

# Proceedings of the 1998 International Computational Accelerator Physics Conference



Monterey, California, USA ▼ September 14-18, 1998

## Proceedings Editors

**Kwok Ko**

*Stanford Linear Accelerator Center  
Stanford, CA*

**Robert Ryne**

*Lawrence Berkeley Nat'l Laboratory  
Berkeley, CA*

Organized by Stanford Linear Accelerator Center (SLAC), Los Alamos National Laboratory (LANL),  
and National Energy Research Scientific Computing Center (NERSC) with support from the  
U.S. Department of Energy

# Contents

Preface .....	vii
Organization .....	ix

## Advanced Accelerator Design and Control

<b>FTU02</b> Experience with the CLASSIC Library for MAD Version 9 .....	1
<i>F. C. Iselin, CERN</i>	
<b>FTU03</b> LEGO - A Class Library for Accelerator Design and Simulation.....	6
<i>Y. Cai, SLAC</i>	
<b>FTU05</b> The Accelerator Description Exchange Format.....	11
<i>N. Malitsky, Cornell</i>	
<b>FTU06</b> Matlab-like Environment for Accelerator Modeling and Simulation .....	16
<i>H. Nishimura, LBNL</i>	
<b>FTU09</b> A Simple Real-Time Beam Tuning Program for the KEKB Injector Linac .....	20
<i>J. W. Flanagan, KEK</i>	
<b>FTU10</b> A Universal Postprocessing Toolkit for Accelerator Simulation and Data Analysis.....	23
<i>M. Borland, ANL</i>	

## Modeling of High-Intensity Beam Phenomena

<b>CTU01</b> Models and Simulation of Beam Halo Dynamics in High-Power Proton Linacs .....	28
<i>T. P. Wangler, LANL</i>	
<b>CTU03</b> A Multigrid-Based Approach to Modeling a High-Current Superconducting Linac for Waste Transmutation.....	33
<i>P. Pierini, INFN</i>	

<b>CTU04</b>	Simulation and Observation of the Long Time Evolution of the Longitudinal Instability in a Storage Ring .....	38
	<i>O. Boine-Frankenheim, GSI</i>	
<b>CTU05</b>	Analytical Model of Halo Formation.....	43
	<i>R. L. Gluckstern, U Maryland</i>	
<b>CTU07</b>	Halo Formation in 3D Bunches with Self-Consistent Stationary Distributions .....	45
	<i>A. V. Fedotov, U Maryland</i>	
<b>CTU08</b>	New Methods in WARP .....	50
	<i>D. P. Grote, LLNL</i>	
<b>CTU09</b>	Unique Features of the University of Maryland Electron Ring and the Necessity of PIC Code Simulation .....	55
	<i>R. A. Kishak, U. Maryland</i>	

### Advances in Electromagnetic Modeling

<b>FWE14</b>	On Solving Maxwellian Eigenvalue Problems for Accelerating Cavities .....	60
	<i>P. Arbenz, ETH-Zurich</i>	
<b>FWE15</b>	Improved GdfidL with Generalized Diagonal Fillings and Reduced Memory and CPU Requirements.....	65
	<i>W. Bruns, TU-Berlin</i>	

### Advanced Algorithms and Methods

<b>FWE21</b>	Self-Consistent Model for the Beams in Accelerators .....	69
	<i>A. Novokhatski, TU-Darmstadt</i>	
<b>FWE25</b>	Using Symbolic Algebra for the Generation of Orbit Simulation Codes from Hamiltonians .....	74
	<i>A. Adelman, PSI</i>	

### New Software Environments and Libraries

<b>CWE13</b>	The Framework of United Accelerator Libraries .....	79
	<i>N. Matlitsky, Cornell</i>	
<b>CWE15</b>	JPP: A Particle Tracking Code with Java .....	82
	<i>R. Hajima, U Tokyo</i>	
<b>CWE16</b>	Implementation of Object-Oriented Design with Fortran Language in Beam Dynamics Studies .....	87
	<i>J. Qiang, LANL</i>	

### Graphic Interfaces and Advanced Visualization

<b>CWE21</b>	An Application Framework and Intelligent Graphic User Interface for Multiple Accelerator Codes .....	92
	<i>B. W. Hill, Gillespie Assoc</i>	

**CWE25** Measurements and Visualization of the Transverse Phase-Space Topology at LEP ..... 97  
*G. Morpurgo, CERN*

**CWE26** Using 3-D Perspectives as a Visualization Tool for Phase Space Data ..... 102  
*G. H. Gillespie, Gillespie Assoc*

### **Status of Computational Accelerator Physics**

**STH04** New Method for Klystron Modeling ..... 107  
*Y. H. Chin, KEK*

### **Modeling Advanced Accelerator Concepts**

**FTH12** Numerical Studies of Wake Excitation in Plasma Channels..... 112  
*B. A. Shadwick, LBNL*

### **Beam-Beam Simulations**

**FTH22** Beam-Beam Simulations with the Gaussian Code TRS..... 117  
*M. A. Furman, LBNL*

**FTH23** ODYSSEUS: A Dynamic Strong-Strong Beam-Beam Simulation for Storage Rings ..... 122  
*E. B. Anderson, Cornell*

**FTH24** Beam-Beam Simulations with GUINEA-PIG ..... 127  
*D. Schulte, CERN*

### **Computing Wakefield Effects**

**CTH11** Transitions Dynamics of the Wakefields of Ultra Short Bunches ..... 132  
*A. Novokhatski, TU-Darmstadt*

**CTH12** Calculations of the Short-Range Longitudinal Wakefields in the NLC Linac ..... 137  
*K. L. F. Bane, SLAC*

**CTH13** The Surface Roughness Wakefield Effect..... 140  
*A. Novokhatski, TU-Darmstadt*

**CTH14** Wake of a Rough Beam Wall Surface ..... 145  
*K. L. F. Bane, SLAC*

### **Particle and Field Simulations**

**CTH23** Calculations of Impedance for Multiple Waveguide Junction Using Scattering  
Matrix Formulation ..... 149  
*V. Dolgashev, Novosibirsk*

**CTH24** The 30 GHz Transfer Structure for the CLIC Study ..... 154  
*G. Carron, CERN*

**CTH25** Field Flatness and Non-Stationary Behaviour of the 4\*7-Cell-Tesla-Superstructure ..... 159  
*H. W. Glock, U Rostock-Germany*

**Posters**

<b>LTU02</b>	Multi-Platform Graphic User Interface for the Marylie Charged Particle Beam Transport Code .....	164
	<i>G. H. Gillespie, Gillespie Assoc</i>	
<b>LTU04</b>	The CERN/SL XDataviewer: An Interactive Graphical Tool for Data Visualization and Editing .....	169
	<i>G. Morpurgo, CERN</i>	
<b>LTU06</b>	Techniques for Robust Nonlinear Delta-F Simulations of Beams .....	174
	<i>A. Friedman, LLNL</i>	
<b>LTU07</b>	Dynamic Space Charge Calculations for High Intensity Beams in Rings .....	179
	<i>J. A. Holmes, ORNL</i>	
<b>LTU10</b>	Map Computation from Magnetic Field Data and Application to the LHC High-Gradient Quadrupoles .....	184
	<i>M. Venturini, U Maryland</i>	
<b>LTU11</b>	Functional Dependence, Broad-Band Fitting, and Ancillary Conditions.....	189
	<i>D. C. Carey, FNAL</i>	
<b>LTU13</b>	Transient Magnetic Field Analysis for the Localization of Electrical Faults in Superconducting Collared Coils .....	192
	<i>P. Komorowski, CERN</i>	
<b>LTU16</b>	Analysis of Emittance Growth and Complex Impedance for Coherent Synchrotron Radiation Shielded by Two Parallel Plates .....	197
	<i>R. Hajima, U Tokyo</i>	
<b>LTU17</b>	Calculations of Eigenmodes in Superconducting Cavities on APE-Supercomputers Using a Software Interface to MAFIA .....	201
	<i>F. Neugebauer, DESY</i>	
<b>LTU19</b>	Investigation of the Impedance and Higher Order Mode Losses for Proposed Beam Pipe Configurations for the Hera Luminosity Upgrade Project.....	204
	<i>M Dohlus, DESY</i>	
<b>LTU20</b>	Design of 2x2 DLDS RF Components for JLC .....	209
	<i>J. Q. Wang, KEK</i>	
<b>LTU21</b>	Electromagnetic Modeling of Fast Beam Chopper for SNS Project .....	213
	<i>S. S. Kurennoy, LANL</i>	
<b>LTU28</b>	Progress Parallelizing XOOPIC .....	218
	<i>P. J. Mardahl, UC Berkeley</i>	
<b>LTU30</b>	Thermal-Photon and Residual-Gas Scattering in the NLC Beam Delivery .....	222
	<i>I. Reichel, SLAC</i>	
	<b>Participants</b> .....	227
	<b>Author Index</b> .....	229

# Preface

The 1998 International Computational Accelerator Physics Conference, ICAP'98, was held September 14-18, 1998, at the Monterey Convention Center in Monterey, California. The meeting continued the tradition of four earlier Computational Accelerator Physics conferences, held in La Jolla, California (1988), Los Alamos, New Mexico (1990), Pleasanton, California (1993), and Williamsburg, Virginia (1996). ICAP'98 was organized by the Stanford Linear Accelerator Center (SLAC), Los Alamos National Laboratory (LANL), and the National Energy Research Scientific Computing Center (NERSC), and received support from the U.S. Department of Energy.

The aim of this conference series is to discuss recent advances in Computational Accelerator Physics and their applications to solving important problems related to present and future accelerators. The ICAP'98 program included plenary sessions on modeling challenges in next-generation accelerators, state-of-the art methods in computational accelerator physics, and a view of what high-end computing would be like in the 21st century. There were also focused sessions on accelerator design and control, modeling high intensity beam phenomena, electromagnetic modeling, advanced algorithms, software environments and libraries, visualization and graphic interfaces, modeling advanced accelerator concepts, modeling the beam-beam interaction, computing wakefields, and combined particle/field simulations. In addition to these oral sessions there was a poster session and a vendor display. Prior to the start of the technical program, a half-day tutorial on High Performance Computing was hosted by NERSC.

The ICAP series recognizes that computing and, in particular, advanced computing, is playing an increasingly important role in the successful design and development of major accelerator facilities, and in research and development leading to advances in accelerator technology. We expect that this new emphasis on high-end computing will continue in future ICAP conferences.

As the conference co-chairs, we would like to extend our appreciation to all the participants. We would also like to thank those persons who helped organize and execute the meeting, including members of the advisory and program committees, the conference administrator, Ms. Eleanor Mitchell and her staff, and those responsible for the conference computing facilities. Finally, we owe special thanks to Sharon West and Thomas Knight for the publication of the conference proceedings.

Kwok Ko and Robert Ryne  
ICAP'98 Co-Chairs

# Committees

## Advisory

P. Colestock (FNAL)  
A. Dragt (U. of Maryland)  
W. Herrmannsfeldt (SLAC)  
I. Hoffmann (GSI)  
T. Kitchens (U.S. DOE)  
S. Myers (CERN)  
G. Peters (U.S. DOE)  
R. Ruth (SLAC)  
H. Simon (NERSC)  
Y. Yamazaki (KEK)

## Program

K. Ko (Co-chair, SLAC)  
R. Ryne (Co-chair, LANL)  
Y. Chin (KEK)  
A. Friedman (LLNL)  
G. Gillespie (Gillespie Assoc.)  
J. Irwin (SLAC)  
T. Katsouleas (USC)  
J. Lagniel (LNS)  
J. Petillo (SAIC)  
J. van Zeijts (TJNAF)

## Conference

E. Mitchell (Administrator, SLAC)  
J. Irwin (SLAC)  
K. Ko (SLAC)  
R. Ryne (LANL)  
V. Srinivas (SLAC)  
IJ. Alessi (BNL)  
N. Angert (GSI)  
B. Aune (CEA, DSM/DAPNIA)  
G. Dutto (TRIUMF)

# EXPERIENCE WITH THE CLASSIC LIBRARY IN MAD VERSION 9

F. Christoph Iselin\*  
CERN, SL/AP group, CH-1211 Geneva, Switzerland

## Abstract

The CLASSIC library is a C++ class library which provides services for building portable accelerator models and algorithms for their analysis. This paper describes the motivations behind the CLASSIC library and its main features. It shows how this library can be used in a large accelerator design program like the new version 9 of MAD written in C++. The possibilities are illustrated by presenting some new developments in MAD version 9, like sophisticated matching features with simultaneous matching of two rings.

The major part of the CLASSIC library is now implemented. Its source code and some preliminary documentation are available from the author.

## 1 INTRODUCTION

The complete design of an accelerator usually requires many different problems to be solved, and to do so, several programs to be run in succession. This implies that accelerator description data must to be sent back and forth between programs, which is complex and error prone due to unnecessary differences in input formats. Data exchange could be avoided if one could pull an algorithm out of one program and plug it into another. However, most of the time this is impossible, since the internal data structures vary wildly between programs.

The format problem was partially solved in 1984, when a “Standard Input Language” was proposed. Nevertheless the data structure problem remained with all problems it causes. To alleviate this problems John Irwin (SLAC) proposed the development of a C++ class library called CLASSIC (Class Library for Accelerator System SIMulation and Control). The project was started during a workshop held in summer 1995 at SLAC as an international collaboration.

Section 2 summarises the history of the CLASSIC project. Section 3 describes the features and use of the CLASSIC library, and Section 4 outlines a few new features in MAD which became possible thanks to the CLASSIC library.

## 2 HISTORY OF CLASSIC

The CLASSIC library was initially started by a large collaboration. Many persons promised to participate, but many of them have not been able to keep their promises due to other commitments. The most important contributions were made by (in alphabetical order)

**Scott Berg** (SLAC/CERN/Indiana University),

**Yunhai Cai** (SLAC),

**Alex Dragt** (University of Maryland),

**James Holt** (FNAL),

**John Irwin** (SLAC),

**Christoph Iselin** (CERN),

**Leo Michelotti** (FNAL),

**Nicolas Walker** (DESY),

**Yiton Yan** (SLAC),

**Johannes van Zeijts** (TJNAF).

During the years 1995 and 1996 these persons met in several workshops to define a structure for the CLASSIC class library. The result was first presented in 1996 [2]. Since then most of the library has been implemented, and is now available from the author.

## 3 DESCRIPTION OF CLASSIC

### 3.1 Terminology

Throughout this description the following terms are used:

**Component:** A single object occurring in an accelerator, like a magnet or a drift space.

**Beam line:** A sequence of accelerator elements. Beam lines can be nested to any depth.

**Element:** Anything that can occur in an accelerator. This can be a single component, or an arbitrary sequence of components.

### 3.2 Structure of CLASSIC

The CLASSIC library consists of several groups of classes called class categories. Each class category provides the user with a logically distinct service. 14 class categories are now completed:

**AbsBeamline:** This category contains a set of abstract classes which define the interface of single components, like drift spaces, magnets, electrostatic elements, RF cavities, etc. The implementations can be found in the category `BeamlineCore`.

---

\*email: [chris.iselin@cern.ch](mailto:chris.iselin@cern.ch)

**BeamlineCore:** This category contains the classes providing concrete representations of components. The instance objects of these classes contain all data needed for performing computations on these components. This category could easily be replaced by another one which provides a different representation. For example, a representation might implement a direct interface to a data base. The separation between `AbstractBeamline` and `BeamlineCore` makes it possible to decouple algorithms completely from component classes.

**ComponentWrappers:** Each “Wrapper” component inherits its interface from the corresponding abstract element. It uses the “Decorator” pattern [3] to attach field changes, like random field imperfections and/or field scaling to a component.

**Beamlines:** The template classes in this category represent the structure of beam lines. Some of them also allow to store arbitrary data in all positions in a structure. In particular they provide mechanisms for attaching misalignments to any element (be it a component of a beam line). They also permit to build tables such as tables of optical functions or of survey data.

**BeamlineGeometry:** This category contains representations for all different geometries which may occur in a component or in a beam line. Each element (component or beam line) contains a geometry object, which defines the design orbit of the accelerator. Classes exist which model geometries comprising Cartesian coordinates (straight geometry), mid-plane geometry (sector bend like), and many more.

**Fields:** This category includes all magnetic and electric field types which can exist in components (dipole, multipole, RF field, etc.).

**Construction:** This category implements the “Factory” pattern [3]. A `ElementFactory` object can construct components and beam lines. It can be called directly by the user, but it is also used in the language parser for constructing components.

**Channels:** A `Channel` implements the “Proxy” pattern [3]. It controls access to arbitrary attributes of components, and may be used in a matching process to provide uniform access to all matching variables and constraints.

**Algorithms:** The `Algorithm` classes implement the “Visitor” pattern [3]. A “Visitor” can be thought of as an algorithm object which walks through a beam line and applies itself to each component in turn. Presently `Algorithms` includes a `Surveyor`, which computes the global geometry, and several mapper algorithms which accumulate linear or non-linear transfer maps using various algorithms. `Algorithms` also

contains a few special integrators which can be attached to a component or beam line.

**Algebra:** Here one finds templates for abstract data types like `Vector`, `Matrix`, `Tps` (Truncated Power Series), `Vps` (Vector Power Series), and maps, all with variable dimensions. Their template parameter selects the data type from which they are built (`int`, `float`, `double`, or even `Tps`). There are also methods for finding fixed points and non-resonant normal form analysis. Further algorithms shall be added in future.

**MemoryManagement:** A pair of classes implementing reference-counted shared objects.

**Parser:** This category contains a simple parser for the Standard Input Language (SIL) [1]. It constructs an accelerator structure from a SIL input file. For this purpose it uses the class `ElementFactory` from class category `Construction`.

**Physics:** The class `Physics` defines a name space defining mathematical and physical constants.

**Utilities:** This category contains a random generator as described by Knuth [5] and a collection of exception objects. The exception objects are used throughout the other categories to handle errors.

### 3.3 Use of CLASSIC

First, an accelerator structure must be built using the parser included with the library, or by defining accelerator components as C++ objects and combining them by executing a C++ program. The resulting structure may contain beam lines and sub-lines nested to arbitrary depth. Within the structure the magnetic and electric fields are represented by classes all derived from a single base class. This permits to treat them all uniformly. The geometry of each accelerator element is described by its contained geometry object.

Next, the user may generate random or systematic misalignment errors for selected elements (components or whole beam lines). Selected components can have field modifiers (random or systematic errors, or scaling) attached.

Special integrators can be defined for any element. When it makes sense for a given algorithm, such an integrator replaces the normal function of the algorithm. An integrator could e. g. use a precomputed map for a complex component or for a complete beam line. Another integrator example automatically splits a component into several thin lenses.

Once the structure is built, the user can apply an algorithm, a C++ object which encapsulates the physics computations to be done. An algorithm can perform any computation on the beam line, such as tabulation of lattice functions. An algorithm is sent to a beam line as a “Visitor” pattern [3]. Both the beam line representation and the algorithm are implemented by abstract classes. This efficiently

hides the implementation details and greatly reduces coupling between classes.

Matching of an accelerator structure is made simple by Channel objects. Once defined, these allow uniform direct access to arbitrary attributes of components. By building an array of channels for access to the variable parameters, and another array of channels for access to the constrained values, the matching process becomes a simple algebraic problem, like

```
// Define an algorithm object.
Mapper visitor(...);
Vector<Channel>variables;
Vector<Channel>constraints;
// Fill in the channels for
// variables and constraints.
...
// Assign new values to the variables.
variables = ...;
// Compute the new constraints.
visitor.execute();
// Extract the new constraint values.
... = constraints;
```

This example is by no means complete, but it should give the flavour of the method.

### 3.4 State of CLASSIC

Since CLASSIC was presented in 1996 as a project [2], the major parts of the library have been completed. The library has been used successfully in a new version of the MAD program, and many minor changes were made to make its use easier. The library compiles cleanly with the latest version of the egcs compiler (1.1b). It does however not yet compile with a purely standard-conforming compiler, since for the namespace features have been omitted. This point will be corrected soon. The CLASSIC source code is available from the author [6] under the same conditions as the CERN program library.

Even though the majority of the library is now complete, some important algorithms, as well as the planned interface to the control system are still missing. Development goes onto fill these gaps.

For the time being the documentation uses doc++-like comments [4]. The resulting documentation is available on the world-wide web [7].

### 3.5 Timing Benchmarks

Most of the CPU time is usually spent for handling truncated power series. Benchmarks have shown that for moderately high orders (about 6) the methods of CLASSIC are comparable in CPU time usage to Berz's FORTRAN routines [8].

For linear maps it turns out that a large fraction of the CPU time is used for memory allocation and deallocation, but studies are under way to overcome this problem. A

linear algebra package and a differential algebra package with fixed dimensions will possibly be added for speed.

### 3.6 Future Plans

In near future, many more algorithms will be defined and added to the library. Some of these will be lifted from MAD, once they are sufficiently tested.

Normal form analysis is now only possible in the non-resonant case. The resonant case will be added shortly.

The CLASSIC structure is now more or less frozen. The documentation can thus be rewritten in a more digestible form. It will include a user's guide and a reference manual.

## 4 ADVANCED FEATURES IN MAD

The CLASSIC library made it possible to implement several advanced features in the new C++ version of MAD. These features are mainly aimed at the design of a collider with two separate rings with two-in-one magnets like the LHC machine, for which many particular problems arise.

- The machine has a global coordinate system, whose arc length  $s$  coincides for both rings, even though one of the beam runs in opposite direction towards decreasing  $s$ .
- Many components in an interaction region are common to both rings. Obviously for physical consistency their order must agree for both rings. This suggests that they are defined by the same sequence definition. The common parts are however traversed in opposite sense by the two beams. This changes the sign of the magnetic forces.
- The layout of the two rings are strongly coupled to each other due to the two-in-one design of the magnets.
- The twin-bore magnets cause strong correlations between the field imperfection in either aperture.
- The small apertures and the lack of space impose very tight matching conditions.
- For the LHC machine, the two rings must fit the existing LEP tunnel. Many present-day accelerators are constrained by similar types of constraints.
- When matching parts of the machine, e. g. interaction regions or arc cells, all algorithms should see consistent imperfections whenever the same accelerator components are seen in different operations using different ranges of the machine.
- Interaction regions must be adjusted to match arc cells at either end, the most efficient method requires backward tracking of lattice functions in the downstream cells. This implies sign changes in the algorithms, which are different from those induced by a beam running in backward direction.

- In order to fulfil all constraints in an optimal way, it is desirable to match both rings in the same time, together with their geometric layout.

#### 4.1 Optical Tables

For efficient matching MAD-9 implements the concept of tables. Their interaction with other objects is shown schematically in Fig. 1. These tables can be used for matching as described below.

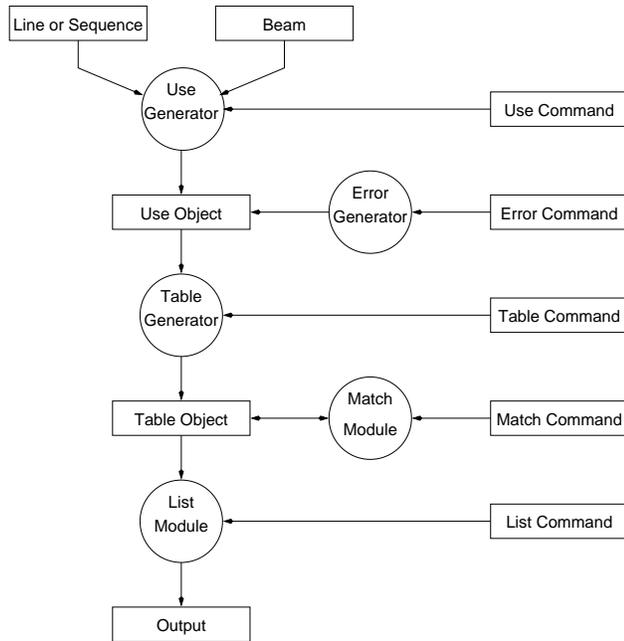


Figure 1: Schematic view of the Interaction between Objects in MAD

The use generator creates a use object; this is a fully instantiated copy of the input beam line, where all components are made unique. The use object also has an attached beam, defining the particles running through the line and their reference energy. The error generator generates misalignment and multipole field errors as desired and attaches them to the use object. The table generator pulls out a flat representation of a selected range of the use object and fills the attached data slots with data such as lattice functions or survey data. The match module interacts with one or more table objects and adjusts their parameters to obey the matching constraints. Finally the list module extracts a formatted listing from the table object.

At the time of writing the following tables are available:

- Periodic linear lattice functions.
- Linear lattice functions with initial values.
- Geometric layout.

A table containing the transfer maps accumulated from its beginning to the current position is being implemented.

#### 4.2 Matching

In a matching process all global variables and all object attributes can be made variable.

Matching constraints can be arbitrary expressions containing:

- Object attributes (e. g. the field strength in a magnet),
- Global Variables (e. g. the global reference momentum),
- Summary values from a table (e. g. the machine tune),
- Entries from tables (e. g. the  $\beta_x$  value in a selected position),
- Vectors built from the above (e. g. the r.m.s. value of dispersion over a selected range in a table).

Very general matching constraints can be constructed. Several tables can be used in the same matching process. By default they are all recomputed whenever a parameter changes, such as to keep all constraints up to date. If a table is known to remain constant, it can be declared as static. It is then not recomputed to save CPU time.

#### 4.3 Status of MAD-9

A beta test version is completed. It is being used to test the early stages of LHC version 6. Its features include:

- Input of the machine structure in a much more flexible way than in MAD version 8.
- Definition of misalignments and/or field imperfections. The program now accepts all order of multipoles, even for elements which are by definition single multipoles.
- Computation of linear lattice functions.
- Computation of geometric layout.
- Computation of arbitrary order maps with normal form analysis. So far only the non-resonant case can be handled.
- Matching of lattice functions or layout.
- Simultaneous matching of lattice functions and layout.
- Simple tracking of particles.
- Interaction with the DOOM data base [10].

#### 4.4 Future Plans for MAD

Plans for near future include:

- The beam line structures of CLASSIC are set up such as to make it easy to run through a beam line in either direction. The `Visitor` pattern encapsulates the complete state of an algorithm. This makes it easy to keep track of the proper sign changes for all four interesting cases:
  1. Run beam forward and track in the same direction.
  2. Run beam backward and track in the same direction.
  3. Run beam forward and track in the opposite direction.
  4. Run beam backward and track in the opposite direction.

The case (1) is complete, the three other cases are being implemented.

- Beams running in reverse direction through a lattice.
- Tracking of particles or lattice functions in the same direction or in the direction opposite to the beam.
- Tables of accumulated maps.
- The documentation [11] must be completed.

## 5 CONCLUSION

It has been shown that the CLASSIC library design is viable, and that it provides the flexibility needed for implementing very sophisticated features in an accelerator design program. Some research is still needed to make the library more efficient, and more algorithms should be implemented based on CLASSIC. The control system interface, as well as a data base interface should be defined and written. For the latter the CORBA specification might be helpful.

## 6 REFERENCES

- [1] D. C. Carey and F. C. Iselin: 'A Standard Input Language for Particle Beam and Accelerator Computer Programs'. 1984 Snowmass Summer Study.
- [2] F. C. Iselin: 'The CLASSIC Project'. Computational Accelerator Physics Conference 1996, Williamsburg, VA.
- [3] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, 'Design Patterns', New York, Addison-Wesley 1995.
- [4] Malte Zöckler, Roland Wunderling: 'DOC++'. URL: <http://www.zib.de/visual/software/doc++/index.html>.
- [5] D. Knuth: 'The Art of Computer Programming', Vol. 2 'Semi-numerical Algorithms', 2nd edition, Addison-Wesley, 1973.
- [6] F. C. Iselin: 'Classic source code'. URL: <http://wwwslap.cern.ch/~fci/classic/2.0>.
- [7] F. C. Iselin: 'Classic Reference'. URL: <http://wwwslap.cern.ch/~fci/classic/doc>.
- [8] M. Berz: 'Differential Algebra Package'. FORTRAN package available from: Department of Physics and Astronomy and National Super-conducting Cyclotron Laboratory, Michigan State University, MI.
- [9] J. Chen, W. Akers, G. Heyes, D. Wu, and C. Watson: *An Object-Oriented Class Library for Developing Device Control Application, Proceedings ICALEPCS 1995*.
- [10] H. Grote: 'The DOOM Project'. URL: [http://wwwslap.cern.ch/~hansg/doom/ap\\_doom.html](http://wwwslap.cern.ch/~hansg/doom/ap_doom.html).
- [11] F. C. Iselin: 'The MAD Program, Version 9.01 (Methodical Accelerator Design) User's Reference Manual'. URL: [http://wwwslap.cern.ch/~fci/mad/mad9/user\\_guide.html](http://wwwslap.cern.ch/~fci/mad/mad9/user_guide.html).

# LEGO - A Class Library for Accelerator Design and Simulation \*

Yunhai Cai

Stanford Linear Accelerator Center, Stanford University, Stanford, CA 94309 USA

## Abstract

An object-oriented class library of accelerator design and simulation is designed and implemented in a simple and modular fashion. All physics of single-particle dynamics is implemented based on the Hamiltonian in the local frame of the component. Symplectic integrators are used to approximate the integration of the Hamiltonian. A differential algebra class is introduced to extract a Taylor map up to arbitrary order. Analysis of optics is done in the same way both for the linear and non-linear cases. Recently, Monte Carlo simulation of synchrotron radiation has been added into the library. The code was used to design the lattices of the PEP-II and now is used for the commissioning. Some examples of how to use the library will be given.

## 1 INTRODUCTION

There were many accelerator design and simulation codes used for designing lattices for the PEP-II[1] largely due to the complexity of the design. It has been always a dream during the design stage to have one code that can handle everything correctly: purposely off-aligned quadrupole inside a solenoid detector, two beams inside a common quadrupole and non-linear chromatic effects with coupling. It is clear that a code with object-oriented design and implementation is the most natural and powerful approach to handle even more complicated modeling efforts during the commissioning and operation of the machines.

We started to design and implement LEGO three years ago to generate an environment to simulate single charge particle dynamics as a primary goal. The first requirement for the design was that all physics calculation directly related to particles shall be handled in a local coordinate system mounted on the accelerator components. The second requirement was to use differential algebra methods to generate maps and analyze beam dynamics whenever appropriate.

We also wanted any applications developed in this environment to be applied to real accelerators in the same way as a simulated machine. Finally, we tried very hard to make our design as simple and modular as possible.

\* Work supported by the Department of Energy under Contract No. DE-AC03-76SF00515.

## 2 SINGLE PARTICLE DYNAMICS

### 2.1 Hamiltonian

Let's consider a sector bending magnet combined with multipole fields. The Hamiltonian that describes single particle dynamics in a curved coordinate system with curvature  $h$  can be expressed as[2]

$$H = -(1+hx)\sqrt{(1+\delta)^2 - P_x^2 - P_y^2} + hx + \frac{1}{2}h^2x^2 - \frac{e}{p_0}A_s \quad (1)$$

where  $\delta = (p - p_0)/p_0$ ,  $P_x$  and  $P_y$  are momenta normalized by the design momentum  $p_0$ . The designed path length,  $s$ , is used as the independent variable. For simplicity, we assume that the electron is an ultra-relativistic particle and the difference between the path length and time of flight is ignored. Our canonical coordinates are

$$\vec{z} = \begin{pmatrix} x \\ P_x \\ y \\ P_y \\ \delta \\ \tau \end{pmatrix}. \quad (2)$$

Please note that we use  $\delta$  as the fifth component because in many situations we will treat it as a static variable. We also assume that the bending angle matches the curvature  $h$ .  $A_s$  is the longitudinal component of vector potential which can be written as multipole expansion

$$A_s = -Re\left(\sum_{n=1} \frac{1}{n}(b_n + ia_n)(x + iy)^n\right), \quad (3)$$

where  $b_n$  and  $a_n$  are normal and skew components of multipoles respectively. In our convention,  $b_3$  is a normal sextupole. The magnetic field can be computed from the vector potential using  $\vec{B} = \nabla \times \vec{A}$ . The result is

$$B_y + iB_x = \sum_{n=1} (b_n + ia_n)(x + iy)^{n-1}. \quad (4)$$

When a machine is large ( $x \ll 1/h$ ) and the energy of the electron is high ( $P_x \ll 1, P_y \ll 1$ ), we can simplify the Hamiltonian in Eq. 1 by expanding the square root and keeping only the quadratic part. The simplified Hamiltonian is

$$H = \frac{1}{2} \frac{P_x^2 + P_y^2}{(1+\delta)} - (1+hx)\delta + \frac{1}{2}h^2x^2 - \frac{e}{p_0}A_s, \quad (5)$$

where a constant of  $-1$  has been dropped since it will not affect any dynamics. This Hamiltonian describes most elements in a storage accelerator since by selecting different parameters, it can describe a drift, dipole, quadrupole or sextupole. It also contains a dispersion,  $-hx\delta$ , and weak focusing,  $(1/2)h^2x^2$ , generated by the bending magnet.

## 2.2 Symplectic Integrator

The simplified Hamiltonian cannot be solved in its general forms without further approximation because of its non-linearity. We choose symplectic integrators as the technique to solve it approximately. One of the advantages of the symplectic integrator is that symplecticity is preserved in the process of its integration. This property is very important when the long-term stability of particles is the issue of concern. Another advantage is that one can easily obtain a transfer map to an arbitrary order by integrating a truncated power series[3] through the element that contains very high-order multipoles.

The idea of symplectic integrator is very simple. It is based on the observation that although the Hamiltonian as a whole can not be solved, but if we separate it into two parts[4]

$$H = H_0 + H_1, \quad (6)$$

where

$$H_0 = \frac{1}{2} \frac{P_x^2 + P_y^2}{(1 + \delta)}, \quad (7)$$

and

$$H_1 = -(1 + hx)\delta + \frac{1}{2}h^2x^2 - \frac{e}{p_0}A_s, \quad (8)$$

then each of them can be solved exactly.  $H_0$  is a ‘‘drift’’ since it depends only on the momentum and  $H_1$  is a ‘‘kick’’ that depends only the coordinates.

To see how these exact solvable solutions can be used to approximate the integration of the whole Hamiltonian we write the integration process as a Lie operator[5]

$$\vec{z}_{out} = e^{-s:H} \vec{z}_{in}, \quad (9)$$

where  $:H : f = \{H, f\}_{poisson}$  denotes the Lie operation on a function  $f$  using the Poisson bracket.

It can be shown by applying the Cambell-Bake-Hausdorf theorem that

$$e^{-lH} = e^{-\frac{l}{2}H_0} e^{-lH_1} e^{-\frac{l}{2}H_0} + O(l^3), \quad (10)$$

The result can be seen as simply placing the integrated kick at the middle of the drift. It does not depend on the specific form of  $H_0$  or  $H_1$ . This integrator is called second-order symplectic integrator since its residual is third order in the length of the integration.

In fact, we can make a fourth-order integrator[6, 7] by using three kicks and four drifts symmetrically as given by

$$e^{-lH} = e^{-c_1 l H_0} e^{-d_1 l H_1} e^{-c_2 l H_0} e^{-d_2 l H_1} e^{-c_2 l H_0} e^{-d_1 l H_1} e^{-c_1 l H_0} + O(l^5), \quad (11)$$

where  $l$  is the length of integration and

$$c_1 = \frac{1}{2(2 - 2^{\frac{1}{3}})}, c_2 = \frac{1 - 2^{\frac{1}{3}}}{2(2 - 2^{\frac{1}{3}})},$$

$$d_1 = \frac{1}{2 - 2^{\frac{1}{3}}}, d_2 = \frac{-2^{\frac{1}{3}}}{2 - 2^{\frac{1}{3}}}. \quad (12)$$

Please note there are two negative drifts and one negative kick used in the formula. In most cases, if the information of where and how the kicks occurred in the physical space is not needed, it is an excellent approximation. It is often used for strong quadrupoles in interaction regions where strong focusing is required. Also  $2c_1 + 2c_2 = 1$  and  $2d_1 + d_2 = 1$  ensure that the total path length and integrated magnetic strength are kept the same as in the total Hamiltonian.

This process can be continued to construct higher order symplectic integrators[8]. In practice, we slice evenly a magnet into a few segments and then for each segment we select a proper symplectic integrator.

## 2.3 Solvable Solutions

To make this paper self-contained, we list the solutions of some useful integrators explicitly. It is trivial to solve the Hamiltonian equation of a drift. The change of phase vector  $\vec{z}$  after the drift described by the Lie operator  $e^{-lH_0}$  is

$$\Delta \vec{z} = \begin{pmatrix} \frac{lP_x}{1+\delta} \\ 0 \\ \frac{lP_y}{1+\delta} \\ 0 \\ 0 \\ \frac{l}{2(1+\delta)^2}(P_x^2 + P_y^2) \end{pmatrix}. \quad (13)$$

The solution of a kick is also well known. The change of phase vector after passing  $e^{-lH_1}$  is

$$\Delta \vec{z} = \begin{pmatrix} 0 \\ -lb_y + lh\delta - lxh^2 \\ 0 \\ lb_x \\ 0 \\ l(1 + hx) \end{pmatrix}, \quad (14)$$

where  $\vec{b} = (e/p_0)\vec{B}$  is the normalized magnetic fields.

## 3 CORE LIBRARY

The library includes 30,000 lines of C++ codes. All important features have been implemented and tested. The main features in the core includes:

- geometry and survey,

- symplectic integrator,
- synchrotron radiation,
- linear optics,
- element by element tracking,
- non-linear map extraction to arbitrary order,
- non-linear map analysis.

It consists of several inter related modules: They are the parser, beamline, processor, integrator and patch. These modules are designed to be used most effectively as parts of a library. However, they can be used independently as well. For example, a beamline can be constructed directly without using the parser module.

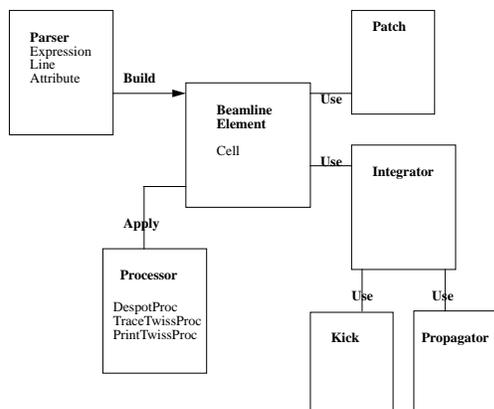


Figure 1: Main Lego Modules

The parser module is for decoding lattice input files. The main function of this module is to read a given input lattice file into tables of parameters, element attributes and symbolic beamlines upon which the beamline with a tree structure for LEGO is built. The module is used to parse many common input formats used in the accelerator community, for instance the MAD input. We will discuss some other formats later in the section on interfaces.

The beamline module is the core of the library. It defines many components commonly used in accelerators and holds places for the integrators and patches required for physics calculation. It also provides the interface and hook for processors to access elements and travel through the tree-structured beamline sequentially. Together with the processor, they form a visitor pattern[9]. This creates a separation between the beamline and its operations. This is a very desirable feature of a library because additional operations on the beamline can be added using a processor without recompiling the core library.

The processor module is the key of the library. A processor uses the hooks provided by the beamline to manipulate the data of elements and beamline. Most data processing performed on elements, integrators or patches is handled by processors. Applications often use processors to interface with the beamline. One of the most important processors in the module sets up the DESPOT integrators for tracking. Actually, we can replace the engine of the computation simply by sending another processor to set up another

type of integrators, for example TEAPOT. Linear and non-linear analysis procedures are unaffected by the swapping of integrators.

The integrator module defines the physics of the beam-transport. An integrator is introduced for the integration of the local Hamiltonian through the body of element including fringe field if needed. The physics of the integrators is outlined in the previous section. Since there are many ways to approximate the integration, the choice of what kind of integrators to use for a given type of element is left for users. In the module, we provide a few processors to set up a consistent set of integrators for instance, DESPOT or TEAPOT. The integrator makes it possible to separate the description of physical components and how they are used in the calculation of physics. This feature is considered to be one of the major achievements of the library.

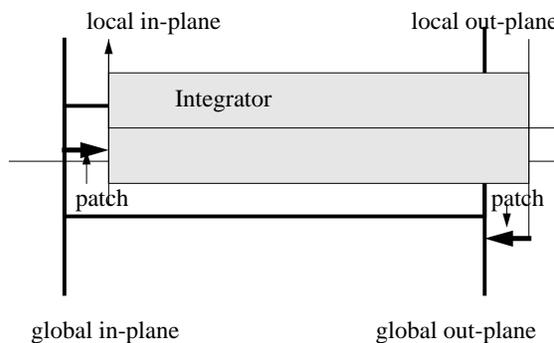


Figure 2: Lego Concept

The patch module handles element or beamline misalignment. Typically, there are two patches for each misaligned component. One is for the entry right before entering the element and another for the exit. Similar to the integrator, the choice of patches are made by users. For instance, a proper selection of the patch allowed us to handle purposely off-aligned quadrupoles inside the solenoid detector.

In addition to these closely related modules, there are many independent modules in the library, such as differential algebra, matrix, vector, geometry, fitting and map modules.

## 4 APPLICATIONS

Building upon the core library, we have written many useful application classes. They are commonly used to evaluate the performance of the machines when many aberrations are present. The main applications are:

- generating alignment and magnetic errors,
- global coupling, tune and chromaticity control,
- three-bumps orbit correction,
- simultaneously correcting orbit and dispersion,
- dynamic aperture search,
- beam-based alignment and optical diagnostics.

Among them, the simplest procedures adjust tunes with two families of quadrupoles and chromaticities with sextupoles. To make a global coupling correction, we implemented a scheme of four families of skew quadrupoles to zero out the four coupled elements in an one-turn matrix.

For control of the closed orbit, we have implemented the widely used three-bump method. Recently, we added a more powerful scheme of correcting orbit and dispersion simultaneously using orbit steering correctors based on eigen-vector decomposition and the MICADO method[10].

Finally, for the commissioning of the high energy ring(HER), we wrote a beam-based alignment package to determine misalignments of quadrupoles and offsets of beam position monitors. The method of analysis is to fit the beam trajectories for several quadrupole configurations differing by a large percentage in strength while treating the circular accelerator as a single-passage beamline.

All the application packages have been simulated for the PEP-II lattice under various conditions. We will show how they are applied to the commissioning of the PEP-II in a later section.

## 5 INTERFACES

In order to use the applications effectively, we wrote many interfaces to the control system of PEP-II and other existing programs. First, in the parser module, we have implemented two builders for decoding MAD input decks and skeleton decks used in the control system. We are defining our own standard input format to accommodate the new types of element allowed in LEGO.

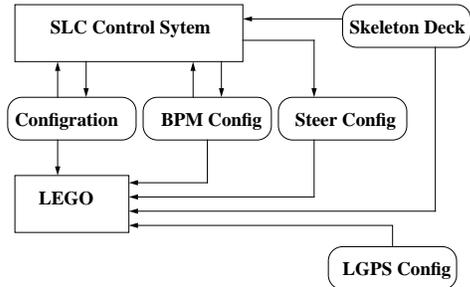


Figure 3: Interface to the control system

Furthermore, we have added the feature of loading configurations and beam position monitor files from the control system to the LEGO beamline so that we can build an off-line model easily in the control room and then apply the application programs to the accelerators. This feature is proven very useful during the commissioning.

## 6 VALIDATIONS

There are many ways to validate a computer program. Common approaches are to compare it to a well known program or an exact analytical solution. For this library, the most numeric tests are performed against its predecessors:

DESPOT and TRACY. The results of the testing are always at the machine precision of the computers when common procedures are applied.

As a design and simulation tool for accelerators, the ultimate validation is against the experimental measurements in the existing machines. During the commissioning of the HER, we made many measurements of the lattices. After a year of commissioning, the recent measurements, including dynamic aperture, are very close to the predictions made by the simulations using this library.

### 6.1 Measurement of Dispersion

Last January, the dispersions of the ring were measured and compared with the simulations. The result of the comparison is shown in Fig. 4. The simulation was carried out as the same way as how the measurement is performed, namely measuring the difference orbit with two different settings of RF frequencies.

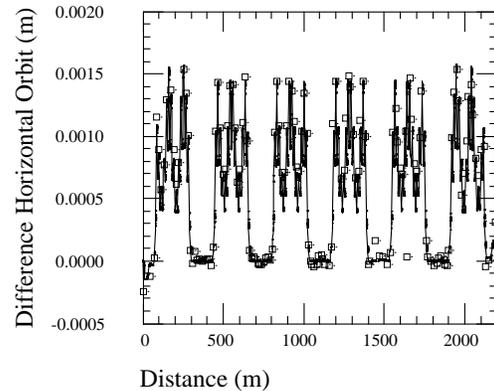


Figure 4: Horizontal dispersion measurement for the HER with -1 kHz change of the RF frequency, the square symbols present the measurement data and solid line for the simulation

### 6.2 Dynamic Aperture Measurement

Last a few years, we routinely perform the dynamic aperture calculation to model the performance of the designed lattices, to specify alignment and magnetic tolerances, and to monitor the field quality of the magnets.

Typically, after alignment and magnetic errors based on the estimations or measurements are introduced into the ideal lattices, we then perform many procedures listed in the section of applications, such as correcting orbit, dispersion, and coupling.

At the end of the correction procedures, the dynamic aperture is determined by tracking the electrons for 1024 turns with  $10\sigma_E$  synchrotron oscillations. A dynamic aperture plot at the injection point is shown on Fig. 5 with the solenoid field of the detector turned off.

At the end of the last July run, we measured the horizontal dynamic aperture by measuring the transverse quantum

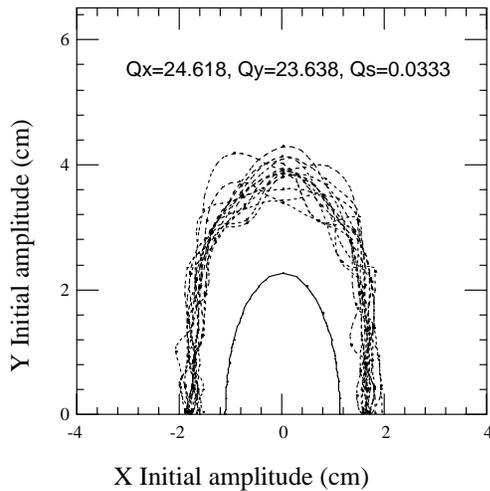


Figure 5: The prediction of the dynamic aperture for the designed lattice with specified alignment and magnetic errors for the HER, the solid line presents 10 sigma of the nominal beam size assuming the vertical emittance is half of the horizontal one and the dotted lines are the dynamic aperture of fifteen randomly generated machines

lifetime after blowing up the horizontal beam size by a factor of two, utilizing the large dependency of the damping partition with respect to the RF frequency[11].

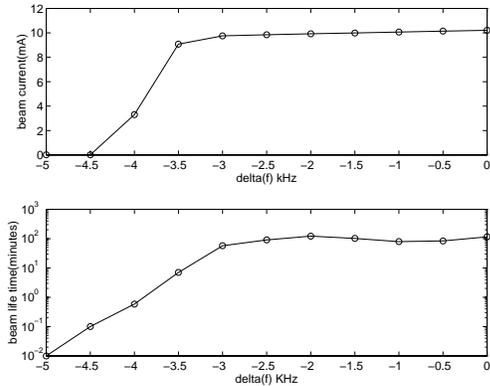


Figure 6: The measurement of the dynamic aperture for the HER

From Fig. 6, we can see that the beam lifetime drops about a factor of two at  $-3.0$  kHz shift of the RF frequency. If we assume that the drop is due to the quantum lifetime of the beam, we conclude that the dynamic aperture is about 5-6 sigma of the enlarged beam. The dynamic aperture with the nominal beam size is estimated from the change of the damping partition due to the change of the frequency. The detailed analysis shows that the dynamic aperture is 13 sigma. That is very close the predicted value shown in Fig. 5 in the horizontal plane.

## 7 SUMMARY

We have created an object-oriented environment for simulating accelerators. It becomes a very efficient tool box to develop new applications both for simulation and operation of accelerators. In this approach, we have achieved five important design specifications. Four of them are related to the modularity of design. They are:

- separation between input language and physical description of element,
- separation between description of element and computational usage of element,
- separation between beamline and its operations,
- separation between analysis of physics and underlining method of transport.

These separations make LEGO very flexible to use and adaptable to challenging design and simulation conditions, like the interaction region of the PEP-II.

The last achievement is the common interface both for simulated machines and real accelerator for all applications.

## 8 ACKNOWLEDGMENT

I would like to thank John Irwin, Martin Donald and Yiton Yan for many useful and exiting conversations from which the ideals of this class library are originated.

## 9 REFERENCES

- [1] "PEP-II: An Asymmetric B Factory," Conceptual Design Report, SLAC-418, June 1993.
- [2] R.D. Ruth, "Single Particle Dynamics in Circular Accelerator," AIP Conference Proceedings 153, Vol.1 p166.
- [3] M. Berz, "Differential Algebra Description of Beam Dynamics to Very High Order," Particle Accel. 24, 109 (1989).
- [4] E. Forest, "Canonical Integrators as Tracking Codes," SSC-138, September 1987.
- [5] A.J. Dragt, in Physics of High-Energy Particle Accelerators, AIP Conf. Proc. No. 87, edited by R.A. Carrigan *et al.* (AIP, New York, 1982)
- [6] R.D. Ruth, "A Canonical Integration Technique," IEEE Trans. Nuclear Science, NS-30, p2669(1983).
- [7] F. Neri, "Lie Algebra and Canonical Integration," Department of Physics, University of Maryland, Preprint(1988).
- [8] H. Yoshida, "Construction of High Order Symplectic Integrators," Phys. Lett. A. Vol. 150 p262 (1990).
- [9] E. Gamma, R. Helm, R. Johnson and J. Vlassides, "Design Patterns, Elements of Reusable Object-Oriented Software," Addison-Wesley Professional Computing Series, 1994
- [10] M. Donald, Y. Cai, H. Shoae and G. White, "An Orbit and Dispersion Correction Scheme for PEP-II," PAC97 proceedings.
- [11] I. Reichel, Ph.D. dissertation, "Study of the Transverse Beam Tails at LEP", PITHA 98/6, July 1998.

# ACCELERATOR DESCRIPTION EXCHANGE FORMAT

Nikolay Malitsky and Richard Talman  
Newman Laboratory, Cornell University, Ithaca, NY 14853

## Abstract

The Accelerator Description eXchange Format (ADXF) is in response to the Iselin-Keil-Talman letter[1], which calls for a new accelerator description standard aimed to provide connectivity between a variety of beam-dynamics programs and heterogeneous data sources. ADXF represents a flat and complete description of the current accelerator state. It has been designed as the additional independent fully-instantiated layer to existing design data structures, in particular the Standard Input Format (SIF). Though the ADXF preserves all SIF element types its open model provides a mechanism for introducing elements with arbitrary attributes. The proposed specification is based on the Extensible Markup Language (XML), an industrial standard for processing Web documents and application-neutral data.

## 1 PREFACE

Basing vertical domain applications on top of framework's reusable components and universal infrastructure is a reliable and effective approach in software development. However, a variety of different views and implementations of accelerator models prevent the direct integration of diverse accelerator programs and comparison of their output results. There is an actual need for a common format to exchange accelerator data between heterogeneous tools and data stores. In 1985 D.Carey and C.Iselin defined a Standard Input Format (SIF[2]) which significantly facilitated the combined usage and development of several accelerator programs. The SIF implementation has been based on the MAD parser and has provided a description of the design model. Changes in computer and accelerator technologies determine new requirements (and solutions) for exchange mechanisms. In January 1998, C.Iselin, E.Keil, and R.Talman issued a letter[1], which called for a new Accelerator Description Standard(ADS) aimed to provide connectivity between a variety of beam dynamics programs and heterogeneous sources. The proposed Accelerator Description eXchange Format is a response to this letter. ADXF represent a flat, complete and independent description of the current accelerator state. The concepts described in this proposal are based on the Standard Input Format (SIF[2]), Standard Machine Format (SMF[3]), Standard eXchange Format (SXF[4]) and experience with actual accelerator applications, such as LHC[5], RHIC[6], CESR[7], FNAL Main Injector, and others.

## 2 RESPONSE TO ADS REQUIREMENTS

This section presents a list of the Iselin-Keil-Talman ADS requirements and their implementation in the ADXF specification.

### 2.1 Backward compatibility with the Standard Input Format

“It should generalize (but not replace) SIF in ways that experience has dictated appropriate.”  
“ADS should mimic SIF where possible, retaining basic accelerator objects and their attributes”

The Standard Input Format focuses on the *design* accelerator description and defines two of its components: *element* and *beam line*. ADXF offers an additional independent layer that deals with the *operational* fully-instantiated accelerator representation. The relationship between SIF and ADXF structures is provided by the references embedded in the ADXF objects. The ADXF format is based on the integration of the MAD *sequence*[2] and SMF *element buckets*[3]. The specification preserves all SIF element types and provides the mechanism for introducing new ones. SIF element attributes have been normalized and grouped in “orthogonal” buckets. The total list of ADXF element types and attributes are presented in Table 1 and Table 2.

“Other feature that have been suggested include: ... nested line preservation from underlying SIF design.”

ADXF represents the accelerator by the flattened tree of accelerator nodes, elements and sequences. Sequences can be nested to an arbitrary depth and may have references to the corresponding design beam lines.

“Minimal completeness. All elements that can influence single particle motion (in their idealized operation) and only those elements are contained.”

The proposed submission contains only SIF elements (see Table 1). However, its open model allows description of arbitrary element types (e.g. muon collider ionization cooling). This ADXF approach is determined by the actual accelerator problems, such as CESR superimposed solenoid and quadrupole elements, LHC and CESR parasitic beam-beam effects, and others.

## 2.2 Extensions of the Standard Input Format.

“Flexibility. Examples are: freedom (but not encouragement) to introduce additional elements or additional attributes to existing elements in a standard (for other purposes ignorable) way.”

Extensibility has been the main criterion of the ADXF design and its implementation approach. The proposed specification allows accelerator description to be extended in two ways: introduction of new element types (e.g. a RHIC helical dipole or a CESR wiggler) and inclusion of new element buckets common for all element types.

“Full-instantiation. Every ring element has its own parameters and may have its own name (laboratory-wide, for example).”

It is provided.

“ ... support for shared lines (such as two rings) “

Each SXF accelerator node, sequence or element, has a unique identifier within the particular accelerator name space, but it may be shared unambiguously by different accelerators (e.g. in interaction regions, injection and extraction systems, and others)

“... provision for definition of “families” of elements”

These families have been initially introduced in the Framework of Unified Accelerator Libraries as communication objects, adjusters and detectors, transported between the accelerator model and correction algorithms. Their specification will be defined after a working prototype of this system has been developed and alternative approaches analysed.

## 2.3 Consistency across different accelerator phase models.

“It should serve from the early phases of conceptual design, through the engineering design and analysis, to the operation of the accelerator.”

The present ADXF version includes only the flat *operational* view of the accelerator; however it provides a mechanism for its integration with the existing site-specific design hierarchical models. This approach stems from the desire to make this format neutral to different conceptual models and adaptable to arbitrary data stores. ADXF can be considered as an additional independent layer to existing design data structures. The interface between the ADXF flat accelerator description and local design models is site-specific and therefore is not a part of this specification.

## 2.4 Separation of the accelerator physics algorithms and approaches.

“Containing only element and lattice description and no beam dynamics, it is to be usable without prejudice by any physical method.”

It is provided.

## 2.5 Separation of technology issues.

“Multiple-realization, in forms optimized for efficient computation (independent of particular computer language), ease of modification, network transmission, database management, and human editing.”

The ADXF conceptual model is described in the modern object-oriented Unified Modeling Language (UML) accepted as an industrial standard by the Object Management Group and other organizations. Expressing ADXF semantics in UML will guarantee multiple-realizations.

## 2.6 Compliance with computer standards.

“It should respect modern computer science standards, especially concerning database management and accessibility over networks.”

The ADXF format is based on the Extensible Markup Language (XML[8]), an industrial standard for processing Web documents and application-neutral data. Its connectivity with disparate data stores (such as ODBMS, RDBMS, file systems, and others) can be implemented using the effective mapping techniques[9] or XML supporting tools.

## 3 ADXF OBJECT MODEL

An accelerator description is a complex system that combines elements with heterogeneous attributes in nested hierarchical structures. The complexity of this organization prompts a variety of different views and implementations of accelerator models. The ADXF is intended to provide a uniform format of accelerator description and interoperability across diverse accelerator applications. This goal determines the major design principles of the ADXF object model. It must be *flat*, *concrete*, and *extendible*.

In most accelerator programs, an accelerator element shares attributes with different elements. These attributes can be inherited from its design prototype and may be stored in global repositories. The choice of an optimal organization is a trade-off between many factors, such as application requirements, software environment, and others. To facilitate mapping diverse accelerator representations into the ADXF format, the ADXF element is described by the *flat* collection of *all* (inherited, global, and local) element attributes. The connectivity with application-specific hierarchical structures is provided by optional references embedded in the accelerator elements. It makes the ADXF

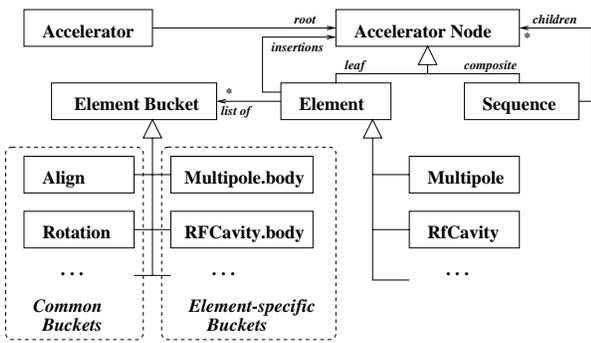


Figure 1: The ADXF object model.

format neutral to different conceptual models and adaptable to arbitrary data stores.

Despite a variety of different element types, accelerator elements have to be organized and processed polymorphically in uniform collections. There are two major approaches for modeling reusable interfaces: inheritance and prototyping. Inheritance is the most natural way of allowing users to change code in object-oriented programming. An architecture based on inheritance has well-defined organization and data types. However, inheritance results in *statically* binding related components. The prototype is a generic class that is able to *dynamically* produce objects with different structure and behavior according to their embedded metadata. This approach is based on the powerful *Reflection* pattern[10] and used in many operation systems and horizontal frameworks, for example, ActiveX Automation, CORBA Dynamic Invocation Interface (DII), and others. For the ADXF object model, an intermediate approach has been chosen. All accelerator element types have a uniform two-level structure: a *dynamic* open collection of *statically* defined buckets with orthogonal fine-grained sets of element attributes. Some buckets belong to particular element types, others can be included in all elements. This approach facilitates selection and classification of well-defined *concrete* data types. On the other hand, it provides a consistent *extendible* mechanism for describing and integrating new accelerator elements and effects. Fig. 1 shows the ADXF object model. Semantics, structure, and the XML declaration of its elements are described in the following sections.

### 3.1 Accelerator

The *accelerator* represents a current accelerator state. It is defined as the root in a hierarchical tree of accelerator nodes, elements and sequences of elements:

```
<!ELEMENT accelerator
  ( sequence | %adxf.element; )* >
```

The accelerator state has several attributes:

id	an accelerator name. The <i>id</i> identifies the accelerator instance and determines the naming context for all its accelerator objects.
version	a version of the current accelerator configuration.
timestamp	a date and a time of the current accelerator configuration.
design	a reference to the design prototype.

In the XML DTD format, these attributes are declared as follows:

```
<!ATTLIST accelerator
  id          CDATA #REQUIRED
  version     CDATA #REQUIRED
  timestamp   CDATA #REQUIRED
  design      CDATA #IMPLIED>
```

Definition of accelerator parameters in the ADXF file is illustrated by the next example:

```
<accelerator
  id = "lhc"
  design = "lhc"
  version = "5.0" timestamp =
  "Mon, 18 Aug 1998 09:27:55">
  ...
</accelerator>
```

### 3.2 Accelerator Node

There are two types of accelerator nodes: sequence and element. They have three common attributes:

id	a name of the accelerator node.
design	a reference to the design prototype. Because the ADXF accelerator description is complete and independent from external sources this link does not affect node parameters and provides only the relationship with site-specific accelerator design models.
at	a longitudinal position of the node entry to the start of the parent sequence. If <i>at</i> is missing, the node is assumed to be butted up against the previous accelerator node.

According to the *Composite* design pattern[11], sequence and element have to be derived from the common class. Since the present XML specification[8] does not support any inheritance mechanism, these attributes have been combined in a sharable XML entity:

```
<!ENTITY % adxf.node.attributes
  "id          CDATA #REQUIRED
  design      CDATA #IMPLIED
  at          CDATA #IMPLIED">
```

### 3.3 Sequence

Sequence is a composite node in an accelerator hierarchical tree. Then its structure repeats the accelerator model, a list of accelerator elements and subsequences:

```
<!ELEMENT sequence
  ( sequence | %adxf.element; )* >
<!ATTLIST sequence
  %adxf.node.attributes; >
```

In contradistinction to the accelerator semantics, each sequence has a parent, and it does not define a separate naming context for its children.

### 3.4 Accelerator Element

Element represents a minimal identified accelerator entity. There are many different element types (such as quadrupole, solenoid, *etc.*). In the object-oriented approach (see Fig. 1), all element types have to be implemented by individual classes derived from the common ancestor Element. It allows one to describe the sequence of heterogeneous elements via a homogeneous collection of Element instances. However, the present XML specification does not support any inheritance mechanism and requires the explicit inclusion of all accelerator types in the accelerator declaration. To satisfy the XML constraints and to provide the ADXF extensions, all element types have been divided into *core* and *extra* categories:

```
<!ENTITY % adxf.element
  "(%adxf.coreElement; |
  %adxf.extraElement;)">
```

The *adxf.coreElement* category contains a list of element types defined in the standardized ADXF DTD file. At this time, it comprises SIF elements[2]:

```
<!ENTITY % adxf.coreElement
  "(marker |
  drift |
  sbend |
  rbend |
  ... )" >
```

Extra elements can be added later in project-specific ADXF extensions. For example, the Wiggler declaration can be included in the local XML document as:

```
<!DOCTYPE accelerator [
  <!ENTITY % adxf.extraElement
    "wiggler" >
  <!ENTITY % adxf.core SYSTEM
    "adxf.dtd">
  %adxf.core;
  ...
]>
```

According to the SMF analysis patterns, all element data are naturally represented via orthogonal buckets of element attributes (such as multipole, aperture, rotation and others). All buckets have a common ancestor and can be stored in a single homogeneous container. As it has mentioned before, the present XML specification does not support any inheritance mechanism. Then the element attributes have been divided into three parts: *frame*, "*element\_type*".*body*, and other buckets sharable by all element types. The *frame* represents the element geometry. The "*element\_type*".*body* bucket can be composed from a mixture of existing and new attributes relevant to the particular element type and recognizable by the corresponding integrators (trackers). The proposed model is able to describe most element types. However, there is a category of accelerator applications using composite elements with superimposed characteristics, such as the intrinsic beam-beam effects, IR quadrupoles installed in the detector solenoid region, and others. To address these tasks, each element type supports addition of an array of element insertions:

```
<!ELEMENT insertions %adxf.element;* >
...
<!ATTLIST quadrupole
  %adxf.node.attributes;>
<!ELEMENT quadrupole (
  frame?,
  quadrupole.body?,
  %adxf.buckets; ,
  insertions?) >
```

The following example illustrates the definition of quadrupole data in the ADXF-compliant document:

```
<quadrupole id = "quad2">
...
  <insertions>
    <multipole id = "entry" at = "0.0">
    </multipole>
    <multipole id = "exit" at = "0.1">
    </multipole>
  </insertions>
</quadrupole>
```

### 3.5 Bucket of Element Attributes

The element bucket encapsulates the minimal set of attributes relevant to the single effect or element feature (such as the magnetic field, aperture, *etc.*). In the ADXF specification, these attributes are explicitly defined in the XML attribute-list constructs. For example, the *frame* bucket is declared as:

```
<!ELEMENT frame EMPTY>
<!ATTLIST frame
  l          CDATA #IMPLIED
  hangle     CDATA #IMPLIED
  vangle     CDATA #IMPLIED
  n          CDATA #IMPLIED>
```

where l, hangle, vangle, and n are the *frame* attributes.

There are two categories of element buckets. Members of the first category, *body* buckets, belong to a particular element type; other buckets can be shared by all elements. Table 1 and Table 2 present the ADXF proposed classification of type-specific and common element attributes, respectively.

The ADXF specification provides a consistent mechanism for integrating new element types. The same approach has been employed for writing new common buckets. These buckets have been split also into *core* and *extra* collections:

```
<!ENTITY % adxf.bucket
  "(%adxf.coreBucket; |
   %adxf.extraBucket;)">
```

The *adxf.coreBucket* collection contains a list of standardized element buckets defined in the ADXF DTD file:

```
<!ENTITY % adxf.coreBucket
  "align | rotation | ... ">
```

The *extra* buckets can be added later in project-specific ADXF extensions.

#### 4 REFERENCES

- [1] F.C.Iselin, E.Keil, R.Talman. 21 January, 1998
- [2] D.C.Carey and F.C.Iselin: *Standard Input Language for Particle Beam and Accelerator Computer Programs*, Snowmass, Colorado, 1984.
- [3] N.Malitsky and R.Talman: *Unified Accelerator Libraries*, AIP 391, Williamsburg, 1996
- [4] F. Pilat *et.al.*: *Standard eXchange Format (SXF) for Accelerator Description*, RHIC/AP/155/, 1998.
- [5] N.Malitsky and R.Talman: *Study of LHC Aperture Dependence on Tune Separation Using Thin Lenses, Phase Trombones, and "Unified Accelerator Libraries"*, LHC Project Note, 1998.
- [6] F.Pilat, S.Tepikian, C.G.Trahern, N.Malitsky: *A model of RHIC using the Unified Accelerator Libraries*, RHIC/AP/146, 1998.
- [7] N.Malitsky and T.Pelaia: *Integration of Unified Accelerator Libraries with CESR*, CBN 98-9, March 1998.
- [8] *Extensible Markup Language (XML), 1.0 WC3 Recommendation*, 10 February 1998, REC-xml-19980210.
- [9] M.Blaha and W.Premerlani: *Object-Oriented Modeling and Design for Database Applications*, Prentice Hall, 1998.
- [10] F.Buschmann, R.Meunier, H.Rohnert, P.Sommerlad, M.Stal: *A System of Patterns*, John Wiley & Sons, 1996.
- [11] E.Gamma, R.Helm, R.Johnson, and J.Vlissides: *Design Patterns: Elements of Reusable Software Architecture*. Addison-Wisley, 1995.

Element Type	Body Attributes	Comments
marker	-	
drift	-	
sbend	kn1	array of normal multipole components
rbend	kt1	array of skew multipole components
	e1	rotation angle for the entrance pole face
	e2	rotation angle for the exit pole face
quadrupole	kn1	array of normal multipole components
sextupole	kt1	array of skew multipole components
octupole		
multipole		
kicker		
hkicker		
vkicker		
solenoid	ks1	solenoid integrated strength
rf cavity	volt	array of RF voltage harmonics
	lag	array of phase lags
	harmon	array of harmonic numbers
elseparator	volt	voltage
hmonitor	-	
vmonitor		
monitor		
ecollimator	-	
rcollimator		

Table 1: Element body attributes

Bucket Type	Attributes	Comments
frame	l hangle vangle n	magnetic length along the design orbit horizontal bend angle vertical bend angle splitting number
align	x y z	offset in the x-direction offset in the y-direction offset in the z-direction
rotation	phi theta psi	rotation around the x-axis rotation around the y-axis rotation around the s-axis
aperture	shape x y	aperture shape horizontal half-aperture vertical half-aperture

Table 2: Common buckets and their attributes.

# Matlab-like Environment for Accelerator Modeling and Simulation\*

Hiroshi Nishimura, Lawrence Berkeley National Laboratory,  
University of California, Berkeley, CA 94720 USA

## Abstract

An interactive GUI and a flexible programmability are equally important for accelerator modeling and simulation studies. However, it is not a trivial task to provide both of them. We discuss the use of Matlab and O-Matrix that is one of the Matlab-like environments on Windows, to obtain a reasonable solution by presenting examples at the ALS.

## 1 INTERACTIVE GUI AND PROGRAMMABILITY

### 1.1 Needs for Programmability

A modern GUI does not always provide flexible programmability for users. This becomes apparent in the case of an accelerator control system that has adopted a modern interactive GUI environment. If the specifications for such a system can cover all the needs of its future users, the software developers would be able to pre-program the required functionality with the support of a GUI. However, it is not realistic to assume such a scenario. Users will find missing functions at the last moment. Therefore, it is essential for an interactive GUI environment to provide some kind of programmability or extendibility to users.

In case of accelerator modeling and simulation studies, a GUI environment is not so popular as for a control system but it rises the same issue. As flexible programmability often becomes more important than a modern GUI, the GUI must cooperate with it.

Since user's algorithms are described by using a kind of language, a modeling program must parse and execute them. There are basically three ways to solve it:

1. Embed a language processing capability in it.
2. Export its application program interface (API).
3. Adopt an existing programmable environment.

The first case is usually realized by providing its own scripting language that is usually a small interpreter. The second case is common for a large scale environment like a machine control system. On the other hand, the third is suitable for a small system like a modeling program.

### 1.2 Experience at ALS

In the ALS[1] design phase, we took the first case to create a simulation program with full programmability. We wrote Tracy[2] by using the Pascal S compiler[3] itself as a framework to access physics library functions but without any graphics capabilities.

During the ALS commissioning phase, we wrote TracyV[4] to simulate the existing ALS storage ring in a modern GUI environment but without programmability. we do not feel too much inconvenience as it is to operate a well-defined virtual machine. At the same time, a C++ class library Goemon[5] was developed and used where a new logic is required.

When the commissioning is completed, Matlab[6] was adopted and has been in heavy use for the ALS machine controls and studies[7]. As its C-like interpreter is suitable for matrix calculations, it carries out orbit control quite effectively. Stimulated by the successful use of Matlab, we started using it for accelerator modeling and simulation studies as described in this paper.

## 2 MATLAB AS A MODELING ENVIRONMENT

### 2.1 Matlab as an Environment

Matlab is widely used on various platforms mainly for numerical calculations. It is an environment with a C-like interpreter that is suitable for matrix calculations with a rich set of library routines that cover mathematics, statistics, optimization and graphics. It also supports external routines that are written in C/C++ by users. Therefore, a user can use Matlab as an environment to manipulate the user's routines.

### 2.2 Matlab for Accelerator Modeling and Simulation

Accelerator optics and orbit calculations are mostly done by matrix and vector manipulations, therefore Matlab fits the needs effectively with various library functions. In addition to it, Matlab makes a lattice definition simple and flexible by providing the list data type, which releases us from writing a lattice parser and allows us to handle the lattice configuration dynamically.

---

\* This work was supported by the Director, Office of Energy Research, Office of Basic Energy Sciences, Material Sciences Division, U. S. Department of Energy, under Contract No. DE-AC03-76SF00098.

TracyML[8] is our first effort of writing an accelerator modeling code using the standard 4x5 matrix formalism entirely in Matlab. Lattice configurations, optics and orbit calculations, and various kinds of fitting and optimization calculations can be flexibly mixed in Matlab programs. Here is an example of defining a beam line in TracyML:

```
D1=drift('D1',0.2);
D2=drift('D2',0.1);
QF=quad('QF',0.3,2.12);
QD=quad('QD',0.1,-2.25);
B=bend('BEND',0.86,3.0,1.5,1.5,0.0);
LINE1=[D1 QF D2];
LINE2=[D2 QD D1];
CELL=[LINE1 B LINE2];
```

As this example shows, the Matlab list structure is appropriate to describe the lattice configuration. A version 5 of Matlab supports user-defined data types and some of the object-oriented features. TracyML uses the Matlab record structure to simplify the support of beam line elements such as magnets. It has been found to be convenient for fitting and optimizing calculations. On the other hand, as it is entirely written in Matlab, the execution speed is not adequate for numerical-intensive tasks like particle tracking. Fig. 1 shows an example of TracyML.

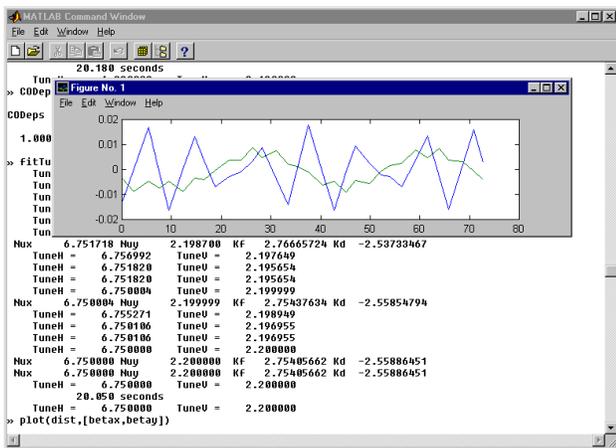


Fig.1 TracyML

### 2.3 Matlab External Calls

When the execution speed becomes a bottleneck in Matlab, we must start using external library calls that are supported by dynamic link libraries (DLL) on Windows. In case of accelerator modeling and simulation, there are two possibilities for the location that holds the virtual machine:

1. Keep the virtual machine in the Matlab layer and use external calls at a low level.
2. Keep the virtual machine in the external DLL layer and use external calls to manipulate it.

As our objective is to speed up the particle tracking speed, we must take the case 2, which leads to the use of a C++ class library Goemon as its external library.

The problem is that Matlab on Windows requires a separate DLL for each individual external call and each DLL carries its own memory space that is inadequate to share the virtual machine object. Possible solutions are, for example:

1. Use one generic external call and dispatch it to multiple Matlab calls.
2. Use one common DLL to interface DLLs for external calls to an external library.
3. Use an other Matlab-like environment that does not carry this kind of problem.

We are in a process of using case 2. At the same time, we found another environment (case 3), O-Matrix[9], and created TracyM[10] to access Goemon library from its Matlab-like environment, as described in the next chapter.

In case of the accelerator controls, we do not have to share the variables in the DLL layer because:

1. The operational context is held in the Matlab layer.
2. The real machine is common to all the external calls.

Therefore, Matlab does not have this kind of problem.

## 3 O-MATRIX AS A MODELING ENVIRONMENT

### 3.1 O-Matrix as an Environment

O-Matrix is a Matlab-like environment available on Windows that allows one DLL to provide multiple external calls. We can effectively wrap multiple C/C++ library routines for use from the environment. Therefore, if we are to hold the context in an external layer for faster execution speed, O-Matrix becomes more suitable than Matlab. On the other hand, one of the demerits of using O-Matrix will be the loss of the data structures that are available in Matlab 5, which leads us to do more in a DLL layer.

### 3.2 TracyM

TracyM is an attempt to obtain both modern GUI environment and programmability for accelerator design, simulation and modeling studies. It is built on top of O-Matrix by using the C++ class library Goemon in a DLL that serves as a physics library. The merits of TracyM over Tracy ML are:

1. Faster execution speed as all the CPU-intensive calculations are done in the DLL.
2. Better C++ Compatibility as it wraps the Goemon C++ class library.

In addition to these merits, it becomes easier to create external GUI tools as described in the next chapter. Here is an example of TracyM calculating the ALS booster ring properties.

```

clear
include TracyM.mat # link to DLL
initTracy
### Define Lattice ###
SYM=marker("")
LBF=drift("LBF",0.496875)
LBD=drift("LBD",0.546875)
LB =drift("LB ",1.05)
### Bending **
BU =bend("BU",0.525,7.5,7.5,0.0, 0.0)
BD =bend("BD",0.525,7.5,0.0,7.5, 0.0)
BB =[BU,BD]
### Quadrupole **
F2 =quad("F2",0.15, 2.7682214)
D2 =quad("D2",0.10,-2.5401249)
### Construct One Superperiod **
DBF =[D2, LBD, BB, LBF, F2]
FBD =[F2, LBF, BB, LBD, D2]
FLD =[F2, LBF, LB, LBD, D2]
DLF =[D2, LBD, LB, LBF, F2]
BL1 =[SYM,DBF,FBD,DBF,FLD,DLF,FBD,DBF,FBD]
### Create a Ring **
ring(BL1,4)

setQforTune(F2,D2)
getTwiss(1)

PrintTwiss
# plot Twiss functions
ginit
x=getVec(0) # S
y=getVec(10) # BetaH
gplot(x,y)
y=getVec(14) # BetaV
gplot(x,y)

calcIntegral # synchrotron integrals

for n=1 to 30 begin
E = n*50*1.0e6
emt=calcEmittance(E)
taux=DampingTimeH(emt)
Emit0=Emittance(emt)
print n*50, taux, Emit0
end

terminateTracy

```

Fig.2 is the result of this program.

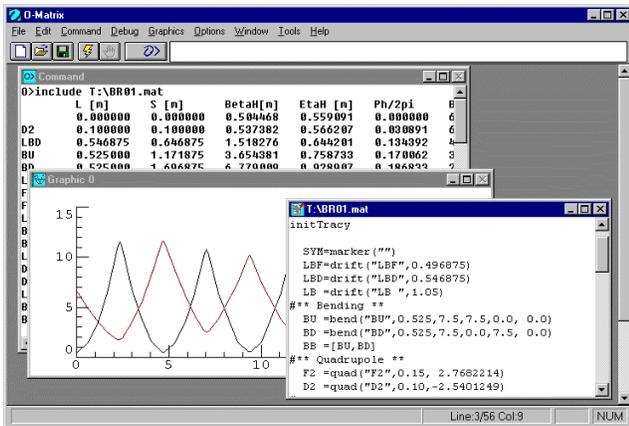


Fig.2 TracyM, Example (1)

Another example is of the dynamic aperture calculation of the ALS storage ring shown in Fig.3. Here is a routine to calculate the dynamic aperture.

```

for j=-20 to 20 begin
for i=-30 to 30 begin
x=i/1000.0
y=j/1000.0
n=quickTurns(512,0.10,x,y)
print i,j,n
if n>=512 then begin
gplot(i,j,"square")
end
end
end
end

```

By using the external routine (quickTurns) we can keep the native speed where numerical calculation becomes intensive.

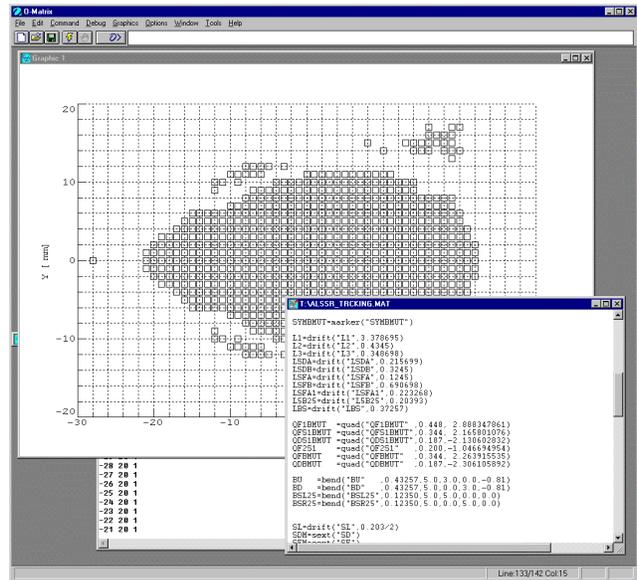


Fig.3 TracyM, Example (2)

## 4 MATLAB-LIKE ENVIRONMENT AND STANDARD GUI

### 4.1 A Matlab-LIKE ENVIRONMENT AND GUI

We have shown the examples of TracyML and TracyM with some graphics displays, but none of them can claim that it has a GUI. It is true that Matlab and O-Matrix do support creation of a GUI with buttons and menus, but we feel more comfortable to have dedicated GUI applications that are natively compiled for interactive use. In the case of accelerator modeling and control, it is possible to prepare a standard set of GUI for common use, such as displaying Twiss functions and closed orbits. For these items, users should not be forced to create event-driven routines every time they need them.

### 4.2 TracyM and its External Tools

From such an aspect, TracyM is accompanied by several external programs that provide standard displays with

GUI. Fig.4 shows two of such programs, Table View and Graph View, displaying the current status of the virtual machine of TracyM by communicating with TracyM through Win32 shared memory.

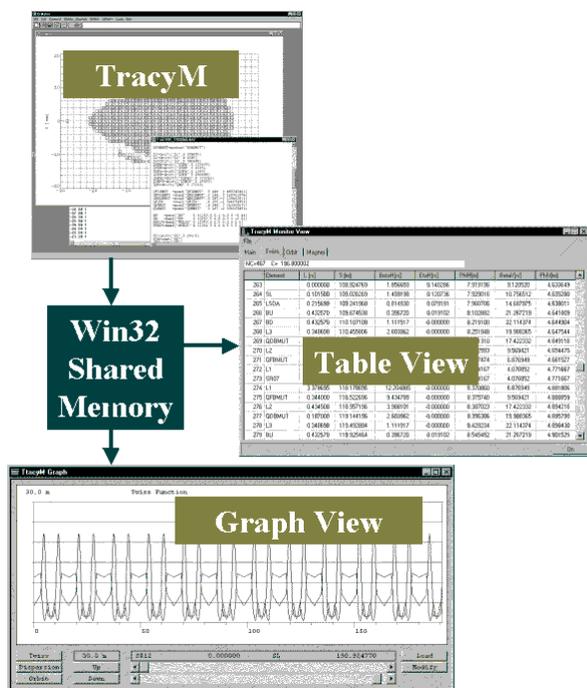


Fig.4 TracyM and External Tools

Another external tool is Command View that manipulates a standard set of knobs of the virtual machine using GUI. In this case, TracyM becomes a server for Command Panel and communicating with it through a Win32 pipe. Although this scheme is still in a development phase, we do see some possibility of creating a GUI application without sacrificing a flexible programmability. In this scheme, the role of TracyM is to create a virtual machine for use from external GUI tools.

## 5 CONCLUSION

A Matlab-like environment such as O-Matrix can serve as a programmable environment for C/C++ libraries that are specialized in various fields. Matlab 5 with user-definable data structure works efficiently for accelerator control, modeling and simulations. Lattice definition becomes simple and dynamic by using the list structure. When the execution speed is required, O-Matrix becomes effective because of its external calling scheme. If we can identify a standard set of displays, separate programs dedicated to them may become advantageous.

## REFERENCES

- [1] "1-2 GeV Synchrotron Radiation Source, Conceptual Design Report," LBL PUB-5172 Rev. LBL,1986. A. Jackson, IEEE 93PAC, 93CH3279-7,1432, 1993.
- [2] H. Nishimura, "TRACY, A Tool for Accelerator Design and Analysis", EPAC 88,803,1989. J.Bengtsson, E.Forest and H.Nishimura, "Tracy Users Manual", unpublished.
- [3] R.E.Berry, "Programming Language Translation", Ellis Horwood Ltd., England, 1981.
- [4] H. Nishimura, " Accelerator Modeling and Control Using Delphi on Windows NT", IWCSMSA96, KEK Proceedings 97-19, 174.
- [5] H. Nishimura, " Taking an Object-Oriented View of Accelerators", IEEE 95PAC, 95CB35843,2162,1996. H. Nishimura, LSAP-153, LBL Internal Report, 1993.
- [6] The Mathworks, Inc. Natic, MA., U.S.
- [7] G. J. Portmann,"ALS Storage Ring Setup & Control Using MATLAB", LBNL ALS/LSAP-248,1990. G. J. Portmann,"Recipe for ALS Storage Ring Operation", LBNL ALS/LSAP-249,1990.
- [8] H. Nishimura,"Introduction to TracyML", LBNL ALS/LSAP-246,1998.
- [9] Harmonic Software, Inc. Seattle, WA., U.S.
- [10] H. Nishimura, "Introduction to TracyM, Part 1", LBNL ALS/LSAP-258,1998.

# A SIMPLE REAL-TIME BEAM TUNING PROGRAM FOR THE KEKB INJECTOR LINAC

J.W. Flanagan, K. Oide, N. Akasaka, A. Enomoto, K. Furukawa,  
T. Kamitani, H. Koiso, Y. Ogawa, S. Ohsawa, T. Suwada,  
KEK, 1-1 Oho, Tsukuba, Ibaraki 305-0801 Japan

## Abstract

A simple real-time beam tuning program has been developed for use at the KEK B-Factory injector linac. The program features the ability to adjust an arbitrary combination of parameters (e.g. magnet currents and RF phases) in order to optimize an arbitrary combination of monitor values (e.g. the ratio of a downstream beam current to gun current). Based on the downhill-simplex method[2][?], it requires no knowledge of beamline details. An additional "persistence" parameter is used to adjust the treatment of pulse-to-pulse variations of monitor values while mapping the parameter-space terrain, and controls the peak-holding performance in the presence of both statistical fluctuations and long-term drift. Preliminary results from the commissioning of the KEKB injector linac and plans for the future are discussed.

## 1 INTRODUCTION

The KEKB injector linac went through the first phase of commissioning from October 1997 to June 1998, following extensive upgrades to the prior facility.[1] During this time several techniques for tuning the beam were employed, such as the introduction of local bumps, energy feedback via klystron phase tuning (using orbital radius in the arc section as a diagnostic), and automatic orbit smoothing with the use of BPMs. These techniques were very successful in achieving the target beam currents. In addition to the above techniques, daily fluctuations in the machine state and the rise of non-linear behavior at higher target currents suggested that a non-model-based optimization tool might prove useful. As a result, a beam feedback tool based on the downhill simplex method was developed.

## 2 SOFTWARE

### 2.1 Environment

Figure 1 shows the software environment in use at KEKB, including the linac and the beam transport line between the linac and the KEKB storage rings. The SADscript environment[4] uses a Mathematica-like syntax with object-oriented extensions and provides easy access to all monitor and control records in the EPICS machine-state database, in addition to the beam optics and tracking functions of SAD. The user interface, shown in Figure 2, is built from a GUI toolkit based on Tcl/Tk.[5] During the Spring commissioning some linac components were accessed via stand-alone commands, though from Fall 1998 all instrumentation controls are based on EPICS, using a uniform

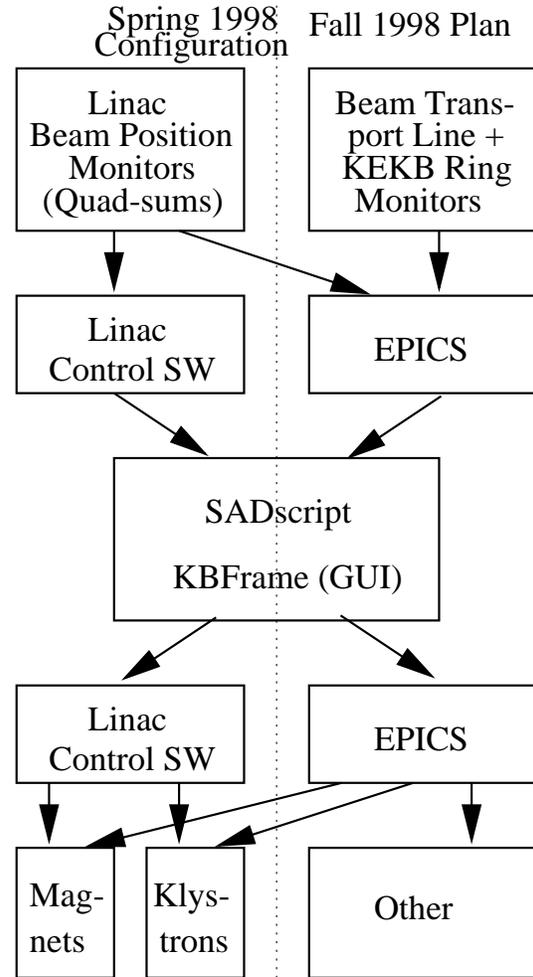


Figure 1: Beam Feedback Software Environment

method of access to monitors and controls and simple extensibility to beam-transport and KEKB ring components.

### 2.2 Downhill Simplex Method

The Downhill Simplex optimization method is well known in the scientific community, being extensively discussed in Reference [2]. Given a function  $F$  of  $N$  parameters, an "amoeba" (simplex) with  $N+1$  "feet" measures the height of the  $N$ -dimensional terrain, and attempts to walk down the slope in search of the bottom of a valley. When the amoeba seems to be near a minimum, it will resize to contract around the suspected minimum. It can also perform other operations depending on the terrain.

The method is quite general, and can find a minimum of

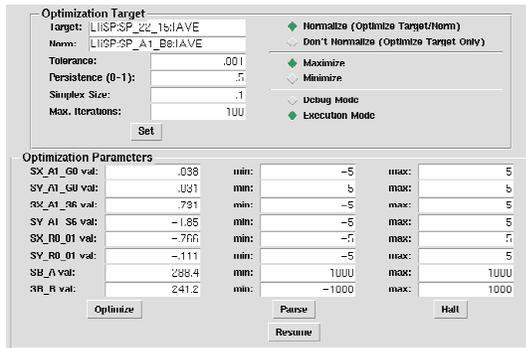


Figure 2: Optimization Tool Control Panel

an arbitrary function in N-dimensions without knowledge of derivatives or any other information about the function. This makes it applicable to cases where the functional dependence on the input parameters is difficult to calculate but relatively constant, making it a good choice as a beam fine-tuning tool. The method was applied to the minimization of vertical emittance during operation of the Tristran ring at KEK.[3] A drawback is that while the downhill-simplex algorithm expects  $F$  to be arbitrary, it does not expect its value at one point in parameter space to fluctuate from measurement to measurement. Our solution to this problem is the introduction of a parameter called “persistence.”

### 2.3 Persistence

There were two goals for the program. One was to fine-tune a beam that had already been tuned using other methods, to correct for non-linear or non-calculable effects remaining. For this purpose, the program is largely operating like a human operator with faster reaction time. The second goal is to maintain a good machine state, continuously adjusting for drifts in machine performance over longer-term intervals.

As mentioned, the simplex consists at any one time of a list of measured values for  $N+1$  locations (feet) in parameter space. Based on the heights of these feet in the terrain, it will attempt to feel out new and better locations and relocate the least-optimally located feet to better positions. In principle, the most reliable measurement for each foot would be the result of averaging over many pulses. For our initial experiments, we used a 10-pulse average at each location. (Such averaged measurements are available in the EPICS database, along with single-shot measurements.) When the amoeba is near a suspected optimum, it will contract around in size around the most optimally placed foot, and continue. At this point, before the contraction, the value of the optimizing function is remeasured at the current optimal location. The value  $F$  at that location is then assigned a weighted average of the old and new values:

$$F = [PF_{old} + (1 - P)F_{new}],$$

where  $0 \leq P \leq 1$  is the persistence. A persistence of 0 means that the algorithm retains no memory of old values, and 1 means that it never remeasures the optimum location twice as long as another foot does not find a better location. The simplex is then resorted to verify that the optimum is still the optimum, or reassign the optimum to a different foot if necessary. This periodic remeasurement only at suspected optima, instead for example of remeasuring all feet, is a time-saving compromise. Also, since each remeasurement of the optimizing function at a point in parameter space requires physically changing the machine parameters to the corresponding state in order to make the measurement, this procedure avoids putting the machine into states that are already believed sub-optimal.

### 2.4 Performance

The performance of the algorithm was tested during the commissioning of the 3.5 GeV positron line. Electrons were accelerated to 1.5 GeV in the first two sectors (“A,” “B”) of the linac, then pass through a 180 deg arc section, and are accelerated to 3.7 GeV by the end of the next two sectors (“C,” “1”), at which point a target is inserted when the linac is in positron mode. The positrons from the target are then accelerated to 3.5 GeV in the remaining 4 sectors (“2”-“5”) for transport to the KEKB low-energy storage ring.

For the case of positron production, the quantity being optimized was the ratio of the beam current downstream of the positron target to the beam current just after the gun. The parameters being varied were one pair of x and y steering magnets just after the gun, another pair after the buncher section, and another pair just at the entrance to the arc section, in order to minimize emittance growth due to transverse wake effects on the beam from injection and beam pipe misalignments. In addition, the RF phases of the first two sectors were varied to minimize the energy spread. The energy acceptance of the arc section is  $\pm 1.5\%$

The results were encouraging, with the highest positron current being recorded for the commissioning period at several percent over previous performance. Typically most of the improvement was seen within the first 50 iterations. The persistence setting was found useful in keeping the machine near an optimum – a persistence of zero led to drift away from optimal settings even after the optimum had been found, and a persistence of one tended to result in the simplex becoming hung up on a false peak due to fluctuations. Figure 3 shows the optimization curve for a persistence setting of 0.5, which shows good peak-holding performance. The speed of the algorithm largely depends on the rates at which the machine state can be changed and the monitor values can be read.

## 3 SUMMARY

A real-time beam optimization tool has been developed for the KEKB injector linac. Performance was found to be enhanced by the introduction of a persistence parameter.

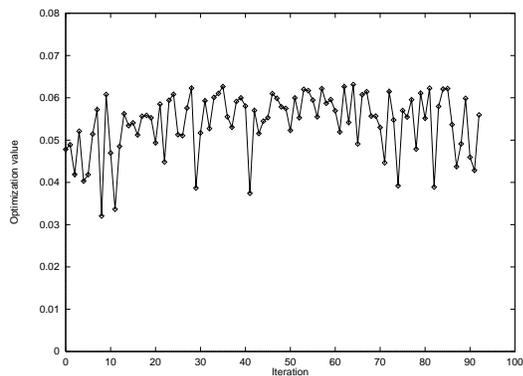


Figure 3: Example Optimization Curve for persistence=0.5, showing the value of the ratio of the current downstream of the positron target to that just after the gun, as a function of iteration.

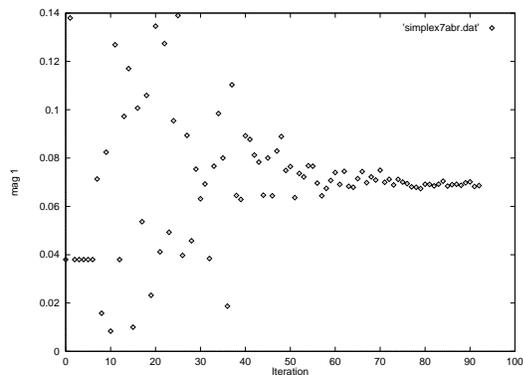


Figure 4: Diagnostic plot of parameter value history during an optimization run. In this example the parameter is the first steering magnet, which stabilizes at a new level by the 70th iteration.

Further testing and extension of applicability to the beam transport line and KEKB rings is underway.

#### 4 ACKNOWLEDGEMENTS

The authors are indebted to the efforts of Dr. Noboru Yamamoto and the KEKB Control Group, who established the EPICS environment at KEK, and Prof. Shin-Ichi Kurokawa for his continual support. Work supported by the Japanese Ministry of Education, Science, Sports and Culture (Monbusho) and the Japan Society for the Promotion of Science.

#### 5 REFERENCES

- [1] Y. Ogawa *et al.*, "Commissioning of the KEKB Linac", KEK Preprint 98-125, and proceedings of Linac98. (1998).
- [2] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, "Numerical Recipes in C", Cambridge University Press, pp. 408-412 (1988). Note that they refer primacy to: J.A. Nelder and R. Mead, *Computer Journal*, vol. 7, pp. 303-313.
- [3] M. Kikuchi, private communication.
- [4] <http://www-acc-theory.kek.jp/SAD/sad.html>

- [5] <http://faraday.kek.jp/KEKBFrame/contents.html>  
Akasaka *et al.*, "Operation Software for Commissioning of KEKB Linac Programmed with SAD," proceedings of APAC98.

# A UNIVERSAL POSTPROCESSING TOOLKIT FOR ACCELERATOR SIMULATION AND DATA ANALYSIS\*

M. Borland, ANL/APS, Bldg 401, 9700 South Cass Avenue, Argonne, IL. USA

## Abstract

The Self-Describing Data Sets (SDDS) toolkit comprises about 70 generally-applicable programs sharing a common data protocol. At the Advanced Photon Source (APS), SDDS performs the vast majority of operational data collection and processing, most data display functions, and many control functions. In addition, a number of accelerator simulation codes use SDDS for all postprocessing and data display. This has three principle advantages: First, simulation codes need not provide customized postprocessing tools, thus simplifying development and maintenance. Second, users can enhance code capabilities without changing the code itself, by adding SDDS-based pre- and post-processing. Third, multiple codes can be used together more easily, by employing SDDS for data transfer and adaptation. Given its broad applicability, the SDDS file protocol is surprisingly simple, making it quite easy for simulations to generate SDDS-compliant data. This paper discusses the philosophy behind SDDS, contrasting it with some recent trends, and outlines the capabilities of the toolkit. The paper also gives examples of using SDDS for accelerator simulation.

## 1 INTRODUCTION

In recent years, we have seen a great increase in the number of accelerator simulation codes. Each code developer faces the same hurdles of developing a user interface and providing tools for the user to view and analyze code output. Many lack the time or expertise to create sophisticated interfaces and postprocessing, and so provide difficult interfaces and little in the way of postprocessing.

In the midst of this, there is no movement toward common standards that permit simulation codes to be used together and share postprocessors. The lack of such a development reinforces the rate of growth, because users often find it easier to write a new code than to work with one or more existing ones.

Simulation codes still rely on an inordinate amount of user interaction and manual work. The time spent by the user to prepare and analyze data from a simulation may exceed the running time by orders of magnitude. A root cause of this is the lack of generally-applicable, powerful postprocessing tools.

In this paper, I describe a system that solves these problems. I will start by discussing common problems with simulation codes. Following this, I describe the structure and advantages of the Self-Describing Data Sets (SDDS) system, which is based on modular tools, self-describing

data, and scripts.

## 2 COMMON SIMULATION CODE PROBLEMS

*Data Storage:* One of the root problems with simulation codes is the wide variation in data storage methods. Much of the output from most codes is user-readable text or graphics. Such output is inherently difficult or impossible to subject to automatic processing, which forces the user to manually transcribe or translate data into another format. Ironically, by providing “user-friendly” text or graphics, the code developer can actually make the sophisticated user’s job much harder.

Some code developers recognize these problems and provide data files intended for automatic processing. However, in many cases, these are for use with a dedicated postprocessor that provides simple, single-purpose graphics and perhaps simple analysis. The user may not be able to count on these files remaining the same between code revisions. Other code developers take the approach of making data files for a specific commercial data analysis or viewing product. Rarely does more than one code use the same approach, let alone use the same data protocol.

Finally, most data files are “fragile,” in that format changes will break programs that use the protocol. For example, a code may output a table of numbers that is intended for automatic processing. A user or the developer may create another code that reads the output of the first. When, however, a new column of numbers is added to the output of the first code, the second code is no longer functional. This creates an unwillingness to upgrade, a multiplication of versions of the same code, and an inability to use codes cooperatively.

*Postprocessing Support:* The time and effort a user spends postprocessing data from a simulation is generally proportional to the number of simulation runs performed. This is largely due to the lack of analysis software that accepts the output of simulation codes directly. This is often true of codes that provide a GUI, since the user of such an interface is expected to guide program actions with the mouse.

The analysis provided by the typical simulation code is often limited and not customizable. The user cannot ask for new computations to be performed upon the code’s existing output data. Graphical output is also generally very specific and predefined. If the user wants something different, he must try to transfer the simulation output to another program.

In a way, this is only sensible. For each developer of each simulation to provide sophisticated graphics and postprocessing for each simulation code would entail an enormous amount of extra work. Hence, it is likely that this state of

---

\* Work supported by U.S. Department of Energy, Office of Basic Energy Sciences, under Contract No. W-31-109-ENG-38.

affairs will continue unless another approach is found.

*User Programmability:* Few codes allow the user to go beyond the “Prepare, Run, Read” or “Click, Run, Look” simulation cycle. For example, there is usually no way to automatically prepare a series of input data sets for a simulation code, followed by automatic collation of the results of the group of runs. This is available with the custom programming environments that some codes provide, but it is unreasonable to expect every code to have such an environment.

What is needed is the ability to treat each simulation code as a computational module that can be embedded within another process (e.g., another program or script). For this to work, the root problem of data storage chaos must first be addressed.

### 3 A SOLUTION

An ideal situation would be the following: 1. All simulation output should be ready-made for machine manipulation and computation. 2. A common set of powerful data processing and display tools should be used for all accelerator simulation codes. 3. The output of any code should be readily used as input to another. 4. It should be possible to enhance and automate the use of a code without changing the code itself.

A system that provides these advantages is available and has been in use at the APS since 1994. The Self-Describing Data Sets [1, 2] system is based on a relatively simple self-describing file protocol. A toolkit is available that comprises about 70 general-purpose programs that use this protocol. There is an additional toolkit of about 20 control-system programs [3, 4].

SDDS has enjoyed wide application at APS, beginning with commissioning of the APS injector and continuing through commissioning of the APS storage ring. It provides computational muscle for most of the high-level applications in use today on the APS controls system, including control, display, and analysis. It is used for almost all experimental data collection and analysis for machine studies. It is also used for much of the accelerator simulation done at APS. In this paper, I will address only the last of these applications.

### 4 SELF-DESCRIBING DATA AND THE SDDS PROTOCOL

The concept of “self-describing data” may be unfamiliar, but it is quite simple. Basically, self-describing data is identified and accessed *by name only*, in contrast with traditional data formats where data is accessed by position (e.g., column number). Positional access methods make for “fragile” data files by requiring one code to “know” the detailed organization of data from another code.

With self-describing files, the code uses the name of the data to request the data values using a subroutine call. The details of where the data is in the file are encoded in the

self-describing file protocol and interpreted by the subroutines that manage such files. As long as the name of the data and the self-describing file protocol are not changed, this system is robust and the data will always be readable no matter what additional data is put in the file.

There are other advantages to using self-describing files. With self-describing files, programs can be generic and operate on *named* data; for example, a graphics program can plot data in general, rather than just beta functions or just phase space diagrams. Self-describing files may also include “meta-data,” such as units, and data type. Using such meta-data makes for more robust programs. For example, codes can check units and data type of data before using it.

An SDDS file consists of three parts: a single line that identifies the file as an SDDS file; an ASCII header consisting of namelist-like entries that define the structure of the data; and zero or more instances of the structure defined by the header. The data itself may be in binary or ASCII format, as declared by the header. The structure defined by the header may contain scalar values, columns forming a table, and arbitrarily dimensioned arrays. The user can access the elements of the structure individually. There are no inherent limits on the number of elements, the length of the tables, or the number of dimensions or size of arrays. Any element may have one of six data types: single- or double-precision float-point, short or long integer, character, or character string.

A familiar example of an SDDS file would be a table of Twiss parameters with corresponding values for the tunes, chromaticities, etc. After four years of experience with SDDS, we’ve found that the vast majority of our data fits this simple model in a natural fashion.

### 5 PROGRAM TOOLKITS

A “toolkit” is a group of programs that can operate on the same type of entity. Such programs can perform the same type of action on numerous entities of the same type. Ideally, any program’s output is usable as input to any other program, so that programs can be used in sequence to process data. This is the chief advantage of such a toolkit. Another advantage is decentralized expansion, since anyone can independently create a tool that operates on the same entities. Such additions *cannot* break existing tools or the applications that use them, in stark contrast to the common situation where a change to one program requires changes to other programs.

The SDDS toolkit [5] is a group of about 70 programs that operate on SDDS files. There are three types of SDDS programs: those that read and write SDDS files, called the “core toolkit”; those that read SDDS files and create non-SDDS data (e.g., text or graphics); and those that read non-SDDS data (e.g., CSV files) and create SDDS data.

Capabilities of the individual programs are many, including graphics, equation evaluation, statistics, histogramming, fitting, interpolation, integration, differentiation, signal processing, smoothing, peakfinding, matrix operations,

and creation of text printouts. Since most programs both read and write SDDS data, many operations can be used sequentially on the same data set. The user can thus create a processing chain that suits his particular needs using the SDDS toolkit programs as modules.

For example, one might want to simply plot some data (using the SDDS program `sddsplot`). However, one might want to FFT the data and plot to FFTs (`sddsfft`, `sddsplot`). Or one might want to FFT the data, smooth the FFTs, find the amplitudes of the peaks, histogram the amplitudes, then plot the histograms (`sddsfft`, `sddsmooth`, `sddspeakfind`, `sddshist`, `sddsplot`). The possible combinations are literally inexhaustible.

Note that creating human-readable output is only a small part of the SDDS toolkit. Graphics and text printouts, while important, should not drive the structure of a simulation code or determine the format of its output. Using SDDS, the simulation developer doesn't need to worry about what kind of graphics or printouts a user might want. SDDS allows the user to produce the graphics and printouts *he* wants.

To give some more explicit examples, suppose that one has an SDDS data file (`data.sdds`, say) containing turn-by-turn particle coordinates from a tracking simulation. Suppose that the data is in a table, with columns called `x`, `px`, `y`, and `py` giving the transverse phase space coordinates, a column called `Turn` giving the turn number, and a column called `delta` giving the fractional momentum offset. (Space does not permit showing the plots from these examples. Examples of plotting output, all directly from the SDDS toolkit and unaltered, are shown in the Figures in following sections.)

- Plot the horizontal phase space:

```
sddsplot -column=x,px -graph=dot \
  data.sdds
```

- Take the FFTs of the horizontal and vertical motion. Plot these together using variable line types on a log scale:

```
sddsfft -column=Turn,x,y \
  -window=hanning data.sdds data.fft \
sddsplot -column=f,FFT* \
  -graph=line,vary -mode=y=log data.fft
```

Note that the FFTs of `x` and `y` are automatically named `FFTx` and `FFTy`. Most of the core toolkit functions this way, providing predictable new names based on the names of the data being processed. In some cases, the user must supply a new name or a template for new names.

- Suppose now that the file `data.sdds` contains multiple pages, with each page corresponding to a separate particle with a different initial momentum offset. The following sequence will give plots of `x` and `y` tune vs momentum:

```
sddsprocess data.sdds -pipe=out \
  -process=delta,first,delta0 \
  | sddsfft -column=Turn,x,y -pipe \
  | sddsprocess -pipe \
  -process=FFTx,max,xTune,posit,func=f \
  -process=FFTy,max,yTune,posit,func=f \
  | sddscollapse -pipe=in data.result
```

```
sddsplot -column=delta0,?Tune \
  -separate data.result
```

While this example looks a little cryptic at first, it is easily understood and is much more terse than doing the same thing in a programming language. First, `sddsprocess` is used to create a parameter (`delta0`) containing the first value of the column `delta`. Second, the `x` and `y` values are FFT'd using `Turn` as the independent variable. Third, `sddsprocess` is used to find the position of the maxima in `FFTx` and `FFTy`, viewed as a function of `f` (the frequency from the FFT); the resulting parameters are called `xTune` and `yTune`. Fourth, `sddscollapse` is used to “throw away” the tabular data and collapse the parameter data across pages to create a new table, containing columns `delta0`, `xTune`, and `yTune`. Finally, the result is plotted.

In order to coordinate toolkit programs and create high-level applications, we often employ a script programming language. SDDS simplifies the development of data processing, while the script language provides an easy interface to the data processing algorithm. Using SDDS is much easier than using a programming language for data processing because the user need not worry about loops, variables, and all the other overhead of programming. SDDS hides all of this from the user and lets the user think in terms of the operations he needs to perform.

## 6 SIMULATION CODES AND SDDS

There are a number of codes that are completely or partially reliant on SDDS for data storage and processing. Among the “fully-compliant” codes are: `elegant` [6], a general-purpose accelerator simulation code; `shower` [7], an EGS4 wrapper for electron-gamma shower simulation; and `spiffe` [6], a fully-electromagnetic PIC code for rf gun simulation. These codes use SDDS for all output and most input. (For input, we use standard format for the accelerator lattice and each code has a custom command-based main input stream.)

As an illustration of the flexibility of SDDS, the code `elegant` uses SDDS for all data except the lattice file and command input. Data include: particle input and output for tracking; turn-by-turn particle data and statistics; FFTs of particle motion; beam moments, transport matrix, and Twiss parameters vs position; coordinates of lost particles; amplification factors; orbits, corrector strengths, and statistics; magnet strength output after tune or chromaticity correction; output of internally-generated error values; input of data for any element parameter; input of kicker waveforms; input of impedances and wake functions.

The three codes noted above have been used cooperatively, a task made much easier by the use of SDDS files. For example, `elegant` can be readily used to track the output of `spiffe` or `shower`. `shower` can be wedged between two `elegant` runs to create a simulation of beam transport and shower creation followed by tracking of shower particles. While `elegant` knows the

names of the data in `spiffe` particle data output and can read it directly, `elegant` doesn't know anything about shower files (which don't use accelerator-type coordinates). Instead an SDDS-based script is used to produce `elegant`-convention 6-D particle data files from shower-convention files. This illustrates how the SDDS toolkit can be used to glue together two disparate programs that use SDDS files but differ in their mathematical convention for describing results. A number of real-world projects have made use of this technique. These include rf gun design and transport line optimization [8] and multilayer positron target design [9].

It isn't necessary for a code to be fully SDDS-compliant in order to reap some of the advantages of SDDS. Several existing accelerator codes (including ABCI [10] and TDA3D [11]) have been modified to create SDDS ASCII files [12]. This is quite easy and gives any code so modified instant access to sophisticated postprocessing and graphics.

For some codes (MAD [13] and MAFIA [14, 15]), we have special-purpose output conversion tools available to convert the code's output into SDDS. For other codes (ACCSIM [16], RACETRACK [17], and GINGER [18]) we can convert to SDDS by prepending an SDDS header to the output data [12].

## 7 EXAMPLES

### 7.1 Enhancing a Code Using SDDS

The code `elegant` performs 6-D canonical tracking of particles, but is unable to compute tune as a function of momentum. It is straightforward to add this capability without modifying `elegant`. The items in parentheses are the names of the SDDS toolkit programs used in each step. The reader may notice a similarity with an example provided above.

1. Prepare input particle coordinates having a range of  $\Delta P/P$  values and small x and y starting values (`sddssequence`, `sddsprocess`).
2. Run `elegant` to track each particle in succession.
3. FFT the turn-by-turn x and y coordinates of each particle (`sddsfft`).
4. Associate each pair of FFTs with the  $\Delta P/P$  value (`sddsxref`).
5. Find the position of the peaks of each FFT, i.e., the tunes (`sddsprocess`).
6. Plot the result (`sddsplot`). Figure 1 shows the plot as it is produced by `sddsplot`.

Based on this, another script was developed to perform chromaticity correction from the results of tracking. Such sequences can be placed into scripts and used easily on different lattices. These examples show clearly how one can enhance a simulation without modifying it or having any knowledge of the source code. This is true of any code that allows fully automated postprocessing of data and preparation of input. One of the great strengths of SDDS is that it provides this capability for any compliant code.

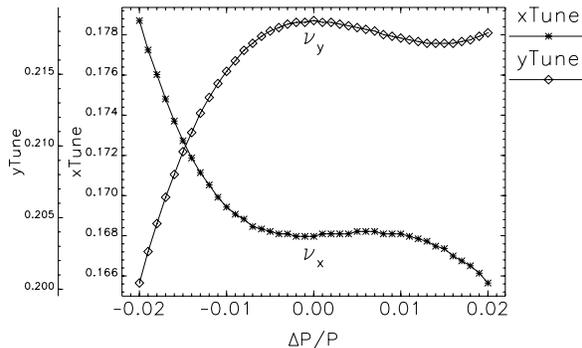


Figure 1: `elegant` tracking data postprocessed using SDDS

### 7.2 Scalable Simulation and Postprocessing

Often in the design stage of an accelerator, one runs simulations with random perturbations of magnet positions, strengths, and other parameters. Typically only 10 or 20 simulations are done, because it is very time-intensive to set up, run, and review the results of a single simulation. With SDDS, this impediment is removed, because all analysis can be done with an SDDS-based script. Once a script is written to postprocess the results of a single run, it is easily adapted to postprocess the results of an arbitrary number of runs.

For example, I simulated 1000 randomized configurations of the APS Positron Accumulator Ring [19] with alignment and strength errors, plus orbit, tune, and chromaticity correction. This was done by simulating 50 randomized configurations on each of 20 workstations. Each workstation performed 50 simulations and produced a set of SDDS files. These results were collated, processed, and displayed using a script employing the SDDS toolkit. Two examples of the results of this script are shown in Figures 2 and 3. (These plots are produced with `sddsplot` and are shown exactly as produced by that program.)

Using 1000 randomized configurations gives distributions of machine properties that simply aren't available with the small number of seeds used in most work. The use of SDDS to support quality simulations like this is not confined to any specific program, but is possible with any program that writes SDDS data (or for which an automatic conversion method exists from native data to SDDS data). In contrast to the typical situation, the amount of human effort involved with the SDDS-based approach is completely independent of the number of random seeds used. Further, the relatively small effort invested in the script pays off whenever simulation of another machine or lattice is needed.

### 7.3 Top-Up Safety Tracking [20]

Top-up operation, wherein beam is injected into a synchrotron light source storage ring with shutters open and beam available to users, is a high-priority goal at the APS. One issue with top-up operation is whether, due to a full or

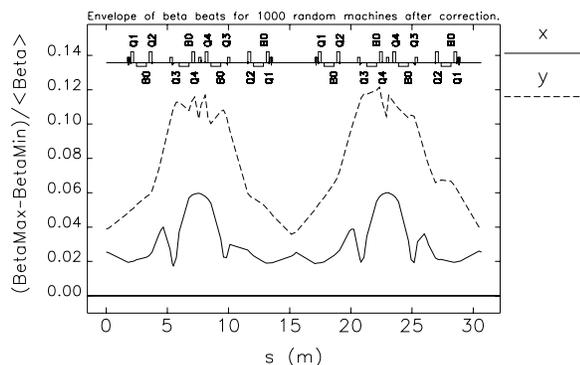


Figure 2: Beta functions from 1000 random machine simulations processed using SDDS

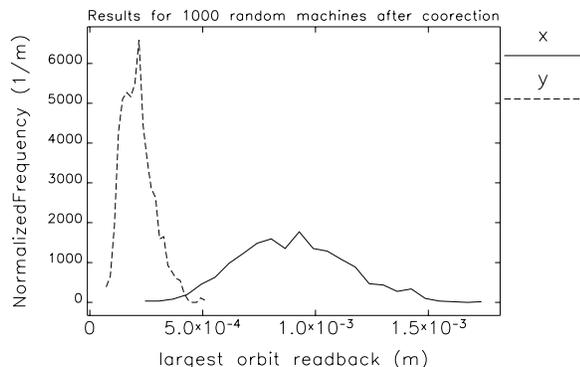


Figure 3: Orbit correction results from 1000 random machine simulations processed using SDDS

partial dipole short, injected beam could be extracted down a photon beamline while maintaining stored beam. If so, then the safety protocol planned by APS to protect users from direct injected beam would be inadequate.

Space does not permit going into the tracking studies in great detail. The approach was to simulate the existence of stored beam and the extraction of injected beam under the same conditions. The goal is to show that not only are the two incompatible, but that there is a gap (expressed in fractional strength error of a dipole) between the conditions that allow one and the conditions that allow the other. The simulations must include scenarios that cover multiple lattices, multiple tunes, and failures of other, nearby magnets in the most unfavorable way.

The location of apertures in each ring sector is part of the input for this endeavor. Each aperture configuration requires about 570 runs of *elegant* to evaluate the failure scenarios we devised. Frequently, one simulation is set up using the results of a prior one (e.g., an injected beam simulation uses the results of a stored-beam simulation that employed orbit correction to find a worst-case dipole kick from a supposed quadrupole short). A single script is used to submit the jobs, which are grouped in directories by scenario type. Each scenario directory has an individual script that prepares and submits jobs. Using 20 workstations, it takes two to three days to run a configuration.

A single script postprocesses the data. This script in fact runs a series of scenario-specific scripts. No manual col-

lation or analysis is required. The results of the 570-odd simulations are reduced to a single number (the safety gap) plus an SDDS file containing summary data for each scenario. If a particular scenario needs to be investigated, the scenario-specific script can be used to make plots.

When faced with such a complex project, many would begin by developing a new simulation code. However, the SDDS-based approach allowed us to use an existing code and complete the simulations in a far more timely fashion.

## 8 REFERENCES

- [1] M. Borland, "A Self-Describing File Protocol for Simulation Integration and Shared Postprocessors," Proc. 1995 PAC, May 1-5, 1995, Dallas, Texas, pp. 2184-2186 (1996).
- [2] M. Borland, L. Emery, "The Self-Describing Data Sets File Protocol and Toolkit", Proc. 1995 ICALEPCS, Oct. 30-Nov. 3, 1995, Chicago, Illinois, pp. 653-662 (1997).
- [3] L. Emery, M. Borland, "Commissioning Software Tools at the Advanced Photon Source," Proc. 1995 PAC, pp. 2238-2240 (1996).
- [4] M. Borland, L. Emery, N. Sereno, "Doing Accelerator Physics Using SDDS, UNIX, and EPICS," Proc. 1995 ICALEPCS, pp. 382-391 (1997).
- [5] M. Borland, "User's Guide for the SDDS Toolkit Version 1.8," <http://www.aps.anl.gov/asd/oag/manuals/SDDStoolkit/SDDStoolkit.html>
- [6] M. Borland, ANL/APS, unpublished program.
- [7] L. Emery, ANL/APS, unpublished program.
- [8] J. Lewellen, ANL/APS, private communication.
- [9] M. White, E. Lessner, "Slow Positron Target Concepts for the APS Linear Accelerator," Proc. 11th International Conf. of Positron Annihilation, Material Sciences Forum Volumes 255-257, pp. 778-780 (1997).
- [10] Y.-H. Chin, CERN-SL-94-02-AP, February 1994.
- [11] T.-M. Tran, J. S. Wuertele, "TDA -A Three-Dimensional Axisymmetric Code for Free-Electron Laser Simulation," Computer Physics Comm. 54, p. 263, 1989.
- [12] Y.-C. Chae, ANL/APS, private communication.
- [13] F. C. Iselin, "Status of MAD (Version 8.5) and Future Plans," CERN-SL-92-46-AP, Sep 1992.
- [14] M. Dehler *et. al.*, "Status and Future of the 3-D MAFIA Group of Codes," COMPUMAG Conf. on the Computation of Electromagnetic Fields, Tokyo, Japan, Sep 3-7, 1989.
- [15] L. Emery, ANL/APS, private communication.
- [16] F.W. Jones *et. al.*, "ACCSIM: A Program to Simulate the Accumulation of Intense Proton Beams," Int. Conf. on High Energy Accelerators, Tsukuba, Japan, Aug 22-26, 1989.
- [17] F. Iazzourene *et. al.*, "RACETRACK USER'S GUIDE VERSION 4.01," Sincrotrone Trieste Report ST/M-97/7, July 1992.
- [18] W.M. Fawley, "An Informal Manual for GINGER and its post-processor XPLOTGIN," LBID-2141, December 1995.
- [19] M. Borland, "Commissioning of the Argonne Positron Accumulator Ring," Proc 1995 PAC, pp. 287-289 (1996).
- [20] M. Borland, L. Emery, private communications.

# MODELS AND SIMULATION OF BEAM HALO DYNAMICS IN HIGH POWER PROTON LINACS\*

Thomas P. Wangler

*Los Alamos National Laboratory, Los Alamos, NM 87545*

*Abstract*

We discuss the application of both multiparticle simulation techniques and analytical models known as particle-core models to the problem of understanding beam halo in high-power proton linacs. We emphasize the importance of multiparticle simulation including the space-charge forces as an essential tool for the description of the beam dynamics in a modern high-intensity proton linac. In addition, we have found that to understand the physics of beam halo, it has been necessary to supplement the simulations with a model known as the particle-core model.

## 1. INTRODUCTION

Within the past ten years, high-power proton linacs have been developed for several applications including neutron spallation sources, tritium production, and nuclear waste transmutation. The most challenging design parameters are those associated with the Accelerator for Production of Tritium (APT) linac [1]. APT is a CW proton linac with a final energy in the 1-GeV range. The average beam current is 100 mA, resulting in a final average beam power in the 100-MW range. Although the beam-physics regime is not very different than that for the LANSCE linac, which operates at Los Alamos, APT has an average current and beam power, which are a factor of 100 greater than for LANSCE. Consequently, designing for low beam loss to avoid radioactivation of the accelerator becomes a high priority to ensure that hands-on maintenance can be carried out, and that availability of the machine will remain high. To limit the average beam loss to the same absolute levels as have been achieved at LANSCE, the beam-dynamics design requirement for APT is to limit the beam loss to less than  $10^{-5}$  total or an average loss per unit length of less than  $10^{-8}/\text{m}$  for energies greater than 100 MeV. To accomplish this we must understand the causes and characteristics of the beam halo, since particles in the halo are those that may be lost on the accelerator walls.

## 2. MULTIPARTICLE SIMULATION

Multiparticle simulations are necessary for two important reasons. First, simulations are needed to describe the beam dynamics that depend on the nonlinear and time-dependent space-charge force, which is related to the evolving particle distribution. Second, simulations are needed to describe the effects of random linac errors or imperfections, which are treated using a Monte Carlo approach.

There is no consensus at present on a precise definition of the beam halo. Generally, the term halo describes the outer low-density edge of the beam in phase space that surrounds a dense central core. Typically, the halo particles are those that lie outside the phase-space boundary of an ellipse with the same shape as the rms emittance and an area of about 8 to 10  $\epsilon_{\text{rms}}$ , where  $\epsilon_{\text{rms}}$  is the rms emittance.

As a result of work done during the past few years, the main cause of beam-halo formation in high intensity proton-linac beams has been identified as arising from the space-charge forces that act in mismatched beams. In particular, this halo mechanism is the result of the coupling of collective oscillations to the motion of single particles. Because of the focusing provided in the three orthogonal directions, mismatch of the rms beam sizes generally excites some admixture of three collective envelope modes of the bunched beam, causing oscillations in the rms beam sizes. These envelope modes are shown in Fig. 1. Two modes have radial transverse oscillations where the transverse rms projections  $x_{\text{rms}}$  and  $y_{\text{rms}}$  move in phase; the longitudinal rms projection  $z_{\text{rms}}$  is either in phase with the transverse motion for a high-frequency or breathing mode, or out of phase for a low-frequency mode. The third envelope mode is the quadrupole mode, in which  $x_{\text{rms}}$  and  $y_{\text{rms}}$  are out of phase, and for this mode there is no longitudinal motion.

The focusing forces also produce oscillations of the individual particles, known as transverse or betatron oscillations, and longitudinal or synchrotron oscillations. When the oscillation frequency of a particle is half the frequency of one of the collective envelope modes, that particle can exchange energy with the mode through parametric resonance. The halo is formed mainly from those particles that are driven to large amplitudes through resonance with one or more of the envelope modes. Of the three envelope modes, the high and low frequency modes have already been identified as important for APT.

In the parameter regime of APT, the plasma parameter, or number of particles in a Debye sphere is much greater than unity, about  $10^6$ . In this regime the multiparticle Coulomb effect is accurately described by an average force, called the space-charge force, which is derivable from a potential that satisfies Poisson's equation. Discrete particle effects represent small fluctuations relative to the space-charge force, and are generally neglected. The space-charge force is calculated in computer codes by using the particle-in-cell (PIC) method, which is essentially a numerical method for solving the coupled Vlasov-Poisson equations that describe the simultaneous evolution of both the particle distribution and the space-charge fields. Before solving

Poisson's equation for a given time step, the charge distribution is transferred to a mesh using the particle distribution and including some form of smoothing to reduce numerical errors that are equivalent to artificial discrete-particle collisions.

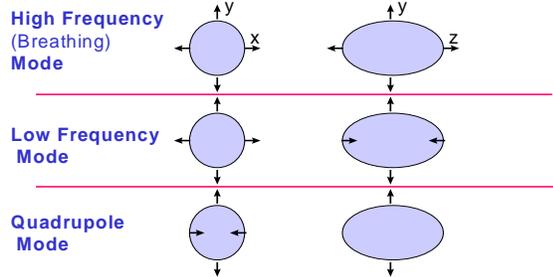


FIG. 1. Envelope modes of a mismatched bunched beam.

The simulation codes that are used for proton linacs typically use axial position as an independent variable, rather than time. For each particle in the bunch, the six phase-space variables, three position and three momentum coordinates, are tracked through the linac. Each time the space-charge subroutine is called, the particle coordinates must be transformed to positions corresponding to a fixed time, and then the particles are Lorentz transformed into the center-of-momentum frame of the bunch. A mesh is superimposed over the bunch and the particle charges are distributed among the mesh cells using an area weighting method that takes into account the position of each particle within its cell to determine the relative fraction of the particle charge that is assigned to each cell. This method provides the smoothing that was discussed earlier. Once the charge distribution has been determined, the electric field components are obtained on each grid point by numerical solution of Poisson's equation. The field components at the location of each particle are then obtained by interpolation of the field components from their values at the grid points. Two different PIC codes have been used, SCHEFF [2], a 2D r-z code, which uses an approximate correction for the effect of an elliptical transverse cross section, and a fully 3D PIC code called 3DPIC [3]. The 3DPIC code treats the 3D effects more accurately than SCHEFF, and up to  $10^7$  particles have been run in simulations on the CRAY T3E parallel computer. These two PIC codes have been compared for the APT linac design. Using  $10^5$  particles and with no random linac errors, excellent agreement has been obtained for all rms quantities and also for the maximum particle displacements. The SCHEFF routine has been benchmarked against the experimental measurements of rms beam properties at the LANSE proton linac; excellent agreement to within about 15% was observed [4].

The extent and the magnitude of beam halo in the linac is dependent on the machine errors or imperfections that produce mismatch. The simulation code uses the Monte Carlo approach to select errors within known tolerances. Many computer-simulation runs are required

from which the results may be combined to obtain probabilistic predictions for the expected beam distribution. For APT, we believe that linac errors could lead to effective mismatches in the range of 20% to 30%. The linac errors include misalignments and energy errors in the injected beam, RFQ higher multipoles and image-charge forces, quadrupole imperfections such as displacement, tilt, rotation, fringe fields, and higher multipoles, and cavity imperfections including phase and amplitude errors and tilts for the cavity fields. Fig. 2 shows the transverse beam size versus energy for APT for 20 runs with 100,000 particles for each run with different random errors. Shown are the aperture radius, the  $x_{rms}$  beam projection and the maximum displacement.

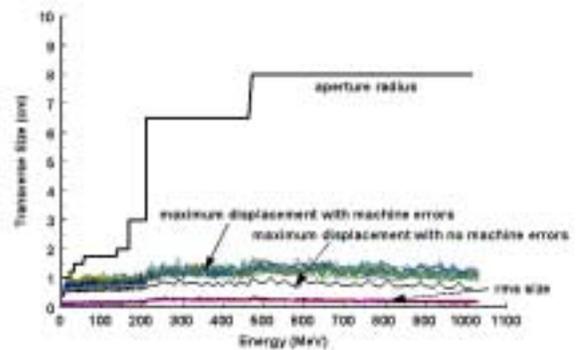


FIG. 2. Transverse rms size and maximum displacement versus energy along the APT linac for 20 simulation runs with different random linac errors and  $10^5$  particles per run. The aperture radius is shown for comparison, and the blue curve just below the 20 curves of maximum displacement is the maximum displacement when no random errors are present.

High performance parallel computing is becoming an important tool for linacs like APT for two reasons. First, the requirement that the total beam loss above 100 MeV must be limited to  $10^{-5}$  implies that to see losses at this level, simulation runs with greater than  $10^5$  particles per run are needed. Runs with  $10^7$  particles per run and greater will be helpful for obtaining even better statistical precision in the halo. Also, the use of parallel computing allows us to use the 3D PIC space-charge calculation, which is important to ensure that physics associated with 3D effects is not missed. Running  $10^7$  particles through the APT linac after the RFQ with the 3DPIC code, using 128 processors on a 64X64X128 grid, takes about 5.5 hours on the CRAY T3E computer. Fig. 3 shows the phase-space plots at the end of the APT linac for a  $10^7$  particle run on the CRAY T3E, including random linac errors.

### 3. PARTICLE-CORE MODEL

Computer simulation is an important tool but should not become a substitute for understanding the physics.

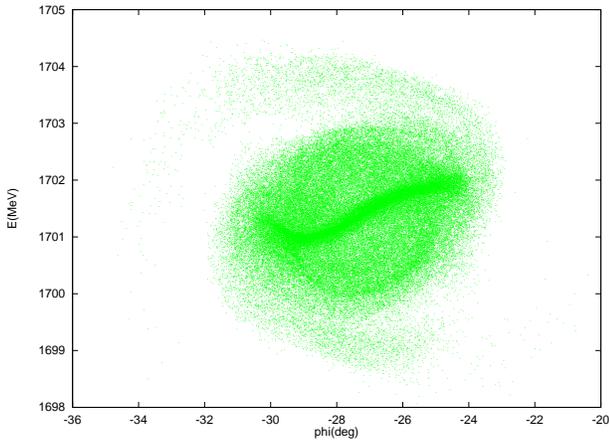


FIG. 3. Final longitudinal phase space plot (phase versus energy relative to the synchronous phase and energy) for a  $10^7$ -particle multiparticle simulation of the APT linac beam, including random linac errors, using the CRAY T3E parallel computer.  $10^5$  particles are included in the plot.

Models are important for providing additional understanding and insight. The particle-core model of beam-halo formation provides a framework for understanding the results of simulation, and provides scaling formulas that lead to guidelines for accelerator design. In this model the collective behavior is represented by the motion of the core. There are several particle-core models that have been constructed with different core geometries, including a 2-D continuous cylindrical beam [5,6], a spherical bunch, and a spheroidal bunch [7,8]. The latter model is the most representative model for a beam bunch in a linac. However, in this paper I discuss some results for the simpler case of a spherical bunch. In the spherical-bunch model, the beam core is represented by a uniform-density sphere in a uniform-focusing channel; the core experiences a linear external focusing force, as well as the defocusing effects from emittance plus the space-charge force. A mismatch of the initial core size is introduced that is symmetric in all three planes. This mismatch excites the radial breathing mode of the core. First, to understand the transverse particle dynamics, we study the motion of test particles that pass through the axis as they interact with the nonlinear space-charge field of the oscillating core and the applied linear external focusing force.

In the sphere model, the equation for the core motion is the envelope equation,

$$\frac{d^2 R}{ds^2} + k_0^2 R - \frac{(4\epsilon_{rms})^2}{R^3} - \frac{\kappa}{R^2} = 0, \quad (1)$$

where  $s$  is the axial coordinate,  $R$  is the core radius, the focusing force is represented by  $k_0$  which is also the wavenumber or phase advance per unit length of particle oscillations at zero current,  $\epsilon_{rms}$  is the rms unnormalized emittance, and

$$\kappa = \frac{q^2 N}{4\pi\epsilon_0 mc^2 \gamma^3 \beta^2}, \quad (2)$$

where  $q$ ,  $m$ , and  $\beta c$  are the charge, mass, and axial velocity of the particles, respectively,  $\gamma$  is the relativistic Lorentz factor,  $N$  is the number of particles per bunch,  $c$  is the speed of light, and  $\epsilon_0$  is the permeability of free space. For a matched beam, the core radius is constant, denoted by  $R = R_0$ . The motion of test particles that pass through the axis is governed by

$$\begin{aligned} \frac{d^2 x}{ds^2} + k_0^2 x - \frac{\kappa x}{R^3} &= 0, \quad x \leq 0, \\ \frac{d^2 x}{ds^2} + k_0^2 x - \frac{\kappa |x|}{x^3} &= 0, \quad x > 0. \end{aligned} \quad (3)$$

The net focusing force, including the space-charge term for a particle that always remains within the core, is represented by the wavenumber  $k$ , given by  $k^2 = k_0^2 - \kappa/R_0^3$ . For small mismatch oscillations the core breathing-mode wavenumber  $k_c$  can be expressed as  $k_c^2 = 3k_0^2 + k^2$ .

For the matched case, the core radius is constant, and there is no net change in the energy of a particle averaged over a complete period of the particle motion. For the mismatched case, the core radius oscillates, and particles can either gain or lose energy with each transit through the core. The particles experience a nonlinear space-charge force when they are outside the core and from Gauss' law this force is independent of the instantaneous size of the core. When the particles pass through the core, they are decelerated by the space-charge force as they approach the axis, and accelerated by the space-charge force as they leave the axis. The net space-charge impulse delivered to the beam is equal to the sum of a core-entrance contribution plus a core-exit contribution. These impulses may be either diminished or enhanced relative to the matched case, depending on whether the core radius is larger or smaller than the equilibrium value at the time the particle passed through. For example, if a particle enters the core when its radius is larger than the matched value, and exits when its radius is smaller than the matched value, a net energy impulse is delivered to the particle. Gluckstern [6], studying a particle-core model for a cylindrical beam, has shown that the effect of the core on the motion of the particles can be described by a nonlinear parametric resonance. The particles resonate with the core when the particle wavenumber  $\nu$  is related to the core breathing-mode wavenumber  $k_c$  by  $k_c = 2\nu$ . Note that  $\nu = k$  for particles that always remain within the uniform core, and  $\nu > k$  for particles with amplitude larger than the core radius, because of the reduced influence of the space-charge force for larger amplitudes. For nonzero beam current, one can show that the resonance condition requires that  $\nu > k$ , i.e. resonant particles must have amplitudes larger than the core radius. Thus, the decrease of the space-charge field with increasing displacement,

experienced by particles that are outside the core, produces an increase of the wavenumber  $\nu$  with amplitude so that the  $k_c = 2\nu$  resonant condition cannot be maintained as the amplitude increases; this effect limits the resonant amplitude growth. It is convenient to define two parameters, a space-charge tune-depression ratio  $\eta = k/k_0$ , and a mismatch parameter  $\mu = R_i/R_0$ , where  $R_i$  is the initial core radius.

Fig. 4 shows displacement versus axial distance for a particle driven by the resonance. The characteristics of the model are also displayed in the stroboscopic plot in Fig. 5, where a maximum amplitude is shown as the maximum displacement of the outer separatrix for particles in the resonance regions that are located between the inner and outer separatrices. By solving the equations of the sphere model numerically, we can determine the maximum amplitude for the resonantly driven particles. Fig. 6 shows a comparison of the maximum amplitudes from multiparticle simulations for an initial spherical Gaussian bunch with the maximum amplitudes obtained from the particle-core model. The agreement is good; the points from the simulation closely follow the general shape of the curves from the model and lie only slightly higher. Empirically we find that the maximum amplitude of the resonant particles satisfies an approximate formula

$$x_{\max} = x_{rms} (A + B|\ln(\mu)|) \quad (4)$$

where A and B are constants,

$$x_{rms}^2 \equiv \frac{\epsilon_{rms}}{k_0 \beta \gamma} [1 + u]^{2/3}, \quad (5)$$

$$u = \frac{q^2 N}{20\sqrt{5}\pi\epsilon_0 mc^2 (k_0 \beta \gamma^3 \epsilon_{n,rms}^3)^{1/2}}, \quad (6)$$

and  $\epsilon_{n,rms}$  is the normalized rms emittance. These results from the sphere particle-core model provide the following guidelines for minimizing beam halo: good beam matching, small initial emittance, small number of particles per bunch (achieved for a given average current by choosing high bunch frequency), large  $k_0$  (strong focusing), and large  $\beta$  and  $\gamma$  (halo amplitudes are reduced at high energy).

