International Journal of Modern Physics C, Vol. 6, No. 5 (1995) 651–661

GIOTTO: A CODE FOR THE NONLINEAR ANALYSIS OF AREA-PRESERVING MAPPINGS*

G. SERVIZI
Department of Mathematics, University of Bologna, P.za di Porta S. Donato 1
Bologna, I-40126, Italy

D. BORTOLOTTI, E. TODESCO
INFN, Sezione di Bologna, Via Irnerio 46
Bologna, I-40126, Italy

M. GIOVANNOZZI
CERN, PS Division
Geneva, CH 1211, Switzerland

M. N. VRAHATIS
Department of Mathematics, University of Patras
Patras, GR-26110, Greece

Received 20 June 1995

An interactive code for the analysis of nonlinear area-preserving mappings is described; several facilities allow the user to draw phase portraits, make zooming, and use colors. The perturbative approach of normal forms and all the standard tools for the analysis of the nonlinear dynamics (Fourier spectra, Lyapunov exponents, fixed points ...) are implemented in a user-friendly graphic environment based on X-window and OSF-Motif. Both simple models and more involved mappings which describe the betatronic motion in a particle accelerator can be implemented.

1. Introduction

In the study of dynamical systems of low dimensionality the geometry of the orbits is a crucial aspect, and the real time visualization obtained through workstations allows one to detect features that are beyond the possibility of analytical investigation. The Poincaré section reduces the dimensionality and brings an Hamiltonian system with two degrees of freedom to a map in a two-dimensional Euclidean space. If the model is itself a mapping, then the computation is usually so fast that both the orbits can be visualized in real time by a simple mouse click to give the initial condition, and also many dynamical indicators such as the winding number or the Fourier spectrum can be interactively computed to obtain a better phenomenological and theoretical understanding of the displayed orbits.

*Work partially supported by EC Human Capital and Mobility contract Nr. ERBCHRXCT940480.

Several topological structures such as closed curves, chains of islands, and chaotic regions appear in two-dimensional maps[1]: due to the hierarchical structure of Hamiltonian nonintegrable systems, a very useful possibility offered by computer graphics is zooming in on a given region, as well as using the colors to mark different orbits. Beyond the direct numerical evaluation of iterates of the mapping, the perturbative tools of normal forms[2] can be implemented: a normal form is a representation of the map where the symmetries are made explicit by a change of coordinates and allows a fast evaluation of the relevant dynamical parameters. The normal forms have been successfully used to analyze the highly complex maps describing the lattice of particle accelerators, and to optimize them analytically[3]; their inclusion in a graphics environment proves to be quite useful.

We have therefore developed a software package "GIOTTO"[4] which is a graphic interactive program for the analysis of two-dimensional discrete dynamical systems of academic interest or arising from applied physics, basically from beam dynamics. GIOTTO allows direct visualization of trajectories of nonlinear mappings of the plane onto itself, as well as zooming on any rectangular region of interest in the plane. It also includes several algorithms for the computation of characteristics of the mapping such as: the localization of fixed points[5]; the conjugation with normal forms[6]; the computation of the winding number, of the Fourier spectra, and of the Lyapunov exponent[7]; the visualization of the homoclinic tangles emanating from hyperbolic fixed points[8]; and the search for bifurcations of fixed points. GIOTTO can analyze both simple built-in maps such as the conservative Hénon mapping,[9] and user programmed maps; complicated one-turn maps relative to magnetic lattices can be provided through standard CERN codes such as MAD[10] and SIXTRACK.[11] GIOTTO can also be a very useful pedagogical support in teaching dynamical systems to students with a very limited programming background.

GIOTTO follows the same philosophy as other codes for the analysis of generic dynamical systems such as DSTOOL[12]: it is based on window managers and graphic servers like the `X11` and `OSF/Motif` that allows one to develop user-friendly application programs; other requirements are the high portability and an easy installation with a very limited number of files. Contrary to DSTOOL, GIOTTO is explicitly build for the analysis of area-preserving mappings which arise from accelerator physics. Therefore, it includes some specific features such as the implementation of the perturbative tools of normal forms and an interface with the CERN standard input files for magnetic lattices. Nevertheless, GIOTTO can also be used to analyze generic dynamical systems (either mappings or flows) whose only constraint is a two-dimensional phase space.

