EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH

CERN–PS/96–12 (PA)
6 February 1996

PLATO: A PROGRAM LIBRARY FOR THE ANALYSIS OF NONLINEAR BETATRONIC MOTION[1]

M. Giovannozzi
and
E. Todesco[2], A. Bazzani[3], R. Bartolini

Abstract

We present PLATO (Perturbative Lattice Analysis and Tracking tOols), a program library for analysing four-dimensional betatronic motion in circular particle accelerators. The 136 included routines provide both the resonant and the non-resonant perturbative series that approximate nonlinear motion (normal forms); standard numerical tools such as the Lyapunov exponent, frequency analysis and evaluation of the dynamic aperture are also available. The aim of the library is to apply these techniques, originally developed for the analysis of nonlinear dynamic systems, to realistic models of particle accelerators. To ensure the highest flexibility, the code is fully compatible with standard tracking programs commonly used in the accelerator physics community.

(Submitted to Nuclear Instruments and Methods in Physics Research)

[1] Work partially supported by EC Human Capital and Mobility Contract No. ERBCHRXCT940480.

[2] INFN, Sezione di Bologna, Via Irnerio 46, I–40126 Bologna, Italy.

[3] Department of Mathematics, Piazza di Porta S. Donato 3, I–40126 Bologna, Italy.

1 INTRODUCTION

The search for confirmations of the Standard Model relies heavily on the new generation of hadron colliders, such as the planned LHC [1]. Higher and higher energy beams are required to produce the particles predicted by the theories of fundamental interactions, and a very intense bending magnetic field is necessary to confine these high-energy beams in a circular machine. The only way to produce strong fields is to use superconducting magnets. Unfortunately this technology has one main drawback (apart from the complex cryogenics needed): it is not possible to design a superconducting magnet producing a high-quality field. This means, for instance, that a dipole generates a magnetic field which, on top of the required uniform field, includes nonlinear multipolar components.

The particles circulating in superconducting accelerators experience nonlinear forces which produce strong instabilities and losses. These effects prevent safe operation of the machine for two main reasons. Firstly, the superconducting magnets are damaged by the energy deposited by the particles hitting the beam pipe and may quench. Secondly, the luminosity of the machine, which is proportional to the intensity of the circulating beam, decreases, thus reducing the rate of production of events.

A theory of nonlinear betatron motion has been developed since the fifties within the framework of the perturbative Hamiltonian theory used originally in celestial mechanics [2, 3], and a first-order analysis has been conducted at CERN [3, 4]. During the last ten years, there have been new developments. On the theoretical side, a perturbative approach for the one-turn maps, based on expansions in powers of the amplitude instead of the field gradients, has been developed [5–8]. This perturbative parameter fits more naturally to the features of the problem. On the computational side, efficient arbitrary-order codes have been written to automatically evaluate the coefficients of the one-turn map [7] and the perturbative series for generic maps, both in the non-resonant [7, 9] and resonant [9] cases.

Moreover, sophisticated numerical tools borrowed from celestial mechanics have proved to be useful indicators of the nonlinear motion: computation of the Lyapunov exponent [10, 11], frequency analysis [12–15], evaluation of the global dynamics through tune footprints [14, 16], and so on.

In this paper we outline the main features of the PLATO (Perturbative Lattice Analysis and Tracking tOols) program library, which contains several analytical and numerical tools which have been applied to the analysis of nonlinear beam dynamics [8, 15–19]. The main goal of PLATO is to allow the user to apply these approaches to generic accelerator lattices. The code is fully compatible with standard tracking programs commonly used in the accelerator physics community.

The paper is organized as follows. In Section 2 we describe the structure of the program library with special emphasis on the implementation. In Section 3 we discuss the structure of the input files and the interface with standard accelerator codes. Section 4 deals with the analysis of the tracking data using numerical tools, and in Section 5 the perturbative tools implemented in the program library are presented. Finally some conclusions are drawn in Section 6.

2 GENERAL FEATURES

2.1 Main aims

The routines in the program library can be grouped into three categories.

- Interface with certain tracking codes used in the accelerator physics community (MAD [4] and SIXTRACK [20]). A key feature of the library is the ability to analyse an arbitrary accelerator structure described using the MAD input language, and to provide the capability of using data produced by other tracking codes.