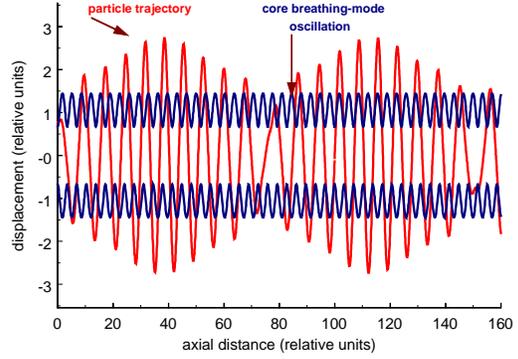


FIG. 4 Transverse displacement versus axial distance showing parametric resonance in the sphere particle-core model for a test particle with initial displacement of unity and initial divergence of zero. The envelope of the uniform-density spherical core is shown oscillating at about twice the frequency of the resonant test particle.

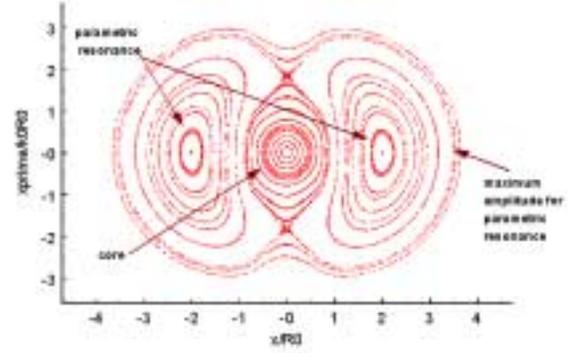


FIG. 5. Stroboscopic transverse phase-space plot for the sphere particle-core model for  $\mu=1.5$  and  $\eta = 0.5$ , showing the core region, the regions of parametric resonance, and the maximum displacement of the resonantly excited particles.

The sphere model presented here treats all three degrees of freedom the same. This model can be modified to account for nonlinear RF focusing in the longitudinal direction. If we assume that the core motion is approximately unaffected by the nonlinear focusing and that only the large amplitude test particles are affected, we can change the equation of motion for the test particles to

$$\frac{d^2 z}{ds^2} - \frac{qE_0 T (\cos(\phi_s - 2\pi z / \beta \lambda) - \cos(\phi_s))}{mc^2 \beta^2 \gamma^3} - \frac{\kappa z}{R^3} = 0, \quad z \leq R,$$

$$\frac{d^2 z}{ds^2} - \frac{qE_0 T (\cos(\phi_s - 2\pi z / \beta \lambda) - \cos(\phi_s))}{mc^2 \beta^2 \gamma^3} - \frac{\kappa z}{|z|^3} = 0, \quad z > R.$$

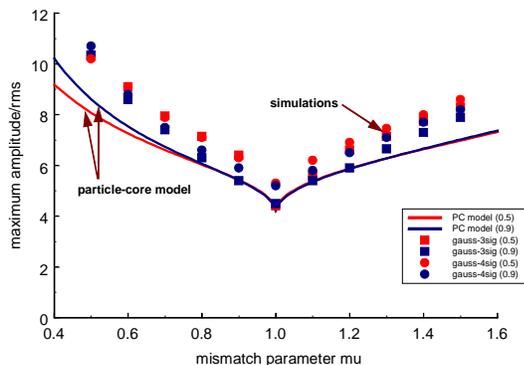


FIG. 6. Comparison of the maximum displacement from multiparticle simulations with the sphere particle-core model. The simulations are for initial spherical Gaussian bunches truncated at  $3\sigma$  and  $4\sigma$ ,  $\mu=1.5$ , and  $\eta = 0.5$  and  $0.9$ .

This change has a significant effect on the dynamics, by weakening the longitudinal focusing and reducing the particle frequencies for the large amplitude particles. The resulting stroboscopic plot for parameters near the APT parameter regime is shown in Fig. 7, for the parameter choices  $\mu=1.5$ ,  $\eta=0.5$ ,  $\phi_s=-30$  deg, and  $\phi_0=6.67$  deg, where  $\phi_s$  and  $\phi_0 = 2\pi R_0/\beta\lambda$  are the synchronous phase and the phase half width of the bunch, respectively.

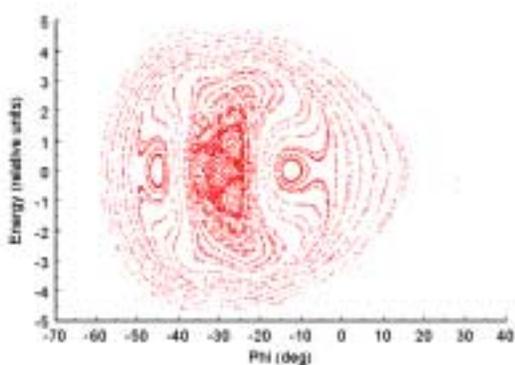


FIG. 7 Stroboscopic longitudinal phase-space plot of energy versus phase relative to the synchronous particle for the sphere particle-core model with a nonlinear rf force applied to the test particles and for parameters  $\mu=1.5$ ,  $\eta = 0.5$ ,  $\phi_s = -30$  deg and  $\phi_0 = 6.67$  deg.

While the sphere particle-core model exhibits many of the important features of the physics, it neglects some important effects which can only be treated properly by assuming a more realistic shape for the core geometry. Improvement is obtained by assuming a spheroidal core shape with radial and axial motion, where typically

$z_{rms} > x_{rms} = y_{rms}$ . This core geometry allows us to account for the presence of both the high and the low frequency modes. The spheroid model is being studied by both University of Maryland [7] and the Lawrence Livermore National Laboratory [8]. For the APT linac it is found that the high frequency mode primarily affects the transverse halo dynamics while the low frequency mode primarily affects the longitudinal halo dynamics.

## 4. CONCLUSION

Although multiparticle simulation is needed for a detailed description the beam halo, the particle-core model has been an indispensable complimentary tool. Combining these two methods has lead to a useful picture of the underlying physics of beam halo. Systematic studies using both simulation and the particle-core models describe the physics over the parameter regime of importance for high-power proton linacs.

## 5. ACKNOWLEDGMENTS

The author acknowledges support from the U.S. Department of Energy, and particularly wishes to thank John Barnard, Robert Gluckstern, and Robert Ryne for useful discussions.

## 6. REFERENCES

- 1 P.W. Lisowski in *Proc. 1997 Particle Accelerator Conf.*, Vancouver, British Columbia, Canada, (IEEE, Piscaraway, NJ, 1998), p. 3780.
- 2 T. P. Wangler, *Principles of RF Linear Accelerators* (John Wiley & Sons, Inc., New York, 1998), pp. 270-272.
- 3 R. Ryne, The U.S. Grand Challenge in Computational Accelerator Physics, to be published in Proc. XIX International Linear Accelerator Conference (LINAC98), August 23-28, 1998, Chicago.
- 4 R. W. Garnett, R. S. Mills, and T. P. Wangler, *Proc. 1990 Linac Conf.*, Sept. 9-14, 1990, Albuquerque, NM, Los Alamos Report LA-12004-C, pp. 347-350.
- 5 For a discussion of the cylindrical beam model and for additional references see T.P.Wangler, K.R.Crandall, R.Ryne, and T.S.Wang, "Particle-Core Model for Transverse Dynamics of Beam Halo", to be published in Physical Review Special Topics – Accelerators and Beams.
- 6 R. Gluckstern, Phys. Rev. Lett. **73**, 1247 (1994).
- 7 R. L. Gluckstern, A. V. Fedotov, S. S. Kurennoy, and R. D. Ryne, Phys. Rev. E. **58**, 4977 (1998).
- 8 J. Barnard and S. Lund, *Proc. 1997 Particle Accelerator Conf.*, Vancouver, British Columbia, Canada, (IEEE, Piscaraway, NJ, 1998), p. 1929.

# A MULTIGRID-BASED APPROACH TO MODELING A HIGH CURRENT SUPERCONDUCTING LINAC FOR WASTE TRANSMUTATION

Paolo Pierini

INFN Milano – LASA, Via Fratelli Cervi, 201, I-20090 Segrate (MI) Italy

## Abstract

High current proton accelerators are nowadays planned for various applications that make use of a large flux of spallation neutrons. The INFN has initiated a study on the conceptual and technical feasibility of a (greater than) 40 MW CW linac for nuclear waste transmutation and energy production.

A safe and reliable operation of these high current machines will require proper control of the beam losses. Accurate simulation codes are needed to assess that the beam losses are maintained to a safe level. As a part of the INFN program, an “ad hoc” code development activity has started, using recent programming techniques and numerical algorithms. The code in development deals, for now, with the proton beam dynamics in the high energy part of the proposed machine, composed by superconducting RF (elliptical) cavities placed between the long drifts of a quadrupole doublet array. The cavities are treated using an on-axis field distribution, either provided in analytical form or as tabular data.

Space charge is evaluated with a fast Poisson solver, that uses a 3D multigrid algorithm. V Cycle or full multigrid algorithms appear to be promising in terms of speed and memory requirements, and can be readily modified for parallel computers. Checks with standard direct point-to-point calculations have been performed.

A major effort has been put in using a modular approach for the data and program design. The code conforms to the F90 syntax and, where possible, makes use of safe programming criteria (controlled scoping of variables through MODULEs and PRIVATE/PUBLIC qualifiers, explicit procedure INTERFACES with INTENT declaration, dynamic allocation of all the data structure for the beam line, the particles and the space charge meshes). Preliminary results of this ongoing work will be presented in this contribution.

## 1 DESCRIPTION OF THE CODE

The code has been developed for the simulation of high current beams in a proton linac with superconducting RF cavities. The chosen independent coordinate is time, expressed in terms of the phase  $\phi = \omega_{RF} t$ , where  $\omega_{RF}$  is  $(2\pi)$  the frequency of the radiofrequency (RF) field. Particle coordinates and momenta are updated in phase steps. If a particle crosses during a timestep two adjacent element boundaries, a substep is performed up to the element end and then the timestep is continued in the

following element. Coordinates at the element output are stored in buffers and written to disk in case phase space plots at given positions in the beamline are desired.

Space charge kicks are applied in phase steps multiple of the particle tracking steps. The space charge kick on each particle can be evaluated either using a direct (point to point) method (with the use of a “screening radius” to prevent the Coulomb divergence of macro particles lying at close distance) or using a fast 3D Poisson solver in the beam frame, based on a multigrid algorithm[1].

### 1.1 Beamline elements

The linac beamline is made of quadrupole, drift space and RF cavity elements. For quadrupoles and drift spaces analytical maps are applied for the evolution during the timestep, whereas in the RF cavities a direct integration of the equations of motion in the space-time dependent cavity electric field is performed.

The RF cavity field is described through an analytical expression of the on-axis longitudinal field of an “ideal  $\beta$  cavity”, as follows:

$$E_z(z,t) = 2E_{acc} \sin(\omega_{RF}t + \phi_c) \sin \frac{\omega_{RF}z}{c\beta_c}$$

where  $E_{acc}$  is the cavity accelerating field,  $\beta_c$  the cavity synchronous velocity and  $\phi_c$  the cavity phase. Thus the energy gain for the synchronous particle with  $\beta = \beta_c$  is  $E_{acc}L$ , where  $L = N\lambda_{RF}\beta_c/2$  is the active cavity length and  $N$  the number of cells. An iterative preprocessor algorithm (before the actual tracking takes place) sets the cavity phases along the beamline to the desired value for the synchronous phase, defined for the reference particle at the cavity center.

A second order expansion for the off-axis electric and magnetic fields is used. A future extension allowing to use longitudinal electric field maps provided by a cavity eigenfield solver as SUPERFISH[2] is planned.

A special beamline element, providing a uniform focussing channel in the three planes, has also been included in order to check analytical predictions and to test the space charge solver numerical properties with different beam aspect ratios (as discussed in Ref. [3]).

### 1.2 Space charge multigrid implementation

The three dimensional space charge calculation is performed solving the Poisson equation in the rest frame of the synchronous particle (the reference particle). The

electrostatic field in the rest frame is then transformed to the electric and magnetic fields in the laboratory frame, where the space charge kick is applied to the individual particles. The charge distribution on the numerical grid is evaluated in the rest frame using a cloud in cell (CIC) charge assignment scheme, and field interpolation on the particle positions from the grid values uses a trilinear interpolation scheme, for consistency[4].

The Poisson equation for the electrostatic potential is solved on a square mesh, typically consisting of  $33 \times 33 \times 33$  or  $65 \times 65 \times 65$  points (extending to at least 6 rms), using a fast multigrid solver.

Multigrid methods are the fastest iterative methods for the solution of elliptic problems and are based on two main components:

- 1 *standard relaxation algorithms* (as the weighted Jacobi or the Gauss-Seidel)
- 2 the discretization of the model problem on *a series of nested grid levels* (obtained, for example, by step doubling).

The multigrid scheme makes use of the smoothing properties of relaxation schemes: the high frequencies (relative to the grid step) of the error decay by orders of magnitude in the first few iteration of the relaxation operator. However, relaxation is very ineffective for the smooth components of the error (again, the smoothness is to be intended relatively to the grid step).

In a multigrid scheme, relaxation is performed for a few cycles (1-4, typically), to nearly eliminate the fast oscillating errors. After that, only smooth error are left and the estimate of the solution is “restricted” to a coarser level, where a portion of these error components are no longer smooth, due to the step doubling. This process is iterated down the grid levels until reaching the elemental grid consisting of  $3 \times 3 \times 3$  gridpoints (and only one unknown), which can be directly solved. The approximate solutions at each grid level are used then to form a better estimate for the solution on the finer level, through a proper “prolongation” operator. The procedure is then iterated all over the grid hierarchy until the desired convergence is reached. This cycling strategy is known as the V-cycle multigrid scheme and is pictorially illustrated in Figure 1. The V-cycle scheme has a favorable scaling with respect to the grid size, since the number of cycles required in order to converge to a specified residual norm is independent on the dimensions of the finest grid.

The multigrid has been implemented in the code using trilinear averaging (known as full-weighting) of the 27 neighboring nodes for the *restriction* operator and trilinear interpolation as the *prolongation* operator. The smoothing operator is the *Gauss-Seidel* algorithm.

Other cycling schemes are possible and were implemented in the code, like the Full Multigrid algorithm[1], but the V cycle allow a simple acceleration mechanism: when using the potential map computed at the previous space charge evaluation the number of

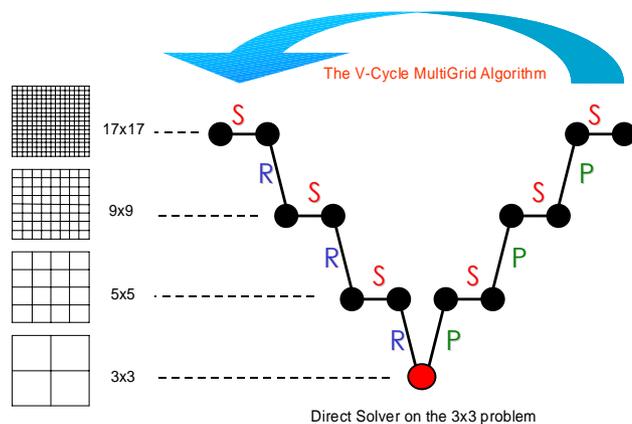


Figure 1: A pictorial representation of the multigrid V-cycling scheme. The S denotes the smoothing operator, R and P denote the restriction and prolongation operators, respectively.

iterations required to converge to the specified norm can be reduced by 30%.

All the multigrid internal routines and data structures are stored in a Fortran 90 Module with PRIVATE attributes and a few PUBLIC interfaces.

### 1.3 Tests and performance of the space charge routine

The space charge solver has been tested both with the results from the direct point to point routine (which can be used with a limited resolution due to the scaling of execution time with the square of the particles number) and with analytical tests. In Figure 2 we show the case of the radial field of a uniformly charged sphere, with a comparison between the analytical solution and the numerical solution of the multigrid routine, interpolated on the position of 10,000 test random positions in the 3D space, in and around the sphere.

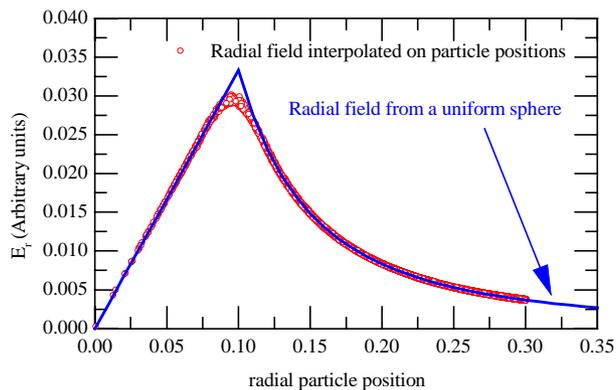


Figure 2: Radial field from a uniformly charged sphere (solid line) and field from the multigrid solver interpolated on the position of 10,000 test positions. The grid extends from -0.5 to 0.5 in x, y, z, the sphere radius is 0.1 and  $33 \times 33 \times 33$  grid points have been used.

Obviously, the numerical solution cannot represent the discontinuous derivative of the exact solution near the sphere edge, due to the charge smoothing on the grid, but is otherwise in excellent agreement with the analytical solution and does not exhibit a directional dependence.

The CPU scaling of the space charge algorithm can be seen from Figure 3, where the time needed for the space charge evaluation (including the charge assignment and the field interpolation) is plotted both with respect to the number of grid points and to the number of particles. Above 100,000 particle most of the execution time is spent in the charge assignment/field interpolation steps, and the space charge step timing is weakly dependent on the finest grid size.

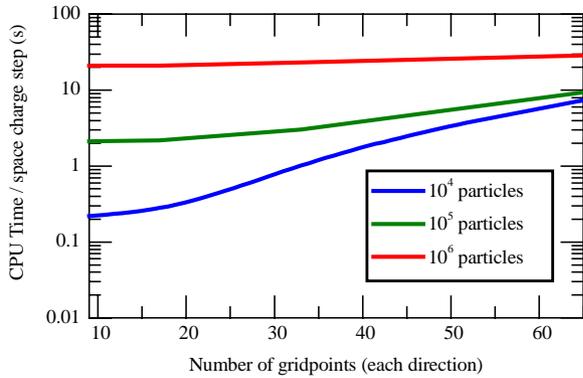


Figure 3: CPU time per space charge step as a function of the number of grid points (on the finest grid) and of the number of particles. Times were measured on a 300 MHz Pentium II 2-processor NT Workstation.

A further improvement for the treatment of the open boundary conditions that allows to increase the mesh resolutions at the beam core by using semi-analytical boundary conditions is currently under test[5].

## 2 THE TRASCO SC LINAC SIMULATIONS

The code has been used to validate the linac design for the superconducting linac of the TRASCO Project[6]. Simulations with up to 100,000 particles have been used for beam dynamics calculations of a 25 mA proton beam in the 100-1600 MeV superconducting accelerator. The linac is split in three sections with five cell elliptical cavities designed for a synchronous  $\beta$  of 0.5, 0.65 and 0.85, grouped in cryostats of 2,3 and 4 structures each.

The matched Twiss beam parameters and the quadrupole and cavity matching parameters between the sections have been determined with adiabatic smooth matching routines that we have implemented in the linear optics code TRACE-3D[7].

Figure 4 shows the rms beam envelopes along the 750 m of the linac beamline. The two section transitions can be clearly identified at approximately 100 and 250 m.

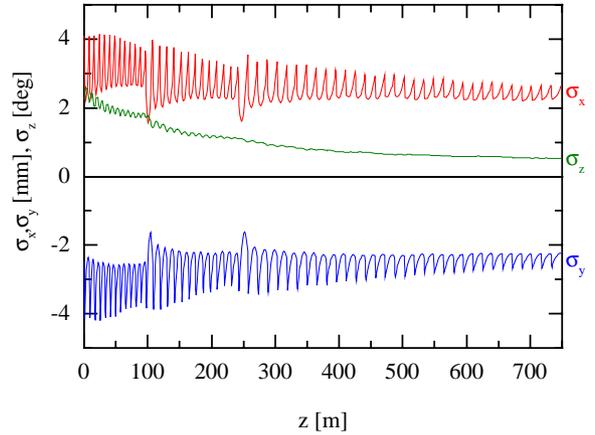


Figure 4: The three rms beam envelopes in the TRASCO linac, matching conditions were determined from TRACE3-D.

Figure 5 shows the horizontal beam size fractions along the linac, normalized by the rms horizontal beam size (in order to get rid of the envelope oscillations in the doublet channel). The curves in the figure refer to the horizontal position containing 90%, 99%, 99.9% and 100% of the beam radius (divided by the horizontal rms beam size). No distribution mismatch can be seen for the whole beam in the first linac section (up to 96 m), where the beam size fractions stay constant, whereas a mismatch in the tails of the distribution is introduced in the transition from the first to the second section. This mismatch is clearly seen by the onset of betatron oscillation from the 99% of the beam and above. The noisy pattern in the 100% curve is due to a poor statistics of the few particles in the outer tail (0.1% of the beam, i.e. only 100 simulation particles).

The simulation used 100,000 particles distributed uniformly in the 6D phase space.

Similar plots can be shown for the other beam profiles (the vertical and the longitudinal), with a distribution mismatch arising from the section transitions.

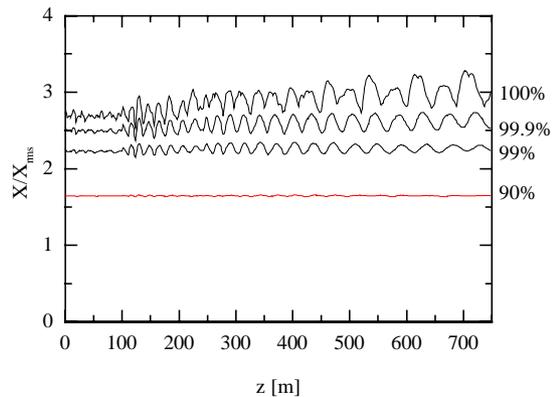


Figure 5: Different horizontal beam ratio profiles along the linac, corresponding to the position containing 90%, 99%, 99.9% and 100% of the beam, divided by the rms value.

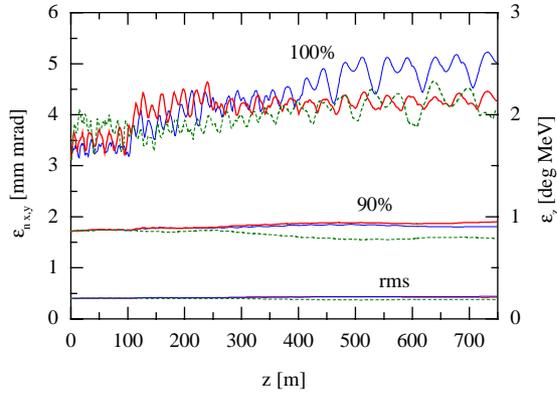


Figure 6: Beam emittances along the linac. Bottom lines are the rms emittances, lines at the middle are the 90% emittances, upper lines are the 100% beam emittances. The longitudinal emittance (to be read on the right axis) is the dashed line, the continuous lines are the horizontal and vertical emittances.

Figure 6 shows the beam emittances along the linac. The figure shows the rms (bottom lines), 90% (intermediate) and 100% (upper lines) beam emittances. No appreciable rms emittance growth can be seen from the plot (when shown in an expanded scale, the rms emittance growth over the whole linac length is limited to well below 10%). An increase of the total emittance at the first transition between the sections can be seen from the plot. This emittance growth at the distribution tails is associated with the distribution mismatch induced by the section transitions.

In Figure 7 the phase space plots at the end of the linac are shown, for the corresponding matched case shown in Figures 3 to 6. Evidences of the beam tails formation in both the longitudinal and transverse planes can be seen from the phase space plots. However, no clear sign of beam filamentation or onset of resonances can be seen from the figure.

Note that the characteristic “rectangular” phase space distribution, corresponding to a moderately tune depressed ( $\nu \approx 0.6$ ) beam, can be seen from Figure 7.

In Figure 8 the same phase space plots at the linac output, as in Figure 7, are shown, for a case where the beam is mismatched in both the horizontal and longitudinal phase spaces. The mismatch factor for this case is 30% (both in the longitudinal and transverse planes). Here, the onset of beam filamentation due to the mismatch can be clearly seen, especially in the longitudinal phase space plane. In this case the rms longitudinal emittance is only 7% higher than the matched case, but the evidence of a strong presence of particles in the beam tails indicates a substantial increase of the total beam emittance.

Simulations with smaller values of the mismatch factors have been performed in order to assess the induced emittance growth. Mismatch factors lower than

10% result in a negligible increase of the rms emittance and a 20% increase of the total (100%) emittances[8].

### 3 CONCLUSIONS

A dedicated tool for the simulation of high current beam dynamics in a superconducting linac has been developed in the framework of the TRASCO Project for a high power linac for nuclear waste transmutation. The code is being used to assist the validation of the linac design and to verify the stringent requirements on beam losses that could cause the linac activation.

A 3D space charge evaluation routine based on an iterative multigrid scheme has been developed and thoroughly tested. The space charge solver efficiency is enough to allow simulations up to 1,000,000 macro particles on a desktop workstation. A future improvement of the code will use parallel processing directives in order to decrease the time consuming space charge evaluation procedures (i.e. the charge smoothing/field interpolation).

The reference design layout for the TRASCO linac has been verified with simulations. No particle losses (with a resolution of 1 to 100,000) have been found, and the emittance growth is limited if the beam mismatch can be controlled to 10%.

For further information on the code and its distribution, please contact the Author[9].

### 4 ACKNOWLEDGEMENTS

I wish to acknowledge the stimulating discussions with all my colleagues of the TRASCO group in Milano-LASA and the contribution of Nicolas Pichoff, from CEA-SACLAY, for the help in the tests of the multigrid algorithm.

### REFERENCES

- [1] ‘A Multigrid Tutorial’, by Briggs, William, published by SIAM, Philadelphia, PA 1987, ISBN 0-89871-221-1, and ‘An Introduction to Multigrid Methods’, by Wesseling, Pieter, published by John Wiley and Sons, New York, 1992, ISBN 0-471-93083-0.
- [2] ‘POISSON/SUPERFISH’, by Billen, James and Young, Lloyd, LA-UR-96-1834, Los Alamos, NM, 1996.
- [3] ‘Simulation Results with an Alternate 3D Space-Charge Routine, PICNIC’, Pichoff, N, Lagniel, J M, Nath, S, in Proceedings of the XIX International Linear Accelerator Conference, August 23-28 1998, Chicago, IL.
- [4] ‘Computer Simulations Using Particles’, by Hockney, R W and Eastwood, J W, published by Adam Hilger, New York, NY 1988, ISBN 0-85274-392-0.
- [5] Nicolas Pichoff, CEA Saclay, private communication.
- [6] ‘Status of the INFN High Current Proton Linac for Nuclear Waste Transmutation’, Pagani, C, Barni, D, Bellomo, G, Parodi, R, Pierini, P, in Proceedings of the XIX International Linear Accelerator Conference, August 23-28 1998, Chicago, IL.
- [7] TRACE3-D Documentation, Krandall, K, Rushtoi, D, LA-UR-97-886, Los Alamos, NM, 1987.
- [8] ‘Beam Dynamics in a High Current SC Proton Linac for Nuclear Waste Transmutation’, Bellomo, G, Pierini, P, in Proceedings of the XIX International Linear Accelerator Conference, August 23-28 1998, Chicago, IL.
- [9] The Author can be contacted by e-mail with the following address: Paolo.Pierini@mi.infn.it

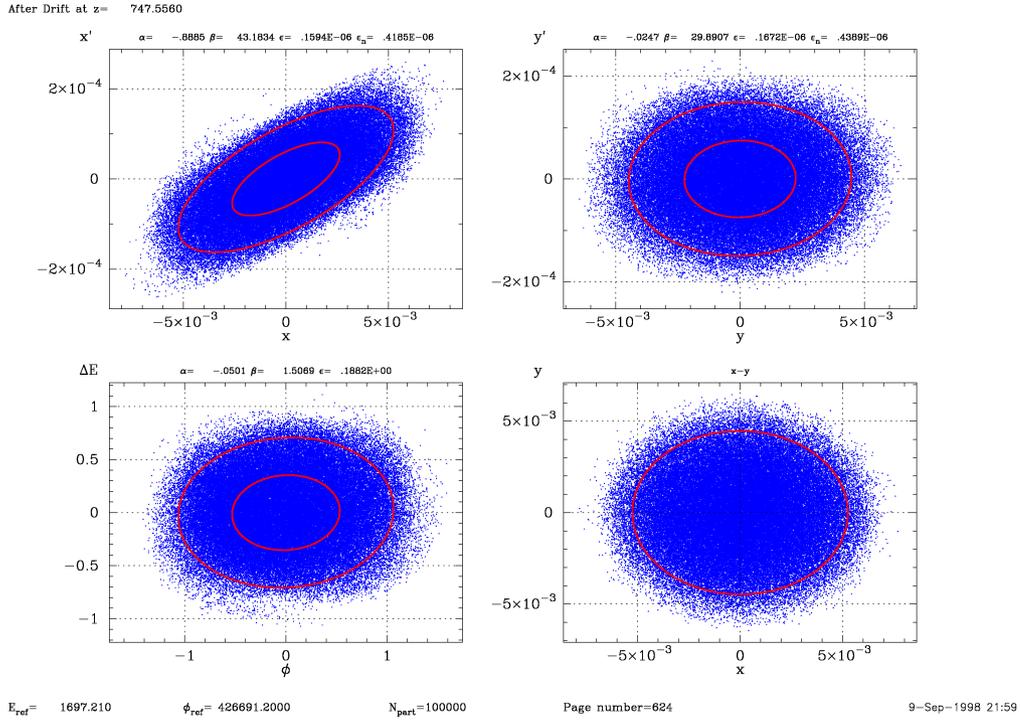


Figure 7: Phase space plots at the linac end (1.7 GeV after 750 m of linac). Upper left: horizontal plane, Upper right: vertical plane, Bottom left: longitudinal plane, Bottom right: transverse distribution. Units are m and rad for the transverse planes and deg MeV for the longitudinal. Reference case for the “adiabatically matched” beam. The ellipses shown in the phase space plots represent the rms emittance and 4 times the rms value.

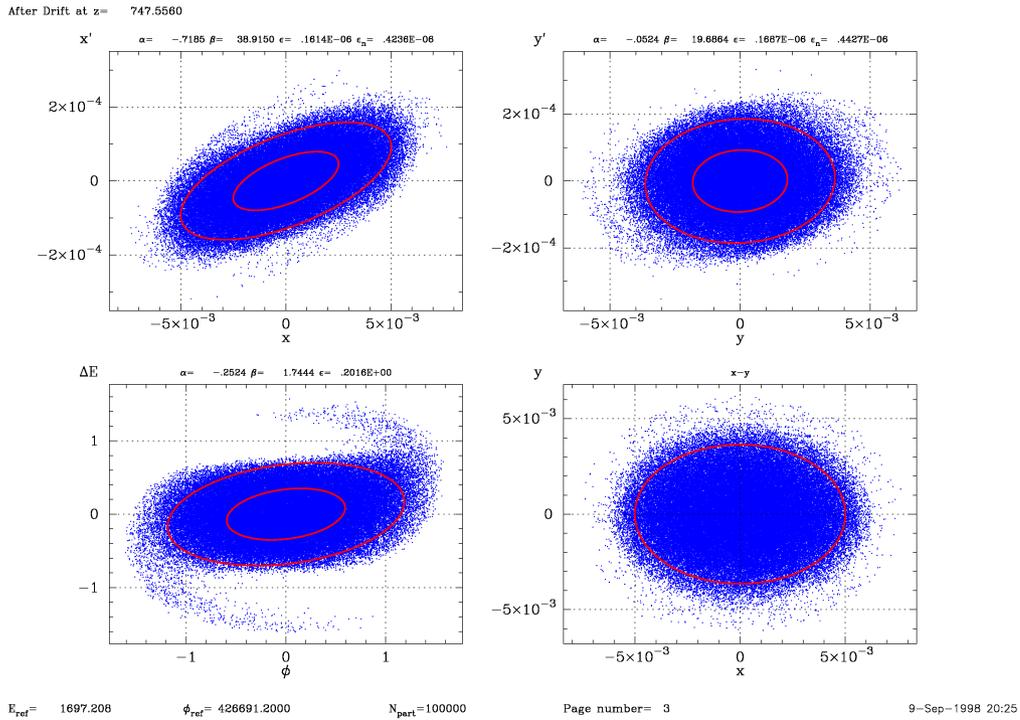


Figure 8: Same case for Figure 7 at the linac output, but for an initially mismatched beam (with a longitudinal and transverse mismatch factor of 30%).

# Simulation and Observation of the Long Time Evolution of the Longitudinal Instability in a Storage Ring

O. Boine-Frankenheim, I. Hofmann

Gesellschaft für Schwerionenforschung (GSI), Planckstr.1, 64291 Darmstadt, Germany

*Abstract*

The longitudinal instability in the long-wavelength range is observed experimentally at GSI. Two different simulation schemes (Particle-In-Cell and direct integration on a grid) are employed to understand the long time evolution of the instability.

## 1 INTRODUCTION

Above a certain threshold beam intensity the interaction of a coasting charged particle beam with the electromagnetic fields induced in ring environment can lead to self-amplification of a initial current perturbation. This longitudinal instability leads to a rapid blow-up of the longitudinal momentum spread. Longitudinal instability appears to be one of the main limiting factors in both longitudinal beam quality of intense charged particle beams and the achievement of very short bunches in ring machines. For high-energy beams in circular machines this phenomenon has been studied extensively by means of analytic theory, kinetic simulation and experiment for many years [1, 2, 3]. Several studies focused on the ‘overshoot-phenomenon’, the dependence of the final phase space blow up on the initial momentum spread [4, 5]. In the framework of the heavy ion fusion driver study [6], the high intensity upgrade at GSI [7] and possible high current proton storage and buncher rings for different applications new interest arose in the effect of space charge on the longitudinal instability. The evolution of the instability can be effected crucially by the presence of space charge, this was first pointed out by I. Hofmann [8] who showed by means of kinetic simulations that the destructive effect of the longitudinal instability in the short-wavelength regime (‘microwave’ instability) on a space charge dominated beam is suppressed. This stabilization was addressed to the coupling between higher modes up the the cut-off frequency  $c/b$  ( $b$  pipe radius,  $c$  speed of light) introduced by space charge. Experimental efforts at GSI focused on the instability in the long-wavelength range (wavelengths much longer than the pipe diameter) caused by the resistive impedance of the RF cavities in the Heavy Ion Storage Ring ESR and Heavy Ion Synchrotron SIS. In the long-wavelength range a broader spectrum of possible collective excitations can be expected than in the short-wavelength range. In a recent storage ring experiment at GSI with a space charge dominated ion beam it was discovered that longitudinal instability leads to long living coherent structures on the beam [10]. Obviously the instability excites a nonlinear wave structure, similar to BGK-like (Bernstein-Green-Kruskal) waves [11] caused by non-linear Landau damping [12] in

ideal plasmas.

Other experimental observations of the longitudinal instability at GSI show the recurrence of the self-bunching amplitudes and ‘spike’ like structures on the current profile. In order to understand the phenomena observed, kinetic simulation codes, based on the Vlasov-Fokker-Planck equation, were developed. The simulation of the long time evolution initiated by the instability is complicated by the presence of space charge. It is a well known fact that space charge together with the noise inherent to typical Particle-In-Cell (PIC) codes leads to artificial heating in these codes [9]. Therefore a direct ‘noise-free’ integration method was implemented, in addition to a usual PIC code, in order to study the long time evolution of the observables.

## 2 KINETIC DESCRIPTION

Let  $\dot{\theta}_0 = \omega_0$  be the angular frequency,  $v_0$  the velocity of the synchronous particle and  $\dot{\theta}_0 + \Delta\dot{\theta}$ , and  $v_0 + v_z$  the angular frequency and velocity of a nonsynchronous particle in a ring machine of the radius  $R$ . The coordinates in a system co-moving with the synchronous particle are

$$z = R\Delta\theta, \quad v_z = \dot{z} = R\Delta\dot{\theta}. \quad (1)$$

The kinetic description is based on the Vlasov equation for the distribution function  $f(z, v_z, t)$  written in the frame comoving with the beam

$$\frac{\partial f}{\partial t} + v_z \frac{\partial f}{\partial z} - \frac{q\eta}{\gamma_0 m} E_z \frac{\partial f}{\partial v_z} = 0, \quad (2)$$

with the frequency slip factor  $\eta = 1/\gamma_t^2 - 1/\gamma^2$ , the relativistic factor  $\gamma_0 = 1/(1 - \beta_0^2)^{1/2}$ ,  $\beta_0 = v_0/c$ , the total longitudinal electric field  $E_z(z, t)$ , the ion charge  $q$  and  $m$  the mass. The line density is given by

$$\rho_L(z, t) = q \int_{-\infty}^{\infty} f dv_z \quad (3)$$

Perturbations on the beam current cause electric fields, acting back on the beam. The coupling between the beam and the ring environment is described in terms of the ring impedance  $Z_{||}(\omega)$

$$E_z(\omega) = -\frac{1}{2\pi R} Z_{||}(\omega) I(\omega) \quad (4)$$

If we deal with coasting beams only, it is sufficient to only account for the current amplitudes  $I_n$  at harmonics  $\omega_n = n\omega_0$  of the revolution frequency  $\omega_0$

$$I_n = \beta_0 c \int_0^L \rho_L(z, t) \exp(inz/R) dz. \quad (5)$$

Eq. 4 then simplifies to

$$E_{nz} = -\frac{1}{2\pi R} Z_n I_n. \quad (6)$$

We consider a narrow band impedance only, with an eigenfrequency tuned exactly to a fixed harmonic number  $h$

$$Z_{cn} = R_s \delta(h - n). \quad (7)$$

The electric field induced in an rf cavity and acting back on the beam is called the ‘beam loading’ field. The space charge electric field acts in addition to the ‘beam loading’ field. Below the cut-off wavelength  $2\pi b$  (pipe radius  $b$ ) the longitudinal electric space charge impedance is given by

$$Z_{sn} = -\frac{ingZ_0}{2\beta_0\gamma_0^2} \quad (8)$$

with  $Z_0 = 377 \Omega$  and the factor  $g = 0.5 + 2 \ln(b/a)$  (beam radius  $a$ ). We can account for electron cooling and intra-beam-scattering (IBS) by adding a Fokker-Planck term to the RHS of the Vlasov equation (Eq. 2)

$$\frac{df}{dt} = \frac{\partial}{\partial v_z} \left( \frac{F_z}{m} f \right) + D \frac{\partial^2 f}{\partial v_z^2}, \quad (9)$$

with the electron cooling force  $F_z$  and the diffusion coefficient  $D$ .

### 3 REVIEW OF THE LINEARIZED THEORY

It is convenient to introduce a scaled impedance according to

$$V + iU = \frac{2I_0q}{\pi mc^2 \beta_0^2 \gamma_0 |\eta| \sigma^2} \left( \text{Re} \left( \frac{Z}{n} \right) + i \text{Im} \left( \frac{Z}{n} \right) \right) \quad (10)$$

with the momentum spread  $\sigma$  and the average current  $I_0$ . For a beam with negligible momentum width or  $|U+iV| \gg 1$  the coherent frequency shift  $\Delta\omega = \omega - n\omega_0$  of a mode with harmonic number  $n$  is [13]

$$\left( \frac{\Delta\omega}{n\delta\omega} \right)^2 = U - iV \quad (11)$$

with  $\delta\omega/\omega_0 = -0.5\eta\sigma$ . The instability growth rate following from Eq.11 is

$$\text{Im}(\Delta\omega) = n\delta\omega \frac{1}{2} (\sqrt{U^2 + V^2} - U) \quad (12)$$

For a space charge dominated beam we require  $|U_{sc}| > 1$ . In all cases where  $|U + iV| \lesssim 1$  one must use the appropriate velocity distribution function to evaluate the dispersion function for  $\Delta\omega$  in order to determine the effect of linear Landau damping on the instability. Landau damping provides a region of stability for sufficiently small  $|U+iV|$ . The most conservative stability criterion is represented by the Keil-Schnell circle (F of the order of unity) [13]

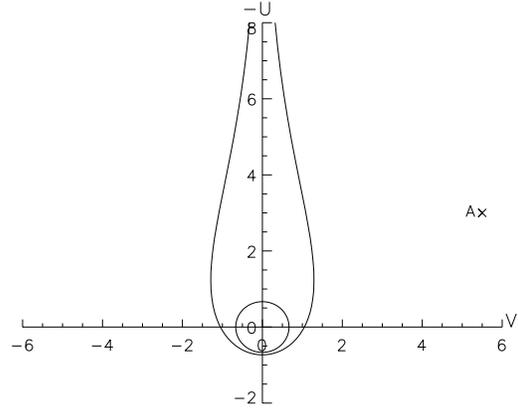


Figure 1: Stability curve for a Gaussian velocity distribution together with the Keil-Schnell stability circle. A: ESR experiment with a  $C^{6+}$  beam ( $\sigma_0 = 1.1 \cdot 10^{-5}$ , 340 MeV/u, 0.3 mA) interacting with one RF cavity ( $R_s=1300 \Omega$ ) tuned to  $n = 1$ .

$$|U + iV| = \frac{2}{3} F \quad (13)$$

The threshold momentum spread following from the Keil-Schnell criterion is

$$\sigma_{\text{th,KS}}^2 = \frac{qI_0}{Fmc^2 \beta_0^2 \gamma_0 |\eta|} \left| \frac{Z}{n} \right| \quad (14)$$

Fig. 1 shows the stability diagram for a Gaussian velocity distribution together with the Keil-Schnell circle. The operating point A in Fig. 1 corresponds to the initial beam parameters used in the ESR experiment [10]. There  $1.65 \times 10^8 C^{6+}$  ions at 340 MeV/u were cooled to  $\sigma = 1.1 \times 10^{-5}$ , interacting with one RF cavity ( $R_s=1300 \Omega$ ) tuned close to  $n = 1$ . The space charge impedance is  $Z_{sc} = -i700 \Omega$ .

### 4 NUMERICAL INTEGRATION SCHEMES

Linear theory can provide us with the initial rise time of the instability. In order to make reliable predictions of inherently nonlinear experimental observables like the final momentum spread and the self-bunching amplitudes we have to go beyond linear theory and integrate the Vlasov-Fokker-Planck equation, together with the self-consistent electric fields numerically. Two different numerical integration schemes are employed, which are described in the following.

#### 4.1 Particle-In-Cell Method

Our PIC code is a general tool to study the longitudinal dynamics of space charge dominated coasting or bunched

beams under the influence of the ring environment represented by a general impedance  $Z_{\parallel}(\omega)$  and arbitrary external RF fields.

In the PIC scheme the beam is represented by a number of macro-particles interacting via space charge and the ring impedances. Due to the random character of this approach the fluctuation level of the observables in time and space often exceeds the fluctuation level of the real beam. The fluctuation inherent to the PIC simulation can cause artificial heating, if the number of macro-particles per cell is too low or the resolution of the grid is not sufficient. On the other side we can take advantage of the simulation noise and use the PIC code to predict the structure of the experimentally observable Schottky noise spectrum.

PIC codes are well established in theoretical plasma physics, where they are used routinely to study various non-linear kinetic phenomena [9]. In one-dimensional applications one usually requires the grid spacing to resolve the Debye length ( $\Delta z \lesssim \lambda_D$ ), otherwise numerical instabilities cause substantial artificial heating [9]. In an ideal plasma the phase velocity of electron plasma waves decreases for lower wave lengths. Waves length of the order of the Debye length correspond to phase velocities close to the thermal electron speed and are strongly Landau damped. Therefore the Debye length can be regarded as the lowest scale length in a ideal plasma. For longitudinal space charge waves on a beam in a conducting pipe below the cut-off frequency the coherent frequency shift depends linearly on the harmonic number (Eq. 11). Therefore the phase velocity is independent on the wave length and all wave lengths down to  $2\pi b$  need to be resolved, in order to avoid unphysical heating. For the time step one has to choose  $\Delta t < \Delta z/v_{\max}$ , with  $v_{\max}$  being the maximum velocity to be resolved.

In our code macro-particles are loaded randomly in  $(x, y, z)$  phase space. For the purpose of purely longitudinal studies it is usually sufficient to evolve the longitudinal particle coordinates only and leave the transverse coordinates unchanged. The particle pusher is divided into several steps. First the position of each particle in space is updated:

$$\mathbf{r}^{t+\Delta t/2} - \mathbf{r}^{t-\Delta t/2} = \mathbf{v}^t \Delta t \quad (15)$$

The updated positions are interpolated on a grid in order to form the charge density  $\rho(r, z)$ . The space charge field  $\mathbf{E}_s(r, z)$  is calculated from the charge density and Poisson's equation assuming cylindrical symmetry

$$\varepsilon_0 \nabla \mathbf{E}_s^{t+\Delta t/2} = \rho^{t+\Delta t/2} \quad (16)$$

The beam current in the frequency domain  $I(\omega)$  is calculated from the time domain by using the time history of previous current values. We can then use the general form (4) to obtain the beam loading field in the frequency domain

$$E_{bz}(\omega) = -\frac{1}{2\pi R} Z_{\parallel}(\omega) I(\omega) \quad (17)$$

In the code external fields can be added at this point to perform various bunch manipulations. The total electric field, including the focusing fields, defined on the grid  $\mathbf{E}^{t+\Delta t/2}$  must be interpolated back to the particle positions to update the longitudinal and the transverse velocities

$$v_z^{t+\Delta t} - v_z^t = \frac{q\eta}{\gamma_0 m} E_z^{t+\Delta t/2} \Delta t \quad (18)$$

$$+ \frac{q}{\gamma_0^3 m} F_z \Delta t + \sqrt{3D_z \Delta t} R_1$$

$$v_{x,y}^{t+\Delta t} - v_{x,y}^t = \frac{q}{\gamma_0 m} E_{x,y}^{t+\Delta t/2} \Delta t \quad (19)$$

$$+ \frac{q}{\gamma_0 m} F_{x,y} \Delta t + \sqrt{\frac{3}{2} D_{x,y} \Delta t} R_{2,3}$$

Here we added an electron cooling force  $\mathbf{F}(\mathbf{v})$  and intra-beam scattering following [14].  $D_{x,y,x}$  are the diffusion constants and  $R_{1,2,3}$  are random numbers.

The simulation noise inherent to the PIC scheme has the undesired effect that it artificially heats the beam. On the other side we can take advantage of this simulation noise to predict the Schottky noise spectrum, which is an important experimental observable.

#### 4.2 Direct 'Noise-Free' Integration on a Grid

A more elegant, but also more elaborate way to solve the Vlasov-Fokker-Planck equation is the direct integration on a grid in longitudinal phase space  $(z, v_z)$ . This approach is 'noise-free', if we disregard the computer noise for the moment. The direct integration has the advantage of equally good resolution everywhere on the grid, whereas in the PIC code it can happen that there are not enough macro-particles in a certain phase space region to resolve a kinetic phenomena (see for example [15]).

In our integration scheme the full time step is split in several steps. First the Vlasov part is evolved by means of the well know time splitting scheme described in [16]. Let  $\Delta t$  be the simulation time step, then the splitting scheme for the Vlasov part is:

Step 1.

$$f^*(z, v_z, t + \Delta t) = f(z - v_z \Delta t/2, v_z, t) \quad (20)$$

Step 2.

$$f^{**}(z, v_z, t + \Delta t) = f^*(z, v_z + \frac{q\eta}{\gamma_0 m} E^* \Delta t, t + \Delta t)$$

Step 3.

$$f(z, v_z, t + \Delta t) = f^{**}(z - v_z \Delta t/2, v_z, t)$$

The interpolation is done by means of cubic splines. The space charge field and the beam loading field are updated using the fast Fourier transformed  $\rho_L^*(z)$  and Eq. 6

$$E_n^* = -\frac{v_0}{2\pi R} Z_n \rho_{Ln}^* \quad (21)$$

In the case of the Vlasov equation, that means in the ‘collision-free’ case,  $f(z, v_z, t + \Delta t)$  is the final distribution function. For the Vlasov-Fokker-Planck equation we still have correct  $f$  for the friction and diffusion terms. Let  $f_j^t$  be the distribution function resulting from the Vlasov step at a grid point  $v_{zj} = j\Delta v$  along the velocity axis. The final distribution function  $f_j^{t+\Delta t}$  is calculated by using the time implicit scheme. This completes the time step.

$$f_j^{t+\Delta t} = f_j^t + \frac{\Delta t}{2m\Delta v} (F_{j+1}f_{j+1}^{t+\Delta t} - F_{j-1}f_{j-1}^{t+\Delta t}) \quad (22)$$

$$+ \frac{\Delta t D}{(\Delta v)^2} (f_{j+1}^{t+\Delta t} - 2f_j^{t+\Delta t} + f_{j-1}^{t+\Delta t})$$

## 5 SIMULATION OF THE LONGITUDINAL INSTABILITY

The instability growth time and the long time evolution of the self-bunching amplitudes were measured in an experiment in the ESR [10]. After the cooling of the beam the eigenfrequency of the RF cavity was tuned near to the revolution frequency, resulting in an operation point close to A in Fig. 1. The beam current signal from a longitudinal beam monitor was sampled with a high resolution and stored over 1 s. The instability growth times as a function of the cavity detuning as well as the self-bunching profiles up to the first wave steepening phase were found in good agreement with PIC simulation results [10].

Our recent work focuses on the collective dynamics governing the long time evolution of the instability. The main tool is the Vlasov-Fokker-Planck simulation described earlier. First we reconsider the ESR experiment. In the simulation we start from the operating point A in Fig. 1, assuming a initial Maxwellian distribution function. The grid size chosen is  $N_z \times N_v = 512 \times 200$ . In the simulation we ignore the residual RF voltage present in the experiment. Therefore the instability rise time will be slightly lower than in the experiment. The cooling time chosen is 400 ms, which is much longer than the instability rise time. The initial equilibrium momentum spread together with the cooling time gives us the equilibrium IBS diffusion coefficient.

Fig. 2 shows the time evolution of the line density and the velocity distribution. In agreement with the experimental observations the simulation shows a remaining coherent signal on the beam. The velocity distribution does not converge to a stationary function either, but shows remaining fluctuations of the characteristic ‘foot’ towards lower velocities.

The simulation enables us to look into the details of the distribution function in the longitudinal phase space. In Fig. 3 snap shots of the distribution function together with the line density and the velocity distribution are shown. The instability first saturates by trapping particles in the self-excited potential. The resulting hole structure has a life time of about 100 ms, before it starts to smooth out due to intra-beam scattering. The excited hole

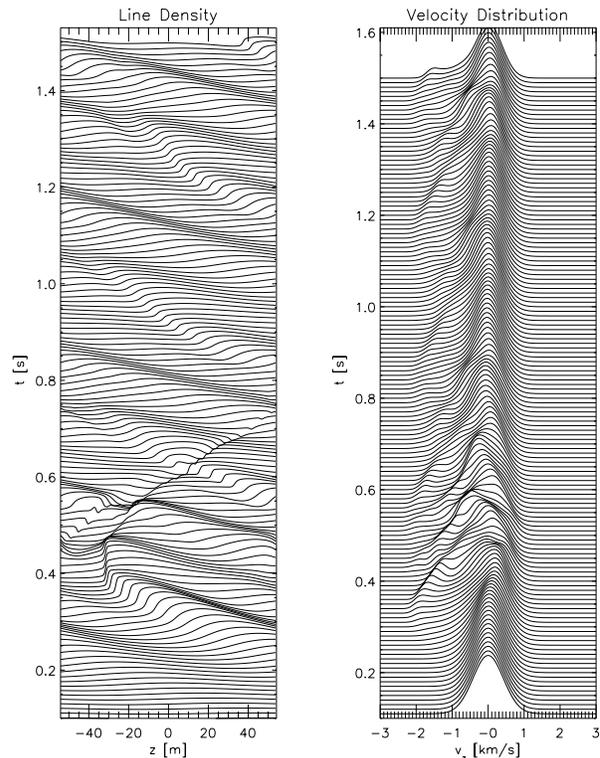


Figure 2: Time evolution of the line density and velocity distribution obtained from the Vlasov-Fokker-Planck simulation of a  $C^{6+}$  beam in the ESR interacting with one RF cavity at  $n=1$ .

structure can be regarded as a quasi-stationary, traveling Bernstein-Greene-Kruskal (BGK) [11] wave, well known from plasma physics. Due to the presence of the resistivity a pure stationary BGK solution cannot be reached, even in the absence of IBS. The ‘holes’ cause local current perturbations that continue interacting with the resistive impedance. After the first saturation stage a new ‘hole’ in the distribution function develops (see Fig. 3).

From a simulation theory standpoint it is interesting to compare the direct integration result with the corresponding result from the PIC simulation, using  $N \approx N_z \times N_v$  macro-particles. The evolution of the averaged velocity distribution is nearly the same in both codes. Nevertheless, the PIC simulation shows the subsequent formation of hole structures, accompanied by small scale ‘bubbles’ (see Fig. 4), that cannot be observed in the Vlasov-Fokker-Planck simulation and in the experiment. Additionally it is difficult to distinguish between the effects of artificial diffusion and ‘real’ intra-beam-scattering in the PIC simulation.

## 6 SUMMARY

The longitudinal instability below transition and in the long-wavelength range was studied by means of two different kinetic simulation schemes. The results can be related to recent experimental observations in the SIS and ESR at

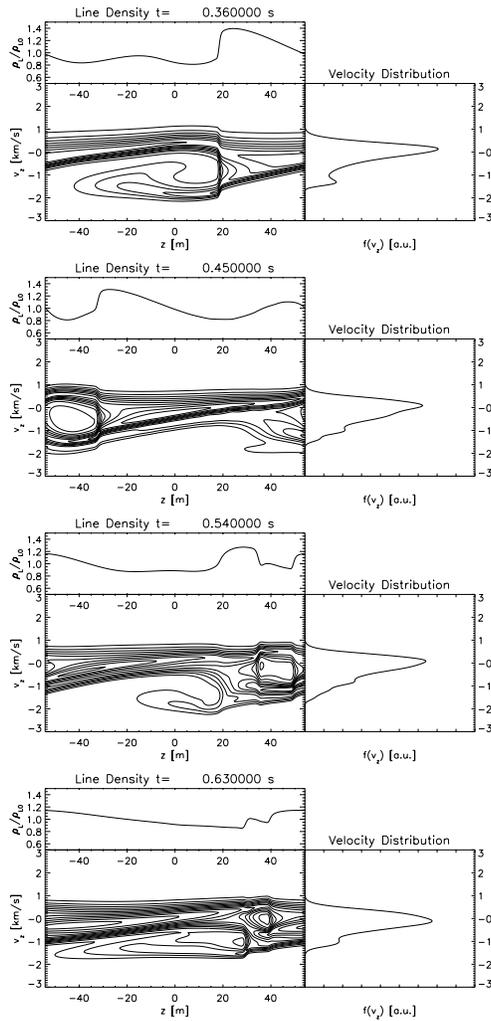


Figure 3: Contour plot of the distribution function together with the corresponding line density and velocity distribution obtained from the Vlasov-Fokker-Planck simulation of a  $C^{6+}$  beam in the ESR interacting with one RF cavity at  $n=1$ .

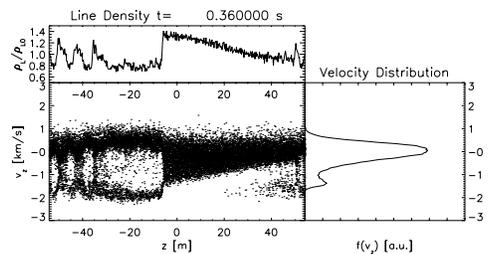


Figure 4: The distribution function together with the corresponding line density and velocity distribution obtained from the PIC simulation of a  $C^{6+}$  beam in the ESR interacting with one RF cavity at  $n=1$ .

GSI shown a remaining coherent signal on the beam. It was pointed out that for a space charge dominated beam the long time evolution of the longitudinal instability is governed by hole structures in the distribution function. These ‘holes’ show up as long-living coherent structures on the beam current and pronounced peaks in the Schottky spectrum. It was mentioned that for space charge dominated beams the more elaborate direct ‘noise-free’ integration scheme becomes more appropriate than the PIC scheme.

## 7 REFERENCES

- [1] V. K. Neil, A. M. Sessler, Rev. Sci. Instrum. **32**, 256 (1961)
- [2] B. Keil, E. Messerschmidt, Nucl. Instrum. Methods **128**, 203 (1975)
- [3] B. Zotter, P. Bramham, IEEE Trans. Nucl. Sci. **20** 830 (1973)
- [4] Y. Chin, K. Yokoya, Phys. Rev. D **28**, 2141 (1983)
- [5] S. A. Bogacz, K. Y. Ng, Phys. Rev. D **36**, 1538 (1987)
- [6] I. Hofmann, G. Plass (ed.), GSI Report **98-06** (1998)
- [7] U. Ratzinger *et al.* GSI Report **95-05** (1995)
- [8] I. Hofmann, Laser Part. Beams **3**, 1 (1985)
- [9] C.K. Birdsall, A.B. Langdon, *Plasma Physics via Computer Simulation*, IOP (1991)
- [10] G. Rumolo *et al.*, Nucl. Instrum. Methods **415**, 363 (1998)
- [11] I. B. Bernstein *et al.*, Phys. Rev. **108**, 546 (1957)
- [12] G. Manfredi, Phys. Rev. Lett. **79**, 2815 (1997)
- [13] M. Reiser, *Theory and Design of Charged Particle Beams*, Wiley (1994)
- [14] P. W. Rambo, R. J. Procassini, Phys. Plasmas **2**, 3130 (1995)
- [15] O. Boine-Frankenheim, J. D’Avanzo, Phys. Plasmas **3**, 792 (1996)
- [16] S. G. Chen, G. Knorr, J. Comput. Phys. **22**, 330 (1976)

# ANALYTIC MODEL OF HALO FORMATION

R.L. Gluckstern

Physics Department, University of Maryland, College Park, MD 20742

## Abstract

An analytical model for halo formation has been constructed based on a beam in a focusing channel which is “breathing” due to a mismatch with the channel. For a 2-D KV beam, an integral of motion can be obtained assuming the dominance of the parametric resonance (breathing mode frequency is twice the frequency of individual particles within the beam). These results correspond to the “peanut diagram” in particle phase space in all respects. The model is then extended to other 2-D distributions, as well as 3-D distributions involving both longitudinal and transverse breathing modes. Numerical simulations are then used to determine the behavior of the longitudinal and transverse halos which occur, and their dependence on the initial phase space distribution, the bunch charge and shape, and the amplitude of the mismatch.

## 1 INTRODUCTION

The need for high current in a variety of new accelerator applications has focused a great deal of attention on understanding the phenomenon of halo formation in ion beams, which can cause excessive radiation of the accelerator. This understanding requires both an analytical model which explains available observations as well as computer simulations to verify both the assumptions of the model and its predictions. Fortunately, the availability of high speed computing, whose application to this problem is one of the major foci of this conference, has allowed rapid progress in the understanding of halo formation in high current linear accelerators.

## 2 2-D MODEL

Early attention was devoted to the analytic study of 2-D round beams in a continuous focusing channel. In particular, the KV distribution [1], a hyperspherical shell in the 4-D phase space with the self-consistent [2] distribution

$$f(H) = N\delta(H_0 - H), \quad (1)$$

where

$$H = \frac{mv^2}{2} + \frac{kr^2}{2} + e\Phi_{sc}(\mathbf{r}), \quad (2)$$

had the useful features of a uniform charge density within the beam, and uniform density in the  $x$  and  $y$  phase space projections. Here  $H_0$  and  $N$  are constants,  $k$  is the constant external focusing gradient, and  $e\Phi_{sc}(\mathbf{r})$  is the potential energy at  $\mathbf{r}$  due to space charge.

Use of the equation for the beam envelope [3] permitted the analytic description of a “breathing” beam, in which the

charge density oscillated between too tight and too loose a match to the external focusing force. These oscillations provided a periodic force to the ion motion, which was simple harmonic as long as the ions remained inside the beam. But for ions which traveled beyond the beam boundary, the oscillations were non-linear. In this case the ion’s non-linear motion in the presence of a periodic force allowed it to be trapped in the parametric resonance, where the breathing frequency was twice the ion oscillation frequency. The analytic model thus predicted the formation of a “halo” [4] for certain combinations of mismatch and tune depression. The numerical simulations using the “particle-core” model confirmed the validity of the models, and pointed as well to the existence of chaotic motion as the tune depression became more severe [5]-[9].

Subsequent work focused on the possible mechanism for particles escaping from the beam into the region of non-linear oscillation [10]. In addition, numerical simulations were run for other, more physical, self-consistent stationary distributions of the form

$$f(\mathbf{r}, \mathbf{v}) = N(H_0 - H)^n \quad (3)$$

with  $n = 0, 1$  [11]. These simulations exhibited the same halo structure and phase space patterns seen for the KV distribution, but with somewhat different quantitative dependence on mismatch and tune depression. The localization of the halo radius to approximately the same value predicted by the KV distribution gave the linac designers confidence that a beam pipe wall could be placed far enough from the beam to avoid intercepting the halo particles.

## 3 3-D MODEL

Attention then shifted to short 3-D beam bunches of ellipsoidal shape with  $c/a = \text{length/width ratio}$  in the range 2-4 [12, 13]. We continued our effort to study the self-consistent phase space stationary distributions of the form

$$f(\mathbf{r}, \mathbf{v}) = N(H_0 - H)^n \quad (4)$$

but this time, for  $n = -1/2$ , the differential equation for the charge density was linear and could be solved analytically [13]. In addition, for  $c/a > 2$ , the “breathing” modes could be approximately separated into transverse and longitudinal modes, each of which was capable of generating a halo. Thus the picture was of a beam bunch which, when mismatched accordingly, generated either a transverse or a longitudinal halo, or both. The signature of the longitudinal halo was the same as that of the transverse halo (a “peanut diagram” in the phase space projection). The

transverse and longitudinal mismatch and tune depression parameter space was extensively explored with numerical simulations, as you will hear in subsequent talks. But a new concern surfaced: Would the longitudinal halo permit the loss of ions from the rf bucket? Unfortunately, the bucket “walls” cannot be moved far away without increasing the length and cost of the linac. You will also hear more about these studies in a non-linear rf bucket in a subsequent talk.

Other issues involving halo formation were looked at, including equipartitioned distributions which were rms matched but not self-consistent [14]. As you will hear, these involved a rapid initial phase space redistribution, leading to a relatively small change in the parameters and extent of the halo formation due to the mismatch. In addition, they also point to the presence of a transverse-longitudinal coupling which allows either kind of halo to develop from either a transverse or longitudinal mismatch [14].

#### 4 SUMMARY

Analytic models have been developed to study halo development in both 2-D beams and 3-D beam bunches in a linac. These models suggest that the most likely explanation for the halos which have been observed and which are likely to be seen in future high current linacs involves the parametric resonance between the collective modes which describe “breathing” and the motion of individual ions. When these models are used in conjunction with multiparticle simulations involving millions of particles, which are now practical with supercomputers and parallel processing, one can have great confidence in the predictions for halo formation and emittance growth which are so crucial for the designs of high current acceleration of short beam bunches.

#### 5 REFERENCES

- [1] I.M. Kapchinsky, *Theory of Resonance Linear Accelerators*, Harwood Academic Publishers, New York, 1985.
- [2] Any distribution which is only a function of Hamiltonian will be self-consistent.
- [3] M. Reiser, *Theory and Design of Charged Particle Beams*, Wiley, New York (1994).
- [4] R.L. Gluckstern, *Phys. Rev. Letters* 73, 1247 (1994).
- [5] J.S. O’Connell, T.P. Wangler, R.S. Mills, and K.R. Crandall, in *Proceedings of the PAC93* (IEEE, Washington, DC, 1993), p. 3657.
- [6] R.A. Jameson, in ‘Frontiers of Accelerator Technology’ (Proceed. of the Joint US-CERN-Japan Intern’l School, 1994), World Scient., Singapore (1996).
- [7] J.M. Lagniel, *Nucl. Inst. Meth. A*345 (1994) 46; A345 (1994) 405.
- [8] A. Riabko, M. Ellison, X. Kang, S.Y. Lee, D. Li, J.Y. Liu, X. Pei, and L. Wang, *Phys. Rev. E* 51, 3529 (1995).
- [9] H. Okamoto and M. Ikegami, *Phys. Rev. E* 55, 4694 (1997).
- [10] R.L. Gluckstern, W-H. Cheng, and H. Ye, *Phys. Rev. Letters* 75, 2835 (1995); R.L. Gluckstern, W-H. Cheng, S.S. Kurennoy and H. Ye, *Phys. Rev. E* 54, 6788 (1996).

- [11] R.L. Gluckstern and S.S. Kurennoy, in *Proceedings of the PAC97*, Vancouver, Canada (1997), (unpublished).
- [12] J.J. Barnard and S.M. Lund (I), and S.M. Lund and J.J. Barnard (II), in *Proceedings of the 1997 Particle Accelerator Conference*, Vancouver, Canada (1997), (unpublished).
- [13] R.L. Gluckstern, A.V. Fedotov, S. Kurennoy and R. Ryne, *Phys. Rev. E* 58, 4977 (1998).
- [14] A.V. Fedotov, R.L. Gluckstern, S.S. Kurennoy and R.D. Ryne, in *Proceedings of the 1998 LINAC Conference*, Chicago, IL, USA; also Univ. of Maryland Physics Dept. Preprint 98-108 (1998).

# HALO FORMATION IN 3-D BUNCHES WITH SELF-CONSISTENT STATIONARY DISTRIBUTIONS

A.V. Fedotov, R.L. Gluckstern

Physics Department, University of Maryland, College Park, MD 20742

S.S. Kurennoy, R.D. Ryne

LANSCE Division, Los Alamos National Lab, Los Alamos, NM 87545

*Abstract*

We have constructed, analytically and numerically, a new class of self-consistent 6-D phase space stationary distributions. Stationary distributions allow us to study the halo development mechanism without being obscured by beam redistribution and its effect on halo formation. The beam is then mismatched longitudinally and/or transversely, and we explore the formation of longitudinal and transverse halos in 3-D axisymmetric beam bunches. Of particular importance is the result that, due to the coupling between longitudinal and transverse motion, a longitudinal or transverse halo is observed for a mismatch less than 10% if the mismatch in the other plane is large.

## 1 INTRODUCTION

The need for high current in a variety of new accelerator applications has focused a great deal of attention on understanding the phenomenon of halo formation in ion beams, which can cause excessive radioactivation of the accelerator. Starting in about 1991, a variety of two-dimensional (2-D) simulation studies [1, 2] have led to the conclusion that halos are formed when a beam is mismatched to a focusing channel, exciting some sort of collective oscillation(s) of the beam which are in resonance with the non-linear oscillation of individual ions.

Most of the simulations studies start with rms matched beams which are *not* stationary solutions of the Vlasov equation (See for example [3]). As a result, the initial beam undergoes some sort of redistribution in phase space, masking the possible development of halos. Our effort has been devoted to populating a stationary distribution in phase space, in the hope that the halo development mechanism can be studied without being obscured by the “relaxation” of the beam in phase space. We have particularly studied initial distributions which are stationary by virtue of being a function only of the Hamiltonian [4, 5].

It is clear that a realistic treatment of halo formation must take into account 3-D beam bunches and 6-D phase space distributions. Recently, Barnard and Lund [6] performed numerical studies with a 3-D beam bunch using the particle-core model, drawing attention to the existence and importance of a longitudinal halo for a spheroidal bunch. However, all studies based on the particle-core model do not address the question of whether halo formation is influenced by the density redistribution which follows for a non-stationary beam, even if it is rms matched [3]. In fact, halo

formation in 2-D due to the redistribution process in rms matched beams was shown, for example, by Okamoto [7] and Jameson [8]. We therefore continue our effort to study the halo development mechanism in 3-D beam bunches in the absence of the redistribution process. Such an approach allows us to study the fundamental mechanism of halo formation associated with the beam mismatch. To accomplish this we have constructed, analytically and numerically, a new class of stationary 6-D phase space distributions for a spheroidal beam bunch. Our present analysis assumes smoothed linear external transverse and longitudinal restoring force gradients,  $k_z$ ,  $k_y$ ,  $k_x$ . In general, the distribution can be chosen to have an approximately ellipsoidal boundary. However, for simplicity, we treat the azimuthally symmetric case ( $k_x = k_y$ ) for which the beam bunch is approximately spheroidal. This is the focus of the present investigation. More details can be found in [9].

## 2 STATIONARY 6-D PHASE SPACE DISTRIBUTION

### 2.1 Distribution and charge density

We take for the azimuthally symmetric 6-D phase space distribution

$$f(\mathbf{x}, \mathbf{p}) = N(H_0 - H)^{-1/2} \quad (1)$$

where

$$H = k_x r^2/2 + k_z z^2/2 + e\Phi_{sc}(\mathbf{x}) + mv^2/2. \quad (2)$$

Here  $\mathbf{p} = m\mathbf{v}$ ,  $r^2 = x^2 + y^2$ , and  $k_x, k_z$  are the smoothed transverse and longitudinal restoring force gradients. The quantity  $\Phi_{sc}(\mathbf{x})$  is the electrostatic potential due to the space charge of the bunch. The distribution is normalized such that

$$\int d\mathbf{x} \int d\mathbf{p} f(\mathbf{x}, \mathbf{p}) = 1. \quad (3)$$

The charge distribution corresponding to Eq. (1) is

$$\begin{aligned} \rho(\mathbf{x}) &= Q \int d\mathbf{p} f(\mathbf{x}, \mathbf{p}) \\ &= NQm^3 \int d\mathbf{v} \left[ G(\mathbf{x}) - \frac{mv^2}{2} \right]^{-1/2}, \end{aligned} \quad (4)$$

where

$$G(\mathbf{x}) \equiv H_0 - \frac{k_x r^2}{2} - \frac{k_z z^2}{2} - e\Phi_{sc}(\mathbf{x}). \quad (5)$$

Performing the integral over  $d\mathbf{v} \equiv v^2 dv d\Omega_v$  in Eq. (4) leads to

$$\rho(\mathbf{x}) = QG(\mathbf{x}) / \int d\mathbf{x}G(\mathbf{x}), \quad (6)$$

where the normalization constant satisfies

$$2\sqrt{2}\pi^2 N m^{3/2} \int d\mathbf{x}G(\mathbf{x}) = 1. \quad (7)$$

From Eq. (5) and Poisson's equation, we write

$$\nabla^2 G(\mathbf{x}) = -k_s - e\nabla^2 \Phi_{sc} = -k_s + (e/\epsilon_0)\rho(\mathbf{x}), \quad (8)$$

where

$$k_s = 2k_x + k_z. \quad (9)$$

Using Eq. (6), we obtain the partial differential equation for  $G(\mathbf{x})$

$$\nabla^2 G(\mathbf{x}) = -k_s + \kappa^2 G(\mathbf{x}), \quad (10)$$

where

$$\kappa^2 = (eQ/\epsilon_0) / \int d\mathbf{x}G(\mathbf{x}). \quad (11)$$

The solution of Eq. (10) for an axisymmetric, spheroidal shaped bunch can most easily be written in the spherical coordinates  $R, \theta$  for which

$$z = R \cos \theta, \quad r = R \sin \theta, \quad (12)$$

as

$$G(\mathbf{x}) = (k_s/\kappa^2)g(\mathbf{x}) \quad (13)$$

where

$$g(\mathbf{x}) = 1 + \sum_{\ell=0}^{\infty} \alpha_{\ell} P_{2\ell}(\cos \theta) i_{2\ell}(\kappa R). \quad (14)$$

Here  $P_{2\ell}(\cos \theta)$  are the even (fore-aft symmetric) Legendre polynomials and  $i_{2\ell}(\kappa R)$  are the spherical Bessel functions (regular at  $\kappa R = 0$ ) of imaginary argument.

Since  $g(\mathbf{x})$  is proportional to the charge density, the edge of the bunch is defined as the border  $g(\mathbf{x}) = 0$ , closest to the origin. We therefore choose the  $\alpha_{\ell}$ 's so that the surface of the bunch reproduces, as closely as possible, the ellipsoidal surface.

We also note that  $m\langle \dot{x}^2 \rangle = m\langle \dot{y}^2 \rangle = m\langle \dot{z}^2 \rangle = m\langle v^2 \rangle/3$  because  $H$  depends only on  $v^2$  and  $\mathbf{x}$ . Thus our choice of a stationary distribution of the form  $f(H)$  automatically corresponds to equipartition (equal average kinetic energy in the three spatial directions).

## 2.2 Numerical implementation

We have developed a 3-D particle-in-cell (PIC) code HALO3D to test the analytic model described above, and to explore halo formation [9]. The single-particle equations of motion are integrated using a symplectic, split-operator technique [10]. The space charge calculation uses area weighting ("Cloud-in-Cell") and implements open boundary conditions with the Hockney convolution algorithm

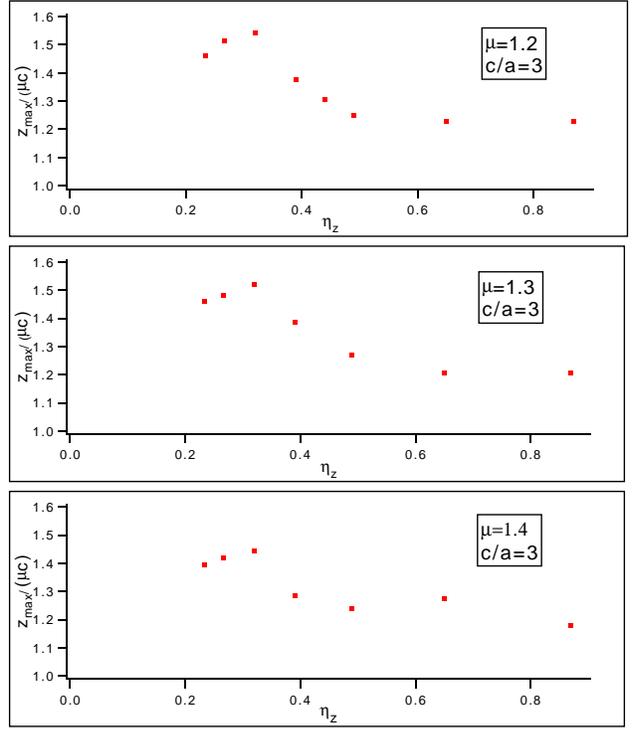


Figure 1: Longitudinal halo extent for different mismatches.

[11]. The code runs on parallel computers, and in particular, the space charge calculation has been optimized for parallel platforms using the Ferrell-Bertschinger method [12]. Some details about the code can be found in [13]. Details about effects of non-linear RF fields can be found in [14].

We initially populate the 6-D phase space according to Eq. (1), and then mismatch the  $x, y, z$  coordinates by factors  $\mu_x = \mu_y = 1 + \delta a/a$ ,  $\mu_z = 1 + \delta c/c$  and the corresponding momenta by  $1/\mu_x = 1/\mu_y, 1/\mu_z$ , with  $a, c$  being the minor and major semiaxes of our spheroidal bunch, respectively.

## 3 ORBIT SIMULATIONS

### 3.1 Longitudinal halo

Due to the fact that longitudinal tune depression is always less than the transverse one for elongated equipartitioned bunches the longitudinal halo is our primary focus. An important quantity is the ratio of the halo radius to that of the matched distribution. We performed a systematic study for different  $c/a$  and mismatch factors in the range of interest [15], by looking at the halo extent at the time when the beam comes to a roughly saturated state after the development of a halo. Our new result is the dependence of the halo extent on tune depression shown in Fig. 1 for  $c/a = 3$  and mismatch parameters  $\mu = 1.2, 1.3, 1.4$  being the same in all directions  $x, y, z$ . One sees a significant increase in halo extent for severe tune depressions. In addition the halo extent clearly depends on the mismatch parameter. The

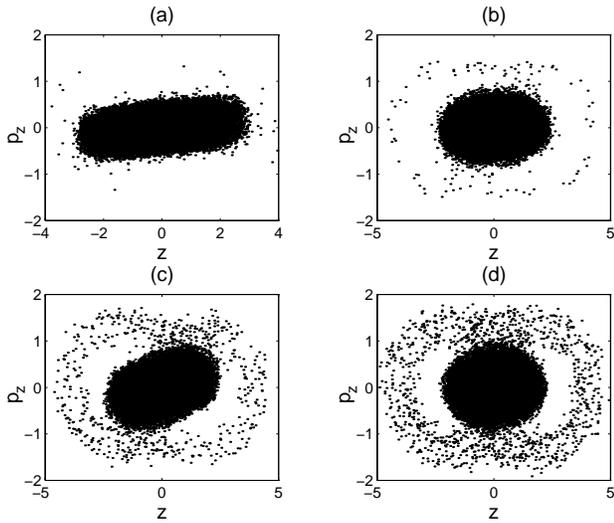


Figure 2: Dependence of halo intensity on the mismatch for  $c/a = 3$ ,  $\eta_x = 0.65$ ,  $\eta_z = 0.49$  (with 32,768 particles plotted). a)  $\mu = 1.1$ . b)  $\mu = 1.2$ . c)  $\mu = 1.3$ . d)  $\mu = 1.4$ .

approximately linear dependence of the halo extent on the mismatch factor  $\mu$  indicates that a serious effort should be made to match the beam to the channel as accurately as possible. Similar investigation for other  $c/a$  can be found in [9].

Simulation results show that the halo intensity (roughly defined as the fraction of particles outside the core in phase space) depends primarily on the mismatch. Figure 2 presents the phase space diagram (with only 32,768 particles plotted) after the halos have stabilized, for  $\eta_x = 0.65$ ,  $\eta_z = 0.49$  ( $c/a = 3$ ) with several mismatches  $\mu = 1.1, 1.2, 1.3, 1.4$ . Severe mismatches lead to several percent of the particles in the halo, which is clearly outside acceptable limits. In Fig. 3 we present the phase space diagram for different tune depressions  $\eta_z = 0.87, 0.65, 0.49, 0.32$  with  $\mu = 1.2$  ( $c/a = 3$ ) for which the fraction of particles in the halo is about 0.5 %. No significant dependence of halo intensity on the tune depression is seen. However, for tune depression  $\eta_z \leq 0.4$  the clear peanut diagram in the longitudinal phase space now has a chaotic behavior.

One more important feature is how fast the halo develops. We first make the observation that for comparable mismatches the longitudinal halo develops much faster than the transverse halo when the mismatches and/or tune depressions are not severe. Such behavior simply occurs because for fixed charge we have  $\eta_z < \eta_x$  for elongated equipartitioned bunches. For severe mismatches and/or tune depressions both the longitudinal and transverse halos develop very quickly. A typical picture is shown in Fig. 4.

Of particular interest is the clear dependence on tune depression. Specifically, for more severe tune depression the halo starts to develop earlier as can be seen in Fig. 5 where the development of the halo is shown for  $c/a = 3$ ,  $\mu = 1.2$  and different tune depressions.

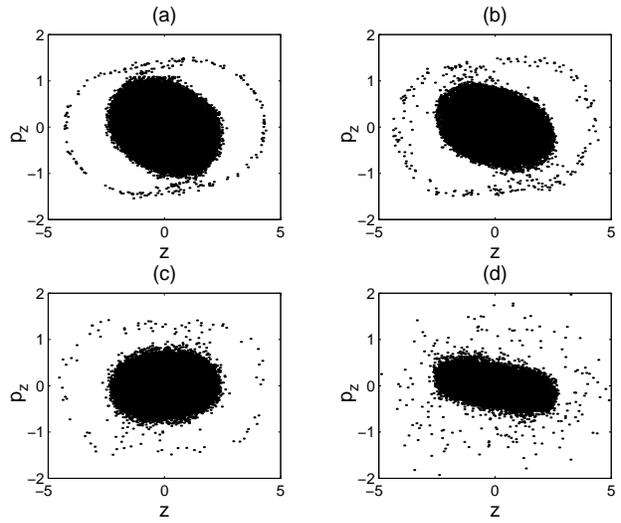


Figure 3: Dependence of halo intensity on tune depression for  $c/a = 3$ ,  $\mu = 1.2$ . a)  $\eta_z = 0.87$ . b)  $\eta_z = 0.65$ . c)  $\eta_z = 0.49$ . d)  $\eta_z = 0.32$ .

Another important characteristic of the longitudinal halo is its dependence on the mismatch when there is no mismatch in the radial direction. The number of particles in the halo drops dramatically with  $\mu_z$ . In fact, we see no halo for  $\mu_z < 1.2$  ( $< 20\%$  longitudinal mismatch). Note that the situation changes when the effect of coupling is significant.

### 3.2 Transverse halo

The transverse halo closely duplicates all the features observed for non-linear stationary distributions in 2-D simulations [16]. The agreement between 2-D and 3-D simulations is very good. The only two significant differences seen are related to the rate of halo development. In the present 3-D simulations there is a clear dependence on the tune depression which was not the case in the corresponding 2-D simulations [16]. The second difference is that the transverse halo in the 3-D simulations develops significantly faster than in 2-D for comparable mismatches and tune depressions. More details can be found in [9].

### 3.3 Coupling effects

In performing 3-D simulations we encounter halo formation in a beam bunch, where we clearly see coupling between the longitudinal and transverse motion. It was already noted [9] that due to the coupling between  $r$  and  $z$ , a transverse or longitudinal halo is observed even for a very small mismatch (less than 10%) as long as there is a significant mismatch in the other plane. Further numerical investigation of this question showed that the effect of coupling becomes extremely important for nearly spherical bunches ( $c/a \leq 2$ ) which is typical of the parameter range of interest for the APT design [15]. For example, for the short bunch with  $c/a = 2$ , with only a longitudinal initial mismatch ( $\mu_z = 1.5$ ,  $\mu_x = \mu_y = 1.0$ ), one finds particles

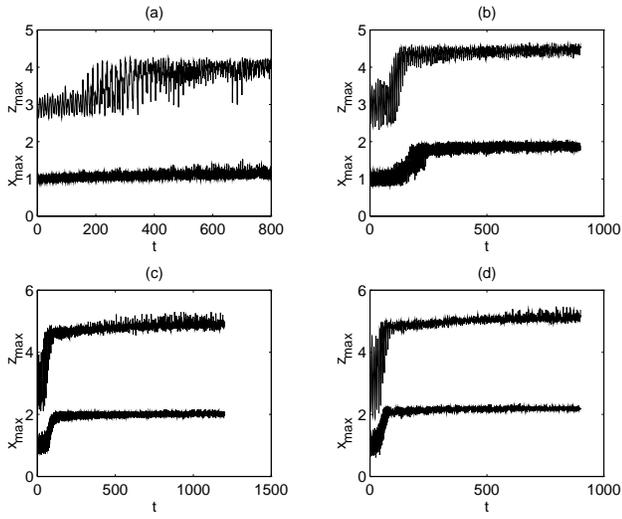


Figure 4: Halo development for comparable mismatches ( $c/a = 3$ ,  $\eta_x = 0.65$ ,  $\eta_z = 0.49$ ). a)  $\mu = 1.1$ . b)  $\mu = 1.2$ . c)  $\mu = 1.3$ . d)  $\mu = 1.4$ .

at large amplitude in both the longitudinal and transverse directions, as can be seen in Fig. 6 for the 6-D stationary distribution. Of course, the intensity of particles in the transverse halo is much smaller than it is when there is in addition a transverse initial mismatch. (In our example in Fig. 6, we have 0.05 percent of the particles in the transverse halo with zero transverse mismatch compared with several percent in the longitudinal halo.) A similar effect due to coupling was seen for the non-stationary distributions [17].

#### 4 SUMMARY

Most of the previous studies were concerned with halos in long beams. In the current work we address the question of halo formation in a beam bunch which is of particular interest for the Accelerator Production of Tritium project where relatively short bunches are proposed [15].

We have constructed, analytically and numerically, a new class of 6-D phase space stationary distributions for an azimuthally symmetric beam bunch of arbitrary charge in the shape of a prolate spheroid [9]. The stationary distribution allows us to study the halo development mechanism in 3-D beam bunches where no phase space redistribution takes place. Our choice of parameters automatically assures equipartition. In our calculations the beam remains equipartitioned through the channel. We therefore study the halo development in 3-D beams which are in thermal equilibrium, without the redistribution introduced by any equipartition process which may take place. Such an approach gives us an excellent chance to investigate the major mechanism of halo formation associated purely with the beam mismatch.

We then use a PIC code with smoothed linear external focusing forces, in which the initial stationary distribution

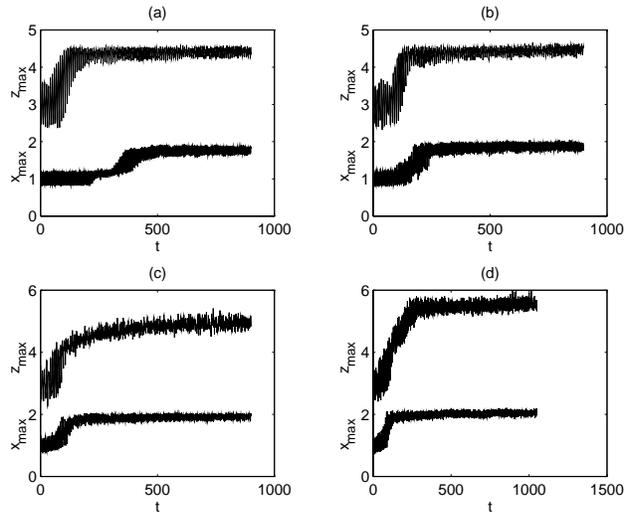


Figure 5: Dependence of the rate of halo development on tune depressions for  $c/a = 3$ ,  $\mu = 1.2$ . a)  $\eta_x = 0.79$ ,  $\eta_z = 0.65$ . b)  $\eta_x = 0.65$ ,  $\eta_z = 0.49$ . c)  $\eta_x = 0.53$ ,  $\eta_z = 0.39$ . d)  $\eta_x = 0.45$ ,  $\eta_z = 0.32$ .

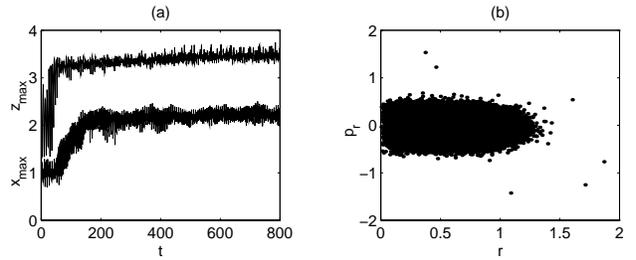


Figure 6: Coupling effect for the 6-D stationary distribution with zero transverse mismatch  $\mu_x = \mu_y = 1.0$ ,  $\mu_z = 1.5$  ( $c/a = 2$ ,  $\eta_x = 0.55$ ,  $\eta_z = 0.45$ ) a) maximum  $x$  and  $z$  b)  $r - p_r$  diagram at  $t = 800$  for particles with the angular momentum  $|L_z| < 0.1$  (with 25,000 particles plotted).

is mismatched in both the transverse and longitudinal directions, and find that both transverse and longitudinal halos can develop, depending on the choice of tune depressions and mismatches. We also found that the effect of coupling between the  $r$  and  $z$  planes is very important in the halo development mechanism and can lead to serious consequences, especially as the bunch shape becomes more spherical.

Our main conclusion is that the longitudinal halo is of great importance because it develops earlier than the transverse halo for elongated bunches with comparable longitudinal and transverse mismatches, and because it occurs even for mismatches of order 10%. In addition, the control of the longitudinal halo could be challenging if the phase width of a beam bunch in the RF bucket cannot be made sufficiently small.

After we established the parameters which lead to halo formation in 3-D beam bunches for the self-consistent 6-D phase space stationary distribution [9], we explored rms

matched distributions which are *not* self-consistent, to determine the extent to which the relatively rapid redistribution of the 6-D phase space contributes to the formation of halos [17, 18].

## 5 ACKNOWLEDGMENT

We wish to acknowledge the support of the U.S. Department of Energy, Division of High Energy Physics, and Division of Mathematical, Information, and Computational Sciences. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the DOE Office of Energy Research. We thank Tom Wangler for helpful conversations. In addition RLG and AVF wish to thank Andy Jason and the LANSCE1 group for its hospitality during part of these studies.

## 6 REFERENCES

- [1] M. Reiser, in Proceedings of the 1991 Particle Accelerator Conference, edited by L.Lizama et al. (IEEE, San Francisco, CA, 1991) p. 2497; A. Cucchetti, M. Reiser and T. Wangler, *ibid.*, p. 251; J.S. O'Connell, T.P. Wangler, R.S. Mills, and K.R. Crandall, in Proceedings of the 1993 Particle Accelerator Conference, edited by S.T. Corneliussen (IEEE, Washington, DC, 1993), p. 3657.
- [2] R.L. Gluckstern, *Phys. Rev. Letters* **73**, 1247 (1994).
- [3] M. Reiser, *Theory and Design of Charged Particle Beams*, Wiley, New York (1994).
- [4] R.L. Gluckstern, W-H. Cheng, and H. Ye, *Phys. Rev. Letters* **75**, 2835 (1995).
- [5] R.L. Gluckstern, W-H. Cheng, S.S. Kurennoy and H. Ye, *Phys Rev. E* **54**, 6788 (1996).
- [6] J.J. Barnard and S.M. Lund (I), and S.M. Lund and J.J. Barnard (II), in Proceedings of the 1997 Particle Accelerator Conference, Vancouver, Canada (1997).
- [7] H. Okamoto and M. Ikegami, *Phys. Rev. E* **55**, 4694 (1997).
- [8] R.A. Jameson, in 'Frontiers of Accelerator Technology' (Proceed. of the Joint US-CERN-Japan Intern'l School, 1994), World Scient., Singapore, 1996, p. 530-560; see also Los Alamos Report No. LA-UR-93-1209.
- [9] R.L. Gluckstern, A.V. Fedotov, S. Kurennoy and R. Ryne, *Phys. Rev. E* **58**, 4977 (1998).
- [10] E. Forest, J. Bengtsson, and M.F. Reusch, *Phys. Lett. A* **158**, 99 (1991).
- [11] R.W. Hockney and J.W. Eastwood, *Computer Simulation Using Particles*, Adam Hilger, NY (1988).
- [12] R. Ferrell and E. Bertschinger, *Int. J. Mod. Phys. C* **5**, 933 (1994).
- [13] R. Ryne, S. Habib, J. Qiang, K. Ko, Z. Li, B. McCandless, W. Mi, C. Ng, M. Saporov, V. Strinivas, Y. Sun, X. Zhan, V. Decyk, G. Golub, in Proceedings of the 1998 LINAC Conference, Chicago, IL, USA (1998).
- [14] J.J. Barnard, S.M. Lund and R.D.Ryne, in Proceedings of the 1998 LINAC Conference, Chicago, IL, USA (1998).
- [15] APT Conceptual Design Report, Los Alamos Report No. LA-UR-97-1329, 1997.
- [16] R.L. Gluckstern and S.S. Kurennoy, in Proceedings of the 1997 Particle Accelerator Conference, Vancouver, Canada (1997).
- [17] A.V. Fedotov, R.L. Gluckstern, S.S. Kurennoy and R.D. Ryne, Univ. of Maryland Physics Dept. Preprint 98-108 (1998).
- [18] A.V. Fedotov, R.L. Gluckstern, S.S. Kurennoy and R.D. Ryne, in Proceedings of the 1998 LINAC Conference, Chicago, IL, USA (1998).

# New Methods in WARP

D. P. Grote, Alex Friedman, LLNL L-645, PO Box 808, Livermore, CA 94550 and  
I. Haber, NRL, Washington DC 20375-5346\*

## Abstract

The WARP[1] code is being developed and applied to simulate the creation and propagation of the high-current, space-charge dominated beams that are required for heavy-ion driven fusion energy (HIF). New methods and capabilities have recently been introduced into WARP, a multi-dimensional particle-in-cell code developed for the study of space-charge dominated beams. We describe: (a) a 2D3V “slice” model (WARPxy) with two novel capabilities: the optional use of 3D applied fields (which can be calculated using the WARP3d solver), and an “exact” treatment of a bent beam pipe, via coordinate transformations; (b) a multigrid fieldsolver which offers internal conductors in 2D and 3D; (c) serial optimizations for cache-based machines which yielded a 20-30% speedup; and (d) a Python interface which has been developed to give the full power of a scripting language user interface in both serial and parallel computing environments.

## 1 INTRODUCTION

A heavy-ion induction accelerator is a promising candidate for a driver for inertial confinement fusion power production. In order to drive the target to ignition, a driver must produce a beam with a high current (several kAs), moderate energy (several GeVs), but very low emittance (several  $\pi$ -mm-mrads). The beams are space-charge dominated and behave like non-neutral plasmas. In order to achieve the required emittance, a thorough understanding is needed of the behavior of the beam and the effects of manipulations and errors. We have developed WARP, a multi-dimensional particle-in-cell/accelerator code, to study the physics of such high-current, low emittance beams.

WARP has been designed and developed to be a flexible simulation tool, allowing simulation of all sections of a driver at various levels of detail and dimensionality. Some of the recently implemented methods and capabilities which enhance that flexibility are described here: new physics models, computational techniques, and computer science issues.

## 2 SLICE MODEL

A slice model is a transverse model of a beam, ignoring some longitudinal effects, primarily longitudinal variations

in the space-charge fields. The slice model can be understood as a model of the behavior of the central portion of the beam (where the beam is fairly uniform axially) versus time, or as the behavior of an infinitely long beam as a function of distance. Slice simulations are not new, in fact the first simulations of beams for heavy-ion fusion were slice simulations, but the model implemented in WARPxy contains several new important features.

The slice code WARPxy was originally adapted from, and is closely coupled to, the WARP3d code, immediately giving it the full power of problem specification and diagnostics of the three-dimensional code. An example is a rich set of methods for specifying the fields of accelerator lattice elements in WARP3d that was easily adapted to work within the slice code. The close coupling allows much sharing of coding which is common to the two models, such as particle moments calculations and other diagnostics.

An important issue in beam dynamics is the presence of an axial velocity spread, which can lead to an increase in transverse emittance and directly affects the final spot size on target. A scheme was adopted in the slice code to include the axial velocity spread as well as changes in the axial velocity, such as from acceleration gaps and from axial force components of other lattice elements. Another issue is the effect of bends on the beam. It is known that when a beam with an axial velocity spread enters a bend, its emittance will increase[2]. Bends are present in the driver designs to guide the beams into the target chamber, and more significantly, in driver designs based on recirculating accelerators. The same scheme adopted for the velocity spread was adapted to include bends, where the time step size of each particle is a function of its radial position.

The underlying integration method for advancing the particles is the leap-frog method. In the slice model, the step size has a constant physical length,  $\Delta s$ , and so the time step size is dependent on the axial velocity of the individual particles. When the axial velocity of the particles changes, the time step-size must be adjusted. The algorithm used in WARPxy is to iterate over the first two steps of the split-leap frog time advance.

$$\vec{v}^{n+\frac{1}{2}} = \vec{v}^n + \frac{1}{2} \frac{\vec{F}^n}{m} \Delta t \quad (1)$$

$$\vec{x}^{n+1} = \vec{x}^n + \vec{v}^{n+\frac{1}{2}} \Delta t \quad (2)$$

Here  $\vec{x}$  and  $\vec{v}$  are the position and velocity,  $\vec{F}$  is the force, including electric and magnetic fields, and  $n$  is the time level and the step size is  $\Delta t$ . After this partial advance, the time-step size is scaled by the amount of over- or under-

---

\* Work performed under the auspices of the U.S. D.O.E. by Lawrence Livermore National Laboratory and Naval Research Laboratory under contracts W-7405-ENG-48, DE-AI02-93ER40799, and DE-AI02-94ER54232.

advance of the axial position,  $z$ .

$$\Delta t' = \Delta t \frac{\Delta s}{z^{n+1} - z^n} \quad (3)$$

The partial advance is then redone with the new time-step size. Once the iteration converges and the correct time-step size is obtained, the particle advance is completed.

$$\vec{v}^{n+1} = \vec{v}^{n+\frac{1}{2}} + \frac{1}{2} \frac{\vec{F}^{n+1}}{m} \Delta t' \quad (4)$$

Our experience is that the iteration converges very rapidly. The code always iterates all of the particles a set number of times, which can be controlled by the user.

When in a bend, the same iteration is done but with the scaling of the time-step size also accounting for the rotation of the slice frame around the bend. The scaling is done in polar coordinates relative to the center of the bend.

$$\Delta t' = \Delta t \frac{\Delta \theta}{\theta^{n+1} - \theta^n} \quad (5)$$

Here,  $\Delta \theta = \Delta s / r_{\text{bend}}$  where  $r_{\text{bend}}$  is the bend radius, and  $\theta^n$  is the axial position of the particle in the polar coordinate system at time level  $n$ . This iteration converges as rapidly as with no bends.

The calculation of the beam self-fields must also include the curvature of the bends. When the self-fields are calculated by solving Poisson's equation, the curvature terms can be treated as source terms, iterating to reach convergence. While alternative direct methods are available to solve Poisson's equation with the curvature terms, the iterative method was chosen since the same method is used in WARP3d. Also, the speed of the field solution is not critical since the two-dimensional simulation time tends to be dominated by the time of the particle advance.

The algorithm has been thoroughly tested. Numerous single particle tests have shown empirically that the method is second order in  $\Delta t$ , the same as the underlying leap-frog advance. Full beam tests have also been carried out. Two tests are described here.

As a test of the bending algorithm, a beam is propagated through a straight lattice but with the coordinate system of the simulation following s-bends. The s-bends deviate from the beam centroid path by roughly the beam radius. The simulated beam behaved as expected, reproducing the correct envelope and showing no anomalous growth in the emittance.

In another test, the behavior of a beam with an axial velocity spread as it entered a bend was examined. The slice simulations were compared with both the three-dimensional simulation and an analytic theory developed to explain the emittance growth in a bend[2]. Good agreement was found among all three. The slice simulations show oscillations in the emittance which agree in amplitude and frequency with the WARP3d simulation and theory. The simulations do show damping in the emittance oscillations that is not included in the theory. The analytic theory does give asymptotic limits on the emittance growth and these agree with the simulation results.

### 3 MULTIGRID POISSON SOLVER

The WARP code uses an electrostatic model of the self-field of the beams which requires solving Poisson's equation to calculate the potential on a Cartesian mesh. The code now has three primary Poisson solvers. The fastest is an FFT based solver which does sine-sine-periodic FFT's to model an infinitely long conducting square pipe. Capacity matrix methods can be used to include simple boundaries such as a round conducting pipe. With more complex conductor geometry, however, the required matrix rapidly becomes too large, and so an iterative method, successive over-relaxation (SOR), was implemented for such cases. The SOR method allows inclusion of arbitrarily complex conductor geometry with much less penalty. Unfortunately though, for the typical mesh sizes used, the SOR method is roughly ten times slower than the FFT method without any conductors, and becomes comparatively worse with larger mesh sizes. In the three-dimensional simulations using the SOR method, the total simulation time was dominated by the Poisson solve. Because of the computational time required to generate large capacity matrices or to use the SOR method, the grid resolution and propagation distance has been limited when using complex conductor geometry.

A third method, the multigrid method[3], has recently been implemented. It promises faster solution while having a small penalty for complex conductor geometry. The multigrid method for solving Poisson's equation on a Cartesian mesh was adapted to include internal conductors. The SOR solver is used to iteratively refine the error on each of the coarse grid levels, allowing use of the same techniques for applying internal conductors which were developed for the full SOR solver. The two techniques used are forcing of the potential inside of the conductor to the desired value each iteration, and a subgrid-scale resolution method in which the finite difference form of Poisson's equation is modified for points within one grid cell outside the surface of the conductors to explicitly include the location of the surface[4].

In order to achieve the rapid convergence of the multigrid method, the conducting boundary conditions must be applied at all levels of coarseness. So, for each level, a list of the grid points which are affected by a conductor is required. For the points inside of the conductor, this offers no difficulty since only a simple check is needed to determine whether the points on the finest grid that are inside of a conductor are on the coarse grid.

The subgrid-scale resolution technique, though, requires additional data. On the coarse grid levels, there will be points within one coarse grid cell of the conductor surface that were more grid cells away on the finest grid. Those points must be included. To generate this data, an algorithm was developed that begins with the subgrid-scale data for the finest grid and scans the grid at each of the coarse levels to gather the data for the coarse levels. For each of the points in the total list, the lowest coarseness level at which a point is on a grid is saved. So, at each level of

coarseness, a point must pass two tests to determine if it is near a conductor: the coarseness level at which the point is on a grid must be less than or equal to the current coarseness level, and the point must be within one grid cell of a conductor.

The multigrid method has the same scaling of operation count with the number of grid cells as the FFT. Timings, see Table 1, show that the FFT is about three to four times faster for all of the grid sizes examined. While the inclusion of conductors does slow the solver down, the solve times are still far below that of the original SOR method. The increase in speed will allow simulations with higher resolution and longer propagation distances than was practical with the original SOR solver.

Table 1. Timings of the field solvers on a single process of a Cray J90. All times are in seconds. The numbers in parenthesis are the number of iterations required to reach the desired convergence.

	Multigrid	FFT	SOR
64x64x64	1.6 (8)	0.41	4.7 (170)
128x128x128	10.6 (8)	2.9	74.0 (340)
ratio of timings 128/64	6.6	7.1	15.7
64x64x64 with conductors	8.7 (14)	-	17.4 (230)

#### 4 OPTIMIZATION TECHNIQUES

Since WARP3d is a three-dimensional code, the simulation time can grow very large. Because of this, much effort has been put into optimizing the code. The optimization can be done by using advanced computational methods to reduce the total amount of computation needed, as with the implementation of the multigrid solver described above, or as with the use of higher-order integration methods[5]. The methods discussed here are methods for increasing the speed at which a set amount of computation is done.

There are a number of simple optimizations that can be done which reduce the time of computation on serial machines. For example: combine constants in loops and remove them completely from loops if possible; remove divides from loops if possible, and if not possible, put the divide as early as possible if the result will be used inside of the loop; treat multidimensional arrays as one-dimensional arrays; use "if" statements to avoid unnecessary work, even inside of loops; and optimize the cache use. The last technique is described more fully.

Most modern day computers have non-uniform memory access: many more CPU cycles are required to bring data from the main memory to the cache, than from cache to the register where the CPU can use the data. The difference is as large as a factor of thirty! The goal, then, is to reuse data as much as possible once it is in the cache. There are three basic techniques. The first is to take advantage of the fact that the data is brought into the cache in chunks. Grouping data together in memory that is used together in

loops or making inner loops over the first indices of arrays (in Fortran) accomplishes this. The second technique is to maximize the number of mathematical operations for each fetch/store from memory. Combining loops is a good way to accomplish this. An implication of this is that the array syntax of Fortran 90 will be bad for cache reuse since it tends to break calculations into shorter loops. The third method relies on knowing the cache line size, the size of the chunk of data that is brought into the cache at once. The arrays should be arranged in memory so that multiple arrays used inside of the same loop do not push each other out of the cache.

The 3-D FFT Poisson solver is used as an example of where cache reuse can improve the execution time. The FFT's are organized as "gang" FFT's - in the transforms over each dimension, the inner loops in the transform are over one of the other dimensions. For the transverse transforms, the outer loop is over the third dimension, while the first two dimensions switch roles between the transformed and the ganged. In both cases, the data is well localized. For the transform over the third dimension, though, the data is not localized, there is a large stride in memory, and there is little cache reuse. Timings of the code bear this out; the transform over the third dimension runs several times slower than the transforms over the first two (which are roughly equal).

The way around the decreased performance is to temporarily rearrange the data to localize it in memory for the transform. The 3-D data is divided into slices along the second dimension, each slice consisting of a plane of data with the first dimension as one axis and the third dimension as the other. Each slice is copied into a temporary array and then transformed along the third dimension. This gives the same locality as in the original transverse transform. Afterward, the data is copied back. The resulting transform time is reduced to be same as the time of the transverse transform. One of the copies is "free" since the array was looped over anyway to multiply it by a constant and so the penalty is the one additional copy. The gain however is a factor of several reduction in the transform time for that dimension. The total gain over the full Poisson solve is typically a 30% reduction in computation time.

#### 5 PYTHON INTERFACE TO WARP

Our experience has shown that to realize the full power and capability of a large code, a high-level, flexible, sophisticated interface is required. A primary requirement of the interface is full language support for user-programmable code control that can be used interactively. An interpreted scripting language provides such an interface. Another required element is visualization - for pre- and post-processing as well as for interactive use. In this section we describe the work we have done to maintain such a scripting language interface in both the serial and parallel computing environments.

### 5.1 Why use an interpreter interface?

An interpreter interface provides a much more flexible means of inputting data and controlling a code than the more traditional command-line arguments and namelist style interface. It does this by allowing the use of high-level language constructs and by giving access to the runtime database and functionality of the code. This makes it easier to design a system where the code developers create a set of packages that users can combine and control to suit their needs, rather than create a code where the developers build in a limited number of options that the users must choose from.

An interpreter allows for more rapid code development. Use of an interpreter removes the need for compile and load steps, directly reducing development time. It also acts as a built in debugger, giving full access to the data, but is more powerful, allowing independent testing of code segments. Once algorithms have been developed and tested, they can be converted into compiled code if there is a problem with speed.

The code size is reduced when an interpreter is used. Most interpreters provide extensive services such as graphics, memory management, and data dump and restart facilities, for example. These are accessible from the interpreted language and do not need to be referenced from the compiled portion of the code. Only the core routines and algorithms need to be built into the code, the rest can be written at the interpreter level. Special features which have a one time use or are required by certain users, for example, can be kept out of the main code and written in the interpreted language.

An interpreter can act as “glue,” linking different codes and packages together. The packages can have separate variable name-spaces and can be developed separately by different developers in different languages. The packages are brought together at the interpreter level; all have the same interface.

WARP was originally built with the Basis code development/interpreter system that was developed at LLNL[6]. Basis has a number of advantages over other interpreters. It has a built in mechanism for generating the interface between the compile code and the interpreter language. It also has a number of other important features built in, such as graphics, memory management, data dumps and restarts. Also, the language is based on Fortran, making it easy to use and convert into compiled Fortran. Another important advantage is the modularity. Basis allows the code to be developed as a set of independent packages with separate name-spaces. The packages are glued together at the interpreter.

There are a number of disadvantages of Basis as well. It is a rather large system which is not very portable. Basis runs only on Unix and Linux machines. It is missing some programming features, such as structures and objects. Also, it has a small number of developers, limiting its breadth of features and development.

### 5.2 Parallel WARP with Basis

In our first attempt at a user interface to parallel version of WARP, we wanted to retain Basis. Unfortunately, Basis has not been ported to the MPP architectures we use. We were able to get around this by running Basis on a local serial workstation and having it spawn and control processes on a remote MPP. Communication was done with PVM[7] since it allowed spawning of processes. The processes on the MPP were event driven - they would wait for commands that the user sends via the Basis interpreter and PVM.

While this system was effective and allowed us to get WARP up and running on an MPP, it had significant drawbacks. Since there was no interpreter running on the MPP side, all possible desired commands had to be preprogrammed, so access to runtime database and controllers was limited to those for which special routines had been written. Because of the requirement for spawning, the code could only be run interactively - batch jobs could not be run. Also, the system required a “close connection” between the workstation and the MPP. Due to computer center policy restrictions, this is not always possible. Because of this last drawback, the parallel version of WARP was in fact inoperable on the primary MPP machine we had access to.

### 5.3 Python

We needed to make use of another interpreter and decided on Python[8]. Python is a recently developed interpreter to which compiled code can be linked and which has a number of advantages over Basis. The language is fully object-oriented (supporting inheritance and polymorphism). It is also small and portable and runs on almost all types of machines, UNIX, PC's running Linux or windows, and Macintoshes. Python also has a large number of developers and has a broad base of available packages, such as linear algebra libraries, interfaces to graphical user interface development libraries, world-wide-web software, and notably, a parallel adaptation.

There are problems with Python, though. A minor problem is the language itself, which has attributes which may be unfamiliar to many scientific users, as discussed below. Python also lacks standards for important features such as graphics and data dumps. There is also no built in method of automatically generating the interface between the compiled code and the interpreter language. While there are packages that do this, they are not flexible enough to meet all of our needs. Note that while the interface can be created by hand, doing so for a large code is not practical considering the size of the interface and the constant need for updating it.

We had to develop our own software to automatically generate the interface between Fortran and Python. The software takes advantage of the work which was done for the Basis interface. The same variable description files were used - a parser was written in Python to extract the information needed from the files to create the interface. The

Fortran preprocessor used with the Basis system is kept so that the original compiled code can be used essentially unchanged. The coarse-grained object-oriented nature of the Basis code is maintained. Each of the original Basis packages is turned into a Python object and has its own namespace.

The most important detail of the interface is the way compiled variables are accessed from the interpreter. In the Python language, all variables are references to objects. The major implication of this is that an assignment, such as "a = b", is actually a re-reference. Before the assignment, the variables "a" and "b" refer to different objects. After the assignment, "a" and "b" refer to the same object, that to which "b" originally referred, and the object to which "a" referred is lost. This means that there cannot be a direct connection between a Python variable and a Fortran variable, since that connection would be lost on an assignment.

The way around the re-referencing is to make use of "attributes" of objects, which are like class members and functions. The getting and setting of attributes can be redefined, allowing the possibility of connecting an attribute to a compiled variable or function via the get and set routines. Get and set routines are defined which perform a search through the list of Fortran variables and subroutines of a package object to find the one associated with the Python attribute.

A generic Python type is created and each package is defined as being an object of that generic type. The definition of the type includes generic functions, such as the get and set, which take the package object as an argument. In creating an object for a package, all of the required information is stored, such as the list of Fortran variables and subroutines.

#### 5.4 Parallel WARP with Python

Python is easily ported to an MPP environment, with one caveat: the input and output must be controlled. Software written by others[9] was obtained and used to do this. This software is designed so that only one processor can read the user input. That process then passes the input to the other processes via message passing. The user can control which processes can print output - the default is only the process that reads the input.

Combining the existing parallelized Fortran code with the Python interface and the input/output package gives a fully interpreter-driven code on the MPP which has the same user interface as the serial code. This combined system is nearly fully operational; nearly all of the functionality which was available with the original serial Basis version is now available with the Python version, both serial and parallel.

The development of the Python interface was a significant step in the evolution of WARP. Since Basis is used in several important LLNL codes, it will likely be around and supported for a long time. However, some of its disadvantages are unlikely to be removed. These include a lack of structures and objects, limited available software packages,

and a lack of portability. For these reasons, we are currently retaining both versions (the source is the same, only the interface and scripts are different). We are converting the Basis scripts to Python as they are needed.

## 6 CONCLUSIONS

While WARP is in some ways a mature code, it is still rapidly evolving. We are adding new physics models, such as the WARPxy model described. We are developing more advanced computational techniques, such as the multigrid method for solving Poisson's equation. We are also adopting modern computer science techniques, including optimization through cache reuse, use of the modern scripting language Python for code steering and user programmability, and massively parallel computation via message passing.

## 7 REFERENCES

- [1] D. P. Grote, A. Friedman, I. Haber, S. Yu, "Three-Dimensional Simulations of High-Current Beams in Induction Accelerators with WARP3d", *Proceedings of the 1995 International Symposium on Heavy Ion fusion, Fusion Engineering and Design*, 32-33 (1996) 193-200.
- [2] J. J. Barnard, H. D. Shay, S. S. Yu, A. Friedman, and D. P. Grote, "Emittance growth in Heavy Ion Recirculators", 1992 Linear Accelerator Conference Proceedings, Ottawa, Canada, vol. 1, p 229, AECL Research, (1992).
- [3] William H. Press et. al., *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, 1986.
- [4] D. P. Grote, A. Friedman, I. Haber, "Methods used in WARP3d, a Three-Dimensional PIC/Accelerator Code", *Proceedings of the 1996 Computational Accelerator Physics Conference*, AIP Conference Proceedings 391, p. 51.
- [5] A. Friedman, D. P. Grote, I. Haber, "Towards Higher Order Particle Simulation of Space-Charge-Dominated Beams", *Proceedings of the 16th International Conference on the Numerical Simulation of Plasmas*, February 10-12, 1998, Santa Barbara, CA.
- [6] P. F. Dubois, The Basis System, LLNL Document M-225 (1988)
- [7] Al Giest, et. al., "PVM3 User's Guide and Reference Manual", technical report ORNL/TM-12187.
- [8] The best reference is the Web page <http://www.python.org/>
- [9] The primary authors are D. Beazley and T. B. Yang.

# Unique Features of the University of Maryland Electron Ring and the Necessity of PIC Code Simulation

R. A. Kishek, S. Bernal, M. Venturini, and M. Reiser,  
Institute for Plasma Research, University of Maryland, College Park, MD 20742  
I. Haber, Naval Research Laboratory, Washington, DC 20375  
D. P. Grote, Lawrence Livermore National Lab, Livermore, CA

*Abstract.*

The Maryland Electron Ring [1] is designed to explore the transport of beams with much higher space charge than other circular machines. In addition, the ring functions as a testbed for design and simulation codes. Applications such as Heavy Ion Fusion and High Intensity Colliders require the preservation of beam quality during transport over large distances. This paper describes the application of self-consistent particle-in-cell code simulations using the WARP suite [2] to the E-ring lattice. The model used includes the nonlinear details of the external magnetic fields, a cylindrical external conductor, and the dispersive effects of the circular lattice on a beam with a non-zero energy spread.

## 1. INTRODUCTION

In the accelerator field, many applications are emerging that require high intensities and good beam quality. Heavy Ion Fusion, for example, requires the transport and acceleration of a high current beam and focusing it onto a tiny spot. Spallation neutron sources and high intensity colliders also require good beam quality, although the intensities needed may be somewhat lower. Furthermore, most of these applications involve some degree of bending of the beam, and some can benefit from the concept of a recirculator to save space and costs. The University of Maryland Electron Ring [1], which is currently in its early construction stages, is a scaled experiment designed to investigate the physics of space-charge dominated beams in a circular geometry. A key goal of the project is its low cost, driving us to use innovative features such as printed-circuit magnets [3] and modularity in design.

Since the E-ring is to be operated in regions of high space-charge previously inaccessible to circular machines, a high priority in this project rests on self-consistent computer simulations to verify the design and further probe the physics. In the on-going process leading up to the commissioning of the ring, the numerical simulations are benchmarked against any experimental measurements available. Because of its low cost and versatility, the E-ring will provide a valuable testbed for computer codes to be used in designing larger machines.

In ref. [4] we had presented results of preliminary simulations in a straight alternating-gradient channel including the magnet nonlinearities. In ref. [5] we extended the simulations to follow the beam along the bent lattice, examining the consequences of lens nonlinearities, mismatches, and dispersion on the beam quality. In this paper, we will expand on some of the features of the E-ring that necessitate the use of particle-in-cell (PIC) methods, then proceed to describe some interesting results observed in the simulations.

In the simulations described herein, we rely primarily on the WARP particle-in-cell (PIC) code [2], which has been developed at Lawrence Livermore National Laboratory for Heavy Ion Fusion applications. An important feature of the WARP code is its ability to efficiently track a space-charge-dominated beam along bends. The particle orbits are integrated self-consistently using the fully nonlinear electrostatic self-field, as well as the fully nonlinear external fields from the bending dipoles and focusing quadrupoles. The WARP code is an attractive choice for the Maryland ring simulations especially because it has been used in simulating the similar Heavy Ion Recirculator at Lawrence Livermore National Lab [2], and is therefore well-suited to such geometries and already contains a variety of lattice element representations.

## 2. RING LATTICE AND SIMULATION SETUP

The design and progress of the electron ring project have been described more fully elsewhere [1]. For the purposes of this paper, it will be sufficient to briefly describe the lattice and nominal beam parameters. The nominal beam current is 100 mA at 10 keV, resulting in a generalized perveance of 0.0015. A nominal (unnormalized  $4\sigma_{rms}$ ) emittance of 50 mm-mrad and nominal average beam radius of 10.2 mm ( $\sigma_0 = 72^\circ$ ) results in a tune depression ( $v/v_0$ ) of 0.16, placing the beam in the strongly space-charge-dominated regime. A future phase is planned where the beam is to be accelerated to 50 keV. Moreover, the ring is designed to run at lower beam currents, allowing us to explore a wide range of tunes.

Figure 1 displays a schematic of the ring lattice, which consists of 36 FODO cells around the 11.52 m circumference ring. Each cell is therefore 32.0 cm long

and contains two evenly-spaced printed-circuit quadrupoles [3] and, in between those, a printed-circuit dipole which bends the beam by  $10^\circ$ . The quadrupole gradient is about 0.078 Tesla/m, while the bending dipole peak field is about 0.00154 Tesla. Three induction gaps used for longitudinal confinement are evenly distributed around ring, while the remainder of the spaces are occupied by diagnostics and pumping ports.

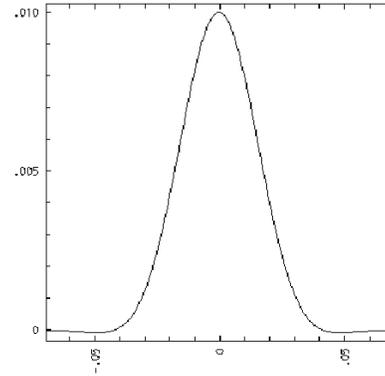


**Figure 1.** Schematic of ring design [1].

Because of space limitations in the mechanical design, the quadrupoles and the dipoles are short relative to the pipe radius (effective length of both  $\sim 3.7$  cm; pipe radius  $\sim 2.5$  cm). Indeed, as Figure 2 shows, the quadrupoles and dipoles consist almost entirely of fringe fields. Although the external fields were designed to be linear in an integrated sense, the question arises whether they may be nonlinear when integrated over the actual particle orbits, and whether such nonlinearities lead to emittance growth. PIC methods that self-consistently include the space charge are powerful tools in answering such questions.

The short effective length of the dipoles implies that the lattice can be thought of as a combination of bends and straight sections. In other words the local radius of curvature of the reference orbit varies as a function of the propagation distance,  $s$ , inside the bend. Furthermore, for induction accelerators, the dominant cost-driving factor is the size of the induction cores, which is partly determined by the bore size and hence the beam size. This expense can be reduced if the beam fills a large cross-section of the beam pipe, but at the expense of image forces. In the Maryland E-ring the ratio of beam radius to pipe radius is about  $2/5$ , and so, as explained

below, the image forces are non-negligible and must be implemented carefully into the model.



**Figure 2.** Profile of printed-circuit quadrupole on-axis gradient as a function of  $z$ .

Since a beam, in general, has a non-zero energy spread, propagation in bends leads to dispersion, i.e., particles at different energies oscillate around different reference trajectories. If not explicitly matched for dispersion, the beam experiences a dispersion mismatch as it enters the ring from the straight injector, or as it leaves it to the extraction section. For a space charge dominated beam, this dispersion mismatch leads to emittance growth, as confirmed earlier by theory and simulation [6-7, 5]. Further studies described below demonstrate that dispersion is more pronounced for space-charge-dominated beams and show the dependence.

For these simulations of the transverse dynamics, we use WARP-XY, the single-slice version of WARP, which solves for the self-fields on a 2-D mesh, but retains the velocity information in all 3 dimensions [2]. The lattice is modeled from first principles by computing the lens magnetic field values on a 3D grid with the aid of a magnetics program, starting with the actual conductor geometry. At every step, WARP interpolates this field data to the particle locations. Thus the dipoles and quadrupoles used in the simulation include the nonlinearities caused by the fringe fields. We load a beam with an initial semi-Gaussian distribution into the field-free region between two quadrupoles, and then follow its evolution along  $s$ , the distance traversed by the beam, for a number of turns (typically 10). The beam is initially rms matched into the lattice using the standard rms envelope equations [8].

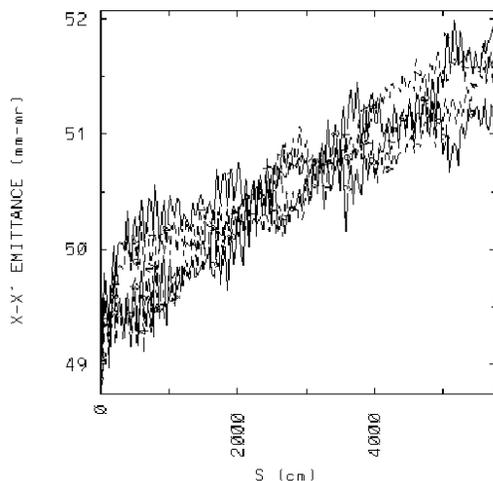
The numerics have been thoroughly tested to ensure convergence. A mesh of  $256 \times 128$  cells across the 5 cm-diameter pipe (symmetry in the vertical direction halves the number of cells needed) provides sufficient resolution of Debye-length-scale variations in the potential. The number of simulation particles is chosen to minimize effects of numerical collisions, producing a

potential that is relatively smooth from cell to cell. Up to 400k particles have been used in some simulations, although it is found that for most purposes, 100k particles are sufficient. Gaussian filtering (see [9]) allows us to get convergent results with even fewer particles ( $\sim 10k$ ). The time step used corresponds to a beam advance by 0.4 cm, which is enough to resolve the fringe fields of the magnets. In addition, we continually benchmark WARP against emerging experimental measurements from prototypes of the ring and the injector. Agreement, which has been excellent so far, is discussed elsewhere [10].

### 3. SIMULATION RESULTS AND DISCUSSION

#### 3.1 Magnet nonlinearities

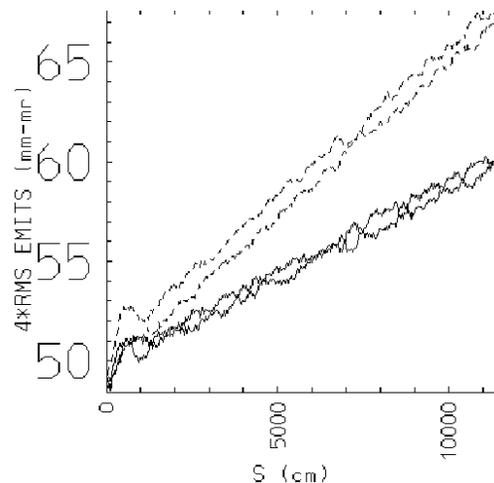
In ref. [4] we had presented the scaling of emittance growth in a straight channel with the number of particles, indicating that whatever growth present is numerical in nature and can be eliminated by increasing the number of particles. In fact, the nonlinearities from the fringe fields of the short quadrupoles are found to contribute very little to the emittance growth. This can also be demonstrated by replacing the full description of the magnets in the model with ideal, hard-edge elements. As Fig. 3 indicates, there is little difference between the two cases, so the assumption that the quadrupole fields are linear when integrated over particle orbits is correct. This is true because, by design, the quadrupole fields are linear when integrated along  $z$ , and the quadrupoles are short enough compared to the (depressed) betatron period that the radii of particle orbits do not change significantly.



**Figure 3.** Evolution of a matched beam with the nominal design parameters in a straight A-G channel over 360 lattice periods, for nonlinear (solid) vs. ideal quadrupoles (dotted). Value of  $4 \times \text{rms } x$  and  $y$  emittances strobed every lattice period in between quadrupoles.

The dipoles, however, are a different matter. After the addition of bends and dipole fields into the WARP model, we observe a somewhat larger emittance growth [dotted lines in Fig. 4]. Although the emittance growth rate is not excessive (30 % over 10 turns after subtracting numerical growth), it is important to understand its causes. [The small short-term emittance growth at the beginning is due to a dispersion mismatch; it can be removed by launching a beam with a smaller energy spread].

Numerical tests suggest that this growth in emittance is real. Although the study conducted so far is by no means exhaustive, we have reasons to believe that this emittance growth is related to the interaction of the beam space charge with the dipole nonlinearities. For instance, in some simulations, we applied appropriate scaling to reduce the effects of the dipole nonlinearities experienced by the beam. This was accomplished by reducing both the beam current and emittance in order to reduce the beam size while maintaining both the matched condition and the tune depression. The boundaries were also shifted to maintain the contribution of the image forces, while the numerics were adjusted to maintain the same resolution and collisionality.



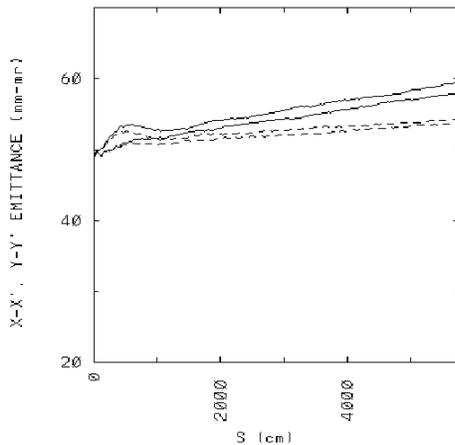
**Figure 4.** Evolution of a matched beam in the ring over 10 turns (360 lattice periods). The dotted line represents a beam with the nominal design parameters while the solid line represents a beam scaled by a factor of 1/2 to reduce the effects of dipole nonlinearities. Value of  $4 \times \text{rms } x$  and  $y$  emittances strobed every lattice period in between quadrupoles. Here, ideal quadrupoles are used to isolate the effects of the dipoles.

The result of this test is that the smaller beam, which samples less of the dipoles' deviations from linearity, exhibits a significantly smaller emittance growth [solid lines in Fig. 4]. The question arises as to why the dipoles, which were also designed to be linear in

the integrated sense, lead to more emittance growth than the quadrupoles. There are two differences worth noting, however, between the dipoles and the quadrupoles. First, the dipoles are typically operated with somewhat stronger fields than the quads (the dipole field is a factor of 2 more than the quad field at the edge of the beam). The second is that the beam, hence the individual particles, follows a curved trajectory inside the dipole, whereas the dipole fields were designed to be linear when integrated along  $z$ , the dipole axis. This difference is small for a  $10^\circ$  bend, yet nevertheless may add up over a large number of lattice periods. Also note that the space charge plays an important role. An emittance-dominated beam with the same size experiences much less emittance growth.

### 3.2 Boundaries

As mentioned earlier, the beam fills a relatively large cross-section of the beam pipe. Furthermore, the local radius of curvature of the beam trajectory varies inside each bend, and so even a well-aligned beam departs momentarily from the pipe axis. This makes image effects quite significant. In WARP, a capacity matrix used in conjunction with the FFT field solver is used to represent the cylindrical boundary [2]. The capacity matrix solver was not originally implemented for use inside bends. Hence, when used inside a bend, that model corresponded to leaving a straight beam pipe around the beam. Later modifications to the field solver\* refined the capacity matrix implementation to model the beam pipe in the bends as a curved conductor.



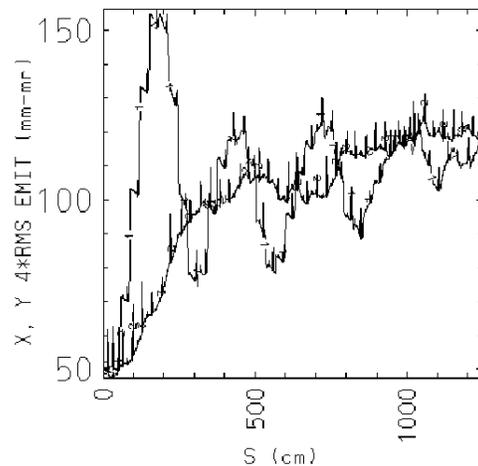
**Figure 5.** The effect of different implementations of the boundary conditions inside the bend on beam emittance; curved conductors (solid) vs. straight conductors (dotted).

\* Done by D. P. Grote at LLNL.

Although the correction is small, the differences add up after propagation through a large number of bends, thus affecting the long-term emittance growth computed by the code. This result is shown in Fig. 5, where we compare the emittance growths from the simulations using the two different implementations of the capacity matrix. Clearly, the different boundaries significantly affect the long-term emittance, resulting in a factor of 2 difference in its growth rate. Therefore, at least for situations like this where the beam covers a large fraction of the pipe cross-section, the boundaries must be accurately represented in order to get accurate answers. This further implies that calculations that totally neglect the boundary may not be as accurate.

### 3.3 Dispersion

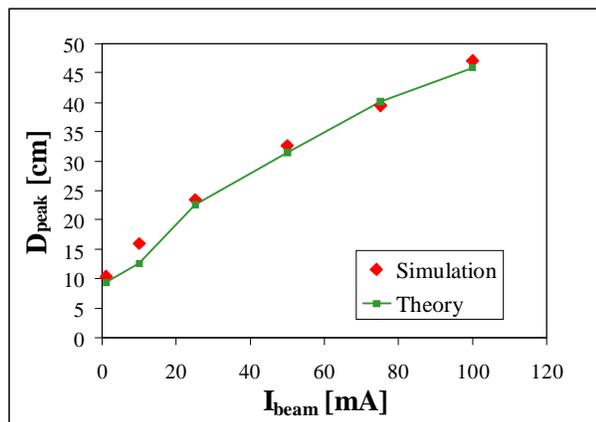
Venturini and Reiser in ref. [6] have derived new envelope equations that are valid in the presence of space charge and dispersions. These equations can be used, for example, to match the beam from a straight injector into a ring. In refs. [5] and [7], WARP simulations have been applied to injection into the Maryland Electron Ring. No attempt has been made to match the dispersion function of the beam, relying instead on the standard RMS envelope equations [8] for matching into the ring. Beams with a spread in energies experience a mismatch in dispersion and an increase in the  $x$ -emittance, followed by damped oscillations [Fig. 6]. Incidentally, this result with WARP-3D is almost identical to that obtained from WARP-XY (which uses a different set of approximations), confirming our confidence in the simulation. The simulations were found to be in excellent agreement with the theory presented in ref. [6], except for the eventual exchange of energy between the  $x$  and  $y$  directions resulting in the damping of the oscillations.



**Figure 6.** X and y emittances over 1 turn from a 3-D simulation of a beam injected into the ring from a straight lattice with an initial velocity spread ( $\Delta v_z/v_z = 0.015$ ).

The emittance increase and oscillations are due to a dispersion mismatch.

That study is extended here to test the theory (ref. [6]) for a wider range of parameters. Figure 7 compares the peak of the dispersion function oscillations calculated from the theory to that observed in simulations as function of the beam current. The good agreement confirms the theory for lower tune depressions. Indeed, for lower tune depressions, the damping seen in the simulations occurs over a longer timescale, which increases the region of overlap between the theory and the simulation. Note that the peak of the dispersion function increases almost linearly with the beam current. Since the dispersion function is a measure of the distortion of reference trajectories of particles according to their energy, this result demonstrates that beams with a higher space charge are more susceptible to dispersive effects. In other words, dispersion becomes more important for space-charge-dominated beams.



**Figure 7.** Dependence of peak of dispersion function on beam current for a velocity spread,  $\Delta v_z/v_z = 0.015$ : theory (solid line) vs. simulation (points).

It is worth noting that Barnard, et. al., at LLNL have proposed a similar theory on dispersion based on the moment description [11]. Although that theory cannot be used to calculate the dynamic evolution of the beam envelope and emittance, their thermodynamic approach provides an accurate estimate of the final beam emittance (i.e., after the oscillations subside) that agrees well with our simulations on the Maryland ring. Barnard's theory also predicts a larger emittance growth from a dispersion mismatch for higher space charge.

#### 4. CONCLUSION

In summary, the Maryland Electron ring contains some unique features that need to be included in a simulation model for a faithful representation of the machine. In this

paper we have addressed the magnet nonlinearities, the outer conductor, and the energy spread of the beam. It is found that the effect of the nonlinearities caused by the fringe fields of the short quadrupoles vanish upon integrating along the particle orbits. Those of the short bending dipoles, however, do not completely vanish and lead to an emittance growth after several turns around the ring. We are conducting additional studies to understand the exact mechanism by which the emittance grows in the dipoles. Since the local radius of curvature of the beam trajectory inside the bends is not constant, and since the beam fills a large cross-section of the beam pipe, the image forces cannot be neglected from the model. Finally, a non-zero energy spread leads to dispersion and perhaps to a change in the matching conditions of the beam. Dispersion is seen to affect the beam substantially more at higher currents, in agreement with the applicable theories.

As experimental data becomes available in the forthcoming commissioning process of the E-ring, it will be interesting to compare with the WARP predictions.

#### ACKNOWLEDGEMENTS

We wish to thank our colleagues John J. Barnard, Alex Friedman, Terry F. Godlove, Steve Lund, and J.G. Wang for many fruitful discussions. The code runs on computers provided by the National Energy Research Scientific Computing Center (NERSC). This research is sponsored by the U. S. Department of Energy (DOE) under contract DE-FG02-94ER40855.

#### REFERENCES

- [1] M. Reiser, S. Bernal, A. Dragt, *et. al.*, *Fus. Eng. and Des.* **32-33**, 293 (1996); J. G. Wang, S. Bernal, P. Chin, *et. al.*, *Nucl. Instr. and Meth. A*, **415**, 422-427 (1998).
- [2] D. P. Grote, *et. al.*, *Fus. Eng. & Des.* **32-33**, 193-200 (1996).
- [3] T. Godlove, S. Bernal, and M. Reiser, *Proc. of 1995 Particle Accelerator Conference*, 2117 (1995).
- [4] R. A. Kishek, S. Bernal, M. Reiser, *et. al.*, *Nucl. Instr. and Meth. A*, **415**, 417-421 (1998).
- [5] R. A. Kishek, I. Haber, M. Venturini, and M. Reiser, presented at the Workshop on Space-Charge Physics in High Intensity Hadron Rings, Shelter Island, NY, May 4-7, 1998, in print (1999).
- [6] M. Venturini and M. Reiser, *Phys. Rev. Lett.*, **81**, 96 (1998);
- [7] M. Venturini, R. A. Kishek, and M. Reiser, *Proc. Workshop on Space-Charge Physics in High Intensity Hadron Rings*, Shelter Island, NY, May 4-7, 1998, in print (1999).
- [8] M. Reiser, *Theory and Design of Charged Particle Beams*, (New York: Wiley & Sons, 1994).
- [9] C. K. Birdsall and A. B. Langdon, *Plasma Physics through Computer Simulation*, (New York: McGraw Hill, 1985).
- [10] S. Bernal, P. Chin, R. Kishek, *et. al.*, *Phys. Rev. ST Accel. Beams* **4**, 044202 (1998); S. Bernal, R. A. Kishek, M. Reiser, and I. Haber, "Observation and Simulation of Radial Density Oscillations in Space-Charge Dominated Electron Beams," to be published (1999).
- [11] J. J. Barnard, G. D. Craig, A. Friedman, *et. al.*, *Proc. Workshop on Space-Charge Physics in High Intensity Hadron Rings*, Shelter Island, NY, May 4-7, 1998, in print (1999).

# ON SOLVING MAXWELLIAN EIGENVALUE PROBLEMS FOR ACCELERATING CAVITIES

Peter Arbenz, Institute of Scientific Computing, Swiss Federal Institute of Technology (ETH)  
8092 Zurich, Switzerland

Stefan Adam, Paul Scherrer Institute, 5232 Villigen, Switzerland

## Abstract

We investigate algorithms for solving large sparse symmetric matrix eigenvalue problems resulting from finite element discretizations of steady state electromagnetic fields in accelerating cavities. The methods have been applied to the new design of the accelerating cavity for the PSI 590 MeV ring cyclotron.