The "name of the game" GIOTTO is not an acronym but refers to the famous Italian painter who was known to draw perfect circles by hand, in the same way as nonresonant normal forms conjugate closed orbits of nonlinear maps with circles. In the next sections we will outline the most relevant features of the code, referring the reader to the User's Manual[4] or to the long write up `giottoHandbook` provided with the program itself for more detail.

2. Area-Preserving Mappings

A mapping of the plane is a recursive transformation of the plane onto itself of the type

$$\mathbf{x}^{(n+1)} = \mathbf{F}(\mathbf{x}^{(n)}) \tag{1}$$

where $\mathbf{x} = (x,\, p)$ is a vector in $\mathbf{R}^2$. The orbit of a mapping is the set of N iterates which can be obtained by the successive application of $\mathbf{F}$ to the initial condition.

GIOTTO primarily deals with area-preserving (or symplectic) mappings, i.e., mappings whose Jacobian determinant is identically equal to one; this is not, however, a mandatory requirement for user supplied mappings. All the built-in mappings have an elliptic fixed point in the origin, and can be written as the composition of a rotation of a winding number ω, plus a nonlinear perturbation:

$$\begin{pmatrix} x' \\ p' \end{pmatrix} = \begin{pmatrix} \cos\omega & \sin\omega \\ -\sin\omega & \cos\omega \end{pmatrix} \begin{pmatrix} x \\ p + h(x) \end{pmatrix} . \tag{2}$$

More complicated mappings can be defined through different procedures, which are outlined in Sec. 4.

In the neighborhood of a fixed point, one can conjugate the nonlinear map to a simpler map which explicitly shows some symmetries and motion invariants: the normal form $\mathbf{U}$. The conjugation is carried out through a nonlinear function $\mathbf{\Phi}$ which transforms $\mathbf{F}$ to $\mathbf{U}$ up to a given order in the powers of the coordinates, which is the perturbative parameter. Close to an elliptic fixed point, one can build two types of normal forms: a nonresonant normal form, which commutes with continuous rotations, or a resonant normal form, which commutes with discrete rotations.[3] In the first case $\mathbf{U}$ is an amplitude-dependent rotation, and in the normal form plane the distance to the origin is the nonlinear invariant since the orbits are circles. In both cases $\mathbf{U}$ can be written as the Lie series of an interpolating Hamiltonian whose level curves (projected in the original plane through the function $\mathbf{\Phi}$) provide the analytical approximations of the orbits of $\mathbf{F}$. The normal form algorithms which are implemented in GIOTTO only deal with area-preserving mappings.

Given an orbit, one can define the tune (or winding number) as the average phase advance per iterate; the tune can also be computed as the position of the main peak of the Fourier transform. We refer to general literature on dynamical systems for the algorithms for computing the fixed points,[5] for the definition of Lyapunov exponent,[7] homoclinic tangles,[8] and for further informations on tune estimates and normal forms.[3]

3. General Features of the Code

3.1. *Technical survey*

GIOTTO is based on the `X11-Window` manager and on the `OSF/Motif` graphic interface; there are no portability problems as long as workstations have the following software products installed:

- a UNIX or UNIX-like operating system;

- a standard C compiler;
- a standard FORTRAN compiler;
- the `X11` server (Release X11R5 or higher);
- the `OSF/Motif` graphic interface (Release Motif1.2 or higher).

GIOTTO has no heavy RAM requirements; they can be reduced to meet smaller workstation's performances by changing the value of the maximum number of windows that are simultaneously managed on the screen. GIOTTO could instead occupy a lot of disk space during a working session, but this depends essentially on how it is used.

