

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].

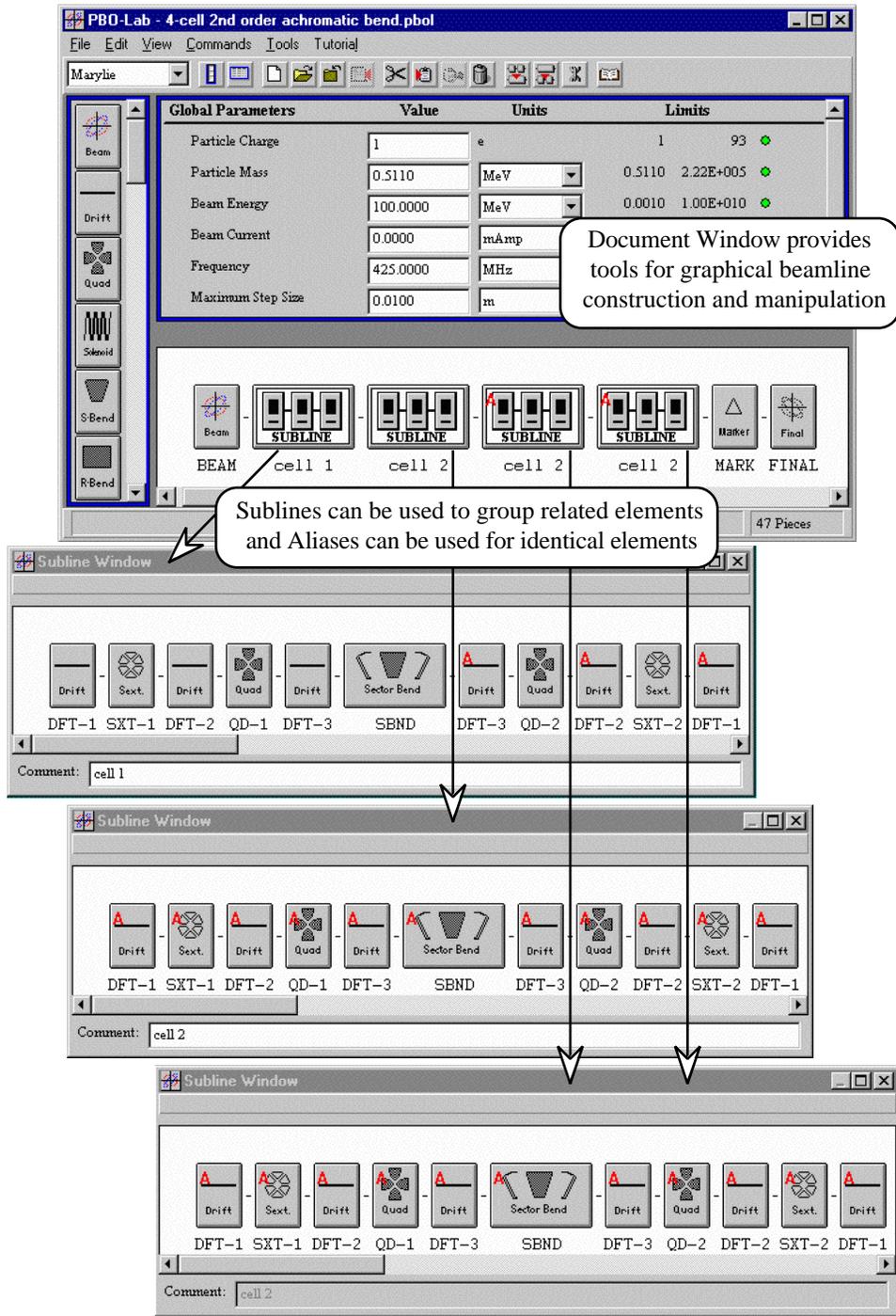


Figure 1. Example Document Window and three Subline Windows of the graphic user interface for MARYLIE.

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 1st 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 2nd 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 3rd and 4th cells, also labeled as “cell 2” since they are identical to the 2nd cell. Subline Aliases may contain Aliases to individual Pieces or other Subline Aliases.