- Analysis of tracking data. The routines enable fine analysis of the results of numerical simulations (element-by-element tracking) to be performed. In particular, the dynamic aperture, the nonlinear tunes, and the maximal Lyapunov exponent can be evaluated. The tracking itself can be performed by routines included in our library.
- Normal forms analysis and related perturbative tools (non-resonant and resonant normal forms, quality factors and resonance analysis). The purpose of these modules is to extract information on particle dynamics from the truncated one-turn map without performing any tracking. Perturbative techniques can be used to obtain analytical expressions for the nonlinear invariants, the tune shifts, the main parameters of the resonances, and so on.

2.2 Structure of the code

The code is written in standard FORTRAN 77, any machine-dependent implementation having been avoided in order to improve portability. The code has been successfully tested on many different CERN platforms (ALPHAVAX, VM, Unix machines). The code is structured as a program library of 136 modules (118 subroutines and 18 functions). Since this library is continually growing, we use the PATCHY [21] format to maintain the source code, which enables us easily and efficiently to keep different versions of the library (e.g. a parallel version of the standard code, which is currently under test).

The user can select from the different subprograms. In the present version two main programs are available. The first allows tracking simulations to be performed using sophisticated numerical tools for post-processing. The second carries out the perturbative computations on the truncated one-turn map. Help files should allow the user to eventually modify his version of the code to fit his needs and to design his own personal version of the main program.

2.3 Help

The program library and the related main codes are not intended for interactive use (even though this is not excluded); therefore there is no online help available. On the other hand, each routine has comments inserted within the code, enabling the user to understand each module and to use it in the correct way. In addition, a long write-up is available in the form of an ASCII file containing a description of the aims of the various routines, a parameter list, and a parameter description. A list of the different common blocks shared by the modules is also included.

3 LATTICE INPUT

The first stage in the analysis of particle motion in an accelerator is a description of the magnetic lattice of the machine. An accelerator is made up of a sequence of magnets. In principle, one record should be sufficient to describe the physical properties (length, magnetic field, etc.) of each component. A simple sequence of such records should be enough to describe a machine but, in practice, the assembling of such a file for a complex machine such as the Large Electron Positron collider (LEP) or the planned Large Hadron Collider (LHC) is tedious and inefficient.

The MAD program [4] contains a simple and powerful language which allows complex operations to be performed on a set of magnetic elements. Therefore, the fundamental elements of a machine (usually a fairly small number) can be defined, and then the whole accelerator can be described using a structured input language with a relatively small number of instructions. In addition, MAD is used to officially maintain the lattices of LEP and LHC, and has become a standard not only at CERN but in a large part of the accelerator physics community. For these reasons we have decided to use the MAD input data in defining the lattice parameters.

3.1 Interface with MAD

The MAD program allows a lattice structure — in the form of a simple sequence of records describing the elements of the machine — to be dumped to a disk file. Therefore, starting from a MAD input file, a second file with the sequential structure of the accelerator can be produced. This output file is the input that provides the lattice to the library: one of the modules reads and decodes it, generating the transfer matrices of the linear elements and the transfer maps of the nonlinear elements. In the latter case, each thick element is replaced by a sequence of a drift, a kick and a second drift without changing the total length. This approach represents the most efficient way to determine the transfer map of a nonlinear element without violating the Hamiltonian structure of the equations of motion. Once the data have been read, they are kept in arrays to be used by the other modules of the library.

3.2 Interface with SIXTRACK

Since a translation program is available to convert a MAD to a SIXTRACK input file, we decided not to develop any routine which uses the SIXTRACK input files. A relevant feature of this program is the capability of computing arbitrary-order truncated transfer maps for an accelerator structure. This option is implemented through the Differential Algebra package [7]. Interface routines that read the maps produced by SIXTRACK in order to analyse them within the library modules are available. Tracking-data files generated by SIXTRACK can be read as well, thus allowing SIXTRACK to be used for the numerical simulations, followed by application of the post-processing techniques implemented in the library.

4 TRACKING ANALYSIS

One of the main aims of the library is to carry out element-by-element tracking of an accelerator structure and to perform a sophisticated analysis of the tracking data. For this purpose, several routines have been grouped in a well-structured main program that performs the following operations.

- Dynamic aperture evaluation. One of the main sources of concern in beam dynamics is the dynamic aperture, i.e. the volume of the domain around the closed orbit where the particles remain confined in the vacuum pipe for the whole storage time. The numerical computation of the dynamic aperture is very CPU-time consuming, as one should scan along the four phase-space variables (x, p_x, y, p_x) to determine the stability domain.