3.2. *Lay-out of the screen*

GIOTTO has an X-windows environment; the keyboard is used to give numerical data and character strings, and the mouse to give initial conditions and to drive the program through the associated widgets. The windows created by GIOTTO can be classified in three main categories:

- Control windows: they determine the actions taken by GIOTTO.
- Graphic windows: they show either phase plots or the related dynamical indicators; any window has a data set attached to it, in such a way that, when many windows are displayed together on the screen, each one can be modified independently.
- Dialog and Information windows: they both set up a colloquy between GIOTTO and the user, but only the former wait for an answer.

The graphic windows have a fixed size which can be chosen by the user, before the execution, by modifying a parameter in the `.giotto` file (see Appendix). When the program starts three windows are displayed on the screen (see Fig. 1):

- A control window called "GIOTTO'S MENU BAR": this is the "main menu" window that contains all the widgets relative to the different actions that the user can perform. Widgets are grouped in a hierarchical structure of pulldown menu cascades which makes them easily accessible also to the inexperienced user. A flow-chart of the menu bar is shown in Fig. 2.
- An semaphore-shaped information window that keeps the user aware of what is the state of the program: green light (idling), red light (working), yellow light (waiting for data input), and all the three lights together (error message). Clicking the mouse inside the semaphore causes a help window to be opened on the monitor screen.
- A graphic window "GIOTTO_01", which is a drawing area to plot the phase portrait of the map.

3.3. *Getting help!*

GIOTTO helps you in many ways. The rightmost cascade button gadget in the main menu window gives a complete review of all the features of the program;

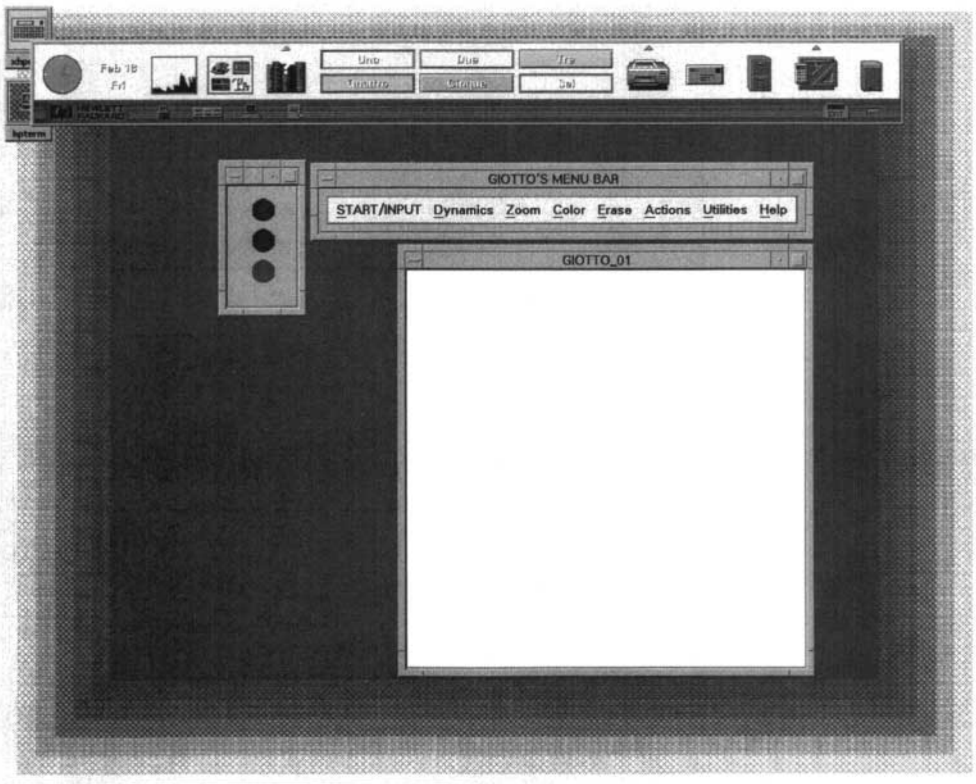

Fig. 1. Lay-out of the screen of GIOTTO: main menu bar, semaphore, and graphic window.

moreover, we already spoke of the semaphore window itself, with the related help window. Furthermore, as a general rule, any window appearing on the screen has a help button inside.

