PLATO: A PROGRAM LIBRARY FOR THE ANALYSIS OF 4D NONLINEAR TRANSVERSE MOTION*

M. Giovannozzi†, INFN Sezione di Bologna, Italy
E. Todesco, A. Bazzani, Dipartimento di Fisica Università di Bologna, Italy
R. Bartolini, CERN, 1211 Geneva 23, Switzerland

Abstract

The PLATO (Perturbative Lattice Analysis and Tracking tOols) program, a program library for analyzing four-dimensional betatronic motion in circular particle accelerators is presented.

The routines included in this library provide both the resonant and the nonresonant 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. To ensure the highest flexibility, the code is fully compatible with standard tracking programs commonly used in the accelerator physics community.

1 INTRODUCTION

The discrete formalism of nonlinear mappings, based on the Lie algebraic tools, has been introduced in accelerator physics in recent years [1], and relevant applications have been made (see Ref. [2] for a review). A perturbative approach for nonlinear one-turn maps has been proposed [3–7], generalizing the Birkhoff series to the case of mappings. The theory of nonresonant normal forms allows one to derive analytical expansions for the amplitude-dependent tuneshift and for the phase-space distortion. Recently, resonant normal forms have been developed, providing a classification of the geometry of resonances [8, 9], and have been applied to derive analytical quality factors to optimize the lattice [10]. On the computational side, efficient arbitrary-order codes have been written to automatically evaluate the coefficients of the one-turn map [11] and the perturbative series for generic maps, both in the nonresonant [5, 12] and resonant [12] cases.

Moreover, sophisticated numerical tools borrowed from celestial mechanics have proved to be useful indicators of the nonlinear motion: computation of the Lyapunov exponent [13, 18], frequency analysis[14–16], evaluation of the global dynamics through tune footprints [10, 15], and so on.

In this paper we outline the main features of PLATO (Perturbative Lattice Analysis and Tracking tOols), a code developed for the analysis of four-dimensional nonlinear betatron motion.

2 GENERAL FEATURES

2.1 Main aims

The routines in the program library can be grouped into three categories.
Interface with tracking codes used in the accelerator physics community (MAD [19] and SIXTRACK [20]).
Analysis of tracking data obtained via numerical simulations (element-by-element tracking). These data are processed to evaluate the dynamic aperture, the nonlinear tunes, and the maximal Lyapunov exponent.
Normal forms analysis and related perturbative tools (nonresonant and resonant normal forms, quality factors and resonance analysis).

2.2 Structure of the code

The code is written in standard FORTRAN 77 and structured as a program library of 136 modules. Two main programs are available. The first allows tracking simulations to be performed using sophisticated numerical tools for post-processing. The second carries out perturbative computations on the truncated one-turn map. Help files 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. A long write-up is also available in the form of an ASCII file.

3 LATTICE INPUT

An accelerator is made up of a sequence of magnets. One record is sufficient to describe the physical properties (length, magnetic field, etc.) of each component. A plain sequence of such records is 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.

As the MAD program [19] contains a simple and powerful language to deal with accelerator structures and, in addition, MAD is used to officially maintain the lattices of LEP and LHC, we have decided to use the MAD input data in defining the lattice parameters.
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. 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 our library.
Interface with SIXTRACK Since a translation program

* Work partially supported by EC Human Capital and Mobility Contract No. ERBCHRXCT940480.

† Present address: CERN PS Division

is available to convert a MAD to a SIXTRACK input file, we decided not to develop any routine which uses the SIXTRACK input files. On the other hand, it is possible to read in the polynomial maps produced by SIXTRACK through the Differential Algebra package [11] as well as the tracking-data files. This allows the analysis through the tools implemented in PLATO.

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 obtained data. The linear part of the code is based on the standard transfer matrices formalism. Nonlinear magnets are always considered as thin elements in the kick approximation. This approach allows one to derive an approximate map that is exactly symplectic: this is crucial for simulating hadron machines where the motion is conservative.

The main program contains several routines that perform the following operations.