The solutions of this kind of eigenvalue problems can be polluted by so-called spurious modes if the divergence-free condition is not treated properly. In this paper we deal with a method that suppresses spurious modes by adding a penalty term to the basic quadratic form. This is the method we had the best experience with [1, 2].

The large sparse eigenvalue problems have been solved with the implicitly restarted Lanczos algorithm. Numerical results obtained on a HP Exemplar X-Class System are reported.

## 1 INTRODUCTION

For the production of high intensity proton, neutron and meson beams used in a wide range of research activities the cyclotron facility of the Paul Scherrer Institute accelerates proton beams to an energy of 590 MeV with an intensity of 1.5 mA in routine operation. The four accelerating cavities in the separate sector cyclotron provide a 50 MHz peak accelerating voltage of 750 kV each. Aiming at a reduction of the power losses in the cavity walls a project for a new design of the accelerating cavities has been started, cf. Fig. 1.

During design the RF engineer is interested in the field shapes and the frequencies of the fundamental and of some few higher modes of the standing electromagnetic waves in a cavity. This corresponds to the wish for access to a Maxwellian eigenvalue solver for some of the lowest eigenvalues and corresponding eigenvectors.

At the Institute for Scientific Computing at ETH Zurich the key questions related to the creation of such an eigensolver have been studied. In order to be able to handle complex geometries with locally fine details, the formulation of the problem was based on finite elements and emphasis has been given to the efficient solution of large problems. It proved to be important to understand the occurrence and to find means to avoid the so-called spurious modes (or ghost modes) that can occur in finite element formulations of Maxwell's equations. This study compared different finite element formulation schemes each used with different

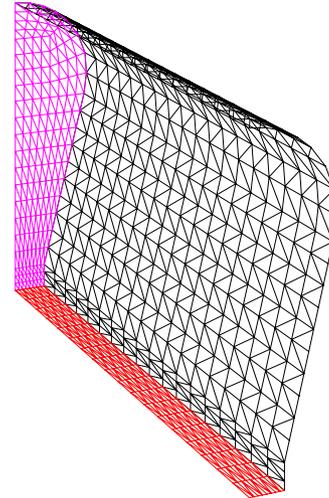


Figure 1: Surface triangulation of an eighth of the new PSI cavity design.

orders of approximation as well as several algorithms for the solution of the corresponding large sparse matrix eigenvalue problems [1, 2].

Different algorithms have been investigated to compute a few of the eigenvalues of the resulting large sparse matrix eigenvalue problems [1, 2]. Here we restrict ourselves to the implicitly restarted Lanczos algorithm that is, together with the Jacobi-Davidson algorithm, the best suited algorithm for this kind of problems. The linear systems of equations that occur in the shift-and-invert mode of the Lanczos algorithm have been solved by the conjugate gradient method in conjunction with a hierarchical basis preconditioner.

## 2 THE MAXWELLIAN EIGENVALUE PROBLEM

Without changing the basic structure of the problem one can assume that the metallic surfaces are perfectly conducting and that the inside of the cavity  $\Omega$  is all in vacuum. The electromagnetic field in the cavity is described by the Maxwell equations [13]. After separation of time and space variables and after elimination of the magnetic field inten-

sity the differential equations

$$\mathbf{curl} \mathbf{curl} \mathbf{e}(\mathbf{x}) = \lambda \mathbf{e}(\mathbf{x}), \quad \mathbf{x} \in \Omega, \quad \lambda := \omega^2/c^2, \quad (1a)$$

$$\mathbf{div} \mathbf{e}(\mathbf{x}) = 0, \quad \mathbf{x} \in \Omega, \quad (1b)$$

$$\mathbf{n} \times \mathbf{e} = 0, \quad \mathbf{x} \in \Gamma := \partial\Omega. \quad (1c)$$

are obtained for the electric field intensity  $\mathbf{e}$ . Equations (1) have solutions only for certain values  $\omega$  called *eigenfrequencies* of the electromagnetic oscillations of the resonator. These solutions are called *eigenmodes*.

The eigenvalue problem (1) can be solved analytically for a few domains, in particular for the case where  $\Omega$  is a rectangular box [1]. We conducted our numerical experiments by means of this simple model which allows to compare the numerical results to an absolute reference. Moreover, for separate sector cyclotrons like the 590 MeV ring cyclotron installed at the Paul Scherrer Institute (PSI) in Villigen, Switzerland, the gross shape of a cavity is actually well approximated by a rectangular box.

The variational form of the eigenvalue problem (1) is given by [1]

$$\begin{aligned} \text{Find } (\lambda, \mathbf{u}) \in \mathbb{R} \times W_0 \text{ such that } \mathbf{u} \neq \mathbf{0} \text{ and} \\ (\mathbf{curl} \mathbf{u}, \mathbf{curl} \mathbf{v}) = \lambda(\mathbf{u}, \mathbf{v}), \quad \forall \mathbf{v} \in W_0. \end{aligned} \quad (2)$$

Let  $L^2(\Omega)$  be the Hilbert space of square-integrable functions over the 3-dimensional domain  $\Omega$  with inner product  $(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \mathbf{u}(\mathbf{x}) \cdot \mathbf{v}(\mathbf{x}) \, d\mathbf{x}$  and norm  $\|\mathbf{u}\|_{0,\Omega} := (\mathbf{u}, \mathbf{u})^{1/2}$ . In (2),  $W_0 := \{\mathbf{v} \in W \mid \mathbf{div} \mathbf{v} = 0\}$  with

$$\begin{aligned} W := \{\mathbf{v} \in L^2(\Omega)^3 \mid \mathbf{curl} \mathbf{v} \in L^2(\Omega)^3, \\ \mathbf{div} \mathbf{v} \in L^2(\Omega), \mathbf{n} \times \mathbf{v} = 0 \text{ on } \partial\Omega\}. \end{aligned} \quad (3)$$

The difficulty with (2) stems from the condition  $\mathbf{div} \mathbf{v} = 0$  as it is hard to find divergence-free finite elements. Therefore, ways have been looked for to get around this *divergence-free condition*. In this process care has to be taken in order not to introduce so-called *spurious modes*, i.e. eigenmodes that have no physical meaning.

In [1, 2] we considered two approaches free of spurious modes, a penalty method and a mixed method. The former was implemented with Lagrange finite elements the latter with Nédélec elements [9, 11]. In this note we restrict ourselves to the penalty method. Results concerning the Nédélec elements will be reported upon later. From our earlier experiments we expect them to be less efficient. In those experiments the computing time needed to get a prescribed accuracy for eigenvalues and vectors was about an order of magnitude smaller with the penalty method than with the mixed method. In the penalty method approach (2) is replaced by [14, 16]

$$\begin{aligned} \text{For } s > 0, \text{ find } (\lambda, \mathbf{u}) \in \mathbb{R} \times W, \mathbf{u} \neq \mathbf{0}, \text{ such that} \\ (\mathbf{curl} \mathbf{u}, \mathbf{curl} \mathbf{v}) + s(\mathbf{div} \mathbf{u}, \mathbf{div} \mathbf{v}) = \lambda(\mathbf{u}, \mathbf{v}), \quad (4) \\ \text{holds for all } \mathbf{v} \in W, \end{aligned}$$

In (4),  $s$  is a positive, usually small parameter. The eigenmodes  $\mathbf{u}(\mathbf{x})$  corresponding to eigenvalues  $\lambda < \mu_1 s$  are

eigenmodes of (2).  $\mu_1$  is the smallest eigenvalue of the negative Laplace operator  $-\Delta$  on  $\Omega$ . Therefore,  $s$  has to be chosen such that  $\mu_1 s$  is above the largest eigenvalue we want to compute.

### 3 FINITE ELEMENT FORMULATION

We triangulate the open, bounded domain  $\Omega \subset \mathbb{R}^3$  with polyhedral boundary  $\Gamma$  by tetrahedrons  $\Omega_e$ . We approximate  $W$  by the well known Lagrange (or node-based) finite elements spaces [9],

$$W_h^{(k)} := \{\mathbf{v} \in W \mid \mathbf{v}|_{\Omega_e} \in (P_k(\Omega_e))^3\}, \quad (5)$$

where  $P_k(\Omega_e)$  is the space of polynomials in  $\Omega_e$  of degree  $\leq k$ . In our numerical experiments [2] we found that trial spaces made up of piecewise quadratic ( $k=2$ ) polynomials are superior to trial spaces made up of piecewise linear ( $k=1$ ) with respect to computing cost relative to accuracy. We will therefore restrict ourselves to  $W_h^{(2)}$  equipped with a hierarchical basis [6, 23],

$$W_h^{(2)} = W_h^{(1)} \oplus V_h^{(2)}. \quad (6)$$

The piecewise linear polynomials in  $W_h^{(1)}$  are determined by their values in the four vertices of the tetrahedron. The piecewise quadratic polynomials in  $V_h^{(2)}$  are determined by the function values in the midpoints of the six edges.

Using  $W_h^{(2)}$  in the Rayleigh-Ritz method for discretizing (4) gives the finite dimensional problem

$$\begin{aligned} \text{For } s > 0, \text{ find } (\lambda_h, \mathbf{e}_h) \in \mathbb{R} \times W_h^{(2)}, \mathbf{e}_h \neq \mathbf{0}, \text{ s.t.} \\ (\mathbf{curl} \mathbf{e}_h, \mathbf{curl} \Psi_h) + s(\mathbf{div} \mathbf{e}_h, \mathbf{div} \Psi_h) = \lambda(\mathbf{e}, \Psi) \quad (7) \\ \text{holds for all } \Psi_h \in W_h^{(2)}. \end{aligned}$$

Let  $\{\Phi_i\}_{i=1}^n$  be a basis of  $W_h^{(2)}$  and  $\mathbf{e}_h = \sum_{i=1}^n \Phi_i \xi_i$ . Then (7) becomes the matrix eigenvalue problem

$$A\mathbf{x} = \lambda M\mathbf{x}, \quad \mathbf{x} = (\xi_1, \dots, \xi_n), \quad (8)$$

where the elements of  $A$  and  $M$  are given by

$$\begin{aligned} a_{i,j} &= (\mathbf{curl} \Phi_i, \mathbf{curl} \Phi_j) + s(\mathbf{div} \Phi_i, \mathbf{div} \Phi_j), \\ m_{i,j} &= (\Phi_i, \Phi_j). \end{aligned}$$

For positive  $s$ , both  $A$  and  $M$  are symmetric positive definite  $n$ -by- $n$  matrices. If the basis functions are chosen properly, then  $A$  and  $M$  are sparse.

Let us consider a rectangular box that is divided into  $m_1 \times m_2 \times m_3$  boxes each of which is in turn subdivided into six tetrahedrons. Then  $\dim W_h^{(2)} \approx 24m_1m_2m_3$ . The matrices  $M$  and  $A$  have up to 65 and 134 nonzero elements per row, respectively. Notice that  $\dim W_h^{(1)} \approx 3m_1m_2m_3$ . We will use  $W_h^{(1)}$  in our ‘coarse grid’ correction in the hierarchical preconditioner of section 5.

#### 4 THE IMPLICITLY RESTARTED LANCZOS ALGORITHM (IRL)

For computing a few, say  $p$ , eigenvalues of a sparse matrix eigenvalue problem

$$A\mathbf{x} = \lambda M\mathbf{x}, \quad A = A^T \geq 0, \quad M = M^T > 0, \quad (9)$$

closest to a number  $\tau$  it is advisable to make a shift-and-invert spectral transformation with a *shift*  $\sigma$  close to  $\tau$  and solve [12, 17]

$$C\mathbf{x} := (A - \sigma M)^{-1}M\mathbf{x} = \mu\mathbf{x}, \quad \mu = \frac{1}{\lambda - \sigma}. \quad (10)$$

instead of (9).  $C = (A - \sigma M)^{-1}M$  is  $M$ -symmetric, i.e., it is symmetric with respect to the inner product  $\mathbf{x}^T M \mathbf{y}$ . The spectral transformation leaves the eigenvectors unchanged. The eigenvalues of (9) close to the shift become the largest absolute of (10). In addition they are relatively well-separated which improves the speed of convergence of Krylov-type subspace methods [20]. The cost of the improved convergence rate is the necessity to solve a linear system of equations involving  $A - \sigma M$ .

In [1, 2] we compared four algorithms for computing a few of the smallest eigenvalues of (10): (i) subspace iteration [20], (ii) the block Lanczos algorithm [12], (iii) the implicitly restarted Lanczos algorithm [22, 8], and (iv) the Jacobi-Davidson algorithm [21, 10]. Subspace iteration was not competitive because of its low convergence rate. The block Lanczos algorithm was performing best for problems of limited size. However, the memory space available was not sufficient for solving our largest problems. The eigensolvers that were able to handle all problem sizes were the implicitly restarted Lanczos algorithm and the Jacobi-Davidson algorithm. The former was in our experiments slightly faster than Jacobi-Davidson by about 10-20%. We therefore only consider IRL in this study. We used the Fortran subroutines from the publicly available ARPACK [15].

To overcome the limitation of the Lanczos algorithm with respect to memory consumption, Sorensen proposed an elegant way to restart the iteration process [22, 8]. (These ideas apply to the Arnoldi algorithm for non-symmetric eigenvalue problems as well.)

The Lanczos iteration process (here with the shift-and-invert approach)

$$\begin{aligned} \mathbf{q}_{j+1}\beta_{j+1} &= \mathbf{r}_{j+1} = C\mathbf{q}_j - \alpha_j\mathbf{q}_j - \beta_j\mathbf{q}_{j-1}, \\ \mathbf{q}_i^T M \mathbf{q}_j &= \delta_{ij}, \quad \alpha_j = \mathbf{q}_j^T M (A - \sigma M)^{-1} M \mathbf{q}_j, \end{aligned} \quad (11)$$

is executed until  $j = p+k$ , where  $k$  is some positive integer. Often  $k = p$  is chosen. Then, the Lanczos relation

$$CV_{p+k} = V_{p+k}T_{p+k} + \mathbf{r}_{p+k+1}\mathbf{e}_{p+k}^T \quad (12)$$

holds where  $V_{p+k} = [\mathbf{q}_1, \dots, \mathbf{q}_{p+k}]$  and  $T_{p+k}$  is a symmetric tridiagonal matrix. Now,  $k$  sweeps of the QR algorithm [20] with shifts  $\mu_1, \dots, \mu_k$  are applied to  $T_{p+k}$  to

obtain  $\hat{T} = \hat{Q}^T T_{p+k} \hat{Q}$  where  $\hat{Q}$  contains the cumulated rotations of the QR sweeps. Multiplication of (12) by  $\hat{Q}$  from the right and considering only the first  $p$  columns yields a new Lanczos relation

$$C\hat{V}_p = \hat{V}_p \hat{T}_p + \hat{\mathbf{r}}_{p+1} \mathbf{e}_p^T, \quad (13)$$

with which the computations are continued. In (13),  $\hat{V}_p$  are the first  $p$  columns of  $V_{p+k}\hat{Q}$ ,  $\hat{T}_p$  is the  $p \times p$  principal submatrix of  $\hat{T} = \hat{Q}^T T_{p+k} \hat{Q}$  and  $\hat{\mathbf{r}}_{p+1}$  is the  $p$ -th column of  $\mathbf{r}_{p+k+1}\mathbf{e}_{p+k}^T \hat{Q}$ . In ARPACK, the shifts  $\mu_1, \dots, \mu_k$  are chosen as the  $k$  eigenvalues of  $\hat{T}_{p+k}$  furthest away from the desired target value  $\tau$ . Then, the eigenvalues of  $\hat{T}_p$  are the  $p$  eigenvalues of  $T_{p+k}$  closest to the target value.

Besides the storage for the matrices  $A$  and  $M$ , the memory requirements of the IRL algorithm are essentially the space needed to store  $\mathbf{q}_j$  and  $M\mathbf{q}_j$ ,  $j = 1, \dots, p+k$ , i.e.,  $2n(p+k)$  floating point numbers.

#### 5 TWO-LEVEL HIERARCHICAL BASIS PRECONDITIONERS

With ARPACK it is possible to use any algorithm to solve the indefinite system of equations  $(A - \sigma M)\mathbf{x} = \mathbf{y} = M\mathbf{q}_j$  in (11). We chose the iterative solver SYMMLQ of Paige and Saunders [19], a variant of the conjugate gradients method designed to handle symmetric indefinite systems of equations that we have to expect if the shift  $\sigma$  is inside the spectrum of  $A$  relative to  $M$ . Preconditioners for SYMMLQ have to be positive definite.

The accuracy to which the linear system in (11) is solved has to be at least as high as the desired accuracy in the eigenvalue calculation in order that the coefficients of the Lanczos three term recurrence are sufficiently accurate [15]. In our experiments, when we computed the transformed eigenvalues to an accuracy of about  $\varepsilon$ , we set the convergence tolerance for the system solver to  $\varepsilon/100$ .

In [2] we experimented with ILU and SSOR preconditioners. In general, they reduced the number of iteration steps. But taking the execution times into account we obtained the best results with diagonal preconditioning. In [4] we study hierarchical basis preconditioners as discussed by Bank [6] and Brenner [7] for box-shaped cavities. Hierarchical basis preconditioners are most natural if finite elements of higher orders are employed. Let us assume that the basis elements  $\{\Phi_i\}_{i=1}^n$  of  $W_h^{(2)}$  are arranged according to the direct sum decomposition (6). Then the equation  $(A - \sigma M)\mathbf{x} = \mathbf{y}$  can be written in the form

$$\begin{pmatrix} A_{11} - \sigma M_{11} & A_{12} - \sigma M_{12} \\ A_{21} - \sigma M_{21} & A_{22} - \sigma M_{22} \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{pmatrix} \quad (14)$$

where the order of the  $(1, 1)$  block is  $n_1 = \dim W_h^{(1)}$ . Let

$$\begin{aligned} D &:= \begin{pmatrix} A_{11} - \sigma M_{11} & 0 \\ 0 & A_{22} - \sigma M_{22} \end{pmatrix}, \\ L &:= \begin{pmatrix} 0 & 0 \\ A_{21} - \sigma M_{21} & 0 \end{pmatrix}. \end{aligned} \quad (15)$$

$n$	$n_1$	$t_C$	diagonal		BJ-pj1		BJ-pj2		BJ-pgs1		BJ-pgs2	
			[msec]	$t$ [sec]	$\bar{n}_{it}$	$t$ [sec]						
17571	2019	14.1	286	187	232	89	272	65	317	64	453	62
29211	3387	27.8	515	189	348	74	447	61	530	60	740	59
39899	4625	45.5	788	194	506	71	612	59	731	58	1051	57
56715	6609	71.0	1188	202	755	71	915	58	1075	56	1527	55
71155	8287	90.0	1627	212	983	72	1112	57	1338	56	1862	55
93147	10887	128.0	2264	228	1374	75	1502	57	1744	55	2486	54

$n$	$n_1$	$t_C$	BGS-pj1		BGS-pj2		BGS-pgs1		BGS-pgs2	
			[msec]	$t$ [sec]	$\bar{n}_{it}$	$t$ [sec]	$\bar{n}_{it}$	$t$ [sec]	$\bar{n}_{it}$	$t$ [sec]
17571	2019	14.1	355	70	279	42	274	36	330	33
29211	3387	27.8	496	56	420	35	462	34	558	31
39899	4625	45.5	710	55	570	34	640	33	787	30
56715	6609	71.0	1084	55	871	34	927	32	1121	29
71155	8287	90.0	1441	57	1100	35	1127	31	1384	29
93147	10887	128.0	1978	59	1497	36	1487	31	1778	28

Table 1: Execution times in seconds and average iteration numbers of outer iteration for various preconditioners. BS (Block Jacobi) and BGS (Block Gauss-Seidel) denote the outer iteration method. pjm (Point Jacobi) and pgs (Point Gauss-Seidel) denote the  $m$ -step interior iteration method.

Preconditioning  $A - \sigma M$  with  $D$  or with  $(D + L)D^{-1}(D + L^T)$ , respectively, amounts to applying a block Jacobi or symmetric block Gauss-Seidel iteration to (14) [6]. We however do not properly apply these block preconditioners. We do solve systems involving  $D_{11} = A_{11} - \sigma M_{11}$  directly with a multifrontal method [5] as the order of  $D_{11}$  is only  $n_1 = \dim W_h^{(1)} \approx n/8$ . The much larger systems involving  $D_{22} = A_{22} - \sigma M_{22}$  are solved by  $m$  steps of a stationary iteration called a smoother, in our case either damped Jacobi or symmetric Gauss-Seidel (i.e. SSOR with  $\omega = 1$ ). The damping factor in the Jacobi iteration was chosen to be  $2/3$ . Notice that with the Conrad-Wallach trick [18]  $m$  steps of symmetric Gauss-Seidel cost only as much as  $m+1$  steps of the ordinary Gauss-Seidel iteration.

## 6 NUMERICAL EXPERIMENTS

We computed the 10 smallest eigenvalues of the eigenvalue problem (7) with a domain closely approximating the shape of the new cavity for the 590 MeV ring cyclotron at the Paul Scherrer Institute, cf. Fig. 1. We computed the eigenvalues to a relative accuracy  $\varepsilon = 10^{-8}$ . In (10) we chose the spectral shift such that  $A - \sigma M$  is positive definite. The penalty parameter in (4) was set to  $s = 1.5$ . The computational results have been obtained on one processor of the PA-8000 processor-based HP Exemplar X-class system (180 MHz cycle, 720 MFlop/s peak) at ETH Zurich. We used the multiprocessor computer for its large memory. Results on a parallel implementation of the code are reported in [3].

In Tab. 1 we present timings for six problem sizes ranging from  $n = \dim W_h^{(2)} = 17571$  up to  $n = 93147$ .  $n_1$  denotes the dimension of the ‘coarse space’  $W_h^{(1)}$ .  $t$  is the overall time to solve a problem; the time for the con-

struction of the matrices is not included. In each iteration of the (restarted) Lanczos algorithm a system of equations has to be solved. All the eigenvalue problems listed in Tab. 1 required 44 steps in the IRL algorithm, i.e. 44 calls to SYMMLQ. SYMMLQ in turn needed in the average  $\bar{n}_{it}$  iterations until convergence which means that  $\bar{n}_{it} + 3$  linear systems of equations have been solved with the preconditioner [19]. The column in Tab. 1 labelled by  $t_C$  shows the times for solving a system with the matrix  $D_{11}$  in (15). Therefore, the time consumed by all coarse grid corrections is about  $44(\bar{n}_{it} + 3)t_C$ . This is 25-32% with BJ-pj1 and 9-13% with BGS-pgs1. The percentage increases with the problem size.

Diagonal preconditioning is satisfactory only for problem sizes smaller than those given in Tab. 1, cf. [4]. As the number of iteration steps increases with the problem size the performance of diagonal preconditioning deteriorates with increasing  $n$ .

As predicted by the theory, the iteration counts of the two-level methods are hardly affected by the problem size. In all problem sizes we observe that increasing the number  $m$  of smoothing steps lowers  $\bar{n}_{it}$  however not to the extent that the additional computing time is outweighed. In our examples, block Jacobi smoothed by one step of damped point Jacobi (BJ-pj1) performed best in all problem sizes. This may be surprising as this combination gave the highest average iteration count of all two-level methods. However, as we start the inner iteration with the zero vector, with Jacobi, the first smoothing step is just a multiplication of a diagonal matrix with a vector. Further steps of the Jacobi smoother as well as all steps of the Gauss-Seidel smoother require the access of two sparse triangular matrices, an operation which has a low flop count per memory access and thus performs at a low Mflop/s rate. There-

fore, the timings obtained with 2-step Jacobi and in particular with the Gauss-Seidel smoothers are higher than with BJ-pj1 although the average iteration count is significantly lower. If we should succeed in increasing the performance of the sparse matrix-vector multiplication, block Gauss-Seidel smoothed with one step of point Gauss-Seidel can be expected to perform best. Notice that because of the symmetry of  $A - \sigma M$  the computation of the residuals in SYMMLQ has a doubled flop count per memory reference.

Our results show that hierarchical preconditioners are well suited for the systems of equations that have to be solved in IRL. The number of iteration steps is independent of the problem size. The experiences made with point Jacobi indicates that a good smoother does not need to access much of  $A - \sigma M$  to be effective. An overlapping Schwarz smoother with small subdomains may therefore help to further improve the convergence rate. The time for the coarse grid correction could be reduced by moving from the two-level to a multilevel algorithm. With larger problem sizes the coarse grid correction may dominate the solution time and memory consumption of the eigensolver. We therefore will in future work investigate more economic ways to solve the coarse grid problem.

## 7 ACKNOWLEDGEMENT

We would like to thank Roman Geus of the Institute of Scientific Computing at ETH Zurich for performing the numerical experiments.

## 8 REFERENCES

- [1] St. Adam, P. Arbenz, and R. Geus, *Eigenvalue solvers for electromagnetic fields in cavities*, Technical Report #275, ETH Zürich, Institute of Scientific Computing, Computer Science Department, October 1997, (Available at URL <http://www.inf.ethz.ch/publications/>)
- [2] P. Arbenz and R. Geus, *A comparison of solvers for large eigenvalue problems originating from Maxwell's equations*, Numer. Linear Algebra Appl. **5** (1998).
- [3] ———, *Parallel solvers for large eigenvalue problems originating from Maxwell's equations*, Euro-Par '98 Parallel Processing (D. Pritchard and J. Reeve, eds.), Springer-Verlag, 1998, (Lecture Notes in Computer Science, 1470).
- [4] ———, *Two-level hierarchical basis preconditioners for computing eigenfrequencies of cavity resonators with the finite element method*, Recent Advances in Numerical Methods and Applications. Proceedings of the 4th International Conference on Numerical Methods and Applications (O. Iliev, M. Kaschiev, S. Margenov, Bl. Sendov, and P. Vassilevski, eds.), World Scientific, Singapore, 1999.
- [5] C. Ashcraft and R. Grimes, *The influence of relaxed supernode partitions on the multifrontal method*, ACM Trans. Math. Softw. **15** (1989), 291–309.
- [6] R. E. Bank, *Hierarchical bases and the finite element method*, Acta Numerica **5** (1996), 1–43.
- [7] S. C. Brenner, *Preconditioning complicated finite elements by simple finite elements*, SIAM J. Sci. Comput. **17** (1996), 1269–1274.
- [8] D. Calvetti, L. Reichel, and D. C. Sorensen, *An implicitly restarted Lanczos method for large symmetric eigenvalue problems*, Electronic Trans. Numer. Anal. **2** (1994), 1–21.
- [9] Ph. G. Ciarlet, *The finite element method for elliptic problems*, North-Holland, Amsterdam, 1978, (Studies in Mathematics and its Applications, 4).
- [10] D. R. Fokkema, G. L. G. Sleijpen, and H. A. van der Vorst, *Jacobi-Davidson style QR and QZ algorithms for the partial reduction of matrix pencils*, SIAM J. Sci. Comput. **20** (1999), 94–125.
- [11] V. Girault and P.-A. Raviart, *Finite element methods for the Navier-Stokes equations*, Springer-Verlag, Berlin, 1986, (Springer Series in Computational Mathematics, 5).
- [12] R. Grimes, J. G. Lewis, and H. Simon, *A shifted block Lanczos algorithm for solving sparse symmetric generalized eigenproblems*, SIAM J. Matrix Anal. Appl. **15** (1994), 228–272.
- [13] J. D. Jackson, *Classical electrodynamics*, 2nd ed., Wiley, New York, 1975.
- [14] F. Kikuchi, *Mixed and penalty formulations for finite element analysis of an eigenvalue problem in electromagnetism*, Comput. Methods Appl. Mech. Eng. **64** (1987), 509–521.
- [15] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK users' guide: Solution of large-scale eigenvalue problems by implicitly restarted Arnoldi methods*, SIAM, Philadelphia, PA, 1998, (The software and this manual are available at URL <http://www.caam.rice.edu/>).
- [16] R. Leis, *Zur Theorie elektromagnetischer Schwingungen in anisotropen Medien*, Math. Z. **106** (1968), 213–224.
- [17] K. Meerbergen and D. Roose, *Matrix transformations for computing rightmost eigenvalues of large sparse non-symmetric eigenvalue problems*, IMA J. Numer. Anal. **16** (1996), 297–346.
- [18] J. Ortega, *Introduction to parallel and vector solution of linear systems*, Plenum Press, New York, 1998.
- [19] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal. **12** (1975), 617–629.
- [20] B. N. Parlett, *The symmetric eigenvalue problem*, Prentice Hall, Englewood Cliffs, NJ, 1980.
- [21] G. L. G. Sleijpen and H. A. van der Vorst, *A Jacobi-Davidson iteration method for linear eigenvalue problems*, SIAM J. Matrix Anal. Appl. **17** (1995), 401–425.
- [22] D. Sorensen, *Implicit application of polynomial filters in a k-step Arnoldi method*, SIAM J. Matrix Anal. Appl. **13** (1992), 357–385.
- [23] H. Yserentant, *Hierarchical bases*, ICIAM 91: Proceedings of the Second International Conference on Industrial and Applied Mathematics (R. E. O'Malley, ed.), SIAM, Philadelphia, PA, 1992, pp. 256–276.

# Improved GdfidL with Generalized Diagonal Fillings and Reduced Memory and CPU Requirements

W. Bruns\*, Technische Universität Berlin, EN-2, Berlin, Germany

## Abstract

A new version of the Finite Difference code GdfidL implements generalized diagonal fillings for the discretization of the material distribution. The new algorithm allows more than 72 different types of material filling for each cell, whereas the common diagonal filling only allows 7 types. With the improved material filling, the discretization error for realistic geometries is reduced by a factor of ten. The improved meshing is implemented both in the resonant solver and the time domain solver. Computed frequencies for some simple geometries are given to show the reduction in discretization error. GdfidL's organisation of the computational volume as a linked list has already reduced the resource requirements to about the half of other codes. A new organisation of the linked list reduces the requirements for time domain computations again by a factor of 0.7.

## 1 INTRODUCTION

We want to compute electromagnetic fields. We want them accurate, we don't want to wait too long for them and we want to compute them within the memory limits of our computers. How to do this?

The first part deals with the accuracy. The next parts introduce a simple scheme to speed up both the FDTD-algorithm and the curl-curl-operator which is applied to find eigenfrequencies and fields.

## 2 ACCURACY

We can distinguish two kinds of error: The error in the approximation of the differential equation by the difference equation, and the approximation of the boundary conditions by the discretized material fillings. In order to have a small total error one has to have both errors at about the same size. When an (almost) homogeneous mesh is used, the error in the solution is proportional to the square of the mesh spacing. → We want to keep the error in the material filling at about the same size. There is little gained when the boundary conditions are discretized "perfectly" (except the mesh might look better). Near the boundaries, though, a "perfect"-mesh will be much better than what I present here.

How large is the error associated with the approximation of the differential equation by difference equations? For example, the error in a discretized rectangular cavity originates only from the error in the approximation of the fields, not from the approximation of the boundaries: Figure 1 shows the computed field in such a rectangular geometry

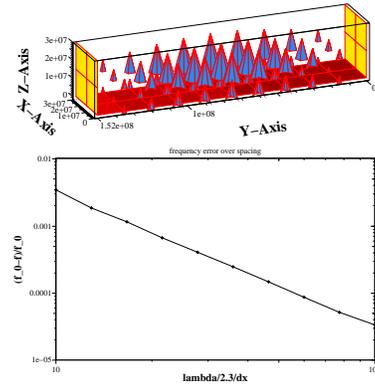


Figure 1: Above: Field in a simple rectangular resonator, Below: The error in the computed frequency. The error is proportional to  $\Delta^2$ .

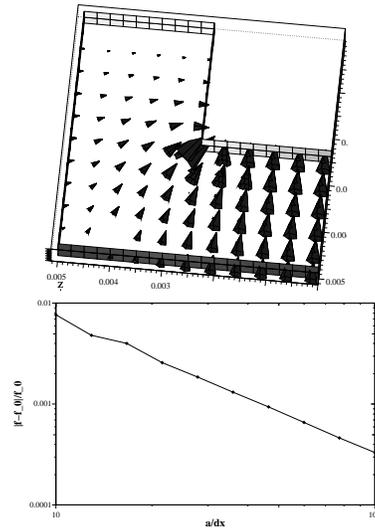


Figure 2: Above: Resonatorfield with singularity, Below: The error in the computed frequency. The error is proportional to  $\Delta^{4/3}$ .

and the error in the computed frequency. The frequency error is of second order, ie. when decreasing the stepsize by a factor of 10, the error is decreased by a factor of 100. This is the best possible error with standard Finite Differences.

In contrast to this, when one computes the field in a resonator with sharp edges of angle 90 degrees, the error is only of order 4/3, since the basis functions of the finite difference approach cannot model the field singularity well. Figure 2 shows such a geometry and the error in the computed frequency.

\* bruns@tetibm2.ee.TU-Berlin.DE

## 2.1 Improved mesh filling

Material-fillings are boundary conditions for the differential equations. The simpler finite difference programs approximate these boundary conditions by using a material filling where every grid cell is assumed to be homogeneously filled with a single material. This is the “staircase” approximation. A better approximation of the boundary conditions can be achieved with prismatic cells, as e.g. the MAFIA [3] group of codes uses them.

The filling with prismatic cells can be generalized. Since the finite difference coefficients for a field component depend only on the material in the immediate vicinity of the edge where the component is defined on, one can work easily with a mesh-filling that is constructed by a boolean combination of prismatic fillings. Figure 3 shows some of the possible discretized material distributions. A similar mesh filling is mentioned in [2]. Figure 4 sketches the procedure to evaluate the FD-coefficient for some field component.

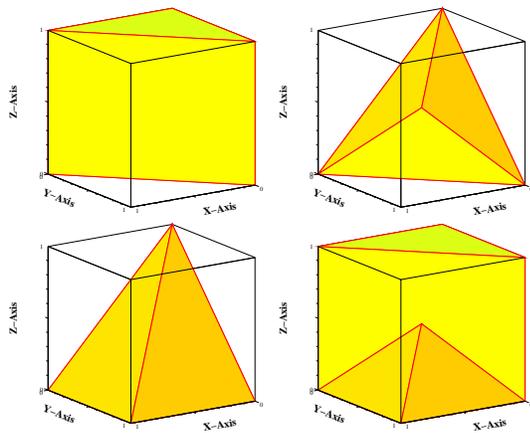


Figure 3: Some examples of the possible inhomogeneous fillings of a cell. Upper left: a prism. lower left: Intersection of two prisms. Upper right: Intersection of three prisms. lower right: Union of “upper left” and “lower left”. The prism in the upper left can be oriented in  $2 \times 3$  different kinds in a cell, the other three material fillings are possible in  $4 \times 3 \times 2$  different orientations.

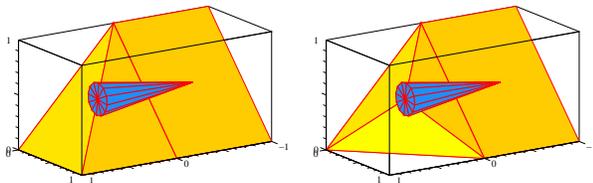


Figure 4: A magnetic field component touching some material boundaries. To evaluate the FD-coefficient for this component, one has to evaluate the effective permeability along the path where the component is defined on. The Finite Difference coefficient for *this* component is the same in both cases.

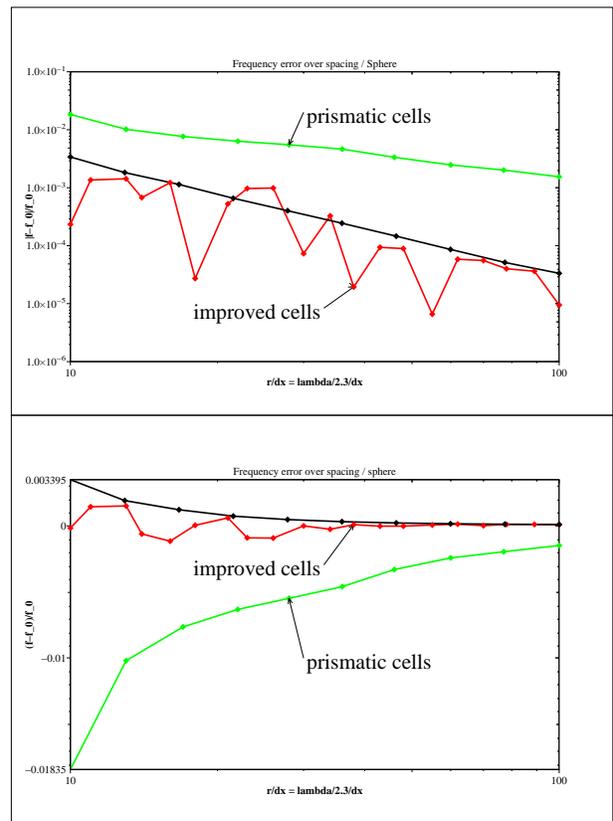


Figure 5: Error in the computed frequency of the lowest mode in a sphere. Above: absolute values in a double logarithmic scale, below: signed values in single logarithmic scale.

## 2.2 Examples, Effect of the new meshing

In order to show the effect of the generalized prismatic filling, figure 5 shows the computed resonance frequency in a sphere as a function of the mesh-spacing. For comparison, the results for prismatic filling and the optimal quadratic behaviour is plotted also. The error with the improved filling is about as low as the optimal quadratic behaviour. If the boundary conditions, ie. the materials would have been discretized perfectly, the result would not be much better.

A more interesting geometry is a reentrant cavity. Figure 6 shows the computed resonance frequency in such a geometry as a function of the mesh-spacing. For comparison, the results for prismatic filling is plotted also. Here the frequency error is not of second order. One can blame this on the small radius of the nose, where the field behaviour is almost singular. The frequency error is comparable to the error as shown in figure 2.

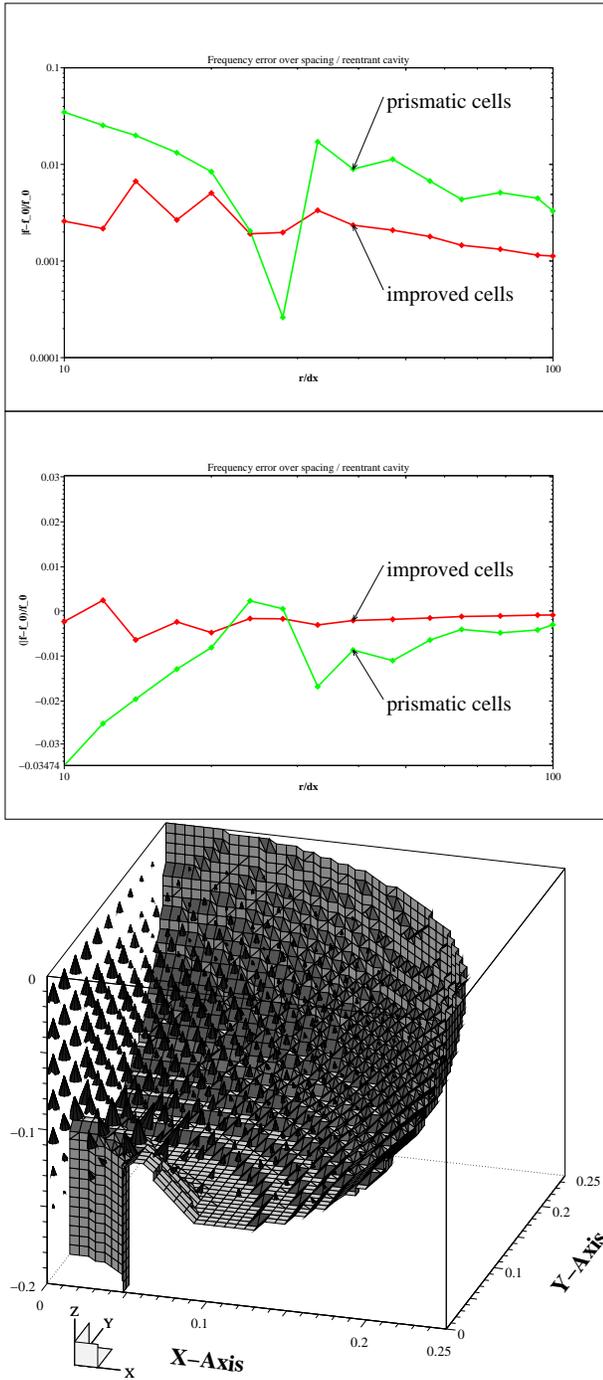


Figure 6: Error in the computed frequency of the first mode in a reentrant cavity. Above: absolute values in a double logarithmic scale, Center: signed errors versus planes / radius in logarithmic scale. Below: The geometry as discretized with 40 meshplanes / radius.

### 3 FASTER FD-IMPLEMENTATION

In addition to the commonly known FD-optimization that computes with  $\int Hds$  and  $\int Eds$  instead of  $H$  and  $E$  [1] two algorithmic improvements have been found and implemented.

#### 3.1 Single sweep through memory

Normally the H- and E-Update in FDTD-codes are performed as follows:

```
do
  Update all H-components,
  Update all E-components,
enddo
```

The above algorithm in every timestep reads all the E components to update all H-components, then reads all the H-components to update all E components. → In every timestep, all field-components are touched twice, but with a long CPU-time distance between.

Careful inspection of the FDTD-update reveals that it is possible to update the H components of a cell, and immediately after that the E-components can be updated, since they are no longer needed to update other H-components. → The second use of the field components will be much faster on cache-based computers, since the field components will already be in the cache. The old E-components can be overwritten, if the mesh is traversed e.g. in the natural order. E.g. for lossfree problems it might look like:

```
REAL, DIMENSION(1:3,0:nx+1,0:ny+1,0:nz+1) :: &
  Eds, Hds, dsoEpsA, dsoMueA
DO iz= 1, nz, 1
  DO iy= 1, ny, 1
    DO ix= 1, nx, 1
      Hds(1,ix,iy,iz)= Hds(1,ix,iy,iz) - dt*dsoMueA(1,ix,iy,iz) &
        * ( Eds(2,ix ,iy ,iz )-Eds(2,ix ,iy ,iz+1) &
          + Eds(3,ix ,iy ,iz )-Eds(3,ix ,iy+1,iz ) )
      Hds(2,ix,iy,iz)= Hds(2,ix,iy,iz) - dt*dsoMueA(2,ix,iy,iz) &
        * (-(Eds(1,ix ,iy ,iz )-Eds(1,ix ,iy ,iz+1)) &
          + Eds(3,ix ,iy ,iz )-Eds(3,ix+1,iy ,iz ) )
      Hds(3,ix,iy,iz)= Hds(3,ix,iy,iz) - dt*dsoMueA(3,ix,iy,iz) &
        * ( Eds(1,ix ,iy ,iz )-Eds(1,ix ,iy+1,iz ) &
          -(Eds(2,ix ,iy ,iz )-Eds(2,ix+1,iy ,iz )))
      !!
      !! Now we can update the Eds-components of the cell since
      !! they are no longer needed for the remaining H-updates.
      !! All used H-components have already been updated:
      !!
      Eds(1,ix,iy,iz)= Eds(1,ix,iy,iz) + dt*dsoEpsA(1,ix,iy,iz) &
        * (-(Hds(2,ix ,iy ,iz ) - Hds(2,ix ,iy ,iz-1)) &
          + Hds(3,ix ,iy ,iz ) - Hds(3,ix ,iy-1,iz ) )
      Eds(2,ix,iy,iz)= Eds(2,ix,iy,iz) + dt*dsoEpsA(2,ix,iy,iz) &
        * ( Hds(1,ix ,iy ,iz ) - Hds(1,ix ,iy ,iz-1) &
          -(Hds(3,ix ,iy ,iz ) - Hds(3,ix-1,iy ,iz )))
      Eds(3,ix,iy,iz)= Eds(3,ix,iy,iz) + dt*dsoEpsA(3,ix,iy,iz) &
        * (-(Hds(1,ix ,iy ,iz ) - Hds(1,ix ,iy-1,iz ) &
          + Hds(2,ix ,iy ,iz ) - Hds(2,ix-1,iy ,iz ) )
        ENDDO
    ENDDO
  ENDDO
```

A similar optimization can be applied in the discretized curl-curl operator that is used for eigenvalue computation. These optimizations save about 30 % CPU-time on typical desktop computers.

### 3.2 Better grid representation

The Finite Difference algorithm can be easily implemented as a computer code if one represents the fields as 4-D arrays. An example how simple the FDTD-update procedure might look was shown in the previous section. This approach has the disadvantage that memory (and CPU) is also used for cells that are totally inside perfect electric or magnetic materials. For many realistic geometries, the volume occupied by electric material is half of the total discretized volume or more.

One wants to use some scheme to only deal with the cells that really have a nonzero field.

One possible scheme could be the use of linked lists, where every field carrying cell carries the information about its neighbours. This needs 6 indices per field cell in addition to the 12 floating point words for the field components and their coefficients. This approach was used in the former "GdfidL" [4] [5].

The improved "GdfidL" has a grid organization that needs only a single index per cell, but also does not compute and store field components of cells that are totally surrounded by perfect electric or magnetic materials. The fields itself are stored in 2D-arrays, where the second index is the number of the cell. An INTEGER array "NrofCell" is used to extract the topology information. The FDTD-update with the single index per cell might look like:

```

REAL,DIMENSION(1:3,0:*) :: &
  Eds, Hds, dsoEpsA, dsoMueA
INTEGER, DIMENSION(0:nx+1,0:ny+1,0:nz+1) :: &
  NrofCell
DO iz= 1, nz, 1
  DO iy= 1, ny, 1
    DO ix= 1, nx, 1
      i= NrofCell(ix,iy,iz)
      IF (i .LT. 1) CYCLE !! skip when no field possible
      !!
      !! indices of neighbour cells in positive directions
      !!
      ipx= NrofCell(ix+1,iy ,iz )
      ipy= NrofCell(ix ,iy+1,iz )
      ipz= NrofCell(ix ,iy ,iz+1)
      Hds(1,i)= Hds(1,i) - dt*dsoMueA(1,i) &
        * ( Eds(2,i)-Eds(2,ipz) + Eds(3,i)-Eds(3,ipy) )
      Hds(2,i)= Hds(2,i) - dt*dsoMueA(2,i) &
        * (-(Eds(1,i)-Eds(1,ipz)) + Eds(3,i)-Eds(3,ipx) )
      Hds(3,i)= Hds(3,i) - dt*dsoMueA(3,i) &
        * ( Eds(1,i)-Eds(1,ipy) -(Eds(2,i)-Eds(2,ipx)))
      !!
      !! indices of neighbour cells in negative directions
      !!
      imx= NrofCell(ix-1,iy ,iz )
      imy= NrofCell(ix ,iy-1,iz )
      imz= NrofCell(ix ,iy ,iz-1)
      Eds(1,i)= Eds(1,i) + dt*dsoEpsA(1,i) &
        * (-(Hds(2,i) - Hds(2,imz)) + Hds(3,i) - Hds(3,imy) )

```

```

Eds(2,i)= Eds(2,i) + dt*dsoEpsA(2,i) &
  * ( Hds(1,i) - Hds(1,imz) -(Hds(3,i) - Hds(3,imx)))
Eds(3,i)= Eds(3,i) + dt*dsoEpsA(3,i) &
  * (-(Hds(1,i) - Hds(1,imy)) + Hds(2,i) - Hds(2,imx) )
  ENDDO
  ENDDO
ENDDO

```

## 4 CONCLUSION

An improved mesh filling has been implemented that reduces the frequency error by a factor of ten as compared to a prismatic filling. The frequency error from the boundary approximation is now in the range of the error due to the discretization of the differential equation itself. The time consuming field updates have been further optimized for modern cache based computers, giving a speed improvement of about 30% over the former GdfidL. The actual GdfidL is now typically 10 times as fast as a conventional FD-program. The memory consumption of GdfidL has been reduced again by a factor of 0.7. The memory consumption is typically only a quarter of that of conventional FD programs.

## 5 REFERENCES

- [1] Th. Weiland, "Ein Verfahren zur Berechnung von Wirbelströmen in massiven, dreidimensionalen, beliebig geformten Eisenkörpern", *etz Archiv*, H. 9 (1979), pp. 263-267
- [2] W. Müller, J. Krueger, A. Jacobus, R. Winz, T. Weiland, H. Euler, U. Hamm, W. R. Novender, "Numerical Solution of 2- or 3-Dimensional Nonlinear Field Problems by Means of the Computer Program PROF1", *Archiv für Elektrotechnik* 65 (1982) pp. 299-307
- [3] F. Ebeling, R. Klatt, F. Krawczyk, E. Lawinsky, T. Weiland, S. G. Wipf, B. Steffen, T. Barts, M. J. Browman, R. K. Cooper, H. Deaven, G. Rodenz, "The 3-D MAFIA group of electromagnetic codes", *IEEE Transactions on Magnetics*, vol. 25, pp. 2962-2964, July 1989
- [4] W. Bruns, "GdfidL: A Finite Difference Program for Arbitrarily Small Perturbations in Rectangular Geometries", *IEEE Transactions on Magnetics*, vol. 32, no. 3, May 1996, pp. 1453-1456
- [5] W. Bruns, "GdfidL: A Finite Difference Program with Reduced Memory and CPU Usage", *Proceedings of the PAC-97, Vancouver*, vol. 2, pp. 2651-2653, <http://www.triumf.ca/pac97/papers/pdf/9P118.PDF>

# SELF-CONSISTENT MODEL FOR THE BEAMS IN ACCELERATORS

A. Novokhatski and T. Weiland

TU Darmstadt, TEMF, Schlossgartenstr. 8, D-64289 Darmstadt, Germany \*

*Abstract*

It is proposed to use Ensembles of particles instead of "macro" particles for modeling the beams in accelerators. Each Ensemble describes the dynamics of the real bunch in the 6 dimensional phase space, taking into account all coupling effects, coming from the relativistic relation between momentum projections and energy. Ensemble parameters include average coordinate and momentum, bunch sizes, momentum spread and all second order correlation parameters of the bunch in the phase space. Self-consistent equations for the Ensemble parameters are derived from Vlasov equation. Examples of application of this model for the TESLA Linear Collider are presented.

## 1 INTRODUCTION

Usually for the beam dynamics calculations, the beam is described by a set of "macro" particles. "Macro" particle is an ensemble of particles for the bunch field calculation. However for the trajectory calculation, "macro" particle becomes a single particle. The motion of the particles inside "macro" particle is not considered. Therefore, for the emittance calculations of the beam with non zero energy spread, relatively large number of "macro" particles is needed.

## 2 PHASE DISTRIBUTION FUNCTION

There is another possibility to describe the beam by the phase distribution function  $f$  of particle density in the phase space of coordinates and momentum:  $(\vec{r}, \vec{p})$

$$f = f(t, x, y, z, p_x, p_y, p_z) \quad \int f(t, \vec{r}, \vec{p}) d\vec{r} d\vec{p} = 1$$

Normalized momentum  $(\vec{p})$  and energy  $(\gamma)$  are used

$$\vec{p} = \frac{\vec{P}}{mc} \quad \gamma = \frac{E}{mc^2} = \sqrt{1 + \vec{p} \cdot \vec{p}}$$

Phase distribution function  $f$  satisfies the Vlasov equation

$$\frac{d}{dt} f = \frac{\partial f}{\partial t} + \overrightarrow{grad}_r(f) \cdot \frac{\vec{p}}{\gamma} c + \overrightarrow{grad}_p(f) \cdot \frac{\vec{F}}{mc} = 0$$

The examples of of the distribution function on the two dimensional phase plane ( Energy- Coordinate), as the solution of Vlasov equation with noise and radiation damping (Fokker-Plank equation) are shown on the Fig.1 and Fig.2. Results of the simulation of the longitudinal instability in the damping ring are presented.

\* Work supported in part by DESY, Hamburg.

Our estimations of the amplitude and frequency of the surface roughness wake field [1] show that this field can be responsible for "saw-tooth" instability in the damping ring. On the pictures the shape of the distribution function is shown at the moment, when energy spread and bunch size get extreme values. Left graphic shows the energy distribution and right - the particle distribution together with cavity RF and wake voltage. The distributions of the bunch with small number of particles are shown by dotted lines. Upper curves show energy spread and bunch size in time, measured in the period of synchrotron oscillations.

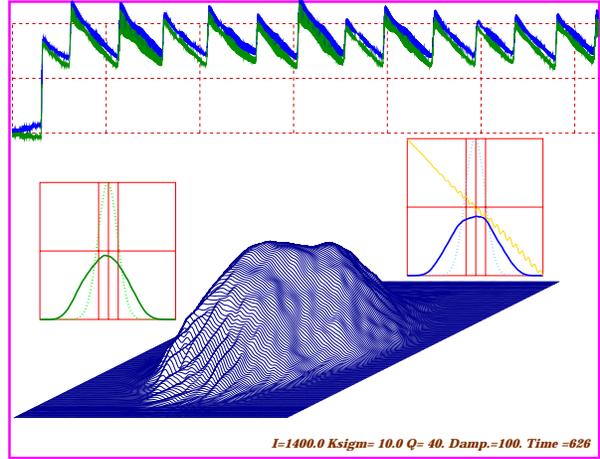


Figure 1: The shape of the distribution function at the moment, when energy spread and bunch size have maximum value.

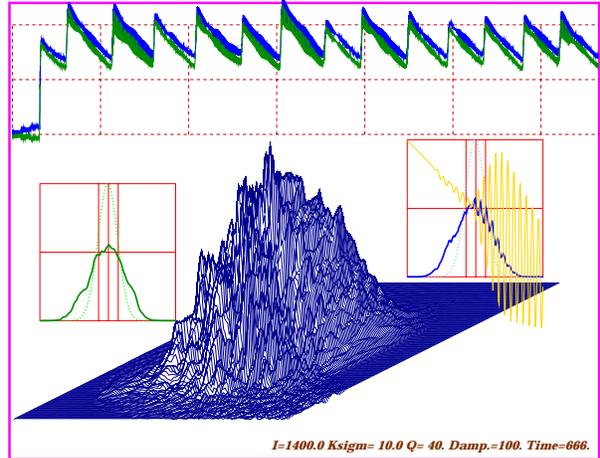


Figure 2: The shape of the distribution function at the moment, when energy spread and bunch size have minimum value.

Unfortunately, the direct numerical solution of the Vlasov equation in 6 dimensional case needs a great amount of the computer memory. If, for finite - difference approximation, we take only 50 mesh points in each

direction, then the total number of mesh points will be  $50^6 = 1.56 \cdot 10^{10}$ , that is equal to the number of real particles in the bunches of Linear Colliders.

### 3 SELF-CONSISTENT MODEL OF THE BEAM

However, the Vlasov equation can be numerically solved more easily for the bunch of particles, interacting with linear forces only. In this case the distribution function is described by a small number of parameters, only by the first and second order moments. Let us define such a bunch as an "Ensemble". If we can have full description of the dynamics of an Ensembles, then by a set of Ensembles we can describe the dynamics of any beam and all kinds of forces. Nevertheless, even one Ensemble can give a lot of information about beam dynamics in the Linear Collider.

The self-consistent equations for the Ensemble parameters are derived from Vlasov equation. Opposite to the usual model for the second order moments (nice description can be find in the book of A.Chao[2]), the model of Ensembles includes longitudinal motion of the particles, and all corresponding correlation with transverse motion. Let us define the Ensemble parameters.

#### 3.1 Ensemble Parameters

First order moments

$$\langle \xi \rangle = \int f(t, \vec{r}, \vec{p}) \xi d\vec{r} d\vec{p}$$

give the Average Position in the Phase Space

$$\langle x \rangle \quad \langle y \rangle \quad \langle z \rangle \quad \langle p_x \rangle \quad \langle p_y \rangle \quad \langle p_z \rangle$$

Second order moments

$$M_{\xi\nu} = \langle \xi\nu \rangle = \int f(t, \vec{r}, \vec{p}) (\xi - \langle \xi \rangle) (\nu - \langle \nu \rangle) d\vec{r} d\vec{p}$$

give Effective Beam Sizes (squared)

$$\begin{aligned} M_{xx} &= \sigma_x^2 = \langle (x - \langle x \rangle)^2 \rangle \\ M_{yy} &= \sigma_y^2 \\ M_{zz} &= \sigma_z^2 \end{aligned}$$

Correlated Sizes

$$\begin{aligned} M_{xy} &= \langle xy \rangle_{avc} = \langle (x - \langle x \rangle)(y - \langle y \rangle) \rangle \\ M_{xz} &= \langle xz \rangle_{avc} \\ M_{yz} &= \langle yz \rangle_{avc} \end{aligned}$$

Correlated Momentum Spread

$$\begin{aligned} M_{xp_x} &= \langle xp_x \rangle_{avc} = \langle (x - \langle x \rangle)(p_x - \langle p_x \rangle) \rangle \\ M_{yp_y} &= \langle yp_y \rangle_{avc} \\ M_{zp_z} &= \langle zp_z \rangle_{avc} \\ M_{xp_y} &= \langle xp_y \rangle_{avc} \\ M_{yp_x} &= \langle yp_x \rangle_{avc} \\ M_{xp_z} &= \langle xp_z \rangle_{avc} \\ M_{zp_x} &= \langle zp_x \rangle_{avc} \\ M_{yp_z} &= \langle yp_z \rangle_{avc} \\ M_{zp_y} &= \langle zp_y \rangle_{avc} \end{aligned}$$

Uncorrelated Momentum Spread (squared)

$$\begin{aligned} M_{p_x p_x} &= \sigma_{p_x}^2 \\ M_{p_y p_y} &= \sigma_{p_y}^2 \\ M_{p_z p_z} &= \sigma_{p_z}^2 \end{aligned}$$

Momentum Correlations

$$\begin{aligned} M_{p_x p_y} &= \langle p_x p_y \rangle_{avc} \\ M_{p_x p_z} &= \langle p_x p_z \rangle_{avc} \\ M_{p_y p_z} &= \langle p_y p_z \rangle_{avc} \end{aligned}$$

#### 3.2 Presentation by a Matrix

At the same time, the second order moments describe the 6 dimensional ellipse in phase space. If we combine the second order moments in a matrix  $M$

$$\begin{pmatrix} M_{xx} & M_{p_x x} & M_{yx} & M_{p_y x} & M_{zx} & M_{p_z x} \\ M_{xp_x} & M_{p_x p_x} & M_{yz} & M_{p_y p_x} & M_{zp_x} & M_{p_z p_x} \\ M_{xy} & M_{p_x y} & M_{yy} & M_{p_y y} & M_{zy} & M_{p_z y} \\ M_{xp_y} & M_{p_x p_y} & M_{yp_y} & M_{p_y p_y} & M_{zp_y} & M_{p_z p_y} \\ M_{xz} & M_{p_x z} & M_{yz} & M_{p_y z} & M_{zz} & M_{p_z z} \\ M_{xp_z} & M_{p_x p_z} & M_{yp_z} & M_{p_y p_z} & M_{zp_z} & M_{p_z p_z} \end{pmatrix}$$

then, the determinant of this matrix will give the volume (squared) of the ellipse

$$V^2 = \det\{M\}$$

As this determinant is invariant for translations and rotation of the ellipse, then the volume  $V$  is the 6 dimensional normalized emittance of the Ensemble. In the case of uncoupling motion the matrix takes the form

$$\begin{pmatrix} M_{xx} & M_{p_x x} & 0 & 0 & 0 & 0 \\ M_{xp_x} & M_{p_x p_x} & 0 & 0 & 0 & 0 \\ 0 & 0 & M_{yy} & M_{p_y y} & 0 & 0 \\ 0 & 0 & M_{yp_y} & M_{p_y p_y} & 0 & 0 \\ 0 & 0 & 0 & 0 & M_{zz} & M_{p_z z} \\ 0 & 0 & 0 & 0 & M_{zp_z} & M_{p_z p_z} \end{pmatrix}$$

Each, non zero  $2 \times 2$  matrix, represents the projection of the emittance on the coordinates plane  $\{xp_x\}$ , or  $\{yp_y\}$ , or  $\{zp_z\}$  and its determinant gives squared value of the emittance, like

$$\epsilon_x = \sqrt{\sigma_x^2 \sigma_{p_x}^2 - \langle xp_x \rangle_{avc}^2} = \sqrt{M_{xx} M_{p_x p_x} - (M_{xp_x})^2}$$

In this case the full emittance is the multiplication of projection emittances

$$V^2 = \epsilon_x^2 \epsilon_y^2 \epsilon_z^2$$

However, when the correlated parameters are excited, the projection emittances have to be changed. For example, when we excite  $M_{xp_z}$ , the full emittance is the difference of positive values

$$V^2 = \epsilon_y^2 (\epsilon_x^2 \epsilon_z^2 - M_{xp_z}^2 \sigma_{p_x}^2 \sigma_x^2)$$

So, to keep it invariant, the x- and z- emittance projections have to be increased.

#### 3.3 Time Equations

Time equations for Ensemble parameters can be derived from the Vlasov equation under two assumptions:

1) The full emittance is invariant only for the forces, that satisfy the condition

$$\langle \mu \bullet \overrightarrow{\text{grad}}_p \frac{\vec{F}}{mc^2} \rangle = 0$$

for any Ensemble parameter  $\mu$ .

2) The energy spread in the beam is not very large and the energy can be presented in the following expanded way

$$\begin{aligned} \frac{1}{\gamma} &= \frac{1}{\gamma_m} - \frac{1}{\gamma_m^3} \sum_n [\langle p_n \rangle (p_n - \langle p_n \rangle) + \\ &\quad + \frac{1}{2} ((p_n - \langle p_n \rangle)^2 - M_{p_n p_n})] \end{aligned}$$

where  $\gamma_m$  is the average beam energy

$$\gamma_m = \sqrt{1 + \sum_n \langle p_n \rangle^2 + M_{p_n p_n}}$$

Under these conditions the average velocity  $\vec{v}$  contains additionally the momentum correlations parts

$$\mathbf{v}_n = \langle \frac{p_n}{\gamma} \rangle = \frac{\langle p_n \rangle}{\gamma_m} - \frac{1}{\gamma_m^3} \sum_k \langle p_k \rangle M_{p_n p_k}$$

Now we can use average value relation

$$\frac{\partial}{c\partial t} \langle \mu \rangle = \langle \overrightarrow{grad}_r \mu \bullet \frac{\vec{p}}{\gamma} \rangle + \langle \overrightarrow{grad}_p \mu \bullet \frac{\vec{F}}{mc^2} \rangle$$

for deriving time equations:

Average Trajectory

$$\frac{\partial}{c\partial t} \langle l \rangle = \frac{\langle p_l \rangle}{\gamma_m} - \frac{1}{\gamma_m^3} \sum_n \langle p_n \rangle M_{p_l p_n}$$

$$\frac{\partial}{c\partial t} \langle p_l \rangle = \langle \frac{F_l}{mc^2} \rangle$$

Coordinate Coordinate Relations

$$\begin{aligned} \frac{\partial}{c\partial t} M_{lk} &= \frac{1}{\gamma_m} (M_{kp_l} + M_{lp_k}) - \\ &- \frac{1}{\gamma_m^3} (\langle p_l \rangle \sum_n \langle p_n \rangle M_{kp_n} + \langle p_k \rangle \sum_n \langle p_n \rangle M_{lp_n}) \end{aligned}$$

Coordinate Momentum Relations

$$\begin{aligned} \frac{\partial}{c\partial t} M_{lp_k} &= \frac{1}{\gamma_m} M_{p_l p_k} - \frac{1}{\gamma_m^3} \langle p_l \rangle \sum_n \langle p_n \rangle M_{p_k p_n} + \\ &+ \langle (l - \langle l \rangle) \frac{F_k}{mc^2} \rangle \end{aligned}$$

Momentum Momentum Relations

$$\frac{\partial}{c\partial t} M_{p_l p_k} = \langle (p_l - \langle p_l \rangle) \frac{F_k}{mc^2} \rangle + \langle (p_k - \langle p_k \rangle) \frac{F_l}{mc^2} \rangle$$

## 4 PROPERTIES OF ENSEMBLE

### 4.1 Fundamental Conservation Laws

We can check the model for realization of the dynamics laws for average values of the bunch. It is easy to show, that from presented equations one obtains:

$$\frac{\partial}{c\partial t} \langle \vec{p} \rangle = \langle \frac{\vec{F}}{mc^2} \rangle$$

$$\frac{\partial}{c\partial t} \langle \gamma \rangle^2 = \frac{\partial}{c\partial t} \gamma_m^2 = 2 \langle \vec{p} \bullet \frac{\vec{F}}{mc^2} \rangle$$

$$\frac{\partial}{c\partial t} \langle \vec{M} \rangle = \frac{\partial}{c\partial t} \langle \vec{r} \times \vec{p} \rangle = \langle \vec{r} \times \frac{\vec{F}}{mc^2} \rangle$$

So, the model fulfills the fundamental conservation laws for momentum, energy and angular momentum.

### 4.2 Emittance Equation

We can also estimate the modification of the full emittance for the case, when the presented assumptions are not fulfilled. When we have only longitudinal force and transverse average momentums are zero, the model gives the following equation for longitudinal emittance

$$\frac{\partial}{c\partial t} \epsilon_z^2 = 2M_{zz} \langle (p_z - \langle p_z \rangle) \frac{F_z}{mc^2} \rangle - 2M_{zp_z} \langle (z - \langle x \rangle) \frac{F_z}{mc^2} \rangle$$

When the force is proportional to the momentum

$$F_z = \alpha mc^2 p_z$$

the equation takes the following form

$$\frac{\partial}{c\partial t} \epsilon_z^2 - 2\alpha \epsilon_z^2 = 0$$

This equation is solved very easily. In agreement with the sign of  $\alpha$ , the full emittance will exponentially increase, or decrease (friction force) with the time

$$\epsilon_z = \epsilon_z^0 \exp(\alpha ct)$$

and  $1/\alpha$  is the effective distance of the effect.

At the same time the transverse emittances are invariant

$$\frac{\partial}{c\partial t} \epsilon_x^2 = 0 \quad \frac{\partial}{c\partial t} \epsilon_y^2 = 0$$

### 4.3 Bunch Compression in Free Space

Here we check the modification of the bunch in free space, when the bunch has negative correlated momentum spread  $M_{xp_x}^0 < 0$ , or  $M_{yp_y}^0 < 0$ , or  $M_{zp_z}^0 < 0$

In case, when  $\langle p_z \rangle \gg \langle p_x \rangle$

$$M_{xx} = M_{xx}^0 + 2 \frac{c\Delta t}{\gamma_m} M_{xp_x}^0 + \left(\frac{c\Delta t}{\gamma_m}\right)^2 M_{p_x p_x}^0$$

$$M_{zz} = M_{zz}^0 + 2 \frac{c\Delta t}{\gamma_m^3} M_{zp_z}^0 + \left(\frac{c\Delta t}{\gamma_m^3}\right)^2 M_{p_z p_z}^0$$

Minimum size of the bunch is determined by the emittance and uncorrelated momentum spread, as it usually is

$$\min(\sigma_x^2) = M_{xx}^0 - \frac{(M_{xp_x}^0)^2}{M_{p_x p_x}^0} = \frac{\epsilon_x^2}{M_{p_x p_x}^0} = \frac{\epsilon_x^2}{\sigma_{p_x}^2}$$

## 5 APPLICATION OF THE MODEL

The derived equations for Ensemble parameters are nonlinear. For some simple cases we can find analytical solution. However, to use these equations for computer simulations correct numerical algorithm is needed. We present examples of analytical and numerical solution.

### 5.1 Ensemble in an Accelerating Structure

From this model we can have the estimation for the frequency  $\varpi$  of the longitudinal oscillations of a bunch in an accelerating field. Let the average center of the bunch move with the speed, equal to the phase velocity of the accelerating wave, having the phase displacement  $\varphi_0$ . The linear part of the force is

$$\frac{F_z}{mc^2} = E + \delta E(z - \langle z \rangle)$$

$$E = \frac{e}{mc^2} E_{acc} \cos \varphi_0 \quad \delta E = \frac{e}{mc^2} E_{acc} \frac{2\pi}{\lambda} \sin \varphi_0$$

For the longitudinal bunch size (squared) the model gives the equations

$$\frac{\partial}{\partial t} M_{zz} = \frac{2}{\gamma_m^3} M_{zp_z}$$

$$\frac{\partial}{\partial t} (\gamma_m^3 \frac{\partial}{\partial t} M_{zp_z}) + 4\delta E M_{zp_z} = \delta E M_{zz} \frac{\partial \gamma_m^3}{\partial t}$$

From which we can estimate  $\varpi$

$$\varpi/c = \sqrt{\frac{\delta E}{\gamma^3}} = \sqrt{\frac{2\pi}{\lambda} \frac{eE_{acc}}{\gamma^3 mc^2} \sin \varphi_0}$$

If the sign of  $\sin \varphi_0$  is negative, we have "exponential" growth of the longitudinal beam size.

## 5.2 Space Charge Effect

The force, acting on particles, moving together is

$$\vec{F} = \frac{e}{\gamma} * \frac{\vec{p}(\vec{p} \cdot \vec{R}) + \vec{R}}{((\vec{p} \cdot \vec{R})^2 + R^2)^{3/2}}$$

Where  $\vec{p}$  is the momentum,  $\vec{R} = \vec{r}_0 - \vec{r}$  is the vector between coordinates of the particles. To find the space charge force in the bunch with homogenous charge density, one have to calculate the force from "uncompensated" charge only. This charge "appears", when the point of observation is shifted from the average center (Fig.3). This approach

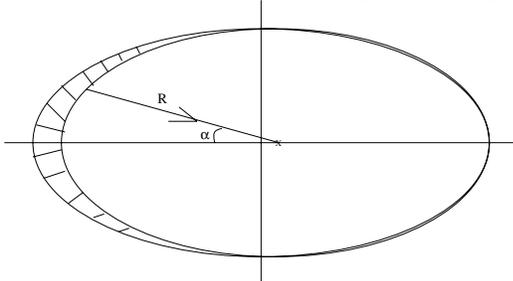


Figure 3: Effective charged area, that forms the force in the point X near the average center.

gives estimation for the space charge force

$$\vec{F} = \frac{eQ}{\gamma^2} \frac{\vec{R}}{V_g}$$

The force is linear with the distance from the average center and inversely proportional to geometrical volume

$$V_g = \det \begin{pmatrix} M_{xx} & M_{xy} & M_{xz} \\ M_{yx} & M_{yy} & M_{yz} \\ M_{zx} & M_{zy} & M_{zz} \end{pmatrix}$$

This formula and the model give the equations for the charged bunch in free space. For the transverse beam size and correlated momentum spread

$$\sigma_x = \sqrt{M_{xx}} \quad \sigma_{p_x} = \sqrt{M_{p_x}}$$

we have

$$\frac{\partial}{\partial t} M_{xx} = \frac{2}{\gamma_m} M_{xp_x} \quad \varkappa = \frac{r_0 N}{\gamma_m^2 V_g}$$

$$\frac{\partial^2}{\partial t^2} M_{xp_x} - \frac{4\varkappa}{\gamma_m} M_{xp_x} = 0$$

For longitudinal beam size and correlated momentum spread we have the same equation, but with the

$$\varkappa_l = \varkappa / \gamma_m^2$$

The equations give growing in time solutions. Effective length  $L_{eff}^\perp$ , where the space charge really enlarges the transverse beam size is

$$L_{eff}^\perp = \sqrt{\frac{\gamma_m}{\varkappa}} = \sqrt{\frac{\gamma_m^3 \sigma_x \sigma_y \sigma_z}{r_0 N}}$$

The effective parameter for the longitudinal size is  $\gamma_m$  larger. However this size does not increase too much, as the transverse enlargement decrease the charge density very soon.

## 5.3 Ensemble in the Homogeneous Magnetic Field

Imagine, that relativistic bunch has initial energy spread  $\delta\gamma$  before injection into the magnetic field. Then the particles with different energy will have different radiuses of rotation and therefore different time for one turn, as they are moving with the speed close to the speed of light

$$T = \langle T \rangle \frac{\gamma}{\langle \gamma \rangle}$$

where  $\langle T \rangle$  and  $\langle \gamma \rangle$  are the average period and energy. The bunch size  $\sigma$  is increasing in time  $t$  as

$$\sigma = \langle R \rangle \sin(2\pi \frac{t}{\langle T \rangle} \frac{\delta\gamma}{\langle \gamma \rangle})$$

where  $\langle R \rangle$  is the average radius. After the time

$$T = \frac{\langle T \rangle \langle \gamma \rangle}{4 \delta\gamma}$$

the bunch takes the circumference of the circle.

Now we use numerical calculations to study this phenomena in the frame of our model. We carry simulations for the bunch with initial energy spread of  $\pm 1\%$ . Comparison of the numerical and analytical results for the bunch size is given in Fig.4. Results for the bunch momentum and energy spread are shown on Fig.5. After 25 turns the center of bunch comes to center of rotation and the beam size reaches the value of rotation radius. This position is not stable, but it is repeated with the period of 50 turns. After 25 turns average momentum and its projections become zero, but energy spread reaches maximum value.

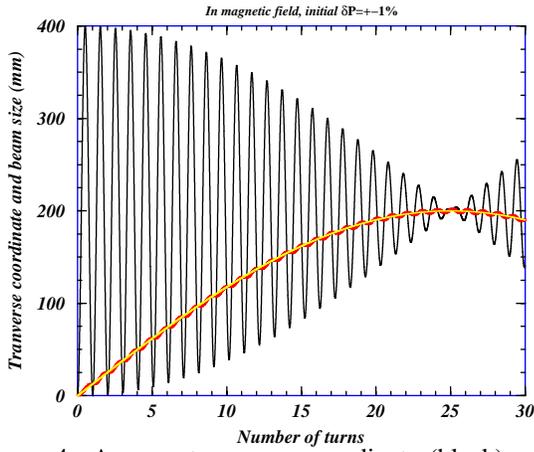


Figure 4: Average transverse coordinate (black) and the bunch size (red). Yellow line presents the analytical formula.

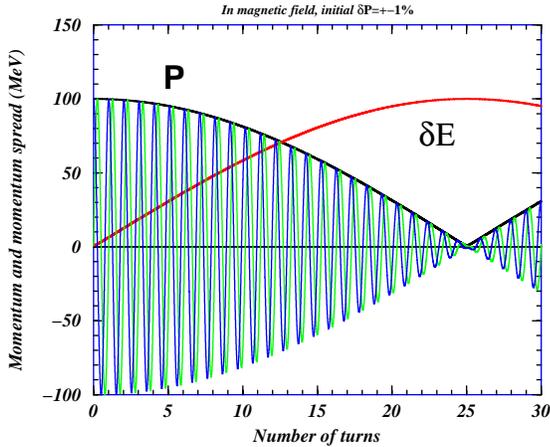


Figure 5: Average bunch momentum (black), its projections (blue and green) and energy spread (red).

#### 5.4 Chicane Bunch Compression

The TESLA bunch compressor consists of four rectangular dipole magnets, where bunch at first is deflecting in transverse direction and then is forwarding back. When the bunch gets the transverse deflection, it also gets correlated moments  $M_{x p_z}, M_{z p_x}$  and  $M_{p_x p_z}, M_{x z}$ , that change the emittance projections. In symmetrical chicane the emittance projections come back to the initial values. If the second pair of magnets have another magnetic strength, then the bunch do not come to the initial transverse position (non symmetric chicane). The transverse beam size increases and the transverse emittance grows up. Emittance growth can be estimated by formula

$$\Delta \epsilon_x^2 = \frac{X^2}{\gamma_m^2} M_{p_z p_z}^0 M_{p_x p_x}^0$$

Where  $X$  is the difference of initial and final transverse coordinate. Another effect, that can change the compression parameters and increase the emittance projections, is the action of the space charge forces. Results of computer simulation for the bunch with the charge of 1 nC are presented on Fig.6 and Fig.7.

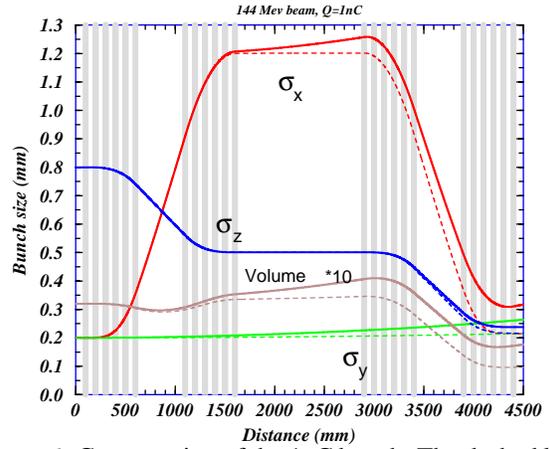


Figure 6: Compression of the 1nC bunch. The dashed lines show results without consideration of the space charge effect.

Longitudinal bunch size is compressed from 0.8 mm to 0.22 mm. Bunch volume is changed mainly in the second half of the chicane. Space charge effect do not greatly change the ratio of the longitudinal compression, but increase transverse bunch size. Space charge forces destroy the symmetry of the compression system. Transverse emittance does not come back to initial value. Emittance increase approximately five times.

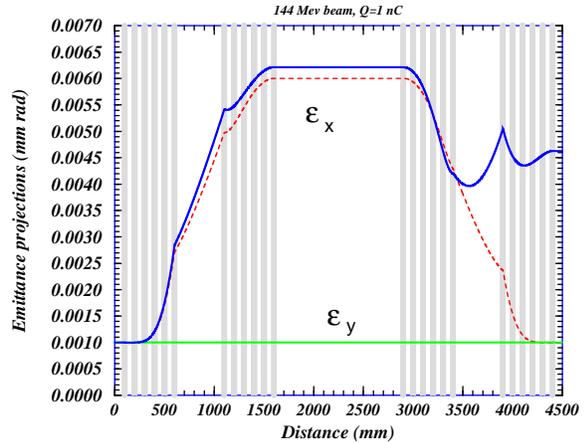


Figure 7: Emittance projections. The dashed lines show results without consideration of the space charge effect.

## 6 ACKNOWLEDGEMENTS

The authors greatly appreciate useful discussions with Reinhard Brinkmann and Jorg Rossbach and Star Track Team.

## 7 REFERENCES

- [1] A.N.Novokhatski, M.Timm and T.Weiland, "The Surface Roughness Wake Field Effect", Proceedings of this Conference.
- [2] A.Chao, Physics of Collective Beam Instabilities in High Energy Accelerators, (J.Wiley & Sons, New York,1993)

# USING SYMBOLIC ALGEBRA FOR THE GENERATION OF ORBIT SIMULATION CODES FROM HAMILTONIANS

Andreas Adelman\* and Stefan Adam,  
Paul-Scherrer-Institute, CH-5232 Villigen, Switzerland

## Abstract

As a part of our new research activity aiming at a detailed understanding of space charge effects in ring cyclotrons and in the corresponding injection beam lines at the Paul Scherrer Institute we are currently developing a three dimensional space charge simulation code.

The use of the Hamiltonian formalism, in combination with a symbolic algebra system (MAPLE), enables us to carry out the entire modelling on a high level. MAPLE easily handles the elaborate derivatives required to form the equations of motion and then casts them into FORTRAN. The subsequent embedding of the FORTRAN code into the MATLAB package is also handled automatically by the procedures of our framework. MATLAB is well suited for running small simulations followed by various post processing activities including graphics.

In order to test this framework, two model cases have been chosen: the double focusing spectrometer and the motion of a single particle in a stable charged cloud. The successful processing of these and some further small model problems encourages us to apply this framework to produce code for the large scale simulation.

The development steps from the Hamiltonian to the FORTRAN subroutine and the resulting simulations are shown for two model cases, as well as the general MAPLE and MATLAB procedures we have developed for this purpose.

Keywords: space charge, symbolic algebra, code generation, Hamiltonian system

## 1 HISTORY AND MOTIVATION

The acronym FORTRAN stands for **FOR**mula **TRAN**slation. The computer language FORTRAN was originally developed by John Backus in 1954 and first released 1957 by IBM [1] [2]. FORTRAN was one of the first high level computer languages in which the programmer specifies the task of the computer in terms of human readable text. This resulted in programs being easier to read, understand and debug. It also allows the programmer to concentrate on the actual problem, rather than getting diverted into details of the underlying computer architecture. If we look more closely however, the formulas written in the natural language of mathematicians and physicist still have to be translated into an artificial language understood by the computer. This need for an additional transforma-

tion step is present for all other programming languages. For a long time this has been a major motivation in the search for a direct way to translate a mathematical description into computer code [6] [8]. The advent of symbolic algebra systems opened the option of entering problems in the mathematical language. With the option to automatically generate code from formulas the two diverging languages approached each other.

We extend that further, towards a automated generation of a small scale orbit simulation starting from the Hamiltonian. The benefits of this are:

1. Time efficient generation of orbit simulation code
2. Quality of the resulting code with respect to correctness
3. Test of the physical and mathematical model, based on analytically solvable examples

## 2 SYMBOLIC MANIPULATION

### 2.1 Problem Frame

Following the definition of a computer experiment [10], we introduce two categories: **1. Large Scale Simulation** and **2. Small Scale Simulation**. The large simulation will meet our scope to simulate the space charge effects of huge a number of particles in a complicated accelerator structure. Given the complexity of the large problem, a successive approach using several small steps is appropriate. Figure 2.1 shows how the two categories interact. After a few iterations of the *loop*, comprising the Small Scale Simulation, the mathematical model is expected to be sufficiently validated to justify the use of the corresponding code in the large scale simulation (see *is used by* in Figure 2.1).

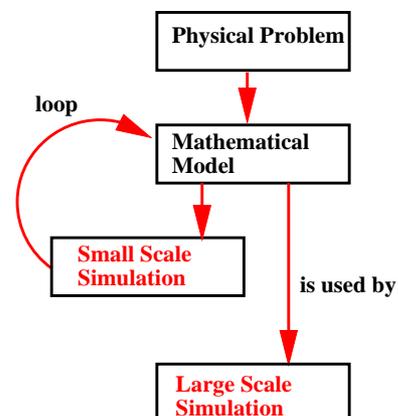


Fig. 2.1 The Frame of Discussion

\* Corresponding Author: Andreas Adelman (Andreas.Adelmann@psi.ch)

The description of the problem starts with its Hamiltonian

$$H(\mathbf{q}, \mathbf{p}; t) \quad (1)$$

from where we can get the corresponding Hamiltonian equations

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i}, \quad i = 1 \dots 3 \quad (2)$$

$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}, \quad i = 1 \dots 3 \quad (3)$$

which then lead to the equations of motions.

## 2.2 Modelling Stages for Orbit Simulations

We are interested in the trajectories of particles guided by various forces, such as: external forces, collective forces (space charge), etc.

$$H(\mathbf{q}, \mathbf{p}; t) = H_{ExternalForces} + H_{SpaceCharge} \quad (4)$$

The effect of a guiding magnetic field on charged particles is best expressed with the vector potential:

$$\mathbf{A}(\mathbf{q}; t). \quad (5)$$

In the sequel we neglect space charge ( $H_{SpaceCharge} = 0$ ) and define the unified notation:

$$\mathbf{y} = (\mathbf{q}, \mathbf{p})^T. \quad (6)$$

With this, the equation of motion form a system of first order differential equations,

$$\frac{d\mathbf{y}}{dt} = F(\mathbf{y}, t) = \left( \frac{\partial H}{\partial p_i}, -\frac{\partial H}{\partial q_i} \right)^T \quad (7)$$

which we finally have to solve.

## 3 THE SIMULATION SHELL

Assume one has (7), out of a symbolic algebra system. To move on and build a small scale simulation, one might consider three alternatives:

1. remain in the symbolic algebra system to solve (7) [7]
2. automatic transfer of (7) into a programming language and coding the simulation (see Figure 2.1).
3. **transfer of (7) into a programming language and joining this code to a numerical simulation package.**

To achieve our goal of validating the mathematical and physical model with minimal programming effort (in the sense of 3rd generation programming languages), we follow the third alternative.

We have chosen the MAPLE package [5] for all symbolic manipulations, and use the extended capabilities of MACROFORT [9] for code generation and apply MATLAB [11] for the numerical part.

As a starting point we have to build symbolic expression for the Hamiltonian (4). With symbolic differentiation, MAPLE can evaluate the equation (7) for the given Hamiltonian. The resulting expressions are **automatically translated** into FORTRAN subroutines.

In order to use external Fortran subroutines within the MATLAB environment, we have to build an intermediate glue layer, the so-called MEX-Interface [11]. The Mex-Interface uses a gateway subroutine in addition to the original external function (see Figure 3).

This glue layer itself is fully defined by the signature of the external function, denoted as  $F$  and  $A$  in our example shown in figure 3.

With this procedure, we generate functions which represent  $F$  in (7) and ( $A$ ) from (5). Those can then act like MATLAB built-in functions.

The MATLAB environment allows to solve (7) numerically using built-in, or external solvers for the system of ordinary differential equations (ODE's), as the problem requires.

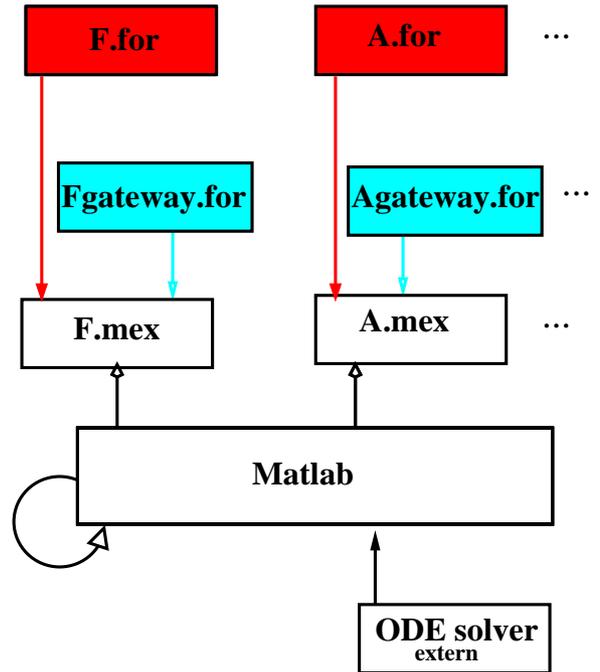


Fig. 3 The Simulation Shell

## 4 EXAMPLES

Two examples will be used to show in detail the steps from the abstract formulation towards a running small scale simulation.

#### 4.1 The double focusing Spectrometer

The mathematical formulation translates one to one into MAPLE syntax, as can be seen below.

The Hamiltonian for a particle in an external magnetic field is given by Barut [3]:

$$H = c \cdot \sqrt{(\mathbf{p} - e_0 \mathbf{A}(\mathbf{q}; t))^2 + m_0^2 c^2} \quad (8)$$

where  $\mathbf{A}$  is the vector potential. The case in consideration does not contain electrostatic fields, therefore we omit those terms.

To obtain the characteristics of the double focusing spectrometer [13], a rotationally symmetric, radially decreasing field has to be chosen, with a radial component of:

$$Br = B \cdot R^n \text{ and } R = \sqrt{(q1^2 + q2^2)}, \quad (9)$$

where the field index  $n$  amounts to 0.5. This field has the corresponding vector potential with the azimuthal component  $A_\theta$  that can be obtained by algebraic manipulation:

$$A_\theta = -\frac{1}{R} \cdot \int R \cdot Br \cdot R^{n_{Field}} \cdot dR + \frac{q_3^2}{2} \cdot \frac{\partial}{\partial R} (Br \cdot R^{Field}) \quad (10)$$

#### 4.1.1 Symbolic Algebra Part

Assuming the expressions for (10) with  $n$  as parameter are stored as elements of the matrix  $A$ , the Hamiltonian can be written in the MAPLE algebra system as follows:

$$H := c * ( (p[1,1] - e0 * A[1,1])^2 + (p[1,2] - e0 * A[1,2])^2 + (p[1,3] - e0 * A[1,3])^2 + m0^2 * c^2 )^(1/2) :$$

This corresponds to the definition of the Hamiltonian (4) where space charge is neglected.

To obtain the equations of motion (7), a set of derivatives has to be formed, entering the MAPLE commands:

```
for i to DIM do
  d_eqs[i] := diff(H, Y[eval(i+DIM)]);
  d_eqs[eval(i+DIM)] := diff(-H, Y[i]);
od;
```

The parameter  $DIM$  defines the dimension of the system to be studied.

The generic code generation procedure  $gen\_F$ , contains an additional parameter  $NPART$ , the number of particles used in the simulation.

---

```
gen_F : proc(DIM, NPART)
  local j, pg;
  0  pg := [];
  1  pushe(['declaref', 'doubleprecision',
  2      [T, Y[2*DIM*NPART], YPP[2*DIM*NPART]]], 'pg');
  3  pushe(['commonf', 'GLOBAL', GLOBALLIST], 'pg');
  ...
  5  for j from 1 to 2*DIM do
  6      pushe([equalf, YPP(j), d_eqs[j]], 'pg');
  od;
  ...
  9  genfor(pg);
end;
genfor(gen_F(DIM, PART)):
```

---

**Fig. 4.1.1** Fragment of the code generation procedure

In this procedure we make use of functions provided by the MACROFORT package [9].

With the MACROFORT commands *pushe* (line 1 - 6) the list *pg* is filled with information related to declarations and statements of the code to be generated. Finally, this information is translated into the FORTRAN subroutine *Ffor* upon the command *genfor* (on line 9). The FORTRAN sub-

routine *Afor* is obtained in a similar way, therefore the detailed description is omitted here.

#### 4.1.2 The Numerical Part

In the MATLAB environment we do the numerical simulation, which starts by defining initial conditions for position and momentum of the particle. To calculate these initial

conditions the vector potential function  $A$  is used to get the mechanical momentum from the canonical one.

```

n      = 0.5;

q10   = 0.0;
q20   = 0.5;
...
p0    = 0.0;
...
Ainit = A(q10,q20,q30,n);

p10   = -p0 + Ainit(1);
p20   = 0.0 + Ainit(2);
...
y0    = [q10 q20 q30 p10 p20 p30 ]';

```

We then use the Runge Kutta Fehlberg (4th order) integrator from MATLAB to integrate the ODE system.

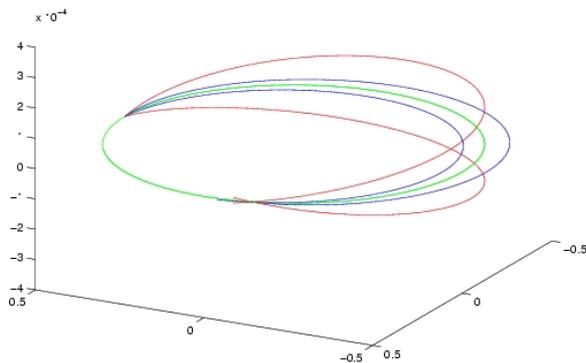
Note that in these samples of Matlab code the functions  $A$  and  $F$  cannot be distinguished from original MATLAB built-in functions.

```
[t1,y1]= ode45('F',TSPAN,y0,options);
```

Finally, one can use a variety of graphical and other post processing functionalities available in MATLAB.

```
plot3(y1(:,1),y1(:,2),y1(:,3),'g');
```

The plot below shows the characteristics of the double focusing spectrometer: horizontally and vertically diverging orbits meet at the azimuthal angle  $\sqrt{2} \cdot \pi$  from the start.



**Fig. 4.1.2** Particle Trajectories in a Double Focusing Spectrometer

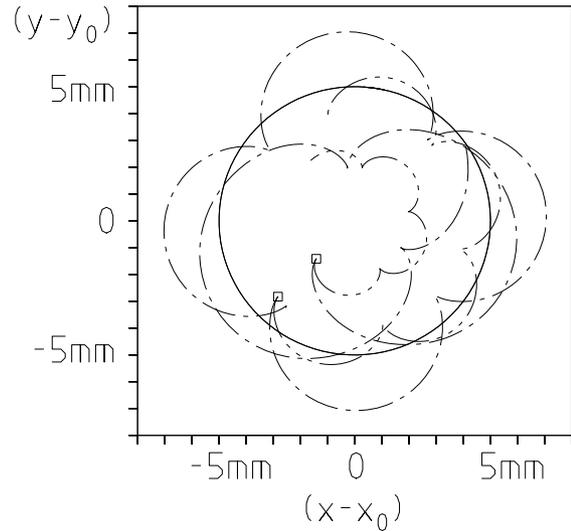
#### 4.2 Particles in a Space Charge Cloud

As another test case to verify the outlined procedure for automatically generating orbit simulation code, we consider a

particle moving in a charged cloud. The governing Hamiltonian becomes:

$$H(\mathbf{q}, \mathbf{p}; t) = H_1 + H_2 + H_{SpaceCharge} \quad (11)$$

We then proceed in a way similar to the previous example and obtain results, which are in agreement with [4], see Figure 4.2.



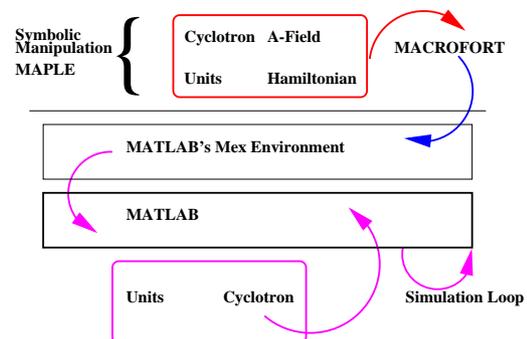
**Fig. 4.2** 2D Particle Motion inside a Charged Cloud

The generality of the presented scheme, for the automatic generation of orbit simulation code, encouraged us to extend the case of particle motion in a space charged cloud to three dimensions.

### 5 CONCLUSIONS AND FUTURE

We have successfully established a framework for the automatic generation of orbit simulation code, without using any 3rd generation programming language.

Starting with the formulation of the Hamiltonian in mathematical notation, the error prone tasks of forming derivatives and of coding in FORTRAN are fully handled by MAPLE. The successive inclusion into the MATLAB environment is done by automatic generation of the MEX-Interface. The essential parts of this framework and their interactions are shown in Figure 5.



**Fig. 5** The Framework of Automatic Codegeneration

This framework is useful for validating physical and mathematical models, by using well-known existing components:

1. MAPLE and MACROFORT
2. MATLAB
3. ODE solvers

Essential software components to be contained in a large scale simulation can be easily produced and tested within this framework:

Equations

⇒ SW-Components

⇒ tested SW-Components

We will further use this approach to generate important code parts for our research project:

*The construction of a three dimensional beam simulation code for high intensity proton beams. This code will allow us to study the behaviour of space charge dominated beams at injection into the PSI Injector 2. [12]*

We plan the extension of this framework in the following aspects:

- Simulations based on symplectic maps
- Interface to a parallel computing environment
- Rewrite MACROFORT to handle the programming language C

### Acknowledgements

We are indebted to the Professors R. Eichler and R. Jeltsch and to Dr. T. Stambach for their support of our project and for many fruitful discussions.

## 6 REFERENCES

- [1] J. Backus. Can programming be liberated from the von Neuman style? *Communications of the ACM*, 21, Aug. 1978.
- [2] J. Backus. The history of FORTRAN I, II, and III. In R. L. Wexelblat, editor, *History of programming languages. Proceedings of the ACM SIGPLAN conference (Los Angeles, Calif., June 1–3, 1978)*, pages 25–74, New York, NY, USA, June 1981. Academic Press.
- [3] A. Barut. *Electrodynamics and Classical Theory of Fields and Particles*. Dover, 1980.
- [4] A. C. Chasman. Space charge effects in a heavy ion cyclotron. *Nuclear Instruments and Methods in Physics Research*, pages 279–283, 1984.
- [5] B. W. Char, K. O. Geddes, G. H. Gonnet, B. Leong, M. B. Monagan, and S. M. Watt. *First Leaves: Tutorial Introduction to Maple*. Springer-Verlag, New York, 1992.
- [6] J. S. Cohen. The effective use of computer algebra systems. In *Transactions of the Sixth Army Conference on Applied Mathematics and Computing*, pages 677–698, 1989.
- [7] E.S.Cheb-Terrab and H. de Oliveira. Poincare sections of hamiltonian systems. *Phys-Pub Universidade de Estado do Rio de Janeiro and submitted to CPC*, January 1996.
- [8] J. Fitch. Mathematics goes automatic. *Physics world*, 6(6):48–52, June 1993.
- [9] C. Gomez. MACROFORT: A FORTRAN code generator for MAPLE. Technical report, Institut National de Recherche en Informatique et en Automatique, 1990.
- [10] R. Hockney and J. Eastwood. *Computer Simulation using Particles*. Adam Hilger, 1988.
- [11] I. T. MathWorks. MATLAB 5. URL: <http://144.212.100.10/>, 1997.
- [12] PSI-Home-Page. Injector 2. URL: <http://www.psi.ch>, 1998.
- [13] K. Siegbahn. *Alpha-, Beta- and Gamma-Ray Spectroscopy*. North-Holland, 1964.

# THE FRAMEWORK OF UNIFIED ACCELERATOR LIBRARIES

Nikolay Malitsky and Richard Talman  
Newman Laboratory, Cornell University, Ithaca, NY 14853

## Abstract

An overview of the framework of Unified Accelerator Libraries (UAL) is presented. The UAL framework is a necessary and logical step in the UAL evolution. It intends to offer a standard-based infrastructure that facilitates compatible and independent implementation of accelerator applications and the development of interoperable Accelerator Simulation Facilities. The paper explain the two major framework components: a uniform mechanism for assembly and reuse of independently developed accelerator algorithms and a uniform infrastructure for optimization and correction approaches.

## 1 RATIONALE

Object-oriented class libraries have provided an important advance in the simulation of accelerator performance but their effective exploitation will rely on the environment or “framework” in which they are employed. In comparison with class libraries, frameworks offer additional benefits: strong “wired-in” interconnection between classes, customization mechanisms, and default behavior and architectural guidance for applications. The strong infrastructure significantly eases the implementation, management, and deployment of critical domain solutions, decreases the amount of standard code that a developer has to program, test, and debug, and frees the scientist from software design to apply his or her expertise on project-specific applications.

The Unified Accelerator Libraries (UAL[1]) have been designed as an open highly flexible environment built from diverse loosely-coupled components. This approach has facilitated the development of new accelerator applications and the selection of the most effective accelerator physics algorithms, analysis and design patterns. The accumulated experience in developing UAL applications, emerging new component-oriented technologies and enterprise-level distributed frameworks and systems have enabled the implementation of the UAL framework, its analysis and design patterns. The UAL framework is a necessary and logical step in the UAL evolution. It is simultaneously a product of the UAL environment and a tool for the development of a UAL-based Accelerator Simulation Facility. The next sections explain in detail the two major framework components: a uniform mechanism for assembly and reuse of independently developed accelerator algorithms and a uniform infrastructure for optimization and correction approaches.

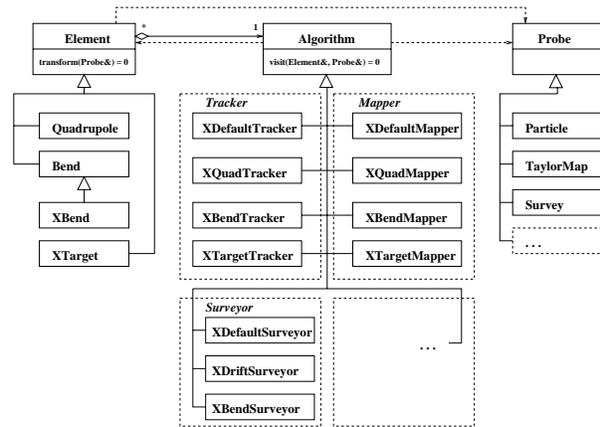


Figure 1: Structure of the Element-Algorithm-Probe analysis pattern.

## 2 UNIFORM MECHANISM FOR ASSEMBLY AND REUSE OF INDEPENDENTLY DEVELOPED ALGORITHMS

The main architectural principle of the Unified Accelerator Libraries is a separation of physical entities and mathematical abstractions from algorithms. The diverse accelerator algorithms are implemented as classes that share data via Common Accelerator Objects (Particle, Twiss, Element, and others). This distributed open architecture has promoted the introduction of the Element-Algorithm-Probe analysis pattern (see Fig. 1).

According to the Element-Algorithm-Probe analysis pattern, Common Accelerator Objects are divided into two categories (Element and Probe), and accelerator algorithms are implemented as separate interchangeable classes<sup>1</sup>. This structure is built after the Strategy design pattern [2] that lets one vary algorithms independently of their context, making them easier to switch, understand, and extend. The Strategy pattern provides many benefits by offering an alternative to subclassing and conditional statements for selecting desired behavior. However, it does not determine a mechanism for selecting algorithms relevant to the concrete element type. There is the Visitor design pattern [2] that declares the *visit* operation for each type of the concrete element and lets one define a new operation without affecting class hierarchies. A Visitor gathers related operations and makes adding new operations easy. However, it prevents subclassing new element types and freezes existing element class hierarchies. Despite the simplicity and

<sup>1</sup>Here we are introducing a (unconventional) view of the particles or beams as “probing” the accelerator structure.

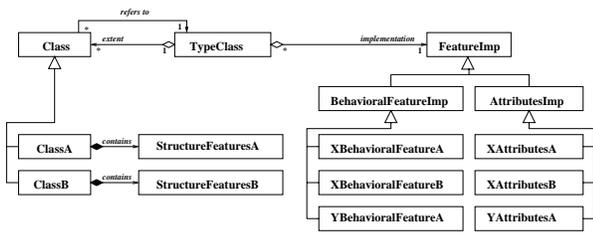


Figure 2: Structure of the Mutable Class design pattern.

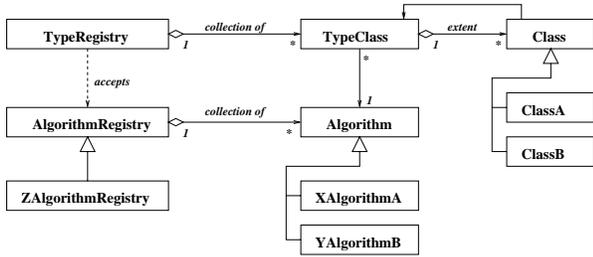


Figure 3: Structure of the Mutable Class Type Registry.

power of the Visitor pattern, this drawback makes it unacceptable for open scientific applications. To address this problem we have derived the Mutable Class design pattern (see Fig. 2).

A Mutable Class is a combination of two design patterns, a Type Object [3] and a Bridge [2]. A Type Object divides a class into two parts, a type class and instance class, and an object’s behavior is delegated to its type. A Bridge decouples an abstraction from its implementation so that the latter can be varied independently. This structure offers an alternative approach to subclassing and results in a highly flexible system configurable at run-time. On the other side, the Mutable Class model is well defined and can be considered as the physical implementation of the Class structural element described in the following Unified Modeling Language (UML) Semantics[4]:

“A Class defines the data structure of Objects, although some Classes may be abstract, i.e no Objects can be created directly from them. Each Object instantiated from a Class contains its own set of values corresponding to the StructureFeatures declared in the full descriptor. Objects do not contain values corresponding to BehavioralFeatures or class-scope Attributes; all Objects of a Class share the definition of the BehavioralFeatures from the Class, and they all have access to the single value stored for each class-scope attribute”.

The TypeClass instances can be grouped together in the uniform collection and be controlled by the separate object, a Type Registry (or Finder). Registry services have been included in many complex design patterns and industrial frameworks. They deal with issues related to querying operations on managed objects. The Mutable Class Type Registry extends its functionality by providing the additional mechanism for dispatching and binding algorithms with the corresponding TypeClass instances (see Fig. 3):

An AlgorithmRegistry is preferable to a Visitor for open

configurable systems. One can consider it as an alternative approach to the Acyclic Visitor pattern [3] that intends to break the dependency cycle of the original Visitor pattern using multiple inheritance. Inheritance is a static mechanism and results in strong coupling between components. The Mutable Class pattern offers a different approach based on a powerful dynamic composition technique and dividing a visiting process into two sequential steps:

1. dispatching and binding algorithms with the TypeClass instances; (in the UAL framework, it corresponds to binding accelerator algorithms with the twenty to thirty ElementType instances).
2. performing operations on elements of the complex structure; (for accelerator applications, the number of operations is determined by the lattice size and the number of repetitions and ranges from 500 to infinity.)

Implementation of the UAL framework based on the Element-Algorithm-Probe analysis pattern and the Mutable Class design pattern will create a universal mechanism for sharing, mixing, replacing, versioning, and extending diverse accelerator algorithms and will support the following actual applications:

- inclusion of new element types or physical effects ( *combined function magnet (FNAL Recycler)*, *superposition of solenoid and quadrupole fields (CESR)*, *parasitic beam-beam effects (LHC, CESR)*);
- uniform access to accelerator data from diverse algorithms ( *conventional tracking*, *spin*, *space charge*, *etc.*);
- optimal single accelerator representation for lattice design and performance modeling ( *optimal combination of “thick” and “thin” elements*).

The proposed structure is intended to facilitate the extensions and maintenance of existing UAL components. For examples, the TEAPOT algorithms for different element types are represented by protected class methods and are selected by the engine according to the set of element attributes. In the Element-Algorithm-Probe framework, the TEAPOT monolithic engine will be split into an extendible collection of tracking algorithms. This promotes the (independent) development of new algorithms for propagating different accelerator “probes”, such as Spin, Space Charge, and others.

### 3 UNIFORM INFRASTRUCTURE FOR OPTIMIZATION AND CORRECTION APPROACHES

The accelerator life cycle has several successive stages: design, construction, commissioning, and operation. In accelerator terminology, optimization procedures correspond to the early design phase, and correction is associated with

accelerator commissioning and operation. Different applications of these approaches result in the principle distinctions in their structure and algorithms. Optimization modules have been implemented as parts of accelerator simulation programs and have access to all kinds of information that is unavailable from the real accelerator. Usually, all data exchanges are hidden inside the simulation code “input language”. The correction modules depend on the accelerator environment (hardware, operating system, *etc.*) and perform actual compensation procedures based only on detector measurements. Despite these differences, both approaches can be described by the same formula: *adjust correction elements in such a way as to minimize the deviation from required behavior measured in detector elements.* The UAL framework intends to use this formula for building a uniform infrastructure for optimization and correction approaches. A single environment for developing simulation and operational software would significantly facilitate its implementation, improve the quality and reliability of algorithms, and promote the exchange and integration of diverse approaches.

TEAPOT [5] is an operational code that was targeted to implement a universal operational algorithm [6] based on the internal accelerator simulator. This approach and these algorithms were successfully reused in the working prototype of the High Level Control Code for the SSC Low Energy Booster [7]. However, the constraints of the FORTRAN and C languages in which the original TEAPOT versions were written, prevented the direct integration of the TEAPOT libraries with operational control systems. Now, modern object-oriented technologies promote the modular structure of accelerator codes. A typical accelerator application can be represented by several cooperating components (Accelerator Simulator, Control, Interface, *etc.*) that communicate via Common Accelerator Objects (Detectors, Adjusters, *etc.*) (See Fig. 4). The UAL framework will provide the services for building these objects and the reliable infrastructure and mechanism to support their operation. In the present C++ version of the TEAPOT program, all correction algorithms are represented by separate class-services. Each service has direct access to the lattice elements and internally selects families of adjusters and detectors. In the new scheme, TEAPOT algorithms will interact with UAL systems via self-defined external Adjusters and Detectors instances and will be organized as replaceable modules of the Control component. This distributed architecture of the UAL environment will reflect the general structure of the accelerator control system and will make straightforward the development of an off-line Accelerator Simulation Facility and the conceptual model of an online Accelerator Simulation Facility (See Fig. 4).

#### 4 REFERENCES

[1] N.Malitsky and R.Talman: *Unified Accelerator Libraries*, AIP 391, Williamsburg, 1996

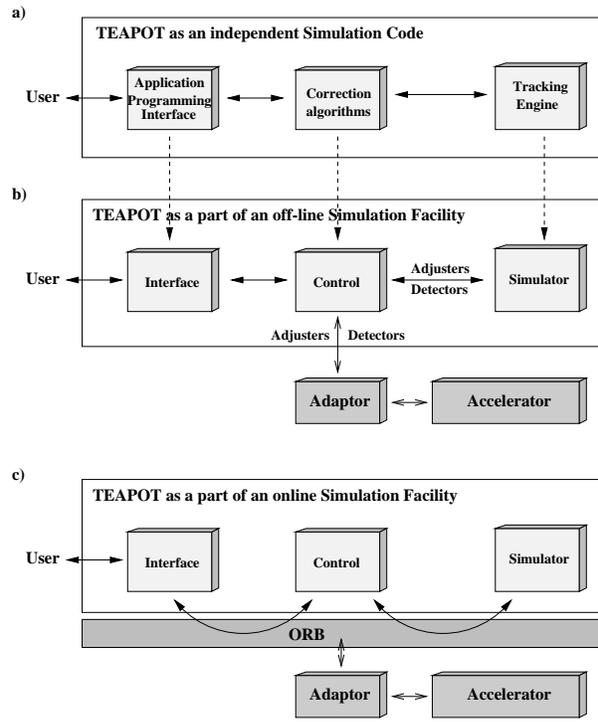


Figure 4: a) An independent Simulation Code vs. b) an off-line Simulation Facility vs. c) an online Simulation Facility (An off-line simulation facility will be the application of the UAL framework).

[2] E.Gamma, R.Helm, R.Johnson, and J.Vlissides: *Design Patterns: Elements of Reusable Software Architecture*. Addison-Wisley, 1995.

[3] *Pattern Languages of Program Design 3*, edited by R.Martin, D.Riehle, F.Buschmann, Addison-Wisley, 1997.

[4] *UML Semantics*. OMG document number: ad/97-08-04.

[5] L.Schachinger and R.Talman: *Teapot: A Thin-Element Accelerator Program for Optics and Tracking*, Particle Accelerators, **22**, 35(1987).

[6] R.Talman: *A Universal Algorithm for Accelerator Correction*, AIP 255, 1991.

[7] G.Bourianoff, A.Reshetov, N.Malitsky: *Object-Oriented Approach for the Design of the Simulation Facility of the SSC*, SSCL-677, 1994.

# JPP: A PARTICLE TRACKING CODE WITH JAVA

Ryoichi Hajima\*,  
Dept. Quantum Eng. & Systems Sci., University of Tokyo,  
Hongo 7-3-1 Bunkyo-ku, Tokyo 113-8656 Japan

*Abstract*

A computer language Java is a possible solution for the standardization of accelerator software to improve software productivity and realize common user interface. In this paper, JPP: a particle tracking code written in Java is presented as an example of Java application for accelerators. We discuss the advantage and disadvantage of developing scientific software with Java and investigate the performance of Java code in comparison with C code.

## 1 INTRODUCTION

In this section, we glance over the trend of software development and its application for accelerators and suggest that Java is a possible solution for the standardization of accelerator software, which improves software productivity and realizes common user interface.

### 1.1 *Software for Accelerators : Development and Utilization*

Computer software is widely used in accelerator field for modeling, designing, controlling, data acquisition and so on. A lot of programming languages and scripts have been suggested to construct the software. Computer hardware for these application is also variety from super computers to personal machines. Even a palm-top PC will be used as a personal terminal for scientific application in near future.

This variety of software and hardware confuses both programmers and users. Programmers often spend much time to translate their software from one platform to another platform, and users must learn minor difference existing in software translated from another platform. This is the reason why we desire the standardization of computer software for accelerators.

A programming language Java is a possible solution for the standardization and has already been utilized widely in industrial applications, because Java has a lot of fascinating features, which are architecture-neutral, tightly specified, robust and secure, and so on [1]. In scientific scene, however, it has been considered that a computer language running on an interpreter such as Java is not acceptable due to its poor performance.

In the present study, we describe a charged particle tracking code written in Java to discuss the advantage and disadvantage of Java in scientific applications, and investigate the validity of existing technologies to speed up Java.

## 2 JAVA FOR NUMERICAL SIMULATIONS

In this section, we discuss the advantage and disadvantage of Java in scientific applications.

### 2.1 *Advantage*

#### 2.1.1 Portable

Portability is the first distinguished feature of Java. Once we compile Java source code into byte code, the byte code is architecture-neutral and executable on any machine and any operating system as far as Java virtual machine (Java-VM) is available.

The portability is also a matter of concern for a program developer, because it promises freedom in choice of developer's environment. He can write and debug a code efficiently with his favorite editor on his familiar computer environment.

#### 2.1.2 Object-Oriented

Java is object-oriented programming language and can be used to make class libraries for accelerator software, which have been constructed with C++.

#### 2.1.3 Robust and Secure

Java provides a robust and secure methods to compose a program. For example of the robustness is that Java has no pointer data types defined in C and C++. The pointer is sometimes useful and permits us to make a tricky routine, but it has potential risk to cause unexpected memory destruction. The simple memory management model in Java releases us from run-time errors caused by abusing of memory management.

#### 2.1.4 User Interface

Providing a common user interface is helpful for application users to learn the usage of new application without confusion. Java has its own graphical package AWT (Abstract Window Toolkit) to realize common user interface on different platforms.

#### 2.1.5 Multithreading

We can write multithreaded applications with Java. Multithreading provides better interactive responsiveness and real-time behavior in interactive applications. Writing a multithreaded program is also enable us to scale up the performance of numerical applications, if the program is executed on a PC of multi-CPU.

---

\* e-mail: hajima@q.t.u-tokyo.ac.jp

## 2.2 Disadvantage

### 2.2.1 Performance

Performance of a program, computing speed, is matter of utmost concern to discuss the superiority of computer language for numerical simulations. It has been considered that Java has great disadvantage on this point, and is not acceptable for a numerical simulation. When we execute a Java program, Java VM interprets Java byte code and passes it to the CPU, in short words, Java is running on an interpreter. Interpreting byte code is generally much slower than executing native code directly. This is the reason why Java has only poor performance in numerical simulations.

People in Java, however, have made successive effort to improve the performance of Java and several solutions are becoming available, which are JIT, native compiler, native method, HotSpot VM.

## 3 SPEED-UP OF JAVA

As seen in the last section, improvement of performance, that is speed-up of calculation, is a key issue to make a numerical simulation with Java. In this section, we discuss some technologies improving the performance of Java.

### 3.1 JIT

The most popular way to speed up Java nowadays is JIT (Just-In-Time compiler). It is one of standard features supported by popular Java virtual machines such as JDK, Netscape Communicator, Internet Explorer. Just-in-time compiler translates whole of Java byte code into optimized native code on-the-fly and executes this native code, then the performance can be improved much better than a classical Java-VM, an interpreter. Since a JIT-VM reads a Java byte code and executes it as a classical-VM does, the portability of byte code is completely satisfied. The start up time overhead in the translation from byte code into native code is not a matter in numerical simulations, because calculation time is generally much longer than the overhead.

### 3.2 Native Compiler

Another straight way to speed up Java is using a native compiler, which translates byte code into native code and saves it as an executable file. One can directly run the executable file without Java VM. The code optimization in native compiler is superior to JIT-VM, because native compiler can spend more time for the optimization. The executable file is not architecture-neutral and the portability of Java is lost.

### 3.3 Native Method

If a specific routine in a Java program spends a large part of calculation time, we can save calculation time by rewriting this time consuming routine in C or Fortran and linking it to Java byte code. This is called as native method. In JDK (Java Development Kit), native method is implemented as

JNI (Java Native Interface). The portability is lost as well as native compiler.

### 3.4 HotSpot VM

A new Java virtual machine named HotSpot is under development and released soon by Sun Microsystems [2]. The HotSpot VM includes several advanced technologies: adaptive optimization, generational garbage collection, hot spot detection and so on. It has been reported that the performance of HotSpot VM is three times as fast as the current JIT VM in JDK for typical benchmark problems [3].

## 4 DESCRIPTION OF JPP

A charged particle tracking code written in Java has been developed as an example of Java application for accelerators and the efficiency of existing technologies to speed up Java has been investigated. In this section, we describe the outline of the particle tracking code: JPP.

### 4.1 Class Hierarchy

The class hierarchy in JPP is designed as shown in figure 1. The root class is AccSystem.class which stores global parameters for calculations such as fundamental RF frequency, the number of active particles, time step for tracking. These parameters are kept as private variables to prevent unexpected overwriting of values in the simulation, but they can be referred from all the subclasses through public methods which return the value of private variables.

We prepare AccElement.class as a prototype of accelerator element and a class for each element such as drift and bending magnet is defined as a subclass of AccElement.class. It is easy to define a new subclass for a new accelerator element.

A routine for space charge calculation is defined in SpaceCharge.class which has several subclasses for various space charge models.

In particle tracking, each particle is treated as an instance of Particle.class, where variables of motion are stored as instance variables.

We also define ParticleSource.class for a particle generator which initializes particle coordinates in six-dimensional phase space of motion according to input command. Another class, Jpp.class is prepared for the main routine of particle tracking.

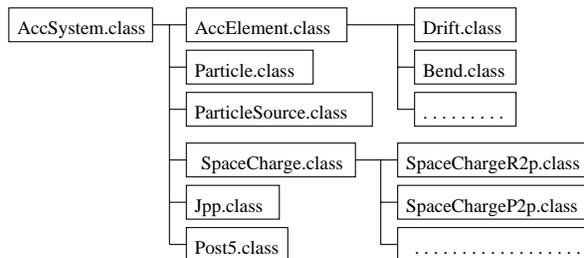


Figure 1: Class hierarchy in JPP.

## 4.2 Instance Methods of Accelerator Elements

All the accelerator elements defined in JPP have three basic instance methods to describe particle motion in the element, which are *inject()*, *eject()* and *advanceParticle()*. These methods are defined in *AccElement.class* and overridden in subclasses as needed.

### 4.2.1 inject()

This method is called when a particle injects into the element and performs necessary procedures at the entrance of element. It checks if an injected particle exists inside the aperture. When the particle is out of the aperture, it is rejected from the active particle list. Some elements require another procedure, for example, transverse momentum of the particle is modified at the entrance of a bending magnet with the fringe field and the edge focus.

### 4.2.2 eject()

This method is called when a particle ejects from the element. The particle coordinates in six-dimensional phase space at the exit of element is stored in output buffer of the element. The particle is, then, passed to the next element. In a bending magnet, the particle momentum is changed before the coordinates are recorded.

### 4.2.3 advanceParticle()

This method describes how a particle moves in the element. Every subclass of accelerator element has its own procedure for this method, because it is an intrinsic method characterizing the element.

## 4.3 Space Charge Calculation

Several subclasses are prepared for space charge calculation. Three different space charge models are implemented in JPP, which are ring-charge model known as SCHEFF in PARMELA [5], point-charge model [6] and line-charge model [7]. All the models are also available as native method with JNI (Java Native Interface), where time-consuming part of space charge routine is written in C and compiled into native code.

## 4.4 Input and Output File

JPP reads an input file in PARMELA-like format, where each line consists of one command keyword and several parameters following the command. Each command represents an accelerator element or simulation condition such as space charge model.

All the calculation results of JPP are recorded in a single output file, which contains the simulation parameters and particles coordinates at the exit of each accelerator element in binary format.

## 4.5 GUI/CUI Post Processor

A post processor POST5 is provided as a part of JPP to visualize simulation results and extract characteristic quantity such as beam emittance and envelope from an output file. The post processor POST5 is written in Java and all the GUI components are realized by *java.awt* (Abstract Window Toolkit), which is the standard API to provide graphical user interfaces in Java programs. Figure 2 shows a screen shot of POST5.

Character based interface is also available in POST5 with command line options, which is useful for quick checking of a specific quantity or running successive calculations in a batch job with logging a brief report of simulation results.

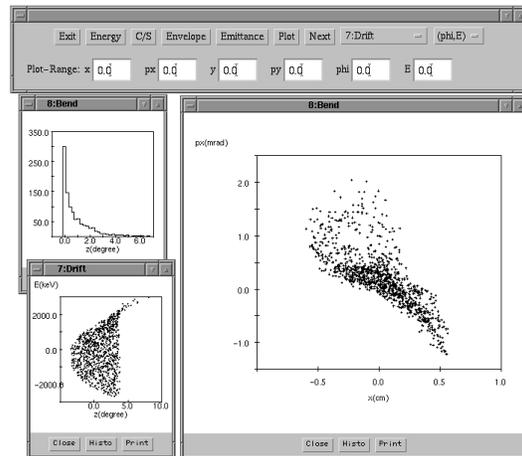


Figure 2: A screen shot of POST5 post processor.

## 5 EXAMPLES AND BENCHMARKS

Two examples of particle tracking have been made as benchmarks to check the validity of the simulation code and to evaluate the performance of Java code in comparison with C code.

### 5.1 Linear Emittance Growth in Simple Drift

Space charge calculation is one of essential routines in a charged particle tracking code. Calculation of linear emittance growth is a good example to check the accuracy of a space charge routine. When an electron bunch travels along simple drift space without external field, growth of transverse emittance occurs due to the self field of space charge. This emittance growth exists even if the bunch has uniform radial distribution, that is a hard edge cylinder, because the bunch has longitudinal discontinuity which causes nonuniform radial field depending on the longitudinal position in the bunch. In case that the electron bunch does not change its transverse size and the emittance growth mainly arises from transverse momentum fluctuation due to self field, the emittance grows linearly with drift length and can be calculated analytically. The linear growth of normalized trans-

verse emittance for an electron bunch having uniform transverse distribution is given by [4]

$$\Delta\varepsilon_n = \frac{I_p S}{4I_a \gamma^2 \beta^2} G(L/a) , \quad (1)$$

where  $I_p$  is peak current,  $I_a = 17kA$  is Alfvén current,  $S$  is drift length and  $G(L/a)$  is geometric function and given by  $G(L/a) = 0.556$  for long Gaussian beam.

The linear emittance growth has been calculated by JPP with ring-charge model and the following parameters:  $E = 5.11MeV$ ,  $I_p = 220A$ ,  $\sigma_z = 7.1cm$  and  $r = 1cm$ . The obtained emittance growth shows good agreement with analytical solution (fig. 3). Slight deviation of the numerical solutions from the analytical curve for long drift is considered as the result of radial expansion which introduces non-linear effect in emittance growth. The expansion of beam radius is about 2% after drift of 60cm in this calculation.

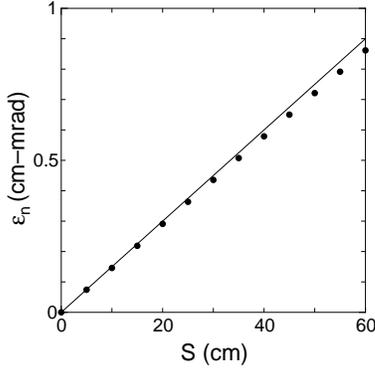


Figure 3: Calculation of linear emittance growth: analytical solution (solid line) and numerical results (dots).

## 5.2 Bunch Acceleration

Another example is bunch acceleration in RF cells. JPP reads RF field data prepared by SUPERFISH and calculates time varying electromagnetic field at arbitrary position during the cells by interpolating field value at 2-D cylindrical grid. In this example, short electron bunch,  $\sigma_z = 3.5mm$  and  $r = 2mm$  is accelerated by 1300MHz 9-cell cavity, where accelerating RF phase is chosen so that maximum energy gain is obtained. Figure 4 shows obtained transverse phase plot and energy spectrum after the cells.

## 5.3 Performance Benchmark

The performance of JPP is compared with another particle tracking code which has the same numerical procedures as JPP but written in C. We also investigate how much the performance of Java program can be improved with JIT and native method.

In the benchmark, we use a personal computer: Pentium-Pro 200MHz, 256kByte L2 cache, 128MByte memory, working on Solaris 2.5.1. Java code is developed and executed on JDK-1.1.6 and C code is compiled by “ProCompiler C 3.0.1” with full optimization.

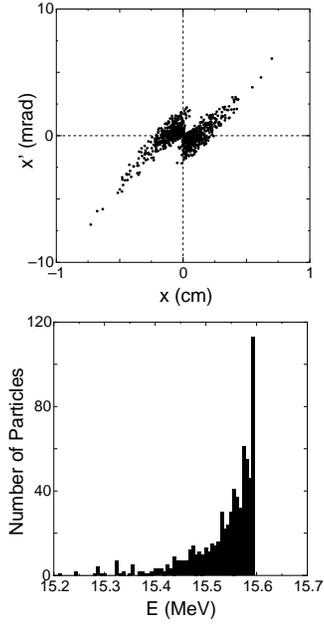


Figure 4: Results of bunch acceleration.

The performance of particle tracking code is measured by whole elapsed time for each run and three different input data are tested:

### (1) bunch acceleration with ring-charge model

This example contains two heavy numerical routines: the calculation of space charge and RF field. The parameters are chosen as : the number of particles  $N_p = 1000$ , the number of time steps  $N_s = 1754$ , the number of 2-D cylindrical grid mesh for the space charge routine  $(r, z) = (10, 30)$ .

### (2) simple drift with ring-charge model

In this case almost calculation time is spent by space charge routine. The parameters are  $N_p = 500$ ,  $N_s = 3000$ ,  $(r, z) = (10, 30)$ .

### (3) simple drift with point-charge model

This is similar to the second example, but space charge effect is calculated from repulsion force between all the pair of two particles. This point charge model is quite heavy calculation. The parameters are  $N_p = 500$ ,  $N_s = 400$ .

Table 1 shows the result of performance measurement, where  $T_a$  is whole elapsed time in second,  $T_s$  is time for space charge calculation,  $T_c$  is time for copying object between Java and C in native method and  $R$  is relative elapsed time in comparison with C.

The performance of Java with classical interpreter is really bad and the elapsed time is more than 16 times of C code in the worst case. It is also confirmed that JIT improves the performance of Java code greatly and the elapsed time is reduced to be two or three times of C code.

The performance of Java can be still improved, if time consuming routine is replaced by native compiled code. In our benchmark, space charge routine has been replaced by native method written in C and the performance has been surely improved as seen in table 1. Native method in JPP

Table 1: The result of performance measurement.

Bunch Acceleration (ring-charge)				
	$T_a$	$T_s$	$T_c$	$R$
Java (interpreter)	959	806	–	11.7
Java (JIT)	203	156	–	2.5
Java (JIT + native)	166	119	80	2.0
C	82	50	–	1.0
Simple Drift (ring-charge)				
	$T_a$	$T_s$	$T_c$	$R$
Java (interpreter)	1238	1204	–	16.7
Java (JIT)	217	205	–	2.9
Java (JIT + native)	98	86	42	1.3
C	74	70	–	1.0
Simple Drift (point-charge)				
	$T_a$	$T_s$	$T_c$	$R$
Java (interpreter)	601	594	–	8.7
Java (JIT)	157	153	–	2.3
Java (JIT + native)	61	58	6	0.88
C	69	68	–	1.0

is implemented by JNI (Java Native Interface) which Sun Microsystems provides as the standard implementation of native method.

Although we expected that space charge calculation in Java with native method is as fast as C code, Java does not catch up with C in some cases. This is because that native method introduces another overhead to copy object array between Java and C. In our calculation with native method, the array of particle object stored in Java working memory is transferred into memory for native method one by one before space charge routine starts and vice versa after space charge calculation is completed. This procedure may become a severe overhead, if we use large number of particles beyond the size of cache. Instead of copying object array one by one, native method could be implemented in other ways in which one can refer and rewrite Java working memory from native method or copy object array at once. However, JNI prohibits these operation and only permits to copy object array one by one, because it is the most secure method which never makes unexpected memory destruction. Performance of native method is, therefore, determined by overhead of copying memory and speed-up due to native code.

Native method contributes to the improvement of performance particularly in the third example of our benchmark, where space charge routine is quite heavy and spends much more time than the memory copy. It has also been confirmed in another study that the more complicated space charge calculation, noninertial space charge force in circular motion of electron bunch, can be available with JPP and native method [8]. In the case of bunch acceleration, on the contrary, native method saves calculation time little, because the number of particle is relatively large and RF field calculation is still conducted by Java byte code.

In conclusion of the benchmark, it is found that JIT sig-

nificantly improves the performance of the particle tracking code written in Java without losing the portability, the Java code, however, still requires calculation time two or three times as great as optimized C code in typical problems.

## 6 CONCLUSIONS

A charged particle tracking code has been developed with Java to investigate possible standardization of accelerator softwares with Java. It has been confirmed that the particle tracking code written in Java has several outstanding features: it is executable on a lot of different platforms without recompiling, providing unified graphical user interface, easy to maintain or extend due to sophisticated class hierarchy and robustness. The performance of the particle tracking code is inferior to optimized C code and it requires calculation time two or three times as large as C code, even if we use Java-VM with JIT (Just-In-Time compiler). The performance, however, will catch up with C code by upcoming HotSpot VM.

## 7 REFERENCES

- [1] J. Gosling and H. McGilton, "The Java Language Environment A White Paper", <<http://java.sun.com/docs/white/langenv/>>(1996).
- [2] D. Griswold, "A White Paper About Sun's Second Generation Java Virtual Machine", <<http://java.sun.com/products/hotspot/index.html>>(1998).
- [3] D. Griswold, "JDK Software: Performance Technologies", JavaOne '98 (1998).
- [4] M.E. Jones and B.E. Carlsten, in Proc. 1987 Particle Accelerator Conf., pp.1319-1321.
- [5] P. Lapostolle et al., Nucl. Inst. Meth. **A379**, pp.21-40 (1996).
- [6] K.T. McDonald, IEEE Trans. Electron Devices, **35**, pp.2052-2059 (1988).
- [7] B.E. Carlsten et al., Nucl. Instr. and Meth. **A304**, pp.587-592 (1991).
- [8] R. Hajima and E. Ikeno, "Numerical analysis of shielded coherent radiation and noninertial space-charge force with 3-D particle tracking", to be published in Nucl. Instr. and Meth. A.

# IMPLEMENTATION OF OBJECT-ORIENTED DESIGN WITH FORTRAN LANGUAGE IN BEAM DYNAMICS STUDIES

J. Qiang, R. Ryne, and S. Habib R. Ryne, LANL, Los Alamos, NM 87545 USA

## Abstract

In this paper, an object-oriented design for beam dynamics simulations in accelerators is implemented using Fortran language. Using module and derived type in F90, we can emulate object concept in the object-oriented design. This gives Fortran code a better maintainability, reusability, and extensibility. The overhead associated with the object-oriented implementation has only a minor effect on performance.

## 1 INTRODUCTION

Object-oriented design is being widely applied in computer software engineering to implement complex codes which possess good maintainability, reusability, and extensibility. This technique also enables the encapsulation of detailed machine specific information, thereby achieving good portability.

In the parallel computing environment, such efforts have mostly been directed to the design of object-oriented frameworks using explicit message passing and C++ [1]. Using such an object-oriented framework reduces the extent of difficulty of parallel programming based on message passing library and also allows good performance to be achieved. However, in the physics community, e.g., the accelerator community, Fortran still remains a popular language for demanding numerical simulations. Even here, implementation of object-oriented design can be useful since using F90 with Message Passing Interface (MPI) in this way encapsulates the detailed communication syntax and eases the design and implementation of parallel simulations.

In contrast to the message passing paradigm discussed above, High Performance Fortran (HPF) as a high level data parallel programming language also has its place in scientific computation. Its advantages of programming ease, reasonable performance, and portability between parallel and serial machines makes it attractive for use in many applications [2, 3, 4]. In HPF, inter-processor communication is handled by the compiler. The programmer generally only needs to explicitly specify the data distribution on parallel processors and parallel loops through directives comments [5]. This makes parallel programming more transparent and allows portability between parallel and serial machines.

Though not designed with object-orientation in view, Fortran 90 already contains some features of object-oriented programming languages with user defined generic data type, pointer, and modules [6, 7]. These features have been successfully applied in plasma simulations to build

object-oriented Fortran codes [8]. Since HPF is supposed to contain all the features of Fortran 90, it is also possible in theory to emulate object-oriented programming using the intrinsic module command and derived types in HPF. Thus, the application of object-oriented design with HPF can combine the traditional advantages of object-oriented methods along with the ease of parallel programming that characterizes HPF.

In this paper: The physical system is described in Section 2, object-oriented design is presented in Section 3, parallel implementation using F90 with MPI is given in Section 4, data parallel implementation with HPF is discussed in Section 5, numerical results on the SGI/Cray T3E-900 and SGI Origin2000 are presented in Section 6, and the conclusions summarized in Section 7.

## 2 PHYSICAL SYSTEM

The physical system for beam dynamics studies consists of the beam and the accelerating system which in turn contains a number of accelerating, guiding, and focusing elements. The forces acting on particles are due to externally applied fields and the inter-particle Coulomb field.

The two-dimensional application we will consider below is a study of the transverse dynamics of an infinitely long intense beam transporting across various focusing elements along the  $z$ -axis. We note that since accelerating, guiding, and focusing elements are arranged along  $z$ , it is usual practice in accelerator simulations to take  $z$  to be the independent variable rather than the time  $t$ . In this case, the original six dimensional equations reduce to a set of four dimensional  $z$ -dependent equations

$$\frac{\partial f}{\partial z} + x' \frac{\partial f}{\partial x} + y' \frac{\partial f}{\partial y} + v'_x \frac{\partial f}{\partial x'} + v'_y \frac{\partial f}{\partial y'} = 0 \quad (1)$$

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = \rho / \epsilon \quad (2)$$

Here,  $f$  is the particle distribution function in phase space,  $\phi$  is the space charge potential,  $\rho$  is the charge density from the distribution function, a prime superscript denotes a derivative with respect to  $z$ . The above equations is solved using a particle-based method. Particle simulations have much lower storage cost (in three spatial dimensions,  $N^3$  vs.  $N^6$ ) and have the crucial advantage of not breaking down when phase space structure falls below the grid resolution.

In the case of a linear focusing system, the Hamiltonian for single particle dynamics is

$$H = \frac{1}{2m} (p_x^2 + p_y^2) + H_{ext} + H_{self} \quad (3)$$



nications among processors when information from more than one processor is required. This gives it the advantages of flexibility and better performance. However, this also increases the difficulty of parallel programming. Applying object-oriented design to parallel message passing programming helps to encapsulate the details of communications and data distributions. This enables the user to manage the applications at a higher level.

To implement the object-oriented design with F90 and MPI, we add some new classes to the original auxiliary module to handle explicit communications. These classes are Pgrid2d, Communication, and Utility. The Pgrid2d class defines a logical two dimension Cartesian processor grid. The Communication class contains the public functions to handle the major communications used in the particle-in-cell simulation using MPI. The Utility class contains three public functions to encapsulate the explicit communications used in the general purpose function operations, e.g. matrix transpose. With the help of auxiliary class module, the particle simulation using beam, field, beam line element, and geometry modules can be built up without knowing the details of the communications. In the following, we give an example of implementing the F90 to emulate polymorphism in the beam line elements in our simulation. The polymorphism is done in an object-oriented language by defining a virtual base class and different derived classes. By assigning the address of a derived class object to a pointer object of base class, the procedure using a single base object name can select the appropriate member function to execute based on the actual class object referenced in the pointer object. In our beam dynamics simulation with F90, we define a base class BeamLineElem, and three derived classes for the drift, focusing, and defocusing beam line elements. The scaled down sketch of these classes are below:

```

module BeamLineElemclass
  use QuadrupoleFclass
  use QuadrupoleDclass
  use DriftTubeclass
  type BeamLineElem
    private
    type (QuadrupoleF), pointer :: pquadf
    type (QuadrupoleD), pointer :: pquadd
    type (DriftTube), pointer :: pdrift
  end type BeamLineElem
  interface assign_BeamLineElem
    module procedure assign_quadf,
      assign_quadd, assign_drift
  end interface
contains
function assign_quadf(pquadf) result(ppquadf)
function assign_quadd(pquadd) result(ppquadd)
function assign_drift(tdrift) result(ppdrift)
subroutine update_BeamLineElem(this,z0,z1)
end module BeamLineElemclass

module DriftTubeclass
  integer, private, parameter :: Nparam = 1
  type DriftTube
    integer :: Nseg
    real :: Length

```

```

    real, dimension(Nparam) :: Param
  end type DriftTube
contains
  subroutine update_DriftTube(this,z0,z1)
end module DriftTubeclass

```

Here, only the drift tube class is given for the derived class since the other two derived classes have a similar structure to the drift tube class. Since there is no direct support of inheritance in F90, we define a derived type in the BeamLineElem base class which contains three pointers to the derived classes as private data members. An overloaded function assign\_BeamLineElem which includes three assignment functions is used to initialize the base BeamLineElem class object with different derived class object addresses. In each assignment function, only one pointer is initialized and the other two pointers are set to null. In the public function update\_BeamLineElem of the base class, updating operations from derived classes are selected according to the different actual object association of pointers in the base class data member. The polymorphism is achieved by calling this subroutine with a constructed base BeamLineElem object in the application. The data members in the derived beam line element class consist of the number of steps of particle movement inside the beam line, length, and strength of the beam line element.