4. Map Input

Map input is driven by the START/INPUT button in the menu bar. Some maps are already implemented in the program, such as the conservative Hénon mapping which modelizes 2D betatronic oscillations of a linear machine with a sextupole kick[9]:

$$\begin{pmatrix} x' \\ p' \end{pmatrix} = \begin{pmatrix} \cos\omega & \sin\omega \\ -\sin\omega & \cos\omega \end{pmatrix} \begin{pmatrix} x \\ p + x^2 \end{pmatrix} \tag{3}$$

Other relevant input options are the followings:

- User programmed map: the user is asked to supply a FORTRAN or a C code which contains the map;

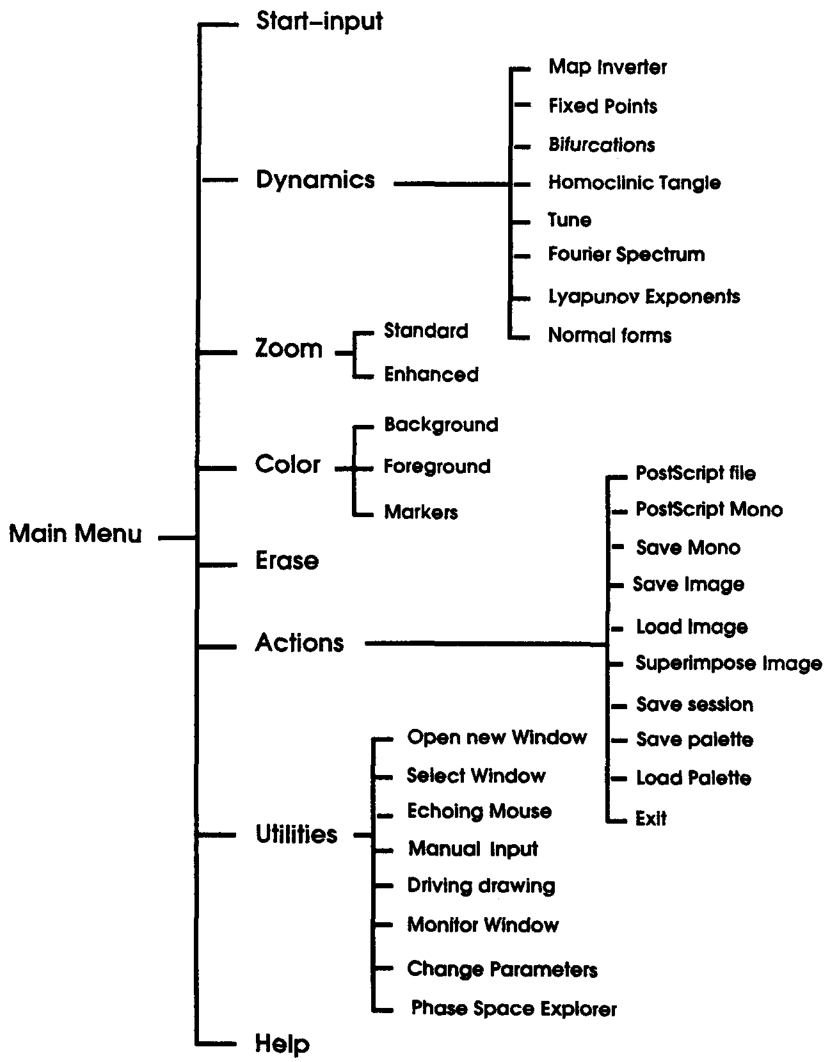

Fig. 2. Flow-chart of the Main Menu Bar.

