



Fermi National Accelerator Laboratory

FERMILAB-Conf-93/124

Accelerator Physics Analysis with Interactive Tools

J.A. Holt and L. Michelotti

*Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510*

May 1993

Presented at the *1993 Particle Accelerator Conference*, Washington, D.C., May 17-20, 1993

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Accelerator Physics Analysis with Interactive Tools

J. A. Holt and L. Michelotti

*Fermi National Accelerator Laboratory[†]
P. O. Box 500, Batavia, IL 60510, USA*

Abstract

Work is in progress on interactive tools for linear and non-linear accelerator design, analysis, and simulation using X-based graphics. The BEAMLINe and MXYZPTLK class libraries, were used with an X Windows graphics library to build a program for interactively editing lattices and studying their properties.

I. INTRODUCTION

There are a number of general-purpose accelerator tracking codes in wide use such as SYNCH [1], MAD [2], or TEAPOT [3]. Each of these programs has an input format or language in which the user formulates the problem to be solved. This language is different from the language the code is written in. Frequently the accelerator physicist wants to solve a problem which does not fit within the constraints of the input language. The physicist must then duplicate the basic lattice handling functions which all codes must have as well as the special functions which solve the problem at hand. Also some problems can be solved more quickly when the input can be changed interactively and both the intermediate and final results be seen graphically. Most codes at present are a batch type operation; an input file is edited, the program is run and the final results are output.

We are developing a set of tools in which the input language is the same as the programming language. The code is written in C++ partly because of the ease in creating new "objects" that behave in every respect like fully functional variables of the language. We have integrated several types of class libraries to form applications customized to individual problems.* Different authors

worked on different parts of the libraries. Because everything was encapsulated in objects, one author did not have to know any details of the objects created by another; only the interface to that object had to be known. As reported elsewhere [4], the MXYZPTLK [7] and BEAMLINe [8] class libraries were used to build a user-friendly, interactive, four-dimensional phase space tracker which, additionally, finds periodic orbits. The interface was based on the OSF/Motif widget toolkit and a C++ class library, MotifApp, written by Young [5]. The MotifApp class library is built upon to create tool classes. These widget classes reuse the `beamline` class by absorption; that is, they contain pointers to beamlines as part of their internal (private) data. The next section describes tools which have been written since writing [4].

II. BASIC TOOLS

Basic to any accelerator analysis program is the ability to edit an accelerator lattice, calculate twiss parameters and fit the tunes among other things.

A text-oriented lattice editor has been developed to interactively edit accelerator lattices. The editor consists of a scrollable list on which the user can insert and delete elements, by double-clicking on an element, the user is prompted for changes to that element by a dialog box. The lattice list can be filtered by element type. For example, one can list only the quadrupoles if so desired. The resulting lattice can be saved or recalled from a file. An additional command enables the user to "sectorize" (create a polynomial map) any portion of or all of the beamline.

A twiss parameter tool has been developed to display twiss parameters in several different formats which can be filtered by element type. There is a text format which can be saved to and recalled from disk, a list format where the user can double-click on a particular line and edit that element similar to the lattice editor and a graphics format where the twiss functions are plotted as a function of length along the beamline. Figure 1 show an example of the lattice editor and twiss tool. The parameter β_x is plotted [9] for the Tevatron low-beta lattice.

[†] Operated by the Universities Research Association Inc, under contract with the U.S. Department of Energy

*This is something in the spirit, if not substance, of LAMBDA [6] but without reference to data bases or specially formatted input files and with a better programming paradigm (object-oriented programming).

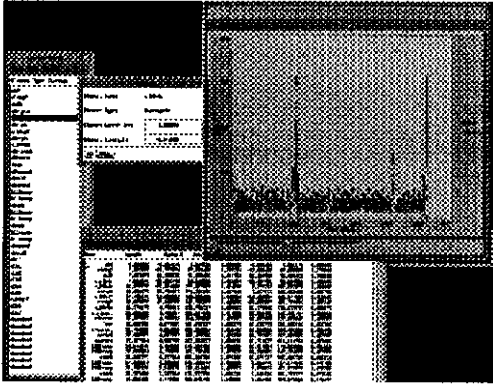


Figure 1: Lattice Editor and Twiss Tool Examples

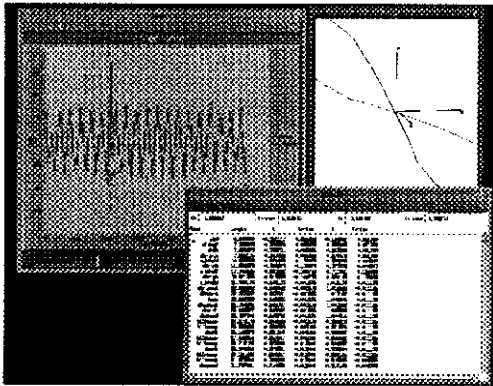


Figure 2: Tevatron Orbits using the Orbit Tool

There is an orbit tool which will receive points from the phase space tracker and plot/list the orbits along the lattice. There is also a 3D orbit plotter [10] which is useful for plotting the Tevatron's helical orbits. Figure 2 Shows the Tevatron antiproton orbit in 2D and both orbits at the D0 interaction region in 3D.