  In [17] we reviewed the definition of dynamic aperture and proposed new methods for calculating it, by scanning the two spatial variables x, y only. The proposed techniques allow the information on the conjugate momenta to be taken into account by using an averaging process or normal forms. Several algorithms for estimating the dynamic aperture are included in the library.
- Tune evaluation. The tune is the ratio of the betatron to the revolution frequencies. It is a crucial parameter since it can drive resonances that endanger the beam stabil-

ity. The standard way to measure the tune of an orbit is based on the FFT algorithm, which allows the frequency corresponding to the maximum in the power spectrum to be determined; unfortunately, however, the accuracy in tune determination is rather poor. Other algorithms are based on the Average Phase Advance [8]. During the last decade, more efficient methods have been proposed [12–14], based either on an interpolation of the FFT results or on the use of harmonic analysis. In [15] a careful analysis of the intrinsic precision of the different algorithms is carried out. Several algorithms are implemented in the library: the most effective are the search for the maximum of the Fourier Transform and the interpolation of the FFT [13–15]. The indicator of 'chaoticity', based on the variation of the instantaneous tune over the orbit [13, 18, 22], is also directly implemented in the library.

- Tune footprints. The tune footprints [14, 16] of a magnetic lattice at a given working point can be generated by starting with a large set of initial conditions distributed in the phase space, determining their nonlinear tunes with a high precision (at least 10^{-4}), and plotting the result in the tune plane. In this way one has a clear picture of the areas in the tune space that are populated by stable particles, and a vivid picture of the resonance net which governs the stability of the system. Figure 1 shows a tune footprint, generated by the library, relating to a simple LHC model [16]. The clusters of points along straight lines reveal the phase locking due to stable resonances, while the depletion of some regions in the frequency space is the sign of hyperbolic structures.

[width=11cm]library1.eps

Figure 1: Tune footprint and resonance lines up to order 7 for the LHC-like cell lattice with random errors.

- Lyapunov evaluation. The Lyapunov exponent is a measure of the rate of divergence of nearby orbits, thus giving an indication of the local 'chaoticity' of a dynamic system [10, 11, 18, 23]. The computation of the Lyapunov exponent is a rather delicate issue and, when using the method of nearby particles, the distance of the two particles must be renormalized to avoid severe underestimates [11]. The evaluation of the maximal Lyapunov exponent through the nearby particles method and the renormalization technique is implemented in the program library.
  As an application of these techniques, we show in Fig. 2 the distribution of the Lyapunov exponent for a set of 1000 initial conditions of the Super Proton Synchrotron lattice used for experiments. This quantity is evaluated at four increasing numbers of turns. Using the tracking module of the code, we have also evaluated the stability of the same initial conditions over 10^6 turns. The unstable particles are represented by the shaded area on the histogram of the Lyapunov. It can be seen that there is a very strong correlation between the 1000 turns Lyapunov estimate and the long-term stability. This technique has recently been used [18] to obtain automated procedures for detecting long-term particle losses.

[width=12cm]library2.eps

Figure 2: Distribution of the Lyapunov exponent (calculated for different numbers of iterates) for a set of initial conditions of the SPS lattice at $\nu_1 = 26.832$, $\nu_2 = 26.799$. Shaded parts of the histograms represent initial conditions which are lost before 10^6 turns.

5 NORMAL FORM ANALYSIS

The library can also carry out arbitrary-order perturbative calculations on complicated accelerator lattices. A wide set of routines is dedicated to the computation of non-resonant and resonant normal forms [9] and other related quantities [5, 6, 8, 16, 19]. As for the tracking analysis, these routines have been included in a main program. The starting point of the perturbative analysis is the one-turn map of the lattice. The complex coordinates which diagonalize the linear part of the motion are denoted by $\mathbf{z} = (z_1, z_1^*, z_2, z_2^*)$, where $z_1 = \hat{x} + i\hat{p}_x$, $z_2 = \hat{y} + i\hat{p}_y$, and the * indicates the complex conjugation. The motion of a single particle is represented by the one-turn map $\mathbf{F}$, which propagates the position $\mathbf{z}$ of a particle at a given section of the machine to the position $\mathbf{z}'$ at the same section after one turn [8]:

$$\begin{aligned} z_1' = F_1(\mathbf{z}) &= e^{i\omega_1} z_1 + \sum_{n\geq 2} \sum_{j_1+j_2+j_3+j_4=n} F_{1;j_1,j_2,j_3,j_4}\, z_1^{j_1} z_1^{*j_2} z_2^{j_3} z_2^{*j_4} \\ z_2' = F_2(\mathbf{z}) &= e^{i\omega_2} z_2 + \sum_{n\geq 2} \sum_{j_1+j_2+j_3+j_4=n} F_{2;j_1,j_2,j_3,j_4}\, z_1^{j_1} z_1^{*j_2} z_2^{j_3} z_2^{*j_4} \ . \end{aligned} \qquad (1)$$

Here ω_1 and ω_2 are the linear tunes. Below we will describe the most important quantities that can be computed for a generic complicated lattice.