- Polynomial (read from file) map: the input map is a polynomial whose coefficients are read from a file provided by the user. Also maps generated through SIXTRACK (see Ref. 11) can be read;
- MAD input file: one can analyze a lattice given as a file generated by the STRUCTURE command of MAD.[10]

5. Drawing Phase Portraits

Once a map is chosen for a graphic window, one can draw and erase orbits, change color and make zooming according to the following options:

- Trajectories are drawn for any initial condition which is clicked on the window itself; 2000 iterations are plotted as default; this value can be changed through the START/INPUT menu, which also contains buttons to change all the defaults of the program.
- Trajectories can be erased by clicking the Erase button; different options are provided: one can erase only the last trajectory, several trajectories at the same time, or all the picture.
- Both the drawing and the background color can be selected using the Color option; a wide variety of colors is available, both built-in and user-defined.
- Using the Zoom button, one can select a square region with sides parallel to the axis (standard zoom) or a generic rectangular region (enhanced zoom) of the graphic window and enlarge it to see in more detail the topology of the orbits. A zooming of the phase space of the Hénon map for $\omega/2\pi = 0.255$ is shown in Fig. 3.
- The Actions pulldown menu contains a list of options which allow the user to save and load pictures and sessions.
  - GIOTTO can produce printable PostScript file of each graphic window both in black and white and in color by activating the PostScript mono of the PostScriptfile button.
  - GIOTTO can save pictures in binary files and load them by activating the Save Image and Load Image buttons.
  - GIOTTO can save all the data relative to a working sessions (i.e., the number of displayed windows with their pictures and the associated maps) by activating the Save session button. The file name is `GIOTTO_SAVE_nm`, where `nm` is a two digits string; to recover a session file you have to type **`giotto nm`**. Files saved by GIOTTO are not compressed, and can be exported to any other installation of GIOTTO; the maximum amount of mass storage required is about 60 kbytes/window.
- The Utilities pulldown menu contains a miscellaneous of the following options.
  - Echoing mouse: the position of every click of the mouse on a graphic window is printed on the standard output. This can be useful to locate structures in phase space.
  - Manual input: the initial condition can be given through the keyboard as two real numbers.
  - Monitor window: a window containing the information relative to the current window is displayed.
  - Change Parameters: this allows one to change the parameters of the map associated to the current window.
  - Phase Space Explorer: this option provides a very fast and effective way to explore the phase portrait and its dependence on the map parameters.

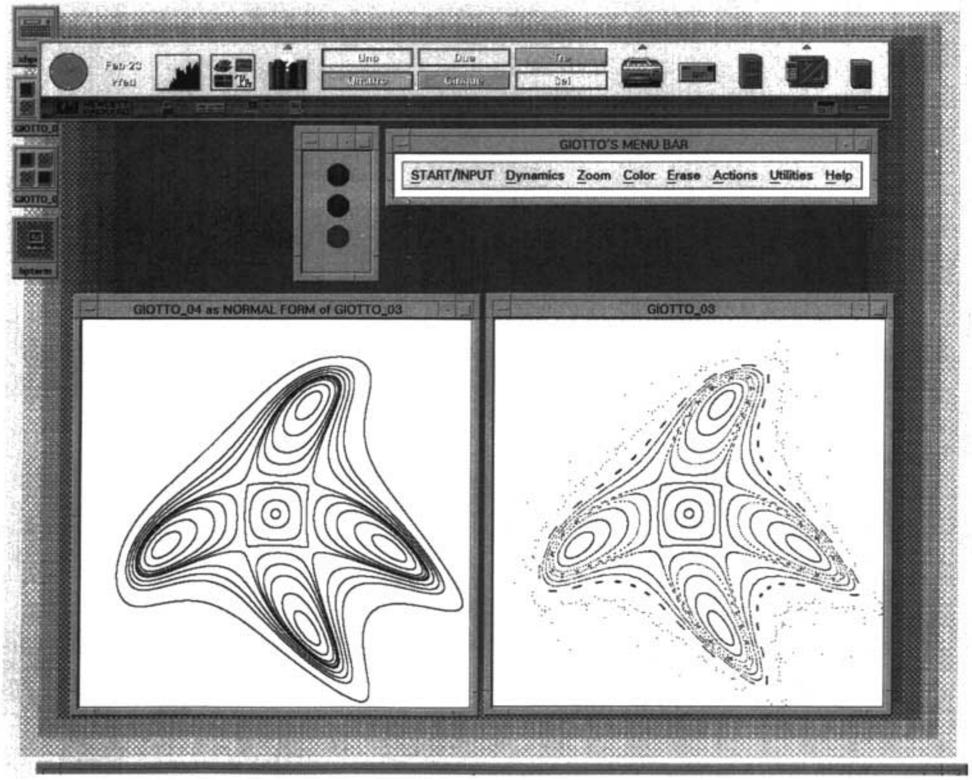

Fig. 3. Zooming of the Hénon map at $\omega/2\pi = 0.255$.