## 5 DATA PARALLEL IMPLEMENTATION WITH HPF

Data parallel programming using HPF provides a relatively easy route to parallel programming. Present compilers, however, are still not fully mature and performance penalties are often encountered.

In principle, using derived type with private data member in an HPF module containing some public member functions, it is possible to emulate a class in the same way as F90 does. Unfortunately, most present HPF compilers do not have adequate support for derived type and dynamically distributed arrays. For example, the PGHPF compiler does not support pointer to derived type, deferred array component in derived type, and parallel distribution of array component to processors in derived type [13]. These restrictions prevent one from defining a generic derived type with dynamic array component inside a module. Emulating object-oriented polymorphism is not possible for the same reason. Therefore, to implement the object-oriented design discussed in the Section 3, we have to modify some classes in our implementation from a generic type to a physical module which contains some private data members and public member functions. The public member functions contained in the physical module provide the behaviors of the module.

In the previous section, we showed that polymorphism could be used in the implementation of beam line elements. However, due to the absence of support for a pointer to derived type in the PGHPF compiler, we have to include the choice of different concrete beam line elements in the BeamlineElem module as separate subroutines. A scaled

down sketch of beam line element module is presented in the following.

```

module BeamLineElem_class
  integer, private, parameter :: Nparam = 1
  type BeamLineElem
    private
    integer :: Nseg
    real :: Length
    real, dimension(Nparam) :: Param
  end type BeamLineElem
contains
  subroutine update_BeamLineElem(this, flag,
                                z0, z1)
  subroutine beamlnDeQuad(this, z0, z1)
  subroutine beamlnFoQuad(this, z0, z1)
  subroutine beamlndefault(this, z0, z1)
end module BeamLineElem_class

```

Parallel loop implementation uses the HPF commands *Do Independent* and *Forall*. *Forall* is used in the case of a single statement with regular array index access. *Do Independent* is used to fuse several statements into one loop to take advantage of the spatial and temporal locality of data in cache. (In the data scattering from grid to particle subroutine, we observed that using *Do Independent* is about a factor of ten faster than using *Forall* in the indirect array index access loop.)

## 6 NUMERICAL RESULTS

The above object-based Fortran programs were applied to the study of proton beam transport through a periodic constant focusing, drift, defocusing, drift (FODO) channel. Here, we report our experience running on the SGI/Cray T3E-900 and SGI Origin2000.

The problem we tested here consists of 10 FODO periods. Our simulation used 250,000 particles with a  $128 \times 128$  field mesh grids. Fig 2 gives the performance of using the object-based F90 with MPI on the SGI/Cray T3E-900.

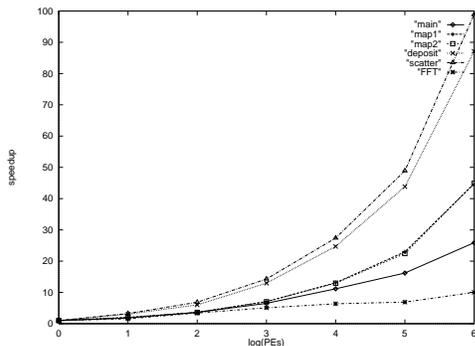


Figure 2: The speedup of the object-based F90/MPI code as a function of the number PEs on SGI/Cray T3E-900.

We measured the speedup of 5 major subroutines scaled by the number of processors. These are map1, map2, scattering, depositing and FFT subroutines. The speedup is calculated as the ratio of time measured on one processor to

the time measured on a given number of processors. The main program speedup reaches about 26 on 64 processors. The total efficiency (ratio of speedup to number of processors) here is relatively low due to the heavy communication costs and small problem size. Checking the performance of separate subroutines, we see that the FFT subroutine is the least scalable due to the global nature of the Fourier transform resulting in communication overhead. On the other hand, depositing and scattering subroutines show a superlinear trend which give a speedup close to 100 on 64 processors. This is because, in the F90/MPI code, since both subroutines are done locally on each processor, the resulting more efficient use of cache may provide superlinear speedup in the operation of these subroutines. The performance of the corresponding HPF code is given in Fig 3.

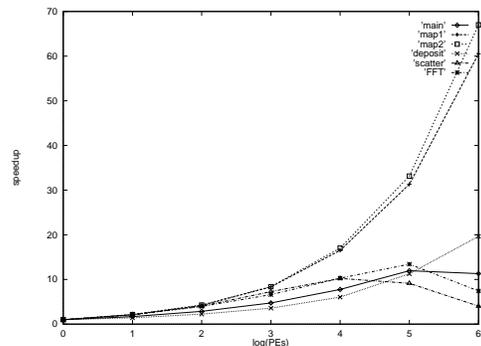


Figure 3: The speedup of the object-based HPF code as a function of the number PEs on SGI/Cray T3E-900.

We see that the main program speedup increases to 32 processors and saturates beyond that. Checking the speedup on individual subroutines, we find that the grid-particle scatter subroutine reaches its maximum speedup on 16 processors due to heavy communication in the indirect array index access. This becomes a bottleneck for the case of a large number of processors. The FFT subroutine reaches its maximum speedup on 32 processors. The map1 and map2 subroutines scale very well with increasing number of processors. The depositing subroutine does not scale well due to the communications in particle deposition. Fig 4 gives the speedup performance of the same code on the SGI Origin2000.

The main program speedup increases to 32 processors and starts to decrease on 64 processors. The speedups of individual subroutines map2, and the depositing and scattering subroutines are superlinear which might be due to local cache effects. The FFT subroutine gives very poor scalability saturating even on 16 processors due to its global nature. This makes the main program less scalable on the SGI Origin2000 than on the SGI/Cray T3E-900. To implement the object-oriented design, we used a number of pointers and dynamics allocated arrays. This will affect the code performance due to loss of compiler optimization comparing with non-object-oriented code. In Fig 5, we give a comparison of the time costs of object based codes using both F90/MPI and HPF with conventional procedure based codes.

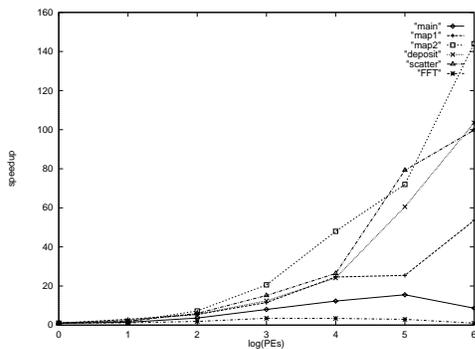


Figure 4: The speedup of the object-based F90/MPI code as a function of the number PEs on SGI Origin2000.

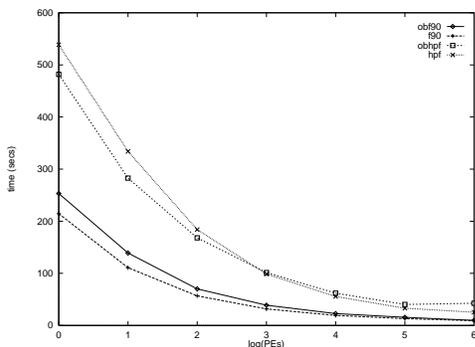


Figure 5: The time costs of object-based and procedure-based programs as a function of the number PEs on SGI/Cray T3E-900.

For F90 with MPI code, on small number of processors, the overhead from object-oriented implementation is about 10 to 20%. This overhead decreases with increasing number of processors. In the case of HPF, the object based code seems to over perform the procedure based code on small number of processors and lose out on a large number of processors.

## 7 CONCLUSIONS

In this paper we have discussed implementations of object-oriented design using Fortran in parallel beam dynamics simulations. As previously stated, implementing the object-oriented design with F90 and MPI encapsulates the details of communication in low level auxiliary classes. This also provides the benefits of better maintainability, reusability and extensibility of software. For example, a new beam line element can be easily incorporated into the BeamLineElem class without affecting the other classes. Using the concept of object, which is implemented using HPF modules, gives the code some advantages of object-oriented programs and also the advantage of ease in parallel programming. This implementation was based on the current status of the PGHPF compiler technology. With further development of compilers, it is possible that the programming barriers we encountered will disappear. In that event, this model will be easily extended to adopt new features in

the future to include more completely object-oriented features.

In our first experience of applying these codes on the SGI/Cray T3E-900 and SGI Origin2000, we obtained a reasonable performance up to 32 processors on both machines. The code written using F90 with MPI seems to be more scalable on the SGI/Cray T3E-900 than on the SGI Origin2000. The overhead of implementing object-oriented design using pointers, types and modules is small. To summarize, it appears to us that implementing object-oriented design with Fortran can achieve both good software quality and parallel programming in scientific applications.

## 8 ACKNOWLEDGMENTS

We acknowledge helpful and stimulating discussions with Dr. Viktor Decyk and the POOMA team, who provided POOMA beam dynamics source code. This work was performed on the SGI/Cray T3E at NERSC and SGI Origin2000 at the ACL and supported by the DOE Grand Challenge in Computational Accelerator Physics.

## 9 REFERENCES

- [1] G. Wilson, L. Paul (ed.): Parallel Programming Using C++, MIT Press, Cambridge (1996).
- [2] R. Ryne, S. Habib, Parallel Beam Dynamics Calculations on High Performance Computers, In: J. J. Bisognano, A. A. Mondelli (eds.): Computational Accelerator Physics, AIP Conference Proceedings 391, Woodbury, New York (1997) 377-389.
- [3] C. H. Ding: HPF for Practical Scientific Algorithms, Preprint for Supercomputing '97, (1997).
- [4] V. V. Elisseev, Parallelization of Three-dimensional Spectral Laser-Plasma Interaction Code Using High Performance Fortran, Computers in Physics **12** (1998) 173-180.
- [5] C. H. Koelbel, B. D. Loveman, R. S. Schreiber, G. L. Steele, M. E. Zosel : The High Performance Fortran Handbook, MIT press, Cambridge, (1994).
- [6] T. M. Ellis, I. R. Philips, T. M. Lahey: Fortran 90 Programming, Addison-Wesley, Harlow, England (1994).
- [7] M. G. Gray, R. M. Roberts: Object-based Programming in Fortran90, Computers in Physics **11** (1997) 355-361.
- [8] V. K. Decyk, C. D. Norton, B. K. Szymanski : Expressing Object-Oriented Concepts in Fortran 90, ACM Fortran Forum, Vol. 16, num 1, (1997).
- [9] E. Forest and R. D. Ruth: Fourth-Order Symplectic Integration, Physica D **43** (1990) 105-117.
- [10] R. W. Hockney and J. W. Eastwood: Computer Simulation Using Particles, Adam Hilger, New York, (1988).
- [11] G. Booch: Object-Oriented Analysis and Design with Applications, Benjamin/Cummings, Menlo Park, CA, (1994).
- [12] <http://www-c.mcs.anl.gov/Projects/mpi/> (1998).
- [13] PGHPF manual, <http://www.nerisc.gov/software/prgenv/compilers/pghpf/docs/>, (1998).

# AN APPLICATION FRAMEWORK AND INTELLIGENT GRAPHIC USER INTERFACE FOR MULTIPLE ACCELERATOR CODES

Barrey W. Hill, Hendy Martono, John M. Moore and James S. Gillespie\*  
G. H. Gillespie Associates, Inc., P. O. Box 2961, Del Mar, CA 92014, USA

\* Permanent address: Intuit, Inc., 6265 Greenwich Drive, San Diego, CA 92121, USA.

## *Abstract*

A multi-platform application framework is being developed for the implementation of a variety of particle physics simulation and analysis codes under a single graphic user interface (GUI) shell. The framework architecture supports plug-in tools such as an interactive particle trajectory module, external data interface tools and hypertext tutorials which interface with installed physics codes. A variety of physics programs are integrated into the framework by separating the computational code, which is implemented as a platform-independent Computational Module, from the I/O requirements, which are replaced by an Interface Module developed with the multi-platform GUI framework. The object-oriented framework provides a sophisticated beamline object model and a rich library of GUI components. Application-dependent components can be derived from the more abstract framework components to support specific requirements of different simulation codes. An overview of the Multi-Platform Shell for Particle Accelerator Related Codes (S.P.A.R.C.-MP) application framework is presented here with illustrations from a Windows 95/NT implementation.

## 1 INTRODUCTION

The Multi-Platform Shell for Particle Accelerator Related Codes (S.P.A.R.C.-MP) is an object-oriented framework which provides the underlying infrastructure for the development of software applications focused on improving the accessibility and ease of use for a variety of particle optics programs used in the accelerator community. Particle physics simulation and analysis programs are made accessible on multiple platforms as Computational Modules in an intelligent graphic user interface shell developed with the S.P.A.R.C.-MP framework. The framework provides a sophisticated beamline object model and an extensive collection of GUI components in order to support a variety of computational codes within a single user interface shell. This provides an environment for the graphical setup of beamline models and execution of multiple optics programs from a single user-friendly interface, without requiring any specific knowledge of the various input formats and syntax required by the different computational programs.

## 2 GRAPHIC USER INTERFACE SHELL

The S.P.A.R.C.-MP framework provides a sophisticated beamline object model and a rich library of GUI components for the development of a unique user interface shell for particle accelerator related codes. Figure 1 illustrates the main document window for the Particle Beam Optics Laboratory (PBO Lab™) [1,2,3,4] which was developed with the S.P.A.R.C.-MP framework.

Beamlines are graphically constructed in the user interface by selecting and dragging beamline Piece icons from the Palette Bar and dropping them onto the Model Space. Individual Pieces or groups of Pieces may be selected for use in other beamline construction tasks, such as dragging a copy to the Work Space, converting between hierarchical and flat representations, creating Aliases or copying between different Documents. Selections from the Work Space may be inserted into the beamline by dragging the selection to an insertion point between beamline Pieces or to the end of the beamline model. Selections of Pieces may also be grouped as Sublines, encapsulating a series of accelerator elements into a single Piece in the beamline model. Sublines may contain nested Sublines, as well as individual Pieces. Sublines can be flattened to expand encapsulated Pieces back to individual icons in the beamline. However, individual Pieces within a Subline can be accessed without flattening the Subline.

Input parameters for beamline elements are accessed by double-clicking Piece icons in the Work Space or Model Space of the Document window. Figure 1 illustrates Piece Windows for the Initial Beam and Quadrupole element. Many beamline elements provide different parameter sets that can be used to specify inputs for the element. For example, a quadrupole may be minimally defined by length and quad coefficient value, or by length, aperture and pole tip magnetic field, or by length and magnetic field gradient. Automatic calculations are made between the different parameters sets, allowing the user to select the parameters for specifying an element independent of the native inputs used by specific simulations. The user interface indicates which parameters are used as native input for the

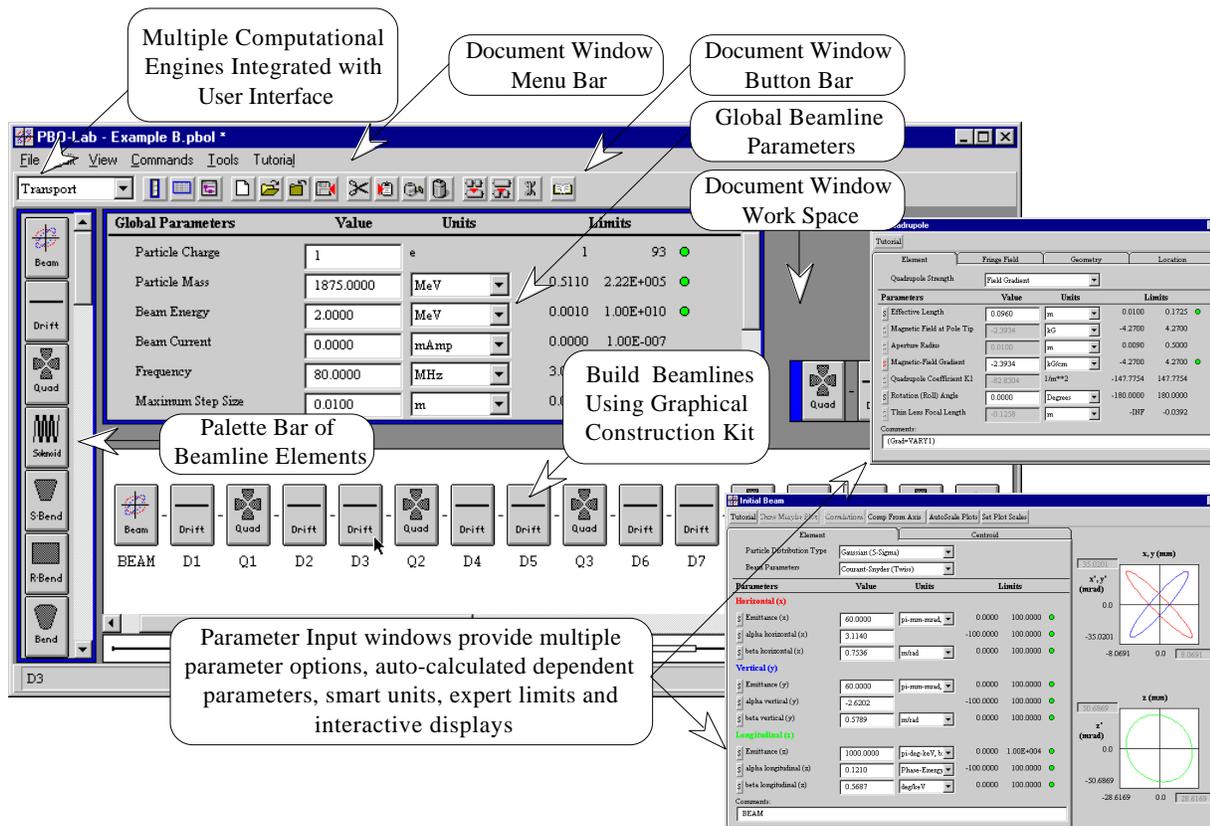


Figure 1. S.P.A.R.C.-MP based PBO Lab for Windows.

selected simulation. A variety of different units options may be selected for each parameter independent of the units required by the different simulations. Expert system type rules provide guidance for editing input parameters, and additional displays, such as effective focal lengths and phase space plots, provide useful feedback to the user.

The framework also provides data visualization components which are used to generate a variety of ellipse plots, scatter plots, line plots, bar graphs, etc. from the raw data generated by the different simulation applications. In addition, the framework architecture supports plug-in tools such as an interactive particle trajectory module, optimization algorithms, an external data interface and hypertext tutorials which interface with and enhance the functionality of installed simulations.

Central to the infrastructure of the S.P.A.R.C.-MP framework is the persistent beamline object model [5]. Fundamentally, the object model is the basis for describing accelerator beamlines at different levels of abstraction and providing the foundation for implementing a host of optics codes that use a variety of different beamline model descriptions. The beamline object model forms the basis for the interactive graphic functionality and the persistent beamline model description. It is also used to generate the native input required by the individual

simulations supported in the framework. The beamline object model describes hierarchical, flat and mixed beamline representations. This model is especially suited to representing beamlines with a large number of repetitive components using Sublines and Aliases. Aliases can be created for any Piece or Subline in the beamline model. Aliases are implemented in the beamline object model with a persistent link to another Piece or Subline and are capable of storing deviations from the original Piece parameters without any duplication of redundant data. A hierarchical organization of the beamline, using Sublines in combination with the ability to create Aliases, allows the user to configure the beamline in any representation that suits their problem. A number of benefits can be realized with this model including more efficient problem setup, compact views of very large models and elimination of redundant data storage within highly symmetrical beamlines.

### 3 SIMULATION APPLICATION MODULES

Different optics programs are implemented as Application Modules in the S.P.A.R.C.-MP framework. A simulation code is selected with a pop-up menu in the Document window button bar as illustrated in Figure 1. This selection is referred to as the application context. The application context is used to provide the user with

various application-specific feedback such as the native inputs for an application. An Application Module has two primary components: an Interface Module and a Computational Module. The Interface Module is the application-specific graphic user interface and data interface code for implementing the input and output requirements of the application. The Computational Module is the computational physics code from the application (with the I/O functionality replaced by the Interface Module). The Computation Module is implemented as a dynamic link library (DLL) that is loaded at run time. Legacy FORTRAN codes, as well as the latest object-oriented codes, can be incorporated into the framework as independent simulation Application Modules. The core graphic user interface components and beamline object model are independent of the installed Simulations. However, application-dependent GUI components can be derived from the more abstract framework components to support any application-specific GUI requirements for the different simulations.

The user interface and data interface requirements for an Application Module are also implemented as independent application-specific components that leverage the functionality of the more abstract framework components. The Interface Modules generate different application-specific beamline descriptions (native input) for executing the Computational Modules, which are treated as black-box computational engines. The framework will support a variety of particle physics simulation and analysis codes such as TRANSPORT [6], TURTLE [7], TRACE 3-D [8] and MARYLIE [9]. Selected characteristics and uses for these applications are listed in Table 1. Additional particle optics applications, as well as system level and facility modeling codes, are also under development for integration with the S.P.A.R.C.-MP framework.

Table 1. Selected Simulation Application Codes.

Code	Selected Characteristics
TRANSPORT	Third-Order Matrix Code. Magnetic Optics, Fitting, Optimization, Aberration Control.
TURTLE	Third-Order Ray Tracing Code. Multi-Particle Simulator. Beam Aperture Control.
TRACE 3-D	First-Order Matrix and Space Charge. Magnetic, RF and Electrostatic Optics. Fitting, Optimization, Envelopes.
MARYLIE	Lie Algebraic Beam Transport. Linear and Non-Linear Optics. Multi-Turn Tracking, Fitting, Optimization.

## 4 FRAMEWORK ARCHITECTURE

The architecture that has been developed for the S.P.A.R.C.-MP framework allows modular software components to be added to, and removed from, the underlying user interface shell. This provides great flexibility for incorporating multiple optics programs as simulation Application Modules in the S.P.A.R.C.-MP framework. The development effort is greatly reduced by reusing verified software components and constructing new components based on more abstract software modules. The S.P.A.R.C.-MP framework provides an abstraction layer with an extensive collection of tested and verified components developed specifically for implementing accelerator design and analysis programs. Figure 2 illustrates the modular architecture that has been developed to support multiple Application Modules in the framework. The Interface Modules and Computational Modules are pictured in the upper right of the Figure, enclosed by a gray outline. The architecture provides the capability to determine what Application Modules are installed and to tailor the user interface for each particular application environment. Simulation Application Modules are registered with the shell so that the user interface can add application-specific menu items, tool bar buttons, tab panels, etc. where and when appropriate based on which simulations have been detected. Framework components and data interface objects can be dynamically modified to support the native requirements and capabilities of specific applications. Framework components that provide GUI features are shared between the different Application Modules. These include the main document window and beamline construction toolkit, as well as the underlying beamline object model, the palette bar of accelerator elements, piece windows with multiple parameter sets, smart units, expert limits and auto-calculated user feedback of dependent parameters. The framework also provides data visualization components that are used to plot data generated from the different simulations. Additional application-specific GUI and data interface requirements are implemented in the Interface Modules.

The modular architecture reduces complexity through abstraction and lends itself to the multiple platform approach adopted for the S.P.A.R.C.-MP framework. Figure 2 illustrates the multi-platform and multi-application aspects of the framework architecture as OS dependent and OS independent layers. GUI code is OS (operating system) dependent. OS independent code is ANSI compatible and independent of the platform and operating system because it contains no references to the native OS libraries. The OS dependent layer is subdivided further into application specific and application independent layers. The multi-platform capability of the S.P.A.R.C.-MP framework is derived from the use of different libraries, each of which encapsulates a specific

operating system. These libraries use a common application programming interface (API) for multiple operating systems. The libraries, which are a collection of basic functions utilizing OS native routines, provide a means of interacting with the platform-specific Operating

System, while maintaining the native “look and feel” of the graphic interfaces associated with the target platform. With the use of these libraries, applications built with the S.P.A.R.C.-MP framework are made platform-independent, since the framework code calls the API

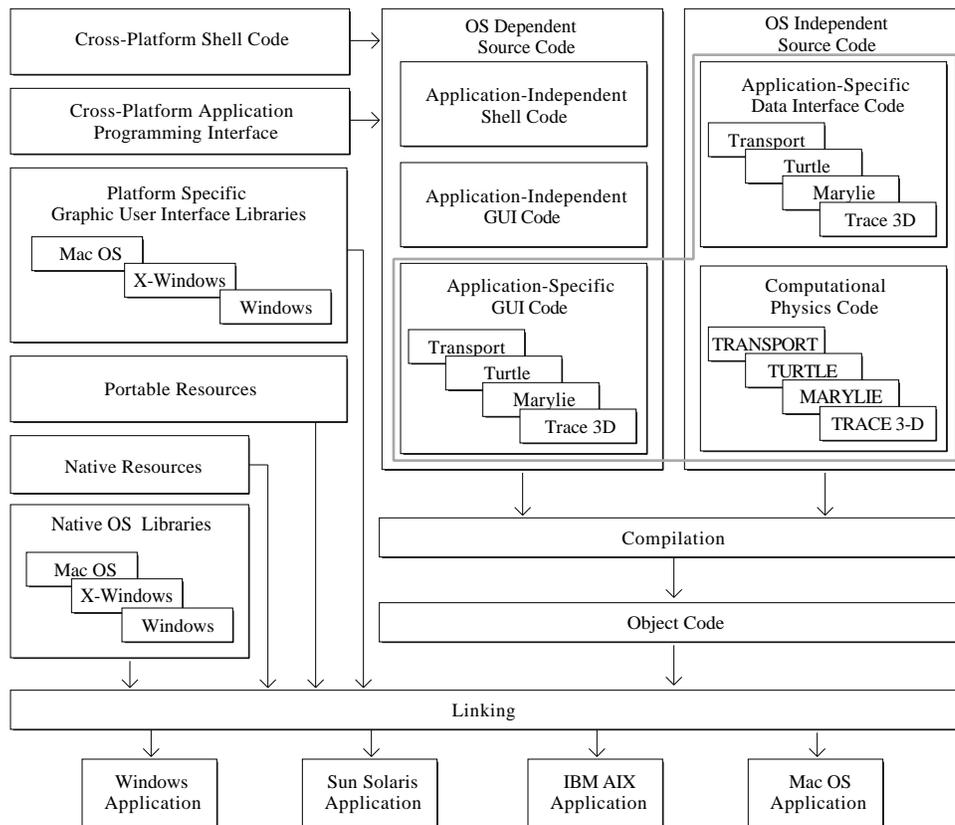


Figure 2. S.P.A.R.C.-MP Multi-Platform Framework Architecture.

routines, not platform specific routines, directly. This approach provides an abstraction layer from the different operating systems and yields a single version of the application source code for all platforms as opposed to developing and maintaining separate source codes for each platform. A platform-specific executable application is created by linking the application code with the appropriate library for the intended platform. Using this approach, the software may be easily ported to a new operating system by utilizing an interface library for that system. The API libraries are implemented as a layer on top of the native toolkits for each of the target platforms. This ensures a native look-and-feel, low overhead and still provides direct access to native toolkits if required. This multi-platform approach has been successfully implemented on personal computers running Microsoft Windows™ 95, 98 and NT, Apple’s Mac OS™ for the PowerPC, as well as IBM and Sun work stations running AIX<sup>a</sup> or Solaris™.

## 5 SUMMARY

The S.P.A.R.C.-MP framework is a multi-platform object-oriented application development environment specifically designed for creating applications which integrate a variety of particle optics and accelerator design codes under a single graphic user interface shell. The framework provides a sophisticated beamline object model and an extensive collection of GUI components, supporting a variety of particle physics simulation and analysis codes such as TRANSPORT, TURTLE, TRACE 3-D and MARYLIE. The Operating Systems supported by the S.P.A.R.C.-MP application framework and graphic user interface include Windows, Mac OS, Solaris and AIX. Platform-specific GUI components are implemented for multiple platforms with a single source platform-independent project. The framework multi-platform GUI components are developed by utilizing a single application programming interface (API) instead of the native software development kits (SDK) for each different platform. The API is implemented for a variety

of computer platforms as platform-specific libraries that interface with the native SDK for each platform. An executable application is built from the single source project by compiling and linking the code on the target platform with the appropriate API. Physics applications are integrated into the framework by separating the computational physics code, which is implemented as a platform independent Computational Module, from the I/O requirements, which are replaced with an Application Module developed with the multi-platform GUI framework components. The S.P.A.R.C.-MP framework is significant not only for its multi-platform capabilities but also for its open and expandable development environment. This environment supports the implementation of multiple optics codes under a single graphic user interface, improving the accessibility and ease of use for a variety of programs used in the accelerator community.

## 6 ACKNOWLEDGEMENTS

Portions of this work has been supported by the U. S. Department of Energy under SBIR grant number DE-FG03-95ER81975.

## 7 REFERENCES

- [1] G. H. Gillespie, B. W. Hill, N. A. Brown, H. Martono and D. C. Carey, "The Particle Beam Optics Interactive Computer Laboratory," AIP Conference Proceedings 391, 264-269 (1996).
- [2] G. H. Gillespie, B. W. Hill, H. Martono, J. M. Moore, N. A. Brown, M. C. Lampel and R. C. Babcock, "The Particle Beam Optics Interactive Computer Laboratory for Personal Computers and Workstations," to be published in the proceedings of the 1997 Particle Accelerator Conference, 3 pages.
- [3] N. A. Brown, G. H. Gillespie, B. W. Hill, M. C. Lampel, H. Martono, and J. M. Moore, "The Particle Beam Optics Laboratory (PBO Lab™): A New Education and Training Aid," to be published in the proceedings of the Sixth European Particle Accelerator Conference, 3 pages.
- [4] The PBO-Lab software is available from AccelSoft Inc., San Diego, California, U.S.A., [www.ghga.com/accelsoft](http://www.ghga.com/accelsoft).
- [5] B. W. Hill, H. Martono and J. S. Gillespie, "An Object Model for Beamline Descriptions," AIP Conference Proceedings 391, 361-365 (1996).
- [6] D. C. Carey, K. L. Brown and F. Rothacker, "Third-Order TRANSPORT - A Computer Program for Designing Charged Particle Beam Transport Systems," Stanford Linear Accelerator Center Report No. SLAC-R-95-462, 295 pages (1995).
- [7] D. C. Carey, "TURTLE (Trace Unlimited Rays Through Lumped Elements) A Computer Program for Simulating Charged Particle Beam Transport Systems," Fermi National Accelerator Laboratory Report No. NAL-64, 45 pages (1978).
- [8] K. Crandall and D. Rusthoi, "TRACE 3-D Documentation," third edition, Los Alamos National Laboratory Rep. LA-UR-97-886, 106 pages (1997).
- [9] A. J. Dragt, D. R. Douglas, J. van Zeijts, F. Neri, C. T. Mottershead, R. D. Ryne, E. Forest, L. M. Healy and P. Schutt, "MARYLIE 3.0 User's Manual: A Program for Charged Particle Beam Transport Based on Lie Algebra Methods," Draft, 543 pages (1998).

# MEASUREMENTS AND VISUALIZATION OF THE TRANSVERSE PHASE-SPACE TOPOLOGY AT LEP

G. Morpurgo, CERN

## Abstract

The LEP Beam Orbit Measurement system (BOM) allows for the acquisition of the beam position at each Beam Position Monitor (BPM) for over 1000 consecutive turns. By synchronizing the acquisition with a kick given to the beam, we can investigate the behavior of the beam under different conditions. In particular, starting from the data of one BPM, we can apply a simple mathematical manipulation to build a "virtual" BPM, with a phase advance of 90 degrees. Plotting the real BPM against the virtual one, we can observe the evolution of the beam in the phase-space. An appropriate coloring technique is used, to help the User finding his way through the data. Fixed points in the phase-space can be put in evidence, as well as the beam behavior in their neighbourhood. Quantities like the tune, the beam detuning as function of the position amplitude and the beam damping can be studied in this way. Significant examples from real life will be shown.

## 1 INTRODUCTION

The LEP Beam Orbit Measurement system (BOM) is made of 500 Beam Position Monitors (BPMs). Each BPM can record the position of each particle bunch in LEP for more than 1024 consecutive turns. Some of these BPMs also record a signal roughly proportional to the bunch currents. The data acquisition is synchronized all around LEP, and it can be triggered by timing events (for instance, by the injection kick). Via software, the information relative to a specific bunch can be extracted. Fig. 1 shows a typical data set, plotting the evolution of the horizontal position of the selected bunch at a given BPM. In this case, the non integer part of the horizontal tune of the beam before the kick was just below  $1/3$ . The large oscillations induced by the kick, combined with the detuning with amplitude effect, initially brought the beam (or part of it) to the third order resonance. Eventually the oscillations were damped, and the beam moved back towards the closed orbit. Our goal is to use this kind of data to get a better understanding of phenomena happening in the transverse phase space[1]. We will explain the method we developed, and we will show a variety of different applications using real data sets from LEP.

## 2 PHASE SPACE REPRESENTATION USING A VIRTUAL PICKUP

In order to visualize the behavior of the beam in the phase space in one plane, we have to plot the position of the beam against its derivative. The kind of phenomena we are interested in (slowly damped betatron oscillations) are

well described in terms of sinus and cosinus functions, and therefore we can approximate the differentiation operator by a phase shift of 90 degrees. While traditionally this was done by selecting two BPMs separated by a phase advance of (approximately) 90 degrees, we found that the most practical way of implementing this phase shift is the creation of a Virtual BPM. This is done by manipulating the position data coming from one single BPM, to reconstruct a representation of their derivative[1](see also [2]).

### 2.1 How to create a Virtual BPM

1. Our starting point is the data array containing the beam position at a given BPM for 1024 turns (Fig.1).

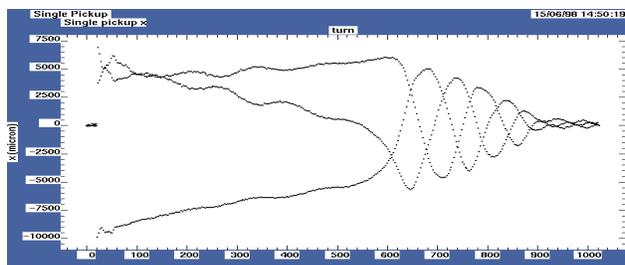


Figure 1: X position for 1024 turns. In this case the beam was kicked close to the third order resonance.

2. We perform an FFT on this position array, to get a frequency spectrum (preserving the sinus and cosinus components of each frequency).
3. We then rotate in the complex plane every spectrum component by 90 degrees. We also apply a linearly decreasing weight to the spectrum tails, to reduce errors (Fig.2).

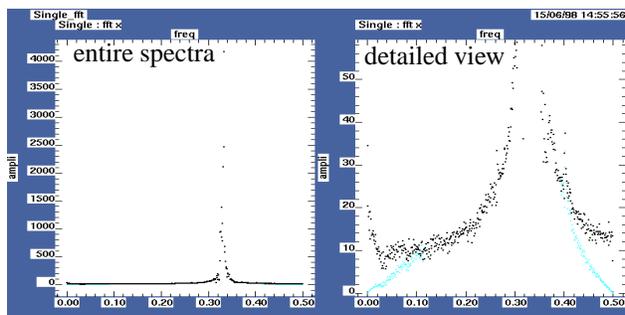


Figure 2: The FFTs of the Real (black) and Virtual (cyan) BPM. In parallel to the rotation operation (amplitude preserving) we apply a linearly decreasing weight to the spectrum tails.

4. Now we perform a Reverse FFT on the rotated spectrum, to get the position array of the Virtual BPM (Fig.3).
5. We plot the Real BPM against the Virtual one (Fig.4). We can also recenter the two arrays around the origin of the phase space, and renormalize on  $\beta$ .

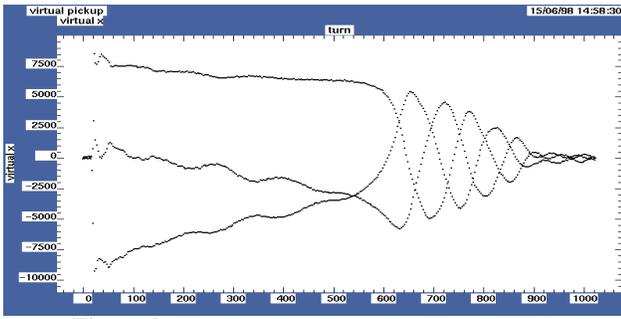


Figure 3: The position array for the Virtual BPM.

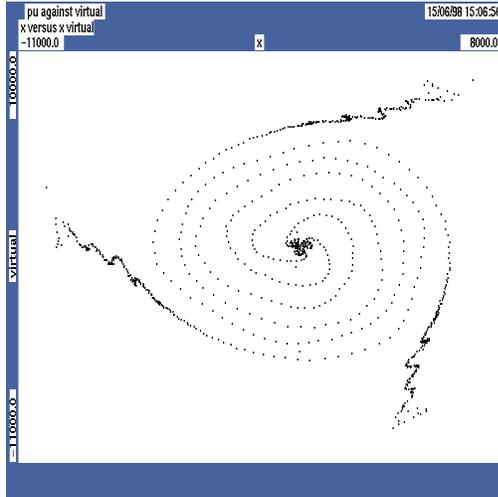


Figure 4: The phase space plot (not yet normalized).

6. We can then add fancy colours depending on the tune value, to emphasize phenomena to be visualized (Fig. 5). In our example, we will use the same colour for every third point, as the dominant phenomena is the oscillation close to the third order resonance. In some cases, connecting points with the same colour may help in understanding the dynamic behavior.

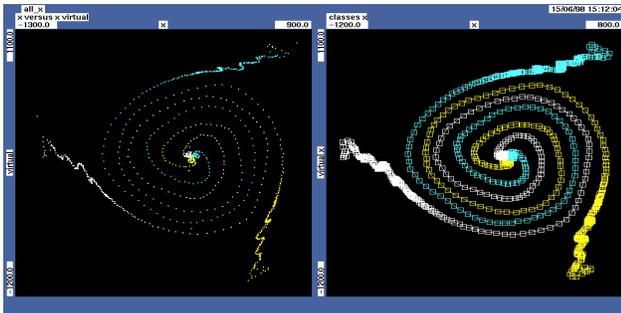


Figure 5: a) Every third point is given the same colour. b) Points with the same colour are connected.

## 2.2 The Stroboscopic Effect

The visualization of the tune evolution within our samples is also helped by what we call the "Stroboscopic Effect". It consists in the fact that every time the tune is close to a fraction with a small denominator, nicely identifiable arms appear in the phase space plot. In Fig. 6 we can identify an outer region, where the tune is close to  $2/7$  (0.286), and where we give every seventh point the same color, and an

inner region, with tune close to  $3/11$  (0.273) where we give every 11th point the same colour.

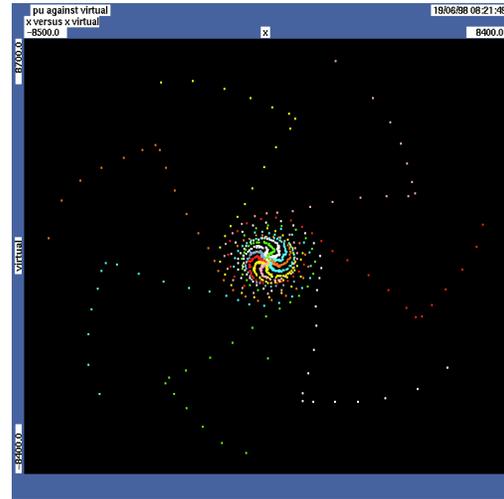


Figure 6: Stroboscopic effect. Tune =  $2/7$  (0.286) outside,  $3/11$  (0.273) inside.

## 2.3 How accurate is the Virtual BPM ?

Due to the discreteness and finiteness of the original sample, the process of generating the Virtual BPM cannot give an exact result. But from the result of some simulations one can see that, apart from a few bad points at the beginning and at the end of the data set, almost all the data points of the Virtual BPM array are correct within 1% (Fig. 7).

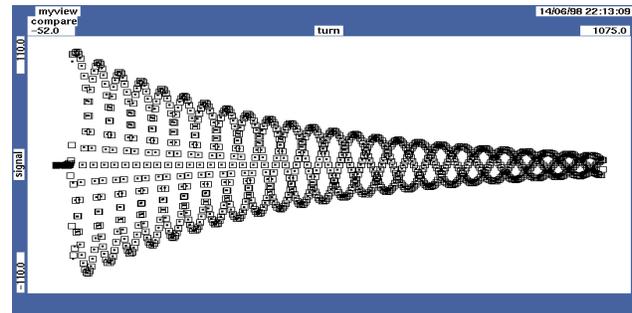


Figure 7: Exponential Decay after kick. Dots = expected BPM, Boxes = result of Virtual BPM Creation algorithm.

## 3 APPLICATIONS FROM LEP

### 3.1 Third Order Resonances

Our first example consists in the observation of the beam approach to the third order resonance fixed points. The beam, originally at a tune below  $1/3$ , was strongly kicked in the horizontal plane, and, because of the tune growth with amplitude, it was trapped into the 3rd order resonance, and partially lost. In the following figures, every third point is given the same colour. In Fig. 8 the data arrays for the Real and the Virtual BPM are shown. In Fig. 9 the two arrays are plotted against each other, showing how the original oscillations around a tune of  $1/3$  are slowly damped and three

fixed points are approached. Fig. 10 represents an enlargement on one of the three areas of the previous picture.

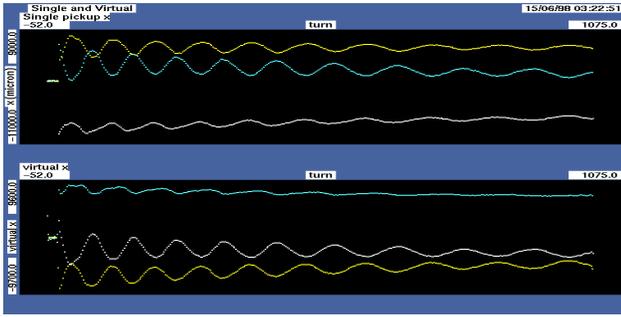


Figure 8: Real and Virtual BPM.

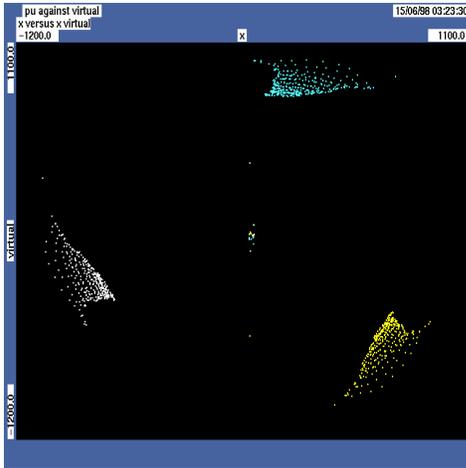


Figure 9: Phase space plot.

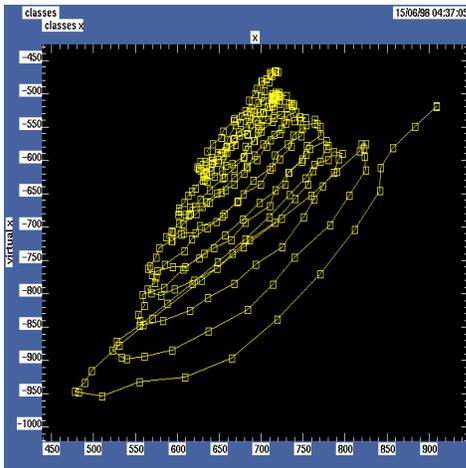


Figure 10: A detail on one of the three regions, as the beam moves toward the fixed point.

### 3.2 Detuning With Amplitude

The angle between consecutive turns in the phase space plot represents the tune at that turn. By plotting it against the Courant-Snyder invariant (kind of normalized amplitude) one can show how the tune changes as function of the

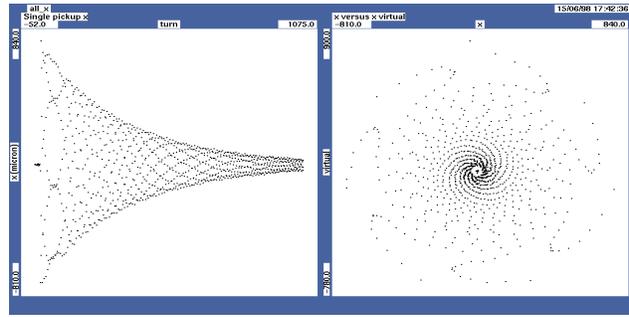


Figure 11: Position data and phase space plot for a good dataset.

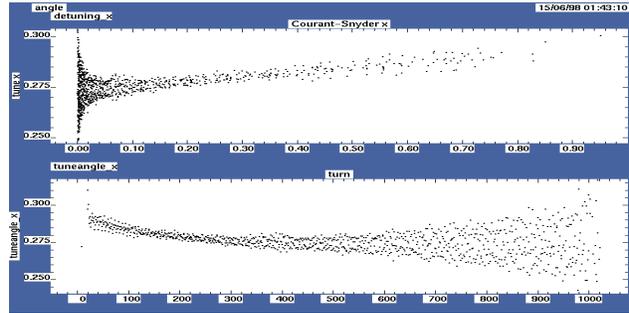


Figure 12: Detuning with amplitude (top).

beam oscillations amplitude. One must anyway be careful to stay away from resonances, which can modify the behavior of the beam and make the measurement meaningless. In Fig. 11 we show a data set suitable for this measurement, with smooth and regular damping of the oscillations after a large kick. Once the phase space plot has been centered and normalized, the Courant-Snyder invariant at every turn is just the square of the distance from the origin of the point corresponding to that turn. In Fig. 12 we plot the detuning with amplitude for this dataset, and we can see that it follows a linear behavior over the entire data range<sup>1</sup>. On the

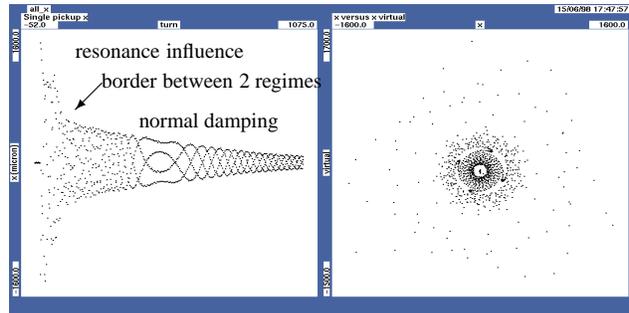


Figure 13: Position data and phase space plot for a bad dataset.

contrary, in Fig. 13 and 14 we can see what happens when we are too close to a resonance. By looking at the BPM data one can already notice that the behavior in the first 150 turns after the kick is quite different from the behavior later. This eventually reflects in the detuning with amplitude plot, where we can identify two different regions : the first, on the right, corresponds to the first turns after the kick, and it shows a slow detuning with amplitude, mainly

<sup>1</sup>Of course the errors become large as the amplitude gets too small.

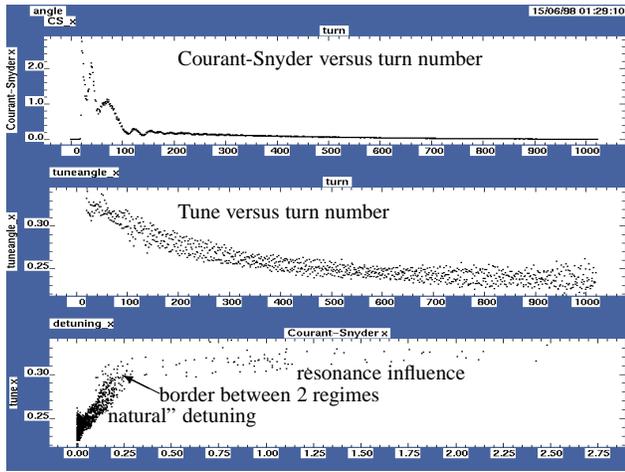


Figure 14: Detuning with amplitude (bottom). Notice two distinct regions.

because the tune is partially locked on the 3rd order resonance. Once the tune unlocks itself from the resonance, the behavior changes, and the detuning gets faster (left part of the picture).

### 3.3 Chaotic Approach To Resonance Islands

In this example we will follow the behavior of one very peculiar beam injection into LEP. For some unknown reasons the injection oscillation was not damped in the normal way (Fig. 15); rather, the beam moved toward some attraction point in the phase space. On its road to these points, a chaotic behavior was clearly observable. Almost all of the injected beam was lost within the first 1000 turns, and the losses were correlated with the period of the synchrotron oscillation, whose frequency is clearly visible in the FFT of the position data (Fig. 16). We can observe this behavior in a much clearer way by looking at the phase space, where we can also observe the same periodicity in the vertical plane (Fig. 17). The beam is captured by the attractors, and it is only released in the last turns of our sample (see Fig. 18). On its way to the attractor, the system oscillates between two states, as shown by Fig. 19. In the left part of the picture, only every 7th point in the phase space is shown (and joined to the previous one). In the right part, only every 14th point is shown, and this eliminates completely the

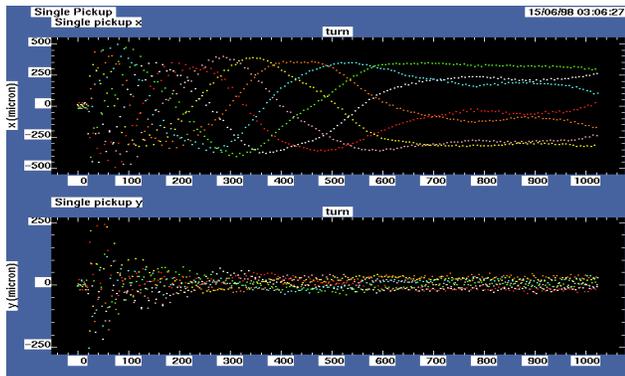


Figure 15: The beam moves towards a 7th order resonance.

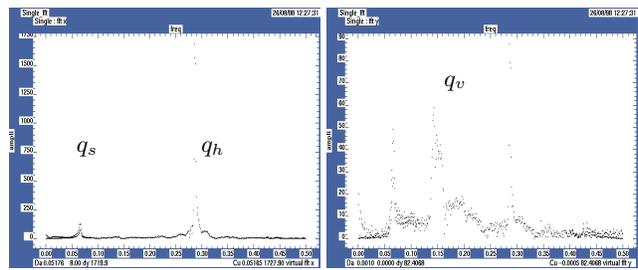


Figure 16: X and Y FFT of the position data.

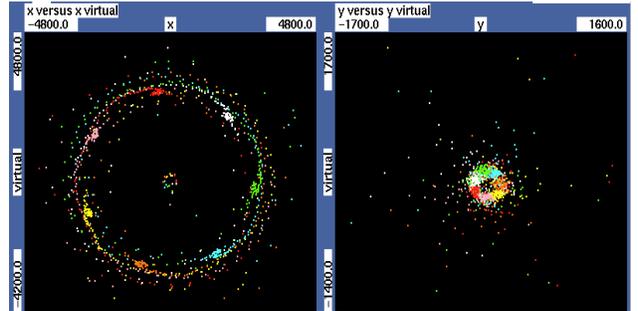


Figure 17: Phase space plots.

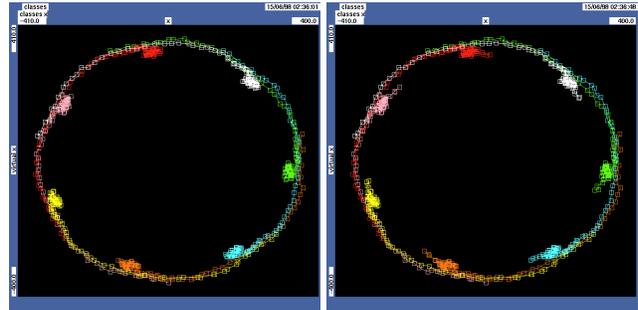


Figure 18: a) Turns 400-980.

b) Turns 400-1020.

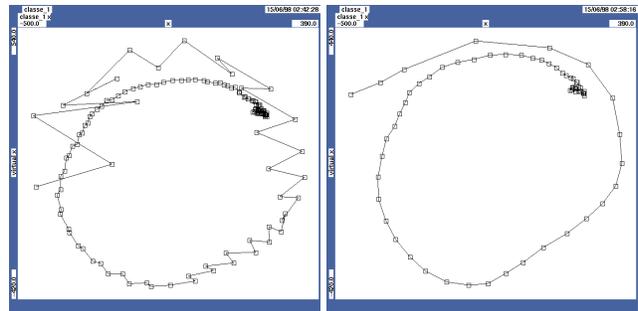


Figure 19: a) Every 7th point.

b) Every 14th point.

oscillations.

### 3.4 Filamentation

One limitation of our BPM system is that we can only observe the behavior of the center of mass of the particle distribution. This is not always representative of the behavior of the individual particles, as it can be seen from the following example. After a strong kick close to a resonance, the

oscillations of the center of mass of the particle distribution decay much earlier than the real oscillations of the individual particles. This effect is due to a rapid spreading of the particle distribution in the phase space ("Filamentation"). In our data set the oscillations seem to decay rapidly, but we are still loosing current at the end of the 1000 turns.

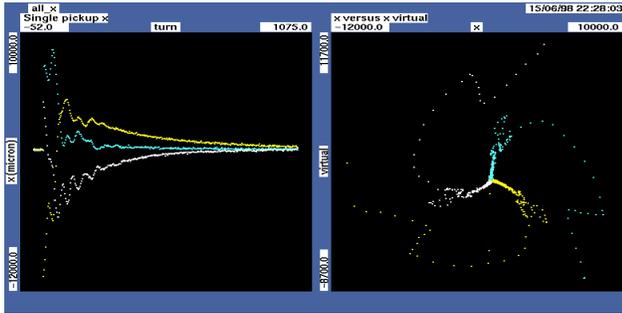


Figure 20: Position and phase space plots.

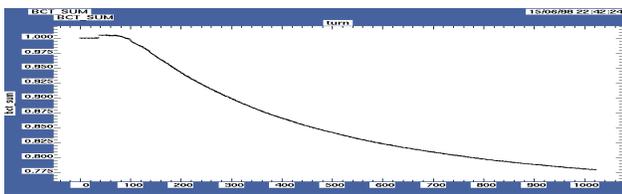


Figure 21: Evolution of bunch current during the 1000 turns.

### 3.5 Coupling

In all the previous examples we were mainly interested in investigating the behavior of the beam in the plane to which the kick was applied, neglecting the other plane. In the data set we will examine now the effect of the coupling is dominating. The energy originally provided to the beam via the horizontal kick is transferred from one plane to the other and viceversa. Amplitude maxima in one plane correspond to minima in the other plane. If we look at the phase space plots of the individual planes, we see that the X and Y distances from the center of the phase space beat strongly. Only by combining the two oscillations together we recover a smooth situation.

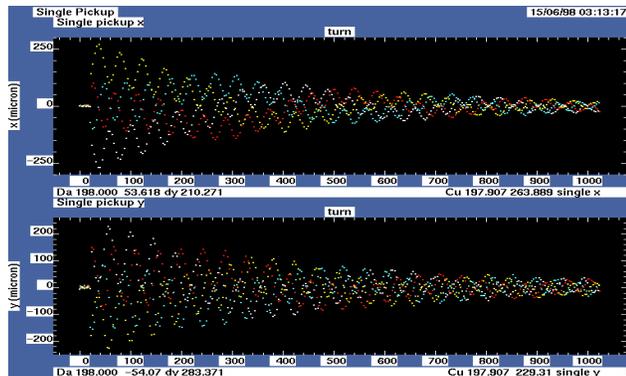


Figure 22: X and Y position.

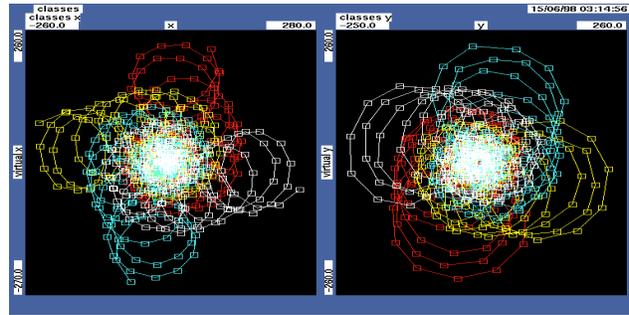


Figure 23: Phase space plots.

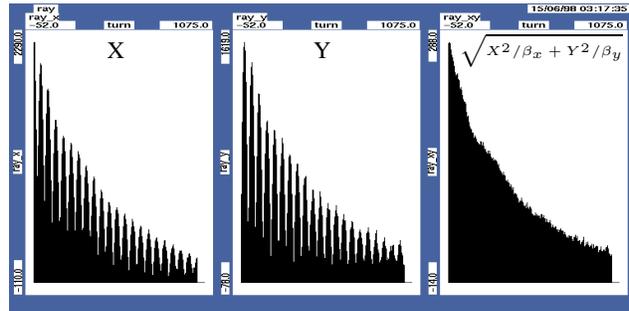


Figure 24: X,Y, and combined distance from the phase space center.

### 3.6 Non-Linearities Due To Beam-Beam Forces

The same method was applied to data obtained by submitting the beam to a continuous excitation of the betatron frequency. In the data set presented here, the two beams were colliding, and the beam-beam forces are probably responsible for the nonlinear behavior.

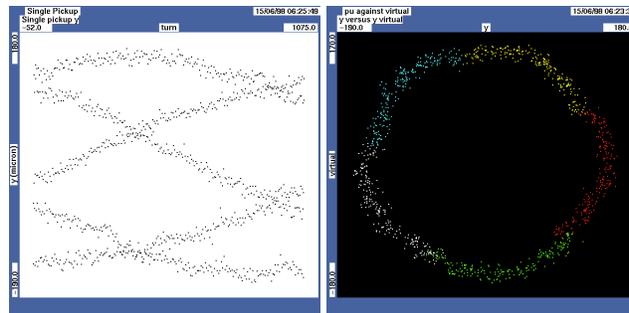


Figure 25: Phase space plot deformation induced by beam-beam forces.

## 4 REFERENCES

- [1] The BOM 1000 Turn Display : A Tool To Visualize The Transverse Phase-Space Topology At LEP, G.Morpurgo; Proceedings of EPAC98, Stockholm, 1998.
- [2] Applications Of Beam Diagnostic System At The VEPP-4, A.N.Dubrovin, A.S.Kalinin, D.N.Shatilov, E.A.Simonov, V.V.Smaluk; Proceedings of EPAC96, Sitges, 1996.

# USING 3-D PERSPECTIVES AS A VISUALIZATION TOOL FOR PHASE SPACE DATA

George H. Gillespie, Barrey W. Hill and Michael C. Lampel  
G. H. Gillespie Associates, Inc., P.O. Box 2961, Del Mar, CA 92014, U.S.A

## *Abstract*

Two-dimensional projections of six-dimensional phase space data are routinely used in the analysis of accelerator beam dynamics phenomena. Plots of the distribution of particle coordinates and momenta in a given phase plane (e.g.  $x, x'$ ), or for the coordinates of beam cross sections (e.g.  $x, y$ ), are among the most commonly used projections. Computer-based visualizations of higher-dimensional projections of phase space data offer the possibility of providing improved insight into complex beam dynamics phenomena. This paper illustrates one of these types of visualizations that use interactive three-dimensional perspective displays.

## 1 INTRODUCTION

There are a number of new tools available that focus on improving the productivity of scientists and engineers involved in the study of particle beams and accelerator beamlines. Increasingly powerful multi-particle simulation codes are being developed that offer unprecedented capability in modeling the beams in accelerator systems. Ryne and collaborators [1] are running three dimensional simulations in parallel computing environments that use as many as 100 million macro-particles, a number which is approaching the actual number of particles that occur within the bunches of real beams. Efficient ways to interpret and assimilate the detailed information contained in the particle distributions produced by these simulations should prove useful.

A prototype interactive three-dimensional (3-D) perspective display tool has been developed to explore its utility. The prototype tool was designed to work with any optics code that generates 6-D phase space distributions, but the work described here uses distributions created with the Particle Beam Optics Laboratory (PBO Lab™). The PBO Lab provides an intelligent graphic user interface that works with several accelerator modeling and simulation programs [2,3].

## 2 EXAMPLE BEAMLINER PROBLEM

To illustrate how the use of 3-D visualization can aid in understanding and designing particle optics beamlines an example problem will be used. The beamline is a four-cell, second-order achromatic bend of the type that has been frequently described in the literature and often used as a building block for many accelerator designs [4,5]. The four cells are identical and are composed of a sector bend, two quadrupoles and two sextupoles, interspersed with drifts. Each cell of the achromat has transverse phase advances of  $90^\circ$  and is sometimes referred to as a quarter-

wave transformer. Figure 1 illustrates an iconic representation of the example beamline.

A scalable version of the example achromatic bend has been set up with the PBO Lab using formulas that incorporate all of the dependent relationships between the different elements. A description of the construction of the example has been published previously [6]. The system has twelve (12) independent parameters, all defined for the first cell. Eight (8) are parameters related to the size of the beamline: the first drift length, the effective lengths and apertures for the first quadrupole, sextupole and dipole magnets, and the bend angle of the first dipole magnet. The remaining four (4) parameters are magnetic field strengths: two quadrupole pole tip fields and two sextupole pole tip fields. Note that the dipole field strength is not an independent parameter in this example. It is computed by one of the expert system rules of the PBO Lab [6], from the specified bend angle and orbit length of the dipole.

For the results presented here we use the same parameters for the achromat that are described in reference [6]. That design is for a 100 MeV electron beam. Each sector dipole has a  $10^\circ$  bend angle, so that the total bend for the 4-cell achromat is  $40^\circ$ . The length of the first drift is 5 cm, and the lengths of the first sextupole and quadrupole are 10 cm. The central trajectory length through the first dipole is also 10 cm. The drift between the first sextupole and quadrupole is set to be equal to the first drift length. The drift between the first quadrupole and dipole is set with a formula that assures that the length of the first half of the first cell is equal to ten (10) times the first drift length. All other lengths in the first cell are specified by algebraic expressions so that the second half of the cell is a mirror image of the first half cell. The first cell thus has a length of 1 meter in this example. This length may easily be adjusted, while still preserving the cell symmetry, by changing the lengths of the first few elements. The second cell of the achromat is completely defined in terms of the parameters of the first cell by algebraic expressions. The third and fourth cells are copies of the second cell. The TRANSPORT Module of the PBO Lab is used in two steps, to initially fit the quadrupole strengths to achieve the desired first-order optical conditions, and then to determine the sextupole strengths to eliminate the second-order aberrations. The third-order TRANSPORT version [7] was used for these fits. The results of this fitting procedure are given in reference [6].

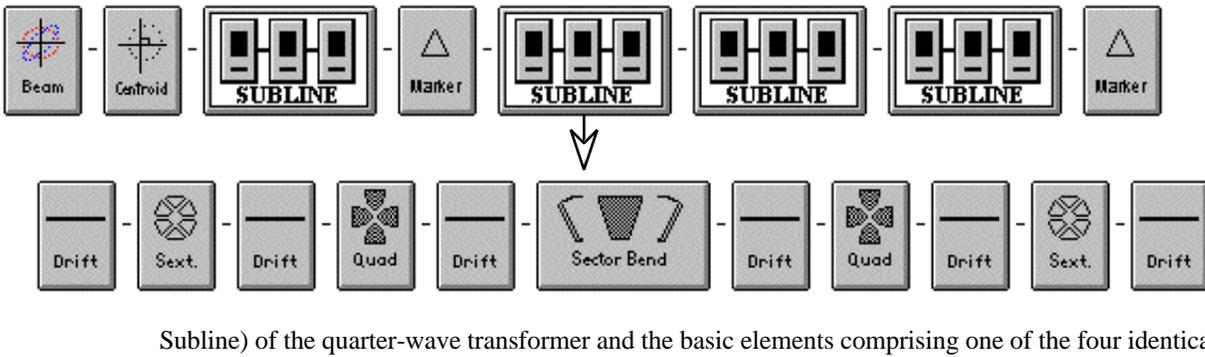


Figure 1. Icon image (from a PBO Lab Document Window) illustrating the four-cell layout (each cell represented by a

Subline) of the quarter-wave transformer and the basic elements comprising one of the four identical cells.

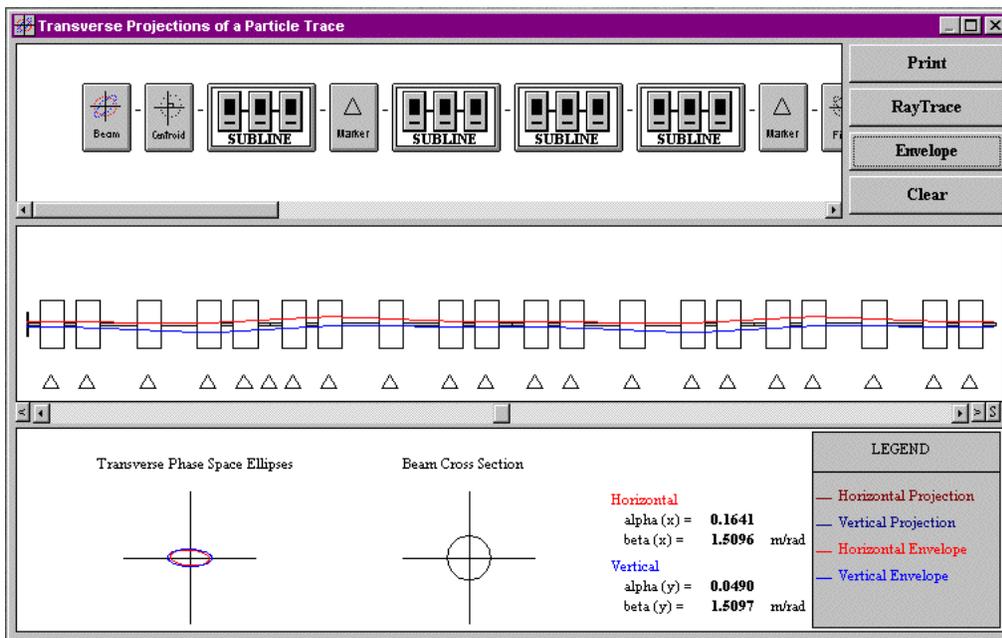


Figure 2. Four-cell achromat layout (top pane), side view scaled image of the achromat with horizontal (x) and vertical (y) envelopes overlain (center pane), and beam ellipses half way through the achromat (lower pane). The window illustrated is from the PBO Lab Trajectories Module.

Figure 2 shows a scaled layout of the example beamline, together with the first-order envelopes, and the transverse phase space ellipses and beam cross section at the mid-point of the achromat.

Table 1. Semi-axes parameters for the initial beam.

Semi-Axis	Value	Units
x	4.000	mm
x'	0.101	mrاد
r12	0.14	-
y	4.000	mm
y'	0.101	mrاد
r34	0.16	-
z	5.000	mm
	3-27	%
r56	-0.09	-

### 3 2-D VIEWS OF 6-D DATA

The TURTLE Module of the PBO Lab was used to generate two-dimensional (2-D) views of a beam propagating through the example beamline. TURTLE is a multi-particle ray tracing program [8] that models beams with a full six-dimensional phase space representation. For this work, an initial beam of 1000 macro-particles was used with semi-axes parameters as given in Table 1. The initial beam bunch is nearly spherical, with an aspect ratio (length/radius) of 1.25, but the correlation parameters are not symmetric. The transverse divergences are small and different values of the momentum spread,  $\delta$ , were used.

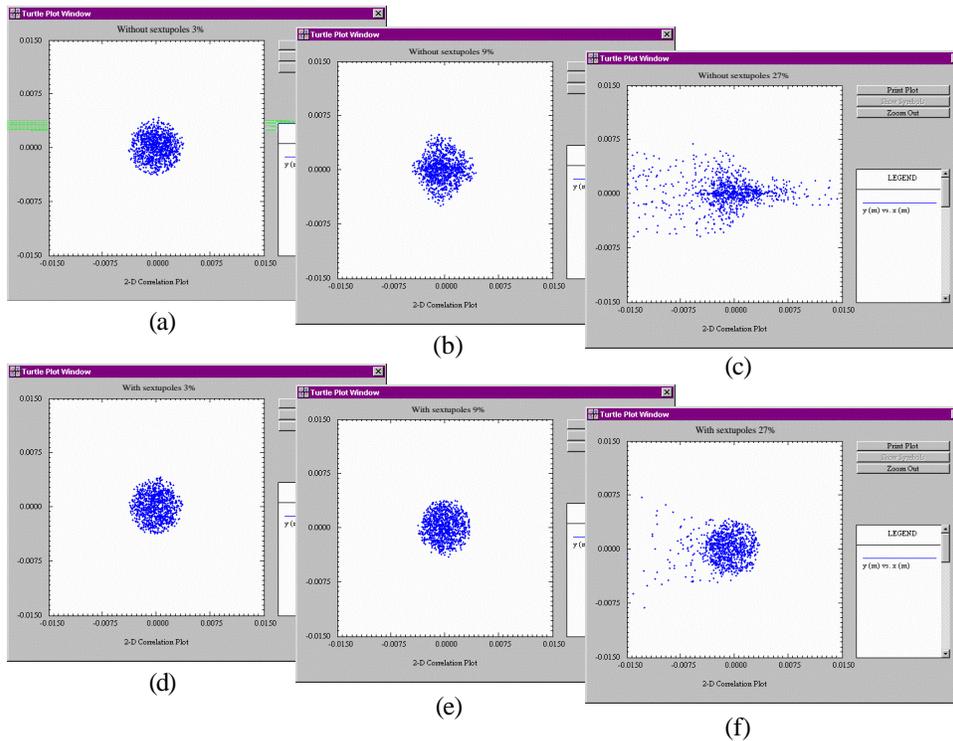


Figure 3. Two-dimensional scatter plots of the beam cross section at the end of the achromatic bend, without sextupole correctors [top (a)-(c)] and with sextupole correctors [bottom (d)-(f)]. In both cases, results are shown for three different values of the initial momentum spread : 3%, 9% and 27%, increasing from left to right. Window images are from the PBO Lab TURTLE Module.

Figure 3 illustrates the results of six different TURTLE simulations of the achromat performance. For the first-order design (without sextupole correction), the cylindrical symmetry of the initial beam is preserved by the achromat for a momentum spread of 3%, but aberrations are apparent at  $\approx 9\%$ , and for  $\approx 27\%$  the beam cross section projection is highly distorted. For the second-order achromat (with the sextupole correction), the cylindrical symmetry remains largely intact throughout the range of momentum spreads illustrated, although for  $\approx 27\%$  some scattering of particles in the bend plane is apparent. For a precision spot imaging system, the results suggest that the first-order design is adequate for  $\approx 3\%$ , that a second-order achromat will be required for  $\approx 9\%$ , and the second-order achromat may be adequate for  $\approx 27\%$  with aperture scraping, if the particle loss can be tolerated. What do these beams look like in 3-D?

#### 4 3-D PERSPECTIVE VIEWS

A 3-D perspective display tool has been developed on a Macintosh computer in order to visualize particle distributions in three dimensions. The tool uses data provided by a 6-column data file of phase space coordinate and momentum values. A simple dialog-type window is used to select the distribution file and the three axes for plotting. Figure 4 illustrates the window.

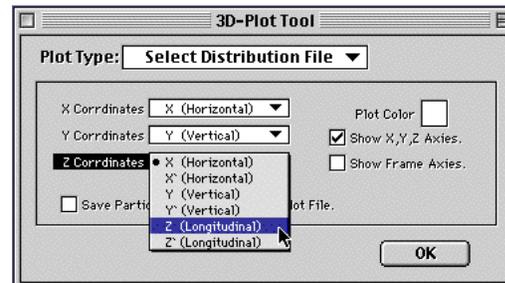


Figure 4. Selection of axes for 3-D visualization tool.

Other settings, such as the degree of the perspective and the size of the data points, can be adjusted interactively from menu selections after the data is displayed. Keyboard commands are used to zoom in and out, to perform controlled rotations about specific axes, and for other tasks. Mouse-based interactive features include the ability to “grab” the beam bunch and rotate it about any axis in order view the distribution from any direction. A variety of other interactive features are also available.

The capability to interactively change the viewing direction is one of the key features that can be used to gain insight into the physical phenomena occurring in the transport of the beam. Figure 5 illustrates a sequence of “snapshots” taken by rotating the viewpoint.

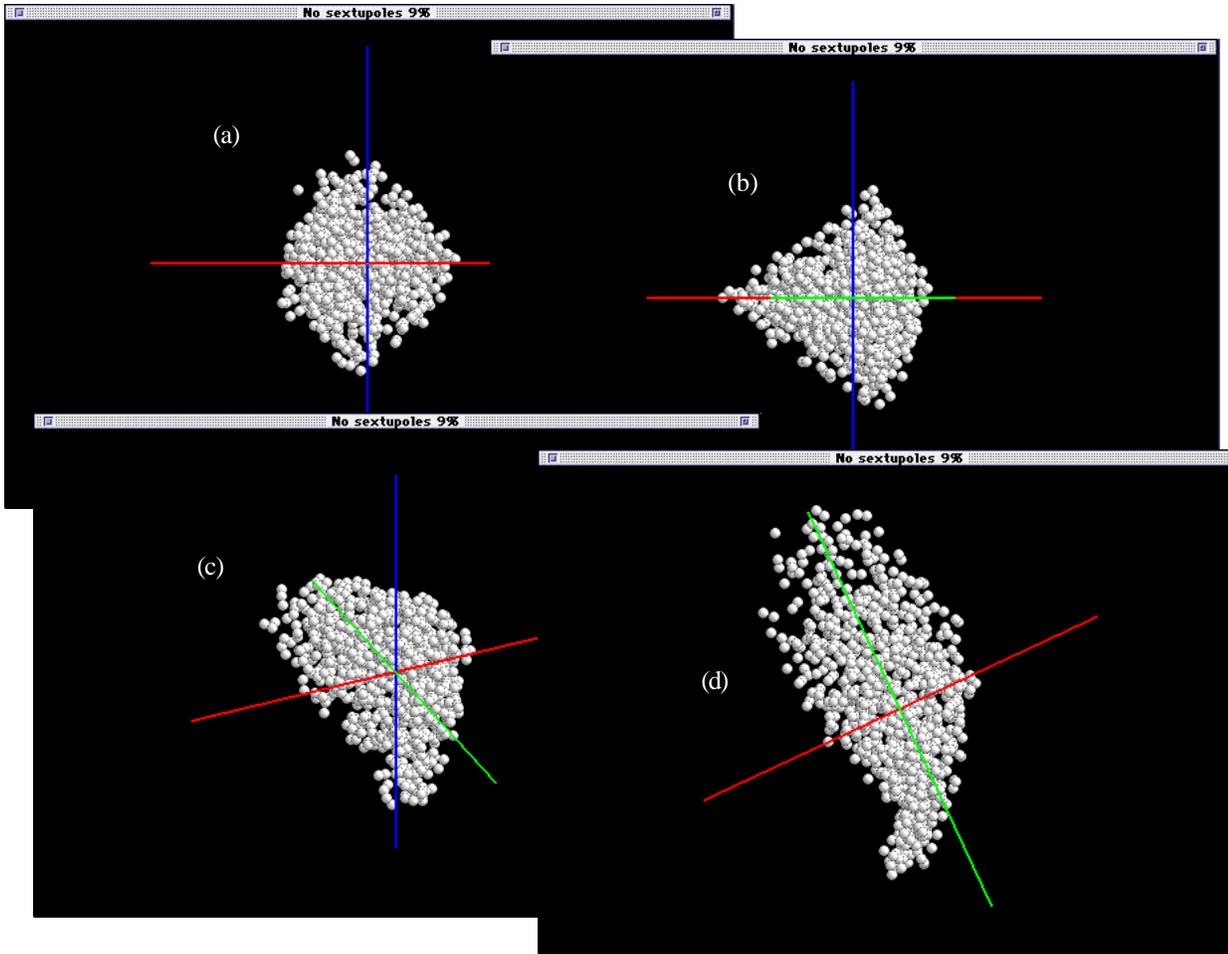


Figure 5. Snapshots of the rotation of the three-dimensional (x,y,z) perspective view of the particle distribution at the exit of the uncorrected achromat. The initial, nearly spherical, beam has a 9% momentum spread. The x-y view in (a), corresponding to Figure 4(b), has been rotated about the y (vertical) axis by approximately  $45^\circ$  in (b), rotated about the new horizontal axis by  $45^\circ$  in (c), and then by  $90^\circ$  in (d). Fanning in different directions at the (longitudinal) ends of the beam, as well as a wrapping of the beam around the y-axis, are apparent.

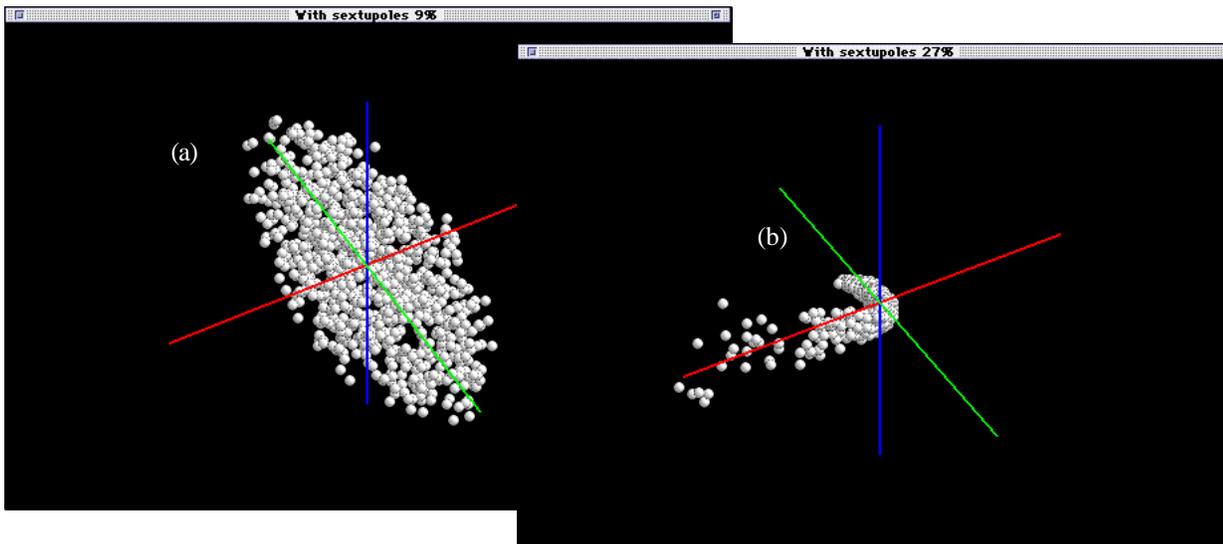


Figure 6. Snapshots of exit distributions of the corrected achromat, for initial beams with momentum spreads of (a) 9% and (b) 27%. The view orientations correspond to that of Figure 5(c), but are not to the same scale.

Figures 5 and 6 illustrate selected views of the three dimensional spatial distribution of the exit beam, for the uncorrected and corrected beamline, respectively. The uncorrected beam illustrated in Figure 5 fans out in different directions: the beam spreads in the horizontal ( $x$ - $z$ ) or bend plane at one (longitudinal) end, but in the vertical ( $y$ - $z$ ) plane at the other end. The beam also tends to wrap around the vertical axis. This same beam (9% momentum spread) is a well-formed 3-D ellipsoid when corrected, as shown in Figure 6(a).

The caption to Figure 5 states that the fanning and wrapping are “apparent” for the uncorrected achromat, but in fact, these effects are difficult to see from a set of snapshots. However, the effects are immediately recognizable by a user interactively rotating the beam bunch on the computer screen; the human eye and brain quickly process the image as a three dimensional object being rotated in space.

At larger momentum spread (27%), the fanning and wrapping become more distorted for the uncorrected beamline (not shown). As Figure 6(b) shows, the wrapping is still apparent for the corrected achromat, while the fanning is effectively confined to one end of the beam. Figure 6(b) also demonstrates that the scattered particles appearing in Figure 4(f) are confined to a narrow fan in the vertical plane at one end of the beam bunch.

## 5 SUMMARY

A prototype interactive tool for visualizing three-dimensional scatter plots has been developed. The tool can be used for displaying any three of the coordinate or momentum variables of a 6-D phase space distribution in a 3-D perspective view. The 3-D display can be interactively manipulated to study the distribution: the image can be rotated about any axis, the perspective can be changed, zoom in and out is supported, and the sizes of the particles and axes can be adjusted. The tool is easy to use and has proven useful in examining the nature of the non-linear structure of beams transported through a classical second-order achromat. The tool has also been used with PARMILA-generated distributions to study the free energy relaxation of non-equipartitioned beams [9] and should prove useful for exploring a variety of other physical phenomena associated with particle beams.

## ACKNOWLEDGMENT

This work has been supported by Internal Research and Development (IR&D) funds of G. H. Gillespie Associates, Inc. The PBO Lab [2,3,6] was developed with support from the U. S. Department of Energy under SBIR grant number DE-FG03-94ER81767 and is now available commercially [10].

## REFERENCES

- [1] S Habib, J Qiang, R. Ryne, K. Ko, Z. Li, B. McCandless, W. Mi, C. Ng, M. Saporov, V. Srinivas, Y. Sun, and X. Zhan, “The Department of Energy Grand Challenge in Computational Accelerator Physics,” to be published in the proceedings of XIXth International Linear Accelerator Conference (Chicago, 1998).
- [2] G. H. Gillespie, B. W. Hill, N. A. Brown, H. Martono and D. C. Carey, “The Particle Beam Optics Interactive Computer Laboratory,” AIP Conference Proceedings 391, 264-269 (1996).
- [3] G. H. Gillespie, B. W. Hill, H. Martono, J. M. Moore, N. A. Brown, and M. C. Lampel, “The Particle Beam Optics Interactive Computer Laboratory for Personal Computers and Workstations,” Proceedings of the 1997 IEEE Particle Accelerator Conference, 2562-2564 (1998).
- [4] K. L. Brown, “A Second-Order Magnetic Optical Achromat,” IEEE Trans. Nuc. Sci. NS-26, 3490-3492 (1979).
- [5] K. L. Brown and R. V. Servranckx, “Applications of the Second-Order Achromat Concept to the Design of Particle Accelerators,” IEEE Trans. Nuc. Sci. NS-32, 2288-2290 (1985).
- [6] G. H. Gillespie, B. W. Hill, H. Martono, J. M. Moore, M. C. Lampel and N. A. Brown, “Using the Particle Beam Optics Laboratory (PBO Lab™) for Beamline Design and Analysis,” to be published in proceedings of the 15th International Conference on Cyclotrons and Their Applications (Caen, France, 15-19 June 1998), 4 pages (1998).
- [7] D. C. Carey, K. L. Brown and F. Rothacker, “Third-Order TRANSPORT- A Computer Program for Designing Charged Particle Beam Transport Systems,” Stanford Linear Accelerator Center Report No. SLAC-R-95-462, 295 pages (1995).
- [8] D. C. Carey, “TURTLE (Trace Unlimited Rays Through Lumped Elements) A Computer Program for Simulating Charged Particle Beam Transport Systems,” Fermi National Accelerator Laboratory Report No. NAL-64, 45 pages (1978).
- [9] N. A. Brown, G. H. Gillespie and B. W. Hill, “Free Energy Relaxation in a Drift-Tube Linac,” Proceedings of 1997 IEEE Particle Accelerator Conference, 1894-1896 (1998).
- [10] *PBO Lab™* is available from AccelSoft Inc., see [www.ghga.com/accelsoft](http://www.ghga.com/accelsoft).

# NEW METHOD FOR KLYSTRON MODELING

Y. H. Chin, KEK, 1-1 Oho, Tsukuba-shi, Ibaraki-ken, 305, Japan

## Abstract

We have developed a new method for a realistic and more accurate simulation of klystron using the MAGIC code. MAGIC is the 2.5-D or 3-D, fully electromagnetic and relativistic particle-in-cell code for self-consistent simulation of plasma. It solves the Maxwell equations in time domain at particle presence for a given geometrical structure. It uses no model or approximation for the beam-cavity interaction, and thus keeps all physical processes intact. With MAGIC, a comprehensive, full-scale simulation of klystron from cathode to collector can be carried out, unlike other codes that are specialized for simulation of only parts of klystron. It has been applied to the solenoid-focused KEK XB72K No.8 and No.9 klystrons, the SLAC XL-4 klystron, and the BINP PPM klystron. Simulation results for all of them show good agreements with measurements. We have also developed a systematic design method for high efficiency and low gradient traveling-wave (TW) output structure. All these inventions were crystallized in the design of a new solenoid-focused XB72K No.10. Its predicted performance is 126 MW output power (efficiency 48.5%) with peak surface field of about 77 MV/m, low enough to sustain a 1.5  $\mu$ s long pulse. It is now in manufacturing and testing is scheduled to start from December 1998.

## 1 JLC KLYSTRON PROGRAM

The 1-TeV JLC (Japan  $e^+e^-$  Linear Collider) project[1] requires about 3200 (/linac) klystrons operating at 75 MW output power with 1.5  $\mu$ s pulse length. The main parameters of solenoid-focused klystron are tabulated in the second column of Table 1. The 120 MW-class X-band klystron program at KEK[2], originally designed for 80 MW peak power at 800 ns pulse length, has already produced 9 klystrons with solenoidal focusing system. To reduce the maximum surface field in the output cavity, the traveling-wave (TW) multi-cell structure has been adopted since the XB72K No.6. Four TW klystrons have been built and tested. All of them share the same gun (1.2 micropervance and the beam area convergence of 110:1) and the buncher (one input, two gain and one bunching cavities). Only the output structures have been redesigned each time at BINP. XB72K No.8 (5 cell TW) attained a power of 55 MW at 500 ns, but the efficiency is only 22%. XB72K No. 9 (4 cell TW) produced 72 MW at 520 kV for a short pulse of 200 ns so far. The efficiency is increased to 31% and no sign of RF instability has been observed. The limitation in the pulse length attributes a poor conditioning of the klystron. The latest tube, XB72K No.10, was designed at KEK, and is being build in Toshiba.

Apart from the solenoid-focused XB72K series, KEK has also started a PPM (periodic permanent magnet) klystron development program. The design parameters are shown in the last column of Table 1. Its goal is to produce a 75MW PPM klystron with an efficiency of 60 % at 1.5  $\mu$ s or longer pulse. The first PPM klystron was designed and build by BINP in the collaboration with KEK. It has a gun with beam area convergence of 400:1 for the micropervance of 0.93. The PPM focusing system with 18 poles (9 periods) produces the constant peak magnetic field of 3.8 kG. The field in the output structure is still periodic, but tapered down to 2.4 kG. There are two solenoid coils located at the beam entrance for a smooth transport of a beam to the PPM section. It achieved 77 MW at 100 ns, but there is a clear sign of RF instability at higher frequencies. The DC current monitor in the collector shows about 30 % loss of particle when RF is on. The second PPM klystron, XB PPM No.1, is being designed at KEK.

Table 1: Specifications of X-band solenoid-focused and PPM-focused klystrons for JLC.

	XB72K	PPM
Operating frequency (GHz)	11.424	11.424
RF pulse length ( $\mu$ s)	$\geq 1.5$	$\geq 1.5$
Peak output power (MW)	75	75
Repetition rate (pps)	120	120
RF efficiency (%)	47	60
Band-width (MHz)	100	120
Beam voltage (kV)	550	480
Perveance ( $\times 10^{-6}$ )	1.2	0.8
Maximum focusing field (kG)	6.5	
Gain (dB)	53-56	53-56

## 2 MAGIC CODE

After a series of disappointing performance of XB72K series, several lessons had been learned. First, KEK should have its own team to specialize the klystron design and overhaul the design process. Second, a new klystron simulation code was needed for a more realistic design of klystron, particularly, that of a TW output structure. The one-dimensional disk model code, DISKLY, had been used by BINP for design of the TW structure from XB72K No.5 till No.9. This code uses an equivalent circuit model (port approximation) to simulate a TW structure and tends to predict the efficiency much larger (nearly twice larger) than the experimental results. For the design of a new klystron, XB72K No.10, we have developed a method to use the MAGIC code[3] to simulate and design a klystron. MAGIC is the 2.5-D or 3-D, fully electromagnetic and

relativistic particle-in-cell code for self-consistent simulation of plasma. It solves the Maxwell equations directly at particle presence by the finite difference method in time like ABCI [4] or MAFIA. It requires only the geometrical structure of the cavity and assumes no model (neither port approximation nor equivalent circuit) for the beam-cavity interaction. The static magnetic field can be applied to a structure. Advantages of MAGIC are its accuracy and versatility. Even an electron gun can be simulated with results in good agreements with measurements. Simulation results can be imported/exported from one section of klystron to another, allowing a consistent simulation of the entire klystron without loss of physics. Only disadvantage is that it is time consuming.

### 3 FUNCTIONAL COMPARISON OF AVAILABLE CODES

Table 2 shows the functional comparison of computer codes available for klystron simulations. MAGIC is the only code that can simulate all parts of klystron from gun to collector. ARSENAL[5] is closest to MAGIC in functional performance, but cannot handle a TW multi-cell structure. CONDOR[6] can simulate a TW structure, but requires a beam input from a gun that needs to be simulated by other code such as EGUN[7]. In the migration of beam and fields from one code to another, two programs must be well matched to avoid any incomplete transfer of information and resulting unphysical phenomena.

Table 2: Functional comparison of available codes.

	Dimension	Gun	Buncher	Single-cell output cavity	Multi-cell output cavity
MAGIC	2.5, 3	O	O	O	O
EGUN	2.5	O	×	×	×
CONDOR	2.5	×	O	O	O
FCI [8]	2.5	×	O	O	×
ARSENAL	2.5	O	O	O	×
JPNDISK	1	×	O	O	×
DISKLY	1	×	O	O	×

### 4 SIMULATION METHOD USING MAGIC

We divide a klystron into three sections:

- Electron gun
- Buncher section (an input, gain and bunching cavities + drift space)
- TW output structure

The simulation techniques are described in detail in Ref. [9]. Here, we briefly summarize them.

#### 3.1 Electron gun

The gun simulation is done by specifying an emission area (cathode) and an applied voltage along a line between a wehnelt and an anode. The number of emitted particles can be specified per unit cell volume and unit time-step. The applied magnetic fields (both  $B_z$  and  $B_r$ ) must be specified over the structure, not just on beam axis. They can be calculated using codes such as POISSON (for solenoid field) and PANDIRA[10] (for PPM). These programs requires the exact configuration of coils, yokes, or permanent magnets and their properties as input.

Figure 1 shows the comparison of beam profile simulated by EGUN and MAGIC for the XB72K-series gun. They look nearly identical. The simulated perveance for three different guns and the measured values are tabulated in Table 3. MAGIC simulations are in excellent agreement with the measurements, while the EGUN tends to produce a 5-10 % larger value than the measurements. This behavior was also reported in simulation of SLAC 50 MW PPM klystron by EGUN [11].

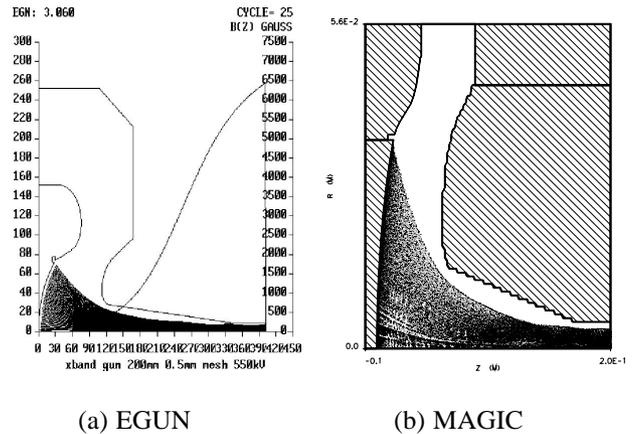


Figure 1: Beam profile from the XB72K gun simulated by (a) EGUN and (b) MAGIC.

Table 3: Comparison of the simulated perveances and the measured values.

Klystron	Frequency (MHz)	Micro perveance		
		MAGIC	EGUN	Measured
XB 72K	11,424	2.03	1.89	2.05
PV3030	2,856	1.19	1.10	1.2
5045	2,856	1.99	1.78	2.0

#### 3.2 Buncher section

The input cavity needs a different treatment from other cavities, because the RF power is given externally, rather

than being induced by a beam. Since a beam stays almost as DC while passing the input cavity, the beam induced voltage is negligible. Therefore, we just need to specify the applied RF voltage along an electric field line between the cavity gap. The field distribution of the fundamental mode should be computed by MAGIC priory and used as input. Other cavities need to be tuned to correct fundamental frequencies by adjusting the cavity aperture on mesh. The beam-induced voltage in cavities are monitored to measure the necessary RF cycles for saturation. In most of cases, about 200-300 RF cycles are enough. To speed up the saturation, a DC beam current from gun is increased smoothly and slowly from zero to the full value at the first 10-20 RF cycles.

Figures 2 (a) and (b) show spatial distributions of beam in the input+gain cavity section and in the bunching cavity section of the XB72K No.10 buncher, respectively. The strong bunching of beam (RF current/DC current  $\approx 1.7$ ) is created toward the end of the buncher section.

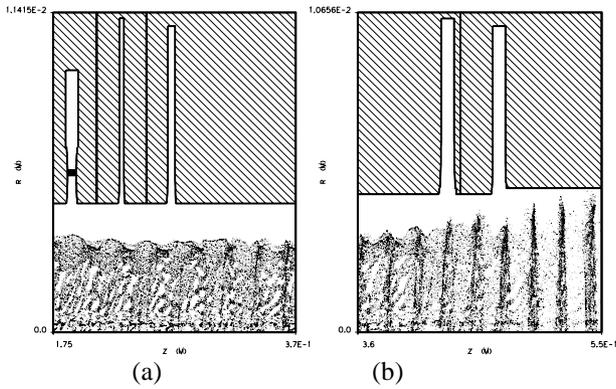


Figure 2: Spatial distribution of beam (a) in the input+gain cavity section and (b) in the bunching cavity section of the XB72K No.10 buncher.

### 3.3 Traveling-Wave (TW) output structure

Simulation of TW output structure is quite straightforward as any other cavity. In order to simulate effects of a non-axis-symmetrical output coupler by the 2.5-D version of MAGIC, we model it by a ring-shaped conductor which has the same complex  $S_{11}$ -matrix element (i.e., the reflection coefficient for amplitude and phase). This is illustrated in Fig. 3. There are three free parameters to fit the frequency dependent  $S_{11}$ -matrix element: the conductance, and the inner and the outer radii of the conductor. For details of the output coupler modeling, refer to Ref.[9]. As shown later, simulation results for many klystrons seem to verify the validity of this approximation.

Before inventing the above conductor approximation, we have considered a use of an axis-symmetrical radial transmission line to model a 3-D coupler. However, this method cuts the output structure into two disconnected parts, and thus an artificial DC voltage is induced by the

DC component of beam at the output cell to which the output couplers are attached. This artificial DC voltage causes a non-negligible effect to the particle dynamics, and results in error. Figure 4 shows the simulation results for the output structure of XB72K No.10.

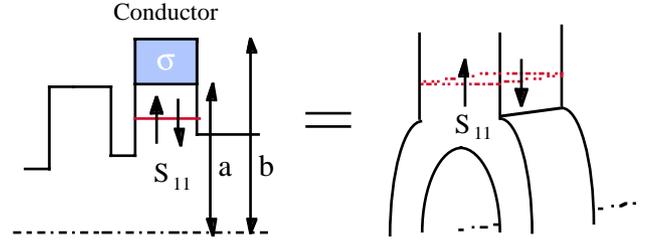


Figure 3: Illustration for 2.5-D modeling of 3-D output coupler using a conductor.

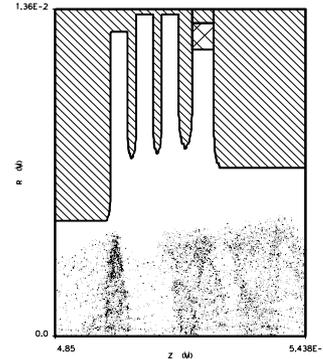


Figure 4: Simulation of XB72K No. 10 in the output structure.

## 5 SIMULATION RESULTS AND MEASUREMENTS

Figure 5 shows the simulation results of MAGIC and the experimental data for the saturated output power vs. beam voltage for XB72K No.8 klystron. Excellent agreements can be seen. The closed triangles in Fig. 5 are DISKLY simulations. It reveals the accuracy limitation of the 1-D disk model code.

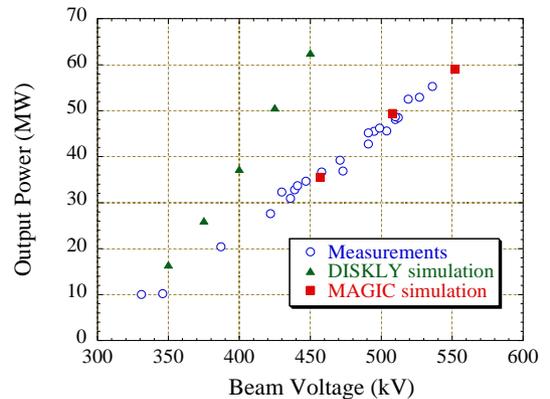


Figure 5: Simulation results of MAGIC and DISKLY and the measurement data for XB72K No.8 klystron.

Let us move to the simulation of SLAC XL-4 klystron. XL-4 klystron produced 50 MW at 400 kV with 1.5  $\mu$ s pulses at 120 pps. It attained 75MW at 450 kV, but the pulse length could go up only to 1.2  $\mu$ s before the RF breakdown in the output cavity. The simulation results for the output power are compared with the measurements in Fig. 6. MAGIC simulations reproduce the measurement data quite well. The CONDOR prediction at 450 kV, denoted by the closed triangle, was at 10% too high. Figure 7 shows the output power vs. the input power for XL-4. It is clear that the simulation reproduces the measured gain curve well.

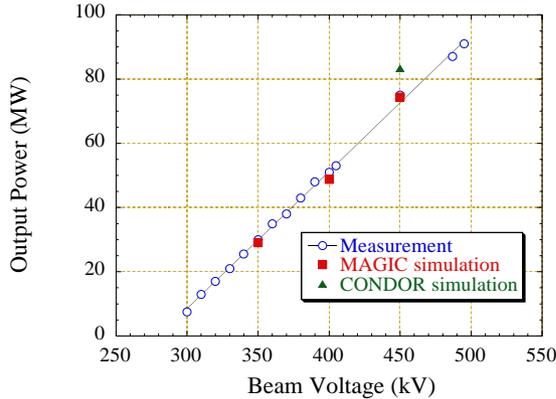


Figure 6: Simulation results of MAGIC and CONDOR and the measurement data for the SLAC XL-4 klystron.

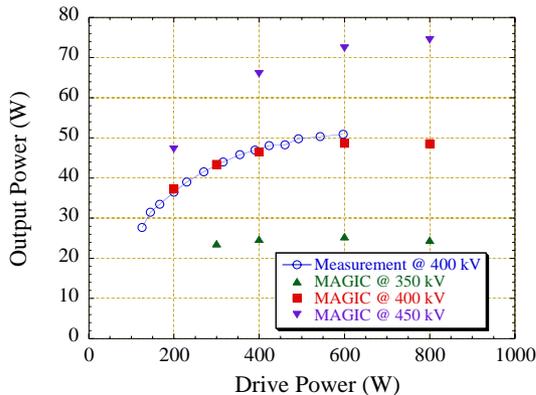


Figure 7: Simulations and measurement data of the output power vs. the input power for the SLAC XL-4 klystron.

Our simulation method can also make an accurate prediction of performance for a PPM klystron. Figure 8 shows the simulation results and the measured values of output power for the BINP PPM klystron. The evolution of DC and RF beam current as a function of distance from the gun is plotted in Fig. 9. The sudden drop of the DC current is due to the particle interception at the final cell of the output cavity. The interception is caused by lack of focusing for particles that drop to the stop-band voltage after losing energy to the traveling-wave. This simulation result explains the experimental observation of significant particle loss described in Section 1.

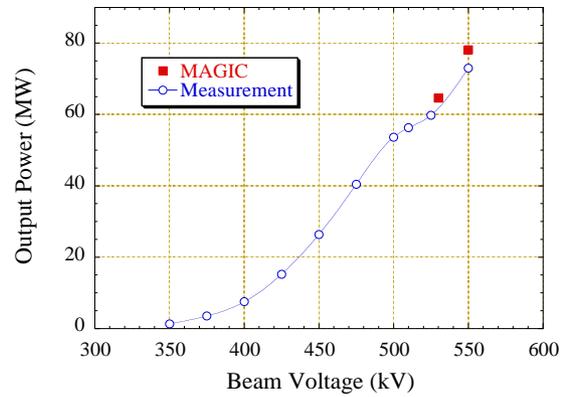


Figure 8: Simulation results for the BINP PPM klystron.

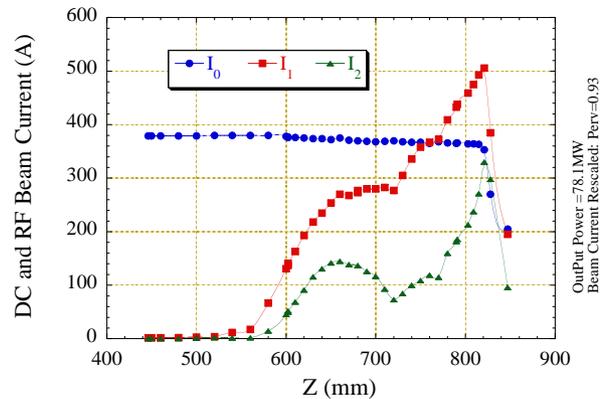


Figure 9: Evolution of the DC and RF beam current in the BINP PPM klystron.

### 3 XB72K NO.10 DESIGN

XB72K No.10 is the last solenoid-focused klystron in the XB72K series. Main changes from the previous XB72K klystrons are the buncher section and the TW output structure. The operational experience with the previous klystrons proved that the gun portion of XB72K has sufficient performance (1.2 micropervance at 2  $\mu$ s pulse length) and no interception of particles has been observed. The old buncher has two gain cavities and only one bunching cavity. It has a poor RF power generation capability: the RF current /DC current is only 1.2 at the entrance of the output structure. In XB72K No.10, one more bunching cavity was added and the drift space was lengthened to 16cm. The stagger tuning of gain cavities was also adopted to increase the band-width to the current specification of 100 MHz.

The most challenging part of XB72K No. 10 design is a high efficiency and low gradient TW output structure. MAGIC is quite useful for getting an accurate estimate of klystron performance, but the design of an effective TW structure is another matter. A systematic design method was needed to avoid getting lost in the freedom of too many parameters.

For this end, we have developed a simple-minded theory of a constant group/phase velocity TW structure.

The idea is to let the power flow with a constant group velocity throughout the structure, while evolving due to merge of the extracted power from a beam. The Q-value at the output port is matched to this group velocity so that the power exits at the same speed as it flows in the structure. This smooth flow of power prevents congestion at local spots and thus the electromagnetic energy density is more equally distributed in the structure

It is also better to keep the phase velocity constant (approximately equal to the average beam velocity) from the first to the last cell, rather than being matched with the declining beam velocity. When the perfect synchronization of traveling-wave and the beam is tried, the beam loses energy too quickly to the wave, and its velocity becomes too slow to be matched with the wave after a few cells (XB72K No. 10 has four cells). The beam then moves to the acceleration phase of the wave and starts to get energy back. The energy extraction efficiency of each cell does not have to be too good. Only the total efficiency of all cells matters. It is more important to keep the beam in the deceleration phase of the wave all the time. In our method, the traveling-wave travels behind the beam at first, and catches it up with in the middle of the structure. It then moves ahead of the beam, but exits from the output port before the beam slips into the acceleration phase of the wave.

We also demand that each cell is operated in  $2/3\pi$  mode at 11.424 GHz. The cell length is also constant except the last cell (slightly longer to reduce the field gradient). As the result, the cells become almost identical. We then tapered up the iris aperture slightly to equalize the field gradient among the cells. In this method, once the group and the phase velocities are chosen, the geometry of the structure are almost uniquely determined. The structure of output port can be adjusted to control the reflection of power to maximize the output power.

The predicted output power vs. the beam voltage is plotted in Fig. 10:

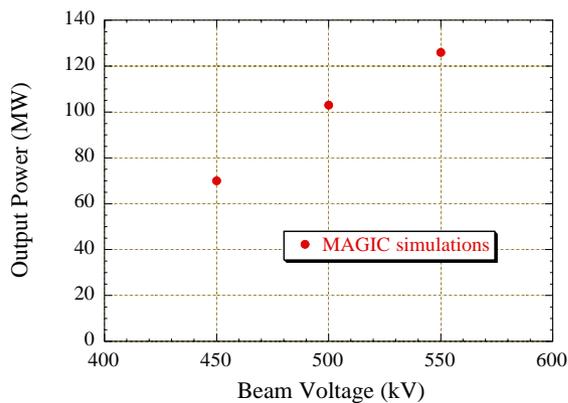


Figure 10: Predicted output power vs. beam voltage for the XB72K No.10.

The predicted performance is summarized in Table 4. Figure 11 shows comparison between XB72K No.10 and

SLAC XL-4 for the saturated power vs. the maximum field gradient in the output structure. Both have similar efficiencies of about 48%, but the maximum gradient of XB72K No.10 is about 20% lower than that of XL-4, though the power is 67% larger. In XB72K No.10, the fairly constant gradient is achieved in the output structure. This comparison indicates that the XB72K TW output structure can attain 120 MW power at a longer pulse than XL-4 at 75 MW without cavity breakdown. At 75MW, XB72K can tolerate an even longer pulse. It is now in manufacturing and testing will begin in November 1998.

Table 4: Predicted performance of XB72K No. 10.

Peak output power	126 MW
Beam voltage	550 kV
Efficiency	48.5%
Maximum field gradient in TW	77 MV/m
Pulse length	1.5 $\mu$ s or longer
Band-width	100 MHz
Gain	53 dB

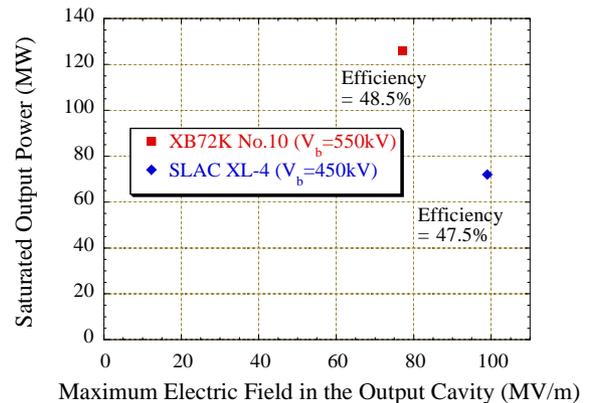


Figure 11: Saturated power versus the maximum field gradient in the output structure for XB72K No.10 and SLAC XL-4.

## 12 REFERENCES

- [1] JLC Design Study, KEK, April 1997.
- [2] Y. H. Chin, et. al., in Proc. of EPAC98, 1998.
- [3] MAGIC User's Manual, Mission Research Corporation, MRC/WDC-R-409, 1997.
- [4] Y. H. Chin, "User's Guide for ABCI Version 8.8", LBL-35258 and CERN SL/94-02 (AP (1994)..
- [5] A. N. Sandalov, et. al., in Proc. of RF96, KEK Proc. 97-1, pp.185-194, 1997.
- [6] B. Aimonetti, et. al., "CONDOR User's Guide", Livermore Computing Systems Document, 1988.
- [7] W. B. Herrmannsfeldt, SLAC-PUB-6498 (1994).
- [8] T. Shintake, Nucl. Instr. Methods A363, p.83, 1995.
- [9] S. Michizono, S. Matsmoto, and H. Tsutsui in this proceedings.
- [10] J. H. Billen and L. M. Young, "POISSON SUPERFISH", LA-UR-96-1834 (1997).
- [11] D. Sprehn et.al, in Proc. of RF96, KEK Proc. 97-1. pp.81-90, 1997.

# Numerical Studies of Wake Excitation in Plasma Channels\*

B. A. Shadwick and J. S. Wurtele

Department of Physics, UC Berkeley and Center for Beam Physics, LBNL, Berkeley CA, USA

## Abstract

The wake fields produced by an intense, short laser pulse propagating in a plasma channel with an arbitrary density profile is investigated. Plasma channels, viewed as accelerating structures, have many desirable features that are not shared by a homogeneous plasma. They are also becoming experimentally realizable. As part of an overall program to analyze plasma channels as accelerating structures, a new fluid simulation code has been developed with the primary purpose of producing fast tools to explore parameter space for both theoretical investigation of accelerator performance as well as the modeling and design of experiments. This code has flexible physics content, for example, the laser can either be fully resolved temporally or treated as ponderomotive force. An important feature, from the accelerator design point of view, is the capability to study beam propagation dynamics. We present preliminary results consisting of a detailed analysis of the transverse structure of the wake for a wide range of experimentally accessible channel profiles and characteristics of the corresponding accelerated beam.

## 1 INTRODUCTION

As accelerating structures, plasmas have the desirable ability to support extremely large gradients without experiencing the electrical breakdown which limits the gradient in conventional structures. Since this concept was first introduced, numerous configurations for particle acceleration in plasmas with uniform density have been proposed. (See Esarey *et al.* [1] for a comprehensive review of plasma accelerators concepts.) In the case of laser driven plasma accelerators, the usable accelerating length is determined by the distance over which the laser pulse maintains a high intensity. That is, the efficient use of the laser energy is limited by diffraction of the drive pulse. At very high intensities, relativistic optical guiding and self-channeling serve to limit diffraction and thereby extend the interaction length of the device. However, these mechanisms are inherently nonlinear; one would prefer not to rely on nonlinear processes for device operation.

Guiding in a pre-formed density channel gives the option to operate in a linear regime, since such guiding is not dependent on the laser intensity. The original theoretical investigation of guiding in channels, which considered the case of parabolic channels, was made by Sprangle and co-workers [2]. The field structure in a parabolic channel is similar to that in a homogeneous plasma; *i.e.*, the fields

have an electrostatic character and the transverse field profile is determined by the driver profile. In contrast, the fields in a hollow channel are electromagnetic with the accelerating gradient being transversely uniform *independent* of the transverse profile of the driver and, to lowest order, the focusing fields are weak and linear. Since the original experimental work by Milchberg [3] numerous results in channels have been reported by many labs worldwide. Now that the experimental techniques for the controlled creation of channels is emerging, it is timely to begin systematic studies of the accelerating characteristics of plasma channels.

A consequence of the central advantage of a plasma accelerating structure, namely the ability to support gradients that would result in electrical breakdown in a metallic structure, is that the “wall” of the structure can exhibit a complex dynamics that must be faithfully modeled in order to determine the electrical properties of the structure.

## 2 CHARACTERIZING PLASMA CHANNELS

An optimized design of a plasma based accelerating structure requires the investigation of a considerable parameter space. By characterizing plasma structures in terms of various figures-of-merit, one can explore this parameter space in a systematic and controlled way using these characteristics as guide-posts. Here we outline the beginnings of such a search. For preliminary studies, we have found two quantitative characteristics to be quite informative [4–6]:  $Q$  and  $[R/Q]$ . Parameterizing the time dependence of the accelerating field as  $E_z \sim \exp[-(i\omega + \gamma)(t - z/c)]$ , we define the quality factor of the cavity,  $Q$ , as

$$Q = \frac{\omega}{2\gamma},$$

which determines the number of electron bunches that can be accelerated. This is of interest for both reasons of efficiency as well for constraints imposed by applications (*e.g.*, interaction region physics issues in a collider).

Following conventional resonator theory, we define the figure of merit  $[R/Q]$  as

$$\left[ \frac{R}{Q} \right] \equiv \frac{E_z(0)^2}{\omega_m \mathcal{U}_m} \sim k_p Z_0,$$

where  $E_z(0)$  is the peak accelerating gradient on axis,  $\omega_m$  is the mode frequency,  $\mathcal{U}_m$  is the mode energy per unit length,  $k_p$  is the plasma wavenumber, and  $Z_0$  is the impedance of free space. (The last relation follows from dimensional analysis.) For the fundamental mode,  $[R/Q]$  characterizes the energy spread imparted to the accelerated beam, whereas, for the higher-order modes,  $[R/Q]$  characterizes beams instabilities [6]. In more pragmatic terms,

\* Supported by the U. S. DoE under grant No. PDDEFG-03-95ER-40936.

$[R/Q]$  can be viewed as a measure of the gradient achieved ( $E_z(0)$ ) for the energy invested in the mode ( $\mathcal{U}_m$ ).

### 3 THE BASIC MODEL

We model the plasma as relativistic, cold, fluid electrons with a neutralizing, immobile ionic background. In addition, we make the following approximations: all field quantities are assumed to be quasi-static, *i.e.*, they are assumed to depend only upon  $\xi = t - z/c$ ; the laser-plasma interaction is modeled ponderomotively; and the assumption of slab geometry. Given these approximations, the fluid momentum balance equations become

$$\left(1 - \frac{v_z}{c}\right) \frac{\partial p_x}{\partial \xi} + v_x \frac{\partial p_x}{\partial x} = q \left( E_x - \frac{v_z}{c} B_y - mc^2 \frac{\partial \gamma_L}{\partial x} \right),$$

and

$$\left(1 - \frac{v_z}{c}\right) \frac{\partial p_z}{\partial \xi} + v_x \frac{\partial p_z}{\partial x} = q \left( E_z + \frac{v_x}{c} B_y + mc \frac{\partial \gamma_L}{\partial \xi} \right),$$

where  $\gamma_L(x, \xi) = \sqrt{1 + a^2}$  and  $a(x, \xi)$  is the dimensionless vector potential of the incident laser. This dimensionless vector potential is related to the vector potential by

$$\mathbf{A}_L = \sqrt{2} \frac{mc^2}{q} a(x, \xi) \cos(\omega_0 \xi).$$

Here the fundamental variables are  $p_x$  and  $p_z$ ;  $v_x$  and  $v_z$  are only a shorthand notation for  $p_x/(m\gamma)$  and  $p_z/(m\gamma)$ , respectively. The electron density,  $n_e$ , satisfies the continuity equation:

$$\frac{\partial n_e}{\partial \xi} + \frac{\partial v_x n_e}{\partial x} - \frac{1}{c} \frac{\partial v_z n_e}{\partial \xi} = 0.$$

The electromagnetic fields are determined by Maxwell's equations coupled to the fluid current density ( $qn_e \mathbf{v}$ ) and charge density ( $qn_e$ ) *viz.*,

$$\frac{\partial B_y}{\partial \xi} - \frac{\partial E_x}{\partial \xi} - c \frac{\partial E_z}{\partial x} = 0, \quad (1)$$

$$\frac{\partial E_x}{\partial \xi} - \frac{\partial B_y}{\partial \xi} + \frac{4\pi q}{m} n_e v_x = 0, \quad (2)$$

$$\frac{\partial E_z}{\partial \xi} - c \frac{\partial B_y}{\partial x} + \frac{4\pi q}{m} n_e v_z = 0,$$

$$4\pi q(n_e - n_i) - \frac{\partial E_x}{\partial x} + \frac{1}{c} \frac{\partial E_z}{\partial \xi} = 0,$$

where  $n_i$  is the background ion density. Note that through Ampere's law, Poisson's equation and the continuity equation are equivalent and thus only one of these is required. Computationally, when solving the full non-linear system, it is advantageous to use Poisson's equation in place of the continuity equation as the former is linear whereas the latter is non-linear. In the absence of the laser driver, this system possess an invariant,

$$\begin{aligned} \mathcal{E} = mc^2 \int dx n_e (\gamma - 1) \left(1 - \frac{v_z}{c}\right) \\ + \frac{1}{8\pi} \int dx \left[ E_z^2 + (E_x - B_y)^2 \right], \end{aligned}$$

which expresses energy balance in the moving frame. (As we will see below this interpretation is more apparent in the linearized theory.)

Numerically, it is desirable to solve the model equations in dimensionless form. Let  $n_0$  be a characteristic plasma density; for example, the plateau value at large transverse distance from the channel. The dimensionless variables used for computation,  $\hat{\xi}$ ,  $\hat{x}$ ,  $\hat{n}_e$ ,  $\hat{n}_i$ ,  $\hat{\mathbf{p}}$ ,  $\hat{\mathbf{E}}$  and  $\hat{\mathbf{B}}$ , are then defined by:  $\omega_p \hat{\xi} = \hat{\xi}$ ,  $k_p x = \hat{x}$ ,  $n_e = n_0 \hat{n}_e$ ,  $n_i = n_0 \hat{n}_i$ ,  $\mathbf{p} = mc \hat{\mathbf{p}}$ ,  $\mathbf{E} = (mc/q) \omega_p \hat{\mathbf{E}}$ , and  $\mathbf{B} = (mc/q) \omega_p \hat{\mathbf{B}}$ . The plasma frequency,  $\omega_p$ , is related to  $n_0$  in the usual way:  $\omega_p^2 = (4\pi q^2/m) n_0$  and  $k_p = \omega_p/c$ .

### 4 LINEAR THEORY

A significant advantage of the pre-formed channel is its ability to guide a laser pulse in the *linear* regime as contrasted with the non-linear processes responsible for guiding in an initially homogeneous plasma. Motivated by the desire to operate in this linear regime, we consider the linearized fluid equations and study linear wake excitation. The linearized momentum equations are

$$\frac{\partial v_x}{\partial \xi} = E_x - \frac{1}{2} \frac{\partial a^2}{\partial x} \quad \text{and} \quad \frac{\partial v_z}{\partial \xi} = E_z + \frac{1}{2} \frac{\partial a^2}{\partial \xi},$$

where we have expanded  $\gamma_L$  under the assumption of small  $a$ , which is in keeping with the premise of linear theory. In the dimensionless system, linearly there is no distinction between  $\mathbf{p}$  and  $\mathbf{v}$ . Here we are linearizing about the quiescent state, so the density appearing in Poisson's equation is that of the perturbation. Thus, the perturbed density only couples through Poisson's equation and is therefore completely determined by the electric field. Let  $n_e^{(0)}$  denote the equilibrium electron density. The linearized energy invariant

$$\begin{aligned} \mathcal{E} = \frac{1}{2} m \int dx n_e^{(0)} (v_x^2 + v_z^2) \\ + \frac{1}{8\pi} \int dx \left[ E_z^2 + (E_x - B_y)^2 \right] \quad (3) \end{aligned}$$

can be understood in terms of Poynting's theorem in the quasi-static approximation:

$$\frac{\partial}{\partial \xi} \left[ \frac{1}{2} n_e^{(0)} v^2 + \frac{1}{8\pi} (E^2 + B^2) \right] + \nabla \cdot \mathbf{S} = 0,$$

where the components of the Poynting flux,  $\mathbf{S}$ , are given by

$$S_x = -\frac{1}{4\pi} B_y E_x, \quad \text{and} \quad S_z = \frac{1}{4\pi} B_y E_x.$$

Combining these pieces, Poynting's theorem becomes

$$\begin{aligned} \frac{\partial}{\partial \xi} \left[ \frac{1}{2} n_e^{(0)} (v_x^2 + v_z^2) + \frac{1}{8\pi} (E^2 + B^2 - 2E_x B_y) \right] \\ - \frac{1}{4\pi} \frac{\partial B_y E_x}{\partial x} = 0. \end{aligned}$$

Integrating this expression over  $x$  yields  $\partial \mathcal{E} / \partial \xi = 0$ . Hence, the invariance of  $\mathcal{E}$  simply expresses conservation of energy for each  $\xi$ -slice.

## 5 COMPUTATIONAL METHODS

For both the linear and non-linear models, we use the Crank-Nicholson technique to discretize the relevant equations. Although this method is implicit, it has three significant advantages: it is (linearly) unconditionally stable, exhibits no amplitude dissipation, and is second-order in both  $x$  and  $\xi$ . These characteristics allow for a large ratio of the  $\xi$  to  $x$  step sizes keeping execution time down even for runs with fine spatial resolution. We take a uniform grid in both  $x$  and  $\xi$ . Let  $f_j^n = f(\xi_n, x_j)$  and let  $\Delta\xi$  and  $\Delta x$  be the  $x$  and  $\xi$  grid spacings, respectively. In this method, the equations are discretized at  $(\xi_{n+1/2}, x_j)$ , i.e., between the  $\xi$  grid-points. The  $\xi$ -derivatives are approximated as

$$\frac{\partial f}{\partial \xi}(\xi_{n+1/2}, x_j) = \frac{f_j^{n+1} - f_j^n}{\Delta\xi} + \mathcal{O}(\Delta\xi^2).$$

All other quantities are evaluated at  $\xi_{n+1/2}$  by averaging their values at  $\xi_n$  and  $\xi_{n+1}$ , viz.,

$$\begin{aligned} \frac{\partial f}{\partial x}(\xi_{n+1/2}, x_j) &= \frac{f_{j+1}^{n+1} - f_{j-1}^{n+1} + f_{j-1}^n - f_{j+1}^n}{4\Delta x} \\ &\quad + \mathcal{O}(\Delta\xi^2 + \Delta x^2), \end{aligned}$$

and

$$f(\xi_{n+1/2}, x_j) = \frac{1}{2} (f_j^{n+1} + f_j^n) + \mathcal{O}(\Delta\xi^2).$$

Discretizing the equations between the  $\xi$  grid-points yields a finite-difference approximation that is second order in  $\Delta\xi$  while only using information from *two*  $\xi$  steps. It this latter property that is responsible for the unconditional stability. Not having  $\Delta\xi$  linked to  $\Delta x$  by a Courant condition turns out to be extremely valuable; in the linear case, we are able to obtain accurate solutions even when  $\Delta\xi/\Delta x$  is quite large ( $\sim 20$ ), significantly reducing both storage requirements and computing time. Explicitly, the Crank-Nicholson discretization of the linear equations is

$$\begin{aligned} v_x^{n+1} - v_x^n - \frac{\Delta\xi}{2} (E_x^{n+1} + E_x^n) &= 0, \\ v_z^{n+1} - v_z^n - \frac{\Delta\xi}{2} (E_z^{n+1} + E_z^n) &= 0, \\ B_y^{n+1} - B_y^n - E_x^{n+1} + E_x^n \\ - \frac{\Delta\xi}{2\Delta x} (E_{z_{j+1}}^{n+1} + E_{z_{j+1}}^n - E_{z_{j-1}}^{n+1} - E_{z_{j-1}}^n) &= 0, \\ E_x^{n+1} - E_x^n - B_y^{n+1} + B_y^n + \\ \frac{\Delta\xi}{2} n_e^{(0)}(x_j) (v_x^{n+1} + v_x^n) &= 0, \\ E_z^{n+1} - E_z^n + \frac{\Delta\xi}{2} n_e^{(0)}(x_j) (v_z^{n+1} + v_z^n) \\ - \frac{\Delta\xi}{2\Delta x} (B_{y_{j+1}}^{n+1} + B_{y_{j+1}}^n - B_{y_{j-1}}^{n+1} - B_{y_{j-1}}^n) &= 0. \end{aligned}$$

Provided that the  $x$ -grid extends a reasonable distance from the channel (so that the transverse gradients of the

fields are negligible at the computational boundary), this discretization will honor the invariance of  $\mathcal{E}$  nearly to machine precision *independent* of the size of  $\Delta\xi$ . While this can be demonstrated explicitly, it can be more simply understood. Consider first discretizing the equations in  $x$  using, say, central differences. This will yield a (large) set of coupled ODE's (in  $\xi$ ) which we then solve using the mid-point rule. This procedure produces the same discretization as the Crank-Nicholson method. It is well-known that the mid-point rule will preserve exactly quadratic invariants. Hence solving these ODE's with the mid-point rule (or equivalently, solving the original PDEs with the Crank-Nicholson method) will preserve any quadratic invariants possessed by the system. Here there is a slight complication due to the transverse integration in (3) resulting in  $\mathcal{E}$  being only an approximate invariant of the spatially discrete system, which is why the preservation of  $\mathcal{E}$  is only nearly exact.

We take a somewhat novel approach to constructing the simulation code. Proceeding from a high-level symbolic description of the PDE's, a custom code generator produces the necessary C++ source to implement the PDE solver. Since the discretized equations are implicit, at its core, the PDE solver must solve a large block-tri-diagonal linear system. The linear system solver is implemented using templates and a traits-like mechanism is used to specify the details of the linear system corresponding to the PDE. The code generator essentially produces a specialization of the traits class describing the specific linear system. This approach strikes balance between complexity of code generator and demands placed on the compiler while still producing code equal in quality to the best hand-optimized code. This approach makes possible optimizations that would be tedious and error prone if carried out by hand. The overall effect is to raise the level of abstraction of the implementation of the PDE solver [7].

## 6 LINEAR RESPONSE

In the linear case, we are fundamentally interested in the plasma response to a ponderomotive impulse as it encapsulates the entire plasma behavior. By integrating the equations of motion analytically from  $\xi = -\varepsilon$  to  $\xi = \varepsilon$  and taking the limit  $\varepsilon \rightarrow 0$  we can convert a ponderomotive impulse into initial conditions for the homogeneous equations. We assume a quiescent state for  $\xi < 0$  and apply a ponderomotive impulse at  $\xi = 0$ . With this driver, the momentum equations become:

$$\frac{\partial v_x}{\partial \xi} = E_x - \frac{1}{2} \frac{\partial a^2(x)}{\partial x} \delta(\xi),$$

and

$$\frac{\partial v_z}{\partial \xi} = E_z + \frac{1}{2} a^2(x) \delta'(\xi).$$

Integrating these equations from  $\xi = -\varepsilon$  to  $\xi = \varepsilon$  we find

$$v_x(\varepsilon, x) = \int_0^\varepsilon d\xi E_x(\xi, x) - \frac{1}{2} \frac{\partial a^2(x)}{\partial x},$$

## 7 HOLLOW CHANNELS

In the ideal hollow channel,  $\omega_p(x) = \omega_{p0}\theta(|x| - b)$ , where  $\theta$  is the Heavyside function. Here it is possible, in the context of linear theory, to obtain an analytical expression for the wake field [9]. In this case, one finds that the channel mode has infinite  $Q$  and oscillates at a frequency,  $\omega_{ch}$ , which is less than  $\omega_{p0}$ . For the non-ideal case, *i.e.*, where the channel walls are not infinitely steep, matters are more complicated. The dielectric function,  $\epsilon = 1 - \omega_p^2(x)/\omega^2$ , is evidently spatially dependent and every  $\omega < \omega_{p0}$  is resonant with the local plasma frequency at some location in the wall. This resonant layer leads to absorption of the wake yielding a low  $Q$ . This, in turn, means a large spread in frequency space about  $\omega_{ch}$ , exciting much of the wall. To examine these effects in detail, we parameterize the equilibrium density as shown in Fig. 2 and assume a ponderomotive impulse with a transverse pulse shape of the form

$$a^2(x) = a_0^2 e^{-2(x_c/w_x)^2}.$$

As an example of the effects of the resonant absorption, consider a channel with  $k_p b = 2$  and  $\alpha = 17^\circ$ . The wake field, shown in Fig. 3, has  $Q \cong 7$ . Resonant absorption transfers energy from the wake field to particle kinetic energy in the region of the channel wall. The velocity fields (currents) are organized in such a way that the corresponding electromagnetic fields are also spatially localized. This process results in the development of fine-scale spatial structure in the velocity and electric fields which requires high numerical resolution in the simulations.

In practice, since the laser driver extends into the bulk plasma, an electrostatic bulk mode will be excited in addition to the electromagnetic channel mode. Furthermore, separating these modes is not a simple matter, thus it is both convenient and reasonable to define an effective  $[R/Q]$  as  $[R/Q]_{\text{eff}} = E_z(0)^2/(\omega_m \mathcal{U}_T)$ , where  $\mathcal{U}_T$  is the total energy in the plasma. As a result of replacing  $\mathcal{U}_m$  with  $\mathcal{U}_T$ , the energy spent exciting the unwanted bulk mode will be reflected in a reduction in  $[R/Q]_{\text{eff}}$ . This effect can be seen in Fig. 4, where we plot  $[R/Q]_{\text{eff}}$  vs. driver width keeping  $a_0^2$  and  $k_p b$  constant. (We normalize  $[R/Q]_{\text{eff}}$  to  $k_p^2 Z_0$ .)

The physical interpretation of this result is straightforward. The channel mode is supported by surface currents in the channel walls so a very narrow driver excites the wall only a small amount, yielding a low  $E_z(0)$ . Conversely, a wide driver excites the wall significantly but the expo-

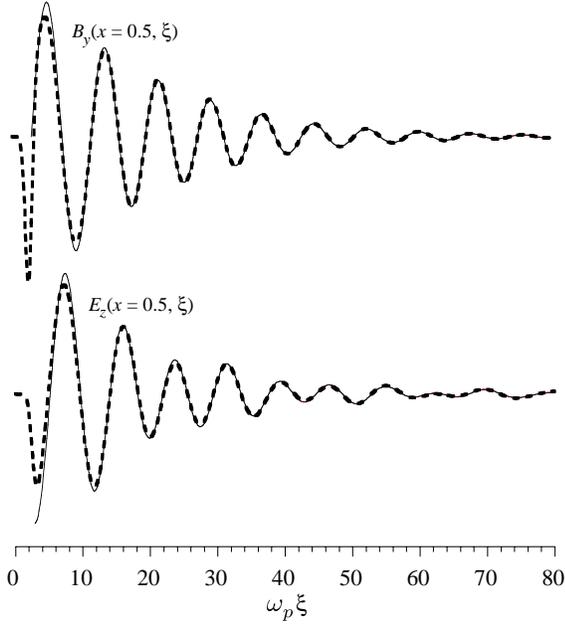


Figure 1: Comparison of the ponderomotively driven case (thick dashed line) and the response function generated from initial conditions (thin solid line) as described in the text. Here the density profile is  $n_e^{(0)}(x) = (k_p x)^2$  and the driver pulse shape is  $\exp(-2(\omega_p \xi - 2.5)^2)$ . The fields are plotted at a transverse location of  $k_p x = 0.5$ .

and

$$v_z(\varepsilon, x) = \int_0^\varepsilon d\xi E_z(\xi, x).$$

We expect the fields to be smooth functions for  $\xi > 0$ , so we may approximate the integrals as

$$\int_0^\varepsilon d\xi f(\xi) = \frac{\varepsilon}{2} [f(\varepsilon) + f(0^+)] + \mathcal{O}(\varepsilon^3).$$

Taking the limit  $\varepsilon \rightarrow 0^+$ , gives

$$v_x(0^+, x) = -\frac{\partial a^2(x)}{\partial x} \quad \text{and} \quad v_z(0^+, x) = 0.$$

In the same way, (1) and (2) lead to  $E_x(0^+, x) = 0$  and  $B_y(0^+, x) = 0$ . Lastly, by adding (1) and (2), and taking  $\varepsilon \rightarrow 0^+$ , we have

$$\frac{\partial E_z}{\partial x}(0^+, x) = n_e^{(0)}(x) v_x(0^+, x),$$

which must then be solved to obtain  $E_z(0^+, x)$ . In Fig. 1, we compare the plasma response to a ponderomotive pulse having a Gaussian envelope in  $\xi$  with the response obtained using these initial conditions. Other than the initial transient (and an overall normalization), the  $\xi$  behavior of the two solutions is identical. This allows for computation of the Green's function without having to deconvolve the driver envelope or to numerically approximate a  $\delta$ -function. Amongst other uses, ready access to the channel Green's function allows for efficient optimization studies [8].

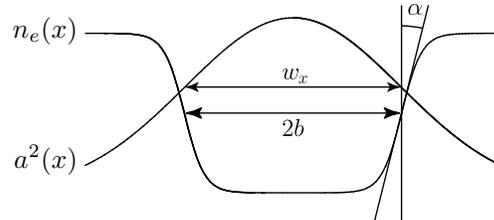


Figure 2: Density and transverse ponderomotive profiles.

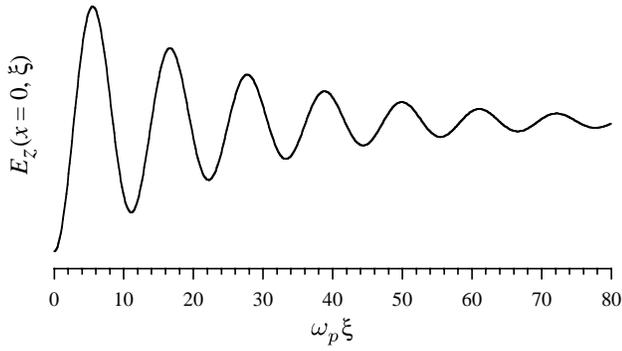


Figure 3: Longitudinal wake field on axis for a channel with  $k_p b = 2$  and  $\alpha = 17^\circ$ .

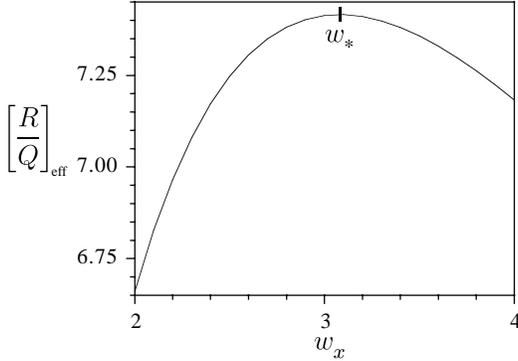


Figure 4: Effect of transverse driver width upon  $[R/Q]_{\text{eff}}$ . The optimal value of  $w_x$  is denoted by  $w_*$ . Here  $k_p b = 2.0$  and  $\tan \alpha = 0.3$

nential “wings” also excite a large bulk mode which contributes to  $\mathcal{U}_T$  but not  $E_z(0)$ . Hence, there is an optimal driver width,  $w_*$ , which balances these competing effects. The effect of the slope of the channel walls is shown in Fig. 5. Clearly,  $Q$  is strongly dependent on this slope. As the walls are made ever less steep, the size of the resonant region grows allowing faster transfer of energy from the wake into the wall region, yielding a lower  $Q$ .