- Truncated one-turn map. The complex coefficients $F_{i;j_1,j_2,j_3,j_4}$ of the one-turn map can be calculated inside the library given the lattice input file; the truncation order $j_1 + j_2 + j_3 + j_4 \leq N$ depends on the number of nonlinear elements, on the maximum available memory, and on the speed of the platform used. The coefficients of the truncated one-turn map can also be externally generated and read by a file.
- Non-resonant normal forms. Once the map coefficients are stored, the conjugating function $\mathbf{\Phi}$ that transforms the one-turn map into its normal form $\mathbf{U}$ [5–8] can be calculated. The normal form is a map that has more symmetries with respect to the original map, and exhibits explicit invariants (with the exception of the double-resonance normal form). It can be expressed as the Lie series of the interpolating Hamiltonian h. One can build either non-resonant or resonant perturbative expansions. In the first case the Hamiltonian is a function of the amplitudes (ρ_1, ρ_2) in the normalized space; the normal form is the direct product of rotations in the two phase planes, whose nonlinear frequencies depend on the distance from the origin within these planes. The two components of the inverse conjugating function, Ψ_1 and Ψ_2, give the approximated nonlinear invariants ρ_1 and ρ_2. Moreover, an analytic expression for the nonlinear tunes is given by the derivative of the Hamiltonian with respect to the amplitudes:

$$\begin{aligned} \nu_x(\rho_1, \rho_2) &= \frac{\partial}{\partial \rho_1} h(\rho_1, \rho_2) \\ \nu_y(\rho_1, \rho_2) &= \frac{\partial}{\partial \rho_2} h(\rho_1, \rho_2) \ . \end{aligned} \qquad (2)$$

- Single-resonance normal forms. In this case one selects a single resonance $(q\nu_x + p\nu_y)$ with $q \in \mathbf{N}$ and p integer. In the following we will use the notation $[q,p]$ to indicate a resonance. The normal form is the Lie series of an interpolating Hamiltonian that is a function of the amplitudes and of one linear combination of angles:

$$h(\rho_1,\rho_2,\theta_1,\theta_2) = \sum_{k_1,k_2,l} h_{k_1,k_2,l}\, \rho_1^{k_1+lq/2}\rho_2^{k_2+l|p|/2} \cos\left[l(q\theta_1+p\theta_2)+\varphi_{k_1,k_2,l}\right] . \tag{3}$$

  The coefficients of these Hamiltonians give important information about which resonances are relevant for the lattice and which are not. Moreover, the position of the resonance line in the space of invariants can be calculated, and the width of the resonance and the eigenvalues of the fixed lines can be determined. In Fig. 3 the average position and the width of the resonance $[3,-6]$ in the space of the invariants are given for a lattice with a single sextupole (Hénon map) in the one-kick approximation at $\omega_1/2\pi = 0.28$, $\omega_2/2\pi = 0.31$. The agreement with the tracking results of the initial conditions that are locked on the resonance (dots) is fairly good, even though the resonance starts from high amplitudes and extends to the dynamic aperture.

[width=8cm]library3.eps

Figure 3: Initial conditions that are locked on the resonance $[3,-6]$ for the Hénon map at $\omega_1/2\pi = 0.28$, $\omega_2/2\pi = 0.31$. Numerical results (dots) are compared with the analytical estimate of the resonance width (solid lines) obtained using resonant normal forms at order 10. Horizontal and vertical nonlinear invariants are plotted in the x and y axes, respectively.

- Double-resonance normal forms. In this case one selects two resonances $(q_1\nu_x + p_1\nu_y)$ and $(p_2\nu_x + q_2\nu_y)$: the normal form is the Lie series of an interpolating Hamiltonian that is a function of the amplitudes and of two linear combinations of angles. This Hamiltonian is not integrable, but can be used to work out the position and the stability of the fixed points that arise when the two single resonances are crossing [19, 24].
- Quality factors. In many optimization problems it is necessary to analyse different versions of a lattice in order to select the one with the highest dynamic aperture. This process is usually very CPU-time consuming. A solution is to find a quality factor (QF), i.e. a quantity having a good correlation with the dynamic aperture, which can be calculated in a short time. The QF can then be used to rate the performance of a lattice [16]. Three QFs based on nonlinear maps and normal forms are directly implemented in the library:
  - the norm of the nonlinear part of the map evaluated at the amplitude A;
  - the average tuneshift at amplitude A evaluated through non-resonant normal forms;
  - the norm of the resonant part of the interpolating Hamiltonian of the single-resonance normal form evaluated at the amplitude A;

  A detailed description of the definition of the quality factors can be found in [16]. Figure 4 shows the correlation between the dynamic aperture and the three quality factors for a simplified version of the LHC lattice, including only random sextupolar errors. We used 100 seeds to generate 100 different error distributions, and for each machine both the dynamic aperture and the quality factor were evaluated. It can be seen that some QFs show a fairly good correlation with the dynamic aperture, and therefore can be used to distinguish between good and bad machines. These techniques have been used to propose a sorting method for the LHC [16].