Dynamic aperture evaluation. The dynamic aperture is the volume of the domain around the closed orbit where the particles remain confined. The numerical computation of the dynamic aperture is very CPU-time consuming. Let ρ_1 and ρ_2 be the nonlinear invariants in the planes (x, p_x) and (y, p_y) respectively, the dynamic aperture reads

$$V(N) = 4\pi^2 \int\int \chi(\rho_1, \rho_2)\; d\rho_1 d\rho_2 \qquad (1)$$

where $\chi(\rho_1, \rho_2)$ is one if the initial condition with nonlinear invariants (ρ_1, ρ_2), is stable for N turns, and is zero elsewhere. In Ref. [17] we have discussed different approaches to compute the integral (1) and these algorithms are included in the library. A typical graphic output is shown in Fig. 1, where the plane (x, y) of initial conditions $(p_x, p_y$ are set to zero) is scanned along a polar grid. Different markers correspond to stable or unstable particles.

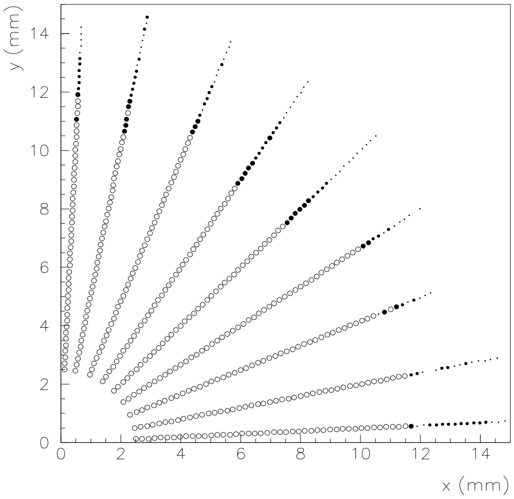

Figure 1: Long-term diagram for a 4D LHC model; particles stable for at least 10^5 turns (empty circles) and lost between 10^2 and 10^5 turns (black circles of decreasing size).

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 stability. During the last decade, efficient methods to compute the tune, have been proposed [14–16] and they are implemented in PLATO. The indicator of 'chaoticity', based on the variation of the instantaneous tune over the orbit [15, 18], is also implemented.

Tune footprint. The tune footprint [10, 15] of a magnetic lattice 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, and plotting the result in the tune plane. This technique gives a vivid picture of the resonance net which governs the stability of the system. Fig. 2 shows a typical tune footprint, generated by PLATO, relating to a simple LHC model [10].

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 [13, 18]. The evaluation of the maximal Lyapunov exponent through the nearby-particles method and the renormalization technique is implemented in the program library.

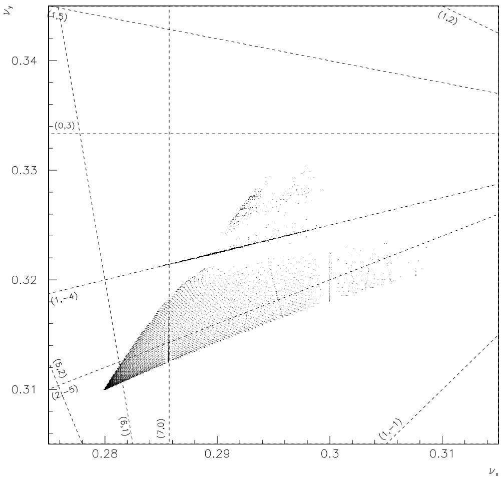

Figure 2: Tune footprint and resonance lines up to order 7 for the LHC cell lattice with random errors.

5 NORMAL FORMS ANALYSIS

A wide set of routines in PLATO is dedicated to the computation of nonresonant and resonant normal forms [12] and other related quantities [6, 8, 10]. Below we will describe the most important quantities that can be computed for a generic complicated lattice.

Truncated one-turn map. The complex coefficients 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.