The circuit editor allows different elements to be grouped together in analogy to an electrical circuit. This is useful in the case of the Tevatron where different quadrupoles are connected to the same electrical bus.

With the tune fitter tool the user interactively selects the focusing and defocusing circuits and is prompted for the desired tunes via a dialog box.

III. COMPOSITE TOOLS

All of the above tools can be used separately or can be combined together to make a more complicated analysis tool. As an example, the Tevatron collider has two low-beta regions for the two experiments CDF and D0. During the present collider run, it was discovered that the beams were not colliding at the center of the D0 detector and that there was a large beta-wave in the machine. A composite tool was developed to look at the perturbations in β when the values of various quadrupoles were changed. Figure 3

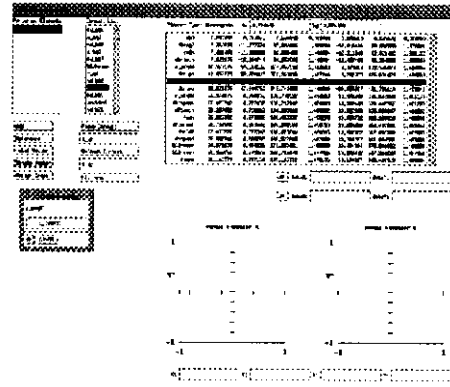


Figure 3: The $\Delta\beta/\beta$ calculation/plotting Tool

shows how the tool is laid out. In the upper right is the twiss tool. Next to it on the left is a circuit editor. Below that is the tune fitter. To the left of the circuit editor is a list of quadrupoles to be perturbed. At the bottom is the vector plot of $\Delta\beta/\beta$ vs. $(\Delta\beta/\beta)'$ for both planes. The buttons on the far left control various functions such as choosing the reference point plotting control etc.

IV. FUTURE WORK

We plan to continue improving the prototype interactive tools developed so far and to add to the collection. The higher priority goals to be addressed include: (a) incorporating procedures into the phase space widgets so that orbit displays can toggle between "ordinary" or normal form coordinates; (b) interactively displaying the layout of the beamline using a three-dimensional graphics environment, such as Phigs; (c) tying that display to the lattice editor tools, so that beamline elements (or families of beamline elements) can be modified after picking them with a cursor; and (d) improving communication between the tools with persistent objects.

REFERENCES

- [1] A. A. Garren, A. S. Kenney, E. D. Courant, A. D. Russell, and M. J. Syphers. *SYNCH: A Computer System for Synchrotron Design and Orbit Analysis. Users Guide*. Los Alamos Accelerator Code Group. Los Alamos National Laboratory.
- [2] Hans Grote and F. Christoph Iselin. *The MAD Program (Methodical Accelerator Design). User's Reference Manual*. European Organization for Nuclear Research, Geneva, Switzerland. CERN/SL/90-13.
- [3] L. Schachinger and R. Talman, *TEAPOT: A Thin Element Accelerator Program for Optics and Tracking*, Superconducting Super Collider Laboratory, SSC-52 December 1985.

- [4] J. A. Holt, L. Michelotti and T. Satogata. *Accelerator Physics Analysis with an Integrated Toolkit* Int. J. Mod. Phys. A(Proc. Suppl.), **2b**, 1007, 1993.
- [5] Douglas A. Young. *Object-Oriented Programming with C++ and OSF/Motif*. Prentice Hall. Englewood Cliffs, NJ. 1992.
- [6] Eric Barr, et al., *Lattice Parameters Database and Operational Simulation at FNAL and SSCL*. IEEE Particle Accelerator Conference 1991, Proceedings. Vol.I pp.207-209.
- [7] L. Michelotti. *WXYZPTLK: A practical users-friendly C++ implementation of differential algebra: User's Guide*. Fermi Note FN-535, Fermilab, January 31, 1990; *WXYZPTLK: A C++ Hacker's Implementation of Automatic Differentiation*. In *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*. SIAM, Philadelphia, PA. 1991.
- [8] L. Michelotti, *WXYZPTLK and Beamline: C++ Objects for Beam Physics*. In *Advanced Beam Dynamics Workshop on Effects of Errors in Accelerators, their Diagnosis and Correction*. (held in Corpus Christi, Texas, October 3-8, 1991) Published by American Institute of Physics, as Conference Proceedings No. 255. 1992.
- [9] Peter Klingebiel and Gregory Bond, *Using the AthenaTools Plotter Widget*, This plotter widget set was obtained via the Internet from a software archive site.
- [10] Mark Edel, Jeff Kallenbach, Joy Kyriakopoulos, Paul Lebrun and Donna Reid, *Visajet*, Fermilab, PU0136, 1992. The 3D widget was taken from the Visajet software which is part of the Nirvana Project.