## 8 CONCLUSIONS

We have begun the systematic study of the accelerating properties of plasma channels by considering  $Q$  and  $[R/Q]$ . These figures-of-merit allow for a well defined optimization of plasma based accelerating structures. Our results are clearly preliminary; we have considered only some of the relevant parameters. In particular we have ignored the constraint on the driver width imposed by the guiding condition which may well require an operating point that differs from the optimal width determined from  $[R/Q]$  considerations alone. Additionally, laboratory channels are unlikely to be completely hollow. In such channels, the base density supports an electrostatic mode in addition to the electromagnetic mode, altering the desirable beam transport properties of the hollow channel and also limiting (*via* wave breaking) the accelerating gradient that can be sup-

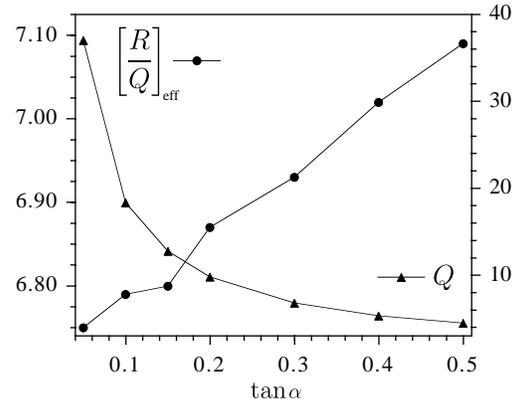


Figure 5: The effect of the wall inclination on  $Q$  and  $[R/Q]_{\text{eff}}$  with optimal driver width. Here  $k_p b = 2.5$

ported. This suggests that transverse pulse shaping (*i.e.*, a higher-order gaussian spatial mode) may prove important in the design of successful hollow channel accelerating structures.

## 9 REFERENCES

- [1] E. Esarey, P. Sprangle, J. Krall, and A. Ting, “Overview of Plasma-Based Accelerator Concepts,” *IEEE Trans. Plasma Sci.* **24**, 252 (1996).
- [2] P. Sprangle, C. M. Tang, and E. Esarey, “Relativistic Self-Focusing of Short-Pulse Radiation Beams in Plasmas,” *IEEE Trans. Plasma Sci.* **PS-15**, 145 (1987).
- [3] C. G. Durfee III and H. M. Milchberg, “Light Pipe for High Intensity Laser Pulses,” *Phys. Rev. Lett.* **71**, 2409 (1993).
- [4] B. A. Shadwick and J. S. Wurtele, “Plasma Channels as Accelerating Structures,” *Bull. Am. Phys. Soc.* **43**, 1123 (1998).
- [5] B. A. Shadwick and J. S. Wurtele, “Numerical Studies of Wake Excitation in Plasma Channels,” in *Proceedings of the Sixth European Particle Accelerator Conference*, Stockholm (IOP, Bristol, 1998), pp. 827–829.
- [6] C. B. Schroeder, J. S. Wurtele, and D. H. Whittum, “Multi-mode Analysis of the Hollow Plasma Channel Accelerator,” *Phys. Rev. Lett.* **82**, 1177 (1999).
- [7] B. A. Shadwick, “The role of automatic code generation in scientific computing,” (1999), in preparation.
- [8] A. Charman, B. A. Shadwick, and J. S. Wurtele, “Optimal Pulse-Shaping in the Laser Wakefield Accelerator,” (1999), in preparation.
- [9] T. C. Chiou, T. Katsouleas, C. Decker, W. B. Mori, G. Shvets, and J. S. Wurtele, “Laser Wake-Field Acceleration and Optical Guiding in a Hollow Plasma Channel,” *Phys. Plasmas* **2**, 310 (1993).

# BEAM-BEAM SIMULATIONS WITH THE GAUSSIAN CODE TRS

Miguel A. Furman\*†

Lawrence Berkeley National Laboratory, MS 71-259  
Berkeley, CA 94720, USA

Abstract

We describe features of the soft-gaussian beam-beam simulation code “TRS” and present sample results for the PEP-II  $e^+e^-$  collider.

## 1 DESCRIPTION OF THE CODE

A basic experimental observation in  $e^+e^-$  colliders in stable operation is that the particle distribution density at the beam core is approximately gaussian, while the density at large amplitudes (a few  $\sigma$ 's away from the center) is not gaussian and is much larger than the extrapolation from a gaussian fit to the core [1, 2].

The code TRS (Two-Ring Simulation) [3] is geared to study the beam core of colliding  $e^+e^-$  beams. Although it can be used to study large-amplitude tail distributions, it is very inefficient at doing so, since the vast majority of the CPU time is spent simulating the gaussian core. The “engine” of this code is similar to that in other codes [4, 5, 6, 7]. The code is written in FORTRAN 77, and is yet to be documented in detail.

### 1.1 Simulation technique

In the simplest case each beam is represented by a single bunch traveling in a separate, distinct ring and collisions occur at only one interaction point (IP). The basic simulation technique consists in tracking a given number (typically 1000–50000) of macroparticles per bunch and computing, at every turn just before the collision, the centroids  $\langle x \rangle_{\pm}$ ,  $\langle y \rangle_{\pm}$  and rms widths  $\sigma_{x\pm}$ ,  $\sigma_{y\pm}$  of the distributions, where the subscript  $+$ ( $-$ ) refers to the  $e^+$ ( $e^-$ ) beam. For the purposes of computing the beam-beam interaction, the code assumes that the transverse distribution of the kicking bunch is gaussian, using the just-computed values of  $\langle x \rangle$ ,  $\langle y \rangle$ ,  $\sigma_x$  and  $\sigma_y$  in the Bassetti-Erskine formula [8] for the electromagnetic field of a gaussian distribution. This formula is then used to compute the kick on every macroparticle of the opposing bunch. The role of the two colliding bunches is then reversed, completing the computation of the beam-beam interaction.

Each macroparticle is then tracked through its corresponding ring lattice, and the process is iterated for many turns, typically corresponding to 3–5 damping times. An aperture “lattice element” intercepts par-

ticles at large amplitude, and these are removed from the simulation.

The main output of the program is a file the with turn-by-turn values of  $\langle x \rangle_{\pm}$ ,  $\langle y \rangle_{\pm}$ ,  $\sigma_{x\pm}$ ,  $\sigma_{y\pm}$  and the remaining number of macroparticles. Simple post-processors can then compute the luminosity and the frequency spectra of  $\langle x \rangle_{\pm}$ ,  $\langle y \rangle_{\pm}$ ,  $\sigma_{x\pm}$  and  $\sigma_{y\pm}$ . The program can also output the  $x$  and  $y$  projections of the time-averaged macroparticle distributions in binned form.

### 1.2 Other features

Nonzero bunch-length effects are taken into account by slicing the bunch longitudinally, so that the beam-beam interaction is represented by several kicks [9] with prescribed locations and weights. In the simplest case (thin-lens approximation) there is a single kick at the center of the bunch. Typically, however, one uses a thick-lens approximation with 5 or more kicks. In between two consecutive kicks, the macroparticles undergo simple drifts as they pass through the opposing bunch. The program only allows for head-on collisions (zero crossing angle), although the centers of the beams are allowed to be offset.

The particle kinematics is fully 6-dimensional; however, the longitudinal component of the beam-beam kick is wholly ignored, which is typically a good approximation. The synchrotron motion of the particles can be chosen to be parametric, i.e. an exact harmonic rotation with a specified synchrotron tune, or can be implemented with an RF cavity kick plus a time-of-flight “lattice element” which is a function of the (specified) momentum compaction factor.

The parameters of the two beams and the two rings are fully independent of each other. The ring lattices can contain arbitrary nonlinear thin elements, apertures, or linear phase-advance maps. In typical applications, however, the entire ring is represented by a simple linear map with specified tunes plus a rectangular aperture.

Radiation damping and quantum excitation are also represented by simple kick elements that typically act once per turn. These elements are constructed so that, in the absence of the beam-beam interaction, the rms beam sizes  $\sigma_{x\pm}$  and  $\sigma_{y\pm}$  would reach specified values  $\sigma_{0x\pm}$  and  $\sigma_{0y\pm}$  after many damping times, regardless of the initial conditions.

Parasitic collisions can be included by an appropriate lattice element whose strength depends on the pa-

---

\* Work supported by the US Department of Energy under contract No. DE-AC03-76SF00098.

† mafurman@lbl.gov

rameters of the opposing bunch.

The code can run in “weak-strong” mode, “strong-strong” mode, or single-particle tracking mode. In the weak-strong mode, one beam (the strong beam) is represented by a static gaussian lens (thin or thick) with specified, unchanging,  $\sigma$ 's, and the other beam (the dynamical, or weak, beam) by a collection of dynamical macroparticles. In strong-strong mode both beams are represented by macroparticles whose distributions vary dynamically under their mutual beam-beam interaction. Single-particle tracking mode is the same as weak-strong mode in which the weak beam consists of a single particle. In this case the output from the program is the turn-by-turn phase space of this particle. This mode is used to study single-particle resonance effects and to produce beam-beam footprints; it has also proven valuable in debugging the code and in allowing for basic comparisons with other codes and with analytic results.

The code also offers choices of algorithms for the computation of the complex error function, which enters the Bassetti-Erskine formula. Typically we use a 3rd-order table interpolation [10], but one can also use a 4th-order interpolation, a Padé approximant [11], or the IMSL<sup>®</sup> function CERFE [12]. Similarly, the program offers the choice of several slicing algorithms to assign the locations and weights of the kicks representing the long-bunch effects, and also two algorithms for the computation of radiation damping and quantum excitation effects.

### 1.3 Tests

The program has been systematically tested against analytic results in single-particle mode, and against other similar simulation codes in their common region of applicability [13, 14]. In single-particle mode, the excellent agreement with analytic results of the amplitude-dependent tune shift validates the basic beam-beam force computation. In weak-strong mode, minor disagreements with other codes can be attributed to differences in details of the codes other than the beam-beam computation.

### 1.4 Speed

In the typical case when the lattice is represented by a linear map, the CPU time used by the program is dominated by the beam-beam computation. For a given run, the CPU time scales according to

$$\text{CPU time} \propto (\text{no. of macroparticles/bunch}) \times (\text{no. of slices}) \times (\text{no. of turns}) \quad (1)$$

A rough idea of the program speed is obtained from informal benchmarks on three computers: in units of CPU-sec/(macroparticle $\times$ slice $\times$ turn), the speed (or, rather, the inverse speed) is  $2.1 \times 10^{-5}$ ,  $1.3 \times 10^{-5}$  and  $8.7 \times 10^{-5}$  on NERSC's C90 Cray computer, on

NERSC's T3E computer “mcurie” in single-processor mode, and on a Sun SPARCstation 20, respectively. These numbers assume that the beams collide once per turn, that both are represented by the same number of macroparticles, and that the complex error function is computed via a 3rd-order table interpolation algorithm.

The above speeds are for strong-strong mode; in weak-strong mode, the program runs roughly twice as fast, as it should be expected.

### 1.5 Drawbacks

Although the gaussian approximation has the advantage of simplicity, its accuracy relies on the assumption that the core of the distribution is gaussian. However, it should be kept in mind that, in certain cases, depending on the values of the tunes and the beam-beam parameters, coherent single-bunch resonances can appear that distort the core distribution significantly away from the gaussian shape. In this case, obviously, the Bassetti-Erskine formula is not expected to be reliable. A more adequate solution is provided by a PIC code in which the electromagnetic kick is computed from the actual macroparticle distribution [15, 16].

Nevertheless, the gaussian approximation is reliable in many cases of practical interest. And, as shown in the examples below for PEP-II, even if the conditions are such that a coherent resonance appears, the gaussian approximation behaves qualitatively differently from the “normal” case, providing a signal that the gaussian approximation should be suspect.

## 2 APPLICATION TO PEP-II

The code TRS has been applied to the PEP-II collider to obtain tune scans whose primary goal is to establish areas in the tune plane with acceptable luminosity performance [17]. For the purposes of this article we use a model of the machine in which the rings are represented by linear arcs of specified tunes, and parasitic collisions are ignored. Table 1 provides a basic parameter list used in the simulations.  $E$  is the nominal beam energy,  $N$  is the number of particles per bunch, the  $\xi_0$ 's are the nominal beam-beam parameters, the  $\beta^*$ 's and  $\sigma_0^*$ 's are the optics functions and nominal beam sizes at the interaction point (IP), respectively, the  $\tau$ 's are the damping times, and the  $\nu_s$ 's are the synchrotron tunes.

Fig. 1 shows the time evolution of the normalized rms beam sizes for the tune values shown. The simulation was run for 25000 turns, i.e.,  $\sim 3 e^+$  damping times, with 50000 macroparticles per bunch using 5 kicks for the bunch length effects. The beam sizes start out being smaller than the nominal sizes owing to the dynamical  $\beta$  effect [18], and the beam blowup behavior is typical of incoherent resonance effects.

Table 1: Basic PEP-II parameters.

	$e^+$	$e^-$
$E$ [GeV]	3.1	9.0
$N$ [ $10^{10}$ ]	5.685	1.958
$\xi_{0x}$	0.03	0.03
$\xi_{0y}$	0.03	0.03
$\beta_x^*$ [cm]	50	50
$\beta_y^*$ [cm]	1.5	1.5
$\sigma_{0x}^*$ [ $\mu\text{m}$ ]	153.0	153.0
$\sigma_{0y}^*$ [ $\mu\text{m}$ ]	4.591	4.591
$\sigma_z$ [cm]	1.0	1.0
$\nu_s$	0.0334	0.0521
$\tau_x = \tau_y$ [turns]	8366	5014

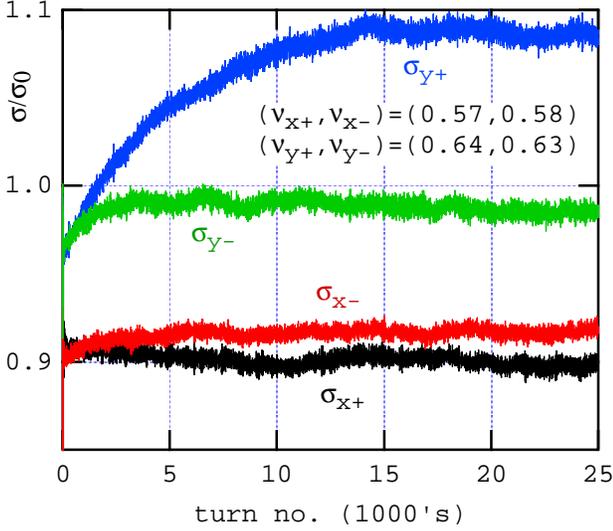


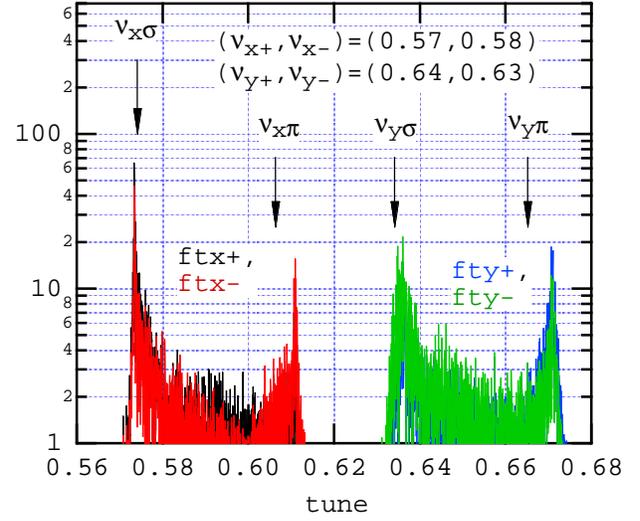
Figure 1: Time evolution of the normalized rms beam sizes for split tunes.

Fig. 2 shows the spectra, in absolute value, of the horizontal and vertical oscillations of the beam centroid for both beams. The arrows indicate the tunes of the  $\sigma$  and  $\pi$  modes computed in the rigid-gaussian small-amplitude approximation, given by [19]

$$\cos \left\{ \frac{2\pi\nu_\pi}{2\pi\nu_\sigma} \right\} = \frac{1}{2}(C_+ + C_-) - \pi(S_+\Xi_+ + S_-\Xi_-) \pm \sqrt{R}, \quad (2)$$

$$R \equiv \frac{1}{4}(C_+ - C_-)^2 + \pi^2(S_+\Xi_+ + S_-\Xi_-)^2 - \pi(C_+ - C_-)(S_+\Xi_+ - S_-\Xi_-) \quad (3)$$

Here the subscript  $+$ ( $-$ ) refers to the  $e^+$ ( $e^-$ ) beam,  $C_\pm \equiv \cos 2\pi\nu_\pm$ ,  $S_\pm \equiv \sin 2\pi\nu_\pm$ , the  $\nu$ 's are the tunes and the  $\Xi$ 's are the coherent beam-beam parameters [20]. For example, the horizontal coherent beam-beam


 Figure 2: Absolute value of the tune spectra, in arbitrary units, of the centroid motions of the two beams for split tunes. The spectra of the two beams almost exactly overlap. The arrows indicate the  $\sigma$  and  $\pi$  tunes computed from Eqs. (2-4).

parameter of the positron beam is

$$\Xi_{x+} = \frac{r_e N_- \beta_{x+}^*}{2\pi\gamma_+ \Sigma_x (\Sigma_x + \Sigma_y)} \quad (4)$$

with corresponding expressions for  $\Xi_{x-}$  and  $\Xi_{y\pm}$ . Here  $r_e$  is the classical electron radius,  $\gamma$  is the usual relativistic factor and  $\Sigma_x = (\sigma_{x+}^{*2} + \sigma_{x-}^{*2})^{1/2}$  with a similar expression for  $\Sigma_y$ .

The  $\sigma$  and  $\pi$  tunes shown by the arrows in Fig. 2 take into account the equilibrium beam sizes in Eq. (4), obtained from Fig. 1. The disagreement of  $\nu_\pi$  with the second peak of the simulated spectra can be explained by the fact that the bunches are not rigid but rather vary dynamically in shape and size [21, 22].

Fig. 3 shows the spectra of the rms beam sizes. The signals of the synchrotron tunes, labeled  $\nu_{s\pm}$ , are clearly seen. Simulations not shown here indicate that, when the tunes are slightly changed, the main peaks move at twice the rate of change of the tunes [23].

If the tunes are chosen pairwise equal, a qualitatively different result obtains. Fig. 4 shows the time evolution of the rms beam sizes for  $\nu_{x+} = \nu_{x-} = 0.57$  and  $\nu_{y+} = \nu_{y-} = 0.64$ . After a transition at  $\sim 5000$  turns, corresponding to the damping time of the  $e^-$  beam, the vertical beam sizes of the two beams are locked together and blow up to a larger value than in the split-tune case. Another feature of the dynamics is shown in Fig. 5, in which the normalized vertical beam sizes are plotted for only 20 consecutive turns. When the starting turn is 2000, the beam sizes are uncorrelated, while if the starting turn is 20000, the sizes are correlated and oscillate in phase with period 3. These behaviors are typical of coherent beam-beam resonances which

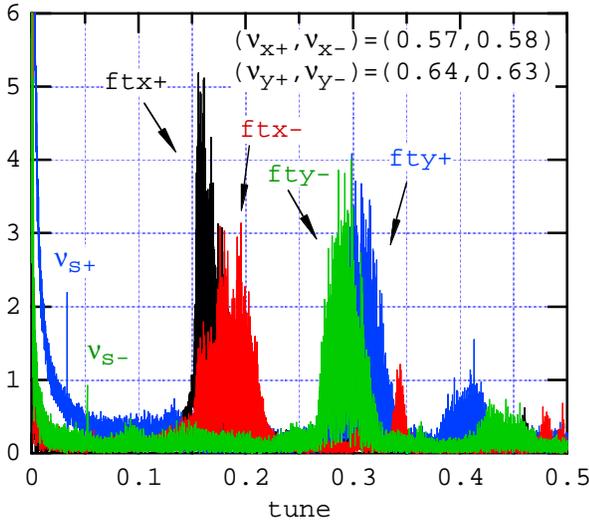


Figure 3: Absolute value of the rms beam size spectra in arbitrary units for split tunes.

can be studied in more quantitative detail by other means, as discussed in Sec. 1.5. In practice, a more realistic simulation of PEP-II requires the inclusion of the parasitic collisions near the IP [17]. These collisions, although relatively weak, are sufficiently strong to destroy the coherent resonance condition, and the actual behavior observed in the simulation is of the “normal” kind, i.e., dominated by incoherent effects, as in the split-tunes example discussed earlier.

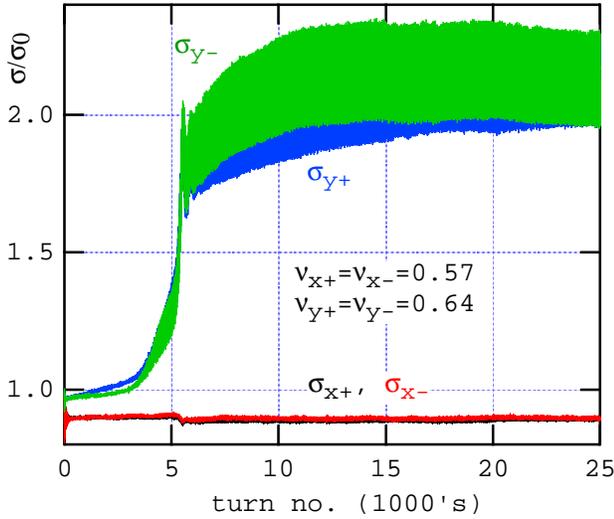


Figure 4: Time evolution of the normalized rms beam sizes for pairwise-equal tunes.

### 3 SUMMARY

We have summarized the main features of the beam-beam simulation code TRS and presented two sample applications to the PEP-II collider. The code has been

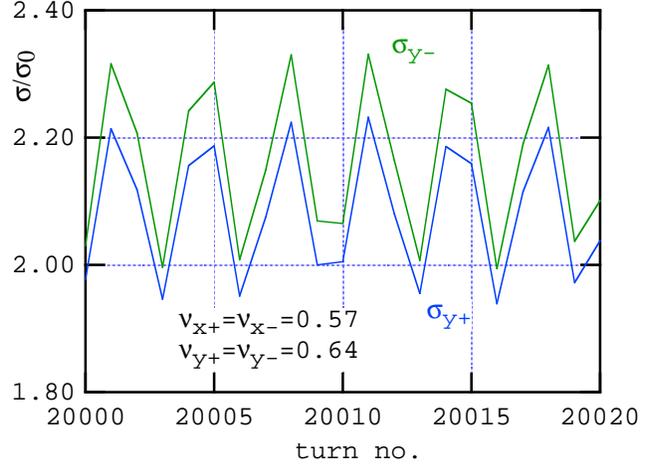
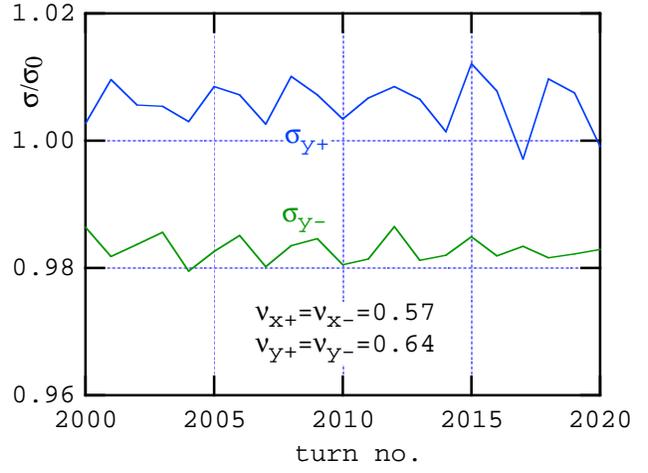


Figure 5: Detail from Fig. 4: time evolution for 20 consecutive turns of the normalized vertical rms beam sizes. Top: starting at turn 2000; bottom: starting at turn 20000.

successfully tested against analytic results and against other simulation codes whenever such comparisons are meaningful.

The soft-gaussian approximation is believed to represent reliably incoherent beam-beam effects. The code has been used to perform studies for the PEP-II collider. For example, simulated tune scans reveal undesirable operating points due to beam blowup from synchrotron sidebands. The dynamical beta effect, clearly seen in these simulations, also influences the choice of a working point. The code has been used to establish the adequate beam separation at the parasitic collision points [24], and has been applied to the proposed muon collider [25], including the effects from the instability of the muon.

In some cases the code clearly reveals coherent behavior; however, the soft-gaussian approximation is quantitatively unreliable in such cases, and other methods are called for.

Present improvement plans include allowing for non-

linear maps to better represent the machine lattice, and for collisions at a crossing angle.

#### 4 ACKNOWLEDGMENTS

I am grateful for many discussions and/or collaboration over time with E. Anderson, A. Chao, T. Chen, Y. H. Chin, J. Eden, E. Forest, W. Herr, K. Hirata, J. Irwin, S. Krishnagopal, E. Keil, R. Li, K. Y. Ng, D. Shatilov, R. Siemann, J. Tennyson, K. Yokoya and A. Zholents.

#### 5 REFERENCES

- [1] J. T. Seeman, "Observations of the Beam-Beam Interaction," Proc. Nonlinear Dynamics Aspects of Particle Accelerators, Sardinia, January 31–February 5, 1985, Springer Verlag Lecture Notes in Physics 247, p. 121.
- [2] D. Rice, "Observations of the Beam-Beam Effect in PEP, SPEAR and CESR," Proc. Third Advanced ICFA Beam Dynamics Workshop (Beam-Beam Effects in Circular Colliders), I. Koop and G. Tumaikin, eds., Novosibirsk, May 29–June 3, 1989, p. 17.
- [3] J. L. Tennyson, undocumented code "TRS," 1989.
- [4] S. Myers, "Review of Beam-Beam Simulations," Proc. Nonlinear Dynamics Aspects of Particle Accelerators, Sardinia, January 31–February 5, 1985, Springer Verlag Lecture Notes in Physics 247, p. 176.
- [5] K. Yokoya and Y. H. Chin, undocumented code.
- [6] K. Hirata, "Analysis of Beam-Beam Interactions with a Large Crossing Angle," Phys. Rev. Lett. 74, 2228 (1995).
- [7] E. B. Anderson and J. T. Rogers, "ODYSSEUS: A Dynamic Strong-Strong Beam-Beam Simulation," CBN 97-33, November 9, 1997, and these proceedings.
- [8] M. Bassetti and G. A. Erskine, "Closed Expression for the Electric Field of a Two-Dimensional Gaussian Charge," CERN-ISR-TH/80-06.
- [9] S. Krishnagopal and R. Siemann, "Bunch-Length Effects in the Beam-Beam Interaction," Phys. Rev. D41, p. 2312 (1990).
- [10] Handbook of Mathematical Functions, M. Abramowitz and I. A. Stegun, Eds., Dover Publications Inc., 9th printing, 1970.
- [11] Y. Okamoto and R. Talman, "Rational Approximation of the Complex Error Function and the Electric Field of a Two-Dimensional Charge Distribution," CBN 80-13, Sept. 1980.
- [12] IMSL Math/Library<sup>®</sup> Reference Manual, v. 10.0.
- [13] M. A. Furman, A. Zholents, T. Chen and D. Shatilov, "Comparisons of Beam-Beam Code Simulations", CBP Tech Note-59/PEP-II AP Note 95.04, July 13, 1995.
- [14] M. A. Furman, A. Zholents, T. Chen and D. Shatilov, "Comparisons of Beam-Beam Simulations," Proc. Seventh Advanced ICFA Workshop on Beam Dynamics, JINR, Dubna, Russia, 18-20 May 1995, (P. Beloshitsky and E. Perelstein, eds.), p. 123.
- [15] S. Krishnagopal and R. Siemann, "Coherent Beam-Beam Interaction in Electron-Positron Colliders," Phys. Rev. Lett. 67, pp. 2461–2464 (1991).
- [16] S. Krishnagopal, "Luminosity-Limiting Coherent Phenomena in Electron-Positron Colliders," Phys. Rev. Lett. 76, pp. 235–238 (1996).
- [17] "PEP-II: An Asymmetric B Factory - Conceptual Design Report," June 1993, LBL-PUB-5379/SLAC-418/CALT-68-1869/UCRL-ID-114055/ UC-IIRPA-93-01.
- [18] M. A. Furman, "Beam-Beam Tune Shift and Dynamical Beta Function in PEP-II," Proc. European Particle Accelerator Conference, London, England, June 27–July 1, 1994, p. 1145.
- [19] K. Hirata and E. Keil, "Coherent Beam-Beam Interaction Limit in Asymmetric Ring Colliders," Phys. Lett. B232, 413 (1990).
- [20] K. Hirata, "Coherent Betatron Oscillation Modes Due to the Beam-Beam Interaction," NIM A269, 7 (1988).
- [21] R. E. Meller and R. H. Siemann, "Coherent Normal Modes of Colliding Beams," IEEE Trans. Nucl. Sci. NS-28 No. 3, 2431 (1981).
- [22] K. Yokoya, Y. Funakoshi, E. Kikutani, H. Koiso and J. Urakawa, "Tune Shift of Coherent Beam-Beam Oscillations," KEK Preprint 89-14; K. Yokoya and H. Koiso, Part. Accel. 27, pp. 181–186 (1990).
- [23] I am indebted to E. Keil for discussions and for unpublished results on this point.
- [24] M. A. Furman, "Parasitic Collisions in PEP-II," Proc. Seventh Advanced ICFA Workshop on Beam Dynamics, JINR, Dubna, Russia, 18-20 May, 1995 (P. Beloshitsky and E. Perelstein, eds.), p. 36.
- [25] M. A. Furman, "The Classical Beam-Beam Interaction for the Muon Collider: A First Look," LBL-38563/BF-19/CBP Note-169, April 25, 1996, Proc. 1996 Snowmass Workshop "New Directions for High-Energy Physics."

# ODYSSEUS: A DYNAMIC STRONG-STRONG BEAM-BEAM SIMULATION FOR STORAGE RINGS

E. B. Anderson, T. I. Banks, and J. T. Rogers,  
Laboratory of Nuclear Studies, Cornell University, Ithaca, NY 14853

*Abstract*

We have developed a simulation of the beam-beam interaction in  $e^+/e^-$  storage ring colliders which is specifically intended to reveal the dynamic collective behavior of the colliding beams. This program is a true 6-dimensional strong-strong simulation in which the electromagnetic fields of longitudinal slices of the colliding beams are recalculated for each slice collision. Broadband wake fields are included and no constraints are placed on the distribution of particles in the beams. Information on tests of the code will be shown. Results will be presented including limiting beam-beam parameters for round and flat beams, deviations from the Gaussian distribution, effects of the beam-beam parameter on head-tail instability thresholds, and Landau damping rates. Possibilities for further improvements will be discussed.

## 1 INTRODUCTION

Our development of a beam-beam simulation program was motivated, in part, by the observation in the Cornell Electron Storage Ring (CESR) of an  $m = -1$  head-tail instability which occurs at lower beam current when beams are in collision. This change seems to indicate that the instability is interacting with the beam-beam effect. The head-tail instability involves particles moving forwards and backwards within the bunch, so any model for this interaction would have to describe the longitudinal dynamics of the bunch. In addition, collective effects arising from the beam-beam force alone can limit luminosity. The flip-flop instability is commonly observed in  $e^+e^-$  storage ring colliders. The DCI storage ring at LAL, Orsay, France, had four colliding beams in which the  $e^+$  beam charge was compensated by the  $e^-$  charge, but the beam-beam limit was not significantly different from that for uncompensated beams. The beam-beam limit in DCI was attributed to a collective beam-beam instability [1]. This suggests that the beam-beam limit for two-beam collisions may also be due, in some cases, to a collective instability.

This paper presents a new beam-beam simulation program, ODYSSEUS (Optimized DYnamic Strong-Strong E-plus e-minUs Simulation). To the author's knowledge, ODYSSEUS is the first six-dimensional strong-strong beam-beam simulation in which no constraints are placed on the beams and is the first to include wake fields. These features make it possible to investigate any mode of oscillation of the colliding beams. ODYSSEUS is designed to serve as a flexible, efficient, and portable tool for investigating beam-beam effects.

## 2 BASIC IDEAS

ODYSSEUS uses macroparticles to model the six-dimensional motion of the particles in the beams. Typically, the number of macroparticles used is on the order of ten thousand in each beam. With this number of macroparticles the speed of the calculation is limited by the electromagnetic field calculations. ODYSSEUS adaptively chooses from a variety of different field computation methods. Different algorithms are used for the core and transverse tails of the beam and for longitudinal slices with large or with small charge. The parameters of the program can be changed to model flat or round beams. Further, inclusion of the longitudinal degrees of freedom and wake fields allows the investigation of previously inaccessible physics.

### 2.1 Particle Tracking

On each simulated turn through the storage ring, each macroparticle is propagated from the collision point and back again through the linear optics of the storage ring, including chromaticity, synchrotron radiation excitation and damping, RF phase focusing, and wake field deflections.

The magnetic optics of the ring are approximated with linear transport theory as described in many sources including a popular article by M. Sands [2]. The fields in the RF cavities are approximated as sinusoids, while the change in position is handled using a momentum compaction. Macroparticles which have migrated past a transverse aperture are no longer considered in the simulation. As they pass the aperture, the positions and velocities of these particles are recorded for later analysis. ODYSSEUS handles the longitudinal variation of the electromagnetic field of the beam by dividing the beam into slices that were typically of equal thickness. Because the particles are moving at ultra-relativistic speeds, it is approximated that the fields are entirely transverse.

Individual macroparticles undergoing longitudinal oscillations may migrate from slice to slice, so on each turn the macroparticles are sorted according to their longitudinal position and reassigned to slices. This is necessary for the calculation of both the electromagnetic wake field and the actual beam-beam force. The longitudinal motion in most accelerators is slow, so during collisions each macroparticle is assumed to remain within its slice. Because the motion of a macroparticle from slice to slice is slow, Heapsort is an effective sorting algorithm [3].

## 2.2 Radiation Effects

There are two major ways that random perturbations due to synchrotron radiation are handled in beam-beam simulations. The first is to use the description of the storage ring to find the cumulative effects of the radiative kicks and apply these to the beam. The second is to pick a radiative kick that gives the known beam size or emittance. In the horizontal direction the design of the magnetic optics closely determines the magnitude of the radiative perturbations. In the vertical direction errors in the optics and small deviations from a horizontal orbit dominate the radiative perturbations. For these reasons, ODYSSEUS uses information derived from the synchrotron radiation integrals to determine the size of the radiative kicks in the horizontal case and phenomenological perturbations are added in the vertical direction to agree with the actual beam size.

## 2.3 Wake Fields

Longitudinal and transverse, single bunch, short-range wake fields are included in the simulation. One of the program inputs is a list of longitudinal and transverse resonators with values for the resonant frequencies, shunt impedances, and quality factors of the resonators. The wake functions are therefore the sum of exponentially damped sinusoids. The wake fields are calculated by summing up the effect of each of the effective resonators.

## 2.4 Collisions

During its passage through the opposing bunch, the transverse position of each macroparticle may change appreciably because the vertical interaction point beta function,  $\beta_V^*$ , in CESR and many other colliders is comparable to the bunch length,  $\sigma_z$ . The simulation collides each pair of slices sequentially, updating the transverse momenta and positions of each macroparticle after each pairwise collision of slices. For each slice collision  $\langle x \rangle$ ,  $\langle y \rangle$ ,  $\langle x^2 \rangle$ ,  $\langle y^2 \rangle$ , and the total charge for the slice distribution are found, and the slice electromagnetic field is calculated.

## 3 FIELD CALCULATION

For purposes of calculating the electromagnetic field from the beam, each beam is divided into longitudinal slices. Although the number of slices can be set arbitrarily, about fifteen are typically used. The field from each slice, integrated over the length of the slice, is calculated independently. The beams are assumed to be ultra-relativistic, so the field due to each slice is transverse and affects only the particles within the region of that slice.

The calculation of the electromagnetic field of each beam is adaptive in order to maximize the speed of the program. One method uses moments of the beam to calculate an approximate electromagnetic field, while others make calculations of the fields on a rectangular grid. Different methods are used depending on whether the field is calculated for the region of the beam core or for the beam tails,

whether the number of macroparticles within a slice,  $N$ , is large or small, and whether the number of grid points,  $N_g$ , used in the field calculation is large or small.

### 3.1 Beam core

#### 3.1.1 Small $N$

If the number of macroparticles,  $N$ , within a slice is very small, the integrated field at a probe beam macroparticle is calculated from the exact radius vector from each opposing source beam macroparticle. The field must be calculated at the position of each macroparticle in the probe beam, so the number of calculations goes as  $N^2$ , making this method efficient only for very small  $N$ . In practice this method is only used when  $N$  is less than fifty.

#### 3.1.2 Large $N$ , Small $N_g$

For larger values of  $N$ , the electromagnetic field is calculated on a rectangular grid using pre-calculated Green's functions for charges on the grid points. Since the Green's function describes the effect of a single unit charge particle, the Green's function on the grid is found by calculating the field at each grid point due to a unit charge at the origin:

$$G(\vec{r}) = \int_{+\infty}^{-\infty} \vec{E}(s, r) ds = \frac{1}{2\pi\epsilon_0} \frac{\hat{r}}{r} \quad (1)$$

where  $\vec{r}$  is the position vector for the grid point. Notice that the Green's function describes the integrated field strength and has been doubled to take into account the effects of the magnetic field. The macroparticle charge is assigned to the grid points using one of two area-weighted techniques (both techniques described are in Section 5). For small values of the number of grid points, the convolution of the charge density and Green's function is done as a summation in real space. The number of calculations required for this convolution goes as  $N_g^2$ . The portion of the code whose speed is dependent on the number of macroparticles is now only linear in  $N$ . This technique is used only when the number of grid points is quite small, generally under two hundred.

#### 3.1.3 Large $N$ , Large $N_g$

For larger values of  $N_g$ , the convolution of the Green's functions and charge density is done as a simple multiplication in wavenumber space. The speed of this method is limited by the speeds of the necessary Fourier transform to wavenumber space and the inverse transform back to real space. The number of calculations goes as  $N_g \log_2 N_g$ . To suppress edge effect problems in the Fourier transforms, the size of the wavenumber space is doubled in both directions and padded with zeros [4].

## 4 BEAM TAILS

The tails of the beam, typically taken to be particles with a displacement of more than  $(10/3)\sigma$  in the horizontal, vertical, or longitudinal directions, are treated differently than

the core particles. The tail particles have very little effect on the beam-beam force. They do, however, respond to the beam-beam force and must be tracked to determine the beam lifetime. Performing a strong-strong calculation for the beam tails with the grid method is computationally inefficient and unnecessary, so a weak-strong calculation is used.

#### 4.1 Longitudinal Tails

Longitudinal tail particles are subject to forces from the core of the opposing beam. This is a weak-strong calculation. A full calculation of the field from the opposing beam slice is performed, as described above for the beam core. The tails are assumed to have no effect on the other beam. It should be noted that the user chooses the number of slices that will be treated in a weak-strong manner and that all slices can be treated as strong-strong slices if the user chooses to do so.

#### 4.2 Transverse Tails

The transverse tail particles are subject to a beam-beam force of similar magnitude to that experienced by the core particles. The fine structure of the charge distribution of the core has little influence on the field in the transverse tails, so the field there is calculated from a two-dimensional Gaussian charge distribution with the same  $\langle x \rangle$ ,  $\langle y \rangle$ ,  $\langle x^2 \rangle$ ,  $\langle y^2 \rangle$ , and total charge as the charge distribution of the slice. The field from this Gaussian charge distribution is calculated from the rational approximation of Talman and Okamoto [5] for the complex error function solution of Bassetti and Erskine [6].

## 5 INTERPOLATION TECHNIQUES

Whenever a grid-based technique is used, it is necessary to interpolate. The charges of the macroparticles must be distributed on a grid for the field calculations, and the fields calculated on the grid must be applied to particles at arbitrary locations. In order to conserve momentum, the same interpolation scheme must be used in these two situations[4].

The lowest order interpolation scheme used in ODYSSEUS is the Cloud-In-Cell (CIC) technique. In this scheme the macroparticle is treated as a uniform cloud the size of a grid rectangle. The portion of this cloud closest to a grid point is assigned to that point. Since the interpolation in ODYSSEUS is done on a two-dimensional grid, this involves the four nearest grid points.

The second-order techniques that are most useful for this type of calculation are the symmetrical five- and nine-point interpolation schemes. A nine-point interpolation scheme has been coded as an option in ODYSSEUS and is typically used instead of the CIC. The nine-point scheme that is used is a natural extension of a one-dimensional technique called Triangular-Shaped Cloud (TSC). In the one-dimensional case, TSC represents a macroparticle with a triangular cloud two grid spaces wide. The fraction of

the area of the cloud that is closest to each of the grid points is assigned to that point. ODYSSEUS uses the two-dimensional extension of the technique. A set of three fractions is found for each dimension, and these are multiplied to find the weights at all nine nearby points.

When higher order interpolation schemes are used, the charge of the macroparticle can be spread out so that it conceals some of the structure of the charge distribution. This is corrected by using a ‘‘sharpening function’’. In ODYSSEUS sharpening is done during the convolution of the charge density and Green’s function in Fourier space. To determine the sharpening function on the grid, a unit charge is placed exactly on a grid point. The interpolation scheme is then used, and some fraction of the charge will be deposited on grid points other than the one where the unit charge is actually located. The Fourier transform of this grid is then found. If  $G$  represents the Green’s function,  $\rho^\ddagger$  represents the spread-out charge distribution,  $S$  represents the spreading function,  $F$  represents the appropriate electromagnetic potential or field, and  $F^\ddagger$  the spread-out potential or field, then:

$$S(\vec{r}) \star F(\vec{r}) = F^\ddagger(\vec{r}), \quad (2)$$

$$F^\ddagger(\vec{r}) = G(\vec{r}) \star \rho^\ddagger(\vec{r}), \quad (3)$$

and

$$S(\vec{r}) \star F(\vec{r}) = G(\vec{r}) \star \rho^\ddagger(\vec{r}). \quad (4)$$

Then in Fourier space it can simply be written that:

$$\tilde{F}(\vec{k}) = \frac{\tilde{G}(\vec{k})\tilde{\rho}^\ddagger(\vec{k})}{\tilde{S}(\vec{k})}. \quad (5)$$

Because the fields are actually spread out twice, once when interpolating to the grid and once when interpolating back to the macroparticles, the actual expression used is:

$$\tilde{F}(\vec{k}) = \frac{\tilde{G}(\vec{k})\tilde{\rho}^\ddagger(\vec{k})}{\tilde{S}(\vec{k})^2}. \quad (6)$$

## 6 PRE- AND POST-PROCESSING

The preprocessing program Penelope was written to maintain consistency between the input variables and provide a easy to use, portable interface. ODYSSEUS was designed to investigate the coherent oscillations of the bunch, so post-processing to analyze the bunch spectrum is necessary. Post-processing and spectral analysis is done in a *Mathematica* [7] notebook.

## 7 TESTING

### 7.1 Field Errors

A small program was written that generates the fields by all the methods used in ODYSSEUS and compares them. There is no difference between the real and Fourier space PIC calculations.

There are significant differences between the time required for field calculations for flat and round beams. With

round beams a field calculation grid can be constructed out of nearly square cells with equal numbers of cells in each dimension. For example, a 32 by 32 grid has only 1024 grid cells, which allows it to run in a reasonable amount of time. In contrast, the often extreme aspect ratios of flat beams force the use of either large numbers of cells or individual cells with poor aspect ratios. For instance, a minimal 8 by 512 grid, appropriate for CESR, has 4096 cells and takes about five times as long as the round beam calculation above.

When trying to accommodate flat beams with a reasonable number of grid cells, one option is to give the individual grid cells aspect ratios other than unity, but there are dangers in this method. Unless the grid cells are perfectly square, there is no choice for the value of the potential at the origin which provides symmetry between the  $x$  and  $y$  components of the gradient of the potential. Using an electric field-based calculation and a separate Green's function for each component removes this problem, but many interpolation techniques will break down as the aspect ratio of the cells increases. ODYSSEUS uses a distinct Green's function for each component of the electric field. The maximum aspect ratio usually allowed was 1.4, but this could be relaxed to significantly larger values for less demanding calculations.

## 7.2 The Number of Slices

The experience of previous investigators [8] had indicated that low numbers of slices, five or less, are necessary. This estimate was not reasonable for ODYSSEUS for two reasons. One reason is that most previous simulations included a significant natural vertical emittance that made many slicing errors insignificant. The second is that the low  $\beta_y^*$  in CESR is comparable to the bunch length, making the hourglass-like effect at the interaction point more important. With round beams the  $\beta_y^*$  tends to be higher, which decreases the hourglass effect and thus the number of slices.

In the case of flat beams, simulations of beams with low, non-colliding, vertical emittances need very large numbers of slices. As discussed below, uniform slicing is partially to blame. Other methods may exist which require fewer slices, which is important since the speed of ODYSSEUS scales as the square of the number of strong slices in each beam.

An analytic estimate was made of the maximum possible tune shift that could occur *solely* from the uniform slicing method. In a particle-tracking simulation using longitudinal slices of uniform width, particles at the front and back of each slice receive a different deflection than would actual beam particles. The deflection error, to first order in the derivatives of  $\rho(z)$ , is

$$\delta y'_m \approx -\Delta y'_m \frac{1}{\rho} \frac{d\rho}{dz} \delta z \quad (7)$$

where  $\Delta y'_m$  is the deflection due to slice  $m$  in the absence of the error, and  $\delta z = z - (z_m + z_{m+1})/2$ . Slice  $m$ , where

$m = -(M-1)/2, \dots, -1, 0, 1, \dots, (M-1)/2$ , spans the interval  $(z_m, z_{m+1})$ . For a Gaussian  $\rho(z)$ ,

$$\delta y'_m = \sqrt{2\pi} w (mw) e^{-m^2 w^2 / 2} \frac{\xi_y}{\beta_y^*} \left( y_0 + y' \frac{\sigma_z}{2} mw \right) \frac{\delta z}{\sigma_z} \quad (8)$$

where  $w = \Delta z / \sigma_z$  is the slice length  $\Delta z$  in units of  $\sigma_z$ . Summing over all  $M$  slices,

$$\delta y'_{tot} = \sum_{m=-(\frac{M-1}{2})}^{(\frac{M-1}{2})} \delta y'_m \quad (9)$$

$$\approx \frac{\sqrt{2\pi} \xi_y y' \delta z}{2\beta_y^*} \int (mw)^2 e^{-(mw)^2 / 2} d(mw) = \frac{\pi \xi_y y' \delta z}{\beta_y^*} \quad (10)$$

Because  $\delta z$  is approximately uniformly distributed on the interval  $(z_m, z_{m+1})$ ,  $\langle (\delta z)^2 \rangle = (\Delta z)^2 / 12$ , and

$$\langle \delta y'^2_{tot} \rangle = \left( \frac{\pi \xi_y}{\beta_y^*} \right)^2 \langle (\delta z)^2 \rangle \langle y'^2 \rangle = \left( \frac{\pi \xi_y}{\beta_y^*} \right)^2 \frac{(\Delta z)^2}{12} \sigma_y^2 \quad (11)$$

Because the  $\delta z$  of a particle changes in a non-periodic way, we will consider it to be random from turn to turn. When the beam is in equilibrium between random excitation and damping,

$$\langle \delta y'^2_{tot} \rangle = 4\delta \sigma_y'^2 \quad (12)$$

where  $\delta$  is the vertical damping decrement. From equations (11) and (12) we find that

$$\xi_{y,max} = \frac{4\sqrt{3}}{\pi} \frac{\beta_y^*}{\Delta z} \sqrt{\delta} \quad (13)$$

This is the maximum value of  $\xi_y$  which will be produced by the simulation due to the finite slice length. Other physical or numerical effects may further reduce  $\xi_y$ .

A series of calculations were made in the Gaussian approximation to investigate the effect of slicing. If increasing the number of slices used does not affect the beam's size, tune-shift, and luminosity, then the original number of slices is probably sufficient. At forty-five slices and more the effect of additional slices drops off rapidly. The noise introduced by slicing is typically unimportant compared to the natural vertical emittance for forty-five slices. For some applications of ODYSSEUS, forty-five slices are prohibitive, and a better slicing technique will need to be implemented.

## 7.3 Head-Tail Modes

Simulated head-tail damping modes for non-colliding beams were in accordance with expectations. With beams in collision, the  $m = +1$  and  $m = -1$  modes were both shifted upwards in frequency by one-half of the  $\sigma$ -mode /  $\pi$ -mode tune shift. This is in accordance with the predictions of Cornelis and Lamont [9].

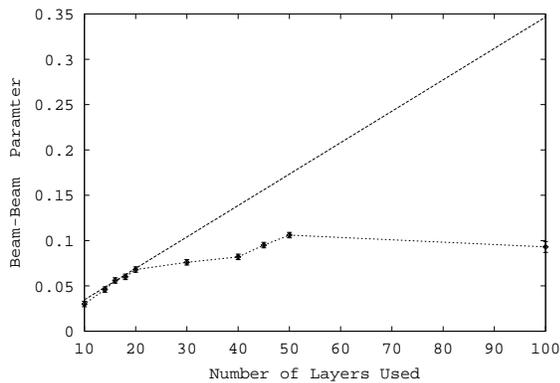


Figure 1: The figure above shows the vertical beam-beam tune shift,  $\Delta\nu_y$ , as a function of the number of slices for a series of runs. The straight line is an estimate of the maximum beam-beam parameter,  $\xi_y$ , from slicing errors alone.

#### 7.4 Speed

ODYSSEUS is currently being run on a 500 MHz  $\alpha$ -chip personal computer running LINUX. On this machine the Gaussian approximation runs in this paper were done in one to four hours, and the PIC methods took from less than a day to five days. Runs lasted for a few radiation damping times, from twenty to eighty thousand turns.

### 8 THE BEAM-BEAM RATIO

A number of predictions have been made for the beam-beam ratio ( $\Delta\nu/\xi$ ) [10, 11, 12, 13, 14, 15]. An interesting initial test for the program was to compare the results of ODYSSEUS with these previous results. The beam-beam ratio was calculated using ODYSSEUS. Flat beams at 8 mA under CESR conditions were used to measure  $\Delta\nu/\xi$ . This ratio was done for beams with a longitudinal extent and also for pancake-like beams. For three-dimensional beams the vertical beam-beam ratio was found to be 1.39, while the ratio was 1.18 for two-dimensional beams. The horizontal beam-beam ratio was 1.0 in both cases.

### 9 CONCLUSIONS

There are a number of important computational advances represented in ODYSSEUS. The most important of these is its adaptive nature. The code dynamically chooses to sum the forces over each particle individually, use a Gaussian charge density approximation, or use a PIC method in real or Fourier space. The grid used in the PIC calculations is pre-generated, then readjusted dynamically as the beam changes. ODYSSEUS also handles wake fields, something that has not been included in similar simulations. Approximations are used in the transverse and longitudinal tails in order to save time on calculations. Runs can be done in reasonable amounts of time, ranging from an hour to a few days depending on the approximations used and beam

shape. ODYSSEUS is a significant advance in the simulation of beam-beam effects, and there are possibilities to improve it further.

There were some surprises in the writing and benchmarking of ODYSSEUS. One surprise was the importance of the individual cells' aspect ratios when deciding what sort of a grid to use for PIC calculations. Another was the importance of slicing algorithms and, consequently, the number of slices required. Improvements in the slicing portion of the code have the possibility of making ODYSSEUS significantly faster.

### 10 ACKNOWLEDGMENTS

This work was supported by the National Science Foundation. The authors extend their thanks to Richard Talman, John Irwin, Bob Siemann, Miguel Furman, Srinivas Krishnagopal, Rob Ryne, Ray Ng, Ray Helmke, Jerry Codner, and the other participants in the Strong-Strong Beam-Beam Workshop at SLAC in the summer of 1998.

### 11 REFERENCES

- [1] J. LeDuff, M. Level, P. Marin, E. Sommer, and H. Zyngier, *Proceedings of the 11th International Conference on High Energy Accelerators*, 707 (1980).
- [2] M. Sands, Stanford Linear Accelerator Center Report, SLAC-121, UC-28, ACC (1970).
- [3] W.H. Press, *et al.*, *Numerical Recipes in FORTRAN: The Art of Scientific Computing* (Cambridge University Press, New York, 1992)
- [4] R.W. Hockney and J.W. Eastwood, *Computer Simulation Using Particles* (Hilger, Bristol, U.K., 1988).
- [5] Y. Okamoto and R. Talman, Cornell Electron Storage Ring Report, CBN 80-13 (1980).
- [6] M. Bassetti and G.A. Erskine, European Laboratory for Particle Physics Report, CERN-ISR-TH/80-06 (1980).
- [7] *Mathematica*, Wolfram Research, Inc., Champaign, Illinois.
- [8] M. Furman, A. Zholenta, T. Chen, and D. Shatilov, PEP-II/AP Note 95.39/LBL-37680/CBP Note-152 (1995).
- [9] K. Cornelis and M. Lamont, *Proceedings of the 4th European Particle Accelerator Conference, London, 1994*, 1150 (1994).
- [10] A. Piwinski, *I.E.E.E. Transactions in Nuclear Science*, **NS-26**, 4268 (1979).
- [11] A. Chao, *AIP Conference Proceedings, 127: Physics of High Energy Particle Accelerators*, (AIP Press, New York, 1983).
- [12] R. Talman, Cornell Electron Storage Ring Report, CLNS-84/610 (1980).
- [13] K. Hirata, *Nuclear Instruments and Methods in Physics Research*, **A269**, 7 (1988).
- [14] R. E. Meller and R. H. Siemann, *I.E.E.E. Transactions in Nuclear Science*, **NS-28**, 2431 (1981).
- [15] K. Yokoya, Y. Funakoshi, E. Kikutani, H. Koise, and J. Urakawa, High Energy Accelerator Research Organization (KEK) preprint 89-14 (1989).

# BEAM-BEAM SIMULATIONS WITH GUINEA-PIG

D. Schulte, CERN, CH-1211 Geneva 23, Switzerland

## Abstract

While the bunches in a linear collider cross once only, due to their small size they experience a strong beam-beam effect. Guinea-Pig is a code to simulate the impact of this effect on luminosity and background. A short overview of the program is given, with examples of its application to the background studies for TESLA, the top threshold scan and a possible luminosity monitor; as well as some results for CLIC.

## 1 INTRODUCTION

In future high energy linear colliders the beams have to be focused to very small sizes in order to achieve the required luminosity. The electro-magnetic field of each bunch will then have a strong effect on the other. In electron positron collisions the resulting forces focus the two beams, leading to an enhancement of the luminosity. Due to the strong bending of their trajectories the beam particles will emit high energy photons, the so-called beamstrahlung. This significantly changes the luminosity spectrum and can lead to an increase in the background.

Electron positron pairs can be produced in different processes during the collision. These particles can be low in energy and their trajectories are strongly affected by the beam fields. This has an important impact on the detector design in order to reduce the resulting background. Other sources of background are bremsstrahlung and the production of hadrons by the two-photon interaction.

The beam-beam interaction can only partially be treated analytically, simulations are therefore crucial. The code GUINEA-PIG [1] (Generator of Unwanted Interactions for Numerical Experiment Analysis—Program Interfaced to GEANT) is written in C and compiles with the GNU C-compiler. A copy of the code can be obtained from the author [2].

## 2 TRACKING ALGORITHMS

In the simulation, the beam particles are replaced by typically 20000 – 500000 macro-particles. The beams are cut longitudinally into a number of slices. Each slice of one beam interacts subsequently with each slice of the other beam. Since the particles are ultra-relativistic, the slice-slice interaction can be treated as a purely two dimensional problem using a clouds-in-cells method. The slice is transversely cut into cells, and the charge of the macro-particles is distributed onto this grid according to their position. Then the potentials on the grid points are evaluated and from these the forces on the particles.

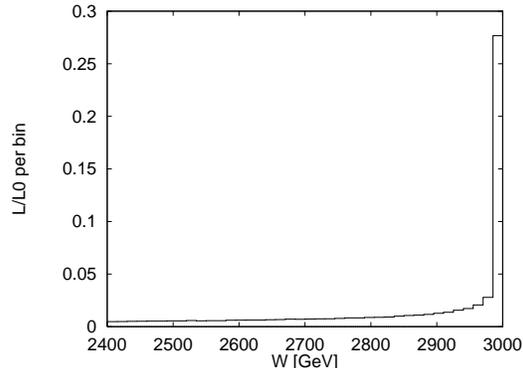


Figure 1: The luminosity spectrum for CLIC at  $E_{cm} = 3$  TeV.

Calculating the potentials is one of the most time consuming operations, the simplest algorithm sums for each cell over all the others. Replacing the summation with a Fast Fourier Transformation increases the speed by a significant factor.

## 3 BEAMSTRAHLUNG

The bending of the beam-particle trajectories leads to the emission of photons (beamstrahlung). This effect is equivalent to synchrotron radiation. Instead of the critical energy  $E_c$  one commonly uses the beamstrahlung parameter  $\Upsilon = 2/3 \cdot \langle E_c \rangle / E_0$  for its description. The designs at 0.5–1 TeV centre-of-mass energy have usually  $\Upsilon \ll 1$ , which is called the classical regime. The fraction  $\delta$  of its energy a particle loses due to beamstrahlung is given in Tab. 1 for different designs. The number of photons emitted per beam particle is of the order of one. Therefore the discreteness of the process has to be taken into account. In the program the produced photons are tracked and the corresponding electron-photon, positron-photon and photon-photon luminosities are calculated as well as the resulting background.

In CLIC at  $E_{cm} = 3$  TeV the beamstrahlung parameter will be much larger than one— $\Upsilon = 8.7$ . The beamstrahlung is in this regime partly suppressed since the beam particles cannot emit photons with a higher energy than the particle has itself. The luminosity spectrum is shown in Fig. 1.

## 4 PAIR PRODUCTION

Electron-positron pairs can be produced by coherent and incoherent processes. For small  $\Upsilon$  the incoherent processes dominate. The three most important are called Breit-

Table 1: Background levels for different collider designs. For TESLA the reference and high luminosity parameters (iL) are given. All designs have no crossing angle or use crab crossing.

		TESLA	TESLA iL	ILC	ILC	CLIC	CLIC	CLIC
$E_{cm}$	[GeV]	500	500	500	1000	500	1000	3000
$f_{rep}$	[Hz]	5	5	120	120	200	150	75
$N_b$		1130	2820	95	95	150	150	150
$N$	[ $10^{10}$ ]	3.63	2.0	0.95	0.95	0.4	0.4	0.4
$\gamma\epsilon_x/\gamma\epsilon_y$	[ $\mu\text{m}$ ]	14.0/0.25	10.0/0.03	4.5/0.1	4.5/0.1	1.88/0.1	1.48/0.07	0.6/0.01
$\beta_x/\beta_y$	[mm]	25/0.7	15/0.4	12/0.12	12/0.15	10/0.1	10/0.1	8/0.1
$\sigma_x^*/\sigma_y^*$	[nm]	845/18.9	554/4.95	332/4.95	235/3.9	196/4.52	123/2.7	40.4/0.58
$\sigma_z$	[ $\mu\text{m}$ ]	700	400	120	120	50	50	30
$\Upsilon$		0.03	0.04	0.10	0.29	0.18	0.56	8.7
$\mathcal{L}$	[ $10^{33}\text{cm}^{-2}\text{s}^{-1}$ ]	6.0	31	6.54	12.9	6.3	13.6	133
$\delta$	[%]	2.5	2.8	3.8	9.1	3.6	9.2	32
$n_\gamma$		2.0	1.65	1.16	1.5	0.8	1.1	2.15
$N_\perp$		31	44	9.8	18.4	2.9	8.0	128
$N_H$		0.13	0.23	0.07	0.33	0.022	0.15	8.0
$N_{MJ}$	[ $10^{-2}$ ]	0.30	0.61	0.20	2.3	0.08	1.27	366

$E_{cm}$ : centre-of-mass energy,  $f_{rep}$ : repetition frequency,  $N_b$ : number of bunches per train,  $\sigma$ : bunch dimensions at IP  
 $N$ : number of particles per bunch,  $\mathcal{L}$ : actual luminosity,  $\beta$ : Beta functions at IP,  $\gamma\epsilon$ : normalised emittances,  
 $\Upsilon$ : Beamstrahlung parameter,  $\delta$ : Average energy loss,  $n_\gamma$ : number of photons per beam particle,  
 $N_\perp$ : Number of particles with  $p_\perp > 20$  MeV,  $\theta > 0.15$ ,  $N_{HadR}$ : Hadronic events,  $N_{MJ}$ : Minijet pairs  $p_\perp > 3.2$  GeV/c.

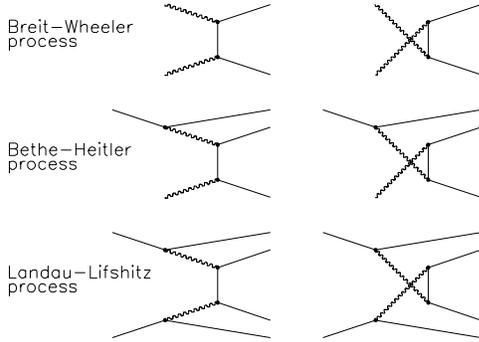


Figure 2: The incoherent pair production processes.

Wheeler, Bethe-Heitler and Landau-Lifshitz process, see Fig. 2. The last two can be derived from the first by the equivalent photon approximation. In this method the initial beam-particle in the Feynmann diagrams are replaced by a spectrum of virtual photons. These photons are treated as being real as long as their virtuality remains below an upper limit. Above this limit they are ignored. In the program the lower boundary of the virtuality  $Q$  is given by  $\hat{Q}^2 = x^2 m^2 / (1 - x)$ . The upper limit  $\hat{Q}^2$  can be chosen to be equal to the electron mass squared  $m^2$  or to depend on the two-photon process. In the latter either the transverse mass squared  $m^2 + p_\perp^2$  of the final state or  $s/4$  can be used as a scale. While the total cross section is not affected very much by the choice, the number of particles with large transverse momentum is. Comparisons of the Landau-Lifshitz process calculated with GUINEA-PIG and the Vermaseren Monte-Carlo show that using  $\hat{Q}^2 = m^2$  significantly underestimates the number of tracks with transverse

momenta of a few MeV [1]. The other two choices are in reasonable agreement but the number of tracks is still somewhat too small, since the imbalance of the transverse momenta of the final state particles is not simulated.

A small transverse momentum of the photon  $q_\perp$  leads to a corresponding uncertainty of its transverse position [3]. If the beams are small enough the photon position can be outside of the beam. This effect was observed at VEPP4 [4] and HERA [5]. In the program it is treated by offsetting the photon with respect to the electron position by  $b = \hbar/q_\perp$ . The total number of pairs is reduced by a factor of about two. The effect on particles with larger angles and transverse momenta, which may enter the detector directly, is weaker.

In the program the produced particles are tracked through the fields of the two beams. Electrons which follow the direction of the electron beam are focused by the positron beam. The effective force of the electron beam is in this case small. The electrons which follow the positron beam are defocused by the electron beam and experience little effect from the positrons. The equivalent is true for positrons. In the program the step size of the particles is adjusted to their energy.

In the coherent pair production process a hard photon turns into an electron positron pair in a strong electromagnetic field. In the high beamstrahlung regime of CLIC at 3 TeV the number of particles from this process is several percent of that of the beam particles. Taking their contribution to the beam fields into account is therefore a necessary extension of the program to be made in the near future.

## 5 BREMSSTRAHLUNG

Another source of low energy particles is the bremsstrahlung process. In this two beam-particles collide and one of them emits a hard photon. In the program this process is implemented again using the equivalent photon approach to replace one of the particles. One then has only to calculate the Compton process. The spectrum of the remnant beam-particles is relatively flat at the low energy end. The particles produced this way always follow the beam with the same charge and are thus focused.

## 6 HADRONIC BACKGROUND

Two-photon collisions also lead to the production of hadrons. The photons interact in these cases not only as point-like particles but also with a certain probability as hadrons. The dependence of the total hadronic cross section on the centre-of-mass energy is thus comparable to the one of hadron-hadron interaction even so its size is considerably smaller. In the program three different parametrisations can be used. The first is the upper estimate taken from reference [7]

$$\sigma_H = 211 \text{ nb} \cdot \left( \frac{s}{\text{GeV}^2} \right)^{0.0808} + 297 \text{ nb} \cdot \left( \frac{s}{\text{GeV}^2} \right)^{-0.4525}$$

A second parametrisation used is taken from reference [8] for comparison. The third is finally again from [7] using the expected cross section rather than the pessimistic case. In Tab. 1 the number of hadronic events  $N_H$  with a centre-of-mass energy in excess of 5 GeV is given for different collider designs. The first parametrisation is used.

A fraction of the hadronic events lead to the production of so-called minijets which have a significant transverse momentum. The simplest of the processes is the direct electro-magnetic production of a quark-anti-quark pair by the point-like photons. It differs from the pair production only by the charge and mass of the final state particles. It is also possible that one of the photons interacts hadronically. In this case a parton of this hadron collides with the other photon producing two jets. The remnants of the hadron form a third jet. If both photons interact as hadrons one finds two of these spectator jets. In total eleven cross sections have to be taken into account for the different quark, gluon and photon combinations.

In the program the real and virtual photons are replaced by spectra of partons for their hadronic interaction—very similar to the replacement of beam particles by virtual photons. The cross section for minijet production rises fast at high photon-photon centre-of-mass energies. To calculate the cross section for events containing minijets one should take into account that the same hadron can produce more than one pair of minijets because more than one of its partons can scatter [9]. This however only affects the distribution of the minijets onto the hadronic events but not the rate itself. One would find fewer events with more jets each

leading to the same total number of jets. In the program this difficulty is ignored and each jet pair is assumed to be produced by a separate hadronic event—this should be a pessimistic approach.

Two parametrisations of the hadronic content of the photon are available in the program, one derived by Drees and Grassi [12], the other by Glück, Reya and Vogt [13]. The assumed values for the scale  $\Lambda_4$  are adjusted automatically to the ones assumed by the authors.

In Tab. 1 the number of minijets  $N_{MJ}$  with a transverse momentum on the parton level of more than  $p_{\perp} > 3.2 \text{ GeV}$  is given. The second parametrisation is used.

The program produces two different files with hadronic output. The first contains the energies of two colliding photons that produce a hadronic event. The second contains the final-state four-vectors of the partons that produced minijets. The fragmentation can be done using PYTHIA [10] or an equivalent program.

## 7 BACKGROUND EFFECTS

Most of the particles from pair production have initially a small angle with respect to the beam axis. Their final angle depends mainly on the deflection by the beam fields. This leads to the distribution shown in Fig. 3 which shows a strong correlation between particle angle and maximal transverse momentum. The vast majority of the particles can be prevented from hitting the detector components by the combined effects of the longitudinal magnetic detector field and limited angular coverage. The solenoid field of the detector will limit the maximum distance a particle can reach from the beam axis depending on its transverse momentum. Particles with large transverse momenta however tend to have smaller angles so they can be avoided by having a dead cone in the detector in the forward region.

Charged particles from the pair production that hit the vertex detector will cause background hits, making the reconstruction of the trajectories from physics events more difficult. Figure 4 shows the number of particles per bunch crossing that will hit the inner layer of the vertex detector in the case of the TESLA high luminosity parameters. In the

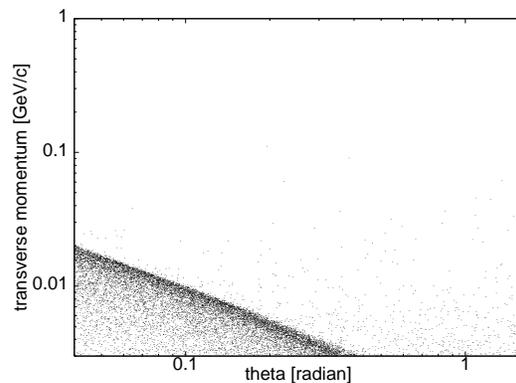


Figure 3: The final angle and transverse momentum of the particles from pair production in the case of TESLA.

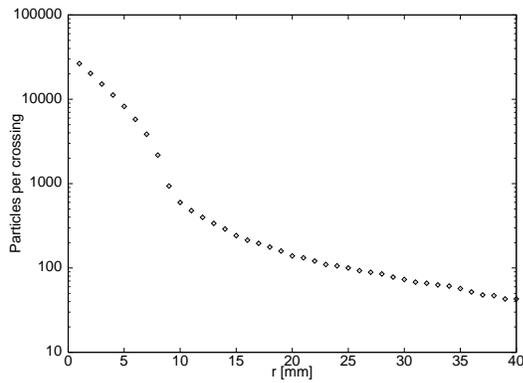


Figure 4: The number of particles that hit the vertex detector as a function of the radius of its inner layer. The magnetic field was assumed to be  $B_z = 3$  T and the opening angle of the detector  $\theta_0 = 200$  mradian.

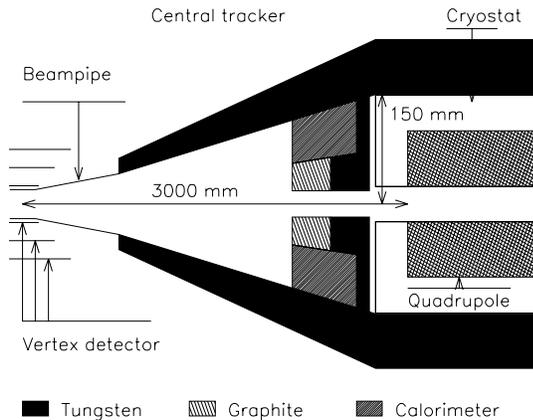


Figure 5: The layout of the mask in the detector of TESLA.

plot the angular coverage of the layer was kept constantly at  $\cos \theta = 0.98$  but the radius was varied. The magnetic field in the detector was assumed to be  $B_z = 3$  T as foreseen in reference [11]. Below a certain radius the number of particles rises drastically. This is because the detector is not any longer only hit by the particles that started with a large angle and transverse momentum. Also those that received their transverse momentum and their angle from the deflection by the beams can produce hits. Using a smaller coverage (larger dead angle) or stronger magnetic field allows to go to smaller radii.

For TESLA it is foreseen to have the innermost layer of the vertex detector at a radius  $r = 25$  mm. Particles can cause more than one hit since they travel along a helix. Using a GEANT based simulation the hit density found is about  $10^{-2} \text{ mm}^{-2}$  for the reference parameters (including a patch factor 1.5 to account for the difference between the Vermaseren Monte-Carlo and GUINEA-PIG). With the high luminosity parameters one finds a hit density of about  $1.5 \cdot 10^{-2} \text{ mm}^{-2}$ .

Also the particles that are not directly entering the vertex detector can cause background. These particles hit the final

quadrupoles on both sides of the detector. The detector can be shielded from the backscattering photons by a tungsten mask as shown in Fig. 5. The charged low energy particles that are backscattered are however led by the main detector field back into the vertex detector—causing an increase in the number of hits by an order of magnitude. This can be prevented by adding an additional inner mask into the main one. This consists of a tungsten layer towards the outside of the detector and a low- $Z$  material towards the inside. The radius of the opening in this mask has to be smaller than the beam pipe to be effective.

## 8 LUMINOSITY MONITOR

One of the important problems in a linear collider is to adjust the beams in the collision point to achieve highest luminosity. Beam-beam deflection scans have been used to this end at the SLC. During this scan however the luminosity production is interrupted.

Three different processes could be used to optimise the collision in future linear colliders during luminosity production [14], namely the bremsstrahlung, beamstrahlung and pair production.

The bremsstrahlung process has the advantage that the event rates are directly proportional to the luminosity except for the beam size effect, which has only a weak dependence on vertical spot size. Either the hard photons or the remnant beam particles which lost a significant amount of energy could be detected in principle. The photon will be accompanied by a large number of softer photons from the beamstrahlung process, so the detector needs to be able to detect only photons above a fixed threshold energy. It seems therefore simpler to detect the low energy remnant particles. Here one has to avoid to detect particles which lost energy due to beamstrahlung or the low energy particles from pair production. As Fig. 6 shows this is possible around  $E = 50$  GeV in the case of TESLA with high luminosity parameters. For CLIC at 3 TeV the bremsstrahlung signal is completely invisible in that of the spent beam, see Fig. 7. One therefore has to find different means to tune the luminosity.

In case of the high luminosity TESLA, the particles from the pair production will deposit about 12 TeV in the inner mask shown in Fig. 5. The measurement of this energy provides a signal which allows to tune the luminosity. The most frequent parameter to tune is the vertical waist of the two beams. Figure 8 shows the luminosity and energy deposited per bunch crossing in the two inner masks if one varies the waist of one of the beams. The other beam is assumed to be at the optimal waist position. The luminosity found from the scan of 20 bunch crossings is lower by a fraction of  $4 \cdot 10^{-4}$  than the optimal one.

## 9 THE TOP THRESHOLD SCAN

Measuring the top production cross section around threshold is useful to determine the top mass  $m_t$ , the strong coupling constant  $\alpha_S$  and the top width  $\Gamma_t$  [15]. The cross sec-

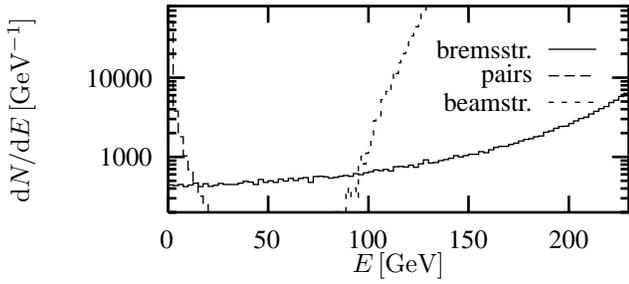


Figure 6: The low energy particle spectra for the TESLA parameters due to the pair production, bremsstrahlung and beamstrahlung.

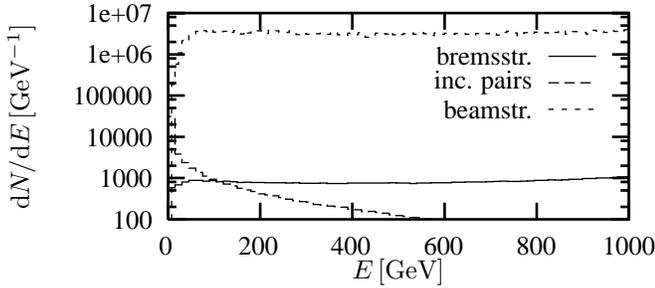


Figure 7: The low energy particle spectra for the CLIC parameters at 3 TeV.

tion is affected by initial state radiation, the energy spread in the beams and the beamstrahlung. The initial state radiation is unavoidable while the other two parameters depend on the machine design. Comparison of TESLA, SBLC and NLC [1] showed some difference which is however not very large.

In Fig. 9 the  $\Delta\chi^2 = 1$  contour of the top scan using nine points is shown in  $m_t$  and  $\Gamma_t$  for fixed  $\alpha_S$ . An integrated luminosity of  $50 \text{ fb}^{-1}$  [15] and the reference parameters of TESLA were assumed. The outer curve corresponds to a background rate three times larger than predicted.

## 10 CONCLUSION

Disruption and beamstrahlung have important effects on the total luminosity and its spectrum in linear colliders. The presented program simulates these and the resulting electro-magnetic and hadronic background. The program was used to evaluate the precision achievable in physics experiments due to smearing of the luminosity spectrum and the background. The possibilities to measure the luminosity were studied, predicting good prospects.

## 11 REFERENCES

[1] D. Schulte. TESLA-97-08 (1996)  
 [2] E-mail address: Daniel.Schulte@CERN.CH  
 [3] G. L. Kotkin, V. G. Serbo, and A. Schiller. Int. J. Mod Phys. A, **7** (1992) 4707–4745.

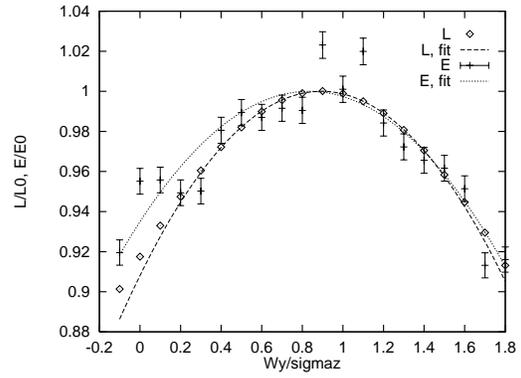


Figure 8: The luminosity and the energy deposited by the particles from pair creation in the inner masks. The position of the vertical waist of one of the beams was varied while the waist of the other beam was in optimal position.

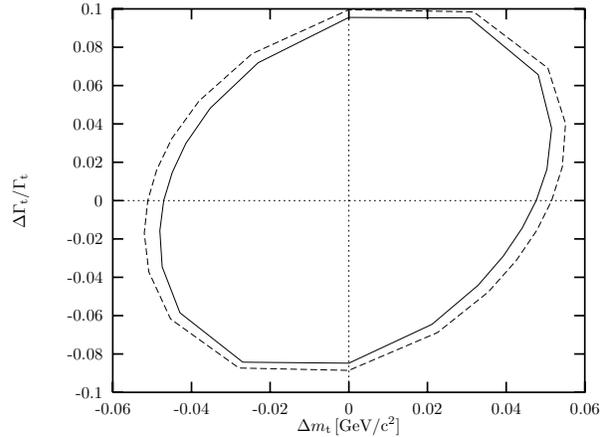


Figure 9: The resolution of the top threshold scan.

[4] A. E. Blinov et al. Phys. Lett. B, **113** (1982) 423.  
 [5] K. Piotrkowski. DESY 95-051 (1995).  
 [6] V. N. Baier and V. M. Katkov. Sov. Phys. Dokl., **17** (1973) 1068.  
 [7] G. A. Schuler and T. Sjöstrand. Z. Phys. C, **37** (1997) 677–688.  
 [8] P. Chen and M. E. Peskin. SLAC-PUB-5873 (1992).  
 [9] J. R. Forshaw and J. K. Storrow. Phys. Lett. B, **268** (1991) 116–121.  
 [10] T. Sjöstrand. Comp. Phys. Comm., **82** (1994) 74-90.  
 [11] R. Brinkmann, G. Materlik, J. Rossbach and A. Wagner (editors). ECFA 1997-182  
 [12] M. Drees and K. Grassie. Z. Phys. C, **28** (1985) 451–462.  
 [13] M. Glück, E. Reya, and A. Vogt. Physical Review D, **46** (1992) 1973.  
 [14] O. Napoly and D. Schulte. Linac 98.  
 [15] A. Juste, M. Martinez and D. Schulte. DESY-97-123-E (1997)

# TRANSITION DYNAMICS OF THE WAKE FIELDS OF ULTRA SHORT BUNCHES

A. Novokhatski, M. Timm and T. Weiland

TU Darmstadt, TEMF, Schlossgartenstr. 8, D-64289 Darmstadt, Germany \*

## Abstract

In the cavities and finite cell structures, ultra short bunches excite very high frequency electromagnetic fields. A fraction of these fields stay in the structure for a very long time. After several reflections another part leaves the structure. The rest part is chasing the bunch. In a time, this field will catch the bunch and take its kinetic energy. The time and the distance, when and where the bunch is caught, is inversely proportional to the bunch length. The time and the distance can be very long for a very short bunch. The analyses of the wake fields in this transient region is given for the Linear Colliders accelerating structure.

## 1 INTRODUCTION

While passing an acceleration section or a multi-cell cavity a short bunch creates high frequency electromagnetic fields in the cells. A fraction of the field, excited in one cell stays there for a long time. After several reflections, another part is leaving the cell. And the rest part is chasing the bunch. In a time this field will catch the bunch and take its kinetic energy. Naturally, this part is also responsible for the excitation of the fields in the next cells. The time or the distance, where the bunch is caught, is inversely proportional to the bunch length. It can be very long for a very short bunch. And the fields in the cells will be also different along this distance. Later, the superposition of the chasing fields will create the "steady state" wake field.

To study the dynamics of the wake fields in this transient region, we used the codes MAFIA[1] and NOVO. The latter was used for the short bunch calculations in the TESLA cavities [2]. A brief description of the algorithm of this code is presented in the last chapter.

## 2 TRANSFORMATION OF THE WAKE FIELDS IN THE SEMI-INFINITE PERIODICAL STRUCTURE

Short bunches interact with single cavity and periodical structure in a different way. In the cavity the loss factor is inversely proportional to the square root of the bunch size. And in the periodical structures loss factor fairly depends upon the bunch size. Usually accelerating sections consist of some number of cavities or cells. And this cavities and cells can be different. Thus the constant gradient NLC accelerating section contains 206 different cells, when the TESLA accelerating cryo - module is the chain of separated 9 cell- cavities. These accelerating structures are not really periodical.

\* Work supported in part by DESY, Hamburg.

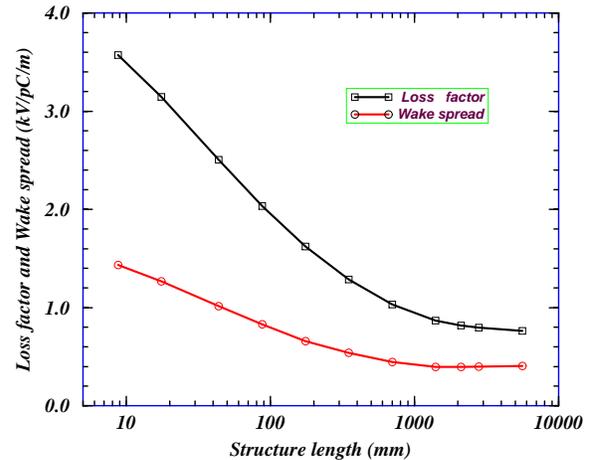


Figure 1: Loss factor and energy spread per unit length of the  $10\mu$  bunch in the periodical iris-loaded structure.

We can assume, that when the bunch comes inside the accelerating section, it has its transverse "self field" and interacts with the cells as with lonely cavities. But in time the fields from excited cells catch the bunch and partially compensate the field, coming into the next cell together with the bunch. For very short bunches this compensation is very strong. The interaction and hence, the loss factor decreases many times. On the Fig.1 the evaluation of the energy loss of the  $10\mu$  bunch in the periodical structure is presented. The period of the structure is  $8.75\text{ mm}$ , the gap is  $6.89\text{ mm}$  and the aperture is  $4.92\text{ mm}$ . The loss factor of the bunch in the periodical structure of TESLA cells is shown on the Fig.2. The bunch length is  $200\mu$ .

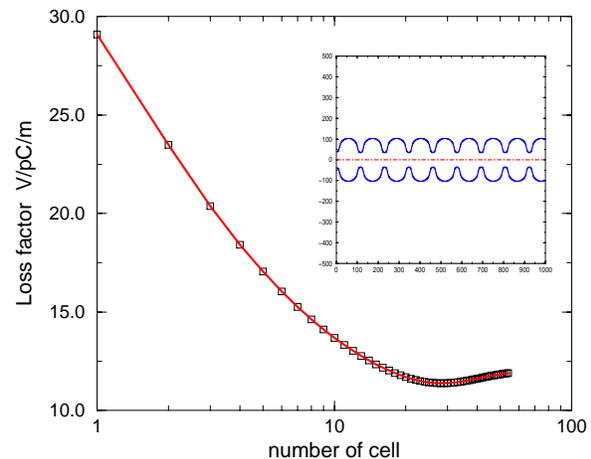


Figure 2: The loss factor of the  $0.2\text{ mm}$  bunch in the periodical structure of the TESLA cells.

Transformation of the wake fields is shown on the Fig.3 and Fig.4 consequently.

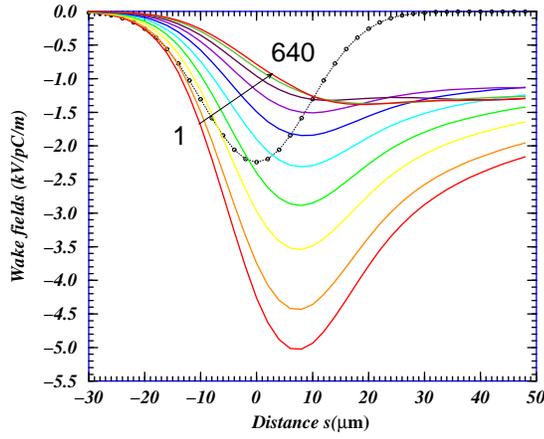


Figure 3: Wake fields of the  $10\mu$  bunch in the periodical iris-loaded structure.

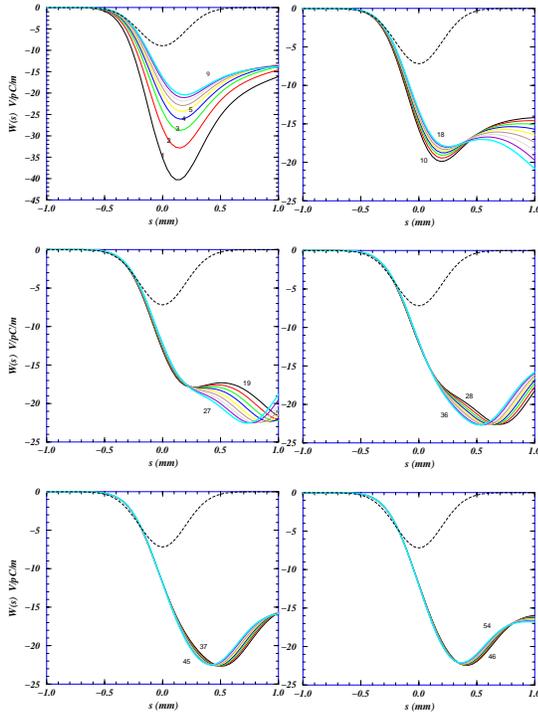


Figure 4: The wake fields of the 0.2 mm bunch in the periodical structure of TESLA cells.

For the iris-loaded structure the wake fields per unit length are given after 1,2,5,10,20,40,80,160, 320 and 640 cells. And for the TESLA cavities results are given for 9 consecutive cells on each picture, making easier comparison with the 9-cell TESLA cavities. This pictures show strong modification of the wake fields along the accelerating structure. Starting from the first cells wake fields are decreasing in amplitude, and the shape is becoming more and more linear around the bunch center, very close to the integral of the charge density.

The same strong modification of the wake fields takes place in bellows as well. In the bellows we have large aperture and small size irises. In the bellows of small number of cells the wake field has inductive character (derivative of the bunch distribution), while in the case of very large number of cells the wake field takes capacitor character (integral

of the bunch distribution).

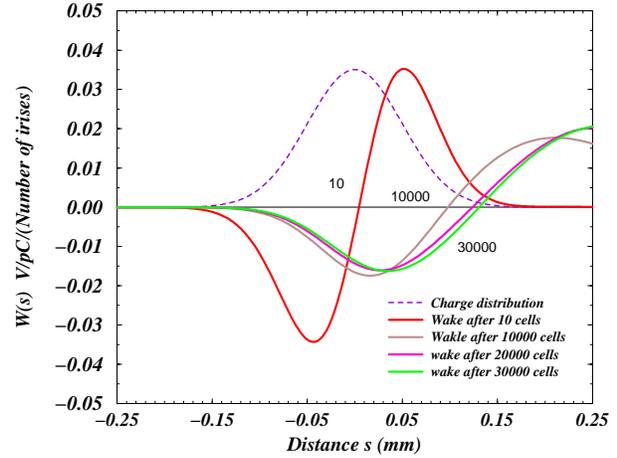


Figure 5: The wake fields of the  $50\mu$  bunch in the bellow of the 5 mm aperture with  $10\mu$  size irises. Wake fields are given after 10, 10000, 20000 and 30000 irises.

This is clearly seen on the Fig.5, where wake fields of the  $50\mu$  bunch in the bellows with small size irises are presented. The aperture of the bellow is 5mm, the gap and the height is  $10\mu$  and the period is  $20\mu$ .

### 3 WAKE FIELD ENERGY

To find real fields, acting on particles, we split the full field  $E_{full}$  in the wake field  $E_{wake}$ , that really acts on the bunch particles and the "self" field  $E_{bunch}$ , that is moving together with this bunch, but does not interact with particles (in the relativistic case).

$$E_{wake} = E_{full} - E_{bunch}$$

The energy distribution of the wake field, following the bunch, can be described by the longitudinal energy density  $\Lambda(s)$ , that is the transverse integral of the energy density at a distance  $s$  from the center of the bunch

$$\Lambda(s) = \frac{\epsilon_0}{2} \int [E_{wake}^2(s) + H_{wake}^2(s)] d\varphi dr$$

When the bunch comes out of the cavity, the integral of this density  $\Lambda(s)$  along the bunch way, shows the energy  $T(s_0)$ , that is following the bunch at a distance  $s_0$  behind

$$T(s_0) = \int_{-\infty}^{s_0} \Lambda(s) ds$$

This integral is approaching the loss factor  $K_{loss}$ , when  $s_0 \rightarrow \infty$

$$K_{loss} = \int_{-\infty}^{\infty} \Lambda(s) ds = T(\infty)$$

While decreasing the bunch length, the loss factor is increasing, and more and more high order modes are excited in a cell. On Fig. 6 the loss factor in one cell of the TESLA cavity is shown over the bunch length, together with the energy integral  $T(s_0)$ .

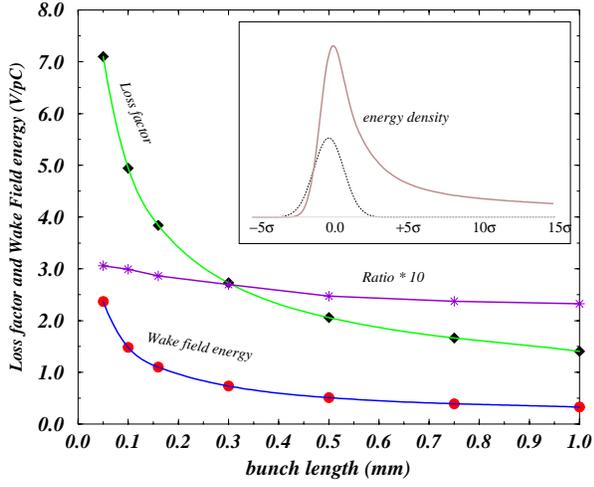


Figure 6: Loss factor and the wake field energy, following the bunch in the tube, after a single cell of the TESLA cavity over the bunch length. Ratio of field energy to the loss factor is multiplied by 10 (to use the same scale).

In this figure, inside the box, the energy density  $\Lambda(s)$  in the tube and the bunch charge distribution are also shown. The energy density has a slowly vanishing tail. The energy of the following field is defined as  $T(s_0 = 15\sigma)$ . The ratio of the energy integral to the loss factor is slightly growing up while the bunch length is decreasing, coming to the value of 30%. So, one third of the "excited energy" immediately leaves the cell with the bunch.

How far will the wake field follow the bunch in the tube? One can predict, that at least, up to the distance  $L$ , where the field "catches" the bunch

$$L = \frac{a^2}{2\sigma}$$

where  $a$  is the radius of the tube,  $\sigma$  is the bunch length. For the regular cell of the TESLA cavity and for the bunch length  $\sigma = 0.5$  mm, this distance  $L = 1.225$  m, is equal to the length of the 9-cell cavity. For the bunch of  $\sigma = 50\mu$ , the distance is more than one accelerating cryomodule length.

As the wake field follows the bunch for a long time, in a multi-cell cavity this field is increasing with the number of cells, that the bunch meets on a distance  $L$ . Results of the computation of the energy integral  $T(15\sigma)$  along the TESLA cavity are shown on Fig. 7.

It can be seen, that the energy of the following field is linearly growing up in the first cells. The number of cells, where the field approaches the asymptotic solution, is determined by the "catch up" length  $L$  and period of structure  $D$

$$N = L/D = \frac{a^2}{2\sigma D}$$

Values for  $N$  for the TESLA cavity (9-cell) are shown in Fig. 7 for different bunch length.

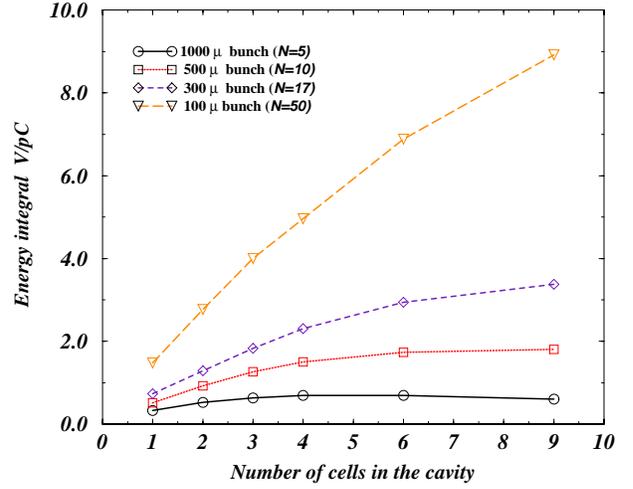


Figure 7: Wake field energy, following the bunch in the TESLA cavity, for bunchlength of 1 mm, 0.5 mm and 0.1 mm.

#### 4 DIRECTED AND REFLECTED FIELDS IN THE TESLA CAVITY

To study the reflected and the directed fields in the TESLA cavity, the model of excitation of one cell was used. The full electromagnetic field is separated and we do calculations only for the wake field. On the surface of the "excited" cavity the wake field has to take the value of the bunch field with negative sign in accordance with boundary conditions. The geometry of the "excited" cell (N2) and the cells around (N1, N3) are shown in Fig. 8. Cell N2 is excited by the bunch of 0.5 mm length. The energy in the cells is calculated and presented in time. When the bunch leaves cell N2 the wake field energy is going with it and excites cell N3; then coming to the end of cell N3, one part of the energy goes to the next cell and the energy in cell N3 is going down. At the same time, the field, reflected from the iris between cell N2 and N3, is crossing cell N2 and coming to cell N1.

We can estimate the reflected and the directed coefficients comparing the field energy in adjoining cells. This coefficients are changing from cell to cell, as the frequency spectrum of the field is changing too. Low frequency modes are slowly moving along the structure, when high frequency modes are traveling with the bunch. The coefficient of transmitted energy from the first cell to the second is around 30%. In the next cell it is approximately 50%. From cell to cell the reflected energy is decreasing and finally high frequency modes are traveling along the structure almost without reflections. If we know the transmission coefficient of the energy  $\Pi$ , then we can estimate the time dependence of the energy in the "excited" cell by

$$\mathcal{E}(t) = \mathcal{E}_0(1 - \Pi)^{ct/D} = \mathcal{E}_0 e^{-t/\tau}$$

and find the energy attenuation time  $\tau$  of high frequency modes in one cell

$$\tau = -\frac{D}{c \ln(1 - \Pi)}$$



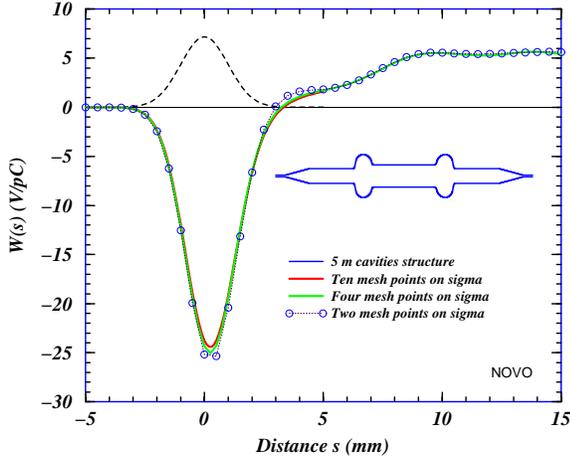


Figure 10: The wake fields of the 1mm bunch in two cavity accelerating structure with tapers. Comparison of the results of calculations with different number of mesh points on the bunch length.

four and two mesh points on the bunch length is given on the Fig.10. It is easy to see, that there is no "mesh dispersion", no modulation with the critical wavelength and no diffusion. To achieve acceptable accuracy even two mesh points on the bunch length will be enough.

The wake potential of a 0.1mm bunch in the constant gradient NLC structure of 206 ellipse cells is shown on Fig.11. Calculations were done for 10 and 5 mesh points on the bunch length. No difference can be noticed between two results. The long distance wake potential is shown on Fig.12. The comparison of the results of 5 and 2 mesh points on the bunch length is shown.

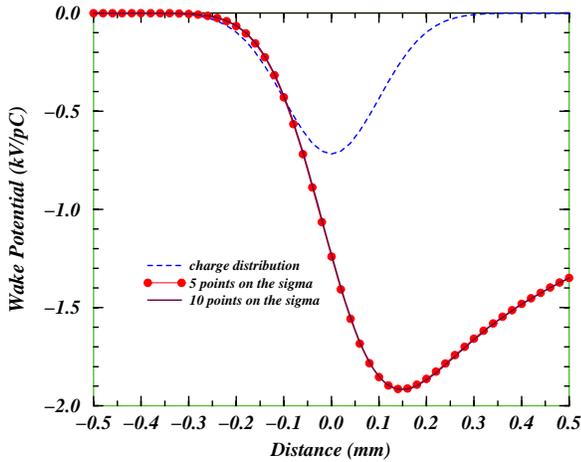


Figure 11: The wake potential of the 0.1 mm bunch in the NCL accelerating section. Comparison of the results of calculations with 10 and 5 mesh points on the bunch length.

The presented scheme can be easily transformed for computation of the fields on the moving mesh points, which is one more advantage for the short bunch wake field calculation. Also it is very easy to make pictures of electric force lines, just to find the lines, where flux of electric field

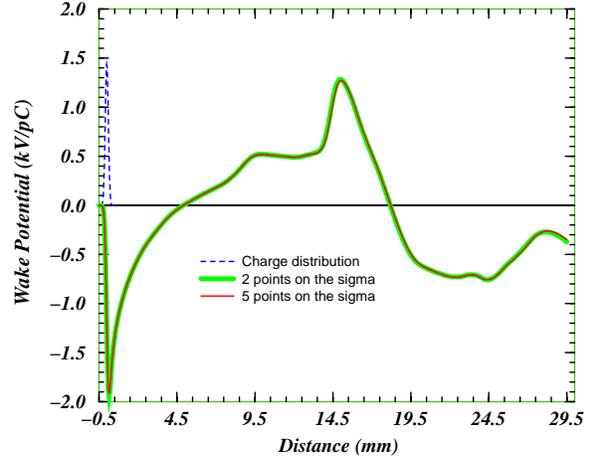


Figure 12: The long distance wake potential of the NLC section, calculated by 5 and 2 mesh points on the bunch length.

is constant

$$\Phi(r, z) = const$$

As an additional example, on Fig.13 the picture of electric force lines of the 100  $\mu$  bunch at the end of the NLC section is presented.

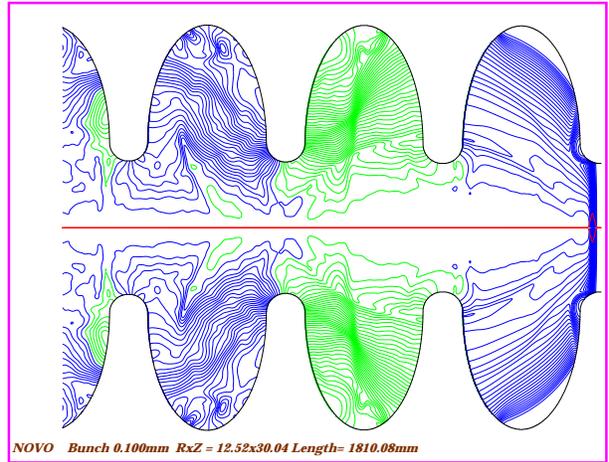


Figure 13: End of the NLC section. Electric Force lines of the field, excited by the 100  $\mu$  bunch.

## 6 ACKNOWLEDGEMENTS

The authors greatly appreciate useful discussions with Reinhard Brinkmann and Jorg Rossbach

## 7 REFERENCES

- [1] The MAFIA collaboration "User Guide", CST GmbH, Darmstadt, Germany
- [2] A.N.Novokhatski and A.Mosnier, "Short Bunch Wake Potentials for a Chain of TESLA Cavities", DAPNIA/SEA-96-08, 1996.

# CALCULATIONS OF THE SHORT-RANGE LONGITUDINAL WAKEFIELDS IN THE NLC LINAC

K.L.F. Bane, SLAC, Stanford, USA; A. Mosnier, Saclay, France;  
A. Novokhatski, TH Darmstadt, Germany; K. Yokoya, KEK, Tsukuba, Japan

*Abstract*

Using two frequency domain and one time domain numerical approaches, we calculate the short-range longitudinal wakefield of the NLC linac accelerating structure, and find that the results agree to  $\sim 5\%$ . We show that our results are consistent with an analytical formula for the impedance at high frequencies. We, in addition, obtain through fitting a simple formula for the short-range wakefield of a linac structure that can be useful in designing linear colliders. Finally, we demonstrate that for the NLC linac cavity the effects on the short-range wake of end conditions, tapering, and rounding of the irises are small.

## 1 INTRODUCTION

In the Next Linear Collider (NLC)[1] trains of short, intense bunches are accelerated through the linac on their way to the collision point. In the linac the transverse modes of the accelerating cavities will be damped and detuned to control long-range wakefield effects. The dominant, remaining current-dependent effects will be those due to the short-range wakefields. A calculation of short-range wakefields in the NLC linac structure has been given in Ref. [2]. However, due to the difficulty in obtaining these functions accurately to the very short distances required—or equivalently, the impedances to the very high frequencies required—we revisit in this report the earlier results, and compare them with those of other calculation methods.

For our purposes an accelerating cavity of the NLC linac can be modeled by a periodic structure. For a periodic structure at high frequencies, the real part of the longitudinal impedance varies as  $\omega^{-3/2}$  and the imaginary part as  $\omega^{-1}$ [3, 4, 5]; correspondingly, the wakefield at the origin  $W_L(0) = Z_0 c / (\pi a^2)$ [5], with  $Z_0 = 377 \Omega$  and  $a$  the iris radius, and for short distances  $s$ ,  $W'_L(s) \sim s^{-1/2}$ . To obtain the short range wakefields of a periodic structure, according to the method used in Ref. [2] (see also[6]), the impedance is first obtained over a finite frequency range through field matching. The high frequency portion is taken to be given by the so-called Sessler-Vaynshtein optical resonator model[3], a model that asymptotically satisfies the appropriate power law. The resulting function is then Fourier transformed to obtain the short-range wakefield.

In this report a second method[7] will be applied to the problem, one that uses a similar approach, though with the impedance calculated along a path slightly shifted from the real frequency axis. This impedance function is much smoother: it is easier to study its asymptotic behavior and easier to Fourier transform. A third method[8], one that uses direct, time domain integration to obtain the wake-

field is also applied. This method has the advantage of being able to find wakes of non-periodic structures. We will, in addition, compare the numerical results to a high-frequency analytical formula due to Gluckstern[5]. Note that we designate the three numerical methods as, respectively, the frequency domain (FD), the complex frequency domain (CFD), and the time domain (TD) approach.

In this report we will also perform parameter studies for a periodic structure, to obtain a simple formula for the wake that may be useful in designing a linear collider. Finally, again using the example of the NLC linac cavity, we study the effects of the rounding of the irises, and the effects of non-periodic features, such as the variation in cell geometry, and the end conditions.

## 2 FREQUENCY DOMAIN METHODS

The NLC accelerating cavity is a 206 cell damped, detuned structure (DDS) operating at 11.4 GHz. It is a disk-loaded structure, with constant period  $L (= 8.75 \text{ mm})$ , and gradually varying gap  $g$  and minimum iris radius  $a$  (the iris edges are rounded). In the version named DDS1, for example, the change in  $a$  follows a Gaussian distribution with rms 2.5%, truncated at  $\pm 2\sigma$ , and  $g$  varies from 7.75 mm to 6.75 mm. The middle cell has dimensions  $a = 4.724 \text{ mm}$  and  $g = 7.25 \text{ mm}$ . Note that since the NLC linac bunch length is very small (with rms  $\sigma_z = 0.1 \text{ mm}$ ), the bunch “sees” only the irises, and the outer cavity shape plays no role in the wakefield. Let us begin by considering a purely periodic model of an NLC cavity. We choose the same model as was used in Ref. [2], *i.e.* one with squared, not rounded, iris edges, and with dimensions  $a = 4.924 \text{ mm}$ ,  $g = 6.89 \text{ mm}$ , and  $L = 8.75 \text{ mm}$ .

According to the FD method, the wave numbers  $k_n$  and the loss factors  $\kappa_n$  for a few hundred modes are obtained by field matching, and the high frequency dependence of the impedance is given by the optical resonator model. The real part of the impedance becomes

$$R_L = \sum_{n=1}^N \frac{\pi \kappa_n}{c} \delta(k - k_n) + \frac{2Z_0 j_{01}^2}{\pi L \zeta^2} \times \quad (1)$$

$$\times \frac{\sqrt{\nu} + 1}{(\nu + 2\sqrt{\nu} + 2)^2} \Theta(k - k_N) \quad k > 0 \quad ,$$

with  $j_{01} = 2.41$ ,  $\zeta = 0.824$ ,  $\nu = 4a^2 k / (\bar{L} \zeta^2)$ , with  $c$  the speed of light and  $\bar{L} = \sqrt{Lg}$ ;  $\Theta(x) = 0$  for  $x < 0$ , 1 for  $x > 0$ . The resonator model combines the power spectrum at the iris edge in the primary field of the beam with diffraction at the edges of a periodic array of thin, circular mirrors. It is a simple model but it has been observed to agree well

with numerical results. The real part of the impedance of our model as obtained by the FD method, with 270 modes averaged over frequency bins, is given by the histogram in Fig. 1. The optical resonator asymptote is given by a dotted curve. We see that at higher frequencies the two agree.

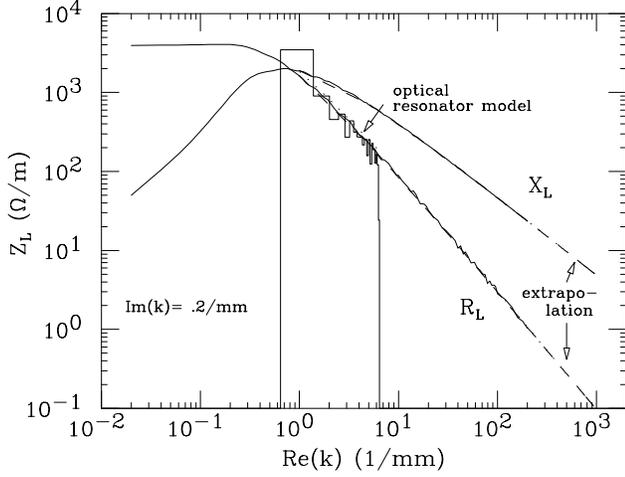


Figure 1:  $R_L$ , when averaged over bins, as given by the FD method (histogram);  $R_L$  and  $X_L$  as given by the CFD method (solid lines) and by Eq. 2 with  $\alpha = 0.52$  (dashes).

The CFD method also finds the impedance by field matching, but along a path slightly shifted off the real  $k$  axis. In Fig. 1 the real ( $R_L$ ) and imaginary ( $X_L$ ) parts of the impedance  $Z_L$ , as obtained by this method, when  $k_I = \text{Im}(k) = 0.2 \text{ mm}^{-1}$ , are shown (the solid curves). We note that the impedance is a relatively smooth function (instead of a sum of delta functions as before), and that the results agree well with those of the FD method and with the optical resonator model. Note that with the CFD method we can accurately go to much higher frequencies than before, and can therefore better study the asymptotic behavior of the impedance. For example, with the FT method, since the density of modes varies  $\sim k$ , we would need to solve for  $3 \times 10^5$  modes to obtain  $Z_L$  up to  $k = 200 \text{ mm}^{-1}$ . We should point out, however, that even with the CFT method to get convergence in the solution at high frequencies, the size of the matrix we need to solve becomes very large: at  $k_R = 200 \text{ mm}^{-1}$  its size is  $\sim 600 \times 600$ .

Gluckstern gives the high frequency behavior of the impedance of a periodic structure as[5]:

$$Z_L \approx \frac{iZ_0}{\pi k a^2} \left[ 1 + (1+i) \frac{\alpha L}{a} \left( \frac{\pi}{kg} \right)^{1/2} \right]^{-1} \quad [k \text{ large}], \quad (2)$$

with the parameter  $\alpha = 1$ . It can be shown, however, that  $\alpha$  is a function of  $g/L$ , with  $\alpha(0) = 1$  and  $\alpha(1) = 0.46$ [9]. (Note that 0.46 is the same numerical factor that has been obtained by Stupakov for a periodic array of infinitesimally thin irises[10].) For normal structures ( $g/L \approx 1$ ),  $\alpha \approx 0.5$ . Eq. 2 with the appropriate  $\alpha$  for our dimensions, 0.52, is shown in Fig. 1 by the two dashed curves. We note good

agreement with the CFD results at high frequencies.

## 2.1 Longitudinal Wakefield

In the FD method the short-range longitudinal wakefield  $W_L(s)$  is obtained by inverse Fourier transforming the impedance. The same is true in the CFD method, except that the result must also be multiplied by the factor  $\exp(k_I s)$ . The results are shown in Fig. 2. There is about a 5% disagreement between the two results. One check is that  $W_L(0)$  should equal  $Z_0/(\pi a^2)$ . The result of the FD method is 6% low, that of the CFD method is 1% low.

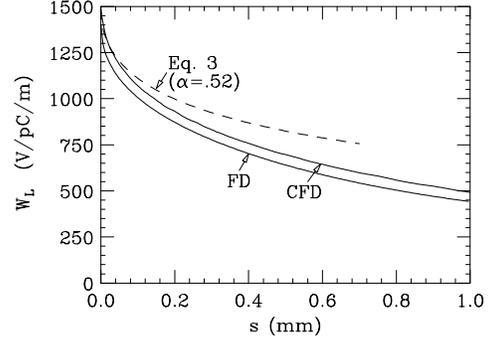


Figure 2: The wakefield of our periodic model, as obtained by different frequency domain methods.

If we inverse Fourier transform Eq. 2 we obtain a prediction for the very short-range wakefield:

$$W_L \approx \frac{Z_0 c}{\pi a^2} \exp\left(\frac{2\pi\alpha^2 L^2 s}{a^2 g}\right) \text{erfc}\left(\frac{\alpha L}{a} \sqrt{\frac{2\pi s}{g}}\right) \quad [s \text{ small}] \quad (3)$$

(given by the dashes in Fig. 2). We see that the approximate result, Eq. 3, agrees well with the CFD result for very short distances, for  $s \lesssim 50 \mu\text{m}$ .

## 2.2 Parameter Study

For designing linear colliders it would be useful to have a simple approximate formula for the short-range wakefield of a periodic structure that is valid, say, up to  $s/L = .15$  ( $s = 1.3 \text{ mm}$  for the NLC), over possible values of  $a$  and  $g$ . For this purpose we repeat the CFD calculation for parameters in the region  $.34 \leq a/L \leq .69$  and  $.54 \leq g/L \leq .89$ . Anticipating the functional form

$$W_L = \frac{Z_0 c}{\pi a^2} \exp\left(-\sqrt{s/s_0}\right), \quad (4)$$

we plot in the left frame of Fig. 3 the values of  $s_0$  fitted to the numerical results (the plotting symbols). We find the data is reasonably well reproduced by taking

$$s_0 = 0.41 \frac{a^{1.8} g^{1.6}}{L^{2.4}} \quad (5)$$

(the dashes in the figure). In the right frame we plot the wakes and the model result for four examples at the corners of our parameter plane. Note that a similar, though different, wakefield model has been proposed in Ref. [11].

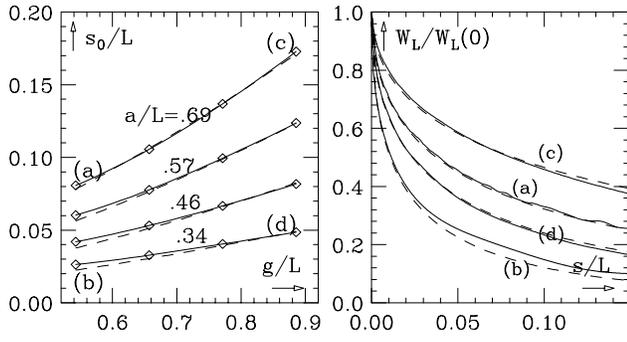


Figure 3: Results of our parameter study.

### 3 TIME DOMAIN CALCULATIONS

The TD method that we employ uses direct time domain integration of Maxwell's equations to obtain the wakefield. It uses an implicit method to solve finite difference equations, taking care to avoid dispersive errors that tend to occur in mesh-based programs at high frequencies. The program finds the wake of a bunch distribution (typically a Gaussian),  $\bar{W}_L$ . It has been used successfully for cases with extremely small bunch lengths, such as in TESLA-FEL, where  $\sigma_z/a \sim 10^{-3}$ [8]. The bunch wake is connected to  $W_L$  through

$$\bar{W}_L(s) = - \int_0^\infty W_L(s') \lambda(s-s') ds' \quad , \quad (6)$$

with  $\lambda(s)$  the charge distribution. In our simulations we will use the nominal NLC bunch length,  $\sigma_z = 0.1$  mm.

First, for our periodic example we let the bunch continue through identical cells until the wake per cell no longer changes. The result of the TD simulation is given in Fig. 4, and compared with that of the CFD method, after it has been convolved according to Eq. 6 (the dashes). The results are almost identical. When comparing total loss factors  $\kappa_{tot}$  the results for the TD, CFD, and FD methods are, respectively, 545, 547, and 512 V/pC/m.

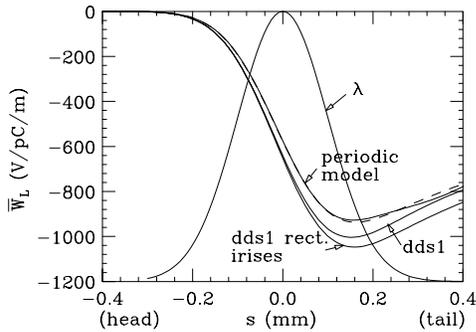


Figure 4: Results of the time domain (TD) calculations.

Features of the DDS1 cavity that are not in our periodic model are transients at the beginning of the cavity, the tapering of the cell geometry in the cavity, and the rounding of the iris edges, the effects of which we can explore with the TD method. As to the transients, it can be

shown[4, 5, 12] that for a finite number of cells greater than a certain critical number  $N_{crit} \gtrsim a^2/(2L\sigma_z)$  the average wake per cell of the finite structure agrees to within a few percent with that of the periodic structure. In our case this corresponds to only 14 cells (out of 206), so this should not be a significant effect. We can also estimate the effect of the tapering on the wakefield. For example, let us consider the effect on  $\kappa_{tot}$ , which for the short NLC bunch scales  $\sim a^{-2}$ . If we integrate this scaling over the Gaussian distribution in  $a$ , we find that over a whole cavity (assuming we can ignore the transients)  $\langle \kappa_{tot} \rangle = \kappa_{tot}(\bar{a})(1 + \sigma^2)$ , with  $\bar{a}$  the average iris radius and  $\sigma$  the rms of the distribution in  $a$  ( $= 2.5\%$ ). That is, the expected result is nearly the same as for a periodic structure with dimension  $a = \bar{a}$ .

We have performed a TD calculation for an entire, tapered DDS1 cavity, once with squared irises and once with the actual (rounded) iris shapes (see Fig. 4). Comparing loss factors, we obtain, for the squared irises,  $\kappa_{tot} = 617$  V/pC/m, which is 13% larger than for the periodic model. However, remember that in the periodic model  $a = 4.924$  mm, which is 4% larger than  $\bar{a}$  in the actual structure, so considering the  $a^{-2}$  scaling of the wakefield there is only a 5% discrepancy unaccounted for, which could be due to the end conditions and/or the tapering. Finally, for the actual DDS1 geometry, *i.e.* with rounded irises, we obtain  $\kappa_{tot} = 601$  V/pC/m, a 3% smaller result.

We conclude that for the NLC parameters, neither the end effects, nor the tapering, nor the rounding of the irises have much effect on the wakefield. More in particular, we also conclude that the longitudinal wakefield obtained through the FD method in Ref. [2], and meant to represent the DDS1 structure, is 15% low, 8% of which is due to not having used the average cell geometry in the calculation.

### 4 REFERENCES

- [1] "Zeroth-Order Design Report for the Next Linear Collider," SLAC Report 474 (1996).
- [2] K. Bane, SLAC-NLC-Note 9, February 1995.
- [3] E. Keil, *Nucl. Instr. Meth.* **100**, 419 (1972); D. Brandt and B. Zotter, CERN-ISR/TH/82-13 and LEP Note 388 (1982).
- [4] S. Heifets and S. Kheifets, *Phys. Rev. D* **39**, 960 (1989).
- [5] R. Gluckstern, *Phys. Rev. D* **39**, 960 (1989).
- [6] K. Bane and P. Wilson, Proceedings of the 11<sup>th</sup> Int. Conf. on High Energy Accelerators, CERN (Birkhäuser Verlag, Basel, 1980), p. 592.
- [7] K. Yokoya, KEK Report 90-21, September 1990, p. 142-150.
- [8] A. Novokhatskii and A. Mosnier, DAPNIA-SEA-96-08, November 1996.
- [9] K. Yokoya, unpublished result.
- [10] G. Stupakov, Proc. of IEEE Part. Accel. Conf., Dallas, 1995, p. 3303.
- [11] A. Novokhatskii and A. Mosnier, "Wakefield Dynamics in Quasi-Periodic Structures," PAC97, Vancouver, 1997.
- [12] K. Bane, *et al*, DESY-M-97-02, January 1997.

# THE SURFACE ROUGHNESS WAKEFIELD EFFECT

A. Novokhatski, M. Timm and T. Weiland,  
TU-Darmstadt, TEMF, Schlossgartenstr. 8, 64289 Darmstadt, Germany\*

## Abstract

In the Linear Colliders FEL projects ultra short bunches are foreseen to be used. In addition to usual wakefields, coming from irregularities in the chamber, these bunches excite fields in transporting lines and undulators due to the surface roughness. This effect can be large for the extruded tubes, usually used in accelerators. Based on computer results it is shown, that the roughness wakefield effect can be described by a simple model for the monopole and dipoles wakefields of a tube with thin dielectric coating.

## 1 INTRODUCTION

The surface roughness wakefield is the field, excited by a bunch traveling at the speed of light in a vacuum chamber with a rough wall surface. These wakefields might play a crucial role for the Linear Colliders and Free Electron Lasers(FEL). Since it is required, that the longitudinal and the transverse emittance is kept small, every additional contribution has to be studied carefully. The roughness depth of an extruded aluminum pipe is in the order of  $0.5 \mu\text{m}$  in average or even  $3 \mu\text{m}$  peak to peak. In FEL operation the bunchlength is below  $25 \mu\text{m}$ . The bunch samples the surface structure of the tube. Wakefields due to the rough surface of the vacuum chamber influence the longitudinal as well as the transverse beam dynamics.

It is shown that the phase velocity of the fundamental tube mode is decreased by the disturbance of a manufacture roughness to the speed of light. Accordingly there is a synchronous wave accompanying the Bunch, which is called the rough tube mode.

## 2 ANALYTICAL APPROACH

To approach a description of the surface roughness effect a cylindrical tube with radius  $a$  is assumed. The boundary of this tube is disturbed by a surface structure with the depth  $\delta$  (Fig. 1). For structures as shown in Fig. 1.I) and Fig. 1.II) the dispersion curve can be calculated easily.

### 2.1 Dispersion of the Fundamental Rough Tube Mode

The surface roughness of the tube decreases the phase velocity of the fundamental mode. The speed of light curve and the dispersion curve are nearly parallel for a wide range of phase advances per cell (Fig. 2). The first higher mode, with a radial dependency of 1, behaves nearly like a mode in a smooth tube. The phase velocity approaches the speed

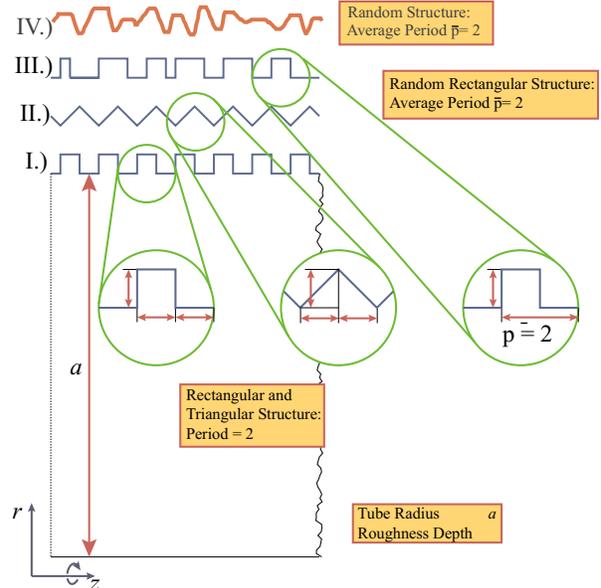


Figure 1: A cylindrical tube with 4 models of surface roughness: I.) periodically rectangular, II.) periodically triangular, III.) random with rectangular shape, IV random in longitudinal and radial direction. The tube radius is  $a$ , the depth of the roughness is  $\delta$ , the period (I., II.) or average period (III.) respectively is  $2 \cdot \delta$

of light curve, but does not cross it. Consequently only the fundamental mode contributes to the wakefields.

The diagram of phase and group velocity Fig. 3 shows, that the curve of the rough surface mode crosses the speed of light line at an arbitrary single, frequency. The group velocity does not reach this line. The relative difference between group velocity  $v_{gr}$  and speed of light  $c$  determines the length of the wake field pulse

$$\Delta t = \frac{c - v_{gr}}{c} \cdot L \quad (1)$$

where  $L$  is the length of the vacuum chamber.

### 2.2 Dielectric Layer Model

To calculate the wakefields inside a tube the model of a wave guide covered with a thin dielectric layer is used. The applicability of this model to tubes with corrugations was demonstrated in [1]. It has been shown that this approach is extendable to the transverse wakefields created by a rough surface [2].

\* Work supported in part by DESY, Hamburg, Germany

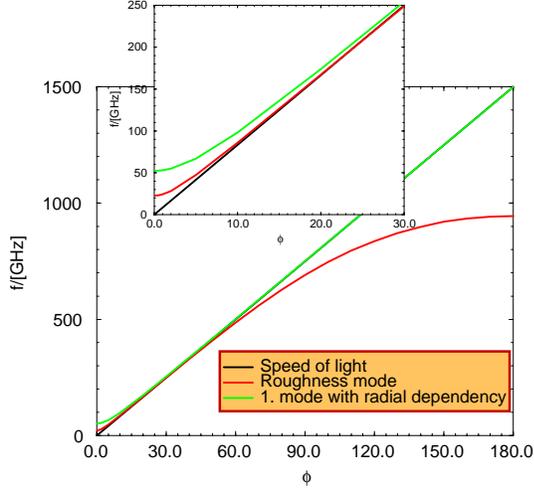


Figure 2: Dispersion diagram of the rough tube mode (structure as in Fig. 1.I). The radius of the tube is  $a = 5$  mm. The period of the roughness is  $100\mu\text{m}$ , the roughness depth is  $\delta = 50\mu\text{m}$ . The frequency is plotted against the phase advance  $\phi$  per cell. The roughness mode curve crosses the speed of light curve at  $\approx 200$  GHz.

### 2.2.1 Monopole Case

In the monopole case ( $m = 0$ ) the wave number of a wave guide covered with a thin dielectric layer is given by

$$k_0^2 = \frac{2\varepsilon}{(\varepsilon - 1)a\delta} \quad (2)$$

where  $\varepsilon$  is the relative permittivity,  $a$  the tube radius and  $\delta$  the thickness of the dielectric layer. The longitudinal wakefunction is

$$W_0^{\parallel}(s) = \frac{Z_0 c}{\pi a^2} \cos(k_0 s) \quad (3)$$

using the impedance of free space  $Z_0 = \sqrt{\mu_0/\varepsilon_0}$ .  $c$  denotes the speed of light.

### 2.2.2 Dipole Case

In the dipole case the wave number is the same,

$$k_1^2 = \frac{2\varepsilon}{(\varepsilon - 1)a\delta} \quad (4)$$

as in the monopole case. The longitudinal wakefunction is given by:

$$W_1^{\parallel}(s) = 2 \left(\frac{r_0}{a}\right) \left(\frac{r_1}{a}\right) \frac{Z_0 c}{\pi a^2} \cos(k_1 s) \quad (5)$$

where  $r_0$  and  $r_1$  are the offset of the driving charge and the witness respectively. Note that for  $r_0 = r_1 = a$  the amplitude of the longitudinal dipole wakefield is twice as large as in the monopole case. Finally the transverse dipole wakefunction reads:

$$W_1^{\perp}(s) = 2 \left(\frac{r_0}{a}\right) \frac{Z_0 c}{\pi a^3 k_1} \sin(k_1 s). \quad (6)$$

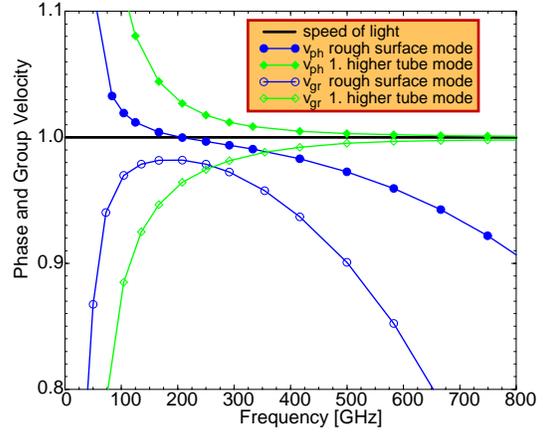


Figure 3: Dispersion diagram of the rough tube mode (structure as in Fig. 1.I). The radius of the tube is  $a = 5$  mm. The period of the roughness is  $100\mu\text{m}$ , the roughness depth is  $\delta = 50\mu\text{m}$ . The phase velocity  $v_{ph} = \omega/\beta$  and the group velocity  $v_{gr} = \partial\omega/\partial\beta$  are plotted against the frequency. The roughness mode curve crosses the speed of light curve at  $\approx 200$  GHz.

To describe the consequences of the transverse wakefields on the bunch in the transfer line the gradient  $G(s)$  of the wakefunction as the quotient of the transverse dipole wakefunction and the offset to the axis is introduced:

$$G(s) = \frac{W_1^{\perp}(s)}{r_0} \quad (7)$$

Assuming that a bunch of an energy  $E$  enters the tube at an offset  $r_0$  it will double its offset after a certain distance, to which in this paper is referred to as the instability length:

$$z_{inst} = \sqrt{\frac{E}{G(2\sigma_z)}} \quad (8)$$

### 2.2.3 Application of the Dielectric Layer Model to Surface Roughness

The applicability of the dielectric layer model to the longitudinal and transverse wakefields of a vacuum chamber with a rough surface modeled as shown in Fig. 1 has been demonstrated in [1, 2]. It is important to note, that neither the appearing of the rf-pulse nor its frequency depend on a strict periodicity of the structure [2].

### 2.2.4 Validity in Three Dimensions

There are many uncertainties in the transformation of this model to three dimensional problems, but the effective roughness depth in 3D is expected to be 3 times less than in 2D.

## 2.3 Normalized Description

The surface roughness wakefield, the loss parameter and the energy spread are given as functions of  $k_0\sigma_z$ . The fre-

quency  $k_0$  can be derived from the model above and is essentially a property of the tube and  $\sigma_z$  of course a bunch property.

The normalization is chosen in this way, that the maximum loss factor equals 1. The wakefunction is:

$$w_0^{\parallel}(z) = 2 \cos(k_0 \sigma_z z). \quad (9)$$

The wakepotential derived from this wake is

$$W_0^{\parallel}(z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^z e^{-\frac{y^2}{2}} \cdot \cos(k_0 \sigma_z (z - y)) dy. \quad (10)$$

Thus the normalized lossfactor is:

$$H(k_0 \sigma_z) = e^{-(k_0 \sigma_z)^2}. \quad (11)$$

It gives the energy transported by the rf-wave traveling along the tube. The normalized energy spread is:

$$\Delta(k_0 \sigma_z) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} W_0^{\parallel}(z) e^{-\frac{z^2}{2}} dz - e^{-2(k_0 \sigma_z)^2} \quad (12)$$

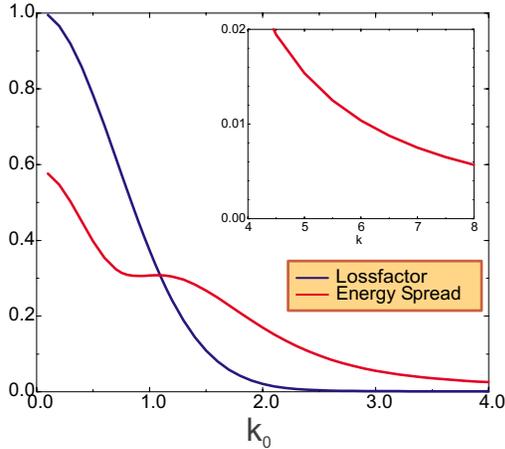


Figure 4: Loss factor  $H$  and Energyspread  $\Delta$  due to the rough tube mode in normalized description.

Depending on the value of  $k_0 \sigma_z$  the wakefields act capacitive  $k_0 \sigma_z \approx 0$  or inductive  $k_0 \sigma_z \gg 1$ . Fig. 4 shows a flat top of the Energyspread in the region  $0.75 < k_0 \sigma_z < 1.25$ . The energy spread stays the same while the loss factor, and thus the energy of the rf-pulse is decreasing.

The amplitude of the wakefield decreases in this region of  $k_0 \sigma_z$ . It is the transition from capacitive to inductive wake characteristic. The tail of the bunch is now accelerated. Therefore the energy spread is not changing.

#### 2.4 Dielectric Layer Model and Normalized Description

To calculate the lossfactor and the energyspread from the normalized description the frequency  $k_0$  has to be determined according to eq. 2 Assuming a radius  $a = 2\text{mm}$  and

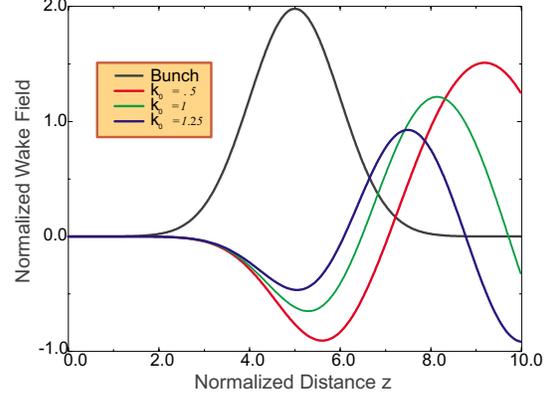


Figure 5: The wakefields in the flat top region of the normalized energy spread (compare Fig. 4).

a roughness depth  $\delta = 100\mu\text{m}$ , the permittivity of the dielectric layer is found to be around  $\epsilon \approx 1.9$  in cases Fig. 1.I.) and Fig. 1.III.) and  $\epsilon \approx 1.4$  in case Fig. 1.II.). Note that the equivalent permittivity depends on different parameters as e.g. the roughness shape.

Subsequently the value of the lossfactor and the energy spread can be calculated by

$$k_{loss} = \frac{Z_0 c}{2\pi a^2} H(k_0 \sigma_z) \quad (13)$$

and

$$\Delta E = \frac{Z_0 c}{2\pi a^2} \Delta(k_0 \sigma_z) \quad (14)$$

respectively.

### 3 NUMERICAL RESULTS

As an example of the application of the dielectric layer model a tube with a radius of  $a = 2\text{mm}$  and a roughness depth of  $\delta = 20\mu\text{m}$ , modeled as Fig.1, is taken. The bunch length  $\sigma_z = 250\mu\text{m}$  is more than 10 times larger than the gaps of the surface roughness.

The longitudinal wakefield resulting from the numerical calculation, is compared to the wakefield, as derived by convolution from the analytical solution 6. The 2 curves show good agreement. The relative permittivity is  $\epsilon_r = 1.515$ . The longer the calculated tube is, the more the amplitude of the wakefield approaches the analytic curve.

The wakefield created by a  $\sigma_z = 250\mu\text{m}$  bunch in side a tube with a surface modeled randomly as well in longitudinal and radial direction is used as another example. The tube radius is  $a = 5\text{mm}$ , the mean value of the random distribution is  $50\mu\text{m}$  in radial and longitudinal direction.

The field lines of the electric field 7 derived by the numerical simulation in the time domain. show a harmonic oscillating field. The wave length of the field is much higher than the period of the surface roughness. The size of a single roughness cavity does not correspond directly to the wave length. This is in agreement to the fact, that the strict periodicity is not necessary. The random distribution

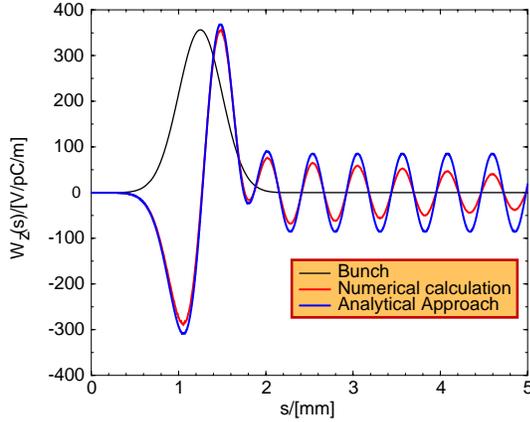


Figure 6: Comparison of the analytical approach to numerical calculations. Wakefield of a bunch with the length of  $\sigma_z = 250\mu\text{m}$ , carrying a charge of  $1\text{pC}$  is passing a tube with radius  $a = 2\text{mm}$ . The structure of the surface is modeled as Fig. 1 period of the roughness is  $p = 40\mu\text{m}$ , the roughness depth is  $\delta = 20\mu\text{m}$ .

does not lead to any decoherence of the rf-pulse following the bunch.

#### 4 APPLICATION OF THE MODEL OF SURFACE ROUGHNESS

The derived theory is now applied to some components of Linear Colliders and Free Electron Laser. As an example the numbers of a three tubes are given. One with a comparatively large diameter and a very smooth surface. with two different roughness depth, and of a very narrow undulator chamber, where wakefields are apprehended in particular, are given.

	Tube 1	Tube 2	Undulator
Tube Radius [mm]	10	50	5
Rough. Depth $\delta[\mu\text{m}]$	0.5	0.1	0.5
$\sigma_z[\mu\text{m}]$	100	10	20
Q [nC]	0.5	1	1
frep [Hz]	200	10	5
Bunch Spacing [ns]	1	100	100
Pulse Length [ $\mu\text{s}$ ]	0.1	1000	1000

Table 1: Electron beam and geometric parameters of the examples for surface roughness wake fields. Numbers are given for a tube of 1 meter length.

Several assumptions are made to derive the number. The roughness depth is has a rectangular shape. For the equivalent permittivity  $\varepsilon$  the value 2 is taken. To calculate the average power different repetition rates of are assumed. This number as well as the bunch spacing and the pulse length, affects the average power only.