[width=12cm]library4.eps

Figure 4: (a) Distribution of the dynamic apertures for an LHC-like cell lattice with random errors (100 seeds). Correlation of the quality factors (b) Q_1, (c) Q_2, and (d) $Q_3[3,0]$ with the dynamic aperture for the LHC model with random errors (100 seeds).

6 FUTURE DEVELOPMENTS

Further modules will be added to this program library shortly. We should implement algorithms to extract useful information from noisy data. These techniques are necessary when analysing real measurements; for instance, to obtain the value of the tune from the knowledge of a turn-by-turn set of data. Furthermore, a tool to analyse non-stationary signals using the wavelet transform is under development. Moreover, the possibility of parallelizing the structure of the routines involved in the tracking part of the library is under study. The preliminary results are quite encouraging.

Acknowledgements

We would like to thank Prof. G. Turchetti for his strong contribution in providing the motivation for this work and for his constructive criticism. Special thanks go to W. Scandale, to F. McIntosh for his constant support and help and to V. Ziemann for providing us with the code used for computing the linear part of the lattice.

References

[1] The LHC study group, The Large Hadron Collider: Conceptual Design Study, CERN/AC/95–05 (LHC).

[2] A. Schoch, CERN 57–21 (1957).

[3] G. Guignard, CERN 76–06 (1978).

[4] H. Grote and F.C. Iselin, The MAD Program (Methodical Accelerator Design) Version 8.16. User's Reference Manual, CERN SL (AP) 90–13 (Rev. 4) (1995).

[5] A. Bazzani, P. Mazzanti, G. Servizi and G. Turchetti, Nuovo Cimento, **B102** (1988) 51.

[6] A. Bazzani, Celestial Mech. **42** (1988) 107.

[7] E. Forest, M. Berz and J. Irwin, Part. Accel. **24** (1989) 91.

[8] A. Bazzani, E. Todesco, G. Turchetti and G. Servizi, CERN 94–02 (1994).

[9] A. Bazzani, M. Giovannozzi and E. Todesco, Comput. Phys. Commun. **86** (1995) 199.

[10] M. Hénon and C. Heiles, Astrophys. J. **69-1** (1964) 73.

[11] G. Benettin, L. Galgani, A. Giorgilli and J.M. Strelcyn, Meccanica **15** (1980) 21.

[12] E. Asseo, J. Bengtsson and M. Chanel, *in* Proc. Fourth European Signal Processing Conference, eds. J.L. Lacoume et al. (North Holland, Amsterdam, 1988) p. 1317.

[13] J. Laskar, C. Froeschlé and A. Celletti, Physica **D56** (1992) 253.

[14] J. Laskar, Physica **D67** (1993) 257.

[15] R. Bartolini, A. Bazzani, M. Giovannozzi, W. Scandale, E. Todesco, Part. Accel., in press (1996).

[16] M. Giovannozzi, R. Grassi, W. Scandale and E. Todesco, Phys. Rev. **E52** (1995) 3093.

[17] E. Todesco and M. Giovannozzi, Phys. Rev. **E53**, in press (1996).

[18] E. Todesco, M. Giovannozzi, W. Scandale presented at LHC95, Montreux, Switzerland, October 1995.

[19] E. Todesco, Phys. Rev. **E50** (1994) R4298.

[20] F. Schmidt, SIXTRACK Version 1.2. Single Particle Tracking Code Treating Transfers Motion with Synchrotron Oscillations in a Symplectic Mamer. User's Reference Manual, CERN–SL (AP) 94–56 (1994).

[21] J. Zoll, Reference manual: PATCHY Version 5.03/14, CERN (1995).

[22] H.S. Dumas and J. Laskar, Phys. Rev. Lett. **70** (1993) 2975.

[23] F. Schmidt, F. Willeke and F. Zimmermann, Part. Accel. **35** (1991) 249.

[24] E. Todesco, Physica D, in press (1996).