Finally, it must be pointed out that GIOTTO does not store the data relative to all the trajectories, but only keeps in memory the array of 512×512 pixels (low resolution installation) or 1024×1024 pixels (high resolution installation) relative to the graphic window.

6. Nonlinear Analysis

Here we give a list of the different analytical and numerical tools which are implemented in GIOTTO; they are all located in the Dynamics button in the menu bar.

- Fixed points of arbitrary order can be computed starting from an initial guess, using a bisection method.[5] The position, the period, the stability, and the eigenvalues of the fixed point are evaluated.
- Fixed points can be followed in their dependence on the map parameter up to the birth of Bifurcations.

- The Fourier spectrum of the iterates of the mappings is evaluated through an FFT and displayed.
- The nonlinear tunes in their dependence on the initial condition are evaluated using the numerical method of the average phase advance or the perturbative approach of normal forms.[3]
- The normal forms series[3] relative to the implemented map are evaluated; approximation of the dynamics provided by the perturbative series can be displayed and compared with the original phase space. Both resonant and nonresonant normal forms are implemented; in Fig. 4 the resonant normal form relative to the Hénon map close to resonance four is shown.
- The homoclinic tangle option allows one to draw the invariant manifolds which emanate from the hyperbolic fixed points computed through the fixed point option, and to analyze the relation with the stability boundary.[8]
- The Lyapunov exponent can be evaluated and displayed as a function of the number of iterates using the method of the nearby trajectories or the tangent map method.[7]

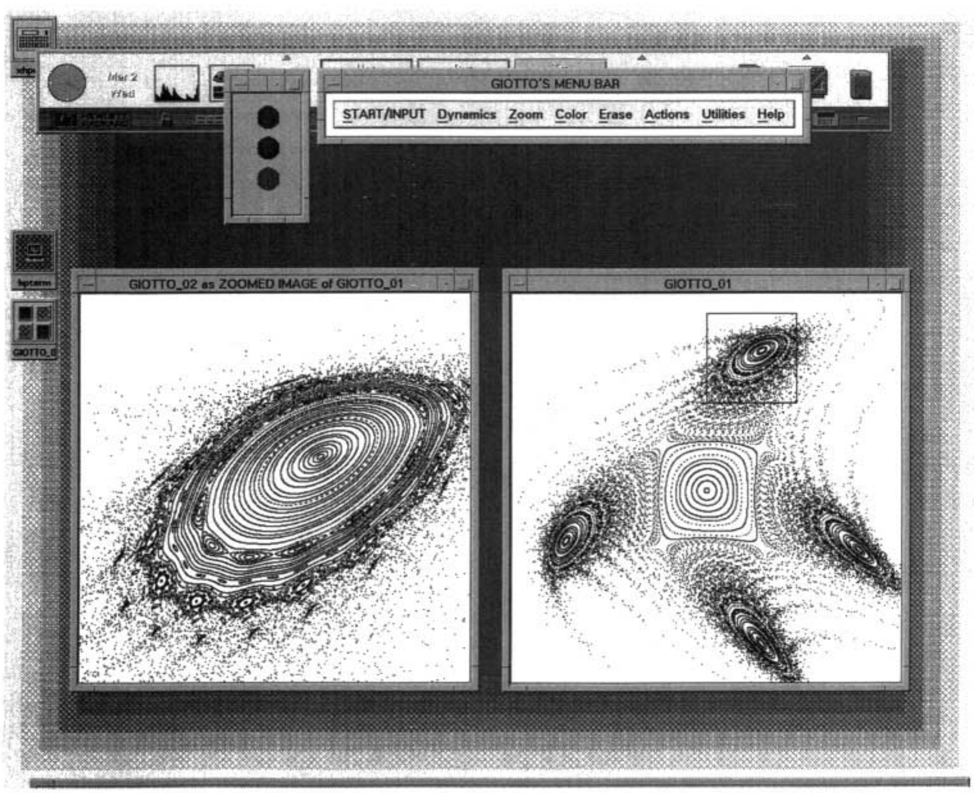

Fig. 4. Phase portrait of the Hénon map at $\omega/2\pi = 0.252$ and approximation provided by the truncated resonant normal form.