The application of the surface roughness model on some

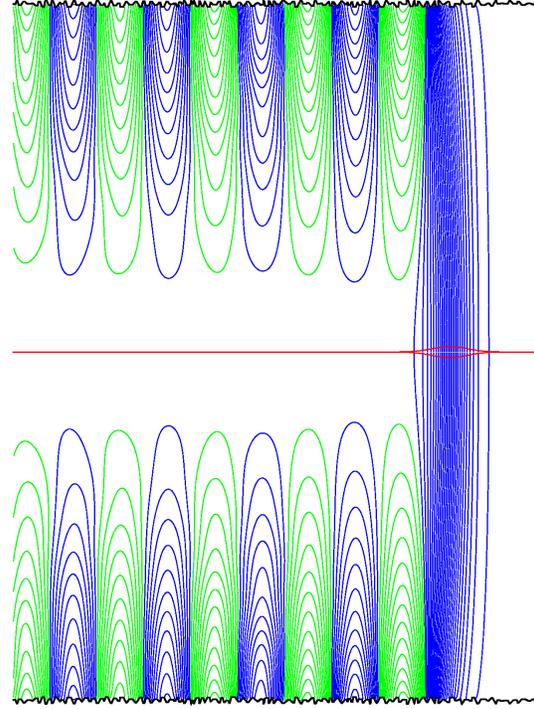


Figure 7: Field lines of the electric wakefield of a bunch with the length of  $\sigma_z = 250\mu\text{m}$ , carrying a charge of  $1\text{pC}$ . The radius of the tube is  $5\text{mm}$ . The average period of the roughness is  $\delta = 50\mu\text{m}$ , the average roughness depth is  $\delta = 50\mu\text{m}$ .

elements which might appear in future Linear Colliders and FEL's shows, that enormous peak powers are achieved even with surfaces usually regarded as smooth (Tab. 4). The peak power stays constant as the structure length increases, because the rf-pulse length depends on the structure length too. Thus the power stored in the rf-pulse grow proportional to the length. Note that the numbers given in Tab. 4 expected to be smaller in a three dimensional structure with an arbitrary shaped surface.

Nevertheless the effect on beam dynamics cannot be neglected.

Furthermore the utilization of this wakefields in the beam dynamics calculation in damping rings shows, that

	Tube 1	Tube 2	Undulator
Energyloss [kV]	24.3	6.6	381.3
Energyspread [MV]	49.8	3.6	216
$k_0\sigma$	1.41	0.28	0.8
Frequency [GHz]	675	1350	1900
Pulselength [fs]	333	13.3	670
Peak Power [MW]	72.9	496	570
Average Power [W]	0.486	0.66	19.4

Table 2: Wakefields due to the vacuum chamber roughness in several Linear Collider components. Parameters are listed in 4.

the saw tooth instability reported in many cases might be a consequence of the surface roughness[3].

## 5 RESISTIVE AND SURFACE ROUGHNESS WAKEFIELDS

The collective effect of the surface roughness and the resistive wall wakefield effect is studied. Therefore a periodic rectangular structure is chosen (Fig. 1 I.). The radius  $a = 2\text{mm}$  and the roughness depth  $\delta = 100\mu\text{m}$  stay unchanged for the comparison. For the conductivity  $\sigma$  several different values were assumed. The lowest conductivity  $\kappa = 132\text{ 1}/\Omega\text{m}$  corresponds to a skin depth of  $\delta_{skin} = 100\mu\text{m}$ , the same value as the roughness depth at the bunch frequency. In steps of 3 times the preceding conductivity,  $\sigma$  is increased, i.e. the skin depth is decreased by  $\sqrt{3}$ . Finally a perfect conducting material is used, as a comparison to the usual surface roughness wakefields.

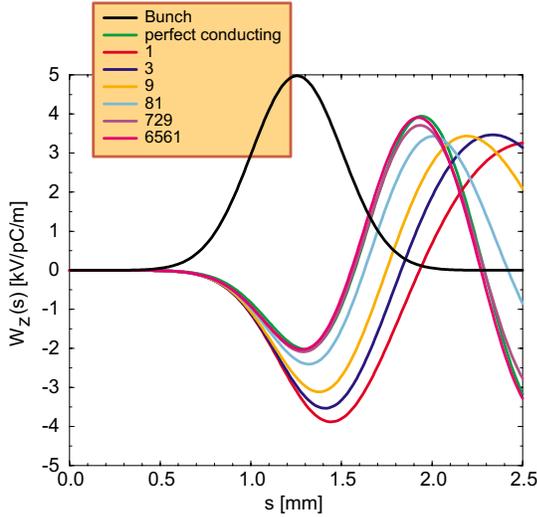


Figure 8: Wakefield of a rough resistive Tube. The tube radius is 2mm the roughness depth is  $100\mu\text{m}$ . The conductivity  $\kappa = 1321/\Omega\text{m}$  corresponds to a skin depth of  $\delta$  at the bunch frequency.

Fig. 8 shows the wakefields in the bunch region. The lower the conductivity is, the higher is the amplitude of the wakefield. The resistive wall and the rough surface effect fortify each other.

Regarding the wakefield in a longer range Fig. 9, the lower conductivity damps the wakefield strongly. The rf-pulse does not reach the length of the perfectly conducting case. The frequency of the pulse is lower. Compared with the description of the resistive wall wakefield in [4], the rough surface wake field is the dominating effect if

$$\delta > \frac{4}{3} \sqrt[3]{a} \left( \frac{c}{2\pi\sigma} \right)^{\frac{2}{3}}, \quad (15)$$

where  $\sigma$ , the conductivity, is  $5.4 \cdot 10^{17} \frac{1}{\text{s}}$ , for a copper tube,  $3.2 \cdot 10^{17} \frac{1}{\text{s}}$  for aluminum and  $1.3 \cdot 10^{16} \frac{1}{\text{s}}$  for stainless steel. In case of the aluminum undulator pipe (Tab. 4) the

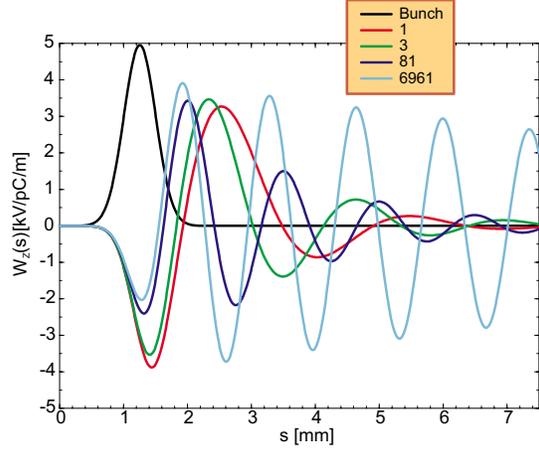


Figure 9: Wakefield of a rough resistive tube. The tube radius is 2mm, the roughness depth is  $100\mu\text{m}$ . The conductivity  $\kappa = 1321/\Omega\text{m}$  corresponds to a skin depth of  $\delta$  at the bunch frequency.

roughness depth  $\delta > 64\text{nm}$  and in case of the transfer line  $\delta > 138\text{nm}$ . In the example above Fig. 8, 9 the transition between the regimes is  $\approx 2.5\kappa$ .

## 6 CONCLUSION

This approach takes into account the accelerators vacuum chamber disturbed by a rough surface. There is a rough tube mode with the phase velocity equal to the speed of light accompanying the Bunch. The bunch does not experience every single detail of the surface corrugation, but averages over the faults. The wakefields due to this mode are large. Estimations of the influences are given.

## 7 ACKNOWLEDGMENTS

The authors want to thank Reinhard Brinkmann, Jörg Rossbach and Holger Schlarb for their interest in this work.

## 8 REFERENCES

- [1] A. Novokhatski and A. Mosnier, 'Wakefields of Short Bunches in the Canal Covered with thin dielectric layer', PAC'97, p. 1444, Vancouver, Canada, May 1997.
- [2] A. Novokhatski, M. Timm and T. Weiland, 'The Transverse Wake Fields in the TESLA-FEL Transfer Line', EPAC-98, p. 515, Stockholm, Sweden, June 1998.
- [3] A. Novokhatski and T. Weiland, 'Self-Consistent Model for the Beams in Accelerators', ICAP'98, Monterey Ca., USA, September 1998.
- [4] K.L.F. Bane and M. Sands, 'The Short-Range Resistive Wall Wakefields', SLAC-PUB-957074, USA, December 1995.
- [5] The MAFIA collaboration "User Guide", CST GmbH, Darmstadt, Germany.

# WAKE OF A ROUGH BEAM WALL SURFACE

K. L. F. Bane and G. V. Stupakov

SLAC, 2575 Sand Hill Road, Menlo Park, CA. 95014 USA

*Abstract*

We review and compare two models recently developed for the impedance calculation of a rough surface.

## 1 INTRODUCTION

Future linear accelerators and FELs tend to use short, intense bunches with small emittances and small energy spreads. For example, the design of the Linac Coherent Light Source (LCLS) at SLAC requires a bunch with a peak current of 3.4 kA, an rms bunch length of 30  $\mu\text{m}$ , a normalized emittance of 1 mm-mr, and an rms energy spread of 0.1% [1]. One concern in such machines is that induced wakefields may significantly increase the beam emittance or energy spread. It has been pointed out in [2, 3] that one major source of wakefields in machines with short bunches might be the roughness of the beam tube surface.

The model developed in Ref. [2] assumes that a rough surface can be represented as a collection of bumps of relatively simple shapes (hemispheres, half cubes, etc.), and the total impedance can be approximated as the sum of the impedances of the individual bumps. Recently, another approach has been developed [4], one using a small-angle approximation in the wall surface profile. It assumes that the wall surface discontinuities are gradual in the direction along the wall surface. In this approach the impedance of the rough surface is expressed in terms of the spectral function of the surface profile. The result represents the contribution of different scales, and can be used to estimate the impedance based on the statistical properties of the surface.

In this paper we review and compare the two approaches. Note that in both models we assume that the depth of the surface perturbations are large compared to the skin depth at the frequencies of interest, and that we can therefore ignore the effect of the resistance of the wall material. Note further that both models yield a total impedance that is inductive in character. Another model, one that says that the effect of a rough surface is similar to that of a thin dielectric layer, and that yields a resonator type of impedance [3], will not be discussed here. Finally, note that, for brevity, we consider here only the longitudinal impedance. In the case of the LCLS undulator beam tube, for example, it appears that this is the dominant wakefield effect. Once the longitudinal impedance is known, however, the transverse impedance of a rough surface on a cylindrical beam tube can be easily obtained, as is shown, for example, in Ref. [2].

## 2 SIMPLE MODEL

In the model developed in Ref. [2], it was assumed that a rough surface can be represented as a random distribution of small bumps and cavities of a certain size — the granularity size — on a smooth surface. Since the impedance of a small bump tends to be significantly larger than that of a cavity of similar size, the effect of cavity-like features was neglected. Then a rough surface can be represented as a collection of bumps, as sketched in Fig. 1. The longitu-

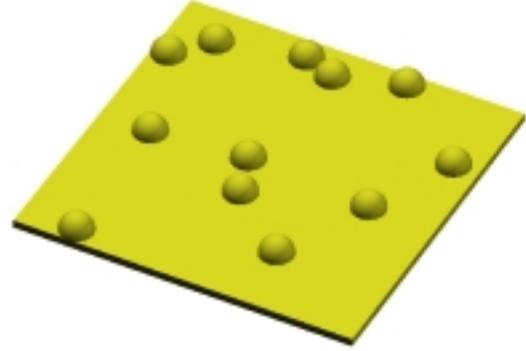


Figure 1: Rough surface is represented as collection of bumps of a given shape randomly distributed on the surface.

dinal impedance of a single hemisphere of radius  $r$  on the surface of a tube of radius  $b$  for  $\omega \ll r/b$  is given by [5]

$$Z_1(\omega) = -i\omega\mathcal{L} = -i\omega\frac{Z_0}{4\pi c}\frac{r^3}{b^2}, \quad (1)$$

where  $\mathcal{L}$  is the inductance,  $\omega$  the frequency,  $Z_0 = 377 \Omega$ , and  $c$  the speed of light. For a small object of a different shape the above formula needs to be multiplied by a form factor  $f$ . Numerically obtained form factors for some simple shapes are given in Table 1 [2]. By comparing the result for a cube and a half cube note a strong, roughly quadratic dependence of  $f$  on bump height.

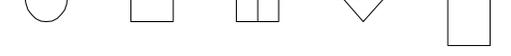
For many bumps, assuming they are separated by at least their size, the total impedance is approximated by the sum of the impedances of the individual bumps. The total impedance *per unit length* of the beam tube then becomes

$$Z(\omega) = -\alpha f \frac{iZ_0\omega}{2\pi c} \frac{r}{b}, \quad (2)$$

with  $\alpha$  a packing factor equal to the relative area on the surface occupied by the bumps. As we see, the longitudinal impedance in this model is purely imaginary (inductive).

Table 1: Form factors for 5 selected objects with the same base area. The figure at the bottom of the table shows the shapes of the respective objects, counted from left to right.

Case	$f$
Hemisphere	1
Half Cube	2.6
Rotated Half Cube	0.6
Wedge	1.1
Cube	10.8



In applications such as the LCLS undulator, an important parameter is the energy spread of the bunch, which can be increased due to the roughness impedance. For a Gaussian bunch with rms length  $\sigma_z \gg r$  the total rms energy spread induced by the roughness of the beam tube is given by [2]

$$\delta E_{rms} = N e^2 L W_{rms}, \quad (3)$$

where  $N$  is the number of particles in the bunch,  $L$  is the beam tube length, and

$$W_{rms} = -\alpha f \frac{c Z_0}{3^{1/4} 2^{3/2} \pi^{3/2}} \frac{r}{b \sigma_z^2}. \quad (4)$$

Using the above expression for the impedance, we can now estimate the effect of the roughness wake in the LCLS undulator using the following parameters: undulator length –  $L = 100$  m, beam charge –  $N e = 1$  nC,  $f = 1$ ,  $\alpha = 0.5$ ,  $\sigma_z = 30$   $\mu$ m,  $b = 3$  mm, beam energy –  $E = 15$  GeV. For the energy spread increase due to the wake  $\sigma_\delta < 0.05\%$ , the height of the bumps should be

$$r < 50 \text{ nm}. \quad (5)$$

If these parameters are accurate, then the requirement on the smoothness of the beam tube surface are severe.

### Small-Angle Approximation

The detailed derivation of the impedance in the small-angle approximation can be found elsewhere [4]. Here we outline the main assumptions and present the final result of this model.

The approach is based on the assumption that the angle between the normal to the rough surface and the radial direction is small compared to unity. If we assume that the rough surface is given by the equation  $y = h(x, z)$ , where  $x$ ,  $y$  and  $z$  are the cartesian coordinates, and  $h$  is the local height of the surface, then the small-angle approximation means that

$$|\nabla h| \ll 1. \quad (6)$$

This assumption allows us to develop a rather general theory of the impedance, which gives good accuracy even when  $|\nabla h| \sim 1$ .

In addition to Eq. (6), we also require that the height of the bumps and their characteristic width  $g$  be small compared to the radius of the pipe  $b$ ,

$$g, |h| \ll b. \quad (7)$$

Evidently, this inequality is easily satisfied for realistic values of  $g$ ,  $h$  and  $b$ . Finally, because typically the size of the surface bumps  $g$  is on the order of microns, and the bunch length  $\sigma_z$  is on the order of at least tens of microns, we also assume that the characteristic frequency of interest  $\omega \sim c/\sigma_z$  is small compared to  $c/g$ ,

$$\omega \ll c/g. \quad (8)$$

Using approximations (6) – (8), one can show that for a single bump of arbitrary shape  $h_0(x, z)$  sitting on the surface of a round beam pipe, the impedance is

$$Z_1(\omega) = -\frac{ikZ_0}{b^2} \int_{-\infty}^{\infty} \frac{\kappa_z^2 |\hat{h}_0(\kappa_z, \kappa_x)|^2}{\sqrt{\kappa_x^2 + \kappa_z^2}} d\kappa_z d\kappa_x, \quad (9)$$

where  $\hat{h}_0$  is a two dimensional Fourier transform of the bunch shape:

$$\hat{h}_0(\kappa_z, \kappa_x) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} h_0(x, z) e^{-i\kappa_z z - i\kappa_x x} dz dx, \quad (10)$$

where the  $z$ -axis is directed along the pipe axes, and the  $x$  axis is locally directed along the azimuthal coordinate  $\theta$ . We note, that due to assumed smallness of the surface structures, we can use the local Cartesian coordinate system  $x$ ,  $y$  and  $z$  in Eqs. (9) and (10) instead of the global cylindrical coordinate system  $\theta$ ,  $r$  and  $z$ .

To describe a rough surface with a random profile, we assume that  $h(x, y)$  is a random function with zero average,  $\langle h(x, z) \rangle = 0$ . Statistical properties of such a surface are characterized by the correlation function  $K(x, y)$ ,

$$K(x - x', z - z') = \langle h(x', z') h(x, z) \rangle, \quad (11)$$

where the angular brackets denote averaging over possible realizations of  $h(x, z)$ . Eq. (11) implies that statistical properties of  $h(x, z)$  do not depend on the position of the surface. An important statistical characteristic of the roughness is the *spectral density* (or *spectrum*)  $R(\kappa_z, \kappa_x)$ , defined as a Fourier transform of the correlation function,

$$R(\kappa_x, \kappa_z) = \frac{1}{(2\pi)^2} \int dx dz K(x, z) e^{-i\kappa_x x - i\kappa_z z}. \quad (12)$$

If the surface is statistically isotropic (all direction in the  $x - y$  plane are statistically equivalent), the spectrum  $R$  depends only on the absolute value  $\kappa$  of the vector  $(\kappa_x, \kappa_z)$ ,  $\kappa = \sqrt{\kappa_x^2 + \kappa_z^2}$ ,  $R = R(\kappa)$ .

The main result of Ref. [4] is that the longitudinal impedance of a circular pipe of radius  $b_0$  with a rough

perfectly conducting surface characterized by the spectral function  $R(\kappa_x, \kappa_z)$  in the frequency range limited by the condition (8) is given by the following equation:

$$Z(\omega) = -\frac{ikZ_0L}{2\pi b} \int d\kappa_z d\kappa_x R(\kappa_x, \kappa_z) \frac{\kappa_z^2}{\kappa}, \quad (13)$$

where  $L$  is the length of the pipe.

The presence of the factor  $\kappa_z^2$  in the integrand of Eq. (13) means that the contribution to  $Z$  of roughness in longitudinal ( $z$ ) and azimuthal ( $x$ ) directions are different. For example, bellow-type variations on the surface have spectral components with  $\kappa_z \neq 0$  and  $\kappa_x = 0$ , and result in non-vanishing  $Z(\omega)$ . On the other hand, ridges going in the longitudinal direction generate a spectrum with  $\kappa_x \neq 0$  and  $\kappa_z = 0$ , and according to Eq. (13) do not contribute to  $Z(\omega)$ .

As an application of Eq. (13), we can calculate the impedance of a rough surface with a Gaussian spectrum

$$S(\kappa) = \frac{l_c^2 d^2}{2\pi} e^{-\kappa^2 l_c^2 / 2}, \quad (14)$$

where  $d$  is the rms height of the roughness and  $l_c$  is the correlation length in the spectrum. Performing the integration, one finds

$$\frac{Z(\omega)}{L} = -\frac{\sqrt{\pi}}{4\sqrt{2}} \frac{ikZ_0 d^2}{l_c b}. \quad (15)$$

It is seen, that the impedance not only depends on the rms height of the bumps, but also on the correlation length  $l_c$ . Increasing this lengths makes the impedance smaller for a given rms height of the roughness. Qualitatively,  $l_c$  can be considered as a typical transverse size of the bumps in the statistical distribution.

### 3 FRACTAL SURFACE

Another model of a rough surface is given by a power spectrum, limited at small wavelengths,

$$\begin{aligned} R(\kappa) &= A\kappa^{-q}, \text{ for } \kappa > \kappa_0, \\ R(\kappa) &= 0, \quad \text{for } \kappa < \kappa_0, \end{aligned} \quad (16)$$

where  $\kappa_0$  is the minimal value of the spectrum,  $q > 0$  is a power index, and  $A$  defines the amplitude of the roughness. For spatial scales much smaller than  $\kappa_0^{-1}$ , this surface gives an example of a fractal landscape with a fractal dimension  $q$ . The parameter  $\kappa_0$  can be related to the characteristic correlation length,  $l_c$ , of the random profile,  $\kappa_0 \sim \pi/l_c$ . We can also relate the factor  $A$  to the rms height  $d$  of the roughness,

$$d^2 = 2\pi \int_0^\infty \kappa d\kappa R(\kappa) = \frac{2\pi A}{q-2} \kappa_0^{2-q}. \quad (17)$$

For convergence of the integral it is required that  $q > 2$ . The shape of the surface for two different values of  $q$  obtained with a help of computer code described in [6] is shown in Fig. 2. It turns out, that increasing the value

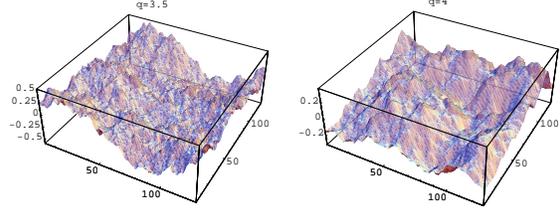


Figure 2: Fractal surfaces for  $q = 3.5$  and  $q = 4$ . Smaller values of  $q$  give more "spiky" profiles.

of  $q$  makes the surface smoother. Using Eq. (13) we can calculate the impedance of such a surface,

$$\frac{Z(\omega)}{L} = -\frac{ikZ_0}{4\pi b} \frac{q-2}{q-3} d^2 \kappa_0. \quad (18)$$

Again, for convergence, we require that  $q > 3$ , otherwise the integral diverges as  $\kappa \rightarrow \infty$ . This requirement is stronger than the convergence condition for Eq. (17), and is due to a relatively slow decay of the spectrum at large  $\kappa$ .

### 4 COMPARISON OF THE TWO MODELS

To compare the two models, we will calculate the impedance of a surface covered by bumps of a given shape, as illustrated by Fig. 1, using the small-angle approximation, and compare it with Eq. (2). For the sake of generality, we will assume an arbitrary shape of the bump given by the function  $h_0(x, z)$ . The bumps are randomly scattered over the surface, with the average number of bumps per unit area equal to  $\nu$ . We will assume that the average distance between the bumps,  $\nu^{-1/2}$ , is much larger than the transverse size of the bump  $g$ ; then we can neglect the events when bumps overlap.

Let us consider a square on the surface of size  $\mathcal{L} \times \mathcal{L}$ , where  $\mathcal{L}$  is large in comparison with the bump width  $g$ , but small relative to the pipe radius  $b$ , so that the effects of curvature are negligible. If this area contains  $N$  bumps, located at positions  $(x_n, z_n)$ ,  $n = 1, 2, \dots, N$ , then the surface profile  $h(x, z)$  is a superposition of all  $N$  bumps,

$$h(x, z) = \sum_{n=1}^N h_0(x - x_n, z - z_n). \quad (19)$$

To calculate the spectrum  $R(\kappa_x, \kappa_z)$  needed in the small-angle approximation model, we will first find the correlation function  $K$ ,

$$\begin{aligned} K(\xi, \zeta) &= \langle h(x, z)h(x + \xi, z + \zeta) \rangle \\ &= \sum_{n,k=1}^N \langle h_0(x - x_n, z - z_n) \\ &\quad \times h_0(x + \xi - x_k, z + \zeta - z_k) \rangle. \end{aligned} \quad (20)$$

Since we neglect overlapping of the bumps, only terms with  $n = k$  contribute to the sum of Eq. (20)

$$K(\xi, \zeta) \approx \sum_{n=1}^N \langle h_0(x - x_n, z - z_n) \times h_0(x + \xi - x_n, z + \zeta - z_n) \rangle. \quad (21)$$

To perform averaging in Eq. (21), we will assume that the probability  $p(x_n, z_n)$  for the bump to be located at the point  $(x_n, z_n)$  within  $dx_n$  and  $dz_n$  does not depend on the position, and is equal  $p = \mathcal{L}^{-2}$ . This assumption corresponds to a uniform distribution of bumps on the surface. Then averaging means integration over the square,

$$\langle f(x, z) \rangle = \mathcal{L}^{-2} \int_{\mathcal{L} \times \mathcal{L}} dx dz f(x, z), \quad (22)$$

and it reduces Eq. (21) to

$$K(\xi, \zeta) = \frac{N}{\mathcal{L}^2} \int_{\mathcal{L} \times \mathcal{L}} dx dz h_0(x, z) h_0(x + \xi, z + \zeta). \quad (23)$$

From Eq. (21) it follows that the correlations function for the randomly distribute bumps is equal to the correlation for a single bump multiplied by the bump density  $\nu = N/\mathcal{L}^2$ . Correspondingly, the spectral function  $R$  is

$$R(\kappa_x, \kappa_z) = (2\pi)^2 \nu |\hat{h}_0(\kappa_x, \kappa_z)|^2, \quad (24)$$

where  $\hat{h}_0$  is given by Eq. (10). Putting this correlation function into Eq.(13) gives

$$Z(\omega) = 2\pi b L \nu Z_1(\omega), \quad (25)$$

where  $Z_1(\omega)$  is given by Eq. (9). This equation tells us that the impedance of a rough surface consisting of a collection of identical bumps randomly scattered over the surface is equal to the impedance of a single bump multiplied by the number of bumps on the surface area. This result agrees with the approach used in the first model. Hence, the only difference between the two models in this limit is due to the calculation of the single bump impedance  $Z_1$ . Indeed, as shown in Ref. [4], for hemispheres, the small-angle approximation theory gives the result that is about two times smaller than the exact solution Eq. (1). Hence, for the rough surface, we will find that the two models agree within the factor of 2, with the small-angle theory giving a smaller impedance.

## 5 CONCLUSIONS

We have shown that the two models of roughness impedance investigated in this report have some similarities and some differences, and can be thought of as being complementary. They both are applicable only when the frequencies of interest are low compared to  $c/r$ , with  $r$  the typical size of the surface discontinuities, and both yield an approximation to the impedance that is purely inductive. The first model approximates a rough surface by a

random collection of non-interacting bumps. It finds the impedance of a single, small bump on a beam tube surface, and then uses averaging to estimate the impedance of a rough surface. The second model, through use of the spectral function of the rough surface analytically finds the impedance, though it is limited to surfaces with slowly varying discontinuities. In the specific case of a surface with non-interacting, smooth bumps the two models will give the same result.

The micro-geometry of a metallic surface—for example, the beam tube in the LCLS undulator—depends on the manufacturing and polishing process that had been applied to that surface. For either of the models discussed in this report to accurately estimate the impedance of a surface requires a specific characterization of the micro-geometry. Once such a characterization is performed, through measurement, one can begin to apply these models to obtain a realistic estimate of the surface impedance and derive conclusions about the effect of the impedance on beam dynamics.

## 6 ACKNOWLEDGMENT

This work was supported by Department of Energy contract DE-AC03-76SF00515.

## 7 REFERENCES

- [1] Linac Coherent Light Source (LCLS) Design Study Report. SLAC-R-521, Apr 1998. 381pp.
- [2] K.L.F. Bane, C.K. Ng and A.W. Chao. SLAC-PUB-7514 (1997).
- [3] A. Novokhatski and A. Mosnier, Proceedings of the 1997 Particle Accelerator Conf., Vancouver, p. 1661.
- [4] G. V. Stupakov, *Phys. Rev. ST-AB* **1**, 064401 (1998).
- [5] S. Kurennoy, *Phys. Rev.* **E55**, 3529 (1997).
- [6] R. E. Maeder. *The Mathematica Programmer II*, Academic Press, San Diego, 1996.
- [7] D. J. Whitehose. *Handbook of Surface Metrology*, IOP Publishing, 1994.

# CALCULATION OF IMPEDANCE FOR MULTIPLE WAVEGUIDE JUNCTION USING SCATTERING MATRIX FORMULATION

V.A.Dolgashev, Budker Institute of Nuclear Physics, Novosibirsk, 630090, Russia

## Abstract

A method of computing the electromagnetic characteristics of a complex 3D cavity consisting of series of waveguides with arbitrary cross section is derived. The scattering matrix formulation is used for the simulation. For calculations of modes in waveguide with arbitrary cross sections a finite element code SLANS is applied. The computer code based on the method is used to calculate scattering parameters, dispersion characteristics of periodical structures, resonances, longitudinal and transverse impedances. The method was originally developed for simulations of long range wake fields in accelerator structures and for calculations of RF windows. Advantages of the technology make possible to simulate long open cavities such as accelerating structures and complex vacuum chambers. The method and some results are presented.

## 1 INTRODUCTION

Design of modern accelerators requires extensive simulation of an influence of accelerator components on the beam. This influence can be characterised by such parameters as resonant frequency, resonant shunt impedance, coupling impedance. A lot of well established methods and computer codes were developed to obtain the parameters of RF cavities, periodical structures, and obstacles in a beam pipe. In general the methods can be divided into three parts. This are grid and finite element based methods, then mode-matching based and equivalent circuit based methods. Up to now, calculation of complex and 3D structures by the first methods is memory-space and computer time consuming problem. Equivalent circuit method is a very effective one but limited mostly in use for structures with simple coupling such as single-mode waveguides or lower passbands of accelerating structure. Mode-matching method has several advantages that make it useful for the simulations. The method permits to simulate complex 3D structures an efficient way and make qualitative analysis of the results simpler then the other methods. It can be used also for non-relativistic particles with curved trajectories. Results of simulation by grid or finite element codes can be naturally incorporated into the method. The method allows to calculate a structure partially as well. An example of such calculation is a matching of accelerator structure couplers. During the

matching one needs to calculate scattering matrix of the structure once and then recalculate only couplers.

The present paper discusses a mode-matching based method. The method uses a scattering matrix technology to determine coupling impedance, scattering parameters, resonant frequency, quality factor, and fields in a RF cavity, as well as periodical structure parameters. The code using the method was developed. The aim of the code was simulation of 3D structures such as RF windows and accelerating structures with 3D couplers. Some results were presented in [1]. The interactive code was written on object oriented language C++ and compiled for PC. The similar approach was developed by S.A.Heifets and S.A.Kheifets [2], and by Ursula Van Reinen[3] for calculation of rotationally symmetrical detuned accelerator structure.

## 2 SCATTERING MATRIX FORMULATION

Theory of mode-matching and multimode scattering matrixes is well known in electrical engineering and will be described briefly. Main steps of the calculation are: to build multimode scattering matrix for each waveguide junction, then to apply the scattering matrix technology to resolve characteristics of the whole structure, and then to post-process the resulted fields. The derivation begins by assuming an expansion of the transverse fields  $E_{\perp}$  and  $H_{\perp}$  in terms of the eigenmodes  $e_l$  in the waveguide as:

$$E_{\perp} = \sum_{l=1}^M (A_l + B_l) e_l, \quad (1)$$

$$H_{\perp} = \sum_{l=1}^M Y_l (A_l - B_l) e_l \times \vec{z}. \quad (2)$$

where  $A$  is a modal amplitude of the incident wave and  $B$  is the amplitude of the reflected wave (Fig.1), and  $Y$  is the characteristic wave admittance of the mode. Normalization of the eigenmodes was chosen so that the modes were orthogonal, that is,

$$\int e_l \cdot e_m ds = \delta_{lm}, \quad (3)$$

where  $\delta$  is Kronecker delta function. Longitudinal electric field is the sum over  $E$  (TM) modes :

$$E_z = \sum_{l=1}^{Me} i \frac{(k_l^e)^2}{\gamma_l^e} (A_l^e - B_l^e) e_{z_l}, \quad (4)$$

where  $k_l^e$  is cut-off wave value of the mode,  $\gamma_l^e$  is propagation value for the particular frequency, and  $ez_l$  is normalized as:

$$(k_l^e)^2 \int ez_l ez_l ds = 1. \quad (5)$$

Derivation of the eigenmodes can be done analytically only for simple geometries such as circle or rectangle. So, for arbitrary cross section another method has to be applied. The method should be able to calculate 2D flat scalar function that represents  $z$  - component of the field in a waveguide and correspondent cut-off frequency. In this particular case a finite element code SLANS [4] was used. After derivation of eigenmodes, applying continuity of fields in common aperture area yields the relation between incident and reflected waves i.e. scattering S matrix. Let us assume that right waveguide has smaller cross-section than the left one (Fig.1).

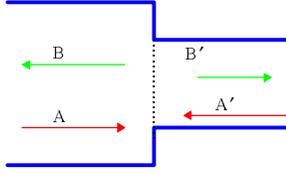


Figure 1: Modes in scattering matrix formulation

At first, we need to build two coupling matrices with the elements:

$$\eta_{ab}^{left} = \int e_a^{left} e_b^{right} ds, \quad (6)$$

where  $a = 1..Mleft$ ,  $b = 1..Mrigth$ ; The values of  $Mleft$  and  $Mrigth$  are governed by the relative convergence phenomena and strongly depend on the geometry.

$$\eta_{ab}^{right} = \delta_{ab}, \quad (7)$$

where  $a, b = 1..Mrigth$ , integration over common cross-section,  $e_a^{left}$  is the  $a$ -th eigenmode of the left waveguide and  $e_b^{right}$  is  $b$ -th eigenmode of the right waveguide. The integration can be provided either analytically, for simple cross-section, or numerically, for arbitrary ones. Next step is derivation of admittance matrices:

$$Y_{ml}^{left} = \sum_{a=1}^{Mleft} Y_a^{left} \eta_{am}^{left} \eta_{al}^{right}, \quad (8)$$

$$Y_{ml}^{right} = \sum_{b=1}^{Mrigth} Y_b^{right} \delta_{ml}, \quad (9)$$

where  $m, l = 1..Mrigth$ . Equation for the full admittance matrix is  $Y = Y^{left} + Y^{right}$ . Scattering matrix of the junction is:

$$S = \begin{pmatrix} (2Y^{-1}Y^{left})^T \eta^{left} - I & (2Y^{-1}Y^{left})^T \eta^{right} \\ (2Y^{-1}Y^{right})^T \eta^{left} & (2Y^{-1}Y^{right})^T \eta^{right} - I \end{pmatrix}. \quad (10)$$

There are several advantages of application of scattering matrices instead of transfer or ABCD matrices[5]. The first one is that on every step of the calculation we have parameters that can be easily measured by a network analyser. Another one is that by the method we can avoid numerical instabilities that appear for frequencies far above cut-off. Next steps in the calculation depend on required results. There are two methods implemented in the discussed code. One is cascading and other sparse matrix LU technology. Cascading leads to effective calculation of dense matrices, and it is used for calculation of scattering parameters and periodical structures. For the impedance calculation a sparse matrix technology [5] was used.

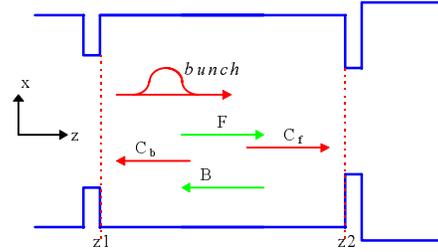


Figure 2: Excitation of a waveguide between two obstacles.

### 3 COUPLING IMPEDANCE

Intensity of beams in an accelerator is limited mostly by the coupling beam impedance. Therefore study and minimization of the impedance are important part of accelerator design. The impedance can be obtained analytically for rather simple geometries only, so numerical methods are applied. Application of time domain methods for 3D or long structures is limited by computing time. The use of cavity codes for the calculation of the impedance is limited by high-Q structures and can be carefully applied for frequencies above beam-pipe cut-off. At the same time properties of the scattering matrix approach allow one to calculate frequency dependent impedance for 3D long structures, frequencies below and above cut-off, and to simulate non-relativistic particles. Computation time for the method does not depend on the length of the structure. Let us consider a waveguide between two obstacles as shown on Fig.2.

The beam excites waveguide modes. The well known formula for the amplitude of the excited mode is [6]:

$$C_{b,f} = \frac{1}{N_s} \int_{z_1}^{z_2} \vec{j}_\omega \vec{E}_{b,f} dv, \quad N_s = 2 \int \vec{E} \times \vec{H} ds, \quad (11)$$

here indexes  $b, f$  denote the modes propagating along forward (+z) and backward (-z) direction,  $\vec{j}_\omega$  - beam current density,  $dv$  - integration over space from  $z1$  to  $z2$ ,  $ds$  - integration over waveguide cross-section,  $E$  and  $H$  - eigenmode fields. Using (11) amplitude of modes in the waveguide for  $z1$  coordinate we have:

$$B = (I - S^{right} S^{left})^{-1} (S^{right} S^{left} C_b + S^{right} C_f^{z1}),$$

$$F = S^{left} (B + C_b), \quad (12)$$

where  $S^{right}$ ,  $S^{left}$  are scattering matrices that resulted from cascading to the right and left from  $z1$ ,  $C_b$  is a vector of modes induced towards (-z) direction, and  $(C_f^{z1})_m = (C_f)_m e^{i\gamma(z1-z2)}$  is a vector of amplitudes excited forward but converted to  $z1$  coordinate. Longitudinal impedance for frequency  $\omega$  is given by:

$$Z_l(\omega) = \int_{-\infty}^{+\infty} E z_s(\omega) e^{-i \cdot k \cdot z} dz, \quad (13)$$

where  $k = 2\pi\omega/\beta c$ ,  $\beta c$  - beam velocity. Longitudinal electric field is:

$$E z_s(\omega) = E z_{mod}(\omega) + E z_{excit}(\omega) + E_{bunch}(\omega),$$

where  $E z_{mod}(\omega)$  follows from (4),  $E_{bunch} = j_\omega/i\omega$  and similar to the (4):

$$E z_{excit}(z) = \sum_{l=1}^{Me} i \frac{(k_l^e)^2}{\gamma_l^e} (C(z)_f - C(z)_b) e z_l. \quad (14)$$

The integration (13) is performed along a witness bunch trajectory. The impedance for structures with arbitrary cross-section has rather complex  $(x,y)$  dependence in the transverse plane[7]. Mapping of the impedance is used in the code. The  $Ez$  field is integrated along five parallel trajectories with coordinates:  $(x_w, y_w)$ ,  $(x_w + dx, y_w)$ ,  $(x_w, y_w + dy)$ ,  $(x_w - dx, y_w)$ ,  $(x_w, y_w - dy)$ . The transverse impedance for ultra-relativistic particles is obtained by taking the transverse derivative of  $Z_l(\omega)$ . Direct integration of the transverse forces can be used for non-relativistic particles. Sum of  $Z_l(\omega)$  over all waveguides, that form the whole structure, gives coupling impedance for the structure.

#### 4 THE USE OF CASCADING FOR IMPEDANCE CALCULATION

New technology of cascading of impedances was developed. The cascading is not restricted to "waveguide to waveguide" junction and then can be applied for arbitrary 3D geometries. The technology is based on separate calculation of parts of a structure and then combining them. The result of the cascading includes the

complete impedance as well as total scattering matrix. The parameters of a part of the structure are a frequency-dependent matrix related to mode-mode (scattering matrix), bunch-mode, mode-bunch, bunch-bunch interaction. This matrix can be obtained by any code, capable to calculate current-field and mode-mode dependencies. Finite-element or grid methods can be used for the purpose. The size of the matrix depends on quantity of waveguide modes and bunch trajectories and rarely exceeds  $100 \times 100$ . Because of the small matrix size the process of cascading requires modest computing time. The cascading is particularly effective for simulation of structures consisting of several identical parts such as periodical accelerating structures or vacuum chambers.

Let us suppose (for simplicity) that the structure consists of two identical parts with scattering matrix:

$$S = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}. \quad (15)$$

Vectors of amplitudes of modes excited by the bunch and values of field integrals caused by particular mode are:

$$V_{bm} = \begin{bmatrix} I_f \\ I_b \end{bmatrix}, \quad V_{mb} = \begin{bmatrix} F_f \\ F_b \end{bmatrix}.$$

Then, according to (12) amplitudes between the parts of the structure are:

$$B = (I - S_{11} S_{11})^{-1} (S_{11} S_{11} I_b \cdot e^{i \cdot k \cdot z} + S_{11} I_f),$$

$$F = S_{11} (B + I_b \cdot e^{i \cdot k \cdot z}). \quad (16)$$

Using (16) we can obtain equation for complete impedance  $Z_c$ :

$$Z_c = (B + I_b \cdot e^{i \cdot k \cdot z}) \cdot F_b + (F + I_f) \cdot F_f \cdot e^{-i \cdot k \cdot z} + 2Z_p,$$

where  $z$  is length of the part,  $k = 2\pi\omega/\beta c$ ,  $Z_p$  is impedance of the part.

## 5 EXAMPLES

### 5.1 Mode trapping above beam-pipe cut-off

The code was used for calculation of modes trapped in a beam pipe above its cut-off frequency. Narrow-band high - Q impedance is responsible for coupling between first bunch and successive bunches. The standard way to describe the impedance above beam-pipe cut-off is low-Q resonances, assuming the loading by radiation into the beam pipes. But simulation shows that low-Q resonances can trap fields with  $Q \approx 10^4$ . Geometry with rotational symmetry was used for the simulation. Longitudinal impedance for monopole modes, resonant frequency and Q-value were calculated assuming perfectly conducting walls, so that Q-values are caused only by radiation into beam-pipes. Geometry consisting of two identical pillboxes was simulated (Fig.3). Dimensions of the pillboxes are  $a = 1 \text{ cm}$ ,  $b = 2 \text{ cm}$ ,  $g = 1 \text{ cm}$ . The pillboxes

are connected by a long beam pipe with  $a = 1\text{ cm}$ , length  $L = 40\text{ cm}$ , and loaded by the same radius beam pipe with infinite length. At first, longitudinal impedance for the one separate pillbox was calculated for frequencies above beam-pipe cut-off. As we can see on the Fig.4 there are no high-Q resonances appeared on the impedance curve.

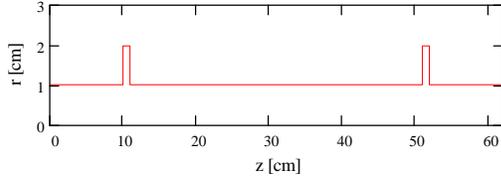


Figure: 3 Geometry of two pillboxes coupled through beam pipe.

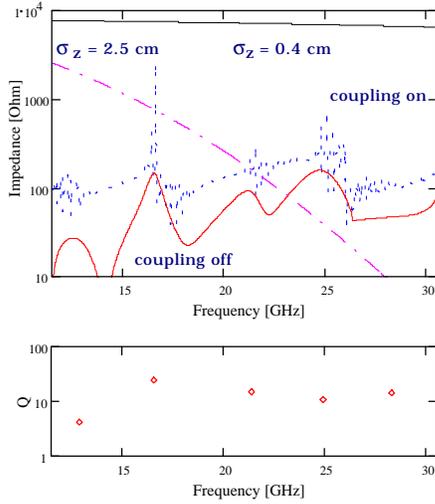


Figure 4: Upper picture: Comparison of longitudinal impedance for two pillboxes above beam-pipe cut-off frequency. Solid line - no coupling of the pillboxes through the beam-pipe. Dot line - coupled pillboxes. Dot-dash line - shape of bunch spectra [a.u.] for  $\sigma_z = 2.5\text{ cm}$ . Upper curve - shape of bunch spectra [a.u.] for  $\sigma_z = 0.4\text{ cm}$ . Lower picture: Q - value for resonances in the one separate pillbox loaded by the beam-pipes.

Then, frequencies and Q-values were calculated. The largest Q does not exceed 50 (lower picture on Fig.4). After that, the two coupled pillboxes were calculated. The results show sharp resonances located near low-Q resonances of the one pillbox. Additional curves on the Fig.4 present shape of beam spectra with r.m.s.  $z$ -length  $0.4\text{ cm}$  and  $2.5\text{ cm}$  to emphasise the effect of the impedance on short bunches. Blow-up of frequency range near the second resonance of the one pillbox is shown on Fig.5. As we can see on the lower picture of Fig.5 the loaded Q-value of the two-pillbox open cavity reaches the value of  $10^4$ . And no specific behaviour appears near the second cut-off frequency of  $26.36\text{ GHz}$ .

The narrow resonances can cause coupled bunch microwave instability. Hence, during the vacuum chamber design, one should pay attention to modes trapped by low-Q “chokes” above beam pipe cut-off frequency.

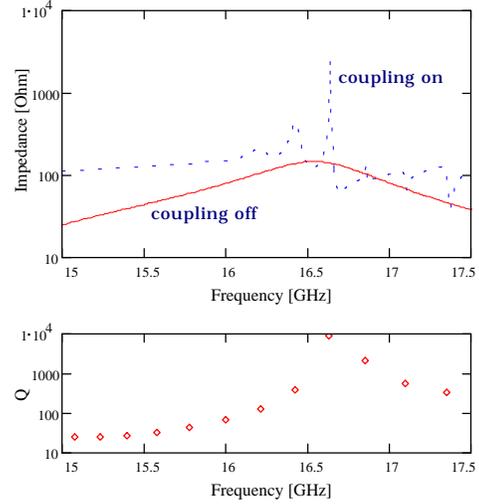


Figure 5: Upper picture: Comparison of longitudinal impedances for two pillboxes above beam-pipe cut-off frequency. Blow-up of curves from Fig.4. Solid line - no coupling of the pillboxes through the beam-pipe. Dot line - coupled pillboxes. Lower picture: Q - value for the resonator consisting of two coupled pillboxes loaded by the beam-pipes.

### 5.2 Transverse field caused by a shifted cell

Next example concerns accelerating structure design. An accelerating structure for modern projects of linear colliders is a disk-loaded waveguide[8][9]. Parameters of the structure is determined, between others, by requirements on the emittance preservation along the linac. Transverse forces inside a structure cause a dilution of the transverse emittance. The forces can be generated both by bunches (wake fields) and by accelerating field. Transverse force created by shifting of a cell of an accelerating structure was estimated. The shifted cell brakes cylindrical symmetry of the structure so that the high gradient accelerating field generates transverse field. Dimensions of the waveguide are: disk radius  $a = 4.5\text{ mm}$ , disk width  $t = 2\text{ mm}$ , cell radius  $b = 10.778\text{ mm}$ , period  $D = 8.7474\text{ mm}$ . That dimensions represent middle cell for X-band ( $11.424\text{ GHz}$ ) detuned accelerator structure for JLC linear collider [8]. First step was matching of a disk-loaded waveguide to simulate a travelling wave. 7-cell stack was excited by monopole mode. Radius  $b$  of the end-cells and radiuses of the outer discs were changed to obtain reflection coefficient less than  $0.02$ . Matched dimensions was  $b = 10.986\text{ mm}$ ,  $a = 5.99\text{ mm}$ . Then the middle cell was shifted as shown on Fig.6. Electric field was integrated along 5

trajectories to obtain map of the resulted fields. Fitting value  $V_{sh}$  from linear fitting of the results is equal to  $8.3 V/mm/\sqrt{W}$ . Then we can estimate transverse voltage  $V_{\perp}$  for full accelerator structure using following parameters: number of cells  $N=150$ , input power  $P=50 MW$ , average shift of cells  $dy=1\mu m$ . Using the equation:  $V_{\perp} = dyV_{sh}\sqrt{N}\sqrt{P}$ , we can obtain  $V_{\perp} = 720V$ .

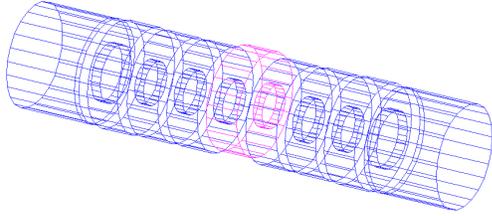


Figure 6: Geometry of the matched disk-loaded waveguide with shifted middle cell ( $dy=1mm$ ).

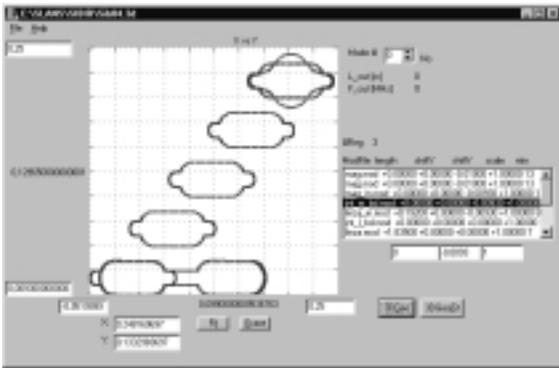


Figure 7: Dump of the preprocessor window. Outer part of the geometry is shown. The geometry was used for calculation of the longitudinal impedance of the vacuum chamber.

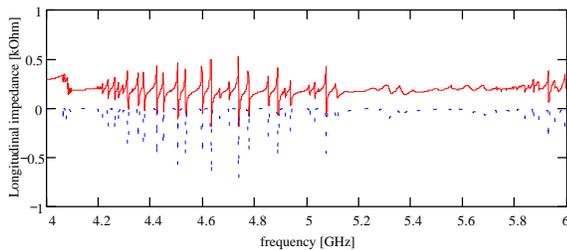


Figure 8: Longitudinal impedance of the vacuum chamber. Solid line -  $Im(Z)$ , dashed line -  $Re(Z)$ .

### 5.3 Impedance of a vacuum chamber

Synchrotron radiation source "Siberia-2" was designed and produced by Budker INP. The longitudinal coupling impedance was calculated for vacuum chamber located inside a quadrupole lens. The vacuum chamber has

length of  $183.5cm$  and cross-section about  $5.2cm \times 10.5cm$ . The cross-section is fitted to the poles of the lens. Both sides of the chamber are connected to dipole magnet's chamber by insertion that is used for adjusting of the cross-sections. The geometry of the junction is shown on Fig.7. Resulted impedance we can see on Fig.8. The chamber has a reach spectrum above  $4GHz$  of the modes trapped inside, in spite of that the lowest cut-off mode has frequency about  $2 GHz$ .

## 6 CONCLUSION

The interactive code was developed for simulation of 3D accelerator components. The code uses scattering matrix method. Performance of the code allows to calculate coupling impedance for complicate structures with arbitrary cross sections. Using the code a technology of cascading of impedances was tested. The cascading can be used in combination with a finite element or grid methods. The cascading is a universal method and it can serve as powerful tool for simulations of complex 3D structures. The code was applied for open cavity consisting of two pillboxes loaded by beam pipe. The calculations show that fields can be trapped by low Q resonances above the beam-pipe cut-off frequency. Transverse voltage caused by shift of a cell in a disk-loaded waveguide was obtained. Calculation of a part of the "Siberia-2" vacuum chamber shows sharp resonances far above the first cut-off frequency.

## 7 REFERENCES

- [1] V.Dolgashev, T.Higo, "Application of Scattering Matrix Formulation for Simulation of Accelerating Structures", Proceedings of LC97, Zvenogorod, Russia, October 1997.
- [2] S.A.Kheifets, S.A.Heifets, "Longitudinal electromagnetic fields in an aperiodic structure", SLAC-PUB-5907, September 1992.
- [3] Ursula Van Reinen, "Higher order mode analysis of tapered disc-loaded waveguides using the mode matching technique", Particle Accelerators, 1993, Vol.41, pp 173-201 [10] M.
- [4] Myakishev D.G., Yakovlev V.P. "An interactive code SLANS for evaluation of RF-cavities and accelerator structures", IEEE PAC, May 6-9, 1991, San Francisco, California, 91CH3038-7, Conference Record. V-5, pp.3002-3004.
- [5] K.Gupta and R.Cadha, "Computer aided design of microwave circuits", Artech House Inc., 1985
- [6] R.E.Collin, "Foundations for microwave engineering", McGraw-Hill, 1966.
- [7] S.Heiffets, A.Wagner, B.Zotter, "Generalised Impedances and Wakes in Asymmetric Structures", SLAC/AP110, January 1998.
- [8] JLC Design Study, High energy Accelerator Research Organisation, Tsukuba, Japan, April, 1997.
- [9] Karl L.F.Bane and Robert L. Gluckstern, "The Transverse Wakefield of a Detuned X-Band Accelerator Structure", Particle Accelerators 42, (1994)123-169.

# THE 30 GHZ TRANSFER STRUCTURE FOR THE CLIC STUDY

G.Carron, A.Millich, L.Thorndahl, CERN, Geneva, Switzerland

## Abstract

In the so-called “Two-Beam Acceleration Scheme” the energy of a drive beam is converted to rf power by means of a “Transfer Structure”, which plays the role of power source. In The Transfer Structure the bunched drive beam is decelerated by the electromagnetic field which it induces and builds up by the coherent interaction of successive bunches with the chosen longitudinal mode. The CLIC Transfer Structure is original in that it operates at 30 GHz and uses teeth-like corrugations to slow down the hybrid TM mode to make it synchronous with the drive beam. The beam energy is transformed into rf power, which travels along the structure and is collected by the output couplers. The 30 GHz rf power is then transported by means of two waveguides to two main linac disk-loaded accelerating structures. This report describes the CLIC Transfer Structure design, 3-D computer simulations, model construction and measurements as well as the prototype construction and testing with the low energy beam in the CLIC Test Facility. The result of this development is a compact, fully passive, relatively simple and low cost device, which offers a readily scalable solution to the problem of rf power extraction from high frequency bunched beams.

## 1 GENERAL DESCRIPTION AND PRINCIPLES

### 1.1 Definition and function of the Transfer Structure (TRS)

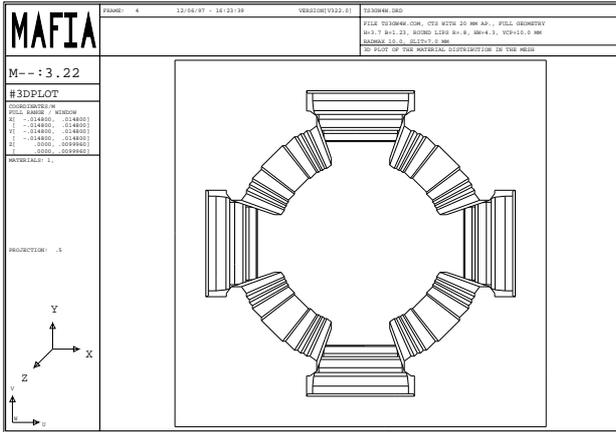


Figure 1 Three cells section of TRS

The transfer structure [1] is a passive rf device in which the bunched electron beam interacts with the impedance of periodically loaded waveguides and excites preferentially one synchronous hybrid TM mode. In the process the beam kinetic energy is converted in electromagnetic energy at the mode frequency which travels along the

structure with the mode group velocity. The rf power produced is collected at the downstream end of the structure by means of couplers and conveyed to the main linac accelerating structures by means of waveguides. In its classic configuration the TRS consists of a cylindrical beam chamber, which is coupled by longitudinal slits to four teeth-loaded waveguides as visible in Figure 1. It shows our reference TRS model with beam chamber diameter 24 mm.

### 1.2 Principle of operation.

When a train of short electron bunches each of charge  $q_b$  traverses a section of TRS  $l_s$  meters long, it builds up a voltage across the structure of peak value

$$U_d = \frac{\omega}{2} \left( \frac{R'}{Q} \right) l_s q_d \quad (1)$$

where  $\omega = 2\pi f$  is the excited mode frequency,  $R'/Q$  is the normalised longitudinal impedance per unit length (expressed in circuit Ohms/m) of the structure at frequency  $f$ ,  $q_d$  is the total beam charge in one drain time  $T_d$  of the structure and  $l_s$  is the structure length. The drain time is simply the time it takes for the energy deposited by one bunch in the fundamental mode to travel out of the structure starting from the moment the bunch has left the structure itself:

$$T_d = \frac{l_s}{c} \left( \frac{1}{\beta_g} - 1 \right)$$

where  $\beta_g = \frac{v_g}{c}$  is the normalised group velocity.

In order for the mode excitation to be coherent and therefore constructive, the bunch spacing must be a multiple of the mode wavelength which is 10 mm and the mode phase velocity must be equal to the speed of the relativistic bunches. The bunch time separation  $T_b$  however, must be much shorter than one drain time  $T_d$  in order for several bunches to contribute to the build up of the voltage  $U_d$ . The rate of energy deposition by the beam or the rf power generated in the TRS is obtained by multiplying the voltage  $U_d$  by the average beam current in one drain time  $q_d/T_d$ :

$$P = \frac{\omega}{2} \left( \frac{R'}{Q} \right) \frac{q_d^2}{T_d} l_s F^2(\sigma) \quad (2)$$

$F^2(\sigma)$  is the power form factor which takes into account the finite length of the gaussian bunches. For a train of bunches lasting much longer than the structure drain time the peak power level in equation (2) stays constant after one drain time has elapsed provided that the charge per

drain time remains constant. Expression (2) therefore gives the steady state power level at the structure output when neglecting the internal wall losses.

## 2 THE REFERENCE TRS.

### 2.1 Transfer structure parameters.

The four-waveguide TRS shape is the result of a development started several years ago, [2] [3]. The 3D simulations using MAFIA [4]. led to the determination of the main geometric and rf parameters.

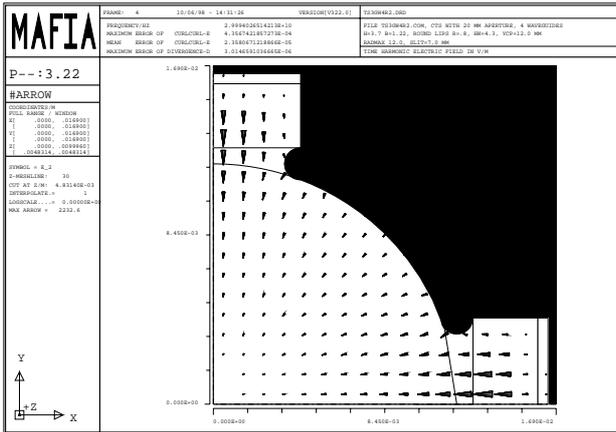


Figure 2 Transverse electric field in the TRS.

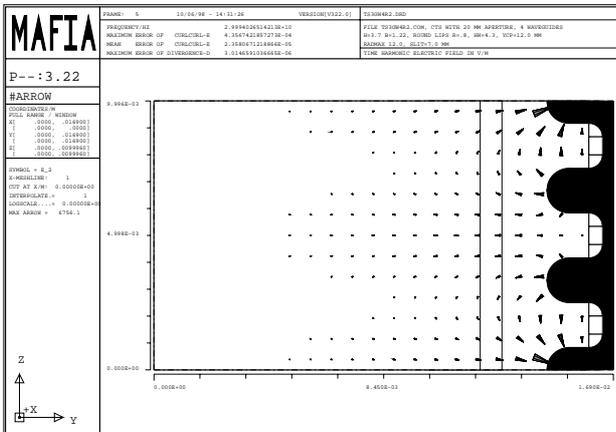


Figure 3 Electric field pattern in the longitudinal plane

The solution found for the resonant hybrid mode appears in Figure 2, which shows the transverse field pattern. The arrows length being proportional to field intensity, we see that most of the mode energy is located in the vicinity of the waveguides, a feature which favours power extraction. The mode phase advance of  $2\pi$  per 3 cells is well illustrated in Figure 3. A six-cell model was used to determine the dispersion curve of the structure and to derive the  $R'/Q$  and group velocity of the  $2\pi/3$  mode. Table 1 shows the main geometric and rf parameters of the transfer structure with 24 mm beam chamber aperture which has been adopted as power extracting structure for the drive beam decelerator [5].

Table 1 Parameters of the reference TRS

Beam chamber diameter	24.00	mm
waveguide width	8.60	mm
waveguide height	3.70	mm
slit aperture	7.00	mm
synch. mode frequency	29.983	GHz
synch. mode $\beta_g$	0.440	
synch. mode $R'/Q$	31.10	Ohm/m
peak transverse wakefield	0.42	V/pC/mm/m
effective structure length	0.80	m
nominal output power*	495.4	MW

\* The output power is computed for a train of bunches with charge 17.4 nC,  $\sigma = 0.5$  mm, spaced 20 mm and with  $F^2(\sigma) = 0.9$

### 2.2 Wakefields in TRS

The transverse wake induced in a 24 cells section of TRS by a gaussian bunch with  $\sigma = 0.6$  mm and charge one pC displaced one mm off center is shown in Figure 4 and its spectrum in Figure 5.

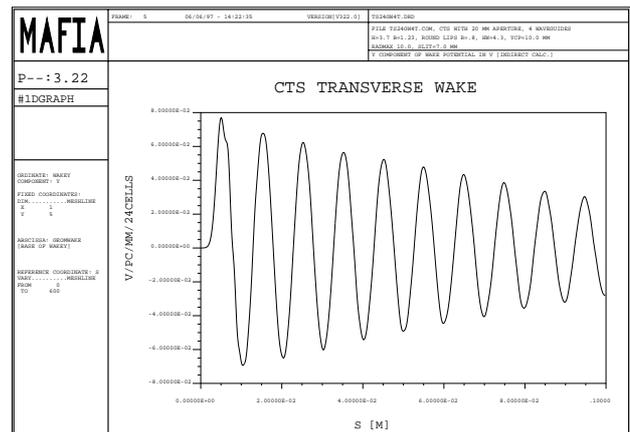


Figure 4 Transverse wakefield in 24 TRS cells

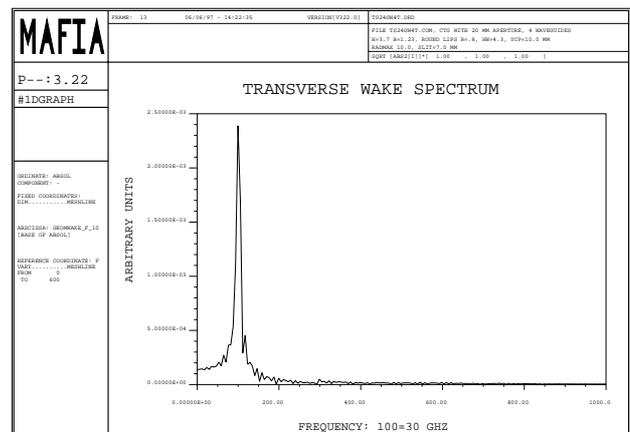


Figure 5 Transverse wake spectrum.

The wake spectrum shows almost no higher order modes. It is therefore justified to assume that practically all the transverse deflection of an off-center beam is caused by

the main deflecting mode, the frequency of which is only a few tens of MHz away from the main longitudinal mode. The value of the peak transverse wakefield, which appears in Table 1, is used in the computation of the transverse stability of the drive beam.

### 2.3 TRS integrated longitudinal electric field uniformity.

Because of the particular geometry of the TRS, the integrated decelerating field varies as a function of the angular and radial position within the beam chamber. The plots in Figure 6 show the variation of the normalised longitudinal integrated field over a three cells section of the TRS as function of the radial position for  $\phi=0$  (towards the middle of the waveguide) and  $\phi=45$  degrees (towards the chamber wall).

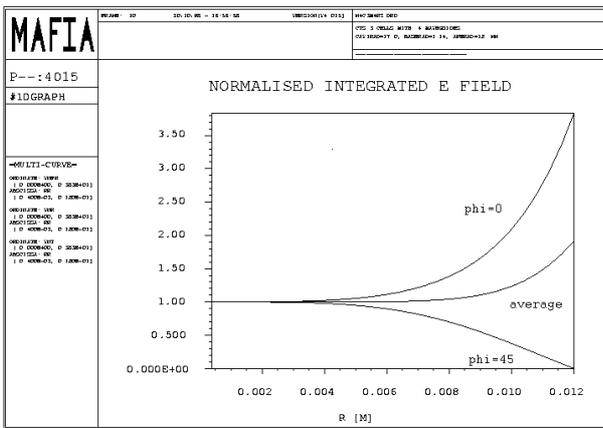


Figure 6 Normalised integrated field at  $\phi=0$ ,  $\phi=45$  degrees and average.

The non uniform beam deceleration causes the particles to receive transverse kicks which are a function of the particle position within the TRS chamber. The overall result found in tracking programs is that the drive beam would be unstable if no cure were found to the problem. One possible simple solution consists in rotating by 45 degrees every other TRS in the decelerating linac so that a particle off centre at  $\phi=0$  in a structure would be at  $\phi=45$  degrees in the following one, thus averaging the field non uniformity. Figure 6 also shows the normalised integrated field when averaged over two rotated structures as described above. The useful effect of the alternate TRS rotation is somewhat reduced by the betatron motion of the particles in the drive linac lattice, however tracking programs have shown that the overall result is beneficial to the transverse beam stability and worth the implementation effort [6]

### 2.4 Transfer structure for the test facility CTF2.

Prototypes TRS were built in 1995 and '96 for the Two-Beam tests to be performed in the CTF2 facility.. The beam charge available for the test being limited to 640 nC in 48 bunches with bunch distance 10 cm, the R/Q of the TRS had to be increased to 550  $\Omega/m$  in order to provide

the nominal output power of 80 MW. This requirement was met by reducing the beam aperture diameter to 15 mm and eliminating the lips between the beam chamber and the waveguides, see Figure 7. The reduced aperture produced an increment of the transverse wakefield which would make the low energy drive beam unstable [7]. It was found necessary to damp the main transverse mode to make the beam go through six transfer structures.

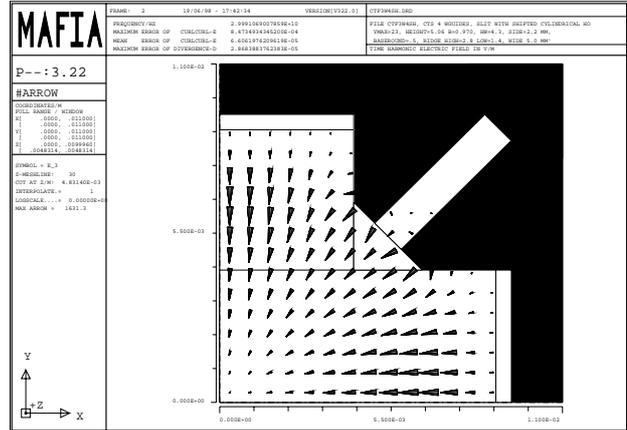


Figure 7 Longitudinal decelerating mode electric field pattern in TRS for CTF2

### 2.5 Transverse mode damping.

The TRS has been equipped with transverse mode dampers, which consist in four corrugated slits oriented at 45 degrees in the transverse plane in order to intercept the image current of the transverse mode. The slits are closed at their outer ends with respect to the beam chamber by rods of SiC forming rf loads. The position of the dampers is chosen in symmetry planes such that the main mode is not affected by their presence. The difference in mode coupling is well illustrated in Figures 7 and 8.

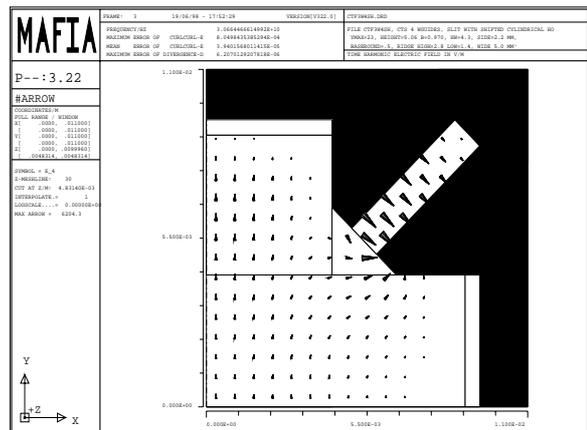


Figure 8 Transverse deflecting mode electric field pattern Model measurements indicate that the Q value of the transverse mode is strongly lowered by the dampers.

## 2.6 TRS construction method

Figure 9 shows the cross-section of the damped TRS installed and tested with beam in CTF2. The structure consists of four copper racks with the periodic corrugations, which are held by four square profiles of Cu-plated stainless steel (for mechanical rigidity). The corrugated damping slits with the SiC slabs are visible. The SiC slabs are pressed into the slots by helical springs running over the whole length inside the cylindrical bore of the profiles.

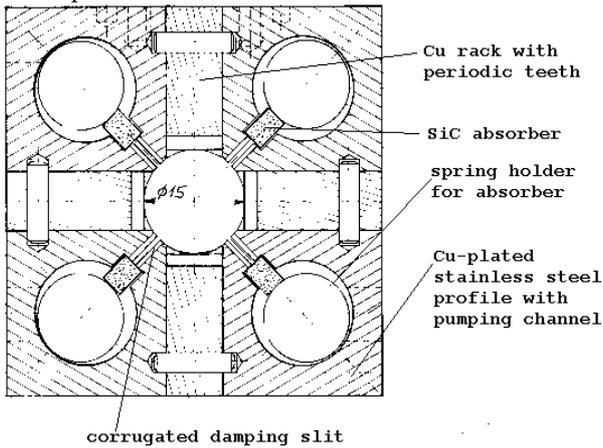


Figure 9 Cross-section of the damped transfer structure used in the CLIC Test Facility

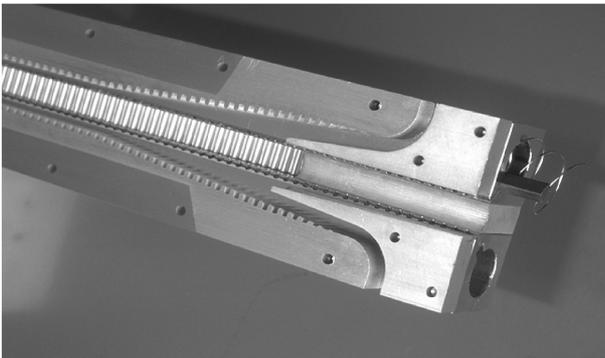


Figure 10 Extremity of open transfer structure with output couplers and channels

All metal parts are vacuum-brazed in a single operation. The power is extracted by means of output couplers visible in Figure 10. The extraction of the mode energy from the beam chamber with very high efficiency is made possible by the shape of the couplers, which present an outward ramp, where the tooth height is increased, thus slowing down the phase velocity locally. Towards the end of the ramp, inside the waveguides but before the last bends, the tooth height is slowly reduced to zero and the rf power is smoothly guided towards the output rectangular slits..

## 2.7 Model Work

Having met with difficulties in calculating the Q-value of damped transverse modes in the TRS because of the uncertain knowledge of the SiC properties, we attempted to measure them using short (to select the relevant mode) resonating models with the real SiC damping material. Such a model is shown in Figure 11. The model was totally encapsulated in metal and equipped with rf probes. In the case of the CTF2 structure the absorbers lowered the Q to values not measurable. In the case of our reference structure values about 60 were obtained.

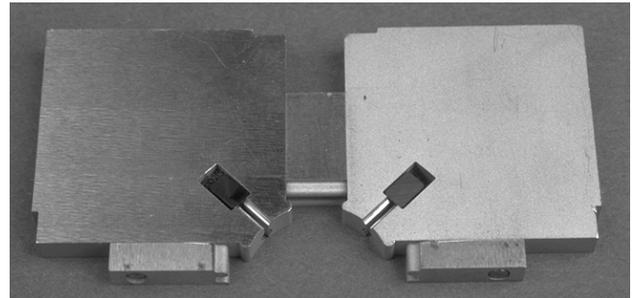


Figure 11 rf model for measurements of Q-values of transverse mode.

## 2.8 Beam Tests in the Clic Test Facility

The prototype four-waveguides TRS was installed in the drive beam line of the CTF2 and produced 27 MW rf power at 30 GHz.

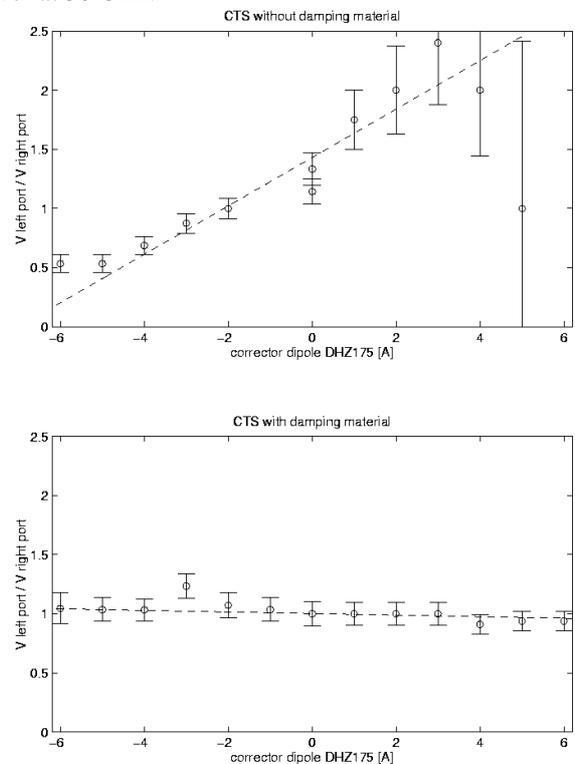


Figure 12 Results of measurements performed in CTF2.

The power was fed to a main linac disk-loaded structure, in which the probe beam was accelerated by 28 MeV [8]. The effort to produce the nominal drive beam intensity, which would provide a TRS output power of 80 MW, is being pursued. Comparisons between signals on the four output channels of the TRS when the beam is displaced laterally show a large beam-position dependence when the structure is not equipped with damping material and rather position-independent signals when the material is present [9]. In Figure 12 we see the ratio between left and right output signals when the beam is moved across the structure aperture. The upper graph is without damping material in the structure, the lower graph with damping material.

#### 4 CONCLUSION

The development of the TRS has resulted in the construction of two prototypes which, when tested with beam, have shown to produce the amount of power predicted by calculations as function of the beam charge. The four-waveguides TRS is a rugged and relatively simple device, which warrants low cost production of large number of units. New and more sophisticated designs being explored foresee structures with six or even eight waveguides, which are more complex to build and present challenging problems for the power extraction. Ultimately the drive beam stability requirements in the decelerator linac will impose the future design choices.

#### REFERENCES

- [1] W.Schnell, The Study of a CERN Linear Collider, CLIC. Linear Accelerator Conference, Williamsburg, October 1988.
- [2] G.Carron, L.Thorndahl, Impedance and Loss Factor Measurements on a CLIC Transfer Structure (CTS). EPAC 92, Berlin, March 1992.
- [3] A.Millich, Simulation of the CLIC Transfer Structure by means of MAFIA,. Computational Accelerator Physics Conference, Pleasanton, February 1993
- [4] T.Weiland et al, The MAFIA User's Handbook. Darmstadt, 1994
- [5] R.D.Ruth, et al, A new method of rf power generation for Two-Beam Linear Colliders, CERN-PS-98-014 (LP), EPAC98, Stockholm, June 1998.
- [6] A.Riche, D.Schulte, The Drive Beam Decelerator of CLIC, LINAC98, Chicago, August 1998.
- [7] C.Johnson, A.Riche, private communication.
- [8] M.Valentini, Experiments on Two-Beam acceleration during the first 1998 run of the CLIC Test Facility, CTF Note 98-19, Sept.1998.
- [9] H.Braun, private communication.

# FIELD FLATNESS AND NON-STATIONARY BEHAVIOUR OF THE 4\*7-CELL-TESLA-SUPERSTRUCTURE

H.-W. Glock, D. Hecht, U. van Rienen  
 Institut für Allgemeine Elektrotechnik, Universität Rostock  
 Albert Einstein-Straße 2, D-18051 Rostock, Germany

## Abstract

A structure of four coupled 7-cell resonators has been proposed [1] to increase the effective gradient of TESLA [2]. Each so-called "superstructure" is fed through a single input coupler. The sensitivity of field flatness against geometrical deviations and the time dependence of the fields during fill- and refill-time are studied by means of MAFIA [3] calculations using an optimized grid. The consistency of MAFIA results for perturbed versus unperturbed fields is confirmed utilizing an analytical relation. Non-stationary fields are expanded in a set of eigenmodes calculated with MAFIA. The method is described in some detail and results are presented in comparison with the respective results on the established 9-cell-structure.

## 1 INTRODUCTION

Overall length is a dominant cost contribution to the total investment for a linear collider. Therefore it is an important task to fill a given length as dense as possible with active, i.e. accelerating, elements. A modified so-called "superstructure" of four coupled 7-cell cavities has been proposed [1] to improve the fill factor of the TESLA [2] accelerator compared to the present design of decoupled 9-cell cavities. Furthermore the superstructure will need only a single input coupler driving the complete string. This paper describes two main aspects of the operation of such a superstructure we are investigating: the question of field flatness sensitivity against shape deviations and the non-stationary behaviour during filling time and beam passage. The following section describes a specialized grid that allows to approximate the cavity shape without staircases of the material distribution. Section 3 contains a method to confirm consistency of MAFIA results on modified boundary shapes. In section 4 the fundamental mode flatness sensitivity against a certain perturbation of a single cell is compared with a similar perturbation of a standard 9-cell resonator. Section 5 gives some detail about the calculation of transients based on eigenmode expansion and shows the results, again in comparison with a 9-cell resonator.

## 2 GRID GENERATION

A proper discretization of the cavity surface is essential for reliable numerical results, especially if small deviations are to be studied. Therefore the grid has been generated in a special manner: At each point of intersection of a certain r-mesh line with the cavity shape

a z-mesh line was placed. Then the boundary is approximated by those mesh cell's diagonals, which are secants of the curvature. Thus no staircases appear in the material distribution. On the other hand this leads to widely spread mesh cell dimensions, since both very high and very low slopes - corresponding to very narrow and very large z-mesh steps - have to be covered. Therefore the r-mesh positions have to be chosen carefully in a manner, that all given points of the structure were hit exactly, and that the local mesh step ratios are kept in reasonable limits.

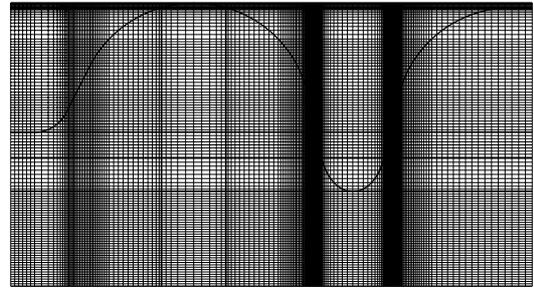


Figure 1: Part of the cavity shape and grid lines. High slopes of the boundary curve lead to dense z-mesh lines, regions of low slopes need dense r-mesh lines.

## 3 MAFIA CALCULATIONS OF CAVITY SHAPE PERTURBATIONS

In order to confirm the consistency of MAFIA-calculated fields and eigenfrequencies of perturbed and unperturbed cavities we calculated either case directly. Further we used expression (1) which is derived from Maxwell's equations without any approximation:

$$\omega - \omega_0 = \frac{i \oint_{\Delta A} (\vec{H} \times \vec{E}_0^*) d\vec{A}}{\iiint_V (\epsilon \vec{E} \vec{E}_0^* + \mu \vec{H} \vec{H}_0^*) dV}. \quad (1)$$

Herein  $(\vec{E}_0, \vec{H}_0, \omega_0)$  stands for the field distribution and eigenfrequency of the unperturbed cavity,  $(\vec{E}, \vec{H}, \omega)$  similar of the perturbed one.  $V$  is the volume of the perturbed cavity;  $\Delta A$  the closed surface of the insertion. If this insertion is located at the boundary of the unperturbed volume (like in the case shown in Fig. 2), only the inner part of  $\Delta A$  contributes to the integral.

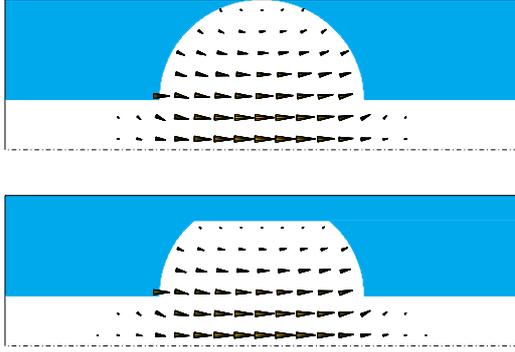


Figure 1: Upper half cross section of model cavity without (upper) and with perturbation. Arrows indicate the fundamental mode E-field, found at 4.3866 GHz (unpert.) and 4.9120 GHz (pert.)

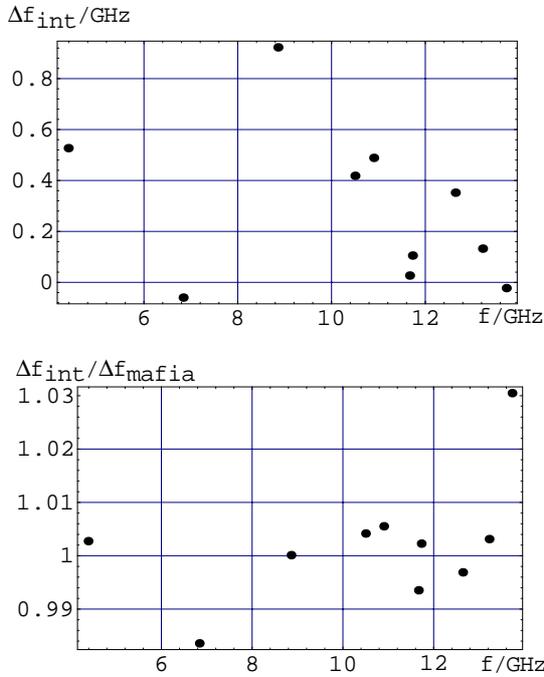


Figure 3 a, b: Frequency deviations  $\Delta f_{\text{int}}$  between perturbed and unperturbed cavities calculated from the right hand side fraction of Equ. (1) (upper); ratio of  $\Delta f_{\text{int}}$  to the difference of MAFIA-eigenfrequencies for the first 10 modes vs. unperturbed frequency (lower graph).

The consistency of the MAFIA-runs with and without perturbation is confirmed by evaluating Eqn. (1). This has been done in the case of 2D-calculated TM<sub>0</sub>-Modes. Then the surface integration (numerator) is reduced to a line integral that can be integrated using the MAFIA postprocessor. Some additional effort has to be spent to perform the cross product of the fields which has to be done in the 2D-case in elementary steps. The volume integral in the denominator is calculated abusing the MAFIA energy integration. To do this one has to prepare two extra fields

with single components equal to

$$\sqrt{\epsilon \vec{E} \vec{E}_0^*} \text{ and } \sqrt{\mu \vec{H} \vec{H}_0^*} .$$

Further attention has to be paid to the sign of the square root's arguments. The result of the right hand side fraction of Equ. (1) is displayed in Fig. 3 normalized to the frequency spread calculated directly for the first 10 modes of the geometry shown in Fig. 2. In general one can observe an agreement within a few percent, mostly below 1%. There is some tendency to smaller errors in case of higher frequency shifts. An unavoidable contribution to the differences comes from the line integral that has to be calculated for technical reasons one mesh line below the surface of the insertion. A second run with doubled distance was made in order to estimate the influence of this effect. Most of the errors were approximately doubled, too, so we assume the reason of the major part of the difference to be caused by the technical difficulty of the testing method.

#### 4 SENSITIVITY OF NORMAL AND SUPERSTRUCTURE

Four cavities are coupled in the superstructure. Therefore it is important to know about the influence of single boundary perturbations on the overall field flatness. In the first cell of the third cavity we applied an insertion at the cell's equator in a manner shown in Fig. 4. The radius of the cell shrunk from 104.935 mm to 104.53 mm. A similar insertion was applied to a standard 9-cell resonator. In either case this corresponds to a full ring of additional material since the calculations were done in 2D. We observe an unflatness of similar amount. The field disturbance covers a wider range in the long structure which one may judge as it's disadvantage.

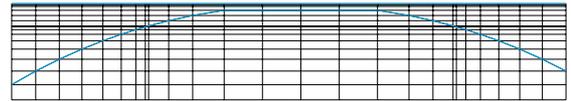


Figure 4: Part of grid and cell boundary with material insertion at the equator.

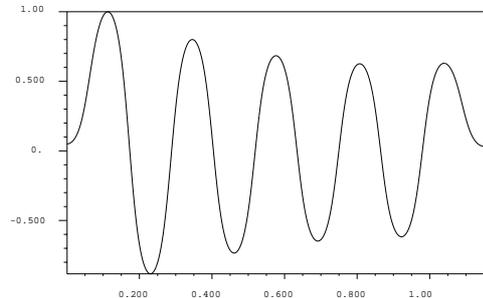


Figure 5: MAFIA-profile of accelerating field in 9-cell resonator with 0.405 mm reduced radius in first cell (cf. Fig. 4).

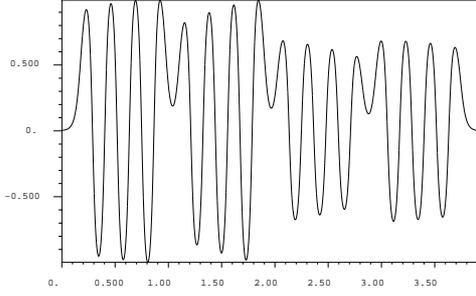


Figure 6: MAFIA-profile of accelerating field in 4x7-cell resonator with 0.405 mm reduced radius in first cell of third resonator ( $\approx 2.1$  m, cf. Fig. 4).

## 5 FILLING AND BEAM LOADING

Recent developments of input couplers and their power transmission capabilities made it possible to feed a complete superstructure with a single coupler. Then one has to look whether the filling process will be completed in the given time. Further the question arises whether the power taken away by the beam is re-established fast enough in order not to shrink the beam energy gain during the bunch train passage. To calculate this one could think of direct time domain simulation. In fact this is practically impossible for a mesh volume of approximately 450.000 points and a time intervall that covers about  $10^6$  oscillations. Therefore we used the approach of eigenmode decomposition (e.g. [4], [5]). A similar calculation with slightly different assumptions and based on another code has been done from Ferrario and Sekutowicz ([6], [7]). Our results are in good agreement with theirs.

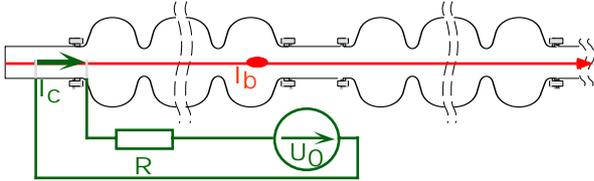


Figure 7: Model used for time domain calculations based on eigenmode expansions.

The model used for the calculation is shown in Fig. 7. The eigenmodes of the resonators are excited by the beam current  $I_b$  and a short transmitter driven current path  $I_c$ . An ideal rf-source with voltage  $U_0$  together with its inner resistance  $R$  are used to describe the transmitter. For reasons of simplicity  $I_c$  is placed on axis in the beam pipe near the beginning of the first cell. This is justified by a proportionality factor, common to all modes, of the axis fields to the fields at the real input coupler position. Since external Qs are about  $10^4$  lower than unloaded ones, resistive losses of the cavities are neglected. So the transmitter resistance is the only damping within the system, that furthermore causes a mode-to-mode coupling. Only the 28 modes of the fundamental passband were taken into account. This is justified by the large frequency gap to the next higher passbands, which appear above 2 GHz.

Starting from Maxwell's equations we find:

$$-\frac{1}{\mu} \Delta \vec{E} + \epsilon \partial_t^2 \vec{E} = -\partial_t \vec{j} . \quad (2)$$

The electrical field is expanded in a set of eigenmodes:

$$\vec{E}(\vec{r}, t) = \sum_{\nu} a_{\nu}(t) \frac{\vec{E}_{\nu}(\vec{r})}{\sqrt{2 W_{\nu}}} . \quad (3)$$

Each eigenmode solves:

$$\Delta \vec{E}_{\nu} + k_{\nu}^2 \vec{E}_{\nu} = 0 \quad \text{with} \quad k_{\nu}^2 = \omega_{\nu}^2 \epsilon \mu \quad (4)$$

and therefore holds:

$$\epsilon \sum_{\nu} \left[ \frac{\vec{E}_{\nu}(\vec{r})}{\sqrt{2 W_{\nu}}} (\omega_{\nu}^2 + \partial_t^2) a_{\nu}(t) \right] = -\partial_t \vec{j} . \quad (5)$$

Applying the orthogonality of the eigenmodes:

$$\iiint_{V_{\text{res}}} \epsilon \vec{E}_{\xi} \vec{E}_{\zeta} dV = 2\delta_{\xi\zeta} W_{\zeta} \quad (6)$$

yields:

$$(\omega_{\nu}^2 + \partial_t^2) a_{\nu}(t) = -\partial_t c_{\nu}(t) \quad (7)$$

with the abbreviation:

$$c_{\nu}(t) := \iiint_{V_{\text{res}}} \frac{\vec{j}(\vec{r}, t) \vec{E}_{\nu}(\vec{r})}{\sqrt{2 W_{\nu}}} dV . \quad (8)$$

Equ. (8) gives the expansion coefficients of the currents in the eigenmode system. The total current is splitted in beam (index b) and transmitter driven current (index c):

$$\vec{j}(\vec{r}, t) = \vec{j}_b(\vec{r}, t) + \vec{j}_c(\vec{r}, t) \quad (9)$$

or respectively:

$$c_{\nu}(t) = c_{\nu,b}(t) + c_{\nu,c}(t) . \quad (10)$$

If the beam travels on-axis with constant velocity  $v$  the following holds:

$$c_{\nu,b}(t) = \int_{z\text{-axis}} \frac{I_b(t - \frac{z}{v}) \vec{e}_z \vec{E}_{\nu}(\vec{r})}{\sqrt{2 W_{\nu}}} dz . \quad (11)$$

Similarly we get for the transmitter current which is assumed to be constant along its path:

$$c_{\nu,c}(t) = I_c(t) \frac{\int \vec{E}_{\nu}(\vec{r}) d\vec{r}}{\sqrt{2 W_{\nu}}} =: I_c(t) K_{\nu} . \quad (12)$$

Eqn. (12) defines the important abbreviation  $K_{\nu}$  that allows to write Kirchhoff's law along the external circuit:

$$U_c(t) + R I_c(t) = U_0(t) \quad (13)$$

as:

$$-\frac{1}{R} \sum_{\nu} a_{\nu}(t) K_{\nu} + I_c(t) = \frac{U_0(t)}{R} . \quad (14)$$

From (7), (10), (12) and the time derivative of (14) a system of coupled differential equations in the  $a_{\nu}(t)$  follows:

$$\begin{aligned} (\omega_{\nu}^2 + \partial_t^2) a_{\nu}(t) + \frac{1}{R} \sum_{\text{m}} K_{\nu} K_{\text{m}} \partial_t a_{\nu}(t) = \\ = -\partial_t \left[ c_{\nu,b}(t) + \frac{K_{\nu}}{R} U_0(t) \right] \quad \forall \nu \end{aligned} \quad (15)$$

Introducing a second set of variables:

$$\mathbf{b}_v := \frac{1}{i\omega_v} \partial_t \mathbf{a}_v \Leftrightarrow \partial_t \mathbf{a}_v = i\omega_v \mathbf{b}_v \Leftrightarrow \partial_t^2 \mathbf{a}_v = i\omega_v \partial_t \mathbf{b}_v \quad (16)$$

(the  $\mathbf{b}_v$  are the amplitudes of the B-fields) and using an additional abbreviation for the inhomogeneity in (15):

$$\mathbf{s}_v := \frac{i}{\omega_v} \partial_t \left[ \mathbf{c}_{v,b}(t) + \frac{\mathbf{K}_v}{R} U_0(t) \right] \quad (17)$$

we arrive at a system of differential equations written in matrix-vector-notation:

$$\partial_t \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{b}_1 \\ \vdots \\ \mathbf{a}_n \\ \mathbf{b}_n \end{pmatrix} - \begin{pmatrix} 0 & i\omega_1 & \dots & 0 & 0 \\ i\omega_1 & -\frac{K_1^2}{R} & \dots & 0 & -\frac{K_1 K_n \omega_n}{R \omega_1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & i\omega_n \\ 0 & -\frac{K_n K_1 \omega_1}{R \omega_n} & \dots & i\omega_n & -\frac{K_n^2}{R} \end{pmatrix} \begin{pmatrix} \mathbf{a}_1 \\ \mathbf{b}_1 \\ \vdots \\ \mathbf{a}_n \\ \mathbf{b}_n \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{s}_1 \\ \vdots \\ 0 \\ \mathbf{s}_n \end{pmatrix} \quad (18)$$

$$\text{or} \quad \partial_t \bar{\mathbf{v}} - \underline{\mathbf{M}} \bar{\mathbf{v}} = \bar{\mathbf{s}}. \quad (19)$$

The corresponding homogeneous system:

$$\partial_t \bar{\mathbf{v}} - \underline{\mathbf{M}} \bar{\mathbf{v}} = 0 \quad (20)$$

has the well known general solution:

$$\bar{\mathbf{v}}(t) = \sum_{j=1}^{2n} (\bar{\mathbf{e}}_j e^{\lambda_j t} u_j) \quad (21)$$

with the number of modes  $n$  used to expand the fields, the eigenvalues  $\lambda_j$  and the eigenvectors  $\bar{\mathbf{e}}_j$  of  $\underline{\mathbf{M}}$  and arbitrary constants  $u_j$ . The calculation of  $\lambda_j$  and  $\bar{\mathbf{e}}_j$  is done numerically which is (beside the field calculation) the only non-analytical step in the procedure. To solve the inhomogeneous system (19) the  $u_j$  are assumed to be time dependent. They can be found with some additional steps and by means of Cramer's rule as:

$$u_j(t) = \frac{\det \left( \bar{\mathbf{e}}_1, \dots, \int_{\tau=0}^t e^{-\lambda_j \tau} \bar{\mathbf{s}}(\tau) d\tau, \dots, \bar{\mathbf{e}}_{2n} \right)}{\det (\bar{\mathbf{e}}_1, \dots, \bar{\mathbf{e}}_{2n})}. \quad (22)$$

In (22) the additional assumption was made that all excitations are vanishing before  $t=0$ . We'll refer to the time dependent vector that replaces  $\bar{\mathbf{e}}_j$  in the numerator determinant as:

$$\int_{\tau=0}^t e^{-\lambda_j \tau} \bar{\mathbf{s}}(\tau) d\tau = \bar{\mathbf{W}}_j(t) = \bar{\mathbf{W}}_{b,j}(t) + \bar{\mathbf{W}}_{c,j}(t). \quad (23)$$

Herein the beam excitation is represented by  $\bar{\mathbf{W}}_{b,j}(t)$ , the transmitter current by  $\bar{\mathbf{W}}_{c,j}(t)$ . For all times  $t>0$  we assume a time dependence of the transmitter voltage like:

$$U_0(t) = V_0 e^{+i\omega_0 t} \quad (24)$$

which allows for an explicit integration:

$$\bar{\mathbf{W}}_{c,j}(t) = \frac{V_0}{R} \bar{\mathbf{K}} \frac{\omega_0}{i\omega_0 - \lambda_j} [1 - e^{(i\omega_0 - \lambda_j)t}] \quad (25)$$

$$\text{with } \bar{\mathbf{K}} = \begin{pmatrix} \vdots \\ 0 \\ \mathbf{K}_v / \omega_v \\ \vdots \end{pmatrix}. \quad (26)$$

The beam is composed as chain of single charges  $q$ , traversing  $z=0$  at different times  $t=t_b$  (an additional charge counting index omitted). Thus we have to replace in (11):

$$I_{b,\delta}(t,z) = q \delta(t - t_b - \frac{z}{v}) \quad (27)$$

After some further steps this leads to:

$$\bar{\mathbf{W}}_{b,j} = i q v \lambda_j \left( \sum_{t_b < (t - L_{res}/v)} e^{-\lambda_j t_b} \right) \times \begin{pmatrix} \vdots \\ 0 \\ \frac{1}{\omega_v \sqrt{2} \bar{\mathbf{W}}_v} \left( \int_{\tau=0}^{L_{res}/v} e^{-\lambda_j \tau} E_{z,v}(v \tau) d\tau \right) \\ \vdots \end{pmatrix} \quad (28)$$

In (28) only the sum and the charge depend on beam parameters; all other quantities are determined from the cavity modes. During a charge's transversal the result of (28) is in error for the contribution of this single charge, which is counted at a whole after the charge left the resonator. The solution of the mode amplitudes searched for is found from Eqns. (21), (22), (23), (25) and (28):

$$\begin{pmatrix} \mathbf{a}_1(t) \\ \mathbf{b}_1(t) \\ \vdots \\ \mathbf{a}_n(t) \\ \mathbf{b}_n(t) \end{pmatrix} = \frac{\sum_{j=1}^{2n} [\bar{\mathbf{e}}_j e^{\lambda_j t} \det (\bar{\mathbf{e}}_1, \dots, (\bar{\mathbf{W}}_{b,j}(t) + \bar{\mathbf{W}}_{c,j}(t)), \dots, \bar{\mathbf{e}}_{2n})]}{\det (\bar{\mathbf{e}}_1, \dots, \bar{\mathbf{e}}_{2n})} \quad (29)$$

The case before a beam transverses the cavity reads as:

$$\begin{pmatrix} \mathbf{a}_1(t) \\ \mathbf{b}_1(t) \\ \vdots \\ \mathbf{a}_n(t) \\ \mathbf{b}_n(t) \end{pmatrix} = \frac{V_0 \omega_0}{R} \sum_{j=1}^{2n} \left[ \bar{\mathbf{e}}_j \frac{e^{\lambda_j t} - e^{i\omega_0 t}}{\omega_0 + i\lambda_j} \det (\bar{\mathbf{e}}_1, \dots, \bar{\mathbf{K}}, \dots, \bar{\mathbf{e}}_{2n}) \right] \frac{1}{\det (\bar{\mathbf{e}}_1, \dots, \bar{\mathbf{e}}_{2n})} \quad (30)$$

In the TESLA scheme 1130 bunches of 5.7267 nC charge are foreseen following each other in a distance of 919 rf periods. The injection of the first bunch happens at rf period 760336 (584.6  $\mu$ s) at half the unloaded steady state voltage. From this an external  $Q = 3446120$  follows. This data allows to calculate the transmitter resistance, which of course is only valid for a certain coupling, defined by length and position of the current path. For the calculations in Fig. 8 a (slightly to low)  $Q = 3433810$  was found from the  $\underline{\mathbf{M}}$ -eigenvalue  $\lambda_j$  according:

$$Q_j = \frac{\omega_j}{2\alpha_j} = \frac{|\text{Im}(\lambda_j)|}{2\text{Re}(\lambda_j)} \quad (30)$$

This results in a voltage decrease (cf. Tab. 1) during the

bunch train and is cited here to illustrate the dependencies. The voltage was calibrated to be 25 MV/m inside the resonators at first injection, equal to 80.6756 MV in total. This total gradient was investigated for each bunch with an improved Q. The remaining deviations appear as jitter of below  $\pm 1000$  V (courtesy M. Dohlus).

Ez/(MV/m)	cell 1	cell 7	cell 28
bunch 1	47.352	48.009	47.426
bunch 1130	47.089	47.831	47.044
decrease	-0.55%	-0.37%	-0.80%

Table 1: Decrease of accelerating voltage caused by slight mismatch of transmitter resistance, i.e. coupling.

Fig. 8 shows as the result of our calculations a sequence of field profiles illustrating the process of build-up, profile stabilization and the influence of the beam. Fig. 9 shows a similar picture under same conditions for a 9-cell structure after  $10^4$  rf periods. As expected, it's field amplitudes are higher at this time and the flatness is better. Nevertheless the field in the superstructure is obviously established fast enough and the profile with beam remains stable.

## 6 CONCLUSION AND OUTLOOK

We studied two main aspects of a 4x7-cell "superstructure" in comparison with the standard TESLA-9-cell cavity. Neither the filling and refilling of the superstructure nor the field flatness sensitivity contradict a successful operation. Further studies will target the higher order mode behaviour.

## ACKNOWLEDGEMENT AND SUPPORT

The authors wish to thank M. Dohlus, DESY, who had a strong impact on the work presented here. This work was supported by DESY.

## REFERENCES

- [1] J. Sekutowicz, M. Ferrario, C. Tang: Superconducting Superstructure; LC97, Sept./Oct. 97, Zvenigorod, Russia
- [2] R. Brinkmann, G. Materlik, J. Rossbach, A. Wagner (Eds.): "Conceptual Design of a 500 GeV  $e^+e^-$  Linear Collider with Integrated X-ray Laser Facility", DESY 1997-048
- [3] MAFIA V. 4.015, CST, D-64289 Darmstadt
- [4] M. Dohlus, R. Schuhmann, T. Weiland: "Calculation of frequency Domain Parameters Using 3D Eigensolutions", International Journal of Numerical Modelling: Electronic Networks, Devices and Fields (invited, to be published)
- [5] H.-W. Glock, P. Hülsmann, M. Kurz, H. Klein: "Rise Time of Amplitudes of Time Harmonic Fields in Multicell Cavities", Proc. 1993 Particle Accelerator Conference, May 1993, Washington D.C., pp. 623-625
- [6] M. Ferrario, A. Mosnier, L. Serafini, F. Tazzioli, J.-M. Tessier, "Multi-Bunch Energy Spread Induced by Beam Loading in a Standing Wave Structure", Part.Acc. 52 (1996), pp. 1-30
- [7] J. Sekutowicz, M. Ferrario, C. Tang: "Superconducting Superstructure for the TESLA Collider", TESLA-Report 98-08, DESY, April 1998

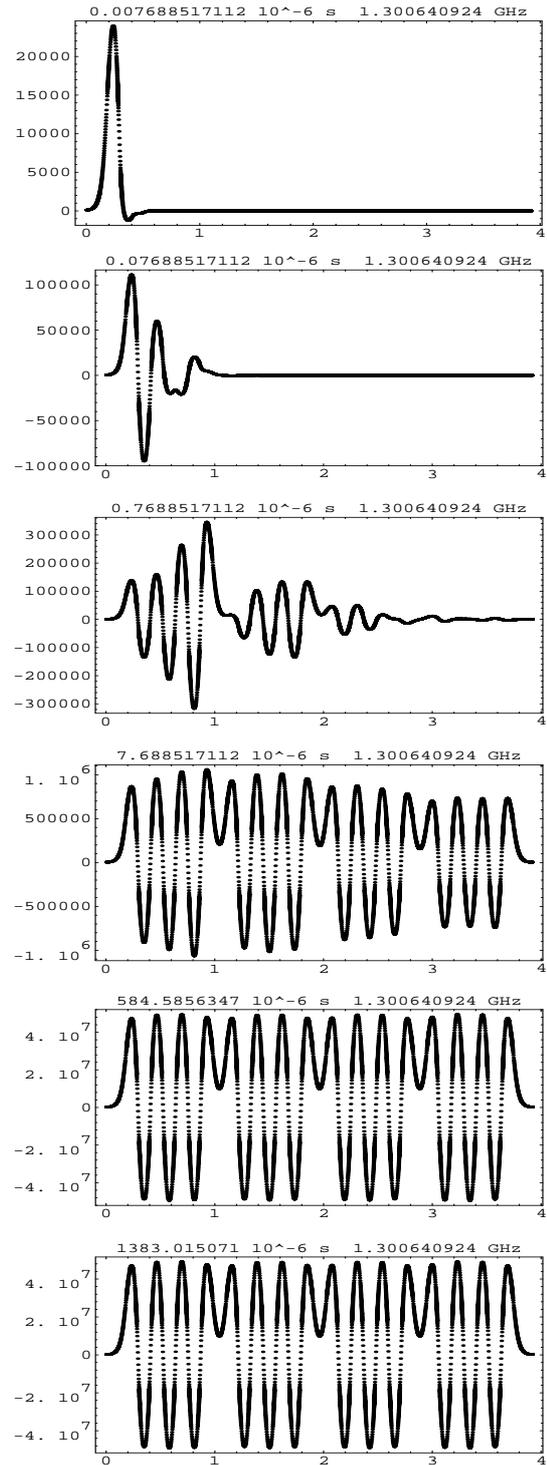


Figure 8: Sequence of field profiles ( $E_z$ /(V/m) vs.  $z/m$ ) at increasing time ( $10^1$  to  $10^4$  rf periods; first, last bunch)

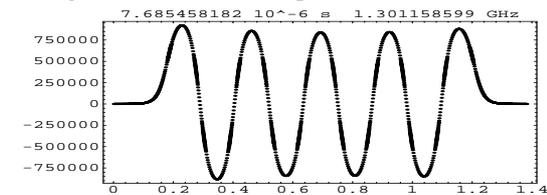


Figure 9: Field profile of 9-cell cavity after  $10^4$  rf periods; the profile is almost stabilized.

# MULTI-PLATFORM GRAPHIC USER INTERFACE FOR THE MARYLIE CHARGED PARTICLE BEAM TRANSPORT CODE

George H. Gillespie, Barrey W. Hill, Michael C. Lampel, Hendy Martono, John M. Moore  
G. H. Gillespie Associates, Inc., P.O. Box 2961, Del Mar, CA 92014, U.S.A

and

Alex J. Dragt

Department of Physics, University of Maryland, College Park, MD 20742, U.S.A.

## *Abstract*

An intelligent graphic user interface (GUI) is being developed for use with the particle optics program MARYLIE [1]. MARYLIE is based on a Lie algebra formulation of charged particle trajectory calculations and is particularly useful for particle tracking and the analysis of linear and nonlinear lattice properties. The GUI for MARYLIE uses the Multi-Platform Shell for Particle Accelerator Related Codes (S.P.A.R.C. MP). S.P.A.R.C. MP is a software framework developed specifically to support accelerator modeling and simulation. Transport element icons are selected from a palette and assembled into beamlines by graphical construction. Optical cells and lattices composed of element groups may be defined as sublines, and any element or subline can be replicated using an alias element. The icon-based beamlines generate entries for the #beam, #menu and #lines components of the MARYLIE Master Input File (MIF). Frequent computations, such as creating maps or generating particle distributions, are encapsulated into interactive GUI commands which create corresponding entries in the #menu, #lines and #labor components of the MIF. An icon-based description of procedural processes is being developed to support the more complex MARYLIE analysis tasks that utilize the #lumps and #loops components. Progress on the development of the GUI for MARYLIE is described and illustrations from the Windows95/NT implementation are presented.

## 1 INTRODUCTION

The S.P.A.R.C. MP software technology provides a framework for implementing cross platform graphical interfaces. Software applications using this framework can be built to run on different operating systems including Windows 95/98/NT, UNIX/X-Windows and the Mac OS. An overview of the approach to developing the cross platform code and a status report on recent developments of the S.P.A.R.C. MP technology are described elsewhere in these proceedings [2]. The Particle Beam Optics Laboratory (PBO Lab™) was the first application [3,4] to be developed with this technology and is now available commercially [5].

S.P.A.R.C. MP was designed to interface with a variety of accelerator modeling and simulation

programs. The software uses a modular approach that allows new programs to be added to the framework, without affecting the functionality of existing programs. The two programs TRANSPORT and TURTLE were integrated with S.P.A.R.C. MP as part of the PBO Lab [3,4]. The graphic interface for MARYLIE [1] is patterned after that developed for the PBO Lab, but several new GUI features have been added in order to support the extensive capabilities for beamline modeling and simulation that are available with MARYLIE. This paper describes some of these new features and illustrates an example of using the MARYLIE GUI.

## 2 GRAPHIC USER INTERFACE (GUI)

The main purpose in developing a GUI for scientific software is to assist researchers in using the software. This includes assistance in setting up problems, in executing the program, and in understanding the results. A useful GUI should provide support for: (a) the easy definition of problems for the target program, (b) a simple and clear means of requesting the program to perform the various tasks that are within its domain, and (c) the display of results from completing those tasks in ways that enhance interpretation and analyses. These three areas of GUI requirements generally parallel the traditional software paradigm based upon text files and command lines that separates the functions of input, execution and output. The goal for S.P.A.R.C. MP based applications is to support all of these areas graphically, without the user needing to prepare any text files or command line input. The development of the MARYLIE GUI focuses on this objective.

### 2.1 Problem Set Up

The GUI provides for the set up and running of MARYLIE without requiring any knowledge of the format, syntax, or similar requirements of the input. Beamlines are graphically constructed on the computer screen using drag and drop icons. Default parameters are incorporated for all required inputs so that both the topology of the beamline and a complete set of input data are defined automatically during the graphical construction. Setting up a specific beamline problem then only involves editing the values of parameters. The S.P.A.R.C. MP graphic user interface shell and the beamline construction kit provide this capability for the MARYLIE GUI.

Some of the basic features of the enhanced beamline construction kit developed for MARYLIE have been

described previously [6]. Figures 1 and 2 illustrate selected elements of the GUI used to set up beamlines for MARYLIE. A Document Window and three Subline Windows are shown in Figure 1; the new

features of the basic Document Window have been described elsewhere [2,6].

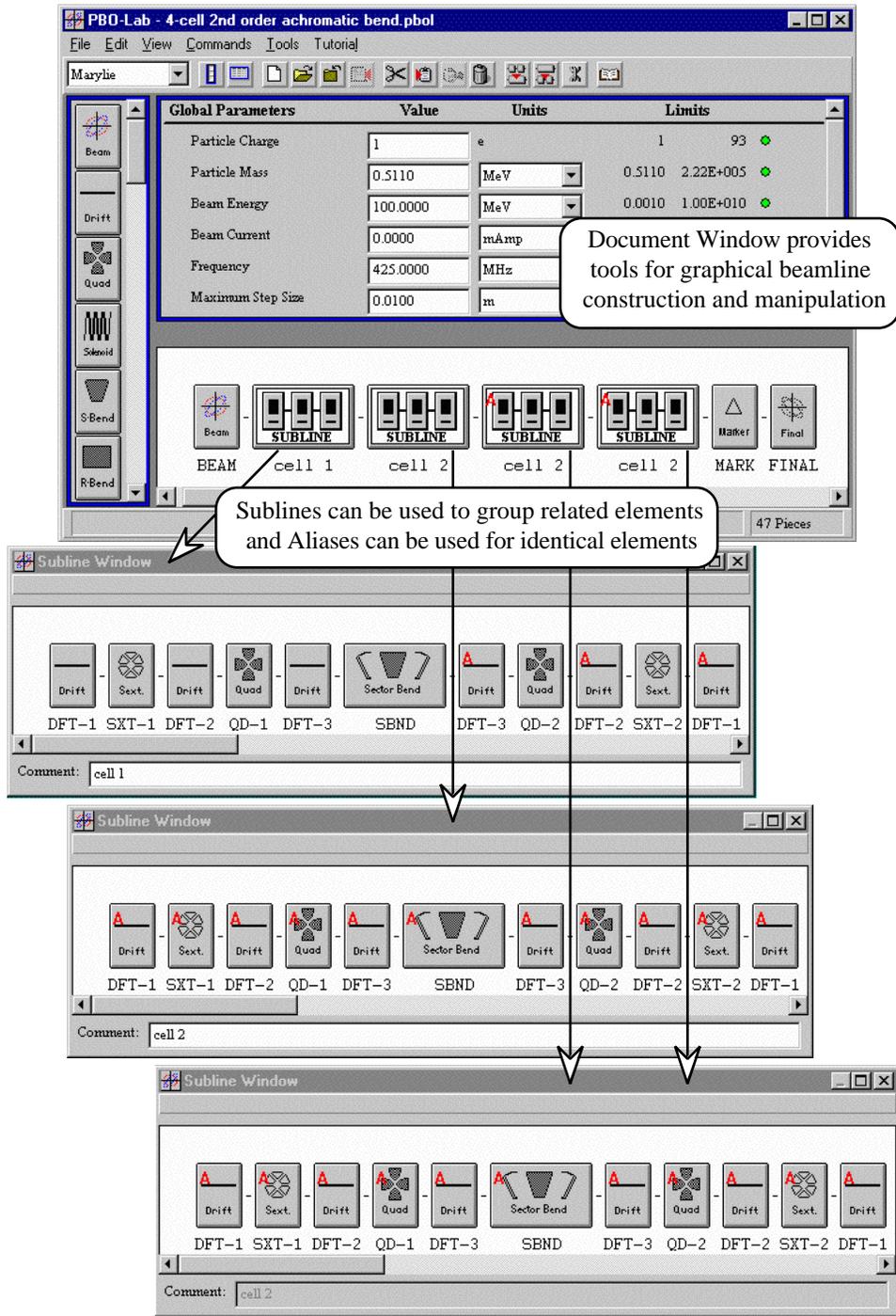


Figure 1. Example Document Window and three Subline Windows of the graphic user interface for MARYLIE.

Figure 1 illustrates a new method for setting up the second-order achromatic bend example described in [7]. The set up of this example capitalizes on the innovative beamline object model [8] that provides the infrastructure

for the beamline construction kit. Individual Pieces, Sublines and Aliases are used in the example. Beamlines can be modeled using hierarchical representations (Sublines), flat representations (individual Pieces), or mixed representations

that contain both Sublines and individual Pieces. An Alias is an individual Piece or Subline, which is represented by a persistent link to the underlying Piece or Subline. In the example discussed here, Aliases are used to provide identical replicas of frequently occurring elements. However, each Alias is also capable of storing deviations from the original parameters (e.g. to represent errors) without duplication of redundant data [2,6]. Note that the Alias Piece is not a copy of a Piece; a copy replicates all of the original Piece data as persistent data for the copy, and does not offer the efficiency in data storage made possible with an Alias Piece.

Figure 1 illustrates several uses of Alias Pieces, for individual Pieces and Sublines. The first Subline (labeled “cell 1”) defines the 1<sup>st</sup> cell of the achromat and contains eight (8) basic elements (three drifts and five magnets) as well as three (3) Aliases, each referring to one of the basic

drift Pieces. Icons for Aliases are indicated by a red “A” in the upper right corner, and the label (displayed below the Alias icon) is the same label as for the underlying original Piece. The second Subline, (labeled “cell 2”) describes the 2<sup>nd</sup> cell of the achromat. This Subline is composed entirely of individual Alias Pieces. The second Subline also has been used to define two Subline Aliases that define the 3<sup>rd</sup> and 4<sup>th</sup> cells, also labeled as “cell 2” since they are identical to the 2<sup>nd</sup> cell. Subline Aliases may contain Aliases to individual Pieces or other Subline Aliases.

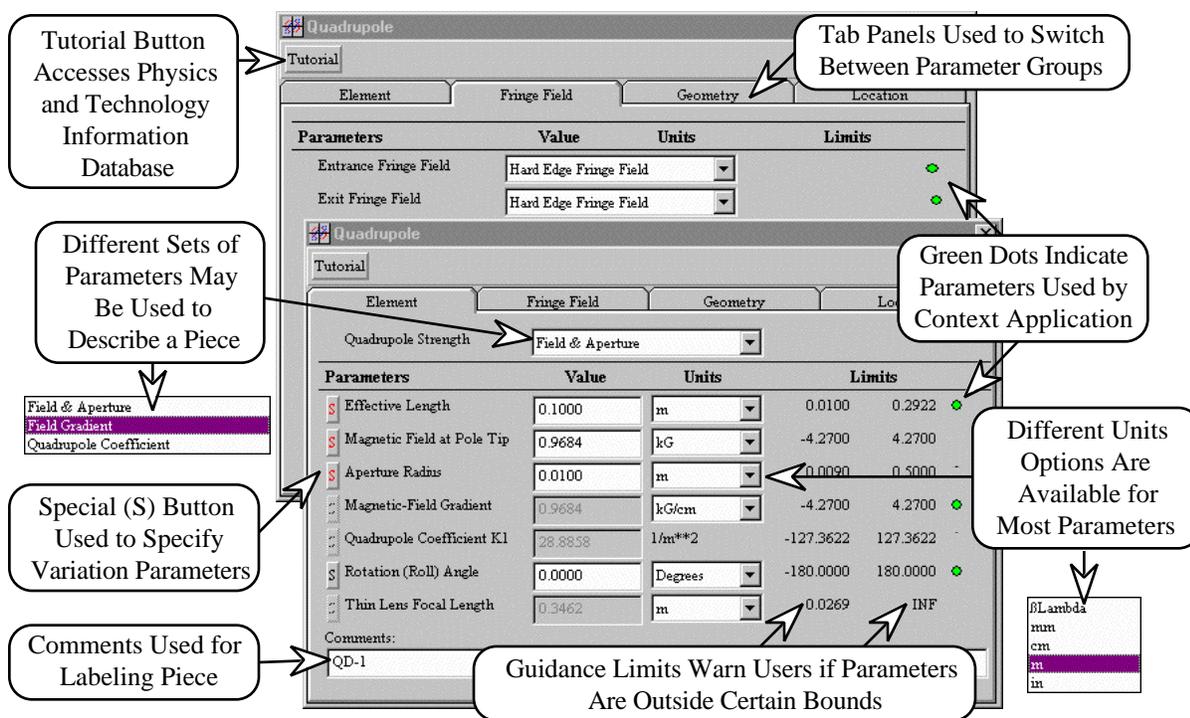


Figure 2. Example of Piece Window (left) and Alias Window (right) for editing parameters of a magnetic bend element.

Figure 2 illustrates the Piece Window for a magnetic quadrupole. Several new features have been added to the Piece Windows for all elements to support MARYLIE as well as other codes. The new features for magnetic elements include tab panels for additional data that describe fringe fields, magnet geometry parameters and beamline location (position and orientation) information. A useful feature of the Piece Windows for magnets is the ability for users to select from several different options for specifying the magnetic strength; the expert system rules automatically compute and display the alternative specifications. While a very convenient capability, MARYLIE only uses one set of magnet

specifications. Green dots are used on the right hand side of each Piece Window to provide feedback to the user on which parameters will be used by MARYLIE.

MARYLIE supports a rich library of commands, operations and procedures to assist users in solving a variety of beam optics problems. The GUI requirements to support these capabilities are still being developed. The next section describes an example that calculates the transfer map for the beamline shown in Figure 1.

## 2.2 MARYLIE Master Input File

The primary input to MARYLIE is provided by data in a Master Input File (MIF). The MIF contains seven

components referred to as #comment, #beam, #menu, #lines #lumps, #loops and #labor [1]. The last six of these MIF components define each MARYLIE run, and the graphic interface writes the necessary data to each MIF component to carry out the task prescribed by the user.

Figure 3 illustrates the MIF generated by the GUI for a user requesting the computation of the non-linear transfer map for the beamline illustrated in Figure 1, and

```

#comment
PBOL-Marylie Model
#beam
0.33526427480817
195.69471624266146
1.000000000000000
1.000000000000000
#menu
drft0001 drft
0.050000000000000
sext0002 sext
0.100000000000000 107.03699999999999
drft0003 drft
0.050000000000000
quad0004 quad
0.100000000000000 9.684390000000000 1.000000000000000
1.000000000000000
drft0005 drft
0.150000000000000
sbln0006 nbnd
10.00000775922950 0.020000000000000 0.450000000000000
0.585147000000000 1.000000000000000 1.000000000000000
quad0007 quad
0.100000000000000 -10.180100000000000 1.000000000000000
1.000000000000000
sext0008 sext
0.100000000000000 -145.66799999999998
mark0009 mark
finl0010 mark
enddata ptm
2 2 2 0 1
echodata pmif
0 12 2

interpret pmif
1 12 2
raytrace rt
13 14 3 0 1 0
track circ
13 14 3 100 100 0
order1 mask
1 0 0 0 0
order2 mask
1 1 1 1 0
return end
#lines
subl0048
drft0001 sext0002 drft0003 quad0004 drft0005 sbnd0006 &
drft0005 quad0007 drft0003 sext0008 drft0001
sbnd0006
sbln0006
subl0049
drft0001 sext0002 drft0003 quad0004 drft0005 sbnd0006 &
drft0005 quad0007 drft0003 sext0008 drft0001
beamline
subl0048 subl0049 subl0049 subl0049 mark0009 finl0010
#loops
testloop
beamline
#labor
echodata
interpret
beamline
enddata
return

```

Figure 3. Marylie MIF data generated by the GUI for a user request to calculate and display the R-matrix and monomial coefficients of the non-linear map for the achromatic bend example illustrated in Figure 1.

The Global Parameters generate entries in the #beam component and provide data for certain #menu component entries. The #menu component defines most of the data used in subsequent components. Individual Pieces generate entries in the #menu component. The various tasks (commands, operations and procedures) defined by the user also create entries in the #menu component. In addition, the GUI generates #menu entries for frequently used commands and for outputting data needed to display results requested by the user. Sublines generate entries in the #lines component. Aliases (with no deviations) are utilized in the data for the #lines component, and the complete beamline also creates an entry in the #lines component. The GUI also generates entries in the #loops component that are needed for certain tasks, such as element-by-element tracking or ray tracing.

The #labor component controls the actual execution of MARYLIE tasks. Most of the entries in this component are derived from entries in the other components, arranged so that the tasks requested by the user will be completed. Several commonly requested tasks, such as the map calculation discussed here, as well as generating distributions, ray tracing, tracking, etc., are

to display the results in a text editable window on the computer screen. It is worth noting that the user does not actually see the MIF text file (unless he or she specifically requests this). The MIF is written, MARYLIE is called and executed, and then the output is displayed as requested.

encapsulated into macros so that they are executed by the selection of single item in the Commands menu of the Document Window. Less frequent or more complex tasks can also be set up graphically. The #labor component entries follow, in left-to-right order, the commands and operations defined by the user graphically on the Model Space of the Document Window. Procedural loops, such as user requests for carrying out iterative fitting or optimization calculations, also generate entries in the #labor component. Both sequential procedures and nested procedures can be defined graphically and the appropriate entries for the #labor component are generated.

### 2.3 Display of Results

The output requested from the MARYLIE run specified in the MIF of Figure 3 is displayed in a text window, but several graphic capabilities for MARYLIE output are also being developed. These include plots of lattice functions and other parameters as a function of position, as well as different views of the 6-D phase space coordinates for beam distributions. Some specialized capabilities are also being developed specifically to support the ease of use of MARYLIE. One of these uses

the new Map Piece [6].

The Map Piece has been developed to provide new functionality for the MARYLIE GUI. The Map Piece can be used to enter a user-specified map, to load a map from data file, and to create and save a map from a given selection of Pieces or Sublines. For example, the map calculation described above can also be accomplished by interactively creating a map piece. To do this, the user

selects the four Sublines on the Model Space of the Document Window shown in Figure 1, and then chooses the “Create a Map Piece” item from the Commands menu. The new Map Piece appears on the Work Space of the Document Window and may then be used like any other Piece. Double-clicking the icon opens a specialized Piece Window that displays the map data. Figure 4 shows selected images from the GUI used for this process.

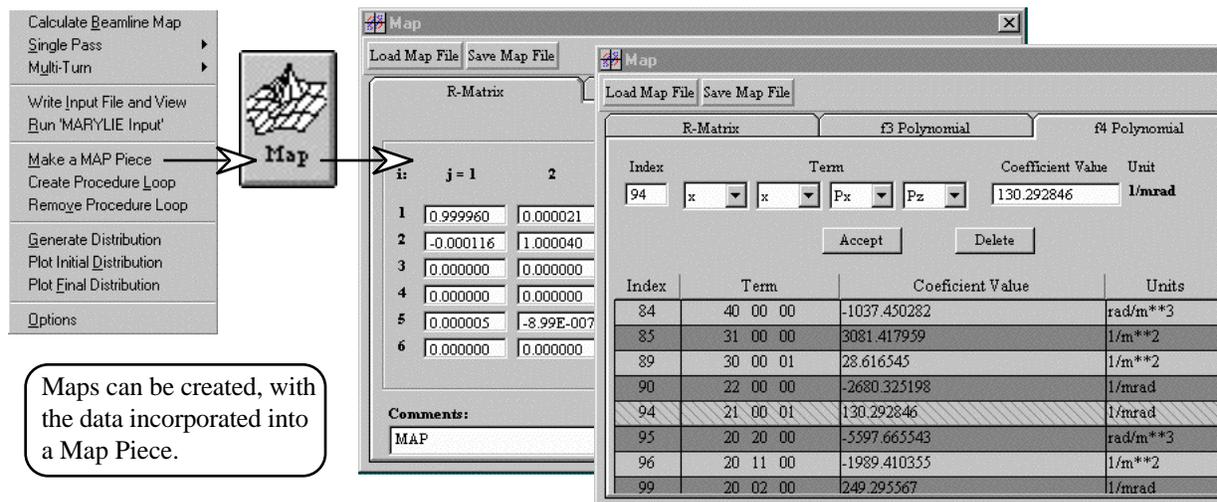


Figure 4. A map can be generated for a group of elements and the data entered automatically into a Map Piece.

### 3 SUMMARY

A graphic user interface is being developed for MARYLIE that promises improved productivity for researchers working on the design and analysis of accelerators and beamlines. Substantial progress has been made in developing a MARYLIE Module for integration with S.P.A.R.C. MP.

### ACKNOWLEDGMENT

This work has been supported by the U. S. Department of Energy under SBIR grant number DE-FG03-95ER81975.

### REFERENCES

[1] A. J. Dragt, D. R. Douglas, F. Neri, C. T. Mottershead, R. D Ryne, E. Forest, L. M Healy, P. Schutt and J. van Zeijts, “MARYLIE 3.0 User’s Manual, A Program for Charged Particle Beam Transport Based on Lie Algebraic Methods,” October draft, 543 pages (1998).

[2] B. W. Hill, H. Martono, J. M. Moore and J. S. Gillespie, “An Application Framework and Intelligent Graphic User Interface for Multiple Accelerator Codes,” these proceedings, 5 pages (1998).

[3] G. H. Gillespie, B. W. Hill, N. A. Brown, H. Martono and D. C. Carey, “The Particle Beam Optics Interactive Computer Laboratory,” AIP Conference Proceedings 391, 264-269 (1996).

[4] G. H. Gillespie, B. W. Hill, H. Martono, J. M. Moore, N. A. Brown, and M. C. Lampel, “The Particle Beam Optics Interactive Computer Laboratory for Personal Computers and Workstations,” Proceedings of the 1997 IEEE Particle Accelerator Conference, 2562-2564 (1998).

[5] *PBO Lab*<sup>TM</sup> is available from AccelSoft Inc., see [www.ghga.com/accelsoft](http://www.ghga.com/accelsoft).

[6] G. H. Gillespie, B. W. Hill, M. C. Lampel, H. Martono, J. M. Moore, K. J. Ryne, and A. J. Dragt, “A Multi-Platform Graphic User Interface for the Particle Optics Code MARYLIE,” to be published in the proceedings of XIXth International Linear Accelerator Conference (Chicago), 3 pages (1998).

[7] G. H. Gillespie, B. W. Hill, H. Martono, J. M. Moore, M. C. Lampel and N. A. Brown, “Using the Particle Beam Optics Laboratory (PBO Lab<sup>TM</sup>) for Beamline Design and Analysis,” to be published in proceedings of the 15th International Conference on Cyclotrons and Their Applications (Caen, France, 15-19 June 1998), 4 pages (1998).

[8] B. W. Hill, H. Martono and J. S. Gillespie, “An Object Model for Beamline Descriptions,” AIP Conference Proceedings 391, 361-365 (1996).

# THE CERN/SL XDATAVIEWER: AN INTERACTIVE GRAPHICAL TOOL FOR DATA VISUALIZATION AND EDITING

G. Morpurgo, CERN

## Abstract

As a result of many years of successive refinements, the CERN/SL Xdataviewer tool has reached its final stage. This graphical tool was especially developed to plot mono- or bi-dimensional arrays of data, and to interact with them, in many different ways. Many pages of graphical information can be handled by the program, in a loose hierarchy of Views, Graphs and Plots objects. Data can be displayed, and interacted with, both in graphical and in text format. Sophisticated built-in Zoom and Data Editing capability is implemented, as well as a flexible Data Output generation facility. A complete C Callable Interface is provided, including a mechanism for feeding back the Application Program with the selections made by the User in the Data Display part. The tool has been written in C language, making use of the standard X Window libraries (Xlib,Xt,Motif). It can be run as a stand alone process, communicating with the Application Program via a shared memory, or it can be embedded in the Application Program itself.

## 1 INTRODUCTION

The Dataviewer was originally developed at CERN/SPS in the late 80's , during the preparation of the software for the new CERN accelerator, LEP. The intention was to provide the Application Programmers and the LEP Operators with an uniform way to implement and use graphical data representations. Many features specific to the LEP context (ex. function editing) were introduced from the first design stage. The implementation was APOLLO specific, as those were the Workstations used at LEP. X-Window did not exist yet. The tool became rapidly popular, and many Applications were built on top of it. Many slightly different variants were developed by different people, until 1990, when Ann Sweeney cleaned and enhanced the code, merging the best features of this "parallel development", and froze the product[1]. Around 1994 the APOLLO workstations were replaced by HP-UX and Xterminals, running Unix and X-Window. Suddently the Dataviewer became unusable. As your author had many Applications built on top of it, he decided to port the Dataviewer to the new environment. More and more Applications used it, and many new features were added to the tool, as a consequence of requirements of new Application. This was done always mantaining the backwards compatibility, so that already existing Applications could always use the latest versions without any modification to the source code[2]. One of these new features is the "Embedded Dataviewer". The Dataviewer was originally conceived as a Stand Alone Process, displaying data that an Application Program had put into a shared memory area. We found that in many cases is useful to merge

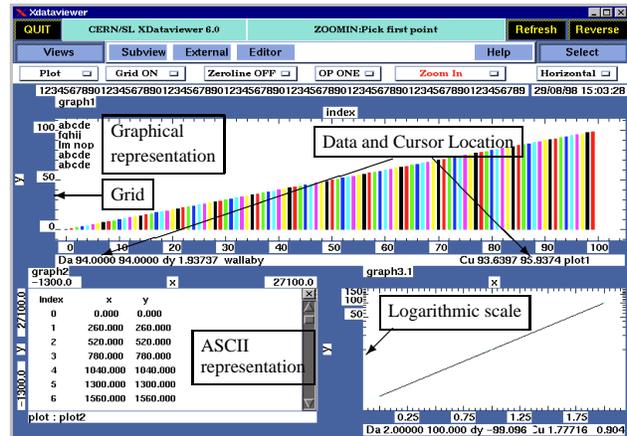


Figure 1: A typical Dataviewer page. A View containing three Graphs is shown

the Application Program with the Dataviewer, and therefore we have implemented a way to embed the Dataviewer inside the Application Program itself. This is done with the auxil of a very few function calls.

## 2 PHILOSOPHY AND REQUIREMENTS

The original design and implementation of the Dataviewer dates from the early days of workstations. At that time, the few commercial products on the market were primitive and expensive, so that there weren't many alternatives but developing this tool. Nowadays the scenario is quite different; many commercial packages are available, they are powerful and claimed to be grosso modo complete, and probably affordable. Nevertheless, they lack at least three of the most important requirements for such a tool in our environment: specific adaptation to our needs, immediate support, and possibility of developing new features as soon as the users need them. This is why if you have a good piece of home-made software, fulfilling your requirements and supported well, you should think twice before dropping it in favour of software on which you do not have any control.

Let us now have a look to the list of specific requirements satisfied by the Dataviewer.

- Ability of showing graphical pages ("Views"), structured in one or more parts ("Graphs"), each of which can display one or mode data arrays ("Plots").
- Possibility of online selection between different Views.
- Each Graph can be displayed in graphical or text format, independently from the other Graphs in the same View. The switch between graphical and text mode is available online.

- Online capability of selective displaying of View subsets.
- Possibility of editing a Plot online (modifying the correspondent data array).
- Possibility of zooming on the data.
- Active cursor, indicating online its position, and the position of the closest data point.
- Dynamic online reconfigurability of Views, Graphs and Plots.
- Access to the data via pointers.
- Optional drawing of axes and grids on the Graphs.
- Optional automatic rescaling of the Graph coordinates, driven by the data to be displayed.
- Possibility of assigning different colours and markers to each data point.
- Interaction with the Application Program, to know when to refresh the data and to inform it when a data point has been selected.
- Possibility of saving the screen data onto a file or to a printer.
- Callable Interface written in C, to enable the Application Program to configure the graphical information and to select the wished features.

### 3 IMPLEMENTATION ISSUES

#### 3.1 Object definition

Four basic entities constitute the hierarchy on which the information visible through the Dataviewer are built :

- *The View.* A View is a graphical page, displayable as a whole on the Dataviewer Window. It can contain many Graphs.
- *The Graph.* A Graph is a window on the x,y space, in which one or more Plots are shown. To be visible, a Graph must be attached to at least a View.
- *The Plot.* A sequence of x,y points. the y values always come from a Data Objects. The x values come either from a Data Object (2-dim Plot), or from the index of the sequence (1-dim Plot). A Plot needs to be attached to at least a Graph.  
Auxiliary Objects can be used to define colours, labels, etc. for each point of the Plot.
- *The Object.* An Object is an array of numeric values, provided by the Application Program. An Object can contain data to be plotted by the Dataviewer, color codes, marker types, error bar values, or labels (in this case it will be an array of character strings). Objects are contained in MOPS structures (see next paragraph), referred to by names and accessed by pointers.

#### 3.2 Object properties

We briefly list here the most important properties for Graphs and Plots

- A *Graph* can be displayed in graphical or text format.
- Its x,y limits might be fixed or automatically adapted to the data.

- It may use linear or logarithmic scales.
- It may have timestamps as x coordinates.
- It may have a Grid and x,y axes.
- It may display also a few 80-char labels.
- A *Graph* can be attached to any number of Views.
- A *Plot* may be 1-dim or 2-dim.
- It may be drawn as an histogram, or as a sequence of markers, optionally connected by a line.
- Every point may have a different colour and label, and also a different marker type (32 colours and about 20 marker types are defined).
- A 2-dim Plot may have its x coordinates increasing monotonously, or being randomly distributed.
- Every point may have error bars.
- Also for a Plot listed in text format, every point can have a different colour.
- Output formats for x and y values can also be specified.
- It is also possible to ask the Dataviewer to show only a selected part of a Plot.
- A Plot can be attached to any number of Graphs.

All of these properties can be controlled by the Application Program via the C Callable interface to the Dataviewer. Default settings for many of these properties are also predefined. Again via functions in the Callable Interface, these defaults can be overridden at creation time both for single or for multiple Objects.

#### 3.3 Data format

All the data required by the Dataviewer (plottable Objects, Views, Graphs and Plots definitions) must be contained into a MOPS[3]. The MOPS (Multiple Object Partitioned Structure) is a reserved memory area (Unix shared memory or internal process memory) dynamically configurable in User defined "Objects". The access to the MOPS and to the Objects is done by name, through a library of C (or FORTRAN) functions. In particular, some functions return pointers to the beginning of each Object Data Area. Data access can be protected by semaphores, if needed. An Object is an array of elements of the same type (int, float, structures, strings,...), and different Objects may have different types. The User can dynamically create or delete objects, change their number of elements, and retrieve or update their values. The data arrays displayed by the Dataviewer are MOPS Objects, and the MOPS access by name capability is used by the Dataviewer Callable Interface. The MOPS Library provides a easy and flexible way for structuring the data and for sharing it between different processes. For instance, there is no need to tell the Dataviewer the number of elements in a plot; it will find it by itself when looking at the MOPS. This solution mirrors very well the structure of many Applications, where the Application Program provides new data and leaves to the Dataviewer the task of showing them to the User.

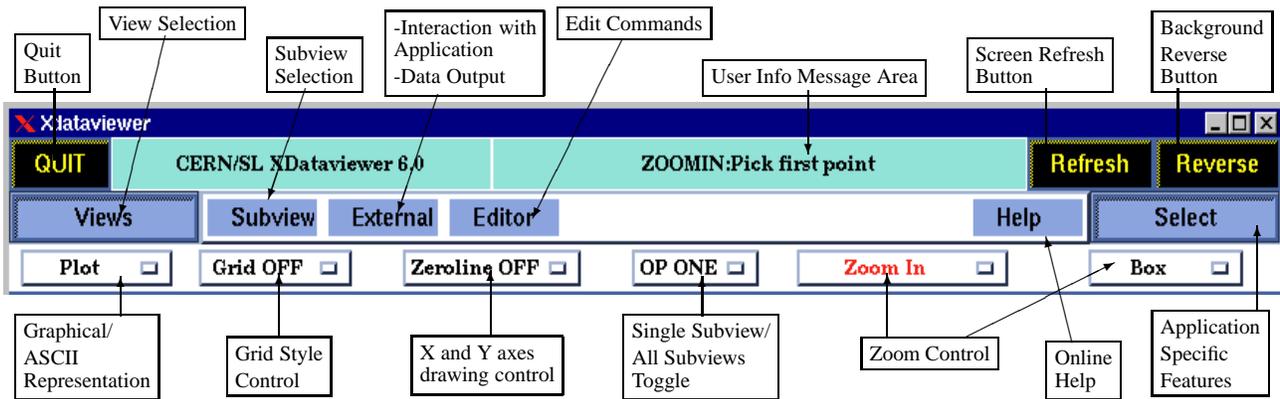


Figure 2: The Dataviewer Control Panel.