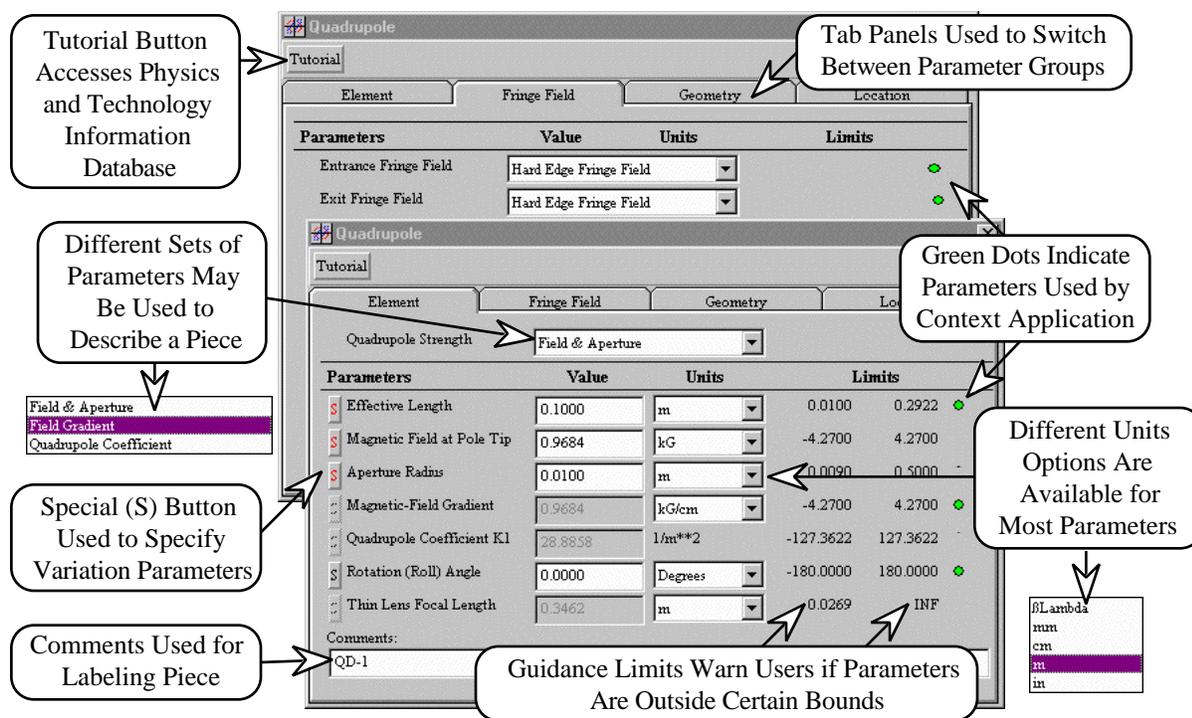


Figure 2. Example of Piece Window (left) and Alias Window (right) for editing parameters of a magnetic bend element.

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.

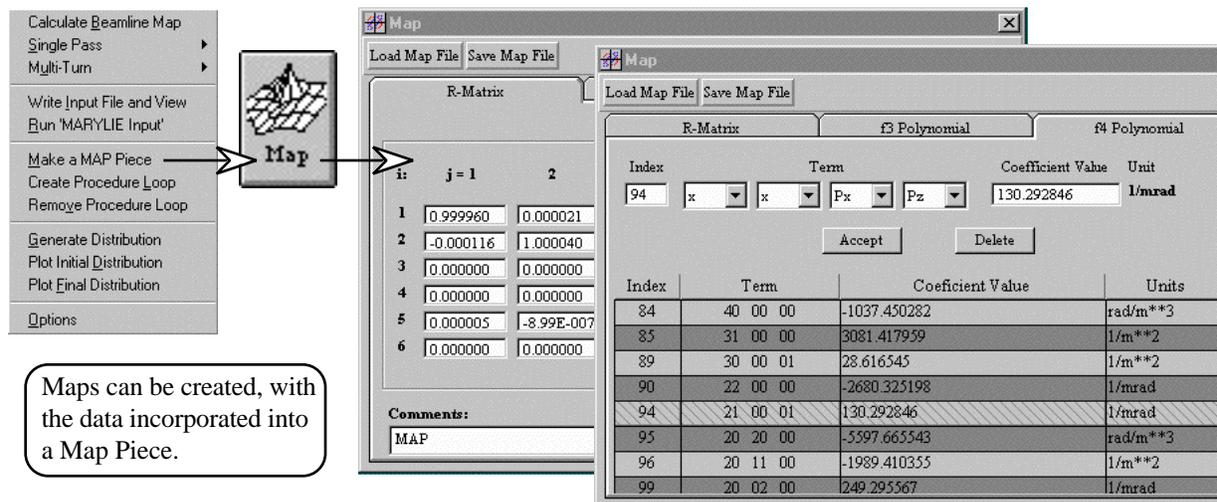


Figure 4. A map can be generated for a group of elements and the data entered automatically into a Map Piece.

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*TM is available from AccelSoft Inc., see 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 LabTM) 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).