Nonresonant normal forms. Once the map coefficients are stored, the conjugating function Φ that transforms the one-turn map into its normal form **U** [3–7] can be calculated. A detailed survey of the algorithms used to evaluate the normal form series are given in Refs. [6, 12]. The Hamiltonian is a function of the amplitudes (ρ_1, ρ_2) in the normalized space only. An analytic expression for the nonlinear tunes is given by the derivative of the Hamiltonian with respect to the amplitudes:

$$\nu_{x,y}(\rho_1, \rho_2) = \frac{\partial}{\partial \rho_{1,2}} h(\rho_1, \rho_2).$$

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. 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:

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

The coefficients of these Hamiltonians give important information about the resonance strength , the position of the resonance line in the space of invariants, the width of the resonance and the eigenvalues of the fixed lines.

Double-resonance normal forms. In this case one selects two different resonances $(q_1\nu_x + p_1\nu_y)$ and $(p_2\nu_x + q_2\nu_y)$. The 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 [8, 9].

Quality factors. In many optimization problems it is necessary to analyse different versions of a lattice to select the one with the best dynamic aperture. An efficient solution is to find a quality factor (QF) having a good correlation with the dynamic aperture and which can be calculated in a short time. The QF can then be used to rate the performance of a lattice [10]. Three QFs based on nonlinear maps and normal forms are directly implemented in the library:

$\underline{Q_1}$ – the norm of the nonlinear part of the map evaluated at the amplitude A;

$\underline{Q_2}$ – the average tuneshift at amplitude A evaluated through nonresonant normal forms;

$\underline{Q_3(p,q)}$ – 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 Ref. [10], where these techniques have been used to propose a sorting method for the LHC. Fig. 3 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.

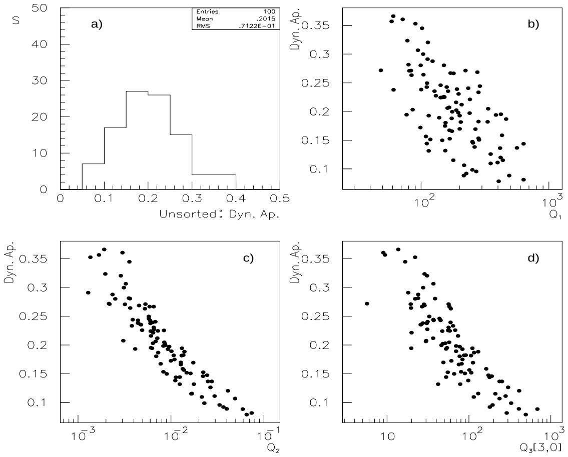

Figure 3: (a) Distribution of the dynamic apertures for an LHC cell lattice with random errors (100 seeds). Correlation between the dynamic aperture and the quality factors Q_1 (b), Q_2 (c) and $Q_3(3,0)$ (d).

6 REFERENCES

[1] A. Dragt and J. M. Finn, J. Math. Phys. **17** (1976) 2649–60.

[2] A. Dragt, Nucl. Instrum. and Methods Phys. Res., A **258** (1987) 339–354.

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

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

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

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

[7] E. Forest and J. Irwin, in: 'Non linear problems in future particle accelerators', edited by W. Scandale and G. Turchetti, World Scientific, Singapore (1990) 46–52.

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

[9] E. Todesco, Physica **D95** (1996) 1–12.

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

[11] M. Berz, Part. Accel. **24** (1989) 109.

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

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

[14] E. Asseo, J. Bengtsson and M. Chanel, *in* Proc. Fourth European Signal Processing Conference, eds. J.L. Lacoume, A. Chehikian, N. Martin, J. Malbos (North Holland, Amsterdam, 1988) p. 1317.

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

[16] R. Bartolini, A. Bazzani, M. Giovannozzi, W. Scandale, E. Todesco, Part. Accel. **52** (1996) 147–77.

[17] E. Todesco and M. Giovannozzi, Phys. Rev. **E53** (1996) 4067–76.

[18] E. Todesco, M. Giovannozzi, W. Scandale, Part. Accel. in press (1997).

[19] H. Grote and F.C. Iselin, CERN SL (AP) 90–13 (Rev. 4) (1995).

[20] F. Schmidt, CERN–SL (AP) 94–56 (1994).