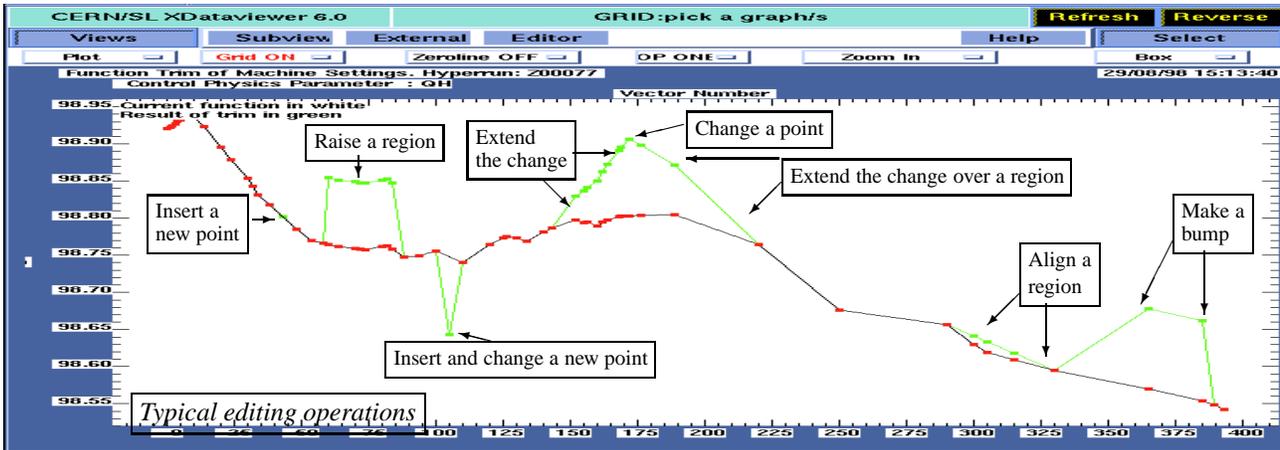


Figure 3: Edit facilities in graphic mode.

### 3.4 The X Window Implementation of the Dataviewer Human Interface and Data Display parts

The Control Panel of the Dataviewer (see Fig. 2) is written using the Motif toolkit, on top of X Window. The actual code is generated by a home made C code generator for X Window applications[4]. The code needed to display the data heavily relies on the Xlib package. As no machine-dependent code is used, the product should be easy to port onto other platforms.

### 3.5 Graphic and Text Data Display

Each Graph can be displayed in two different modes : the graphic mode (default), where each Plot in the Graph is drawn on the Dataviewer part allocated to that Graph in the way specified by the Application Program (Markers, Histograms, Lines...), and the text mode, where the values of the points of one or all of the plots contained by the Graph are written in columns using a default format or a specified one. In text mode, a scrollbar is also displayed, to help the User when moving through a long Plot. If a Graph contains more than one Plot, and the X coordinates of the different Plots are not homogeneous, all these coordinates are merged so that every Plot will find an entry for each of

its points.

### 3.6 Editing Facility

The need for being able to edit a plot derives from the usage of the Dataviewer as a Function Editor, when defining and modifying the behavior of the different LEP equipment during the energy ramp. Both in graphical (via the cursor) and in text (via an input shell) modes it is possible to add, delete, move single points, to align a range of points, to interpolate modifications through a range of points (see Fig. 3).

### 3.7 The Zoom Facility

Using the cursor and the mouse, the User can zoom in and out the data. The axes (X and Y, only X, only Y) to be used for the zoom operation are controlled either by the Interface or by the Application Program. The Dataviewer maintains a linked list of zoom operations, so that it is possible to reverse these operations or to come back to the original configuration.

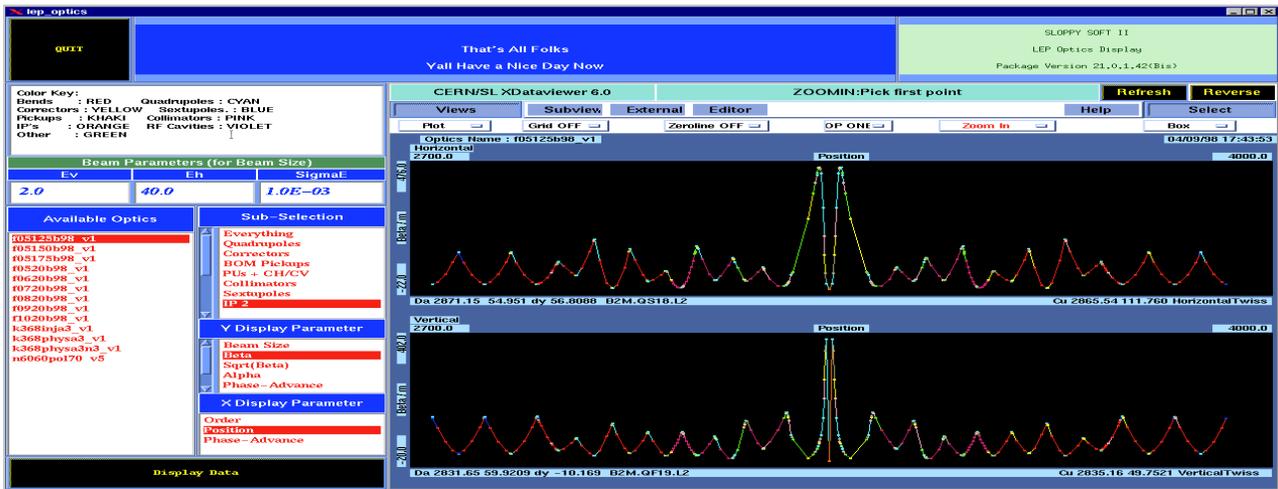


Figure 4: An Application using the Embedded Dataviewer.

### 3.8 The Grid Facility

If specified, a self adaptive Grid will be plot around or across a graph. The spacing of the Grid tics will depend on the graph limits and on the number of screen pixels available for the graph. The format will also be automatically adapted to the data, case by case.

### 3.9 The Help Facility

The Application Programmer can provide, in a text file containing special delimiter lines, explanations on the meaning of the different Plots, Graphs, Views. This information will be displayed on request with the help of a popup window.

### 3.10 Output Facility

The Dataviewer provides utilities to plot the contents of the screen on a printer (either colour or black and white). It may also produce .ps or .eps files containing the same information. Finally, it may print or save onto file the values of all the points in the different Plots in a selected Graph.

### 3.11 Communication with the Application Program

The two typical situations requiring communication between the Dataviewer and the Application are a) the production by the Application of new data to be displayed, and b) the need for the Application to be informed about some User selection on the Dataviewer data display part. In the former case the Application sends a Unix signal to the Dataviewer, which will refresh the data displayed. In the latter case the Dataviewer sends a signal to the Application, which, by means of a C function, may retrieve the relevant information. The Application may also decide to disable this mechanism.

### 3.12 The Embedded Dataviewer

Originally the Dataviewer was always running as a Stand Alone process. In some cases, however, it is desirable to

integrate the graphical part within the same window as the rest of the Application. Due to some features of our code generator, this is actually straightforward. All the widgets composing the Dataviewer part are created within a single C function. A few other functions specific to the Embedded Dataviewer (command line arguments equivalent, or communication between the Application part and the Dataviewer part) have been made available to the Programmers.

### 3.13 The Callable Interface : (Keep It Simple, Stupid !)

The Golden Rule for insuring success to a tool is its easiness of use. Application Programmers like to concentrate on how to solve their problems, and not on how a tool works. Although about 100 C functions are available to tailor everything to the user's need, less than 10 are enough to start :

- `dv_init` : to allocate space for the Dataviewer data structures.
- `dv_vwcreate` : to create a view.
- `dv_grcreate` : to create a graph.
- `dv_grattach` : to attach a graph to a view.
- `dv_plcreate1(2)` : to create a mono(bi)dimensional plot, to display data objects.
- `dv_plattach` : to attach a plot to a graph.
- `dv_kick` : to inform the Dataviewer that new data are ready.

To these functions, one should add less than 10 C functions from the MOPS library (to create, initialize and access a MOPS, and to create and access objects inside it) These 15 functions are enough to start, and to produce perfectly well working Applications; most of the others are used to modify the properties of individual objects (Views, Graph or Plots). Some functions modify the default properties to be assigned to a newly created object, and other will modify the way the Dataviewer works.

## 4 CONCLUSIONS

The Dataviewer has reached the stage of maturity, and the rate of new user requirements is now very low. It is used by most of the Application Programs in the LEP and SPS Control rooms, and it has contributed in saving many man-years of software efforts. It also makes life easier for the Operators, by presenting them different graphical information in an uniform and coherent style.

## 5 ACKNOWLEDGEMENTS

So many persons have provided input to the original Dataviewer design and to all the successive refinements, that is impossible to mention here all of them. Thanks to them, the tool has become very versatile. Ann Sweeney deserves, above everyone else, the most merit for having synthesized all the different ideas and for having implemented them in a clean and well structured piece of code, easy to port, to modify, to extend.

## 6 REFERENCES

- [1] The Dataviewer Programmer's Guide, Ann Sweeney, CERN/SL/CO Note/90-13
- [2] New features for the HP\_UX Dataviewer. Unpublished internal note available from the author (giulio.morpurgo@cern.ch).
- [3] M.O.P.S. User Guide for "C" programs, Werner Herr, CERN-SPS/88-43 (AMS) Revised January 1993.
- [4] Xcreator: a C code generator for AppliXation Programs. Unpublished internal note available from the author (giulio.morpurgo@cern.ch).

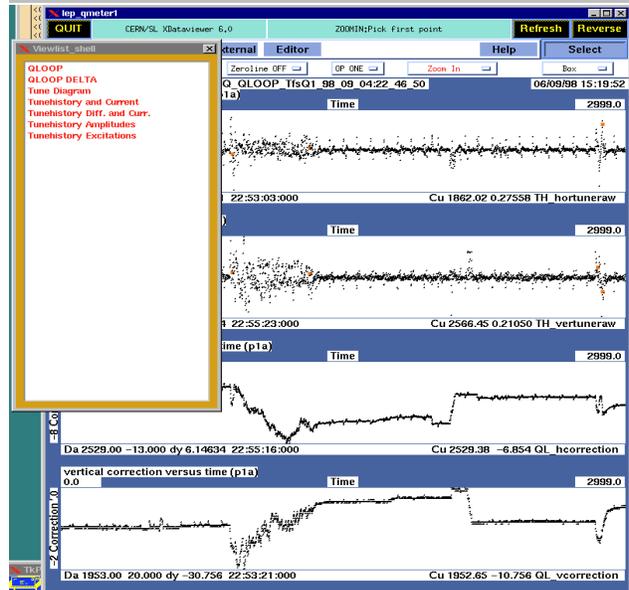
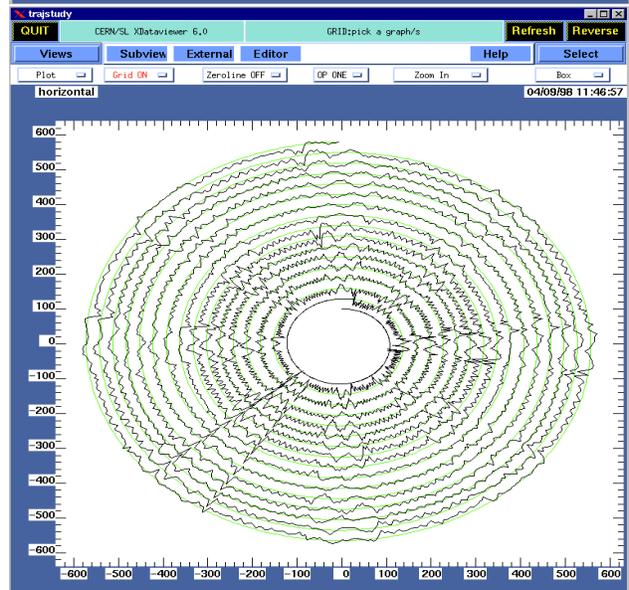
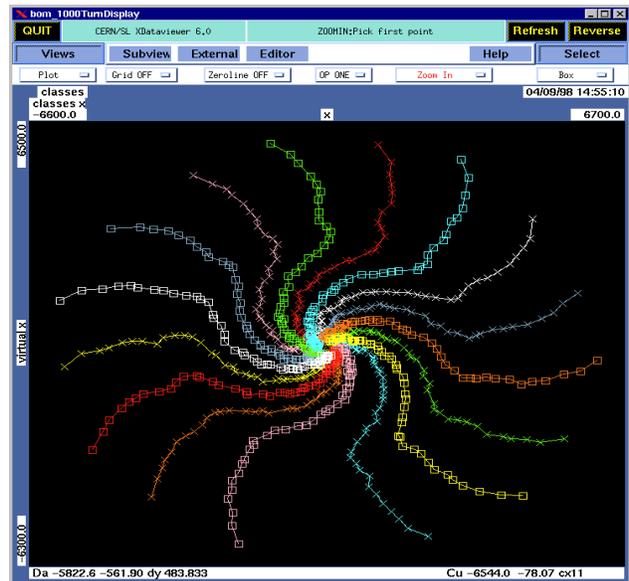


Figure 5: Some Dataviewer applications at LEP

# TECHNIQUES FOR ROBUST NONLINEAR DELTA-F SIMULATIONS OF BEAMS\*

Alex Friedman, John J. Barnard, and David P. Grote  
LLNL, P.O. Box 808, Livermore, CA 94551 USA

## Abstract

We describe means by which the range of applicability of nonlinear-characteristic  $\delta f$  methods to beams may be enhanced, so as to faithfully describe a sharp-edged beam or a smooth beam whose edge moves by more than its scale length. As others have done, we follow a population of Lagrangian characteristic “marker” particles in the total (equilibrium plus perturbation) field. However, in contrast with usual practice, our marker distribution is not proportional to the physical particle distribution. We introduce “ghost” particles: a population of markers loaded into regions of phase space where the equilibrium  $f_0$  is zero or very small. In addition, we do not numerically evolve either  $\delta f$  or a “weight”  $w$ , but rather we use knowledge of the marker positions in phase space and of the functional form of  $f_0$  to evaluate  $\delta f$  anew at each timestep for each marker. We describe the application of our formalism to the model problem of an oscillating displaced beam. We show that marker loading with phase-space density proportional to  $f_0$  leads to inconsistencies, while our modified marker loading is consistent and (for modest displacements) can be statistically “quieter” than conventional particle-in-cell simulation.

## 1 INTRODUCTION

Nonlinear-characteristic  $\delta f$  methods [1-4] have proven highly effective in the simulation of plasmas. However, the application of  $\delta f$  methods to charged particle beams has proven challenging, despite their early use [1] and recent progress [5]. This has been the case largely because few beam equilibria are known. While axisymmetric equilibria are readily constructed, most beams are non-axisymmetric due to the alternating gradient focusing employed. For such beams the only known equilibrium is the “K-V” distribution [6], which is highly singular and exhibits a number of instabilities which are not present in real beams. Furthermore, beams evolve farther from their equilibrium distribution ( $f_0$ ) than do most of the neutral plasmas to which  $\delta f$  methods are applied. Beams are bounded; while some have extended spatial tails, many (those with very strong space charge, or near apertures) do not. Beam parameters can change dramatically along an accelerator, and externally-applied forces can have complicated structure. At least in some regions of phase space, the perturbation  $\delta f$  can be large (sometimes infinitely large) relative to  $f_0$ .

In this paper we explore means by which the range of applicability of  $\delta f$  methods to beams may be enhanced. Despite the above mentioned difficulties, such methods

may still prove to have real utility because the offending regions with large perturbations may be localized, and so integral quantities may be insensitive to such local errors; this must be evaluated on a problem-by-problem basis.

Traditional  $\delta f$  methods, using a marker loading proportional to the physical particle distribution, cannot consistently describe the behavior of a sharp-edged beam undergoing a displacement. We sought revised  $\delta f$  methods which can work even in the extreme limit of a sharp-edged beam. We suggest that two techniques are key. The first is the use of “ghost” particles in a marker distribution not proportional to  $f_0$ , a technique anticipated in earlier  $\delta f$  work [2] but apparently never used for beams. The second is the use of marker coordinates and the functional form of  $f_0$  to evaluate  $\delta f$  anew at each timestep for each marker, rather than evolving  $\delta f$  in time. This saves the solution of an ODE for each particle, with its attendant errors and possible timestep constraints. It follows a comment by Aydemir [3] and is in contrast with what seems to be common practice in the use of  $\delta f$  methods. These methods should also work well when the edge of a smooth-edged beam moves by more than its characteristic density fall-off scale length, typically the thermal Debye length.

In the following sections of this paper we describe our formalism; the model problem of an oscillating displaced beam in slab geometry; results for a water-bag distribution in the zero-space-charge limit; and results for a Maxwell-Boltzmann distribution for a space-charge-dominated beam. Finally, we offer closing comments.

## 2 FORMALISM

The physical distribution  $f(x,v)$  integrates to the total number of particles,  $N$ :

$$N = \int dx dv f(x, v) \quad (1)$$

An idealized marker distribution  $f_m(x,v)$  — the number of markers per unit phase space volume — integrates to the total number of markers,  $N_m$ :

$$N_m = \int dx dv f_m(x, v) \quad (2)$$

We consider  $f$  and  $f_m$  to be continuous and finite; below, we discuss the correspondence to a set of discrete markers  $\{i\}$ . The Vlasov equation in the electrostatic limit is:

$$\frac{df}{dt} = \frac{\partial f}{\partial t} + v \cdot \nabla f + \frac{q}{m} \mathbf{E} \cdot \nabla_v f = 0 \quad (3)$$

and the evolution equation for the perturbation  $\delta f = f - f_0$  is:

$$\frac{d}{dt} \delta f = - \frac{df_0}{dt} = - \frac{q}{m} \delta \mathbf{E} \cdot \nabla_v f_0 \quad (4)$$

(we need not time-advance this!). The unperturbed and perturbation particle number densities are:

$$n_0(x) = \int dv f_0(x, v) \quad \delta n(x) = \int dv \delta f(x, v) \quad (5)$$

The total field, used to advance the markers via a split-leapfrog step, is  $E = E_0 + \delta E$ . The probability densities are  $p(x, v) = f(x, v)/N$  and  $p_m(x, v) = f_m(x, v)/N_m$ ; these integrate to unity and represent the likelihood that a single particle or marker is to be found in a given unit volume of phase space. We define two sets of constants along the marker trajectories:

$$f_i = f(x_i, v_i); \quad f_{mi} = f_m(x_i, v_i) . \quad (6)$$

Assigning a phase-space volume  $V_{mi} = 1/f_{mi}$  to each marker, the ‘‘Klimontovich’’ distribution is:

$$\begin{aligned} f_{Klim}(x, v) &= \sum_{i=1}^{N_m} f(x_i, v_i) V_{mi} \delta(x - x_i) \delta(v - v_i) \\ &= \sum_{i=1}^{N_m} \frac{f_i}{f_{mi}} \delta(x - x_i) \delta(v - v_i) \end{aligned} \quad (7)$$

The moment of a quantity  $A$  is approximated by

$$\begin{aligned} \langle A \rangle &= \frac{1}{N} \int A(x, v) f(x, v) dx dv \\ &\approx \frac{1}{N} \int A(x, v) f_{Klim}(x, v) dx dv \end{aligned} \quad (8)$$

giving:

$$\langle A \rangle \approx \frac{1}{N} \sum_{i=1}^{N_m} A(x_i, v_i) \frac{f_i}{f_{mi}} . \quad (9)$$

Similarly, the Klimontovich distribution for markers is:

$$f_{m,Klim}(x, v) = \sum_{i=1}^{N_m} \delta(x - x_i) \delta(v - v_i) . \quad (10)$$

We replace spatial delta functions  $\delta(x - x_i)$  with the particle ‘‘shape function’’  $S(x_j - x_i)/\Delta x$ , where  $\Delta x$  is the cell size. The charge density in grid cell  $j$  in a conventional PIC code is:

$$n_j = \sum_i \frac{N}{N_m} \frac{S(x_j - x_i)}{\Delta x} \quad (11)$$

or, for non-uniform particle weighting:

$$n_j = \sum_i \frac{N}{N_m} \frac{p_i}{p_{mi}} \frac{S(x_j - x_i)}{\Delta x} = \sum_i \frac{f_i}{f_{mi}} \frac{S(x_j - x_i)}{\Delta x} , \quad (12)$$

where  $p_i = f_i/N$  and  $p_{mi} = f_{mi}/N_m$  remain constant along trajectories, just as do  $f_i$  and  $f_{mi}$ . In a  $\delta f$  calculation, our form for the perturbed charge density is:

$$\delta n_j = \sum_i \left( \frac{f_i - f_0(x_i(t), v_i(t))}{f_{mi}} \right) \frac{S(x_j - x_i)}{\Delta x} \quad (13)$$

or, introducing a perturbed ‘‘weight’’ for each particle  $i$ ,

$$\delta n_j = \sum_i w_i \frac{S(x_j - x_i)}{\Delta x} , \quad (14)$$

where

$$w_i(t) = \frac{N}{N_m} \frac{\delta f_i}{f_i} \frac{p_i}{p_{mi}} = \frac{\delta f(x_i(t), v_i(t))}{f_{mi}} . \quad (15)$$

This matches Aydemir’s eq. (19), noting the differing notation, *e.g.*, his ‘‘ $N$ ’’ is our  $N_m$ . It also matches Parker and Lee’s eq. (7), again with differences in notation. It differs from the common expression for PIC-like loading in that the denominator is  $f_{mi}$  instead of  $f_i$ .

Each particle makes a contribution to  $\delta n$  proportional to  $(f_i - f_0)/f_{mi}$ . Here, the first term simply advects around with the markers and always contributes positively to  $\delta n$ . The second term is zero or contributes negatively to  $\delta n$ , as necessary to cancel the unwanted equilibrium  $n_0$  in regions not currently being occupied by the beam.

Direct evaluation is simpler than time-evolution. Still following Aydemir, from the equation for  $d\delta f/dt$  we find:

$$\frac{dw_i}{dt} = - \left[ \frac{1}{f_{mi}} \frac{df_0}{dt} \right]_{(x,v)=(x_i,v_i)} . \quad (16)$$

As Aydemir notes, it is ‘‘under normal circumstances’’ not necessary to solve this, since  $\delta f_i$  or  $w_i$  at any time can readily be computed from:

$$\begin{aligned} f_0(x_i, v_i) & \text{(by direct evaluation of the specified } f_0), \\ f_i & \text{(constant, by Liouville’s theorem), and} \\ f_{mi} & \text{(constant, by Liouville’s theorem).} \end{aligned}$$

This is a considerable advantage for beams, where  $f_0$  may change radically along the exact orbits. Two quantities ( $f_i$  and  $f_{mi}$ ) need to be stored for direct evaluation, just as two quantities ( $w_i$  and  $f_{mi}$ ) need to be stored for time-evolution. Only in the case of PIC-like loading of markers, where  $f_i/f_{mi}$  is the same number for all particles, is it possible to dispense with one of these stored quantities. This savings is unlikely to be significant.

### 3 MODEL PROBLEM

A sheet beam subjected at  $t=0$  to a constant, uniform sideways force illustrates the relevant principles. For the moment, we assume infinitesimal space charge, using the accumulated number density only as a measure of the effectiveness of our methods (this assumption will be relaxed in Section 5, below). The beam moves with constant axial velocity  $v_z = v_0$ . The transverse force balance is between thermal pressure and an applied linear confining field  $E_0(x) = -E'_0 x$ . The betatron frequency is  $\omega$ , where  $\omega^2 = qE'_0/m$ . A ‘‘slice’’ of particles passes through  $z=0$  at time  $t=0$ , at which time a transverse electric field  $\delta E$  is applied; the effect is to shift the bottom of the electrostatic potential well by a distance  $x_c = q\delta E/m\omega^2 = \delta E/E'_0$ . See Figure 1. Since there is no damping, the beam’s centroid overshoots to  $2x_c$ , and the system rings forever at frequency  $\omega$ . See Figure 2, which shows an extreme case. In phase space, the beam precesses around  $(x_c, 0)$ ; see Figures 3 and 4.

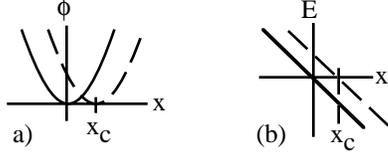


Figure 1. For the model problem in the zero-space-charge limit, (a) potential and (b) electric field vs. transverse coordinate  $x$ . Solid line is unperturbed, dashed perturbed.

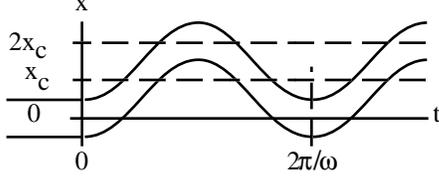


Figure 2. Beam motion in  $\{x,t\}$  space for model problem.

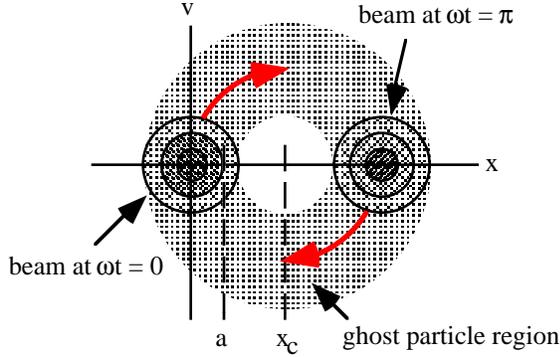


Figure 3. Beam motion in phase space for model problem.

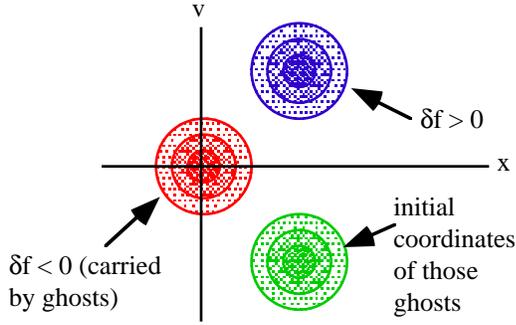


Figure 4. Phase space at  $\omega t = \pi/2$  for model problem.

#### 4 WATERBAG DISTRIBUTION IN ZERO SPACE CHARGE LIMIT

A waterbag distribution illustrates the issues. This is a “sharp-edged” equilibrium with negligible space charge. We set the phase space density constant out to a cutoff circle at energy  $h_0$  in the phase space  $\{\omega x, v\}$ :

$$f_0(x, v) = \begin{cases} \frac{N\omega}{2\pi h_0}, & v^2 + \omega^2 x^2 \leq 2h_0 \\ 0, & v^2 + \omega^2 x^2 > 2h_0 \end{cases}, \quad (17)$$

leading to a particle density:

$$n_0(x) = \begin{cases} \frac{N\omega}{\pi h_0} \sqrt{2h_0 - \omega^2 x^2}, & |x| \leq \frac{\sqrt{2h_0}}{\omega} \\ 0, & |x| > \frac{\sqrt{2h_0}}{\omega} \end{cases}. \quad (18)$$

The displacement of this distribution cannot be simulated consistently with a  $\delta f$  simulation using PIC-like loading.

We consider two marker loadings which afford consistent  $\delta f$ -derived densities for the waterbag problem. For both, the boundaries of the marker region are time-invariant, since  $E$  is time-invariant and markers are uniformly loaded along characteristics (in the presence of the displacing force). Loadings such as these only work for negligible space charge. They are:

(1) A “disk” loading, which uniformly populates a “circle” in phase space of radius  $r_d = x_c + \sqrt{2h_0} / \omega$  centered at  $(x_c, 0)$ :

$$f_m = N_m / \pi (x_c + \sqrt{2h_0} / \omega)^2 = \text{const.} \quad (19)$$

(2) An optimized “annulus” loading, wherein the markers uniformly populate an annulus of outer and inner radii  $r_{\pm} = x_c \pm \sqrt{2h_0} / \omega$  centered at  $(x_c, 0)$ :

$$f_m = N_m / \pi \left[ (x_c + \sqrt{2h_0} / \omega)^2 - (x_c - \sqrt{2h_0} / \omega)^2 \right] = \text{const.} \quad (20)$$

In carrying out these waterbag tests we were able to take advantage of the fact that the orbits in the applied field are readily computed. Rather than time-stepping as in an actual PIC code, we evaluated marker locations at  $\omega t = \pi$  analytically using:

$$x - x_c = (x_0 - x_c) \cos(\omega t) + \frac{v_0}{\omega} \sin(\omega t) \quad (21)$$

$$v = -\omega(x_0 - x_c) \sin(\omega t) + v_0 \cos(\omega t)$$

This is possible because (for this model) the space charge is negligible, used merely as a diagnostic. A 1-d version of area weighting was used; results for nearest-grid-point charge deposition are similar. Best agreement was obtained by using the analytic density averaged over a length  $\Delta x$  centered at each grid point as a reference density; again, differences are minor. Results are shown for conventional PIC, delta-f using PIC-like loading (marker density proportional to  $f_0$ ), and the two marker loadings described earlier. In all cases, the “physical” particle number  $N = 10^9$ , number of markers  $N_m = 25,000$ , number of grid cells  $n_g = 400$ , and limits of the computational domain  $x_l = -5$  and  $x_h = 15$ .

In Figure 5 the density at  $\omega t = \pi$  is shown. Note the failure of  $\delta f$  with PIC-like loading to cancel the unwanted zero-order density at the left. The relative error vs.  $x$  for  $x_c = 0.05$  is shown in Figure 6, and its variation with  $x_c$  in Figure 7. Even annulus-loading  $\delta f$  is inferior to PIC beyond  $x_c \sim 0.4$ , though it remains consistent.

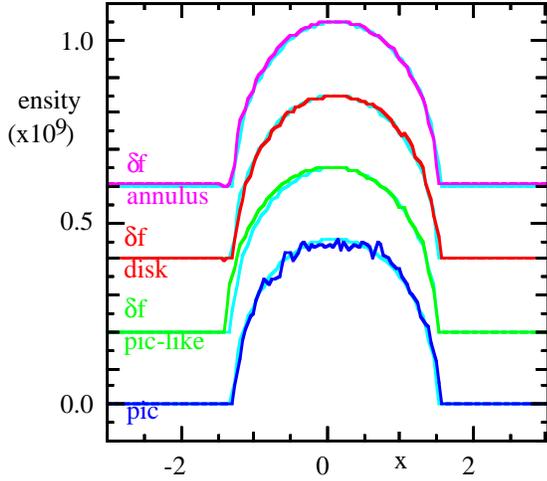


Figure 5. Waterbag problem: density vs.  $x$  at  $\omega t = \pi$  for  $x_c = 0.05$ ; an offset between the ordinates aids clarity. The analytic density is shown in light color.

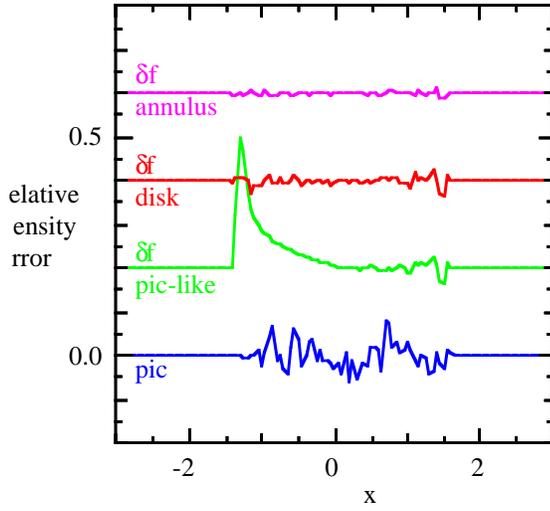


Figure 6. Density error normalized to peak density vs.  $x$  at  $\omega t = \pi$  (note offset)

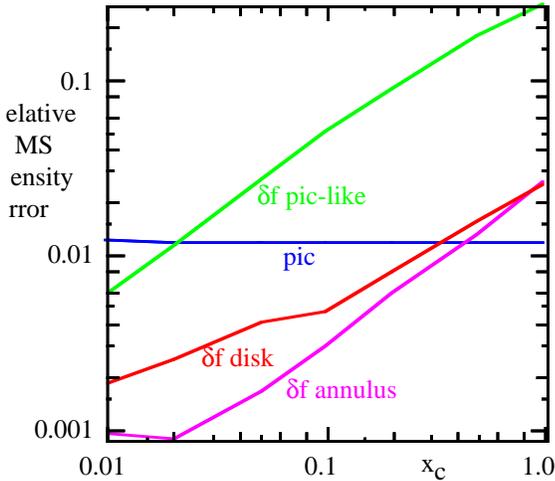


Figure 7. RMS over  $x$  of density error (normalized to peak density) vs. offset  $x_c$  at  $\omega t = \pi$ .

## 5 MAXWELL-BOLTZMANN DISTRIBUTION WITH SPACE CHARGE

When space charge is not negligible, marker loading requires care.  $E$  varies with time, and loadings “centered” about  $x_c$  are not stationary. For the model problem, quasi-stationary marker distributions can be obtained by loading markers uniformly along the equilibrium orbits in the unperturbed field. Since the applied perturbation field shifts all orbits (physical and marker) by the same phase-space offset, the marker distribution is stationary about the moving beam centroid. If the marker distribution is made “large enough,” the phase-space area vacated by the precessing beam is always “well-covered” with markers and the unwanted  $n_0$  is canceled. However, such loadings are in general less “efficient” than the annulus loading that works well for negligible space charge.

We are testing various marker loadings; here we use a rescaled PIC-like loading:  $x_j \leftarrow [(2x_c + a_0)/a_0]x_j$ ;  $v_j \leftarrow [2x_c\omega\beta_0/v_{th}]v_j$ . The distribution value  $f_{mj}$  for each marker  $j$  is reduced relative to that of PIC-like loading by the product of the factors in  $[\ ]$ . This sub-optimal “blob” loading is not uniform along unperturbed orbits, and the marker distribution is not quasi-stationary. Nonetheless, it covers the required phase space. Work in progress (to be described in a future publication) indicates that improved results (in some cases significantly better than those of PIC) are obtainable from a cut-off uniform marker loading.

We have done PIC and  $\delta f$  simulations of a space-charge-dominated Maxwell-Boltzmann (M-B) sheet beam. The equilibrium, which must be computed numerically, is:

$$f = f_0 \exp\left(-\frac{H_{\perp}}{k_B T_{\perp}}\right) = f_0 \exp\left(-\frac{mv_x^2/2 + q\phi_{tot}(x)}{k_B T_{\perp}}\right). \quad (22)$$

We chose a strongly tune-depressed case, with ratio of depressed to undepressed phase advance  $\sigma/\sigma_0 = 0.2$ . Also:  $a_0 = 0.01$  m (edge position of RMS-equivalent uniform beam);  $n_x = 512$  (number of cells across  $x$ );  $x_c = 0.002$  m (offset of equilibrium  $x$  due to displacing force);  $n_m = 32768$  (number of markers);  $\Delta t/\tau\beta_0 = 1/16$  or  $1/32$  (the latter gives visibly better results).

In figure 8 the initial PIC and marker locations are shown; the “ghost” region is sizable (these figures are best viewed online, in color). In Figure 9 the initial PIC beam is again shown, now overlaid with the PIC beam and the markers at  $\omega\beta_0 t = \pi$ . In Figure 10 the markers with the most significant perturbations are displayed in dark colors (for clarity, more markers were used in this run; when a smaller time step is used, the regions become roughly circular). In Figure 11 the profile at  $\omega\beta_0 t = \pi$  is shown for the three methods. The failure of  $\delta f$  with PIC-like loading is evident. When the modified loading is used,  $\delta f$  is smoother than PIC over most of the beam, but noisier at the beam edges (improved loadings can fix this).

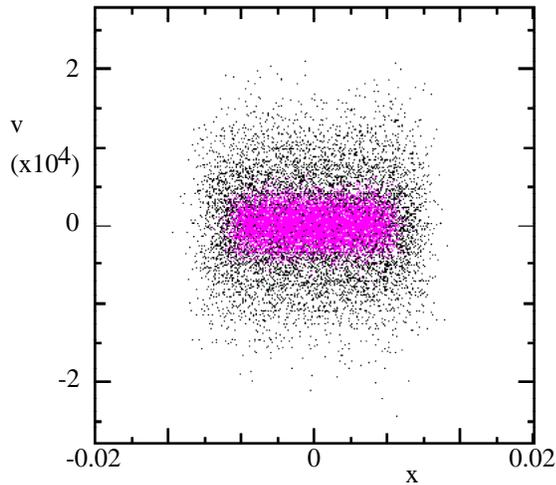


Figure 8. Initial PIC beam (magenta) and markers (black) for “blob” loading of M-B beam with space-charge.

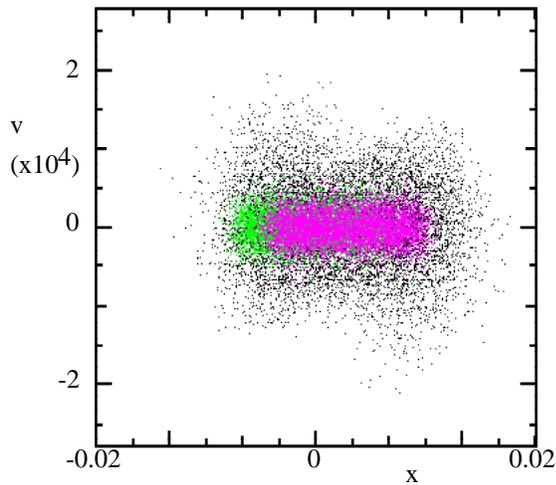


Figure 9. Initial PIC beam (green), PIC beam at  $\omega_{\beta 0} t = \pi$  (magenta), and markers at  $\omega_{\beta 0} t = \pi$  (black).

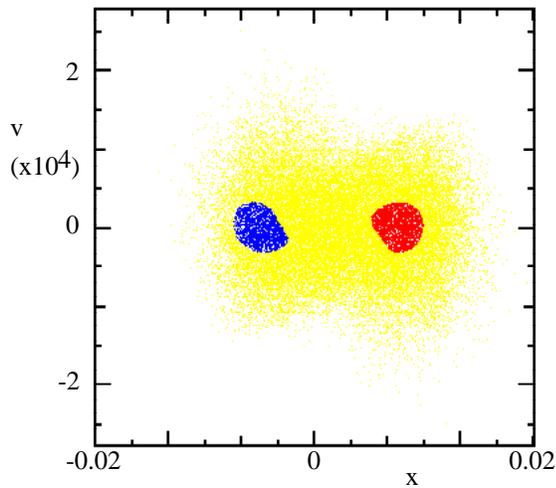


Figure 10. Markers at  $\omega_{\beta 0} t = \pi$ ; Red denotes  $\delta f \geq 0.2 f_{\max}$ ; blue  $\delta f \leq -0.2 f_{\max}$ ; yellow all other

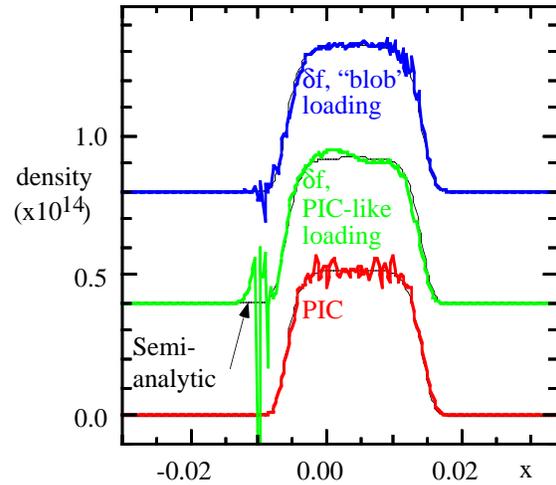


Figure 11. Density profiles at  $\omega_{\beta 0} t = \pi$  for M-B beam.

## 6 DISCUSSION

We have identified  $\delta f$  methods which handle even a sharp-edged beam, and shown that use of PIC-like marker loading leads to inconsistencies in such cases. The K-V distribution is sharp-edged (and singular, in phase space). If  $\delta f$  methods are to be reliably applied to it, great care must be taken to ensure cancellation of undesired  $n_0$  as the thin shell in phase space deforms. This may prove very difficult to do consistently.

Timestep constraints associated with accurately tracking the markers in the total field remain. In many cases these constraints resemble those of PIC, since the marker advance is identical with that of PIC, except that the field solution is unconventional. In other cases the constraints may be more severe, since (as was shown) markers must sometimes be followed in regions of phase space that remain unpopulated in PIC calculations.

Because of their difficulties when the beam evolves considerably, we should not expect these methods (at least in their present form) to become a general replacement for PIC beam simulations. Their extra “quietness” makes these robust  $\delta f$  variants attractive for special purposes, e.g., the detailed study of particular modes on a beam.

## REFERENCES

- \* Work performed under the auspices of the U.S. D.O.E. by LLNL under contract W-7405-ENG-48.
- [1] W. M. Fawley, *BAPS* **27**, 1034 (Oct. 1982).
- [2] S. E. Parker and W. W. Lee, *Phys. Fluids B* **5**(1), 77-86 (1993).
- [3] A. Y. Aydemir, *Phys. Plasmas* **1**(4), 822-31 (1994).
- [4] G. Hu and J. A. Krommes, *Phys. Plasmas* **1**(4), 863-74 (1994).
- [5] P. H. Stoltz, R. C. Davidson & W. W. Lee, *Phys. Plasmas* **5**, 1998 (in press).
- [6] I. Kapchinskij and V. Vladimirskij, in *Proc. Int. Conf. on High Energy Accel. and Instrumentation* (CERN Scientific Information Service, Geneva, 1959), p. 274.

# DYNAMIC SPACE CHARGE CALCULATIONS FOR HIGH INTENSITY BEAMS IN RINGS\*

J. A. Holmes, J. D. Galambos, D. Jeon, D. K. Olsen, J. W. Cobb,  
SNS Building, Oak Ridge National Laboratory, Oak Ridge, TN 37831-8218  
M. Blaskiewicz, A. U. Luccio, and J. Beebe-Wang,  
Brookhaven National Laboratory, Upton, NY 11973, USA

## Abstract

Space-charge-induced emittance growth and halo generation could lead to unacceptably high beam loss in high intensity rings, such as the SNS [1]. In such accelerators, uncontrolled losses to the walls as small as one part in  $10^4$  would lead to activation, making maintenance difficult. For this reason it is essential to understand the effects of space charge on beam dynamics, and halo generation in particular, in high intensity rings. We have undertaken the study of space charge dynamics in high intensity rings using a particle tracking approach, with self-consistent evaluation of the space charge forces through a particle-in-cell model. Because of the stringent loss requirements, it is necessary to thoroughly guarantee the reliability of these calculations to high precision through comparison with experiments and through convergence studies. In this paper we present the results of convergence studies in the parameters of the model, namely, the number of macroparticles, the resolution in the adopted FFT algorithm, the smoothing parameter, and the time step size. Although present calculations have been extended to more than  $10^5$  macroparticles on individual UNIX workstations, it will be necessary to increase another one to two orders of magnitude to obtain the necessary precision. To accomplish this, we have constructed and are now using a LINUX parallel computer from low cost components.

## 1 INTRODUCTION

The operating requirements of some new rings, such as the Spallation Neutron Source (SNS) accumulator ring, include high intensity beams, low uncontrolled losses, and a considerable beam flight path in the ring. In these circumstances, space-charge-induced halo generation is a potential loss mechanism, and requires study. High intensity rings are characterized by the separation of longitudinal and transverse scales. In SNS, for example, the longitudinal bunch length is on the order of  $100m$ , compared with transverse beam dimensions of a few  $cm$ ; and the longitudinal tune is about  $10^{-3}$ , compared with transverse tunes of about 5.8. For this reason it is possible, with good approximation, to separate the longitudinal and transverse dynamics in high intensity rings and, for the study of space charge effects, to

consider the transverse dynamics. This simplification allows the description in terms of four, rather than six, dimensional phase space, which necessitates a smaller solution space to obtain a given numerical precision than is required in six dimensions.

In order to study transverse beam dynamics in high intensity rings, and space charge effects in particular, we have adopted a particle-tracking approach [2]. The integration scheme is chosen to be second order symplectic, with a matrix representation of all linear focusing elements, including dispersion, and the inclusion of all nonlinear effects as kicks. Our treatment of space charge uses a particle-in-cell (PIC) model [2] with fast Fourier transforms (FFTs) to evaluate the forces. This is carried out in a modified version of the particle tracking and injection code, ACCSIM [3], and in a new code, SAMBA, which we are now developing.

In PIC calculations, numerical convergence and accuracy must be carefully assured. We present in this paper the results of convergence studies in the parameters of the model. Section 2 presents the equations and numerical solution of the model; Section 3 presents the convergence results; and Section 4 contains discussion and conclusions.

## 2 EQUATIONS AND NUMERICAL SOLUTION

The essence of our transverse particle tracking model is the following pair of dynamic equations for the macroparticle coordinates  $x$  and  $y$  :

$$\begin{aligned} x'' + K_x(s)x &= F_x^{sc} + F_x^{nl} + \frac{1}{\rho(s)} \frac{\delta p}{p_0} \\ y'' + K_y(s)y &= F_y^{sc} + F_y^{nl} \end{aligned} \quad (1)$$

where  $s$  is the azimuthal coordinate,  $K_{x,y}(s)$  are the linear magnet focusing strengths,  $F_{x,y}^{sc}(s)$  are the space charge forces,  $F_{x,y}^{nl}(s)$  are the external nonlinear forces, and  $\frac{1}{\rho(s)} \frac{\delta p}{p_0}$  is the dispersion term. The linear focusing forces, external nonlinear forces, and bending radius are all dependent on the lattice; while the momentum

deviation  $\frac{\delta p}{p_0}$  is determined by the longitudinal dynamics. The space charge forces are evaluated self-consistently:

$$\begin{aligned} F_x^{sc} &= P \sum_{i=1}^N \frac{x-x_i}{(x-x_i)^2 + (y-y_i)^2 + \mathcal{E}^2} \\ F_y^{sc} &= P \sum_{i=1}^N \frac{y-y_i}{(x-x_i)^2 + (y-y_i)^2 + \mathcal{E}^2} \end{aligned} \quad (2)$$

where the summation is over the number of macroparticles  $N$ ,  $P = \frac{2Z^2 r_p \lambda}{\gamma^3 \beta^2 AN}$ ,  $Z$  is the beam charge number,  $A$  is the mass number,  $r_p$  is the classical radius of the beam particle,  $\gamma$  and  $\beta$  are the relativistic kinematic parameters,  $\lambda$  is the longitudinal beam particle density, and  $\mathcal{E}$  is a numerical smoothing parameter.

The numerical model solves the dynamic equations using an explicit, second order, symplectic integration scheme:

- Divide the lattice into  $N_{az}$  linear elements (drifts, bends, quadrupoles);
- Transport macroparticles through these elements using a transport matrix approach, including dispersion;
- Treat nonlinear forces as kicks, applied at the ends of linear elements with strengths appropriate to a second order symplectic scheme;
- External nonlinear forces can be applied independently to each macroparticle;
- Space charge forces involve interaction of all beam macroparticles, and require special treatment.

The space charge forces are evaluated as nonlinear kicks using a PIC model and FFTs:

- At each azimuth  $s$ , select a regular rectangular  $(x, y)$  mesh of  $N_{FFT} \times N_{FFT}$  points, centered on the beam. Because of the periodicity of the FFT, the mesh must be at least twice the extent of the beam in each direction;
- Obtain the particle density on the mesh by bilinear distribution of the macroparticle charges to adjacent mesh points;
- Obtain the FFT of space charge forces at mesh points as the convolution of the FFT of the particle density that of the force due to a unit charge. The inverse FFT of this quantity gives the space charge forces on the mesh points.

- Obtain the space charge force on each macroparticle as a bilinear interpolation of the forces at the mesh points to the location of the macroparticle.

The advantage of using this FFT procedure is one of speed, with the number of operations scaling as  $N$  for particle distribution and force interpolation and as  $N_{FFT} \times \ln(N_{FFT})$  for the transforms. A limitation of the FFT is that it is not conducive to the inclusion of wall effects.

The study of space charge using this numerical model involves a number of parameters:

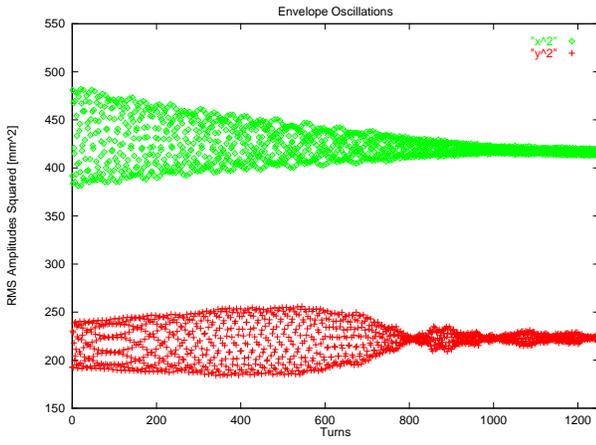
- $N$  the number of macroparticles;
- $N_{FFT}$  the spatial resolution (grid parameter for the FFT algorithm);
- $\mathcal{E}$  the smoothing parameter; and
- $N_{az}$  the azimuthal integration step size.

We now study the convergence properties of an example calculation with respect these parameters.

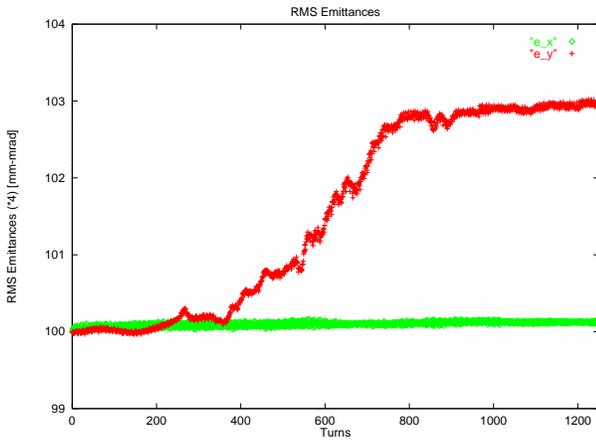
### 3 CONVERGENCE OF THE MODEL FOR AN EXAMPLE CASE

We now study the effect of the parameters of the numerical scheme on the convergence of the model for an example case, namely halo generation by the parametric resonance [4] in a doublet lattice. We consider a doublet lattice with fourfold symmetry similar to the SNS FODO lattice, length  $220.668m$ , having linear focusing only (48 quadrupoles, 32 sector bends, and 80 drifts) and bare tunes of  $\nu_x=5.85$  and  $\nu_y=5.70$ . We consider a coasting beam with a K-V distribution, energy and energy spread  $E_0=1GeV$ ,  $\Delta E_{max}=9.4MeV$ ,  $\Delta E_{rms}=4.7MeV$ ,  $x$  and  $y$  emittances  $\mathcal{E}_{x,y}=100\pi mm\text{-mrad}$ , and number of particles  $N=3.08 \times 10^{14}$ . The initial beam is rms mismatched, resulting in envelope oscillations of about 10% around the matched values. We follow the subsequent evolution for 1250 turns.

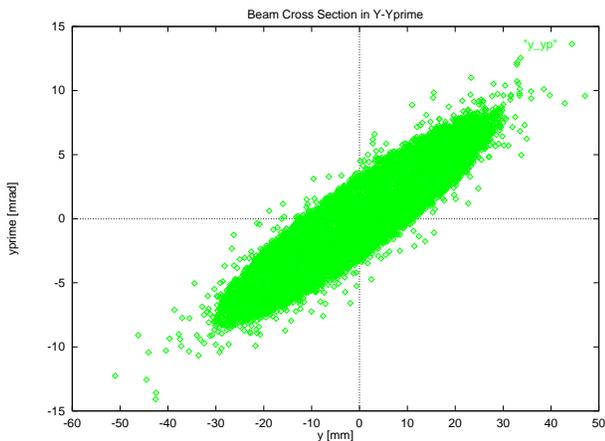
The evolution is observed to be as follows: Because the beam is mismatched envelope oscillations occur, as shown in Fig. 1 which plots the beam averages of  $\langle (x-\langle x \rangle)^2 \rangle$  and  $\langle (y-\langle y \rangle)^2 \rangle$  ( $\langle (\Delta x)^2 \rangle$  and  $\langle (\Delta y)^2 \rangle$ ). Because the  $y$  tune is less than the  $x$  tune, the focusing is slightly weaker in  $y$  than in  $x$ , and through coupling the oscillation energy is transferred from the  $x$  direction into the  $y$  direction. As the beam relaxes some particles cross the separatrix of the parametric resonance, driven by the oscillations in  $\langle (\Delta y)^2 \rangle$ , becoming halo particles and leading to a growth in the rms  $y$  emittance (Fig. 2). This halo



**Figure 1.** Average values of  $\langle (\Delta x)^2 \rangle$  and  $\langle (\Delta y)^2 \rangle$  plotted once each turn and showing envelope oscillations.



**Figure 2.** Transverse rms emittances  $\varepsilon_x$  and  $\varepsilon_y$  versus turn number.



**Figure 3.** Final beam distribution in  $y - y'$  phase space showing beam core and halo particles.

generating process removes energy from the envelope oscillations, which then diminish. As this driving force is removed, the instability saturates (Fig. 2) with the beam in a new steady state with halo. Figure 3 plots the final beam cross section in the  $y - y'$  phase plane.

That the observed evolution is a result of physics, and not a numerical instability, was tested by carrying out the same calculation with the only difference being the method for distributing the charge to the FFT grid. In this latter method, each particle is regarded as a representative of a population of macroparticles, all having the same action values  $(J_x, J_y)$  but distributed uniformly over betatron oscillation phases  $(\phi_x, \phi_y)$ . In this “phase averaged” model envelope oscillations are suppressed so that the parametric resonance is not driven, but no changes are made to numerical methods or parameters. The result of the phase averaged calculation is emittance conservation and no halo generation, as expected for a stable numerical scheme. This indicates that the observed instability is a result of the dynamics.

We also performed convergence studies in the four numerical parameters  $N$ ,  $N_{FFT}$ ,  $\varepsilon$ , and  $N_{az}$ . Figures 1-3 are taken from the “base case” having  $N=7680$  macroparticles,  $N_{FFT} \times N_{FFT} = 32 \times 32$  FFT grid cells,  $\varepsilon = 4.42 \text{mm}$  ( $>$  cell size at all azimuths), and  $N_{az} = 488$  steps/turn. Convergence studies included the following ranges of parameters:

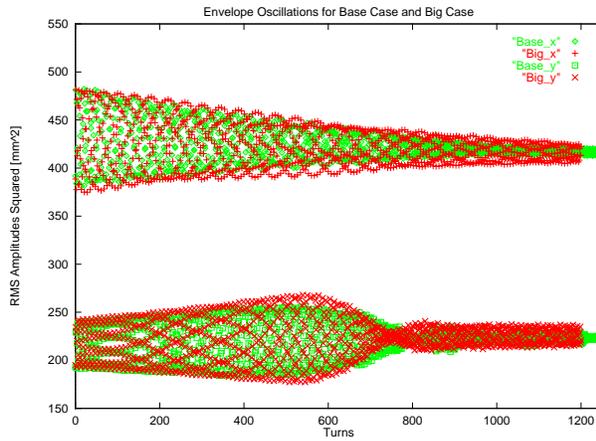
- $N = 7680, 30720, 122880, 245760$
- $N_{FFT} \times N_{FFT} = 32 \times 32, 64 \times 64, 128 \times 128$
- $\varepsilon = 6.25, 4.42, 3.125 \text{mm}$
- $N_{az} = 488, 880$

In the macroparticle variation, the main effects of increasing the number of macroparticles are a delay in the onset of the instability and a faster rise of the instability following onset. The initial and final configurations are basically unchanged as are the main features of the evolution. The same effect is observed when the number of FFT grid points is increased. In both of these cases, the passage to higher resolution decreases the numerical viscosity, thus sharpening the observed evolution.

Choice of the smoothing parameter is perhaps one of the more subjective aspects of PIC calculations by our chosen method. Too small a value obscures the collective dynamics in noise due to binary interactions of macroparticles with nearby FFT nodes. Too large a value smooths away the dynamics. As a guiding constraint, we maintained the values of the smoothing parameter to be larger than the FFT cell size in all cases. As the value of the smoothing parameter was varied, it was observed that increasing the smoothing parameter results in a decreased saturation level of the instability, both in terms of rms emittances and fraction of beam particles in the halo.

Finally, increasing the number of azimuthal integration points resulted in no basic changes in the results.

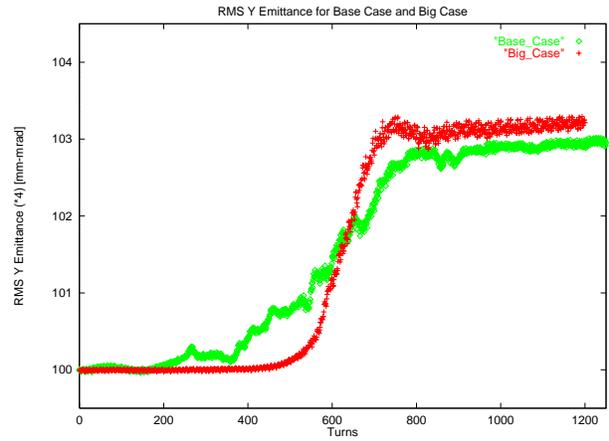
In order to illustrate the effects of the numerical parameter variations, we now compare the beam evolution for the “base case” and a “big case” having the following parameter values:  $N = 245760$  macroparticles,  $N_{FFT} \times N_{FFT} = 128 \times 128$  FFT grid cells,  $\mathcal{E} = 4.42 \text{mm}$ , and  $N_{az} = 880$  steps/turn. In both cases the main features of the beam evolution are the same. In the “big case” the onset of instability is delayed, the growth of the instability is faster, and the beam halo at saturation contains a slightly higher fraction of the beam. Figure 4 shows the beam averages of  $\langle (\Delta x)^2 \rangle$  and  $\langle (\Delta y)^2 \rangle$  for the two cases plotted versus turn number. The most noticeable difference is in the decrease of the vertical fluctuations driving the instability. In the “big case” this decrease is both delayed and sharper relative to the “base case”. Figure 5 plots the evolution of the rms  $y$  emittances for the two cases, and it is again seen that the rise in the emittance for the “big case” is delayed but sharper in comparison with the “base case”. The difference in saturation value of the two emittances is very small, with that of the “big case” being slightly larger.



**Figure 4.** Average values of  $\langle (\Delta x)^2 \rangle$  and  $\langle (\Delta y)^2 \rangle$  plotted once each turn and showing envelope oscillations. Plots are for the “base case” and the “big case”.

#### 4 DISCUSSION AND CONCLUSIONS

Convergence studies have been carried out for the calculation of halo generation via the parametric resonance driven by rms mismatch. In these studies the number of macroparticles per FFT grid cell varied from a minimum of 7.5 to a maximum of 30, although the effective number in the occupied cells is at least four



**Figure 5.** Transverse rms emittances  $\epsilon_y$  versus turn number for the “base case” and the “big case”.

times higher since the FFT grid is twice the extent of the beam in both directions. The number of integration points per quadrupole pair is 20 in the case of 488 integration steps and 37 for 880 steps. It was found that the “base case” of 7680 macroparticles resulted in a reasonable description of the overall dynamics of the beam. Increased resolution revealed a delay in the onset of the instability together with a faster rise time. However, in the case studied here more than 1% of the beam macroparticles migrated to the halo and the vertical rms emittance growth was more than 3%. In high intensity rings, such as SNS, the limit on uncontrolled losses is  $10^{-4}$ , or two orders of magnitude below the observed losses here. To accurately calculate the dynamics of cases with such small losses, high resolution will be a requirement.

In order to perform high resolution calculations in a routine fashion, we will require greater computing resources than have been used here. The present calculations were performed on a variety of IBM RS-6000 and DEC Alpha workstations. The fastest of these machines was a 400MHz Alpha, and the timings on this machine were: approximately 2 turns/minute, or 10 hours for 1250 turns, for the “base case”; and about 3 turns/hour, or 400 hours for 1250 turns, for the “big case”. In order to perform high resolution calculations routinely, we have taken two steps: we have obtained accounts at NERSC, and we have assembled our own parallel computer, the SNS Wonderland Cluster. This cluster consists of five 533 MHz DEC Alpha computers (one gateway node and four compute nodes), each with 4MB of 9ns SRAM Cache, 256MB memory, 4.5GB SCSI Hard Drive, and 10/100 Ethernet. The gateway node contains an additional 18GB storage, and the cluster is connected by a 10/100 Ethernet switch. The cluster is running the RedHat Linux operating system and supports PVM and MPI message passing interfaces. Because particle tracking calculations are CPU and cache bound, the 533MHz processor speed and 4MB cache are well chosen.

Also, the cluster is easily upgradeable, so that more compute power can be added when desired.

## 5 ACKNOWLEDGEMENT

\* Research on the Spallation Neutron Source is sponsored by the Division of Materials Science, U.S. Department of Energy, under contract number DE-AC05-96OR22464 with Lockheed Martin Energy Research Corporation for Oak Ridge National Laboratory.

## REFERENCES

[1] National Spallation Neutron Source Conceptual Design Report, Volumes 1 and 2, NSNS/CDR-2/V1,2, (May, 1997); On the World Wide Web at

- <http://www.ornl.gov/~nSNS/CDRDocuments/CDR.html>
- [2] R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles*, Adam Hilger, IOP Publishing Ltd. (New York: 1988); C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation*, McGraw-Hill Book Company (New York: 1985).
- [3] Jones, F., *Users' Guide to ACCSIM*, TRIUMF Design Note TRI-DN-90-17, (1990).
- [4] J. S. O'Connell, T. P. Wangler, R. S. Mills, and K. R. Crandall, Proc. 1993 Part. Accel. Conf., Washington, DC (1993) 3657; J. M. Lagniel, Nucl. Inst. Meth. A345 (1994) 46; J. M. Lagniel, Nucl. Inst. Meth. A345 (1994) 405; R. L. Gluckstern, Phys. Rev. Lett. 73 (1994) 1247.

# MAP COMPUTATION FROM MAGNETIC FIELD DATA AND APPLICATION TO THE LHC HIGH-GRADIENT QUADRUPOLES

M. Venturini, D. Abell, and A. Dragt,

Physics Department, University of Maryland, College Park MD 20742 USA

*Abstract*

In many cases the most accurate information about fields in a magnet comes either from direct measurement or from a numerical computation done with a 3D electromagnetic code. In this paper we show how this information can be used to compute transfer maps with high accuracy. The resulting transfer maps take into account all effects of real beamline elements including fringe-field and multipole error effects. The method we employ automatically incorporates the smoothing properties of the Laplace Green function. Consequently, it is robust against both measurement and electromagnetic code errors. As an example, we apply the method to a study of end effects in the High-Gradient quadrupoles for the low-beta insertion in the Large Hadron Collider (LHC). The map computation for the quadrupoles has been carried out using MARYLIE's GENMAP routine and a newly written MARYLIE user-defined routine. For long-term tracking we used Cremona symplectification techniques as implemented in CTRACK.

## 1 INTRODUCTION

The motion of charged particles through any beam-line element is described by the transfer map  $\mathcal{M}$  for that element. Through aberrations of order  $(n - 1)$  such a map has the Lie representation [1, 2]

$$\mathcal{M} = \mathcal{R}_2 \exp(: f_3 :) \exp(: f_4 :) \cdots \exp(: f_n :). \quad (1)$$

The linear map  $\mathcal{R}_2$  and the Lie generators  $f_\ell$  are determined by the equation of motion  $\dot{\mathcal{M}} = \mathcal{M} : -H :$  where  $H = H_2 + H_3 + H_4 + \cdots$  is the Hamiltonian expressed in terms of deviation variables and expanded in a homogeneous polynomial series. The deviation variable Hamiltonian  $H$  is determined in turn by the Hamiltonian  $K$ . In Cartesian coordinates with  $z$  taken as the independent variable, and in the absence of electric fields,  $K$  is given by the relation

$$K = -[p_t^2/c^2 - m^2 c^2 - (p_x - qA_x)^2 - (p_y - qA_y)^2]^{1/2} - qA_z.$$

Here  $\mathbf{A}$  is the magnetic vector potential. We therefore need a Taylor expansion for the vector potential components  $A_x$ ,  $A_y$ ,  $A_z$  in the deviation variables  $x$  and  $y$ . How can the coefficients of the Taylor expansion for the vector potential be determined from a knowledge of the magnetic field? In this paper we review the method we proposed in [3] and present an application to the High-Gradient quadrupoles in the interaction region of the LHC. The method uses information about the fields coming from either direct measurement or numerical computation done with a 3D electromagnetic

code. It is based on the calculation of Fourier integrals with suitable kernels derived from the Green function of the Laplace equation. Our approach is different from and more accurate than other methods based on numerical differentiation (see [4]). A pleasant feature is relative insensitivity to the presence of noise in the magnetic field data, which makes the method capable of providing accurate computations of high order terms in the desired Taylor expansion. An additional advantage is that it applies, with minor modifications, to both magnet data obtained by numerical computation and measured data found with spinning coils (see [3, 5] for more details on this aspect). A similar approach is also followed in [6].

## 2 DETERMINATION OF THE VECTOR POTENTIAL

In a current-free region the magnetic field  $\mathbf{B}$  can be described most simply in terms of a *scalar* potential  $\psi$  (with  $\mathbf{B} = \nabla\psi$ ) obeying the Laplace equation  $\nabla^2\psi = 0$ . In cylindrical coordinates the general solution to this equation (that is regular for small  $\rho$ ) has the expansion

$$\psi = \sum_{m=0}^{\infty} \int_{-\infty}^{\infty} dk e^{ikz} I_m(k\rho) \times [\hat{b}_m(k) \sin m\phi + \hat{a}_m(k) \cos m\phi], \quad (2)$$

where the functions  $\hat{a}_m(k)$  and  $\hat{b}_m(k)$  are arbitrary, and  $I_m$  is the modified Bessel function. This is a ‘‘cylindrical multipole’’ expansion, where  $m$  is related to the order of the multipole, and should not be confused with a spherical multipole expansion. The first term on the RHS of (2) describes a purely solenoidal field ( $m = 0$ ). The other terms in the series correspond to the dipole ( $m = 1$ ), quadrupole ( $m = 2$ ),  $\cdots$  components. For simplicity we will treat the terms with  $m \geq 2$ . The solenoidal term requires a separate, but analogous, treatment that entails no new complications. The dipole case is more complicated. In the sometimes restrictive case that the sagitta of the design orbit does not exceed the radius  $R$  introduced in Sec. 3, the methods of this paper also apply. However other methods are required if the sagitta is larger.

If  $\psi$  is given in the form (2), a suitable corresponding vector potential is easily found. Since there is gauge freedom, a possible convenient choice, in the absence of a solenoidal component, is to work in a gauge satisfying  $A_\phi = 0$ . Suppose  $\psi$  as given by (2) is rewritten in the form

$$\psi = \sum_{m=1}^{\infty} \psi_{m,s}(\rho, z) \sin m\phi + \psi_{m,c}(\rho, z) \cos m\phi \quad (3)$$

with

$$\psi_{m,s}(\rho, z) = \int_{-\infty}^{\infty} dk e^{ikz} I_m(k\rho) \hat{b}_m(k). \quad (4)$$

[ $\psi_{m,c}$  has the same form, with  $\hat{a}_m(k)$  replacing  $\hat{b}_m(k)$ .] Then it is easily verified that the remaining components of the vector potential are given by the relations

$$\begin{aligned} A_\rho &= \sum_{m=1}^{\infty} \frac{\cos(m\phi)}{m} \rho \frac{\partial}{\partial z} \psi_{m,s} - \frac{\sin(m\phi)}{m} \rho \frac{\partial}{\partial z} \psi_{m,c}, \\ A_z &= \sum_{m=1}^{\infty} -\frac{\cos(m\phi)}{m} \rho \frac{\partial}{\partial \rho} \psi_{m,s} + \frac{\sin(m\phi)}{m} \rho \frac{\partial}{\partial \rho} \psi_{m,c}. \end{aligned}$$

From the two equations above it is clear that finding Taylor expansions for the vector potential components  $A_x$ ,  $A_y$ , and  $A_z$  (what we need) is equivalent to finding Taylor expansions for  $\psi_{m,s}$  and  $\psi_{m,c}$  in the variable  $\rho$ . This is easily done by a two-step process: first, we expand the modified Bessel functions  $I_m(k\rho)$  appearing in (4) as Taylor series in the quantity  $(k\rho)$ . Doing so produces an expansion in powers of  $\rho$  with coefficients that involve integrations over various powers of  $k$ . Second, we observe that the powers of  $k$  can be replaced by multiple differentiation with respect to the variable  $z$ . The net results of these two steps are the relations ( $\alpha = c, s$ )

$$\psi_{m,\alpha}(\rho, z) = \sum_{\ell=0}^{\infty} (-1)^\ell \frac{m!}{2^{2\ell} \ell! (\ell + m)!} C_{m,\alpha}^{[2\ell]}(z) \rho^{2\ell+m}. \quad (5)$$

The index  $[2\ell]$  indicates the  $2\ell$  derivative with respect to the longitudinal variable  $z$ . The functions  $C_{m,\alpha}^{[0]}(z)$  are the generalized on-axis gradients. Note that the generalized gradients depend on the longitudinal variable  $z$ . For fields produced by long well-made magnets, however, the  $z$  dependence will be significant only at the ends.

We conclude that the dynamics of a charged particle passing through a region of space occupied by a magnetic field described by the scalar potential (2) is completely determined by a knowledge of the generalized on-axis gradient functions  $C_{m,\alpha}^{[0]}(z)$  and their derivatives. We will now describe two ways for computing the generalized gradients and their derivatives.

### 3 COMPUTATION OF GENERALIZED GRADIENTS FROM FIELD DATA

Suppose the radial component of the magnetic field  $B_\rho$  is known, either by measurement or computation, on the surface of some infinitely long cylinder of radius  $R$ . Moreover, suppose that the field is given in terms of an angular Fourier series,

$$B_\rho(\rho = R, \phi, z) = \sum_{m=1}^{\infty} B_m(R, z) \sin(m\phi) + A_m(R, z) \cos(m\phi).$$

It can be shown [5] that the generalized on-axis gradients appearing in the expansion coefficients for the scalar potential (8) can be written as

$$C_{m,s}^{[n]}(z) = \frac{i^n}{2^m m!} \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dk e^{ikz} \frac{k^{m+n-1}}{I'_m(kR)} \tilde{B}_m(R, k). \quad (6)$$

The expression for  $C_{m,c}^{[n]}(z)$  has  $\tilde{A}_m(R, k)$  replacing  $\tilde{B}_m(R, k)$ . Here  $\tilde{B}_m(R, k)$  and  $\tilde{A}_m(R, k)$  are the Fourier transforms of  $B_m(R, z)$  and  $A_m(R, z)$ , e.g.,

$$\tilde{B}_m(R, k) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} dz e^{-ikz} B_m(R, z). \quad (7)$$

In the case where the magnetic field is produced by an iron dominated magnet, and is therefore localized in space, the integrals (7) can be considered to have, in practice, finite limits of integration. With some care, an effective cut-off can also be found even if the fields extend to infinity since they fall off sufficiently rapidly at infinity. Also, since the generalized Bessel function  $I'_m(w)$  increases exponentially for large  $|w|$ , there is also, in effect, a cut-off in  $k$  for the integral (6) defining the generalized gradients.

### 4 NUMERICAL TEST AND INSENSITIVITY TO ERRORS

The method described in Section 3 has been implemented in the code MARYLIE 5.0 [2] as a user-defined routine. The routine reads from an external file the functions  $A_m(R, z)$  and  $B_m(R, z)$ , evaluated on a discrete set of points  $z_i$ . It then generates the corresponding transfer map by using the built-in routine GENMAP to integrate the map equation. Since MARYLIE 5.0 is a 5<sup>th</sup> order code, only the multipoles through  $m = 6$  need be considered.

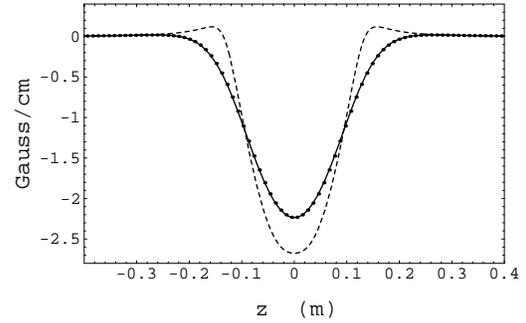


Figure 1: Plot of the scaled harmonic  $B_2(R, z)/R$  (dashed line) and the on-axis gradient  $2C_{2,s}^{[0]}(z)$  as calculated from surface data (dots) and analytically (solid line) for an ideal Lambertson quadrupole.

The integration algorithm of GENMAP is based on a 11<sup>th</sup> order multistep (Adams) method. Because the algorithm uses a fixed step size, one needs to provide values of the generalized gradients and their derivatives only at the predetermined locations in  $z$  required by GENMAP. We

emphasize that no interpolation of the generalized gradients is required by GENMAP.

In order to test both the routines and the method we treated the case of an ideal iron-free Lambertson quadrupole. The use of this case as an example has the virtue that the various  $C_{m,\alpha}^{[n]}(z)$  can also be determined analytically (see [5]). First we computed the surface data  $B_\rho(\rho = R, \phi, z)$  using the Biot-Savart law, and made an angular Fourier analysis to calculate the harmonics  $B_2(R, z)$  and  $B_6(R, z)$  for various values of  $z$ , (because of the symmetries involved only the harmonics of order  $m = 2, 6, 10, \dots$  are non-vanishing). We set the radius  $R$  of the cylinder to be at 75% of the magnet aperture  $r = 0.128$  m, with the length of the magnet being  $2r$ , and the step size  $\Delta z$  used in the calculation of the harmonics = 6.44 mm. Next, we used these data to reconstruct the multipole field expansion by applying our method and compared the results to those predicted by the analytic formulas.

Results are shown in Figs. 1 and 2. In Fig. 1 the dashed line is the function  $B_2(R, z)/R$  as calculated numerically by using the Biot-Savart law. The solid line represents the on-axis gradient, which is equal to  $2C_{2,s}^{[0]}(z)$ , as calculated analytically, while the dots represent the same function as calculated from the surface data. The deviation between  $B_2(R, z)/R$  and  $2C_{2,s}^{[0]}(z)$  is due to terms in the multipole expansion containing derivatives of  $2C_{2,s}^{[0]}(z)$ . These terms are the so called pseudo-multipoles. This can be seen by writing the multipole expansion for  $B_\rho$  through 6<sup>th</sup> order in  $\rho$ :

$$B_\rho = \left( 2C_{2,s}^{[0]}\rho - \frac{1}{3}C_{2,s}^{[2]}\rho^3 + \frac{1}{64}C_{2,s}^{[4]}\rho^5 \right) \sin 2\phi + 6C_{6,s}^{[0]}\rho^5 \sin 6\phi. \quad (8)$$

In Fig. 2, as an indication of the reliability of the method, we report the 8th derivative of the generalized gradient (needed for a 9<sup>th</sup> order code) calculated from the surface data (dots) compared to the analytical profile (solid line). The accuracy in the calculation of the generalized gradient is mirrored by the accuracy with which the elements of the transfer map can be determined. For this case we found relative errors ranging from  $10^{-6}$  for  $\mathcal{R}_2$  to  $10^{-4}$  for  $f_5$  and  $f_6$ .

In practical situations magnetic field data, whether coming from measurements or from numerical computations, are unavoidably affected by errors. What effect do these errors have on the determination of the generalized on-axis gradients and their derivatives? The effect of these errors is relatively mild. This relative insensitivity to errors arises from a basic property of solutions to Laplace's equation: the value of  $\psi$  at some interior point is an appropriately weighted average of its values over any surrounding boundary. Consequently,  $\psi$  is smoother in the interior of a region than it may be on a boundary of this region. Correspondingly, errors in boundary values are averaged. A way to see the smoothing mechanism in action is to look at (6) and observe the presence of the kernel  $1/I'_m(kR)$  acting as

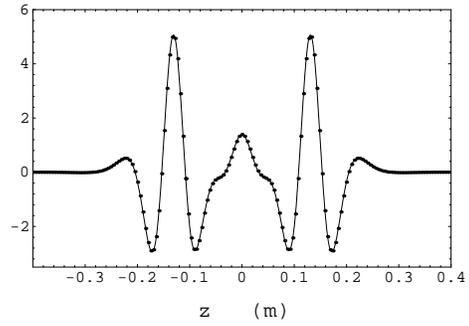


Figure 2: The function  $C_{2,s}^{[8]}(z)$  (in units of  $10^{-5}$  Gauss/cm<sup>9</sup>) as calculated from surface data (dots) and analytically (solid line) for an ideal Lambertson quadrupole.

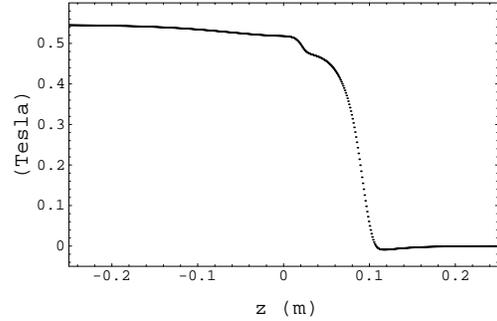


Figure 3: Harmonic  $B_2(R, z)$  (quadrupole field component) for the Return End;  $R=3$  cm.

a high frequency filter  $[I'_m(kR)]$  grows exponentially with  $kR$ . Incidentally, note that for the purpose of noise reduction it is desirable to choose  $R$  as large as possible.

We tested noise insensitivity by considering a simple noise model for which surface fields had values given by

$$B_2^{rnd}(R, z_i) = B_2(R, z_i)[1 + \epsilon_2(z_i)], \quad (9)$$

and similarly for  $B_6^{rnd}(R, z_i)$ , where  $\epsilon_2(z_i)$  is a random variable uniformly distributed in the interval  $[-\epsilon/2, \epsilon/2]$ , and  $B_2(R, z_i)$  is the same as before. Then we calculated the generalized gradients and transfer map using the noisy data. For  $\epsilon = 10^{-2}$  we found relative errors, ranging from  $5 \times 10^{-4}$  for  $\mathcal{R}_2$  to  $10^{-2}$  for  $f_5$  and  $f_6$ , which are at worst on the order of the noise itself.

## 5 APPLICATION TO LHC HIGH-GRADIENT QUADRUPOLES

In this final Section we describe an application of our method to the calculation of the transfer maps for the High-Gradient (HG) superconducting quadrupoles located at the interaction region of LHC (see [8] and references therein). Our purpose here is two-fold: We want to (i) provide an illustration of the method in a case of physical interest and (ii) test the adequacy of the way magnets are usually modeled in tracking studies against our more realistic modeling.

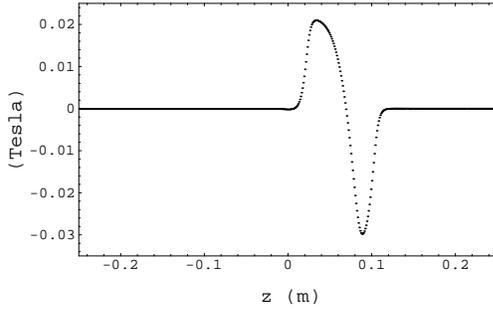


Figure 4: Harmonic  $B_6(R, z)$  (duodecapole field component) for the Return End;  $R=3$  cm.

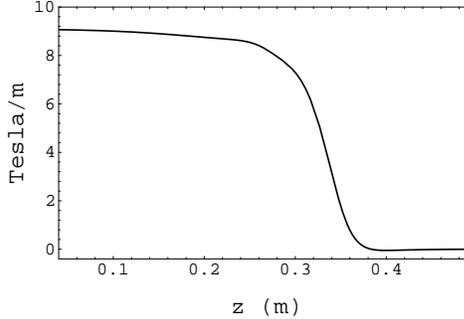


Figure 5: Generalized gradient  $C_{2,s}(z)$  for the Return End. ( $z = 25$  cm in this picture corresponds to  $z = 0$  in Figs. 3 and 4.)

The HG quadrupoles provide the focusing necessary to compress the beam at each Interaction Points (IP) so as to enhance luminosity. This can be done only by allowing the beam to expand considerably in the regions adjacent to the IP's where the HG quadrupoles are located. As a result, the beam is particularly sensitive to the aberrations associated with the HG quadrupoles.

All the HG quadrupoles have same gradient (200.415 T/m) and are identical except for their lengths (details on the LHC lattice can be found in [7]).

We first carried out the harmonic analysis of the magnetic field data provided by [9] for the two ends of the magnets. Because of differing mechanical constraints the geometry of the two ends and the resulting fields are different. The two ends are referred to as 'Lead End' and 'Return End'. The magnetic field was calculated using a code that integrates the Biot-Savart law and approximates the effects of the iron yoke in the limit of infinite permeability.

The  $2^{nd}$  and the  $6^{th}$  harmonics for the Return End (the other harmonics of order lower than 6 are negligible) are shown in Figs. 3 and 4. From these harmonics one can recover the generalized gradients by applying the formulas reported in Section 3 and implemented in MARYLIE. In Fig. 5 we show the generalized gradient  $C_{2,s}(z)$ . In the pictures the fields are scaled by an arbitrary factor. Notice that the profile of  $C_{2,s}(z)$  is smoother than the profile of  $B_2(R, z)$ . This is due to the contribution of the higher order

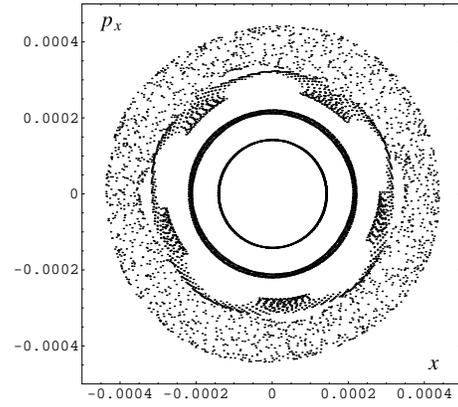


Figure 6: Projection of the Poincaré surface of section on the horizontal plane at the IP (normalized coordinates in units of  $m^{\frac{1}{2}}$ ). Map calculated from the magnetic field data.

derivatives of  $C_{2,s}(z)$  to the field expansion [see Eq. (8)]. The integrated duodecapole component  $S_d$  of the fields at the two ends, defined by  $S_d = \int_{END} C_{6,s}(z) dz$  was respectively  $5.134 \times 10^3$  and  $8.148 \times 10^4$  T/m<sup>4</sup> for the Return and Lead Ends.

Having computed the generalized gradients we calculated the transfer maps associated with the end regions and combined them with the transfer map for the body of the magnet to obtain the transfer map of the full quadrupole. By combining the transfer maps for the various elements in the IP region and the rest of the machine we constructed the one-turn map. This map we used for tracking studies.

Since the focus of our study was on the field modeling of the HG quadrupoles we considered a simplified version of the LHC lattice in which all the lattice elements, with the exception of the HG quadrupoles, were described in the linear approximation. Thus the only sources of nonlinearities (besides the purely geometric terms depending on the canonical momenta) were the intrinsic aberrations carried by the fringe fields [i.e. the 'pseudo-multipoles' or more specifically the aberrations generated by the terms  $C_{2,s}^{[n]}$  with  $n > 0$  in Eq. (8)] and the duodecapole field components at the HG magnet ends.

The result of a tracking study using this model is presented in Fig. 6, which shows the projection on the  $x$ - $p_x$  plane of the Poincaré surface of section at one IP. The orbits of four particles are displayed over  $10^5$  turns, with the outer orbit being on the edge of the dynamic aperture. Tracking was done using the code CTRACK [10]. Normalized coordinates are used (i.e.  $x_{norm} = x_{phys} / \sqrt{\beta_x}$  with  $\beta_x = 0.5$  m, etc.). This picture should be compared to a similar projection of the Poincaré surface of section obtained with a model of the HG quadrupoles in which no duodecapole field components are present (see Fig. 7). One can observe the dynamic aperture increases by a factor of 5, suggesting that in the present design the duodecapole nonlinearities dominate over the intrinsic aberrations carried by

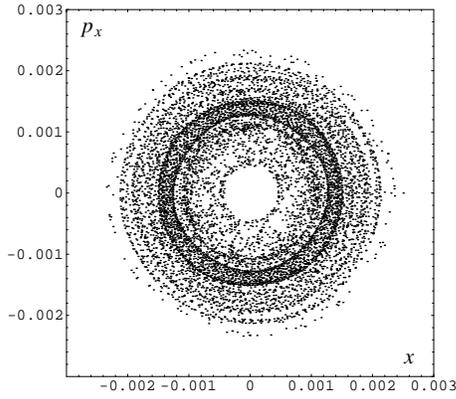


Figure 7: Projection of the Poincaré surface of section on the horizontal plane at the IP (normalized coordinates in units of  $m^{\frac{1}{2}}$ ). Map calculated from the magnetic field data with the duodecipoles turned off.

the quadrupole field components. And last Fig. 8 was obtained by tracking with a ‘Standard Model’ for quadrupole magnets. In this model the action of the duodecapole field components is approximated by thin kicks with strength  $S_d$  and the aberrations due to the quadrupole field components are evaluated though third order in the hard-edge limit. One can observe that the phase space portraits in the Figs. 6 and 8 are very similar. Since the nonlinearities are mostly due to the duodecapole field components the implication is that a thin kick approximation to model the duodecapole terms seems to be adequate. This is confirmed by an inspection of the Lie generators of the transfer map for an individual HG quadrupole in the two different models. In the Table below we show two such Lie generators. The generator  $x^6$  is determined mostly by the duodecapole field components. The two values are within a few percents. On the other hand the generator  $x^2 p_x$ , which measures the third order aberrations associated with the quadrupole field components, differs in the two cases by a factor 7. This shows that a more realistic modelling of the magnet fields may be necessary if the intrinsic quadrupole aberrations are found to be important.

Lie generator	Realistic Model	‘Standard Model’
$x^2 p_x$	0.54011D-04	0.72959D-03
$x^6$	-0.14337D+01	-0.13358D+01

## 6 ACKNOWLEDGEMENTS

We have benefitted greatly from many conversations with P. Walstrom, the reading of some of his internal Technical Notes, and the use of his Fourier integration and Bessel function routines. We are also grateful to T. Mottershead, F. Neri, and P. Walstrom for their many contributions to MARYLIE 5.0. Finally, we thank G. Sabbi for providing us with magnetic field data for LHC quadrupoles. This

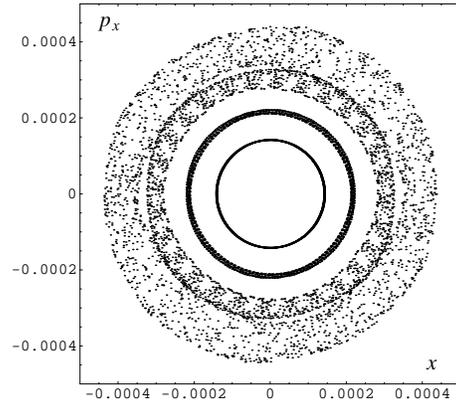


Figure 8: ‘Standard model’ for the HG quads: The quadrupole field component aberrations are treated in the hard edge limit and the systematic duodecipoles in the kick approximation. (Normalized coordinates in units of  $m^{\frac{1}{2}}$ ). To be compared with Fig. 6.

work was supported by US Department of Energy Grant DEFG0296ER40949.

## 7 REFERENCES

- [1] A. Dragt, Lie Methods for Nonlinear Dynamics with Applications to Accelerator Physics, University of Maryland Physics Department Report (1998).
- [2] A. Dragt, F. Neri, G. Rangarajan, D. Douglas, L. Healy, and R. Ryne, Lie Algebraic Treatment of Linear and Nonlinear Beam Dynamics, Ann. Rev. Nucl. Part. Sci. **38** (1988) pp. 455-96.
- [3] M. Venturini, A. Dragt, Accurate Computation of Transfer Maps from Magnetic Field Data, to be published in NIM-A.
- [4] S. Caspi, M. Helm, and L.J. Laslett, The Use of Harmonics in 3-D Magnetic Fields, IEEE Trans. on Magnetics, vol.30, no. 4, (1994).
- [5] M. Venturini, Lie Methods, Exact Map Computation and the Problem of Dispersion in Space Charge Dominated Beams, Ph. D. Dissertation (1998), Physics Department, University of Maryland.
- [6] R. Trines, Modelling the Fringe Fields of a Multipole Device, Master Thesis (1998), Eindhoven University of Technology.
- [7] The LHC lattice can be found on the Web at ‘<http://www-ap.fnl.gov/lhc/database/lattices>’.
- [8] G. Sabbi, HGQSO3 End Field Analysis, Fermilab Report TD-98-010 (1998).
- [9] G. Sabbi, Private Communication.
- [10] D. Abell, Analytic Properties and Cremona Approximation of Transfer Maps for Hamiltonian Systems, Ph. D Dissertation (1995), Physics Department, University of Maryland.

# Functional Dependence, Broad-Band Fitting, and Ancillary Conditions

David C. Carey

Fermi National Accelerator Laboratory\*  
P.O. Box 500, Batavia, Illinois 60510

## Abstract

The ability to make multiple passes through a charged particle optical system, as a single step in a mathematical procedure, opens up new computational capabilities. At the simplest level, the functional dependence of any transfer matrix of any order, or any beam phase-space parameter can be plotted as a function of any other parameter used to describe the optical configuration. Secondly, broad-band fitting can be done on aberrations, where all orders are considered simultaneously according to their importance on the final phase-space beam distribution. Finally, ancillary conditions may be imposed in the calculation of functional dependences. For example, the dependence of a matrix element on a beam line parameter may be calculated and plotted, subject to constraints imposed on other matrix elements. The computer program TRANSPORT now has these capabilities. Additional examples will be given.

## 1 INTRODUCTION

Functional dependence, broad-band fitting, and ancillary conditions are all useful concepts in a charged particle optical system. The computer program TRANSPORT [1] now has these three capabilities. In this section and the following we describe each of these three items, what they represent mathematically, and how to implement them with TRANSPORT.

The functional dependence can involve any of the parameters used to describe the physical beam line and any of the transfer matrix elements which can be calculated. One parameter can be stepped over an interval. As it is stepped, the values of the transfer- and beam- matrix elements will vary systematically. Plots can be made of any matrix element vs the stepped parameter, or of one matrix element vs another.

Ancillary conditions can be imposed in the form of constraints on the matrix elements. With constraints, a fitting process is carried out for each value of the stepped parameter. The constraint will now make the stepped parameter affect the value of any parameters which are varied or calculated from parameters which are varied. Now there will be functional dependence between parameters, and plots can be made of one parameter vs another. The plots will be made with the values of the fitted parameters at the end of the fitting process.

In broad-band fitting, there is no longer a separate fitting procedure for each value of the stepped parameter. Rather, the entire set of values of the stepped parameter are included in a single fitting procedure. Broad-band fitting is useful for eliminating aberrations. In broad-band fitting, aberrations

are not separated into orders, but minimized as they actually occur in a charged particle optical system.

In the following sections, we will give examples of the three capabilities: functional dependence, broad-band fitting, and ancillary conditions. Functional dependence and ancillary conditions will be treated together in the next section since they are strongly related. Broad-band fitting will be treated in a separate section.

## 2 FUNCTIONAL DEPENDENCE AND ANCILLARY CONDITIONS

Consider a symmetric quadrupole triplet. The pole-tip fields are varied so as to obtain a first-order focus in both transverse planes. The triplet is placed so that its longitudinal midpoint is two meters upstream of the longitudinal midpoint of the beam line.

Now the triplet is treated as a rigid body and slid forward and backward. The triplet continues to be symmetric. At each location the pole-tip magnetic fields are adjusted to give foci in the two transverse planes at the end of the beam line.

We define a parameter  $DS$  which indicates the longitudinal displacement of the triplet from the point two meters before the midpoint of the beamline. If  $DS$  is decreased, then the triplet moves toward the beginning of the beam line. If  $DS$  is increased, then the triplet moves toward the end.

Figure (1) is a plot of the two pole-tip magnetic fields of the symmetric triplet vs the longitudinal displacement  $DS$ . Note that the fields are minimized when the parameter  $DS$  is 2.0 meters. At that point the configuration is longitudinally symmetric. The triplet is equidistant from the beginning and the end of the beam line.

The plot shown is made with TRANSPORT [1] and TOPDRAWER [2]. The entire process of making the plot is automated. Starting with a data file for TRANSPORT and a skeleton data file (all instructions except the data), the plots are produced by invoking a single procedure.

The functional dependence of the pole-tip fields is shown clearly. The ancillary conditions are embodied in the requirement that there be a focus in both transverse planes at the end of the beam line.

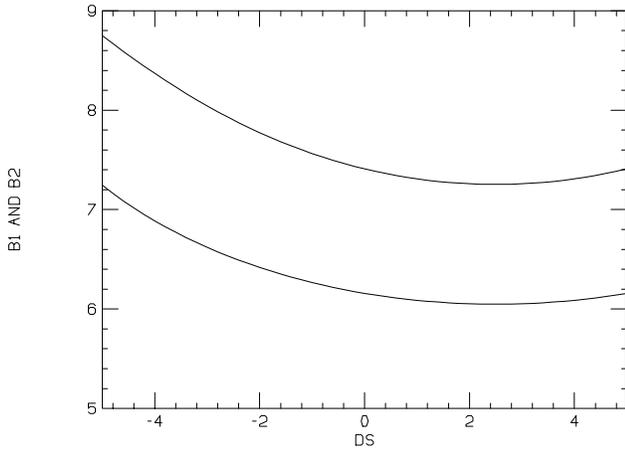


Figure (1): The dependence of the two pole-tip magnetic fields of a symmetric triplet on the longitudinal position of the triplet.

### 3 OFF-MOMENTUM TRANSFER MATRICES

In the theory of charged-particle optics, the action of an optical system on a charged particle is represented by a matrix expansion. The final coordinates  $X_i(1)$  are given in terms of the initial coordinates  $X_i(0)$  by the expression [3] [4]:

$$\begin{aligned} X_i(1) &= \sum_j R_{ij} X_j(0) \\ &+ \sum_{jk} T_{ijk} X_j(0) X_k(0) \\ &+ \sum_{jkl} U_{ijkl} X_j(0) X_k(0) X_l(0) , \end{aligned} \quad (1)$$

The three quantities  $R$ ,  $T$ , and  $U$  represent respectively the first-, second-, and third-order matrices of the Taylor-series expansion. The six beam coordinates are taken to be  $(x, x', y, y', \ell, \delta)$ . Here  $\delta$  is defined as  $\Delta p/p_0$ , the fractional deviation from the central momentum. When we speak of the displacement of a trajectory, we include a displacement in  $\delta$ , even though the term “displacement” sounds like a geometric consideration and  $\delta$  is a kinematic quantity. A trajectory which follows the axis of a quadrupole channel can still have a displacement in  $\delta$  if the trajectory momentum is different from the reference momentum.

Let us now shift our reference to a trajectory which is displaced from the original reference trajectory. The new reference trajectory is denoted as  $X_r$ , and the difference between an arbitrary trajectory and the new reference trajectory is given by  $\Delta X$ . In terms of these new quantities, equation (1) can now be rewritten as:

$$\begin{aligned} X_{1r} + \Delta X_1 &= X_{1s} + R(X_{or} + \Delta X_o) \\ &+ T(X_{or} + \Delta X_o)(X_{or} + \Delta X_o) \\ &+ U(X_{or} \Delta X_o)(X_{or} + \Delta X_o)(X_{or} + \Delta X_o) \end{aligned} \quad (2)$$

Subtracting the equation for the reference trajectory, we derive

$$\begin{aligned} \Delta X_1 &= R \Delta X_o + T(2X_{or} \Delta X_o + \Delta X_o \Delta X_o) \\ &+ U(3X_{or} X_{or} \Delta X_o + 3X_o \Delta X_o \Delta X_o \\ &\quad + \Delta X_o \Delta X_o \Delta X_o) \\ &= (R + 2TX_{or} + 3UX_{or} X_{or}) \Delta X_o \\ &+ (T + 3UX_{or}) \Delta X_o \Delta X_o + U \Delta X_o \Delta X_o \Delta X_o \end{aligned} \quad (3)$$

From equation (3), we can define new first- and second-order transfer matrices by

$$\begin{aligned} R^* &= R + 2TX_{or} + 3UX_{or} X_{or} \\ T^* &= T + 3UX_{or} \end{aligned} \quad (4)$$

These redefined matrices for each element can then be accumulated to produce transfer matrices for the entire magnetic optical system. The transformation of a particle trajectory through the system can now be represented by a transformation similar in appearance to equation (1).

$$X_1 = X_{1r} + R(t)X_o + T(t)X_o X_o + U(t)X_o X_o X_o \quad (5)$$

Here the matrices  $R(t)$ ,  $T(t)$ , and  $U(t)$  are calculated as products of the matrices  $R^*$ ,  $T^*$ , and  $U$ , as defined in equation (4). They are expressed relative to the transformed original reference trajectory  $X_{1r}$ . The original reference trajectory is transformed through the system using equation (1) on an element-by-element basis.

The off-axis expansion can be used to explore chromatic effects. If a nonzero  $\delta$  is specified, the first-order transfer matrix will be with respect to the off-momentum centroid. If there is dispersion in the system, the centroid will, at some point, be displaced from the reference trajectory. The result from an off-axis expansion can differ substantially from that of an on-axis calculation. The effect of many orders higher than second or even third can be seen.

Let us consider the example of a very long focusing system, with many intermediate foci. The value of the  $R_{12}$  matrix element at the end of such a system is shown in figure (2). First, we use the traditional procedure and accumulate the transfer matrices about the original reference trajectory. Then, at the end, we use equation (4) to define a new momentum-dependent first-order transfer matrix for the entire system. The value of the  $R_{12}$  matrix element is given by the straight line (labeled simply “2nd order”), passing through the origin. The inclusion of third order makes no visible difference.

For an exact calculation, with a ray tracing program, the behavior is quite different. For a slight deviation from the reference momentum, the value of  $R_{12}$  will, as before, grow linearly with the value of  $\delta$ . As the momentum deviation continues to increase, one of the intermediate foci will move downstream and the system will once again be focusing.

We can also calculate the off-momentum first-order transfer matrix for each element using equation (4). If we

then accumulate these off-momentum  $R$  matrices over the entire beam line, then we get the two curves shown in figure (2). Both curves are labelled “SBA”, which means shift before accumulating. The curve labelled “2nd order” includes only the second-order terms in equation (4). The curve labelled “3rd order” also includes the third-order term in equation (4). Since the third-order curve is indistinguishable from the exact calculation, no higher orders in the expansion are necessary.

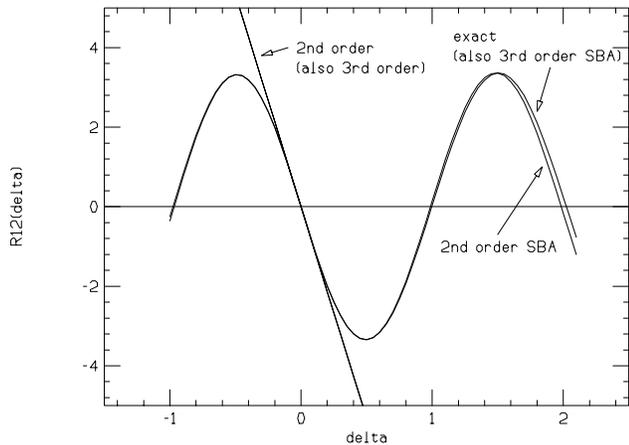


Figure (2): The magnitude of the sinelike trajectory, as a function of relative momentum deviation  $\delta$ , at the end point of a long beam line. The various meanings of the curves in the figure are explained in the text.

#### 4 BROAD-BAND FITTING

Broad-band fitting resembles the combination of functional dependence with ancillary conditions in a very simple way. Both involve fitting, and both involve stepping of a parameter over an interval. With functional dependence with ancillary conditions, the fitting loop is inside the stepping. In other words, there is a separate fitting procedure for each value of the stepped parameter.

With broad-band fitting, there is a single fitting procedure which includes all the values of the stepped parameter. The chi-squared for the the configuration is equal to the sum of the individual chi-squareds for the different values of the stepped parameter. In the example given above, the stepped parameter is the relative momentum deviation  $\delta$ .

A broad-band fitting procedure then imposes a constraint over an interval in  $\delta$ . The example given is a constraint for a point-to-point focus. The off-momentum transfer matrix, according to equation (4), contains contributions from the higher-order matrix elements. Therefore a broad-band constraint on the first-order matrix elements  $R_{12}$  and  $R_{34}$  is simultaneously a constraint on all higher-order chromatic matrix elements affecting the focusing.

To effect the broad-banded fitting in TRANSPORT, one need do only three things. First, one must indicate that a first-order fit is to be made on the matrix elements  $R_{12}$  and  $R_{34}$ . Secondly, one must specify that the parameter  $\delta$  is to be stepped, and also give the limits of the interval and the

step size. Finally, one must include a line in the data which has on it the word BROAD, indicating that the fit is to be broad-banded.

TRANSPORT can then optimize the chromatic characteristics of the beam line by adjusting the strength of sextupoles, octupoles, and any other optical element influencing higher-order terms. The optimization will be imposed uniformly across the entire momentum interval specified, instead of favoring the origin, as one does when one fits the different orders separately.

#### 5 REFERENCES

- [1] David C. Carey, Karl L. Brown, and Frank Rothacker, *TRANSPORT, A Computer Program for Designing Charged Particle Beam Transport Systems*, SLAC Report No. 95-462.
- [2] R.B. Chaffee, “Introduction to Top Drawer”, SLAC Computation Group Report CGTM No. 189.
- [3] David C. Carey, *The Optics of Charged Particle Beams*, Harwood Academic Publishers, New York, 1987.
- [4] Karl L. Brown, *A First- and Second-Order Matrix Theory for the Design of Beam Transport Systems and Charged Particle Spectrometers*, SLAC Report No. 75, Rev. 4, 1982.

\*Operated by Universities Research Association, Inc., under contract with the United States Department of Energy.

# TRANSIENT MAGNETIC FIELD ANALYSIS FOR THE LOCALIZATION OF ELECTRICAL FAULTS IN SUPERCONDUCTING COLLARED COILS

Piotr Komorowski\*, Davide Tommasini  
CERN, LHC Division, CH-1211 Geneva 23

\*Faculty of Physics, Warsaw University of Technology, Warsaw, Poland

## Abstract

The follow-up of the construction of superconducting magnets for accelerators requires setting up of powerful diagnostic tools to detect weak electrical points in the superconducting coils at various stages of their fabrication. In particular some electrical short circuits well detectable after collaring of the magnet often disappear after the coil is uncollared for repair; therefore it is preferable to localize this kind of electrical faults before disassembling the magnet.

An R&D work on detection and localization of inter-turn short circuits is being carried out at CERN in view of the series production of the LHC magnets. The diagnostic methods under study include pulse propagation, time domain reflectometry and transient magnetic field analysis. In this paper special emphasis is put on the analysis of the magnetic field distortions created by the short circuits during a pulsed discharge. A model of LHC dipole allowing the simulation of different fault conditions in the coils has been implemented in ROXIE (static case) and in OPERA-2D (transient case). The model has been verified experimentally on a dedicated short dipole magnet equipped with micro-switches to trigger short circuits in different areas of the coils.

## 1. INTRODUCTION

The pattern of the magnetic field in modern superconducting accelerator magnets is mainly determined by the geometry of the coils. In an ideal case a current of the type  $I(\varphi) = I_0 \cos(\varphi)$  distributed around the aperture produces a pure dipole field. For practical reasons multi-layer coils graded with longitudinal wedge-shaped spacers inserted between the conductor blocks approximate this condition (Fig. 1). The LHC dipole coils are composed of two layers connected with each other to form a pole. This paper is focussed on the study of the single aperture superconducting short models for the LHC in view of applying the results of the investigation to the 15-m long double aperture magnets.

Since 1989 a test and evaluation program of short superconducting dipole magnets for the LHC is under way at CERN. It was intensified in 1995 when a production of a new series of 1-m long dipole magnets was launched [1]. The design is based on a dismantlable structure allowing easy implementation of variants (Fig.2). The coils are confined in a fixed volume by the collars with a pre-compression sufficient to counteract the electrodynamic forces during magnet excitation. The outer layer of the coils is separated from the collars by 0.5

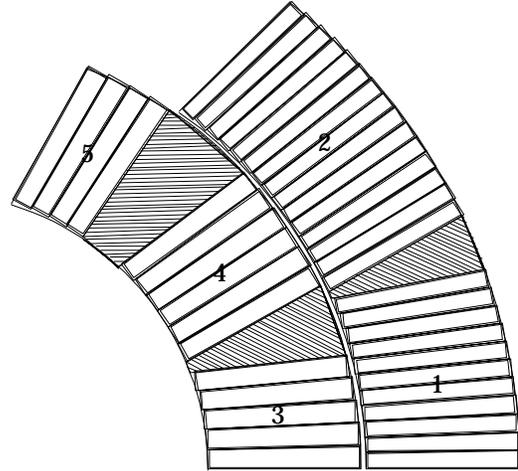


Fig. 1. Quadrant cross section of a 5-block geometry coil; 2 conductor blocks in the outer and 3 in the inner layer

mm of ground insulation and by a 0.7 mm thick austenitic steel protection sheet called collaring shoe. A ferromagnetic yoke is clamped around the collared assembly by an external bolted stainless steel cylinder. The yoke enhances the magnitude of the magnetic field induction inside magnet's aperture and prevents the collars from expanding when submitted to the Lorentz forces exerted on the coils.

The superconducting cable is insulated by one layer of polyimide film followed by a layer of adhesive tape. The external adhesive bonds adjacent turns of the coil. The effective thickness of the cable's insulation is about

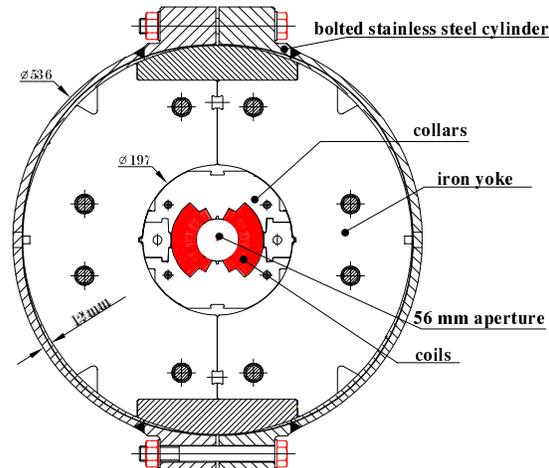


Fig. 2. Cross section of a single aperture dipole magnet

125  $\mu\text{m}$ ; i.e. two adjacent turns are separated by about 250  $\mu\text{m}$  of insulation. This dielectric barrier can be damaged during one of the magnet assembly phases, resulting in an inter-turn short circuit. The change in the distribution of the current in the coils of the magnet due to the fault produces a distortion of the magnetic field. This effect is used here for the longitudinal localization of the short circuits by means of transient magnetic field analysis.

## 2. INTER-TURN SHORT CIRCUITS

Often inter-turn short circuits are well detectable when the magnet is collared. The presence of the fault in the magnet can be diagnosed by means of pulse propagation methods at ambient temperature. A voltage impulse is produced between the terminals of the magnet

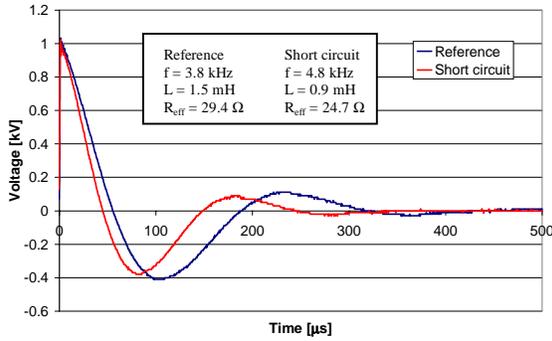


Fig. 3. Discharge curves at 1 kV

by a discharge generator. The coil's response in form of voltage and current oscillations is then registered and analyzed in terms of pseudo period of the oscillations, inductance and effective resistance of the coils. Fig.3 shows the discharge curves acquired on the collared assembly during a test at 1 kV. An equivalent electrical circuit of the magnet under test can be represented as an RLC distributed line excited by a charged capacitor. The oscillations produced during the discharge are gradually damped down by the resistive components of the line, which correspond to the different dissipation processes in the coils and in the surrounding magnet structure.

In case of an inter-turn short circuit, the current is redirected to the neighboring cable turn through the short circuit path. Energy is dissipated additionally at the point of fault. For the propagating impulse the coil appears to be shorter by a length of a cable turn affected by the short circuit. The value of the magnet's inductance decreases, whereas the damping factor and the frequency of the oscillations are increased. The existence of the electrical fault can be thus verified, nevertheless the method does not allow to localize the position of the short circuit. Traditional localization procedures based on resistance and voltage ratio measurements require dismounting of the collars. Often after the collars are removed and the internal prestress in the coils released the short circuit disappears. Time consuming methods involving local controlled pressure increase over small sections of the coils must be employed thereafter to reestablish and localize the short circuit.

## 3. MAGNET MODELING

### 3.1 Multipole expansion of the magnetic field

The quality of the magnetic field is analyzed in terms of multipole field expansion. According to the design project of the LHC [2], the complex magnetic field induction is expanded as follows:

$$B_y + iB_x = \sum_{n=1}^{\infty} (B_n + iA_n) z^{n-1} \quad (1)$$

Coefficients  $B_n$  and  $A_n$  indicate normal and skew field multipoles, respectively. In terms of radial and angular field components expansion (1) is equivalent to:

$$B_r(r, \theta) = \sum_{n=1}^{\infty} r^{n-1} (B_n \sin n\theta + A_n \cos n\theta) \quad (2)$$

$$B_\theta(r, \theta) = \sum_{n=1}^{\infty} r^{n-1} (B_n \cos n\theta - A_n \sin n\theta)$$

By introducing a reference magnetic field, equal to the main dipole field  $B_1$ , and the normal and skew multipole coefficients relative to the main field at a reference radius  $r_0$ ,

$$a_n = \frac{A_n r^{n-1}}{B_1} \Big|_{r=r_0} \quad b_n = \frac{B_n r^{n-1}}{B_1} \Big|_{r=r_0} \quad (3)$$

the radial and angular components of the field can be expressed as:

$$B_r(r, \theta) = B_1 \sum_{n=1}^{\infty} \left( \frac{r}{r_0} \right)^{n-1} (b_n \sin n\theta + a_n \cos n\theta) \quad (4)$$

$$B_\theta(r, \theta) = B_1 \sum_{n=1}^{\infty} \left( \frac{r}{r_0} \right)^{n-1} (b_n \cos n\theta - a_n \sin n\theta)$$

The main normal coefficient of the multipole expansion ( $b_1$ ) is thus always normalized to 1. Higher harmonic coefficients are given in units of  $10^{-4}$  relative to the dipole  $B_1$ .

### 3.2 Reference magnet models

Both static and transient models of the 5-block collared dipole magnet without iron yoke were developed and analyzed using two different CAD programs. In the static model the conductors are excited with  $I = 20$  A direct current. Since the collars are non-magnetic the calculation of the magnetic field harmonics in the static case is based uniquely on the geometry of the coils and the distribution of the current in the conductors.

Transient model simulates the behavior of the magnet during a 1 kV pulsed discharge in which the peak current reaches a value of  $i_{\max} = 20$  A. In this case the frequency of the oscillations of the current decaying in the coils, the skin and the eddy current effects in the conductors, copper wedges, collaring shoe and the collars are taken into account during field computations.

### 3.2.1 Static case

The static model was created with The Routine for the Optimization of Magnet X-Sections, Inverse Problem Solving and End Region Design (ROXIE) program developed at CERN [3]. Data concerning the characteristics of the superconducting cable for the external and internal layers of the coils are taken directly from the ROXIE database. The program features optimization of the coil geometry as a function of the desired magnetic field pattern. Advanced conductor positioning options, like the alignment of the cables on the inner or outer radius of the given layer, are also available. The conductors are subdivided into the regions corresponding to the number of strands in the cables, i.e. 2 rows of 18 elements for the external layer cable, and 2 rows of 14 elements for the internal one. The current line representing the strand is located in the center of each region. Magnetic field resulting from the coils is calculated directly from the Biot-Savart law (Fig. 4.)

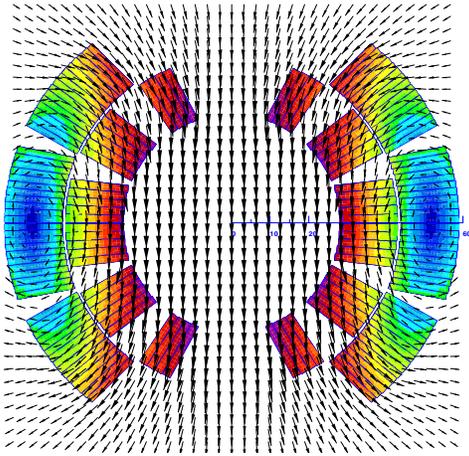


Fig. 4. Magnetic vector field computed with ROXIE

Two cases are considered below. The case with the current grading takes into account the keystone (trapezoidal) form of the cable's transversal cross section. As a consequence of this shape the strands near the inner perimeter of the cable are closer to each other than the ones near the outer edge. This results in grading of the current density throughout the conductor. The case without grading assumes uniform current density distribution in the cable. The outcome of the harmonic analysis of the radial field component at the reference radius of  $r_0 = 10$  mm is given in Table 1.

Table 1. ROXIE harmonic analysis of the magnetic field

Current grading	$B_1$ [ $\cdot 10^{-4}$ T]	$b_3$	$b_5$	$b_7$
On	-122.8	-0.96	0.032	0.008
Off	-122.2	-0.52	0.017	0.013

Due to the dipole symmetry of the coils neither skew multipole terms nor even-ordered normal multipoles are present in the field expansion. Grading of the current gives a difference of 0.44 units for the sextupole in comparison to the uniform current density case.

### 3.2.2 Transient case

The geometry of the coils defined in ROXIE served as a reference for the transient model implemented in OPERA-2D [4]. In addition to the conductor blocks the model includes the copper wedges, the collaring shoe, and the collars (Fig. 5). OPERA-2D uses FEM approach to solve the electromagnetic problem and calculate the redistribution of the currents due to the eddy current effects.

The effective conductivity of the collars was adjusted in the model in order to fine-tune the computational

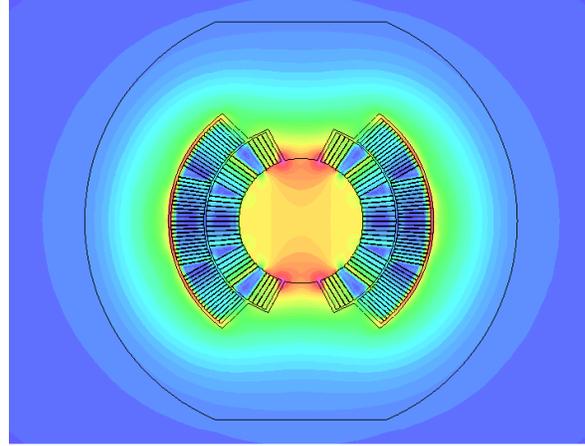


Fig. 5. OPERA-2D transient model

results to the experimental data acquired by measuring the peak field induction inside magnet's aperture and around the collars. For setting the correct value of the conductivity of the cables an additional test was carried out to evaluate the magnitude of the eddy current effects. Separate layers of the coils were excited one at a time to analyze the screening effects. Matching between the model and the experiment was achieved by assigning to the conductors the conductivity values measured with 1 A direct current, i.e.  $3.63 \cdot 10^7$  S/m for the inner layer cable, and  $3.23 \cdot 10^7$  S/m for the outer layer cable.

Relative multipole coefficients obtained during harmonic analysis of the magnetic field are of the order of  $10^{-4}$  and their computation requires high accuracy. The validity of the results relies on the precise definition of the coil geometry and correct implementation of the current density distribution into the model. Therefore the optimized OPERA-2D model was verified in the static case to check the convergence of the results with the ROXIE reference model.

Table 2. Comparison of static models at  $r_0 = 10$  mm

Static Case	$B_1$ [ $\cdot 10^{-4}$ T]	$b_3$	$b_5$	$b_7$
ROXIE	-122.2	-0.52	0.017	0.013
OPERA-2D	-119.3	-0.55	0.020	0.013

Table 2 shows that the model implemented in OPERA-2D and analyzed in the static case corresponds to the ROXIE analysis without current grading. The results of the harmonic analysis for the transient case are listed in Table 3.

Table 3. OPERA-2D transient field analysis

Transient Case	$B_1$ [ $\cdot 10^{-4}$ T]	$b_3$	$b_5$	$b_7$
10 mm	-74	-201	20.1	0.04

The effect of the eddy currents induced in the transient case deteriorates significantly the quality of the field and attenuates the main dipole component in comparison to the static case.

#### 4. SHORT CIRCUIT INDUCED FIELD DISTORTIONS

In case of short circuit the current follows a path different from the nominal one. Fig. 6 depicts schematically an inter-turn short circuit occurring between turns 30 and 31 of the magnet. The current is redirected at the point of fault through the short circuit path. In the cross-section preceding the fault (X-section 1) the current flows in conductor 30 but not in 31. It returns in opposite direction carried by conductor 154 only. In the cross-section behind the short-circuit (X-section 2) conductor 30 is deprived of current.

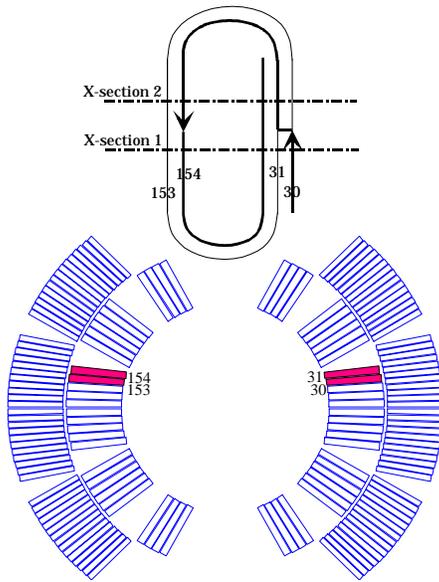


Fig. 6. Schematic representation of an inter-turn short-circuit

Table 4. Short circuit analysis at 10 mm

Normal	Reference	X-section 1	X-section 2
$B_1$	$-74 \cdot 10^{-4}$ T	$-59 \cdot 10^{-4}$ T	$-59 \cdot 10^{-4}$ T
$b_2$	0	3.3	0
$b_3$	-201.0	-257.9	-260.0
$b_4$	0	0.9	0
$b_5$	20.1	23.1	22.7
$b_6$	0	0.2	0
$b_7$	0.04	-0.06	-0.11

Skew	Reference	X-section1	X-section 2
$a_1$	0	6.4	0
$a_2$	0	37.1	33.1
$a_3$	0	1.2	0
$a_4$	0	5.5	5.2
$a_5$	0	0	0
$a_6$	0	0.7	0.7
$a_7$	0	-0.03	0

Table 4 illustrates how the short circuit affects the multipoles of the magnetic field in comparison to the reference field in this particular example. In cross section 1 the distribution of the current on both sides of the coil is asymmetric due to the fault. As a result measurable normal multipoles of even orders like quadrupole ( $b_2$ ), octupole ( $b_6$ ), and skew field harmonics ( $a_6$ ) with high value of skew quadrupole ( $a_2$ ) are developed. These field harmonics normally are not present in a dipole magnet without electrical faults. In cross section 2 the current distribution becomes symmetric again; even orders of the normal multipoles and odd orders of the skew harmonics vanish from the field pattern. This effect can be used for the longitudinal localization of the point of fault.

#### 5. EXPERIMENTAL SET-UP

##### 5.1 Detection method

The development of the field harmonics during a pulsed discharge can be detected by a system of pick-up coils (Fig.7). The detection method takes advantage of the vector field geometry characteristic for the even order multipoles. When the coils are positioned on the median plane in the center of the aperture the pulsed magnetic field will induce the same voltage in the coils C1 and C3



Fig. 7. Printed circuit pick-up coils

if the even orders of the normal field harmonics are not present in the field pattern. Otherwise, if the even-ordered harmonics are developed, a difference in the voltage induction between the two coils will be observed. The contribution of the even order multipoles cancels out in the central coil. Since higher order odd multipoles are negligible close to the center of the aperture, coil C2 picks up mainly the dipole field.

##### 5.2 Instrumentation

One of the MBSMS dipole magnets was equipped with micro-relays soldered on the internal layers in two clusters centered around the cross section in the middle of the magnet, and 280 mm away from the head region of the coil (Fig. 8). The relays, excited with 12 V DC one at a time, create short circuits between adjacent cable turns in well-defined positions. Pulsed current is supplied from a discharge generator by releasing the energy



Fig. 8. Coils equipped with micro-switches

accumulated in a  $1\mu\text{F}$  capacitor, charged at 1 kV, into the magnet. The voltage induced in the pick-up coils is measured during the discharge and integrated thereafter to obtain the time evolution of the magnetic field induction. The pick-up coil platform slides on two vetronite rods aligned inside the aperture of the magnet. The entire detection system can be rotated inside the magnet for the measurement of skew fields.

### 5.3 Results

Experimental results concerning the short circuit discussed in previous section (Fig. 6) are presented in Fig. 9. A measurable difference between the peak field inductions indicated by the coils C1 and C3, higher than the one predicted with the model, was registered in the

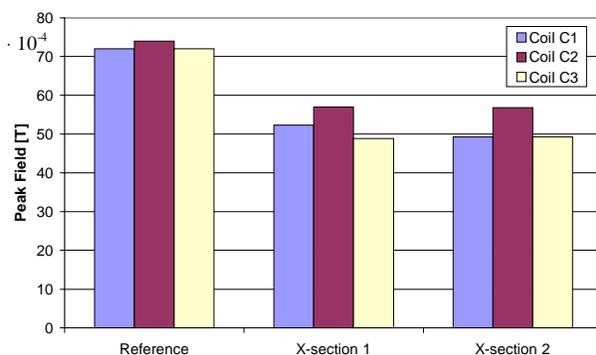


Fig. 9. Pick-up coil indications

area preceding the position of the short circuit (X-section 1). The induction difference vanished immediately, as expected, when the outer edge of the pick-up platform was moved behind the point of fault (X-section 2).

Indications of the central coil (C2) reflect further attenuation of the dipole field under fault conditions. As a consequence of the current redistribution and increased frequency of the current oscillations the field-screening effects in the magnet are enhanced and a degradation of the dipole field occurs. Increased difference between the indications of the coil C2 and the remaining coils of the pick-up system in comparison to the reference signals are attributed to the influence of the large negative normal sextupole predicted in the model. The magnitude of the sextupole at 10 mm is enhanced by 30% due to the short circuit.

## 6. CONCLUSIONS

Transient magnetic field distortions created by electrical faults in superconducting collared coils were analyzed in view of the localization of inter-turn short circuits. The detection method was based on the study of the development of the field harmonics induced by current redistribution. The experimental procedure has been simulated with OPERA-2D and tested on a dedicated dipole magnet equipped with micro-relays capable of activating artificial short circuits in the internal layer of the coils.

The longitudinal position of the short circuits was successfully established for all of the installed micro-switches with the precision of a few millimeters. An abrupt change of the normal even-ordered magnetic field harmonics was observed in the proximity of electrical faults. The magnet pole affected by the short circuit was also identified in each case. The possibility of an exact azimuthal localization of the inter-turn short circuits is under study.

## ACKNOWLEDGMENTS

The authors would like to thank N. Siegel and C. Wyss for their constant support and H. Kummer for his contribution to the assembly of the test magnet.

## REFERENCES

- [1] N.Andreev et al. " State of the short dipole model program for the LHC", presented at the 6<sup>th</sup> EPAC, Stockholm, June 22-26, 1998
- [2] The LHC Conceptual Design, CERN/AC/95-05.
- [3] S. Russenschuck et al., "Integrated design of superconducting accelerator magnets. A case of study of the main quadrupole", The European Physical Journal 1998, AP 1, 93-102
- [4] OPERA-2D, Unix version, Vector Fields Ltd.

# ANALYSIS OF EMITTANCE GROWTH AND COMPLEX IMPEDANCE FOR COHERENT SYNCHROTRON RADIATION SHIELDED BY TWO PARALLEL PLATE

Ryoichi Hajima\*,  
Dept. Quantum Eng. & Systems Sci., University of Tokyo,  
Hongo 7-3-1 Bunkyo-ku, Tokyo 113-8656 Japan

*Abstract*

Self-interaction of electron bunch in circular motion is numerically studied. It has been shown that the self-interaction is expressed in terms of Liénard-Wiechert potential formula and can be calculated with three-dimensional particle tracking. The relation between shielded radiation force and complex impedance of beam pipe is also discussed.

## 1 INTRODUCTION

In recent years several studies have shown that emittance growth due to self field of electrons traveling through circular motion may become a severe problem in the design of X-ray free-electron lasers, where electron-beam of extremely low emittance generated by photo-cathode RF-gun is accelerated in a linear accelerator without dilution of emittance and compressed by magnetic bunch compressors to achieve high peak current. The self field in circular motion is called as coherent synchrotron radiation (CSR) force or noninertial space charge force.

Coherent synchrotron radiation has been studied since 1940s when high energy accelerators were innovated [1]. The early studies were conducted mainly to estimate the amount of energy loss caused by radiation, because the analysis of energy loss was relevant to the construction of a high energy accelerator. In the present days CSR is an important issue regarding to emittance growth in a bunch compressor for X-ray FELs.

There have been several studies on this subject: one-dimensional analysis of CSR force on short electron bunches [2], transient analysis including finite length of magnets [3], and CSR in the motion of small period such as undulator [4]. As for numerical studies, it has been shown that Liénard-Wiechert potential together with particle tracking can be applied to the calculation of CSR and noninertial space charge force [5]. Emittance growth by shielded CSR has also been discussed [6] [7] [8].

In the present study, self-interaction of microbunch in circular motion is numerically studied with a three-dimensional particle tracking code. The relation between emittance growth due to shielded CSR force and complex impedance of beam pipe is also introduced.

## 2 ENERGY CHANGE AND EMITTANCE GROWTH IN A CIRCULAR PATH

When a relativistic electron bunch moves along a straight path, field generated by each electron is propagating with speed of light and hugging the source electron, then it does not much affect on the motion of other electrons. Self-interaction of the bunch, therefore, can be neglected for a relativistic energy.

In a circular path, however, field generated by tail of bunch can overtake head of bunch as shown in figure 1, and this self-interaction may affect the electrons motion. Electric field applied on single electron can be obtained by integrating contribution from whole of the bunch which is expressed in Liénard-Wiechert potential formula. In one-dimensional model, electron bunch can be expressed with line charge along the circular path. For line charge of Gaussian distribution, averaged energy loss of an electron in the bunch due to the self-interaction is given by

$$\Delta E_{ave} = \frac{0.34 q e L_d}{\rho^{2/3} \sigma^{4/3}}, \quad (1)$$

where  $q$  and  $\sigma$  are charge and characteristic length of the bunch,  $L_d$  and  $\rho$  are path length and radius of the circular motion. Total energy loss of the bunch is consistent with the power of coherent synchrotron radiation emitted from the bunch, then the self-interaction is often called as coherent synchrotron radiation force.

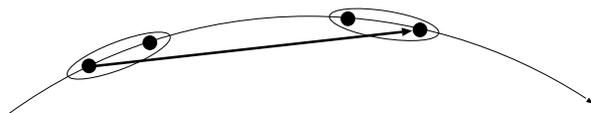


Figure 1: Schematic view of noninertial space charge field.

Estimation of emittance growth is more complicated than energy loss calculation. Although emittance growth can be discussed in terms of additional energy spread in the circular motion and expressed with a simple formula [9], it is limited to ideal situation. Therefore, numerical simulation is required for practical design analysis of bunch compressors.

The emittance growth due to self-interaction in circular motion can be numerically calculated with particle tracking method. Radiative and static field of relativistic electrons is obtained in the form of retarded potential, which is reduced to Liénard-Wiechert potential in case of small line

\* e-mail: hajima@q.t.u-tokyo.ac.jp

charge of density  $\lambda$  traveling through circular trajectory of radius  $R$  and azimuthal electric field in the static frame is expressed as [5]:

$$E_\theta = \frac{\lambda}{4\pi\epsilon_0} \frac{1}{r_{ret} - \vec{r}_{ret} \cdot \vec{u}_{ret}/c} \left( \frac{1}{\gamma^2} - \frac{x}{\rho} \beta^2 + \frac{r}{\rho} \beta^2 [1 - \cos(\zeta')] \right) \Bigg|_{\zeta_r}^{\zeta_f}, \quad (2)$$

where the first term is usual space charge force proportional to  $\gamma^{-2}$  which is often neglected for high energy electron beam, the second term is centrifugal force which does not appear in one-dimensional analysis, and the third term represents CSR force. The retarded position of source particle,  $\zeta'$ , is obtained from geometrical consideration. A three-dimensional particle tracking code **JPP** [10] has been modified to calculate the self-interaction in circular motion, where the numerical routine for the self-field is based on Eq.(2).

We consider an electron beam transport through two-dipole dog-leg as shown in figure 2. Parameters are chosen as: radius of circular path  $R = 2.292m$ , path length in each magnet  $L_d = 20cm$ , energy of electrons  $E = 400MeV$ . The electron bunch is assumed to be Gaussian shape in longitudinal direction,  $\sigma_z = 0.4mm$ , uniform in transverse direction,  $r_b = 1mm$ , and  $q = 1nC$ . Figure 3 shows calculated longitudinal phase plot after the dog-leg, where negative  $s$  means head of bunch. The results of one-dimensional theory is also plotted as a solid line. Emittance growth in the dog-leg is 0.44 mm-mrad, while the growth is 0.002 mm-mrad without the centrifugal force and CSR force.

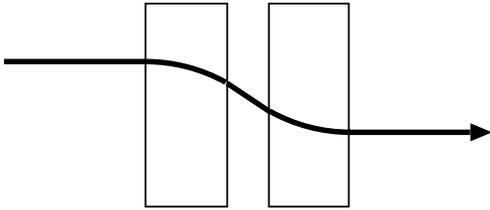


Figure 2: Two-dipole dog-leg consisting of two rectangular magnets.

### 3 EMITTANCE GROWTH BY SHIELDED RADIATION FORCE

In this section shielded CSR force is calculated by **JPP** code and compared with the results of impedance analysis.

The energy loss of electron bunch by CSR force in a metallic beam pipe, can be expressed as

$$\begin{aligned} \left( \frac{dW}{dt} \right)_{CSR} &= -(qc\beta/\rho)^2 \sum_{n=-\infty}^{\infty} |f_n|^2 \text{Re}[Z_n] \\ &= -(qc\beta/2\pi\rho)^2 \mathcal{R}, \end{aligned} \quad (3)$$

where  $q$  is charge of the bunch,  $Z_n$  is complex impedance

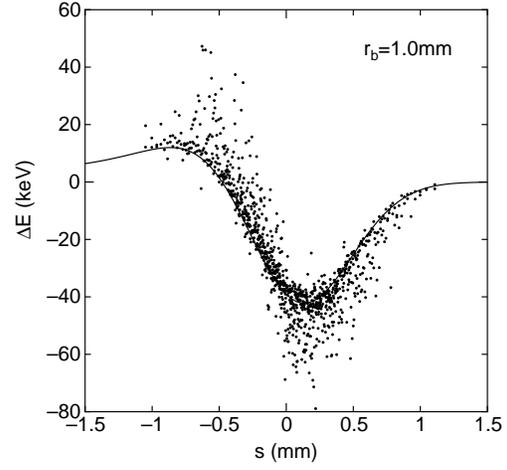


Figure 3: Longitudinal phase plot of electron bunch after the dog-leg.

of the beam pipe,  $\mathcal{R}$  is radiation resistance and  $f_n$  is a longitudinal Fourier component of electron bunch defined as

$$f_n = \frac{1}{2\pi} \int_0^{2\pi} \exp(-in\theta) \lambda(\theta) d\theta, \quad (4)$$

where  $\lambda(\theta)$  is longitudinal charge distribution of the bunch. For infinite two metallic plates placed parallel to the orbit plane, the impedance can be expressed as [1]

$$\begin{aligned} \frac{Z_n}{n} &= \frac{2\pi^2 Z_0 \rho}{\beta h} \sum_{j=1,3,\dots} \left[ \beta^2 J'_n (J'_n + iY'_n) \right. \\ &\quad \left. + \frac{(j\pi\rho/h)^2}{(n\beta)^2 - (j\pi\rho/h)^2} J_n (J_n + iY_n) \right], \end{aligned} \quad (5)$$

where  $h$  is the distance between two plates, the argument of Bessel functions is  $\sqrt{(n\beta)^2 - (j\pi\rho/h)^2}$  and we assume perfect conductivity of the plates.

The effect of conducting plates on the self field in a particle tracking code can be completely simulated by image charge method. Although infinite number of image charges are required to fulfill the boundary condition on the two parallel plates, we can truncate the number of image charges without degradation of accuracy in the simulation, regarding the field dependency on the distance between source and observer. The number of image charges in the following calculations is carefully determined not to introduce numerical error arising from this truncation. In the calculation of retarded position, a tentative solution with ‘‘pencil beam’’ approximation is refined through iterative algorithm, the approximation is, however, not appropriate for image particles far from central axis of the trajectory. A special approximation, therefore, is introduced to obtain a tentative retarded position for image particles [10].

We consider a two-dipole dog-leg having a beam line surrounded by infinite two parallel plates to study the shielding effect on coherent radiation and emittance

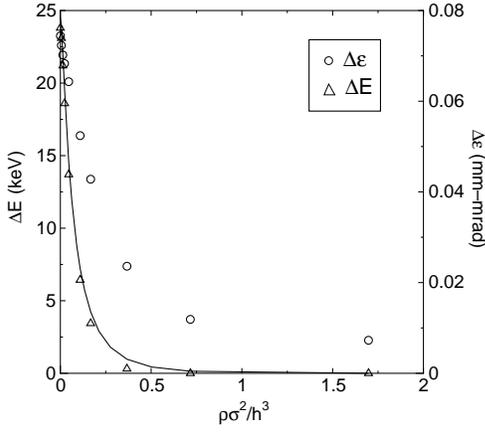


Figure 4: Averaged energy loss of an electron and emittance growth as a function of dimensionless shielding parameter: energy loss ( $\Delta$ ) and emittance growth ( $\circ$ ) calculated by **JPP** code, energy loss calculated from impedance analysis (solid line).

growth. The energy loss of electrons and normalized emittance growth are calculated with various distance between two plates. Figure 4 shows the calculated energy loss and emittance growth as a function of dimensionless shielding parameter  $\rho\sigma^2/h^3$ , where the simulation parameters are chosen as:  $E = 400\text{MeV}$ ,  $r_b = 0.1\text{mm}$ ,  $q = 1\text{nC}$ ,  $\sigma_z = 0.4\text{mm}$ ,  $\rho = 2.292\text{m}$ ,  $L_d = 0.2\text{m}$ .

