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 appropriated.

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.

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]

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

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

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

Please note that we use δ 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

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

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

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

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

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

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

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 nonlinearity. 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]

$$H = H_0 + H_1, \tag{6}$$

where

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

and

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

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]

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

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

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

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

$$\begin{aligned} e^{:-lH:} \quad = \quad & e^{:-c_1lH_0:}e^{:-d_1lH_1:}e^{:-c_2lH_0:}e^{:-d_2lH_1:} \\ & e^{:-c_2lH_0:}e^{:-d_1lH_1:}e^{:-c_1lH_0:} + O(l^5), \end{aligned} \tag{11}$$

where l is the length of integration and

$$\begin{aligned} c_1 &= \frac{1}{2(2-2^{\frac{1}{3}})}, c_2 = \frac{1-2^{\frac{1}{3}}}{2(2-2^{\frac{1}{3}})}, \\ d_1 &= \frac{1}{2-2^{\frac{1}{3}}}, d_2 = \frac{-2^{\frac{1}{3}}}{2-2^{\frac{1}{3}}}. \end{aligned} \tag{12}$$

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

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

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

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

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.

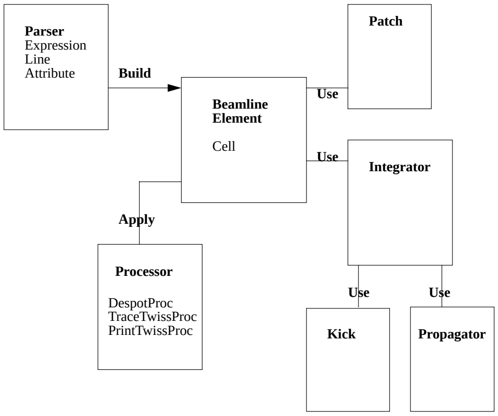

Figure 1: Main Lego Modules

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 nonlinear 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.

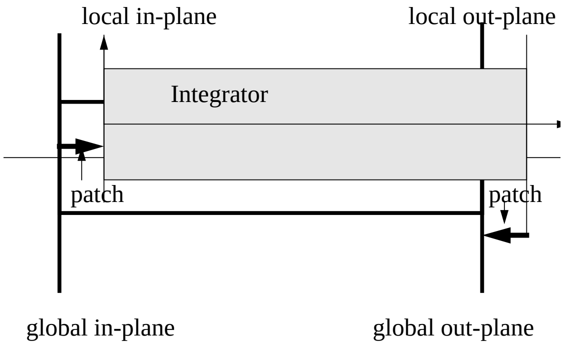

Figure 2: Lego Concept

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 additional 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.

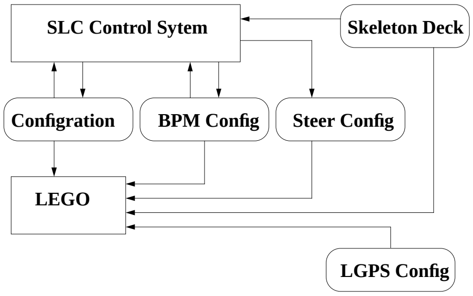

Figure 3: Interface to the control system

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.

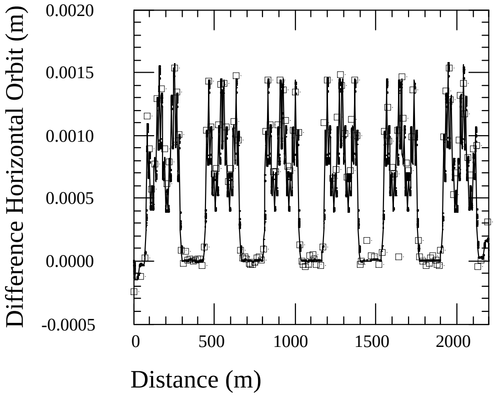

Figure 4: Horizontal dispersion measurement for the HER with -1 kHz change of the RF frequency, the square symbols present the measurement data and solid line for the simulation

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

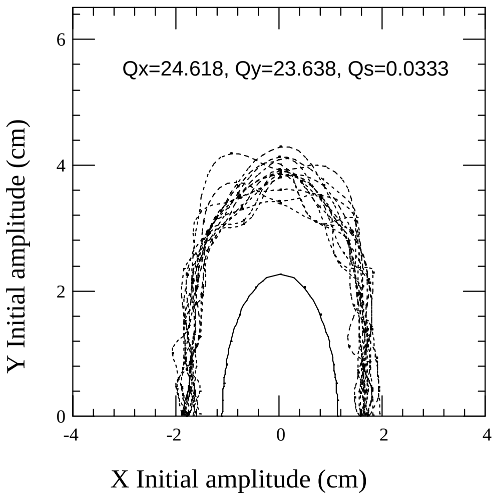

Figure 5: The prediction of the dynamic aperture for the designed lattice with specified alignment and magnetic errors for the HER, the solid line presents 10 sigma of the nominal beam size assuming the vertical emittance is half of the horizontal one and the dotted lines are the dynamic aperture of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

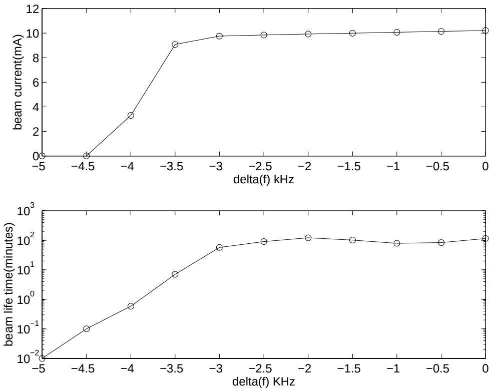

Figure 6: The measurement of the dynamic aperture for the HER

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 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. Vlissides, "Design Patterns, Elements of Reusable Object-Oriented Software," Addison-Wesley Professional Computing Series, 1994

[10] M. Donald, Y. Cai, H. Shoaee 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.