7. Future Developments

Other tools will be implemented in a short time: among them, we cite the evolution of the phase portait and of the dynamical indicators relative to a distribution of initial conditions, and the possibility of defining models through generic symbolic Hamiltonians, Lagrangians, and ODE's, choosing the integration method.

The ultimate challenge is to generalize the code to the analysis of four-dimensional symplectic mappings which modelize the complete betatronic motion: whilst the algorithmic part is already available,[6] the major difficulties come from the visualization of a four-dimensional phase space and from the necessity of having a high number of iterates for each orbit, which affects the interactivity of the program for complicated models.

Acknowledgments

We would like to thank Prof. G. Turchetti for providing motivations to this work and for constructive criticism; we also would like to acknowledge Dr. A. Bazzani for important contributions.

References

1. J. D. Meiss, *Rev. Mod. Phys.* **64**, 795 (1992).
2. A. D. Bryuno, *Russ. Math. Surv.* **43**, 25 (1988).
3. A. Bazzani, G. Servizi, E. Todesco, and G. Turchetti, *CERN Yellow Report* **94-02** (1994).
4. D. Bortolotti, G. Servizi and E. Todesco, *CERN SL Internal Note (AP)* **94-33** (1994).
5. M. N. Vrahatis, G. Servizi, G. Turchetti, and T. C. Bountis, *CERN SL (AP)* **93-06** (1993).
6. A. Bazzani, M. Giovannozzi and E. Todesco, *Comp. Phys. Comm.* **86**, 199 (1995).
7. G. Benettin, L. Galgani, A. Giorgilli, and J. M. Strelcyn, *Meccanica* **15**, 21 (1980).
8. M. Giovannozzi, *Physics Lett. A* **182**, 255 (1993).
9. M. Hénon, *Q. Appl. Math.* **27**, 291 (1969).
10. H. Grote and F. C. Iselin, *CERN SL (AP)* **90-13** (1990).
11. F. Schmidt, *CERN SL (AP)* **94-56** (1994).
12. J. Guckenheimer *et al.*, "DSTOOL: A dynamical system toolkit with an interactive graphical interface", Center for Applied Mathematics, Cornell University (1992).

Appendix

GIOTTO's tar file `giotto.tar` can be obtained by sending a message to the e-mail address `servizi@bo.infn.it`; it is advisable to put it in a new, empty, directory to avoid interactions with existing files; once it has been expanded through the standard UNIX command `tar -xf giotto.tar` a number of files are created in the current directory. To perform installation you have to type `./INSTALL`.

Two questions are issued by the procedure. Firstly, you are inquired on which workstation you are working: this is done to take care of the small differences between the different UNIX "dialects"; in the present release GIOTTO has been suited for HP 9000/700 under HP-UX A.09.05, Silicon Graphics under IRIX 5.3, AlphaVax under OSF1 V3.2 and DecStations 4000 under ULTRIX 4.3. Next you have to select the resolution of the drawn pictures: 512×512 (low resolution) or 1024×1024 (high resolution).

After installation is completed, GIOTTO has created in the directory where the tar file has been expanded to 11 files, which should not be modified; a 12th file, called `.giotto`, is created in the `$home` directory of the user: this file contains some settings of the code such as the size of the windows and their colors, and can be modified by the user.

The command which starts GIOTTO is `giotto` eventually followed by optional arguments. You'll get a summary of the command line format by typing `giotto help` which has the same effect as reading the `README` file in the GIOTTO's directory.