It can be seen that obtained energy loss by **JPP** code agrees with the result of impedance analysis, where energy loss can be neglected for large shielding parameter as predicted before [11]. It seems that the reduction of emittance growth by shielding of metallic walls is not as drastic as the energy loss. While the energy loss due to coherent radiation for  $h = 10\text{mm}$  ( $\rho\sigma^2/h^3 = 0.37$ ) is less than 1keV, 1/30 of energy loss without shielding, the emittance growth is about 1/3 of open duct. This different property of shielding effect between energy loss and emittance growth can be explained by longitudinal phase plot after the dog-leg as shown in figure 5.

It shows that relatively large energy modulation still exists, while total energy loss of the electron bunch is almost zero. This energy modulation arisen in the dog-leg results in emittance growth. The estimation of emittance growth, therefore, requires amount of energy modulation of the electron bunch. It means that we should consider the imaginary part of impedance in the emittance analysis as well as the real part which is directly related with total CSR power. We introduce a concept of complex radiation resistance to include imaginary part of the impedance:

$$\mathcal{R}_c = (2\pi)^2 \sum_{n=-\infty}^{\infty} |f_n|^2 (\text{Re}[Z_n] + i \text{Im}[Z_n]). \quad (6)$$

Figure 6 shows obtained emittance growth by **JPP** code as a function of inverse of magnitude of complex radiation resistance calculated from Eq.(6). It seems that the emittance

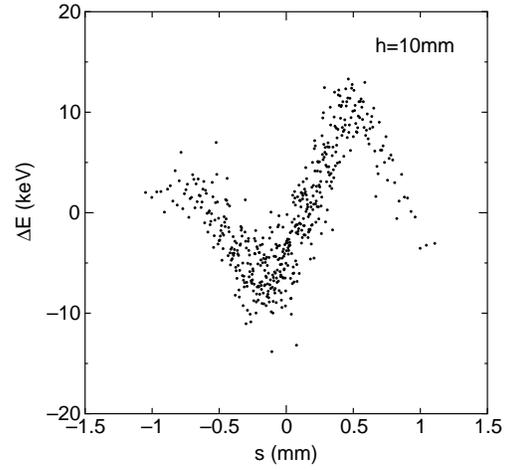


Figure 5: Longitudinal phase plot of electron bunch after the dog-leg.

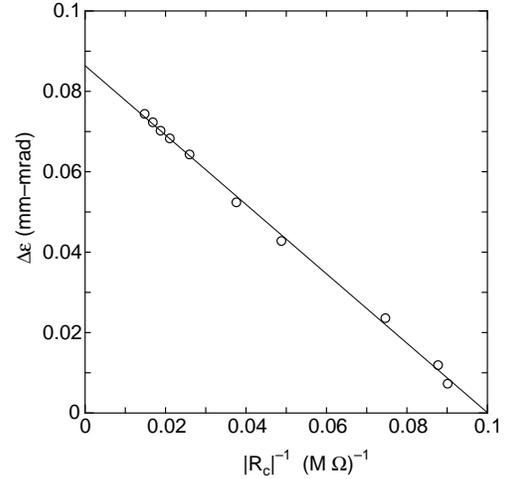


Figure 6: Obtained emittance growth by **JPP** code as a function of inverse of magnitude of complex radiation resistance.

growth of an electron bunch due to shielded coherent synchrotron radiation can be expressed as a function of complex radiation resistance.

## 4 SUMMARY

In the present study we have applied three-dimensional particle tracking together with Liénard-Wiechert potential to the calculation of self-interaction of microbunch in circular motion, which should be considered in the design of bunch compressors for X-ray FELs. Averaged energy loss of an electron due to CSR force is calculated with the simulation code and found to be consistent with one-dimensional analytical results. In the case of shielded CSR force, the obtained energy loss agrees with the result of impedance analysis. It is also found that emittance growth of an electron bunch caused by shielded CSR force can be expressed as a function of complex radiation resistance of beam pipe.

## 5 REFERENCES

- [1] J.Schwinger, "On Radiation by Electrons in a Betatron", unpublished (1945).
- [2] Ya. Derbenev, et al., "Microbunch Radiative Tail-Head Interaction", DESY-print TESLA-FEL 95-05 (1995), unpublished .
- [3] E.L.Saldin et al., "On the Coherent Radiation of an Electron Bunch Moving in an Arc of a Circle", DESY-print TESLA-FEL 96-14 (1996), unpublished.
- [4] E.L.Saldin et al., "Radiative Interaction of Electrons in a Bunch Moving in an Undulator", DESY-print TESLA-FEL 97-08 (1997), and in Proc. of 20th FEL Conf.
- [5] B.E.Carlsten, Phys. Rev. E54 (1996), 838.
- [6] R.Li et al., "Shielded Transient Self-Interaction of a Bunch Entering a Circle from a Straight Path", Proc. SPIE, San Diego (1997).
- [7] M.Dohlus et al., "Uncorrelated Emittance Growth in the TTF-FEL Bunch Compression Sections Due to Coherent Synchrotron Radiation and Space Charge Effects", Proc. EPAC Conf. 1998, Stockholm, and DESY-M-98-060.
- [8] R. Hajima and E. Ikeno, "Numerical analysis of shielded coherent radiation and noninertial space-charge force with 3-D particle tracking", to be published in Nucl. Instr. and Meth. A.
- [9] B.E.Carlsten, Phys. Rev. E51 (1995), 1453.
- [10] R.Hajima, "JPP: A PARTICLE TRACKING CODE WITH JAVA", in Proc. of Int. Computational Accelerator Physics Conf. (ICAP'98) (1998).
- [11] S.A.Kheifets and B.Zotter, "Shielding Effects on Coherent Synchrotron Radiation", CERN-SL-95-22-AP (1995), unpublished.

# CALCULATION OF EIGENMODES IN SUPERCONDUCTING CAVITIES ON APE-SUPERCOMPUTERS USING A SOFTWARE INTERFACE TO MAFIA

F. Neugebauer, DESY Zeuthen, Platanenallee 6, D-15738 Zeuthen, Germany  
U. van Rienen, Universität Rostock, Fachbereich Elektrotechnik und Informationstechnik,  
Albert-Einstein-Straße 2, D-18051 Rostock, Germany

## *Abstract*

To design modern accelerators a profound knowledge of eigenmodes of RF-cavities is required. For normal conducting as well as for superconducting cavities MAFIA is a well established tool to determine the eigenmodes by numerical means. However, the 3-dimensional treatment of multicell cavities lacks from available computer power on a usual high end workstation. Therefore the present approach uses a parallel SIMD supercomputer (APE-100) to compute the numerical expensive part of the MAFIA-algorithm. The system matrix, incorporating geometry and material information, is transferred to the APE-100 during a normal MAFIA-session using a command provided by the MAFIA toolkit (MTK). Then, on the APE-100 the lowest eigenvalues and their corresponding eigenvectors of the system matrix are determined by means of a conjugate gradient algorithm [3]. The result of the diagonalization procedure is then read back to the MAFIA host where further data analysis and visualization can be done.

## 1 INTRODUCTION

The construction of modern accelerators is usually supported by the numerical determination of eigenmodes in the accelerating cavities. Often the rotational symmetry of the cavity is used to simplify the numerical simulation. However, in cases where the cavity plus attached rf-components lacks rotational symmetry a fully 3-dimensional treatment of Maxwell's equations is necessary which requires more computer power than is available on a normal high end workstation. In addition the 3-dimensional approach allows for the simulation of fabrication errors and surface roughness which are usually not considered to have rotational symmetry.

In the framework of the Finite Integration Technique (FIT) developed by Weiland and coworkers[1] Maxwell's equations in integral representation are transformed to a set of matrix equations. Using rectangular grids the discretization volume is partitioned in two sets of cells which can be considered dual. In the case of determining the eigenmodes of a cavity the grid voltages along neighboring gridpoints are the degrees of freedom of the resulting eigenvalue problem. It turns out that the so-called system matrix connects grid voltages of a single cell only to grid voltages of adjacent cells. This "next neighbor connection"-property makes the eigenvalue problem especially well suited to be solved on an APE-100 supercomputer for this type of com-

puter is capable of a very fast data exchange between neighboring nodes.

APE-100 supercomputers are mainly used for in QCD theory where a profound experience in solving eigenvalue problems [3] does exist. However, the parallel structure of the computer requires the use of special programming tools and a language (TAO) dedicated to the computer topology which is inefficient in programming advanced file IO, string evaluation and managing pointers.

Therefore the parsing of the geometry input, which is mainly a linear task, is left to MAFIA which is running on a usual workstation. The resulting system matrix incorporating geometry and material information is transferred to the APE-100 by means of the MAFIA toolkit (MTK). Then, on the APE-100 supercomputer the numerical expensive task of finding the lowest eigenvalues and corresponding eigenvectors of a large sparse matrix is performed. The result of the diagonalization procedure is then read back to the MAFIA host where further data analysis and visualization can be done.

The paper is organized as follows: in section 2 a short overview of the Finite Integration Technique and the APE-100 topology is given. In the next section the matrix vector multiplication which is crucial for the used algorithm is considered in detail. In section 4 a performance analysis of the conjugate gradient algorithm used in this approach is given.

## 2 FINITE INTEGRATION TECHNIQUE AND THE APE-100 TOPOLOGY

The Finite Integration Technique is based on a discretization of Maxwell's equations using a set of two rectangular grids which can be considered dual to each other [1]. The integral representation of Maxwell's equations is transferred to a discrete version by specifying the integration paths as to be along the edges of the discretization cell. For the case of area integrals the 6 bordering rectangles of the cell are chosen as the integration area. The degrees of freedom in the discretized version of Maxwell's equations are not the fields itself, moreover for example the grid voltage along neighboring grid points or the flux over a cell border are used. Therefore the discretized Maxwell's equations remain mathematically equivalent to the continuous case. There is no discretization error and the discretized Maxwell's equations exactly obey the conservation law for charge and current density.

An important feature of the matrix equations is its locality. Actually this is due to the fact that the chosen integration space is restricted to the neighboring cells of the selected degree of freedom. As a consequence the resulting system matrix which eigensystem has to be determined is sparse with a priori known pattern of entries. A detailed treatment of the theory yields that 13 elements of the system matrix are non zero for each degree of freedom. These elements connect to degrees of freedom belonging to neighboring cells.

The locality of the matrix equations can be exploited on APE-100 in a quite natural way. The nodes of the APE-100 supercomputer are arranged on a three-dimensional rectangular grid (see Fig.1) as is the grid used for the discretization of Maxwell's equations. The cells of the discretization volume are distributed to the nodes so that each processor is responsible for its own segment of real space.

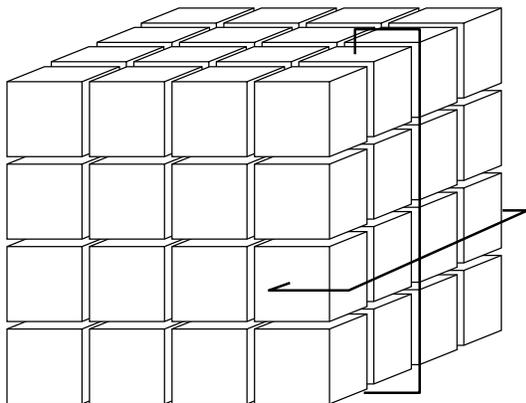


Figure 1: Topology of the APE-100 (4x4x4 nodes). Between adjacent processors there is a fast data transfer which is about 4 times slower than local memory access. The whole cube is subject to periodic boundary conditions in x-,y- and z-direction resulting in a hyper torus. Due to the SIMD character of the APE-100 no latency time occurs on data transmission.

The matrix-vector multiplication is then mainly a local operation on each node. Only in cases where the cell lies on the segment boundary data exchange with neighboring nodes will occur. The APE-100 is perfectly suited to such a situation because the SIMD character of the supercomputer accounts for a very fast data transfer with neighboring nodes without latency.

One drawback of numerical determination of eigenmodes is that the solution space is composed of two solution spaces – one holding the eigenmodes which are searched for and one holding the so-called ghost modes. For FIT there is a workaround excellently described in [2]. Another difficulty arises from the periodic boundary conditions which are built in to the APE-100 topology but are not implemented in the MAFIA package for all coordinate directions. Therefore boundary flags are necessary in the present approach.

### 3 DATA DISTRIBUTION STRATEGY OF THE SYSTEM MATRIX

Heart of the conjugate gradient algorithm described in[3] is the Ritz method applied to the functional  $\mu$

$$\mu(\vec{z}) = \frac{\langle \vec{z}, \hat{A}\vec{z} \rangle}{\langle \vec{z}, \vec{z} \rangle}, \quad (1)$$

where  $A$  denotes the system matrix and  $z$  is the vector of grid voltages. Given a random initial vector the algorithm searches for the minimum of  $\mu$  in the orthogonal subspace of all previously determined eigenvectors. Acceleration of the algorithm is achieved by using exact diagonalization in the subspace spanned by the numerically computed eigenvectors.

As usual for algorithms determining the eigensystem of large sparse matrices the efficient coding of the matrix vector multiplication routine is crucial for a high performance of the algorithm. Therefore the matrix-vector multiplication routine was coded using the extract-replace method of TAO allowing for efficient use of all of the 128 registers of the FPU.

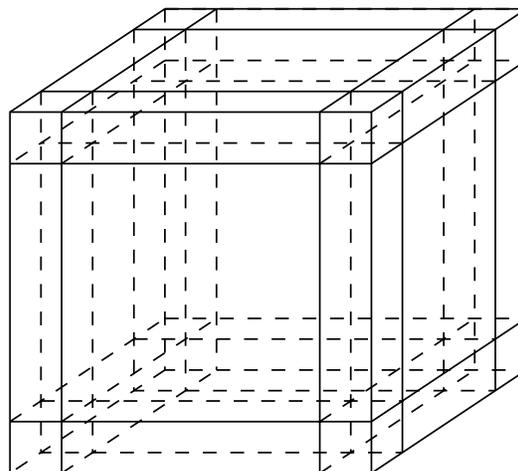


Figure 2: To avoid "IF" statements and flags in the TAO source code the total number of cells residing on one node is divided into 27 parts with definite neighboring relations. For the bulk region (fully inside the box) all neighboring cells are in the local memory of the node. For cells in other pieces it is known at programming time which part of the matrix-vector multiplication requires remote data access.

In addition care had to be taken in the multiplication routine where boundaries of the segments were involved. To simplify the coding of the multiplication loop by avoiding IF statements and flags definite neighbor relations of the cells were introduced by splitting up the whole segment into 27 pieces (see Fig.2). As a consequence the source code of the matrix-vector multiplication is somewhat blown up since every single multiplication operation has to be written 27 times for every piece of the segment.

## 4 PERFORMANCE OF THE ALGORITHM

The APE-100 supercomputer is mainly used for long running simulations in the field of lattice QCD and similar theories in high energy physics. To give easy access to a community familiar with MAFIA a software interface to the APE-100 has been written. After defining the geometry in the M (mesh) module of MAFIA, the E module is started and a new command “MaxqSolve” is issued which writes the system matrix to disk and submits the appropriate script to the APE-100 queue. After the solver on the APE-100 has finished the user can read in the resulting eigensystem by giving the “MaxqGetResults” command to the E module. The whole data transfer to and from the APE-100 is hidden from the user.

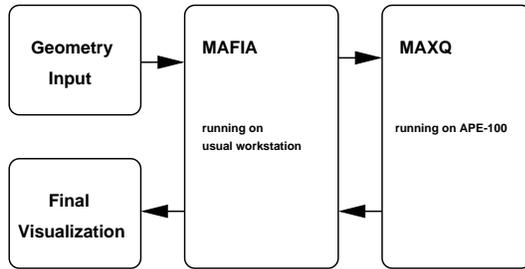


Figure 3: Schematic view of the software interface

The total number of gridpoints which can be used in the discretization is restricted due to memory limitations. Furthermore the used eigensolver needs additional memory for every eigenvector to be found. The memory needed per gridpoint is 28 octets for the system matrix and 12 octets for every eigenvector. This leads to

$$M_{total} = 12 * N * (n + 5) \text{ octets} \quad (2)$$

for an expression for the total memory used.  $N$  is the number of gridpoints and  $n$  denotes the number of eigenvectors to be found.

On the QH2 (8x8x4) with 16 MB per node eq. (2) results in:

number of eigenvectors	available gridpoints
1	60.000.000
2	51.000.000
5	36.000.000
10	24.000.000
20	14.000.000
50	6.500.000
100	3.400.000

In principle the last table shows the applicability of the proposed approach. However, investigations considering the role of the single precision floating point arithmetic of the APE-100 are to be done. The scalar products are reported to be sensitive to the single precision / double precision problematics. Therefore the scalar products are coded using a software emulation of double precision arithmetics.

The same procedure has already been applied in the original MAFIA package to save memory and keep rounding errors at a minimum.

## 5 CONCLUSIONS

The calculation of eigenmodes on a three-dimensional basis is crucial for the simulation of accelerating cavities. Only a fully three-dimensional treatment of Maxwell’s equations can account for effects connected to fabrication errors and surface roughness and most important to effects which arise from devices such as input couplers or HOM-couplers. These coupling devices inevitably break the rotational symmetry of the cavity cannot be neglected for the determination of eigenmodes of the cavity. However, the lack of computational power on high end workstations normally avoids the inclusion of the three-dimensional effects described above. Therefore in the present paper an approach to a supercomputer solution of the eigenmode problem of superconducting cavities has been made.

Though there is no real example of the approach until now the performance analysis of the algorithm shows that determining the eigenmodes of superconducting cavities on APE-100 supercomputers is a reasonable idea to overcome the limits of high end workstations. Work for the application to parts of the TESLA structure is in progress.

## 6 ACKNOWLEDGEMENTS

The authors wish to thank Prof. Th. Weiland and his colleagues for many helpful discussions on MAFIA and H. Simma for helping implementing the eigensystem solver and for many hints concerning the APE-100. The support given by DESY in providing computing time for the MAXQ project (Maxwell’s Equations on Quadrics) is gratefully acknowledged.

## 7 REFERENCES

- [1] Th. Weiland, ‘On the Numerical Solution of Maxwell’s Equations and Applications in Accelerator Physics, Part. Acc. , **15**, 245–291 (1984)
- [2] Th. Weiland, ‘On the Unique Numerical Solution of Maxwellian Eigenvalue Problems in Three Dimensions’, Part. Acc. ,**17**, 227–242 (1985)
- [3] Th. Kalkreuther, and H. Simma, ‘An accelerated conjugate gradient algorithm to compute low-lying eigenvalues – a study for the Dirac operator in SU(2) lattice QCD’, DESY 95-137, HUB-IEP-95/10 (1995)

# INVESTIGATION OF THE IMPEDANCE AND HIGHER ORDER MODE LOSSES FOR PROPOSED BEAM PIPE CONFIGURATIONS FOR THE HERA LUMINOSITY UPGRADE PROJECT

M. Dohlus, S. G. Wipf  
 Deutsches Elektronen Synchrotron (DESY)  
 Notkestrasse 85, 22607 Hamburg, Germany

## Abstract

The impedance and wakefield effects and higher order mode losses have been investigated for the Luminosity Upgrade Project at HERA for three regions where only minimal heating can be tolerated. These are two versions of the beam pipe configuration at the interaction region of the experiment Zeus and also for a region where superconducting magnets are to be installed. As the structures are very long (up to 4.9m) it was not possible to calculate all modes up to the cut-off frequency directly, thus long term wake calculations (100nsecs) in the time domain were used to pinpoint potentially dangerous modes. The frequency and band-width information thus obtained could be used to obtain the impedance of these modes in the frequency domain. The calculations were carried out using the MAFIA [1] programs.

## 1 IMPEDANCE CONSIDERATIONS

Impedance calculations were carried out for the beam tube of the interaction region at ZEUS and also in the adjoining region of the superconducting magnets, using the MAFIA programs [1]. Both wakefield (time domain), and resonant field (frequency domain) calculations were made. In the frequency domain the resonant fields were calculated for as many modes as possible, these fields can then be used to obtain the loss parameter, shunt impedance and power loss. Assuming that only one mode will be excited exactly on resonance at any one time, the maximum values give a good worst case estimate of the potential power loss.

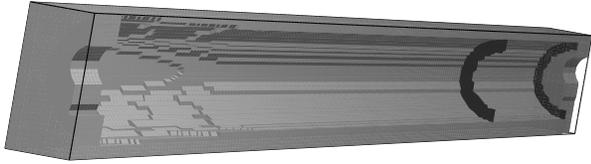


Figure 1: Version I: A foreshortened view of the interaction region at Zeus, the absorbers are shown as dark arcs.

As the damping time of the wakefields is longer than the time between bunches, the single pass loss, calculated from wakefields, does not give an *absolute* value of the power loss which can be expected. It is possible that the wakefields from each bunch accumulate and that the final figure

is higher than the calculated transient losses, however this figure has been used as a measure of comparison for different beam tube geometries and for different longitudinal bunch lengths.

The Version II design entails considerable geometrical alterations. A much smoother beam tube, with tapers at either side of the collimators helps to reduce the losses due to wakefields and an asymmetrical absorber at  $z = 1.65m$  has the effect that many fewer modes are actually trapped in the interaction region. However the change from normal to superconducting magnets reduces the tolerance to heating due to resonant and transient losses further along the beam pipe. Thus it was necessary to consider the impedance in the region of the superconducting magnets in addition to that in the interaction region.

## 2 DEFINITIONS

### 2.1 Time Domain

The wake potential of a point charge,  $q$ , travelling at the speed of light,  $c$ , as seen by a test charge at a distance,  $s$ , behind it, is defined as [2],

$$W(s) = \frac{1}{q} \int_{-\infty}^{\infty} E_z(x, y, z - s, t = z/c) dz \quad (1)$$

Provided that the wake fields from one bunch have died away when the next bunch arrives, then the power loss,  $P$ , is given by

$$P = \frac{U}{T} = -\frac{1}{T} \int_{-\infty}^{\infty} W(s) \cdot c\lambda(s) ds \quad (2)$$

where  $T$  is the time between successive bunches,  $U$  the energy and  $\lambda(s)$  the charge distribution (usually Gaussian).

### 2.2 Frequency Domain

In the frequency domain it is unusual for modes where  $\omega > \omega_{cutoff}$  to be trapped. The loss parameter  $k_\nu$  for a particular mode,  $\nu$ , with frequency,  $f_\nu$  and  $\omega_\nu = 2\pi f_\nu$  is given by

$$k_\nu = \frac{|\int_{-\infty}^{\infty} E_{z\nu}(z) \cdot dz \cdot e^{i\omega_\nu z/c}|^2}{4 \cdot TotalEnergy(\nu)} \quad (3)$$

and  $Q_\nu$  the quality factor.

$$Q_\nu = \frac{\omega_\nu \cdot StoredEnergy(\nu)}{TotalLosses(\nu)} \quad (4)$$

Thus the worst case power loss for a single mode is given by

$$P = 2 \cdot I_0^2 \cdot R_{shunt} \quad (5)$$

where  $R_{shunt} = 2kQ/\omega$  represents the shunt impedance at a resonant frequency with  $I_0 = \text{DC beam current}$  and  $2\pi\sigma \ll c/f\nu$ .

These quantities are all obtained from the MAFIA eigenmode calculation and are, in addition, only weakly dependent on changes in frequency caused by small geometrical changes. The maximum shunt impedances occur mainly at lower frequencies.

It has been assumed that the probability, that the frequency of more than one mode coincides with a line of the beam spectrum at any one time, is very small, so that one is justified in using values for the mode with the highest shunt impedance for the worst case estimate.

### 3 GEOMETRY

#### 3.1 Version I:

A 3.2 meter long section was modeled around the ZEUS Interaction Point ( $z = 0.0m$ ), which formed a resonant cavity due to the changes in cross section at either end. It was only necessary to model half the structure as it is symmetrical with respect to the x-z plane. The electron (or positron) beam travels in the positive z-direction.

The geometry consisted of, from left to right, (see Figure 1):

- keyhole beam pipe, an elliptical beam tube inside the combined function magnet with lateral extension for synchrotron light;
- opening up into a conical section with an entry radius of  $54mm$  at  $z = -1.3m$  and an exit radius of  $46.25mm$  at  $z = -0.62m$ ;
- this is followed by a long cylindrical central beam pipe from  $z = -0.62$  to  $1.3m$ ;
- at  $z = 1.3m$  there is a sudden change in cross section to an elliptical beam tube inside the other combined function magnet with a wider extension than on the left hand side.
- Two synchrotron radiation absorbers were positioned at  $z = 0.75m$  and  $1.3m$ .
- keyhole beam pipe

A second configuration was also investigated using an elliptical central beam pipe from  $z = -1.3m$  to  $1.3m$ . The height of the beam tube at the interaction point was  $20mm$  as opposed to  $46.25mm$  for the cylindrical cross-section, the width remained unchanged.

#### 3.2 Version II - Interaction Region:

A 3.9m stretch of beam tube was calculated. The geometry consisted of, again starting from the left, (see Figure 2):

- an elliptical beam tube  $30mm$  high with a small post or "finger" absorber on one side of the horizontal axis at  $z = -1.75m$ ;

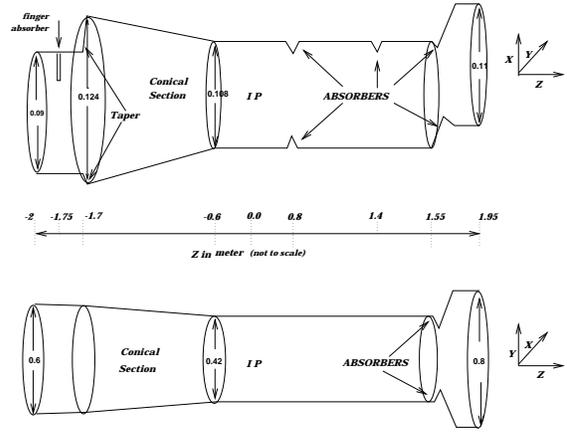


Figure 2: Version II: A diagram of the Interaction Region.

- a step-wise increase in width from  $91$  to  $123mm$  at  $z = -1.7m$ , masked by r.f. shielding, is followed by an elliptical cone which continues until  $z = -0.6m$  where the height is  $21mm$ ;
- this is followed by a long elliptical central beam pipe, width  $108mm$  from  $z = -0.6m$  to  $1.55m$ ;
- at  $z = 1.55m$  there is an assymetrical tapered absorber;
- a pair of tapered synchrotron radiation absorbers were positioned either side of the horizontal axis at  $z = 0.8m$  and a single absorber at  $z = 1.4m$ .

The symmetry with respect to the x-z plane could also be taken advantage of here.

#### 3.3 Version II - Superconducting Magnets:

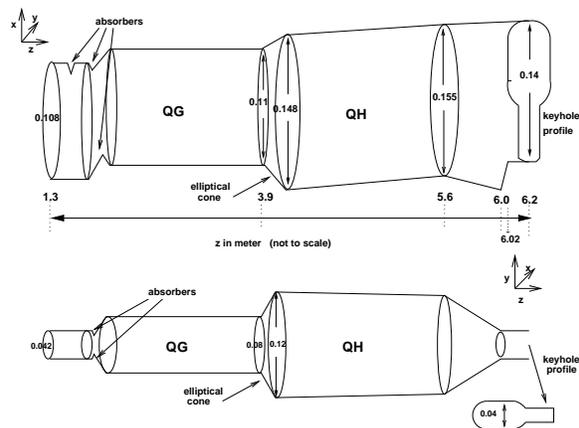


Figure 3: Version II, Superconducting Magnets: A diagram (not to scale) of the beam tube at the Superconducting Magnets . (The last 2 absorbers at the end of the interaction region can be seen at the beginning of this section.)

For the superconducting magnets the section from 1.3m to 6.2m was modelled, from left to right (see Figure 3):

- the rightmost end of the interaction region beam tube including the absorber at 1.4m and the staggered absorbers between 1.55m and 1.66m
- the elliptical beam tube in right hand side magnet with vertical height of 40mm
- the conical connection between the magnets, where the vertical height changes from 40mm to 60mm between  $z = 3.9m$  and 4m
- the larger elliptical beam tube continues through the second magnet to  $z = 5.6m$ . As the path of the beam is curved in this section, slight shifts were included in the geometry so that the distance of the beam from the walls remained correct.
- a gradually tapering section starting at 5.6m connects to the keyhole chamber of next magnet at 6m where extra space is needed to accomodate the synchrotron radiation

#### 4 TIME DOMAIN - INTERACTION REGION

For the Version II configuration, the damping time for a typical mode trapped in the Interaction Region with frequency 3954MHz and Q-factor 14000 is given by

$$\tau = 2Q/\omega \approx 1\mu sec \text{ (11bunches)}.$$

As the bunch separation time is only 96 nanoseconds, this means that the wakefields from one bunch are still present when the next bunch arrives. The two extreme cases are when these fields are either in phase or out of phase by  $\pi$  with the wakefields from the next bunch. In the former case the fields accumulate while in the latter case they cancel. The losses drop dramatically with increasing bunch length, so that maximum losses can be expected at injection. In HERA the longitudinal bunch length of the electrons or positrons (in contrast to the protons) is shortest at injection and longer at higher energies. The longitudinal  $\sigma$  for positrons or electrons varies with beam energy as shown in Table 1.

Table 1: Variation of Longitudinal  $\sigma$  for positrons/electrons with beam energy

Energy:	12 GeV	27.5 GeV	29.8 GeV
Voltage:	90 MV	110 MV	140 MV
Sigma:	3mm	12.3mm	13.4mm

In order to simulate the beam, the time domain program represents the bunch as a rigid longitudinal Gaussian charge distribution. For Version I, 4mm was the smallest value of the  $\sigma$  of the distribution which could be calculated directly, as the extreme ratio of height to length of the beam tube begins to cause computational problems. The 3mm bunch length was calculated in two sections. For the left hand side section it was necessary to take the group

velocity of the wakefields into account and to calculate to the point where the tail of the bunch has outstripped the wakefields,  $z = 0.9m$ .

These very lengthy calculations were only repeated for the Version II geometry for a  $\sigma$  of 4mm, all other Version II calculations were carried out with a bunch  $\sigma$  of 8mm. The Version I values which are included in Table 2 indicate the relationship between bunch length and transient losses and the effect of reducing the height of the beam tube. The steep increase of transient losses with a 3mm bunch length is expected to be similar for the Version II configuration. The Version II single pass transient losses were calculated to be 47 Watt for a  $\sigma$  of 8mm. The bunch spectrum for a 3 mm and an 8 mm  $\sigma$  bunch only differ in their high frequency content, for frequencies well above 5 GHz. The increased transient losses need not be trapped in the interaction region, although there will be reflections at absorbers and other changes in cross-section. Thus only a small portion of the losses are deposited in the walls.

Table 2: Summary of the Single Pass Losses and of the Maximum Single Mode Resonant Power Losses

Beam Tube	Bunch Length	Single Pass Power Loss	Resonant Power Loss
<b>Version I:</b>	mm	Watt	Watt
<i>Cylindrical</i>	3	1870	1993
	4	1338	1978
	7	597	1910
	8	493	1881
	13	203	1684
<i>Elliptical</i>			
	3	873	
	8	138	420
<b>Version II:</b>			
<i>Interaction Region</i>	4	187	
	8	47	3
<i>s.c. magnets</i>	8	108	10
<b>Present Set-up</b>			
	8	395	1212

#### 5 FREQUENCY DOMAIN - INTERACTION REGION

The first 75 modes which lie between 2.2 and 4.14 GHz could be calculated. The majority are not trapped and the energy can flow away and dissipate in the rest of the ring, however one trapped mode was identified which resonated at the "finger" absorber at 2.58GHz. If this mode were to be excited, 94Watt would be dissipated at that position. Two variations of this absorber were calculated, the first extending it to the walls on both sides so that it could not so easily resonate and the second adding an additional taper on the down-stream side. In the former case a similar mode was found at 2.83 GHz, reducing the losses to

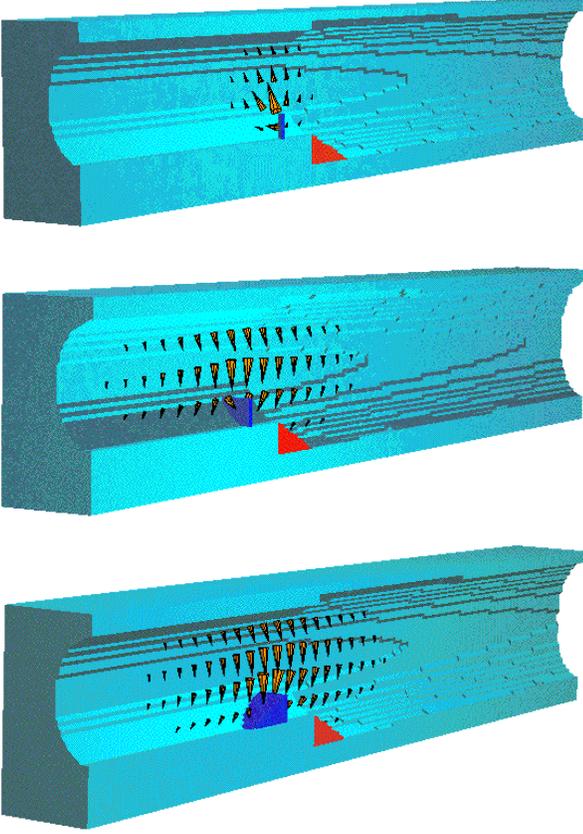


Figure 4: Three versions of the absorber at  $z=-1.75m$ . Top: the original "finger" design - resonant loss=94.5 Watt, frequency=2.58 GHz. Middle: "sector" design - resonant loss=17 Watt, frequency=2.83 GHz. Bottom: "sector with taper" - resonant loss=3 Watt, frequency=2.82 GHz.

17 Watt, while the taper produced a further reduction to 3 Watt at a frequency of 2.82GHz (see Figure 4). The long range wake was calculated for 80 nsecs. Examination of the fourier transformation showed no sharp resonances beyond the range of frequencies which could be calculated in the frequency domain, so no trapped modes are expected at higher frequencies.

### 5.1 Time Domain - Superconducting Magnets

It was possible to calculate the longitudinal wake fields with reasonable accuracy, although, with an over 4m section, one enters a regime where the step size has to be chosen much smaller in relation to  $\sigma$  in order to keep the numerical dispersion error small, on account of the asynchronisation of the bunch and the wakefields. For  $\sigma = 8mm$  a transient loss of 108 Watt was calculated.

### 5.2 Frequency Domain - Superconducting Magnets

As the beam tube is only 21mm high at the right hand side of the IP region and 20mm in the keyhole section, the superconducting magnet section forms a quasi-cavity

for the beam. Table 3 shows the chamber heights and cut-off frequencies for the various sections. Due to the greater vertical height of the beam tube in the right hand side superconducting magnet and the small opening of the keyhole vacuum chamber beyond, a large range of modes can be trapped in this region. Thus the 80 modes which can be calculated without overstepping the file and memory limitations lie below 2.5GHz and another 200 modes can be expected to lie between 2.5GHz and the cut-off frequency, 4.39GHz. Within these 80 modes all resonant losses were very moderate, ( $< 4Watt$ ).

Table 3: Vertical chamber height and cut-off frequency from IP to the keyhole vacuum chamber

IP beam pipe, (-2, -1.6):	2.97Ghz	.0295m
IP beam pipe, (-0.6,1.55):	4.19Gh	.021m
first s.c.magnet beam pipe:	2.2Ghz	.04m
second s.c.magnet beam pipe:	1.46Ghz	.06m
Keyhole:	4.39Ghz	.02m

Modes can be calculated at higher frequencies but the reliability of the results depends on knowing the number of modes which lie in a particular bandwidth. Thus there is no guarantee that *all* modes have been found, whereas when the whole frequency range is used modes are only rarely missed. 10 modes spanning 70 MHz were calculated at 4 GHz using this method and a maximum resonant loss for a single mode was found to be 10 Watt, this is the figure which is entered in Table 4 but the possibility of higher resonant losses cannot be excluded.

Table 4: Superconducting Magnets: Resonant Losses for 10 modes near 3950 MHz

Frequency (MHz)	Q-factor	Kz (V/C)	Rshunt (Ohm)	Loss (Watt)
3942.351	4540	1.035E+09	379.0	2.73
3944.285	6143	3.595E+08	178.0	1.28
3945.393	4265	2.653E+08	91.3	0.657
3947.076	8359	1.031E+08	69.5	0.500
3950.215	5345	2.807E+09	1209.0	8.71
3951.592	4810	3.288E+08	127.0	0.917
3960.004	6958	3.850E+08	215.0	1.55
3963.570	5705	1.311E+09	600.0	4.32
3965.430	5136	1.456E+09	600.0	4.32
3979.007	4415	5.219E+07	18.4	0.133
4012.014	5312	3.420E+09	1441.0	10.4

To aid in the analysis of the resonant losses in the superconducting magnets at higher frequencies, a long range wake calculation was carried out. A similar method was used in reference [3]. The wake potential was calculated

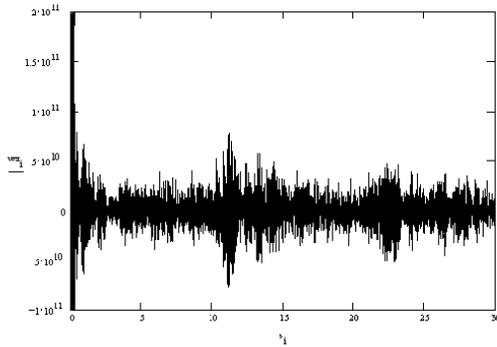


Figure 5: Left: The wake potential, calculated for 30m after the bunch head, plotted against distance from the head of the bunch, abscissa: 0 to 30m.

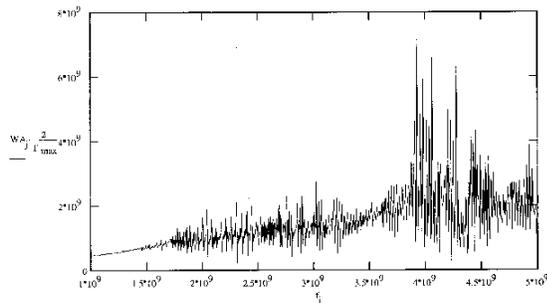


Figure 6: Superconducting magnet beam tube: Long term wake calculation for 100 nsecs. Fourier transformation of the long term wake in the frequency domain, ordinate:  $2 \cdot k_{\nu}$ -parameter, 0 to  $8 \cdot 10^9$ , abscissa: 0 to 5 GHz.

for 100nsecs and the amplitudes of the fourier transformation were analysed. The wake potential is shown in Figure 5 while Figure 6 shows the fourier transformation of the long range wake. The first 80 modes, which were directly calculated in the frequency domain, can be seen as distinct peaks in the frequency range up to 2.5GHz, while there is also a band of sharp resonances which can be seen at 4GHz.

The wake potential is given by

$$W(s) = \sum_{\nu} 2k_{\nu} \cdot \cos \frac{\omega_{\nu} s}{c} \cdot e^{-(\omega_{\nu}/2Q_{\nu}) \cdot s/c} \text{ for } t > 0 \quad (6)$$

The amplitude normalisation in Figure 6 is chosen so that the ordinate corresponds to the amplitude of an undamped oscillation. Assuming that the damping ( $\exp(-\omega/2Q \cdot s/c)$ ) can be neglected for the calculation range ( $s=0 \dots 30m$ ), the spectral peaks can be taken as  $2 \cdot k_{\nu}$  (V/C). An approximate shunt impedance can be calculated for a mode with a sharp peak in the frequency spectrum. For example the highest peak of the spectrum at 3.9 GHz has an amplitude of  $\approx 7.3 \cdot 10^9$ . The shunt impedance is given by  $2kQ/\omega$  so using  $Q = 5500$  (obtained from the frequency

domain calculations for this band mentioned above) gives a shunt impedance of 1683 Ohm and a power loss of 12 Watt, for the HERA design current of 60mAmp for electrons. This value is of the same order of magnitude as that obtained from the frequency domain calculations and thus one would expect the maximum resonant mode loss to be not greater than 12 Watt per excited mode.

## 6 COMPARISON WITH THE PRESENT CONFIGURATION

Similar calculations were made for a change in the beam tube at the interaction region at the end of 1994, the configuration which is now in place. Modes were then trapped in the region from  $z = -2.24m$  and  $z = 2.5m$ . The Version I calculations showed an increase in single pass impedance of approximately 20% for a bunch length of 8 mm. For Version II these losses have been reduced to a third of the value which was calculated for the Version I configuration with an elliptical beam tube.

## 7 CONCLUSION

The resonant losses are caused by modes which are completely trapped in the part of the beam pipe with the largest cross-section and therefore with the lowest cutoff frequency. The losses of higher spectral components (e.g. above 10GHz) are estimated by the 'single pass power loss'. This quantity describes the power loss of the beam, but only a fraction of this energy is absorbed in the walls of the interaction region. Most of the energy propagates upstream and downstream into the beam pipe and is dissipated over a great length. The intersection of absorbers can be tolerated, if their surfaces are small compared to the pipe cross-section. The following modifications could reduce the transient losses in the superconducting magnets

- A more gradual taper into the keyhole section,
- A reduced vertical dimension in the right hand side section
- A more gradual taper between the sections
- Breaking the x-z symmetry so that fields can couple to lower waveguide modes.

The main problems are likely to occur at injection when the electron or positron bunch is shortest.

## 8 REFERENCES

- [1] The MAFIA Collaboration, CST GmbH, Lauteschlägerstr. 38, 64289 Darmstadt, Germany.
- [2] 'Wake Fields and Impedances', by T.Weiland, R.Wanzenberg, DESY M-91-06, May 1991.
- [3] 'HOM Heating at the PEP-II B-Facility IR Beryllium Vacuum Pipe', X.Lin, C.-K.Ng, K.Ko, SLAC-PUB-7658, Sept 1997.

# DESIGN OF 2×2 DLDS RF COMPONENTS FOR JLC

J.Q. Wang, Y.H. Chin, S. Kazakov<sup>†</sup>, S. Yamaguchi and H. Tsutsui  
KEK, Oho, Tsukuba-shi, Ibaraki-ken, 305-0801, Japan

## Abstract

We have studied a multi-mode Delay Line Distribution System (DLDS)[1][2] as the RF power distribution system from klystrons to RF structures for linear colliders. In particular, a 2×2 DLDS has been proposed and studied at KEK for Japan Linear Collider(JLC). It has been proved that the 2×2 DLDS is simple, but has good transmission efficiency. We have designed RF components of a basic unit of a DLDS using the High Frequency Structure Simulator (HFSS) code[3]. They include the TE<sub>01</sub> extractor, the TE<sub>11</sub> to TE<sub>01</sub> converter, and the TE<sub>11</sub> to TE<sub>12</sub> converter for TE<sub>12</sub> mode. HFSS calculation of the system, which consists of TE<sub>01</sub> extractor and TE<sub>11</sub> to TE<sub>01</sub> converter, shows that the transmission efficiency of each mode is better than 95%. A low power test model for the mode stability experiment in 55m long wave guide in DLDS is also being developed. Further study is underway.

## 1 INTRODUCTION

The Delay Line Distribution System (DLDS) invented by KEK has been considered for the compression and distribution of the RF power from klystrons to accelerator structures in the proposed projects of linear colliders, such as the Japan Linear Collider (JLC)[4] and Next Linear Collider (NLC)[5]. In DLDS, the long pulse of combined klystron output is subdivided into a train of shorter pulses by proper phasing of the sources and each subpulse is delivered to accelerating structures at varying distances from the sources through a delay line distribution system. This system utilizes the delay of the electron beam in the accelerator structure of the linear collider to reduce the length of the waveguide assembly. A conceptual improvement is proposed by SLAC to further reduce the length of waveguide system by multiplexing several low-loss RF modes in a same waveguide. Thus, the subpulse in the distribution waveguide are carried by different waveguide modes so that they can be extracted at designated locations according to their mode patterns. Based on the SLAC multi-mode DLDS, a 2×2 DLDS[6] is proposed in KEK for JLC. The advantage of 2×2 DLDS is that it's simple and easy to be expended to accommodate combinations of more klystrons, and also it has good transmission efficiency.

In this paper, we present the design of the main RF components of a basic unit of 2×2 DLDS. Since the

present schemes of multi-mode DLDS have similar feature to handle the over-moded waveguide system, the components may be of common design. The main components include the mode launcher, the TE<sub>01</sub> extractor, and the TE<sub>11</sub> to TE<sub>01</sub> mode converter. The mode launcher converts the power from four rectangular waveguide feeds to separate modes i.e. TE<sub>01</sub> and TE<sub>11</sub> modes, in a multi-moded circular guide through coupling slots. The TE<sub>01</sub> extractor extracts the TE<sub>01</sub> mode in the circular waveguide but is transparent to other modes. For long distance transmission, the low loss TE<sub>12</sub> mode is preferred, so a TE<sub>11</sub> to TE<sub>12</sub> mode converter is also presented. The High Frequency Structure Simulator (HFSS) code, which evaluates in frequency domain with 3D finite element method, is used to design the above components. A basic system of DLDS that consists of TE<sub>01</sub> mode extractor and the TE<sub>11</sub> to TE<sub>01</sub> mode converter is also studied. The results show that the transmission efficiency in the system for both TE<sub>01</sub> and TE<sub>11</sub> modes are better than 95%.

## 2 TE<sub>01</sub>-TE<sub>11</sub> MULTI-MODE LAUNCHER

For TE<sub>01</sub>-TE<sub>11</sub> multi-mode launcher, we adopt the same design as that proposed by Zenghai Li, et al[7]. The launcher has four rectangular input ports and one cylindrical output port. The input power is TE<sub>10</sub> mode of rectangular waveguide, but the output power can be either TE<sub>01</sub> or TE<sub>11</sub> mode due to the phase coding of the input signals. Fig. 1 is the HFSS 1/4 geometry solid model.

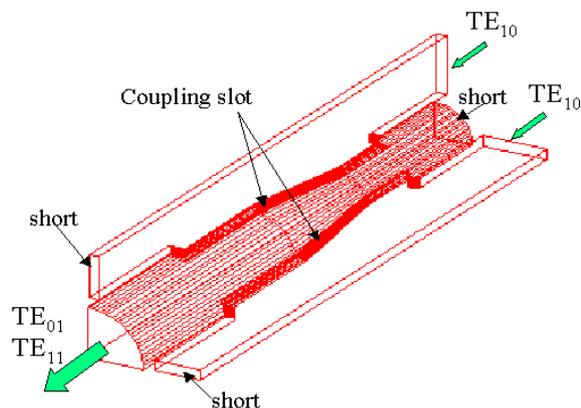


Figure 1: 1/4 geometry solid model of TE<sub>01</sub>-TE<sub>11</sub> mode launcher.

<sup>†</sup> On leave from Branch of Institute of Nuclear Physics, Protvino, Russia.

The four rectangular waveguides run parallelly to the cylindrical waveguide and are spaced  $90^\circ$  apart azimuthally around it. Each of the rectangular waveguide is coupled to the cylindrical waveguide through a single coupling slot. The original design is modeled with MAFIA[8], so HFSS is used to check the performance of the same geometry as the MAFIA model. But the result got by HFSS is not so good as that predicted by MAFIA: though the power transmission efficiency from the rectangular waveguide to circular waveguide for  $TE_{01}$  mode in the can be 97.4%, the efficiency for  $TE_{11}$  mode is only about 90%. We think that the deviation may be caused by the different mesh method between MAFIA and HFSS. By perturbation study [9], we found that it's possible to improve the HFSS result either by increasing the length of the coupling slot on the circular waveguide for 0.35mm or adjusting both short positions in the rectangular and circular waveguide for 0.35mm from the original MAFIA geometry. Then the requirement for high efficiency and low surface field can be satisfied. Table 1 gives the comparison of the results. This confirms that the launcher design has the flexibility to improve the performance practically by adjusting the short position, and suggests that the prototype of the launcher should have the short position tunable. The test result of the prototype will come out soon.

Table 1: Comparison of the transmission efficiency of the mode launcher between MAFIA and HFSS results.

Mode	MAFI A	HFSS		
		Id. M	SI+0.35	Sh+0.35
$TE_{01}$	98.5%	97.4%	99.5%	99%
$TE_{11}$	98.5%	90%	98%	98.7%

Id.M: The identical model as MAFIA  
 SI+0.35: Slot length increases 0.35mm  
 Sh+0.35: Short position adjusts 0.35mm

The present launcher scheme consists of two parts: the  $TE_{21}$  extractor and the  $TE_{01}$ - $TE_{11}$  launcher, so the MAFIA model of the  $TE_{21}$  extractor[7] is also checked with HFSS. The HFSS result coincides with the MAFIA simulation. The  $TE_{21}$  extractor extracts 97.2% of the  $TE_{21}$  mode power and is transparent to other modes, thus meets the requirement.

### 3 $TE_{01}$ EXTRACTOR

The  $TE_{01}$  extractor design is based on the so called wrap-around converter[2]. When  $TE_{01}$  and  $TE_{11}$  modes pass through the extractor, the  $TE_{01}$  is extracted into another parallel waveguide, while the  $TE_{11}$  mode is not affected. The circular waveguide is tapered down to cutoff the  $TE_{01}$  mode while allowing the  $TE_{11}$  to go through, and the parallel one was shorted at one end to control the direction of  $TE_{01}$  mode transmission. Fig. 2 is the HFSS solid model. The rectangular waveguide is warped around the circular waveguide as shown. There are 6 coupling holes spaced  $60^\circ$  apart in the azimuthal direction around the circular wave guide. The size of the coupling

hole is the same as the cross section of the rectangular waveguide. The distance between the center of every two holes is near the wavelength of the  $TE_{10}$  mode in the rectangular waveguide, so that the azimuthal resonant coupling between the two waveguides can be achieved. Thus the  $TE_{01}$  mode in the circular waveguide can be extracted efficiently into the wrap-around rectangular waveguide. Due to reciprocity, the extracted power in the rectangular waveguide can be converted back to  $TE_{01}$  mode in the parallel circular waveguide. The symmetry of the structure prevents the  $TE_{11}$  mode in the circular waveguide being affected.

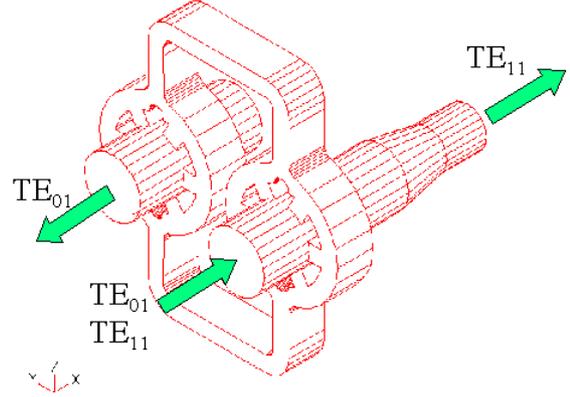


Figure 2: Solid model of  $TE_{01}$  extractor.

The performance of the extractor is mainly decided by the following geometrical parameters: the distance between the coupling holes in the rectangular waveguide, i.e. the bending radius of the wrap-around rectangular waveguide, the distance between the coupling holes on the wall of circular waveguide, i.e. the radius of the circular waveguide, as well as the radius of the round nose at the bifurcation of the rectangular waveguide. The above parameters are optimized with HFSS, and the present geometrical dimensions of the extractor are: the radius of circular waveguide 18.034mm, the tapered down one 12.7mm, the bending radius of rectangular waveguide 26.416mm, and the radius of the nose 4.572mm. The rectangular waveguide is the 22.86mm $\times$ 10.16mm one. The present results show that about 99.4% of the power of  $TE_{01}$  mode can be extracted, 97.2% of  $TE_{11}$  mode goes through without perturbation, while 2.3% converted to  $TM_{01}$  mode. The peak electric surface field at 600MW is around 80MV/m, which locates in the bifurcation area in the rectangular waveguide. Improvement of the design aiming to decrease the peak electric field at high power, and increase the transparency to  $TE_{11}$  mode is underway. In stead of using the cutoff section to extract  $TE_{01}$  mode, the option to cascade two extractor together so as to handle much higher power is being studied, too.

### 4 $TE_{11}$ TO $TE_{01}$ AND $TE_{11}$ TO $TE_{12}$ CONVERTER

In DLDS, the  $TE_{11}$  mode has to be converted back to  $TE_{01}$  mode so that it can be extracted efficiently. The "Serpentine" style structure is adopted[10]. The radius of

the circular waveguide keeps constant, but the axis of the propagation is deformed as a sinusoidal wave shown in Fig. 3(a). The diameter of the waveguide is 20mm, the amplitude of the sinusoidal deformation is 2.33mm, and the one periodical length is 80.4mm. The conversion efficiency can be better than 99.5% when four periods cascaded together. Here, We reproduced the SLAC design[11]. An novel idea has been proposed which can simplify the manufacture, that to cut the waveguide slantingly and then weld them in a right way, as shown in Fig. 3(b). With the slant angle chosen properly, the  $TE_{11}$  to  $TE_{01}$  conversion efficiency can be better than 99%. That can be a hopeful candidate in 2×2 DLDS scheme. A primary design has been confirmed by HFSS. Efforts are still continuing for further improvement in the performance.

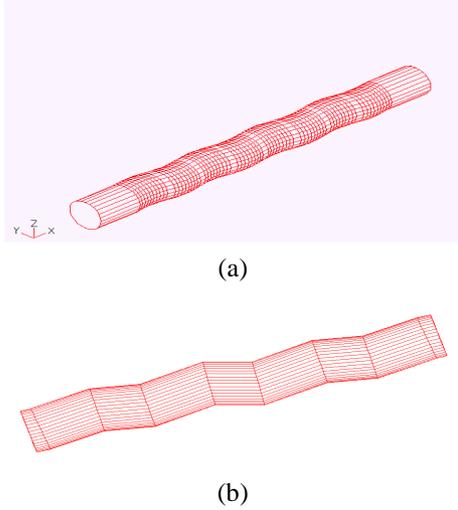


Figure 3: Solid model of  $TE_{11}$  to  $TE_{01}$  mode converters.

In order to reduce the resistive loss in long distance transmission, the  $TE_{11}$  mode are converted to the low loss  $TE_{12}$  mode in the circular waveguide. So  $TE_{11}$  to  $TE_{12}$  converter is needed. The structure with rippled diameters[12] has adopted. The radius of the waveguide varies longitudinally as sinusoidal wave. To avoid the conversion between  $TE_{01}$  and  $TE_{02}$ , the diameter of the waveguide is chosen to vary from 50.2mm to 57.8mm. Nine wave periods are needed. The design procedure is as follows: the S-matrix of one period is calculated by HFSS, then the S-matrix of the nine periods structure is got by cascading the S-matrix of single period given by HFSS. Optimization is done on the length for one period. We found that when the length of one period is 64mm, the converter has the highest efficiency. Then, the whole structure is modeled with HFSS, it's confirmed that more than 99.6% of the power of  $TE_{11}$  mode will be converted to  $TE_{12}$  mode after it goes through the converter. Fig 4. is the HFSS solid model. In HFSS calculation, only 1/4 solid model is needed due to symmetry.

The diameters of waveguides in the above components differ from each other, so tapers are needed to connect them. Tapers are also designed by HFSS. Since

the variation of diameter is not very large, the simple linear transition of waveguide diameter is feasible to achieve transmission efficiency better than 99.5%.

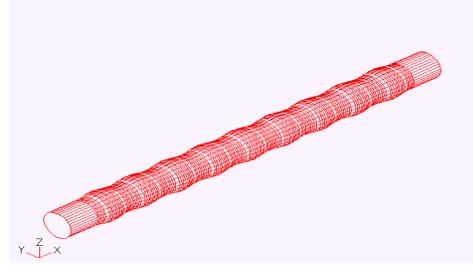


Figure 4: Solid model of the  $TE_{11}$  to  $TE_{12}$  converter.

## 5 TEST UNIT OF 2×2 DLDS

To verify the principle of multi-mode 2×2 DLDS, a test unit is proposed and studied. This unit includes the mode launcher, the  $TE_{01}$  extractor and the  $TE_{11}$  to  $TE_{01}$  converter. Fig. 5 is the schematic layout of the unit.

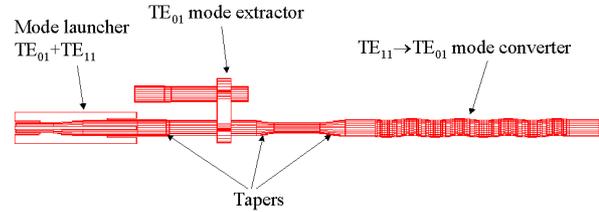


Figure 5: Schematic layout of 2×2 DLDS unit.

Due to the computer memory limit, we only simulate the performance of the system consisting of the  $TE_{01}$  extractor, the  $TE_{11}$  to  $TE_{01}$  converter, as well as the tapers between them. Fig. 6 shows the electric filed pattern propagating in the system.

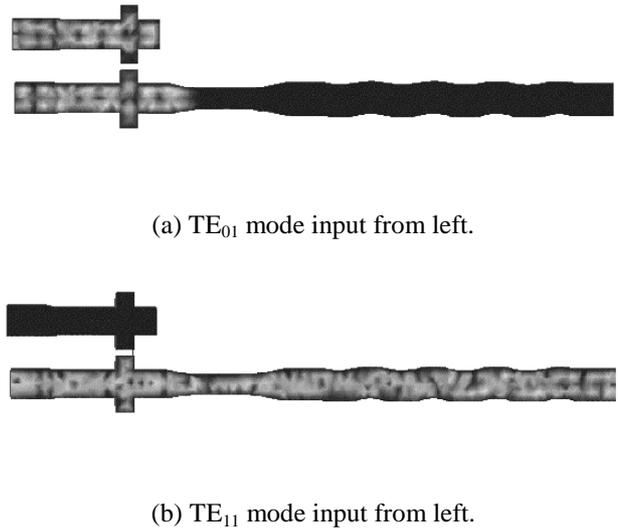


Figure 6: Electrical field patterns in the test unit.

The power of  $TE_{01}$  mode is extracted to a parallel waveguide with efficiency better than 96% as shown in Fig. 6(a), while the power of  $TE_{01}$  mode goes through the extractor directly and then is converted to again to  $TE_{01}$  as shown in Fig. 6(b). The transmission efficiency is better than 95%. With the wall loss taken into account, it's expected that the transmission efficiency in DLDS can be better than 90%. All the above components are being manufactured. The low power test will be done soon. Then the performance of each component, as well as the whole system will be measured.

An absorber is needed at the end of the test unit to absorb the residue power. It's also designed with HFSS. The 1/2 solid model is shown in Fig. 7. A recycled product of SiC is put concentrically in a 20mm circular waveguide. HFSS results show that 99% of incoming power of both  $TE_{01}$  and  $TE_{11}$  modes can be absorbed. To check its sensitivity of performance to the assembly errors, a 1mm displacement of the SiC from the axis of the waveguide is assumed and then simulated, the efficiency of absorption is still higher than 99%.

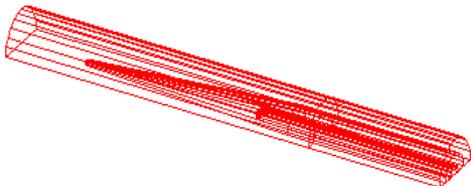


Figure 7: 1/2 solid model of absorber.

The experiment to test the stability of  $TE_{12}$  mode transferring for long distance and its sensitivity to all kinds of error is being planned. The main purpose is to measure the purity of  $TE_{12}$  mode in a 55m long, diameter equal to 4.75 inches waveguide. The experiment needs an  $TE_{11}$  launcher and the  $TE_{11}$ - $TE_{12}$  converter.  $TE_{11}$  mode in circular waveguide can be converted from  $TE_{10}$  mode in rectangular waveguide, the converter, whose main part is the smooth taper from the 22.86×10.16mm rectangular waveguide to the 50.2mm circular waveguide, is designed by HFSS. Fig. 8 is the 1/2 solid model of the launcher. The conversion efficiency is better than 99.2%. The experiment will be done in KEK in near future.

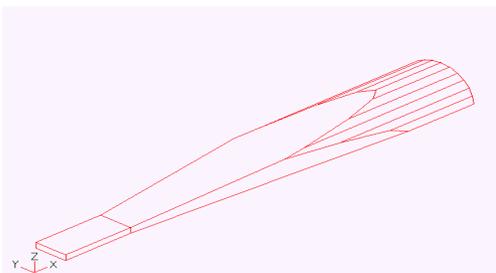


Figure 8: 1/2 solid model of  $TE_{11}$  launcher.

## 6 CONCLUSION

The main components have been designed using HFSS code. According to HFSS result, the power transmission efficiency in the components can at least meet the requirement of DLDS at low power, so the DLDS principle works. A low power test facility will be setup in KEK, and the components will be manufactured according to the HFSS design. The performance of the components as well as the principle of the DLDS will be test experimentally in near future. Meantime, the experiment to study the mode stability after long distance microwave transmission in DLDS is being developed. Improvement on the design of components of DLDS for high power test, or practical use is continuing.

## 7 ACKNOWLEDGEMENT

The author would like to thank Dr. S.G. Tantawi, Dr. Zenghai Li for their kind help.

## REFERENCES

- [1] H. Mizuno and Y. Otake, "A New RF Power Distribution System for X Band Linac Equivalent to an RF Pulse Compression Scheme of Factor  $2^N$ ", LINAC94 Proceedings.
- [2] S.G. Tantawi, et al., "A Multi-Moded RF Delay Line Distribution System for the Next Linear Collider (NLC)", EPAC98 Proceedings.
- [3] HP High-Frequency Structure Simulator 5.0 User's Reference, Sept,1997, Hewlett-packard.
- [4] JLC Design Study Group, "JLC Design Study", KEK Report 97-1, 1997.
- [5] The NLC Design Group, "Zeroth-order Design Report for the Next Linear Collider, SLAC Report 474, 1996.
- [6] Y.H. Chin et al, LCC-005, available at <http://lcdev.kek.jp/JLC/ISG/ISG2.html>.
- [7] Zenghai Li, et al., "Mode Launcher Design for the Multi-moded DLDS", *ibid*.
- [8] MAFIA User's Guide Version 3.20, CST GmbH, Darmstadt, Germany.
- [9] Zenghai Li, private communication.
- [10] Eric Giguet et al, "An original Approach to Mode Converter Optimum Design", IEEE Trans. On Microwave Theory and Tech. , Vol. 46, No.1, Jan. 1998. p. 1-9.
- [11] S. Tantawi, private communication.
- [12] K. Eppley, et al., "A Four port Launcher for a Multimode DLDS Power Delivery System", *ibid*.

# ELECTROMAGNETIC MODELING OF FAST BEAM CHOPPER FOR SNS PROJECT\*

S.S. Kurennoy

Los Alamos National Laboratory, Los Alamos, NM 87545, USA

## Abstract

High current and stringent restrictions on beam losses in the designed linac and storage ring for the Spallation Neutron Source (SNS) require clean and fast – with the rise time from 2% to 98% less than 2.5 ns – beam chopping in the linac front end, at the beam energy 2.5 MeV. The development of new traveling-wave deflecting current structures, based on meander lines, for the SNS fast chopper is discussed. Three-dimensional time-domain computer simulations with MAFIA are used to study transient effects in the chopper and to optimize current structure design.

## 1 SNS CHOPPER SYSTEM

The SNS is a next-generation pulsed spallation neutron source designed to deliver 1 MW of beam power on the target at 60 Hz in its initial stage [1,2]. It will consist of a 1-GeV linear  $H^-$  accelerator and an accumulator ring. The SNS storage ring accumulates the linac beam during a few hundred turns (a macropulse, about 1 ms) using  $H^-$  injection through a carbon foil. The beam injected into the ring is stacked into a single long bunch, and the linac macropulse must be chopped at near the ring revolution frequency 1.188 MHz to provide a gap required for the kicker rise time during a single-turn ring extraction. The final clean beam chopping in the linac is to be done in the Medium Energy Beam Transport (MEBT) line.

The MEBT transports 28 mA of peak beam current from a 2.5-MeV 402.5-MHz RFQ to a drift-tube linac. A 0.5-m space is allocated for the chopper that deflects the beam into a beam stop during the 35% beam-off time. The chopper parameters are summarized in Table 1.

Table 1: MEBT Chopper Specifications

Parameter	Value	Comment
Beam energy	2.5 MeV	$\beta=0.073$
Length	$\leq 0.5$ m	shorter is better
Gap	$\geq 1$ cm	adjustable
Pulser voltage	$\pm 900$ V	currently achievable with FETs
Deflection angle	18 mrad	
Chopping period	841 ns	
Duty factor	35 %	65 % beam on
Rise / fall time	$< 2.5$ ns	2–98 % (final goal)

To mitigate the effects of a partial chopping or small errors in the timing system, an identical “anti-chopper” will be placed in the MEBT line at an optically symmetric point from the chopper to return uncollimated beam to the axis. The preliminary chopping stage in the Low Energy Beam Transport (LEBT) line, at 100 keV, see [2], is introduced to reduce the beam power deposited at the MEBT beam stop. While the LEBT electrostatic chopper system has much slower rise and fall times (tens of ns) than the MEBT one, it is easier to absorb the bulk of the chopped beam at the low LEBT energy.

At any given moment as the beam passes through the MEBT chopper, there are about ten bunches along the chopper length. Even with an “ideal” pulse generator, the only way to avoid partially chopped bunches is to apply a traveling-wave current structure. The deflecting electric-field pulse fills the chopper with the phase velocity along the beam path matching the beam velocity and propagates together with the beam. The bunches following the pulse front are fully deflected while those ahead of the front are not disturbed. Providing the field-pulse front (and its end) shorter than the bunch-to-bunch spacing (2.5 ns, or about 5 cm) is the most challenging requirement to the chopper system. As an initial goal, the rise/fall time below 5 ns is acceptable; it will lead to one partially chopped bunch at the front and the end of each chopper pulse.

## 2 CHOPPER CURRENT STRUCTURE

### 2.1 Coax-to-plate LANSCE chopper

A traveling-wave chopper for  $H^\pm$  beams at 750 keV [3] has been working successfully for many years at LAMPF (now LANSCE) in Los Alamos. It provides the rise time of about 7 ns, mostly due to the pulse modulator. The coax-plate current structure itself is capable of providing a pulse front about 2-3 ns with an overshoot on the 10% level ringing for a few ns. The 1-m long structure consists of two parallel plates, each interfaced with many small strip segments connected with coaxial cables on the reverse side of each plate to form a continuous circuit along the structure, see Fig.1.

The voltages on the upper and lower plates are synchronized and have opposite signs so that the resulting vertical electric field deflects the beam that travels between the plates. The structure rise and fall time limita-

\* Work supported by the US Department of Energy.

tions are caused by stray capacitance between the segments and by multiple coax-to-segment transitions.



Figure 1: Photograph of traveling-wave current structure used in the LANSCE beam chopper.

## 2.2 Meander-line current structure

A new current structure based on a meander line with separators (Fig.2) has been proposed [4]. A strip transmission line forms the meander, which works as a slow-wave structure. The strip itself can be either straight or notched, as the one shown in Fig.2.

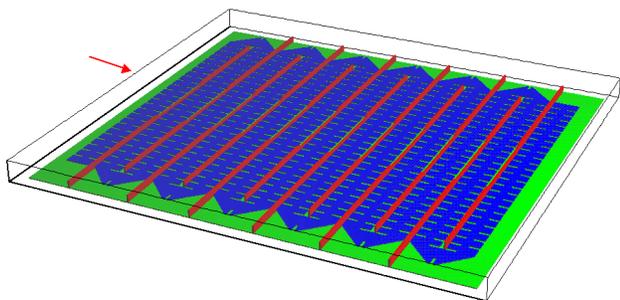


Figure 2: 1/4-length MAFIA model of meander current structure: notched meander strip line (blue) above the ground plate (green) with separators (red), cf. Fig.3. Only the lower plate is shown. The arrow indicates the beam path in the upper plane of the box drawn.

The line parameters are adjusted to provide the line characteristic impedance  $50 \Omega$ . The meander bends are chamfered to avoid pulse reflections. The separators (or guard barriers) rising from and electrically connected to the ground plane are used to reduce the coupling between the adjacent sections of the meander line, see in Fig.3. The new design has no multiple coax-to-plate transitions and is easier for manufacturing.

A proper ratio of the meander period to its width provides the required phase velocity of the voltage pulse,  $v = \beta c$ , along the beam direction. For a straight strip, the meander width transverse to the beam is simply  $b = (w+g)(1/\beta-1)$ , where  $w$  is the strip width and  $g$  is the gap width between the adjacent strips.

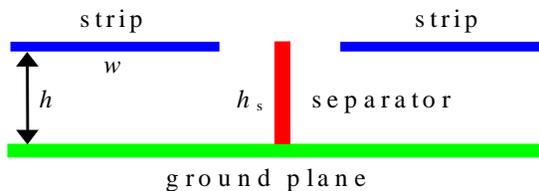


Figure 3: A partial vertical cut in the beam path plane of the meander current structure: the separator is inserted between two adjacent pieces of the transmission line.

3-D time-domain modeling with the electromagnetic simulator MAFIA [5] has been used to study transient effects in the current structure. The simulations [4] used MAFIA version 3.20 package. Essentially, it was done similar to S-parameter calculations: the strip TEM wave was loaded into the structure, and the voltages and the electric fields on the beam path were recorded as it propagated along the chopper. This approach required additional calculations of the 2-D eigenmodes at the entrance and exit ports. And these ports themselves had to be artificially introduced to simulate waveguides connected to the structure. The time dependence of the TEM-wave amplitude can be chosen as required. We used either harmonic signals at a fixed frequency to analyze frequency response, or step-function pulses to study the rise and fall times. In the last case the step was smoothed by  $\sin^2$  for about 0.1 ns to filter out very high frequencies. The same smoothing was applied to finite-length pulses.

It was shown [4] that even without separators, the meander structure has a rise time 2–2.5 ns; with separators it can be reduced down to 1–1.25 ns, depending on the separator height, see Fig.4. The structure fall time was found to be about the same as the rise time. High separators, however, reduce the effective field on the beam path by 10–20% depending on their height  $h_s$ , cf. Fig.4.

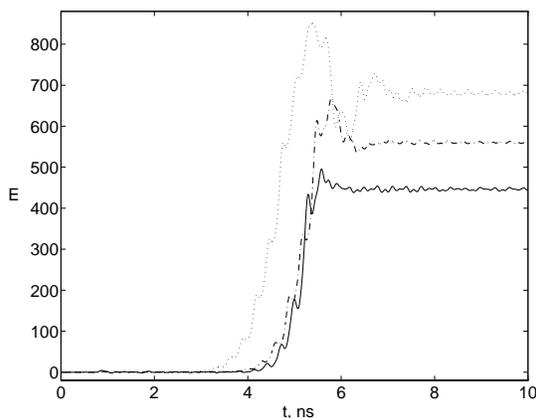


Figure 4: Vertical electric field, in arbitrary units, at a fixed point on the beam path versus time in the half-length meander structure with separators  $h_s = 2h$  (solid),  $h_s = h$  (dash-dotted), and without them (dotted). The voltage amplitude is the same for all cases.

### 2.3 Simple prototype test

To check our simulation results, a very simple model structure with a straight-strip meander line on a printed-circuit board (PCB) has been manufactured at LANL. The PCB with the meander on its bottom surface was placed on 1.25-mm barriers-separators sticking out of the aluminum ground plane, see Fig.5. The prototype length is 12.5 cm. Obviously, the presence of the dielectric introduces some additional dispersion, and it will stretch the voltage pulse front and end. Nevertheless, this prototype was chosen because of manufacturing simplicity.

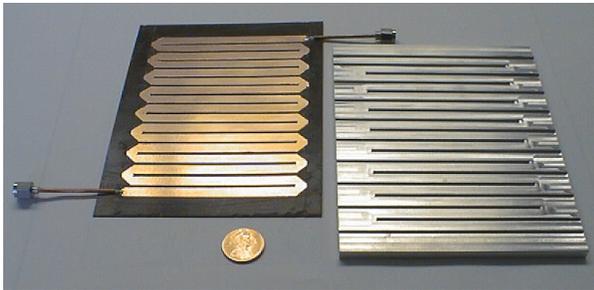


Figure 5: Photograph of the prototype structure: straight-strip meander line on PCB (left) and ground plate with separators (right).

Corresponding MAFIA time-domain simulations have been performed for the prototype structure taking into account the dielectric properties of the PCB material. The transient effects in the prototype have been measured with a TDR and were found to agree well with the MAFIA calculations, with all transitions within a 2-ns range, see Fig.6. Some small differences in the pulse shape in two pictures are due to stronger high-frequency components (tens of GHz) in calculation results. Clearly, these very high frequencies have been filtered out in the experimental setup by wires and connections.

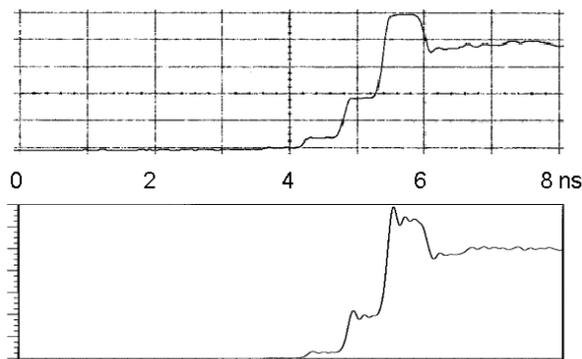


Figure 6: The voltage pulse front in the prototype structure: TDR measurements (top) and MAFIA simulations (bottom).

### 2.4 Notched-strip meander line

Our early work [4] dealt mostly with straight-strip meander structures. Using a notched strip line in the meander instead of a straight one has some advantages. First, the notches provide an additional inductive load that slows down the wave along the strip. A notch of depth  $d$  and width  $a \leq h$  adds the inductance  $L_1 = (\pi/2)\mu_0 h d^2 / w^2$  [6]. In a first approximation, the capacitance per unit length  $C'$  remains the same as for the straight strip, i.e.  $C' \approx C'_0 = \epsilon_0 w / h$ , while the inductance increases from  $L'_0 = \mu_0 h / w$  to  $L' = \mu_0 h / w + 2L_1 / p$ , where  $p$  is the notch spacing period. Then we estimate the phase velocity  $v_{ph} = 1 / (L' C')^{1/2}$  and adjust the impedance  $Z = (L' / C')^{1/2}$  to be  $50 \Omega$ . The ratio  $v_{ph} / c \approx (L'_0 / L')^{1/2}$  depends on the notch depth and the number of notches per unit length. For example, the TEM wave propagates along the notched strip shown in Fig.2 and Fig.7 with the phase velocity about  $0.75c$ .

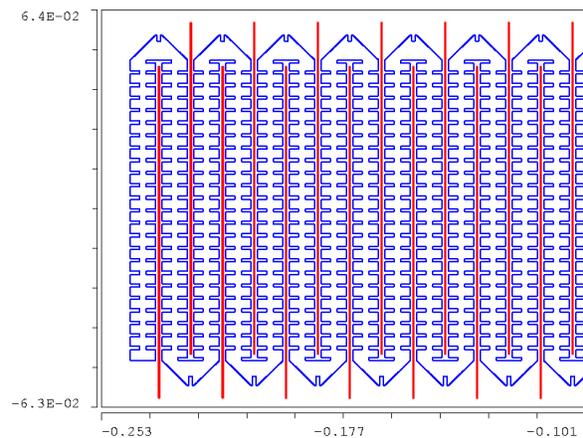


Figure 7: One quarter of the full-length MAFIA model for the notched-strip meander current structure with separators (red lines): 2-D cross section in the strip plane, cf. Fig.2. All dimensions are in meters.

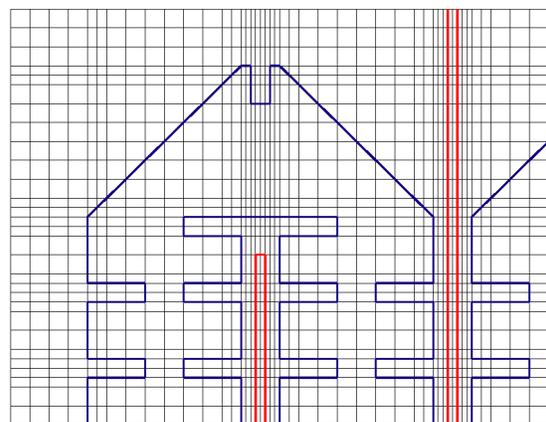


Figure 8: Details of Fig.7 (upper left corner) with mesh lines shown.

As a result, we need a smaller number of bends in the notched-strip meander and can use wider strips for a fixed meander width. The larger ratio of the strip width  $w$  to the strip-to-strip gap width  $g$  increases the structure field efficiency. For the meander in Fig.7, the notched-strip width  $w$  is 8 mm compared to 5 mm for a straight line. The meander width transverse to the beam is about 11 cm,  $h=1$  mm and  $g=2$  mm in both cases. In addition, the notches also reduce the magnetic coupling between adjacent strips since the wave magnetic field is concentrated closer to the strip center. Our recent efforts [7,8] have been directed toward optimizing the notched-strip design.

In modeling the full-length structure, the detailed mesh – like the one in Fig.8 – included up to 3 millions mesh points. Most of our MAFIA computations have been performed on SUN workstations Ultra-1 and Ultra-2. In earlier simulations [4] with MAFIA 3.20 we needed to load a pre-calculated TEM-mode into the structure. The MAFIA version 4 [5] allows simply to feed the strip with a voltage from a filament with some RLC-parameters, which connects the strip to the ground plate. The voltage can be an arbitrary function of time. We have used voltages profiled as either step-functions (smoothed by  $\sin^2$  to filter out very high frequencies) or as finite-length pulses. As the voltage pulse propagates along the structure, the electric field on the beam path is recorded. As an example, Fig.9 shows the deflecting field created by a voltage pulse with 1-ns  $\sin^2$  front, flat top at 1 kV for 3 ns, and 1-ns  $\sin^2$  end, in the full-length 50-cm model of the type shown in Figs. 2, 7 and 8. Such a pulse would kick out exactly two linac bunches.

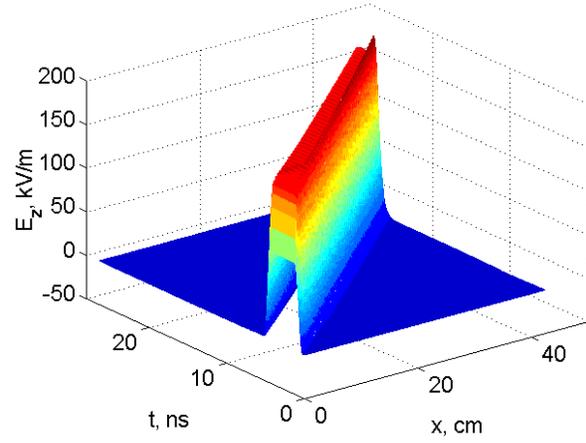


Figure 9: Deflecting field on the beam path versus time and position in the notched-strip meander structure for a 1-3-1-ns 1-kV driving pulse.

As the pulse propagates, its shape is slightly distorted by developing an overshoot. However, the pulse front and end both remain well within 2-ns range, see also Fig.10.

Cross-sections of the surface plot in Fig.9 for a given position  $x$  along the structure show the time dependence of the field at this location, see Fig.10, left. The development of the overshoot is clearly visible for the pulse in the middle and near the structure end. Straight-strip meanders produce slightly larger pulse distortions [7], as one can see in Fig.10, right. The same driving voltage pulse was used for simulations in both cases. Comparison of the pulse amplitudes in Figs.10 (also Figs. 11) confirms that the notched-strip meander provides slightly higher field efficiency than the straight-strip one.

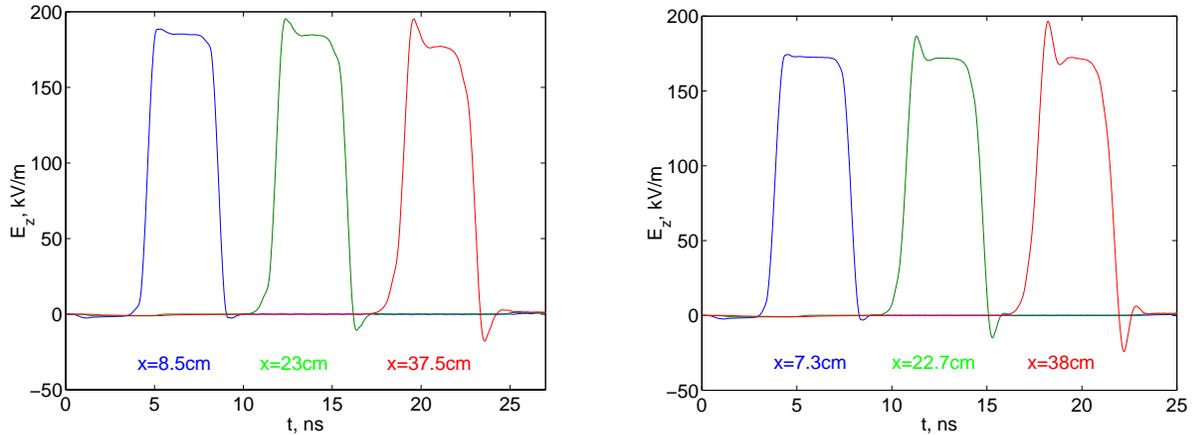


Figure 10: Deflecting field versus time in 3 different points on the beam path in the notched-strip (left) and straight-strip (right) meander structures.

Cross-sections of Fig.9 taken at given time  $t$  produce snapshots of the deflecting field as shown in Fig.11, left. Small wiggles on the pulse tops are due to differences of the field in points above the middle of the strip and above the separators. In fact, these field variations will even

help to spread the deflected beam slightly on the beam stop. A similar picture for the straight-strip meander is plotted in Fig.11, on the right side. The number of wiggles in this case is larger because the structure has a

shorter period along the beam axis,  $w+g=7\text{mm}$ , compared to 10 mm for the notched-strip meander.

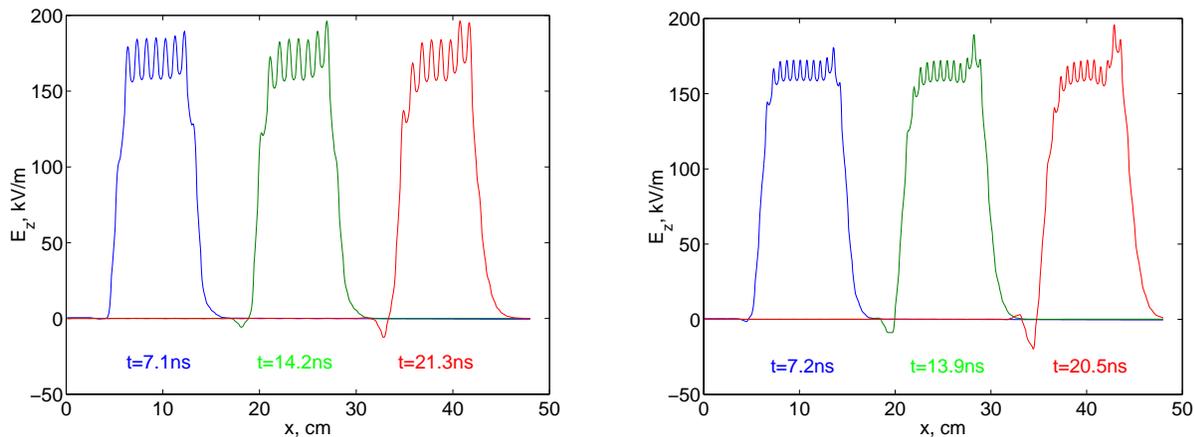


Figure 11: Snapshots of the deflecting field on the beam path in the notched-strip meander structure (left), cf. Fig.9, and in the straight-strip meander structure (right).

While Fig.11 presents only three “snapshots” of the field pulse shape, it is relatively easy to produce a movie showing how the pulse propagates along the structure. It can be done by recording a large number (a few hundred) of frames – similar to those in Fig.11 – separated by short time intervals. The MAFIA postprocessor provides such an option for video recording; however, the corresponding movie file is extremely large. We found a more convenient way to keep this information: recording the MAFIA movie from the screen (we use an X-windows emulator on PC) with a simple PC utility Hypercam produces a very compact avi-file, which can be played later with any media player.

Our design optimization [7,8] leads to the notched-meander structure shown in Figs.2, 7 and 8. Starting from initial analytical estimates for the geometrical parameters, we used MAFIA simulations to adjust the line characteristic impedance and the pulse phase velocity along the beam. The notches are 3-mm deep, 1-mm wide, and spaced by 4 mm. Additional 2-mm deep notches on the bends have been introduced to eliminate pulse reflections. To increase the effective field on the beam path without increasing the rise time, the profiled separators have been introduced: they are low (i.e., flush with the strip line) in the middle, near the beam axis, and higher near the sides, see Figs.2-3.

### 3 CONCLUSIONS

A new current structure based on a meander line has been developed. The 3-D time-domain modeling shows that the structure is capable to provide the rise and fall times on the order of 1 ns. Further simulations will include more engineering details like mechanical supports, as well as beam dynamics and PIC-simulations. Manufacturing of the full-length prototypes and their measurements are also planned.

The voltage generator development for this fast chopper remains an important and challenging issue. We will proceed with the proof-of-principle pulser design using currently available technology, while continuing to work with manufacturers on development of faster powerful FETs.

### 4 ACKNOWLEDGEMENTS

The author would like to thank John Power, who performed measurements of the prototype and contributed a lot to its manufacturing. Thanks to Glen Lambertson for suggesting the use of a notched strip in the transmission line. Useful discussions with Andy Jason and Frank Krawczyk are gratefully acknowledged.

### REFERENCES

- [1] B.R. Appleton et al., *Proceed. of EPAC (Barcelona, 1996)*, p.575. – IOP Publishing, Bristol (1996).
- [2] NSNS Collaboration, “NSNS Conceptual Design Report.”– NSNS/CDR-2/V1, Oak Ridge, TN (1997).
- [3] J.S. Lundsford and R.A. Hardekopf, *IEEE Trans. NS*, **NS-30**, 2830 (1983).
- [4] S.S. Kurennoy, A.J. Jason, F.L. Krawczyk, and J.F. Power, in *Proceed. PAC97 (Vancouver, BC, 1997)*.
- [5] *MAFIA release 4.00*. – CST, Darmstadt, 1997.
- [6] W.J.R. Hoefer, *IEEE Trans. MTT*, **MTT-25**, 822 (1977).
- [7] S.S. Kurennoy, “Beam Chopper System for SNS”, Report LA-CP-98-156, Los Alamos (1998).
- [8] S.S. Kurennoy and J.F. Power, in *Proceed. LINAC98 (Chicago, 1998)*.

# Progress in Parallelizing XOOPIC

Peter J. Mardahl, John P. Verboncoeur  
Cory Hall #173, EECS Department  
University of California  
Berkeley, CA 94720-1772, USA

*Abstract*

XOOPIC [1] (Object Oriented Particle in Cell code for X11-based Unix computers) is presently a serial 2d 3v particle-in-cell plasma simulation. This effort focuses on using parallel and distributed processing to optimize the simulation for large problems. The benefits include increased capacity for memory intensive problems, and improved performance for processor-intensive problems.

The MPI library enables the parallel version to be easily ported to massively parallel, SMP, and distributed computers. The philosophy employed here is to spatially decompose the system into computational regions separated by “virtual boundaries”, objects which contain the local data and algorithms to perform the local field solve and particle communication between regions. This implementation reduces the impact of the parallel extension on the balance of the code.

Specific implementation details such as the hiding of communication latency behind local computation will also be discussed, as well as code features and capabilities.

## 1 GOALS FOR PARALLEL XOOPIC

XOOPIC has been successful as a single-processor code, and is able to simulate many interesting devices including relativistic klystron oscillators, electron guns, DC discharges with gas chemistry, plasma display panel cells, and highly relativistic beams in accelerators. However, particle-in-cell simulations are very computationally intensive, and on a single processor, some problems may take months to complete. The goals, therefore, for parallel XOOPIC are:

- Reduce run-times for large, complex simulations from weeks to days.
- Distribute memory demands across machines, allowing larger simulations than possible otherwise.
- Cross platform portability (networks of workstations, massively parallel machines, and SMP machines).
- Identical usage and feature set for parallel and non-parallel versions of XOOPIC, and largely shared source code.
- Complete source code availability to the general public.

## 2 PARALLELIZATION STRATEGY

The strategy for parallelization of XOOPIC is a coarse-grained spatial decomposition of the physical model into computational regions, as shown in Figure 1. Each computational region has its own mathematical mesh and particle arrays. A coarse-grained partitioning as shown has advantages over other partitioning strategies: in order to update the electromagnetic fields on the mesh points, it is necessary to know the fields on neighboring mesh points. If the neighboring mesh points are on other CPUs, communication is required between CPUs, and typically, communication is slower than computation on a parallel machine. In a block-cyclic decomposition, for example, many more mesh points would have to have non-local data communicated in order to update fields, than would with a coarse-grained partitioning.

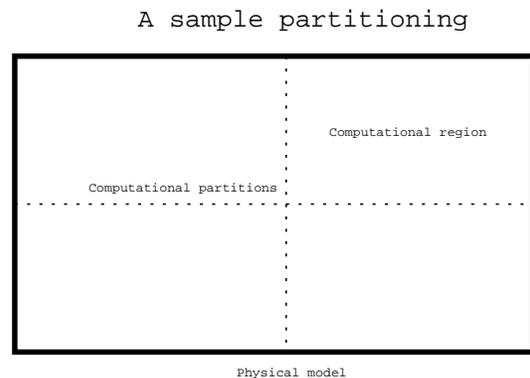


Figure 1: Example of partitioning a model into computational regions.

Another advantage of coarse-grained partitioning is reuse of code. Boundaries of the computational regions are treated as boundary conditions. Therefore, the special case of updating the fields on a computational boundary may be encapsulated in a new boundary condition called SpatialRegionBoundaries (SRBs), minimizing the number of modifications to the existing, already tested non-parallel version of XOOPIC.

Coarse-grained partitioning also allows ready identification of mesh points with only local dependencies: i.e., mesh points which require no data from remote CPUs to update. These “interior” mesh points may be updated while data from other CPUs is transmitted, allowing useful work to be done while messages are in transit. This is an important optimization to perform if parallel resources are to be used efficiently, and has been done in parallel XOOPIC, as shown in Figure 2.

Slow implementation    Faster implementation

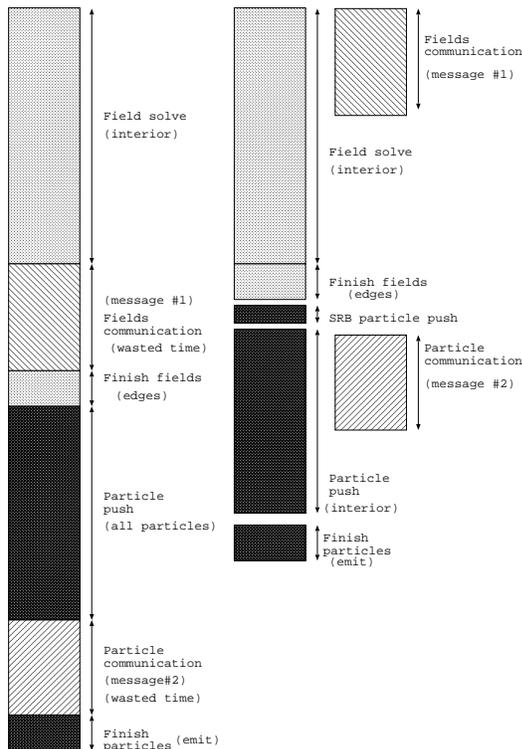


Figure 2: Hiding communication time behind local computation.

Particles also require field data in order to update their positions. Communicating field data every time a particle needed it would be a fatal mistake if performance were a consideration. However, particles only need field data from adjacent mesh points, so if particles are assigned to the same CPU on which the fields it needs reside, communication is not required to update their positions. The exception to this is when particles cross from one region to another: in this case, the particle data is communicated to the destination CPU. In XOOPIC, particles are distributed this way, so that the fields are nearly always local, but it is not possible to easily identify which particles will cross an SRB. This is unfortunate, because it makes another optimization difficult: if crossing particles could be identified early enough, they could be moved first, communicated to their destination, and while the data was in transit, particles needing only local fields could be updated (see *Faster implementation*, Fig. 2).

XOOPIC at present uses the *Faster implementation* for the fields, and the *Slower implementation* for the particles.

Coarse-grained partitioning strategies lend themselves to the use of parallel libraries such as MPI [2] (Message Passing Interface) or PVM (Parallel Virtual Machine). The MPI library in particular is widely portable and gives good performance, and is the library used for parallel XOOPIC. Other tools for parallel programming exist, such as split-C and High Performance Fortran, but these require special

compilers which are often not freely available on platforms of interest. Use of the split-C and HPF compilers would also require extensive modifications to the XOOPIC code.

### 3 SPATIAL REGION BOUNDARIES

As remarked above, much of the work in parallelizing XOOPIC is encapsulated in a special boundary condition: the Spatial Region Boundary (SRB). Figure 3 shows the fields on the mesh near and on an SRB. SRBs are created in linked pairs, with each virtual boundary (a boundary defining a computational region, not a physical boundary) requiring two SRBs, one per computational region. Each SRB is responsible for sending and receiving the necessary fields and passing and receiving particles which cross them.

In order to compute the field components correctly on the SRB (the fields in the dotted box), the field components external to the region must be communicated to it (the fields  $lE1$ ,  $lE2$ ,  $lE3$ ,  $lB1$ ,  $lB3$ , and the currents,  $J1$  and  $J3$ , which are in the same locations as  $E1$  and  $E3$  respectively).

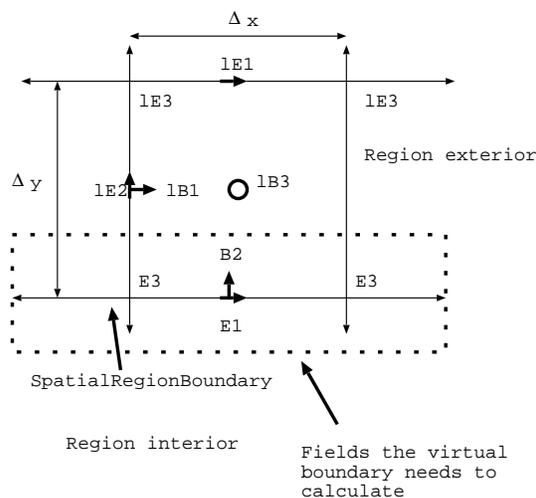


Figure 3: Fields on and near a horizontal Spatial Region Boundary.

The field solve on the boundary is actually a three-step process:  $lE1$ ,  $lE2$ , and  $lE3$  from the region exterior are used to update  $lB1$  and  $lB3$ , which are stored in the SRB.  $E1$  and  $E3$  on the SRB are then computed using the updated ghost values of  $lB1$  and  $lB3$ .  $B2$  on the SRB is updated using the updated  $E3$ s on the SRB by the interior field solve. It is only necessary for the SRB to set  $E3$  properly. These are the computations performed by the SRB shown in Figure 3:

$$lB1^{t+\frac{1}{2}} = lB1 - \frac{\Delta t}{2\Delta y}(lE3 - E3) \quad (1)$$

$$lB3^{t+\frac{1}{2}} = lB3 - \frac{\Delta t}{2} \left( \frac{E1 - lE1}{\Delta y} + \frac{lE2_{j+1} - lE2}{\Delta x} \right) \quad (2)$$



## Particle Pushes per Second vs. #CPUs

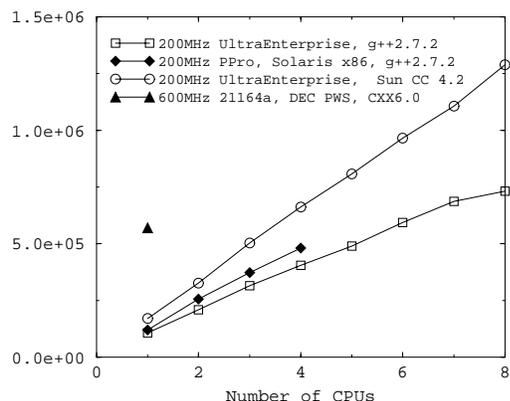


Figure 5: Parallel XOOPIC displays near-linear scaling with number of CPUs.

- No electrostatic models work.
- Dielectric triangle objects are not split correctly across regions.
- Beam emitter boundaries cannot be split by virtual boundaries.
- Particles see nearest-grid-point B3 near SRBs rather than linearly weighted B3.

## 5 FUTURE WORK

Efforts to extend parallel XOOPIC are ongoing. Areas being worked on include:

- Fixing identified bugs.
- Checkpointing of simulations to allow recovery of data should XOOPIC be interrupted.
- Improved diagnostics.
- A parallel Poisson solve, so that electrostatic models may be treated.
- Dynamic load balancing, for best use of parallel resources.
- Elimination of the need to wait for the particle update to complete before identifying which particles must be communicated.
- Extension of XOOPIC to 3-dimensional models.

Checkpointing of simulations is an important capability. Parallel machines have a statistically increased probability of hardware failures, simply because there are more components which may fail. Having the simulation preserve its state to disk may allow recovery of most computation: only the computation done since the last checkpoint would be lost.

Implementation of the parallel Poisson solve is under way, using the the PETSc library [3] with BlockSolve95 [4]. This will allow parallel electrostatic models as well as electromagnetic models. Completion of the parallel Poisson solve is anticipated by early 1999.

Elimination of the need for the particle update to complete and dynamic load balancing are performance optimizations which will extend the usefulness of parallel XOOPIC. Presently, scalability to many CPUs is limited: all computation must wait for the particles to be transferred between SRB pairs, and no dynamic load balancing is done. Efficient use of parallel resources thus requires some planning on the part of the user at present: the type of auto partitioning used can ensure good load balancing, and for some models, reasonable load balancing may not be possible. Dynamic load balancing will improve matters by distributing workload at runtime.

## 6 ACKNOWLEDGMENTS

This work has been funded by AFOSR F49620-94-1-0387, and AFOSR/MURI F49620-95-1-0253. We are also grateful to Jim Havranek (formerly at AFRL) and Gerry Sasser (AFRL) for their useful comments and suggestions.

## 7 REFERENCES

- [1] J.P. Verboncoeur, A.B. Langdon, and N.T. Gladd, "An object-oriented electromagnetic PIC code." *Computer Physics Communications* 87 (1995) 199-211.
- [2] "The Message Passing Interface (MPI) Standard", <http://www.mcs.anl.gov/mpi/>
- [3] "The PETSc Library", <http://acts.nersc.gov/petsc/main.html>
- [4] "BlockSolve95", <http://www.mcs.anl.gov/mpi/>

# Thermal-Photon and Residual-Gas Scattering in the NLC Beam Delivery

I. Reichel, F. Zimmermann, P. Tenenbaum, T.O. Raubenheimer\*  
Stanford Linear Accelerator Center, Stanford University, California 94309

*Abstract*

Without collisions, the largest contribution to the beam lifetime in LEP is Compton scattering off thermal photons. Even if only a few particles are scattered in a single pass, the potential background generated could make this effect important for the NLC as well. We used a modified version of the tracking program DIMAD, which includes a Monte Carlo simulation for the Compton scattering on thermal photons, to calculate the fraction of scattered particles that are intercepted by downstream aperture restrictions and to determine the loss locations. We also studied particle losses due to other scattering processes. For all processes, the effect of additional collimators in the final focus was investigated.

## 1 INTRODUCTION

Without collisions, the largest contribution to the beam lifetime in LEP is Compton scattering off thermal photons. The effect was predicted by Telnov [1]. The backscattered photons were measured [2], they were observed also at HERA [3]. The beam lifetime measured at LEP is in good agreement with the simulated effect of scattering on thermal photons. The typical energy loss induced by this 'inverse' Compton scattering increases in proportion to the beam energy. Even if only a few particles are scattered in a single pass, the potential background generated could make this effect important for the NLC. Assuming a typical photon energy, in the Zeroth Order Design Report (ZDR) [4] we estimated that about 36 particles per bunch train would scatter over a distance of 2 km, the length of the final focus. If this number is approximately correct, the Compton scattering on thermal photons could become a significant (and unavoidable) background source for the NLD detector.

We give the basic formulae describing the thermal-photon scattering and improve our previous crude estimate of the effect, by numerically integrating over the actual photon energy and angular distributions. We then use a version of the tracking program DIMAD [5], which was modified at CERN [6] and SLAC, and which includes a Monte Carlo simulation for the Compton scattering on thermal photons [7], to calculate the fraction of scattered particles that is lost and to determine the loss locations.

In Section 3, we employ the same version of DIMAD to study particle losses caused by two other scattering processes, namely by elastic and inelastic beam-gas scattering. Also here we discuss the total number of lost particles and

their distribution. The effect of elastic scattering on atomic electrons is also estimated.

## 2 SCATTERING ON THERMAL PHOTONS

The total cross section [1] for Compton scattering of a beam electron and a photon is

$$\sigma = \frac{2\pi r_e^2}{x} \left[ \left( 1 - \frac{4}{x} - \frac{8}{x^2} \right) \ln(x+1) + \frac{1}{2} + \frac{8}{x} - \frac{1}{2(x+1)^2} \right] \quad (1)$$

$$\text{where } x = \frac{4E\omega_0 \cos^2(\alpha/2)}{m^2 c^4} \quad (2)$$

$$\approx 15.3 \left[ \frac{E}{\text{TeV}} \right] \left[ \frac{\omega_0}{\text{eV}} \right] \cos^2(\alpha/2)$$

with  $\alpha$  the incident photon angle with respect to the beam in the laboratory system,  $E$  the beam energy, and  $\omega_0$  the incident photon energy. For  $x \ll 1$  the total cross section  $\sigma$  is roughly equal to the Thomson cross section  $\sigma_0 = 8\pi r_e^2/3 = 6.65 \times 10^{-25} \text{ cm}^2$ .

The energy spectrum of the scattered photons (and hence the energy loss of the electrons) is given by [1]

$$\frac{1}{\sigma} \frac{d\sigma}{dy} = \frac{2\sigma_0}{x\sigma} \left[ \frac{1}{1-y} + 1 - y - 4r(1-r) \right] \quad (3)$$

where  $y = \omega'/E$  is the relative energy of the scattered photon, and  $r = y/(x(1-y)) \leq 1$ .

The energy loss spectrum extends between 0 and a maximum value  $y_{max}$

$$0 \leq y \leq y_{max} = \frac{x}{1+x} \quad (4)$$

Only if  $x$  is sufficiently large can the scattered particles lose enough energy to hit some aperture. This becomes more likely the higher the beam energy and the higher the temperature of the vacuum chamber.

In Fig. 1 the total Compton scattering cross section and the maximum relative energy loss per scattering event are depicted as a function of the beam energy. In the following we always assume a beam energy of 500 GeV, unless noted otherwise.

The density and energy distribution of the thermal photons is given by the Planck formula:

$$dn_\gamma = \frac{\omega_0^2 d\omega_0}{\pi^2 c^3 \hbar^3 (e^{\omega_0/kT} - 1)} \quad (5)$$

and the total number of photons per  $\text{cm}^3$  is  $n_\gamma \approx 20.2 T^3 \text{ cm}^{-3}$  [1] with an average photon energy of  $\bar{\omega}_0 \approx 2.7kT$ . At 300 K, this is about 70 meV.

\* Work supported by the U.S. Department of Energy under the contract DE-AC03-76SF00515.

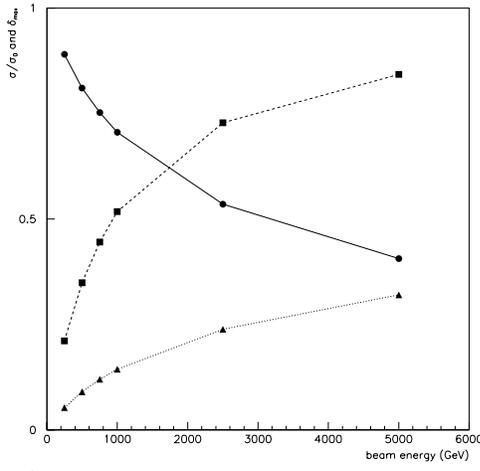


Figure 1: Total Compton cross section in units of the Thomson cross section (—, ●) the maximum relative energy loss for head-on collision with a 0.07 eV thermal photon (- - -, squares), and the average relative energy loss (· · ·, triangles), all as a function of the beam energy. The total cross section and the average relative energy loss were obtained by a Monte-Carlo simulation generating  $10^5$  scattering events.

We consider a beam of  $N$  particles propagating through a distance  $L$ . The energy distribution of the scattered photons is obtained by integrating the product of cross section and Planck spectrum over the solid angle and over all photon energies [1]:

$$\frac{dN}{dy} = NL \int_0^\pi \int_{\omega_{min}}^\infty \frac{d\sigma_c}{dy} (1 + \cos \alpha) dn_\gamma(\omega_0, T) d\omega_0 \frac{d\Omega}{4\pi} \quad (6)$$

where the factor  $(1 + \cos \alpha)$  accounts for the relative motion of electron and photon.

The minimum photon energy is a function of  $y$ ,

$$\omega_{min} = \frac{y m_e^2 c^4}{4(1-y)E \cos^2 \alpha/2} \quad (7)$$

and, to integrate Eq. (6), one must express  $x$  in terms of  $\omega_0$  and  $\alpha$  using Eq. (3).

Figure 2 shows the energy loss spectrum,  $dN/dy$  or  $dN/d\delta$  (where  $\delta = y$  is the relative energy loss of the electron), for  $N = 10^{12}$  particles, a total length  $L = 5$  km, and three different beam energies. By integrating  $dN/d\delta$  over  $\delta$ , we estimate a scattering probability of about  $2.7 \times 10^{-14} \text{ m}^{-1}$  per particle, so that that about 100 particles will be scattered per bunch train. A significant fraction of these will suffer an energy loss large enough to hit some downstream aperture.

Figure 3 compares the energy-loss spectra for thermal-photon scattering with that for inelastic beam-gas scattering, assuming 10 nTorr of CO, both obtained from a Monte-Carlo simulation.

To gain more insight, we have performed simulations using a special version of DIMAD developed for LEP [6]: We tracked a few thousand particles through the NLC beam delivery system, with an optics as depicted in Figs. 4 and 5.

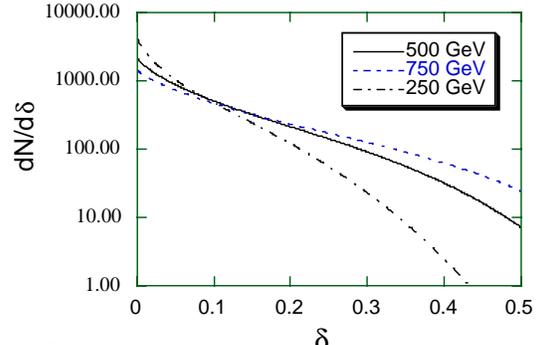


Figure 2: Energy loss spectrum due to Compton scattering off thermal photons. The curves were obtained by numerical integration of Eq. (6). A beam pipe at room temperature was assumed.

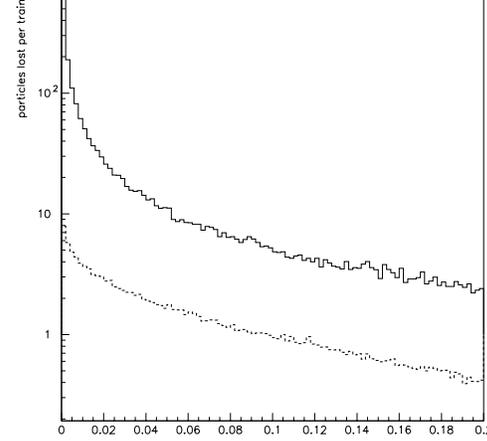


Figure 3: Energy-loss spectrum due to thermal-photon scattering (- - -) and inelastic beam-gas scattering (—).

The nominal number of NLC collimators was used: 4 horizontal plus 2 vertical spoilers, and 8 horizontal plus 6 vertical absorbers. They are adjusted to collimate at  $5\sigma_x$ ,  $36\sigma_y$ , and 4% in energy. Table 1 lists more details, including the locations of spoilers and absorbers in the collimation system (the first 2.5 km of the beam line). The collimation parameters chosen agree with the prescription of the ZDR. In order to explore the effect of additional collimation in the final focus proper, we consider 6 optional absorbers, two each for horizontal, vertical and energy collimation. Table 1 lists parameters of these final-focus collimators. Their locations are indicated in Figs. 4 and 5. For the three different scattering processes (thermal photon scattering, elastic and inelastic beam-gas scattering), we compare the particle losses obtained without any collimation in the final-focus, with all 6 final-focus collimators, and with only 4 transverse final-focus collimators.

The DIMAD simulations predict about  $117 \pm 2$  scattering events per bunch train and 5 km distance. For 500 GeV beam energy,  $51 \pm 1$  (44%) of these particles are lost somewhere along the beam line. The loss distribution as a function of longitudinal position  $s$  is shown in Fig. 6. As can be seen by comparing with Fig. 5, most of the scattered particles are lost in the high dispersion regions of the collimation section and in the horizontal chromatic correction section (CCX) of the final focus. On average only

collimator type	coll. depth	half gap [mm]	location $s$ [m]
collimators in the collimation system			
IP 1, hor. spoil. (Ti)	$5 \sigma_x, 4\%$	2	383, 603
IP 1, vert. spoil. (Ti)	$36 \sigma_y$	1.8	441
IP 1, hor. abs. (Cu)	$5 \sigma_x, 4\%$	2	499, 719
IP 1, vert. abs. (Cu)	$36 \sigma_y$	0.75	509
IP 1, vert. abs. (W)	$36 \sigma_y$	1.8	661
FD 1, hor. spoil. (Ti)	$5 \sigma_x, 4\%$	2	910, 1130
FD 1, vert. spoil. (Ti)	$36 \sigma_y$	1.8	968
FD 1, hor. abs. (Cu)	$5 \sigma_x, 4\%$	2	1026, 1246
FD 1, vert. abs. (Cu)	$36 \sigma_y$	0.75	1036
FD 1, vert. abs. (W)	$36 \sigma_y$	1.8	1188
IP 2, hor. abs. (W)	$15 \sigma_x$	2.4	1431, 1652
IP 2, vert. abs. (Cu)	$150 \sigma_y$	4.3	1490
FD 2, hor. abs. (W)	$6 \sigma_x$	1.2	1967, 2187
IP 2, vert. abs. (Cu)	$40 \sigma_y$	1.1	2025
optional collimators in the final focus			
hor. abs. at SX1 (2nd)	$6 \sigma_x$	3.6	4097
hor. abs. at QY2 (2nd)	5%	4.5	4630
vert. abs. at SY1 (2nd)	$45 \sigma_y$	4.8	4715
hor. abs. at SII	5%	4.5	4798
hor. abs. at QFT6	$7 \sigma_x$	3.3	5040
vert. abs. at QFT5	$50 \sigma_y$	4.4	5056

Table 1: The collimators included in this study. The beam line starts with the post-linac diagnostic section at  $s = 0$  m. The interaction point (IP) is at  $s = 5210$  m. The labels 'IP' and 'FD' refer to collimators at the same betatron phase as the IP and the final doublet, respectively.

$2 \pm 0.3$  particles are lost over the last 500 m, primarily in the quadrupole QFT6M (at 5041 m), and also near Q2M, the first quadrupole of the final doublet. At the latter place only about  $0.2 \pm 0.08$  particles are lost per bunch train. These particle numbers do not change dramatically with beam energy: at 250 GeV about  $40 \pm 1$  particles are lost per train, and at 750 GeV the number is  $56 \pm 1$ .

The simulations are performed with a version of DIMAD which correctly treats the transverse dynamics even for large energy deviations. Figure 7 presents a result that was obtained for identical conditions with the original DIMAD program that is accurate only to 2nd order in  $\delta$ . The difference between Figs. 6 and 7 is small.

Figure 8 presents the results of simulations including 6 or 4 final-focus collimators, shown in the left and right picture, respectively. In these two cases, about 66% of the scattered particles are lost, compared with 44% in the absence of final-focus collimation. But now, there are no particles lost near the final doublet, whereas without final-focus collimation the loss there was about 0.2 per train. This suggests that the additional final-focus collimators would be effective.

### 3 BEAM-GAS SCATTERING

#### 3.1 Elastic Scattering

The cross section for elastic scattering (Mott scattering) is given by [6]

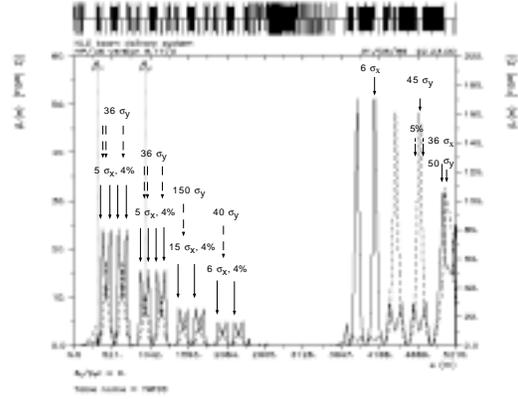


Figure 4: Horizontal and vertical beta functions in the NLC beam delivery system. The locations of spoilers and absorbers in the collimation system (the first 2.5 km) and possible locations in the final focus proper (the last 1.8 km) are also indicated, along with the respective collimation depths.

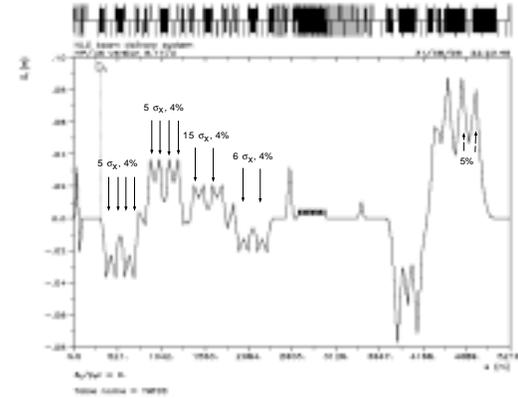


Figure 5: Dispersion function in the NLC beam delivery system. The locations of spoilers and absorbers for energy collimation are marked as in Fig. 4.

$$\frac{d\sigma_{el}}{d\Omega} = \left( \frac{Z\alpha\hbar c}{2pv} \right)^2 \frac{1 - \beta^2 \sin^2 \theta/2}{\sin^4 \theta/2} \quad (8)$$

Integrated above some minimum angle  $\theta_{min}$  the total cross section is approximately

$$\sigma_{el} \approx \frac{\pi\alpha^2\hbar^2 c^2 Z^2}{E^2 \sin^2 \theta/2}. \quad (9)$$

A minimum angle determined by the screening due to the atomic electrons is given by  $\theta_{min} \approx \hbar/(pa) \approx 20$  nrad, with  $a \approx 0.22\lambda_c/\alpha Z^{1/3}$  and assuming  $Z \approx 7$  (nitrogen, or carbon monoxide molecules)<sup>1</sup>. The total cross section is then  $\sigma_{el} \approx 10^{-23} \text{ m}^{-2}$ . At a CO pressure of 10 nTorr, this translates into a scattering probability of  $8 \times 10^{-9} \text{ m}^{-1}$ , or  $4 \times 10^7$  scatters per bunch train. The simulations show that the particles scattered at angles below  $1 \mu\text{rad}$  are not lost. Instead of 20 nrad, we thus assumed a limit  $\theta_{min} = 2 \mu\text{rad}$ , for which the scattering probability is  $8 \times 10^{-13} \text{ m}^{-1}$ , resulting in about 4000 scattering events per train. About 30% of these lead to a particle loss (1000 particles per

<sup>1</sup>The minimum beam divergence at the high beta points is much smaller than this, only about 0.5 nrad.

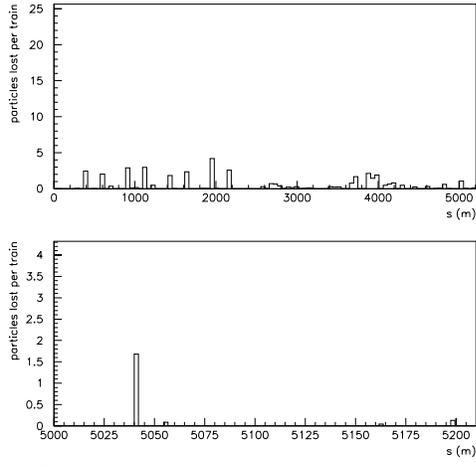


Figure 6: Number of particles per bunch train lost due to thermal-photon scattering at different locations along along the beam line. Top: the entire beam-delivery system, bottom: the last 200 m prior to the IP.

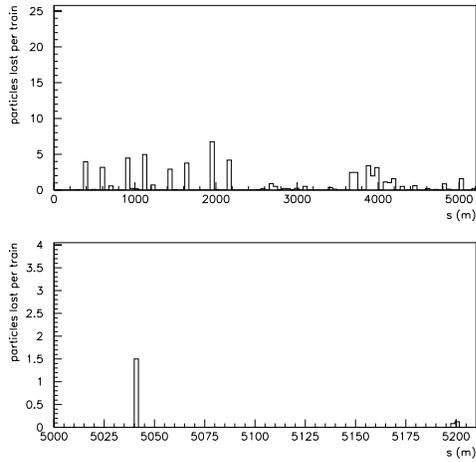


Figure 7: Same figure as Fig. 6 but calculated with the original DIMAD program. The latter is chromatically correct only to 2nd order in  $\delta$ .

train in the entire 5 km beam-delivery section). The loss distribution is illustrated in Fig. 9. Most losses occur at the spoilers and absorbers of the collimation section.

In Fig. 10 we depict simulation results obtained including final-focus collimators. With or without these collimators, there are no losses in the immediate vicinity of the final doublet. The losses 150–200 m upstream of the final doublet are reduced by the additional collimation. The fraction of scattered particles (at an angle larger than  $2 \mu\text{rad}$ ) which are lost is about 43 %, a factor of 1.5 higher than without final-focus collimation.

### 3.2 Inelastic Scattering

The differential cross section for inelastic scattering (Bremsstrahlung) is approximately given by

$$\frac{d\sigma}{d\delta} = \frac{A}{N_A X_0} \frac{1}{k} \left( \frac{4}{3} - \frac{4}{3}k + k^2 \right) \quad (10)$$

where  $N_A$  is the Avogadro number,  $A$  the atomic mass, and

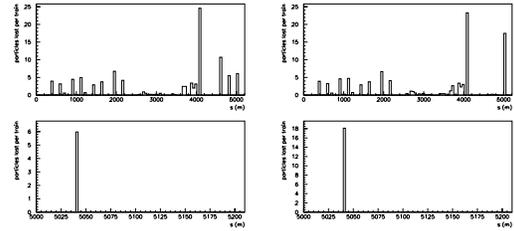


Figure 8: Number of particles per bunch train lost due to thermal-photon scattering at different locations along along the beam line. Left: with 6 additional final-focus collimators; right: with 4 final-focus collimators.

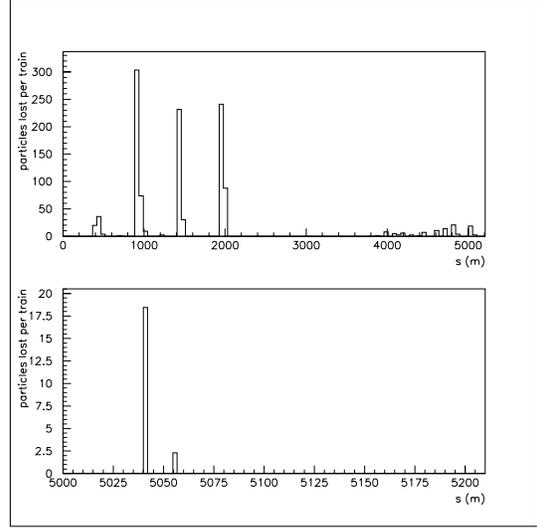


Figure 9: Number of particles per bunch train lost due to elastic beam-gas scattering at different locations along the beam line, without final-focus collimation.

$$\frac{1}{X_0} \approx 4\alpha r_e^2 N_A \frac{Z^2}{A} \ln \left( \frac{183}{Z^{1/3}} \right) \quad (11)$$

the radiation length. The total cross section for scattering with a relative energy loss larger than  $\delta_{min}$  is given by [8]

$$\sigma_{inel} \approx -\frac{16}{3} r_e^2 \alpha Z^2 \ln \delta_{min} \ln \left( \frac{183}{Z^{1/3}} \right) \quad (12)$$

For  $\delta_{min} = 1 \%$  and CO or  $N_2$  molecules, this cross section is about 6 barn. At a pressure of 10 nTorr, the scattering probability is  $2 \times 10^{-13} \text{ m}^{-1}$ . This corresponds to 1000 events per bunch train over 5 km. About half of these (500 per train) are lost by hitting aperture or collimators,

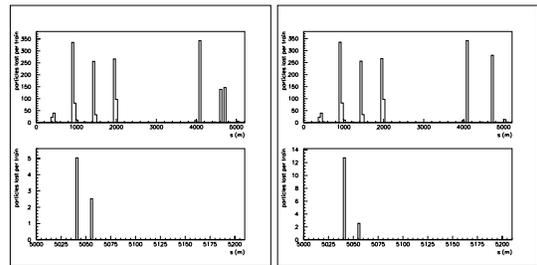


Figure 10: Number of particles per bunch train lost due to elastic beam-gas scattering at different locations along the beam line. Left: with 6 additional final-focus collimators; right: with 4 final-focus collimators.

and almost  $20 \pm 3$  hit apertures within 200 m from the IP. The simulated loss distribution is shown in Fig. 11.

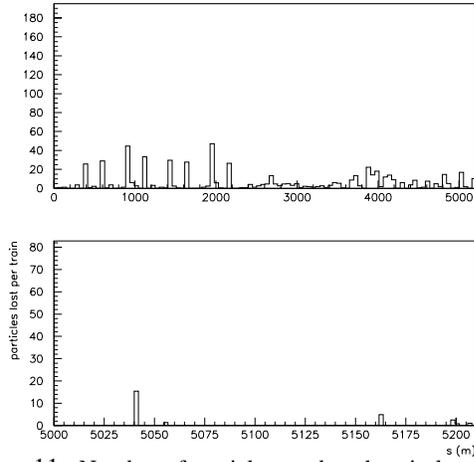


Figure 11: Number of particles per bunch train lost due to inelastic beam-gas scattering at different locations along the beam line, without final-focus collimation.

Figure 12 again shows the effect of final-focus collimators. Also in this case, considerably more scattered particles are lost in the final focus than would be without final-focus collimation. In total, about 80 % of the particles scattered with an energy loss above 1 % are lost. Without final-focus collimation this fraction was 58 %. The final-focus collimators do however reduce the number of particles lost within 50 m from the final doublet by roughly a factor of 2, from about 10 to about 5 (note the different scale of the two bottom pictures).

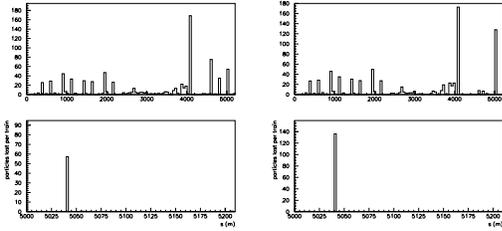


Figure 12: Number of particles per bunch train lost due to inelastic beam-gas scattering at different locations along the beam line. Left: with 6 additional final-focus collimators; right: with 4 final-focus collimators.

### 3.3 Scattering on Atomic Electrons

If an electron scatters elastically on an atomic electron, in the laboratory frame it can lose a significant fraction of its energy. The cross section for scattering on atomic electrons is

$$\begin{aligned} \frac{d\sigma}{d\delta} &= -\frac{2\pi r_e^2 n_{atom} Z}{\gamma} \left\{ \frac{1 - 2\delta + 3\delta^2 - 2\delta^3 + \delta^4}{\delta^2(1 - 2\delta + \delta^2)} \right\} \\ &\approx -\frac{2\pi r_e^2 n_{atom} Z}{\gamma \delta^2} \end{aligned} \quad (13)$$

where  $Z$  is the atomic number of the residual gas atoms,  $n_a$  the number of atoms per molecule,  $r_e$  the classical electron radius and  $\delta > 0$ . Integration yields the total cross section for losing an energy larger than  $\delta_{min}$ :

$\sigma = 2\pi r_e^2 Z / (\gamma \delta_{min})$ . For example, with  $Z = 8$  and  $\delta_{min} = 1\%$ , the cross section is  $\sigma = 8 \times 10^{-28} \text{ cm}^2$ . This would amount to only 0.1 scattered particles per bunch train, a negligible effect. The energy loss  $\delta$  and the scattering angle in the center-of-mass frame are related by  $\delta = (\cos \theta^* - 1)/2$ . The scattering angle in the laboratory frame is

$$\tan \theta = \frac{1}{\sqrt{2}\gamma} \frac{2\sqrt{\delta + \delta^2}}{1 - 2\delta} \quad (14)$$

As an example, for  $\delta \approx 0.1\%$  the scattering angle would be  $\theta \approx 44 \mu\text{rad}$ .

## 4 CONCLUSIONS

In the 5 km long NLC beam delivery system, about one thousand particles per train are lost due to elastic beam-gas scattering and about five hundred due to inelastic beam-gas scattering (Bremsstrahlung), assuming 10 nTorr average CO pressure. Of these particles, only about 10 and 20, respectively, are lost over the last 200 m prior to the IP. These are 10 times more than the number of particles lost by scattering on thermal photons, which are about 50 in total, and 2 over the last 200 m. At a vacuum pressure of about 1 nTorr, the losses due to beam-gas scattering and due to thermal-photon scattering would be equal.

## 5 ACKNOWLEDGEMENTS

We thank our colleagues at CERN, in particular Helmut Burkhardt and Ghislain Roy, who helped writing the modified version of DIMAD used for these studies.

## 6 REFERENCES

- [1] V.I. Telnov, "Scattering of Electrons on Thermal Radiation Photons in Electron-Positron Storage Rings", Nucl. Instr. Methods A, p. 304 (1987).
- [2] B. Dehning et al., "Scattering of high energy electrons off thermal photons", Phys. Lett. B249 p. 145 (1990).
- [3] M. Lomperski, "Compton Scattering off Blackbody Radiation and other Backgrounds of the HERA Polarimeter", DESY-93-045 (1993).
- [4] C. Adolphsen et al., "Zeroth Order Design Report for the Next Linear Collider," *SLAC-Report 474* (1996).
- [5] R.V. Servranckx, K.L. Brown, L. Schachinger, D. Douglas, "User's Guide to the Program DIMAD", SLAC-285 (1990).
- [6] I. Reichel, "Study of the Transverse Beam Tails at LEP", Ph.D. thesis, RWTH Aachen (1998).
- [7] H. Burkhardt, "Monte Carlo Simulations of Scattering of Beam Particles and Thermal Photons", CERN SL/Note 93-73 (OP) (1993).
- [8] A. Piwinski, "Beam Losses and Lifetime", in CERN Accelerator School, Gif-sur-Yvette, France, CERN 85-19 Vol. II (1985).

# Participants

Adam, Stefan (PSI)  
Adelman, Andreas (PSI)  
Anderson, Edwin (Cornell)  
Arbenz, Peter (Swiss Fed. Inst)  
Bailey, David (NERSC)  
Bane, Karl (SLAC)  
Barnard, John (LLNL)  
Becker, Ulrich (CST)  
Bondarev, Boris (Moscow Radio. Inst)  
Bruhwiler, David (Tech-X Corp)  
Bruns, Warner (TU-Berlin)  
Carey, David (FNAL)  
Cary, John (Tech-X Corp)  
Cendes, Zoltan (Ansoft)  
Chin, Yong Ho (KEK)  
Chou, Weiren (FNAL)  
Cleland, Timothy (LANL)  
Colstock, Pat (FNAL)  
Cooke, Simon (SAIC)  
Corbett, Jeff (SLAC)  
DeFord, John (STAR, Inc)  
Dolgashev, Valery (BINP)  
Durkin, Alexander (Moscow Radio. Inst)  
Fedotov, Alexei (U Maryland)  
Friedman, Alex (LLNL)  
Gillespie, George (Gillespie Assoc)  
Gluckstern, Robert (U Maryland)  
Grote, Dave (LLNL)  
Habib, Salman (LANL)  
Heifets, Samuel (SLAC)  
Herrmannsfeldt, William (SLAC)  
Hill, Barrey (Gillespie Assoc)  
Holt, James (FNAL)  
Humphries, Stanley (U New Mexico)  
Illy, Stefan (ITP, Karlsruhe)  
Irwin, John (SLAC)  
Iselin, Christoph (CERN)  
Iwashita, Yoshihisa (Kyoto U)  
Jones, Frederick (Triumpf)  
Kalkanis, Gregory (Varian)  
Katsouleas, Tom (UC Santa Cruz)  
Keil, Eberhard (CERN)  
Kirk, Harold (BNL)  
Kishak, Rami (U Maryland)  
Ko, Kwok (SLAC)  
Komorowski, Piotr (CERN)  
Krawczyk, Frank (LLNL)  
Kurennoy, Sergey (LLNL)  
Lampel, Michael (Gillespie Assoc)  
Lee, Martin (SLAC)  
Li, Zenghai (SLAC)  
Loewen, Rod (SLAC)  
McCandless, Brian (SLAC)  
McCrory, Elliott (Fermilab)  
Mi, Wanjun (SLAC)  
Millich, Antonio (CERN)  
Mondelli, Alfred (SAIC)  
Moore, John (Gillespie Assoc)  
Morpurgo, Giulio (CERN)  
Nelson, Eric (LANL)  
Ng, Cho (SLAC)  
Nishimura, Hiroshi (LBNL)  
Novokhatski, Alexandre (TU-Darmstadt)  
Petillo, John (SAIC)  
Pilat, Fulvia (BNL)  
Pedobedov, Boris (SLAC)  
Qiang, Ji (LANL)  
Raubenheimer, Tor (SLAC)  
Ruggiero, Francesco (CERN)  
Ruth, Ron (SLAC)  
Ryne, Robert (LANL)  
Salop, Arthur (Varian)  
Saparov, Misha (SLAC)  
Schoessow, Paul (ANL)  
Sharp, William (LLNL)  
Shasharina, Svetlana (Tech-X Corp)  
Simon, Horst (NERSC)  
Simpson, Jim (SLAC)  
Srinivas, Vinay (SLAC)  
Stevens, Rick (ANL)  
Stupakov, Gennady (SLAC)  
Sun, Yong (SLAC)  
Tatchyn, Roman (SLAC/SSRL)  
Thompson, Kathleen (SLAC)

PARTICIPANTS

Timm, Martin (TU-Darmstadt)  
van Rienen, Ursula (Rostock U)  
Venturini, Marco (U Maryland)  
Vlieks, Arnold (SLAC)  
Walker, Nicholas (DESY)  
Walling, Linda (Ansoft)  
Wang, Chun-Xi (SLAC)  
Wang, Fuhua (MIT)  
Wang, Juwen (SLAC)

Wangler, Thomas (LANL)  
Warnock, Robert (SLAC)  
Weiland, Thomas (TU-Darmstadt)  
Wipf, Susan (DESY)  
Wolf, Michael (SLAC)  
Wong, Jonathan (SLAC)  
Wong, Ron (SLAC)  
Woodley, Mark (SLAC)

# Author Index

## A

Abell, D., 184  
Adam, S., 60, 74  
Adelmann, A., 74  
Akasaka, N., 20  
Anderson, E. B., 122  
Arbenz, P., 60

## B

Bane, K. L. F., 137, 145  
Banks, T. I., 122  
Barnard, J. J., 174  
Beebe-Wang, J., 179  
Bernal, S., 55  
Blaskiewicz, M., 179  
Boine-Frankenheim, O., 38  
Borland, M., 23  
Bruns, W., 65

## C

Cai, Y., 6  
Carey, D. C., 189  
Carron, G., 154  
Chin, Y. H., 107, 209  
Cobb, J. W., 179

## D

Dohlus, M., 204  
Dolgashev, V. A., 149  
Dragt, A. J., 164, 184

## E

Enomoto, A., 20

## F

Fedotov, A. V., 45  
Flanagan, J. W., 20  
Friedman, A., 50, 174  
Furman, M. A., 117  
Furukawa, K., 20

## G

Galambos, J. D., 179  
Gillespie, G. H., 102, 164  
Gillespie, J. S., 92  
Glock, H. W., 159  
Gluckstern, R. L., 43, 45  
Grote, D. P., 50, 55, 174

## H

Haber, I., 50, 55  
Habib, S., 87  
Hajima, R., 82, 197  
Hecht, D., 159  
Hill, B. W., 92, 102, 164  
Hofmann, I., 38  
Holmes, J. A., 179

## I

Ilselin, F. C., 1

## J

Jeon, D., 179

## K

Kamitani, T., 20  
Kazakov, S., 209  
Kishek, R. A., 55

Koiso, H., 20  
Komorowski, P., 192  
Kurennoy, S. S., 45, 213

## L

Lampel, M. C., 102, 164  
Luccio, A. U., 179

## M

Malitsky, N., 11, 79  
Mardahl, P. J., 218  
Martono, H., 92, 164  
Millich, A., 154  
Moore, J. M., 92, 164  
Morpurgo, G., 97, 169  
Mosnier, A., 137

## N

Neugebauer, F., 201  
Nishimura, H., 16  
Novokhatski, A., 69, 132, 137, 140

## O

Ogawa, Y., 20  
Ohsawa, S., 20  
Oide, K., 20  
Olsen, D. K., 179

## P

Pierini, P., 33

## Q

Qiang, J., 87

## R

Raubenheimer, T. O., 222

Reichel, I., 222  
Reiser, M., 55  
Rogers, J. T., 122  
Ryne, R. D., 45, 87

## S

Schulte, D., 127  
Shadwick, B. A., 112  
Stupakov, G. V., 145  
Suwada, T., 20

## T

Talman, R., 11, 79  
Thorndahl, L., 154  
Timm, M., 132, 140  
Tenenbaum, P., 222  
Tommasini, D., 192  
Tsutsui, H., 209

## V

van Rienen, U., 159, 201  
Venturini, M., 55, 184  
Verboncoeur, J. P., 218

## W

Wang, J. Q., 209  
Wangler, T., 28  
Weiland, T., 69, 132, 140  
Wipf, S. G., 204  
Wurtele, J. S., 112

## Y

Yamaguchi, S., 209  
Yokoya, K., 137

## Z

Zimmermann, F., 222