# Scientific Visualization of Poincaré Maps [1]

Wu, Shin-Ting[1], Sidney P. Campos[1] and Marcus A. M. de Aguiar[2]

[1] *DCA/FEEC/UNICAMP - C.P. 6101*
*13083-970 Campinas - SP*
ting,sidney@dca.fee.unicamp.br
[2] *IFGW/UNICAMP - C.P. 6165*
*13083-970 Campinas - SP*
aguiar@ifi.unicamp.br

Poincaré maps have proved to be a valuable tool in the analysis of several dynamical systems modeled by differential equations. These maps are generated by reducing the continuous flow to a two-dimensional discrete dynamics. From a map it is possible to identify the chaos phenomenon in a system under the influence of an external parameter. If this external parameter is variable, one can study the behavior of the system by interpolating the set of corresponding Poincaré maps. Despite of its usefulness, the computer graphics work carried out so far has been limited to display and plot of Poincaré maps.
In this paper a prototype for the computer analysis of Poincaré maps is described. We show that, from the point-of-view of computer graphics, we can process Poincaré maps as noisy images. This approach not only facilitates the partition of Poincaré maps into regular and chaotic regions but also offers possibilities to visualize the continuous evolution of a system by varying the external parameter. Some results are given to illustrate the functionalities of the prototype.

## 1 Introduction

The behavior of non-linear dynamical systems can be highly unpredictable and strongly dependent on initial conditions and on possible parameters. This kind of unpredictability is known as deterministic chaos and its study is essential to several research areas such as Physics, Biology, Chemistry, Medicine and Economy. A useful tool to identify this phenomenon is provided by Poincaré maps, which reduces the study of a continuous flow to that of a two-dimensional discrete map. A classical example of such map is the so called *stroboscopic map*, in which the position and velocity of a particle (which are continuous functions of the time) are annotated only at certain values

[1] A short version of this paper has already been published in the proceedings of the WSCG conference [2].

of time, say every five seconds.

If the dynamical system depends on a parameter, this dependence will be reflected in the corresponding Poincaré map: each Poincaré map corresponds to an instance of the parameter. If, for a given initial condition, the behavior of the system is unpredictable, the points plotted on the map have a chaotic structure, in the sense that we cannot distinguish any geometric curve shape. On the other hand, if the system is predictable, the distribution is typically regular, i.e. they build well-defined curves on the plane. Through the comparisons between the maps we can also investigate the variation in the behavior of the system subjected to distinct instances of the parameter. However, to our knowledge, the visualization of this variation is done in a very archaic way without any additional computational support.

Since Poincaré maps are given by scattered sampling data, the bottleneck for the advance in the visualization of Poincaré maps is to distinguish automatically the border between the chaotic and regular regions, which is an important issue in Physics. Some existing brute-force techniques based on the calculation of *local Lyapunov exponents* are very time-consuming and inhibits its application[1]. This paper presents a first step towards overcoming this problem. We developed a technique for the following special purposes:

- partition of a Poincaré map into regular and chaotic regions,
- comparisons of Poincaré maps,
- interpolation of Poincaré maps and
- visualization of interpolated Poincaré maps from different points-of-view.

The breakthrough of our work lies on the realization that:

- if there are sufficient sampling data the chaotic regions should be completely filled and the regular regions should contain a set of well-defined curves. However, depending on the disposable computational resources or on the initial parameter values, a typical Poincaré map presents several lacunes, regions not visited by any sampling data, as shown in Fig. 1.
- the interest on the Poincaré maps is restricted mostly to the geometry of the spatial distribution of sampling data and not on the behavior of an isolated point. It means that, even if there is a huge point set, our focus is on the envelopes of these points.

Therefore, we may model a map as a highly noisy image, whose *missed* sampling data are recovered from a space-filling algorithm. After this recovering, it is possible to apply image processing techniques to partition a Poincaré map into regular and chaotic regions. Following this partition, we can contrast two or more Poincaré maps by comparing the envelopes of regions or we can visualize the behavior of a system along the variation of external parameter by interpolating these envelopes.

This paper is organized as follows. In Section 2 a brief explanation of

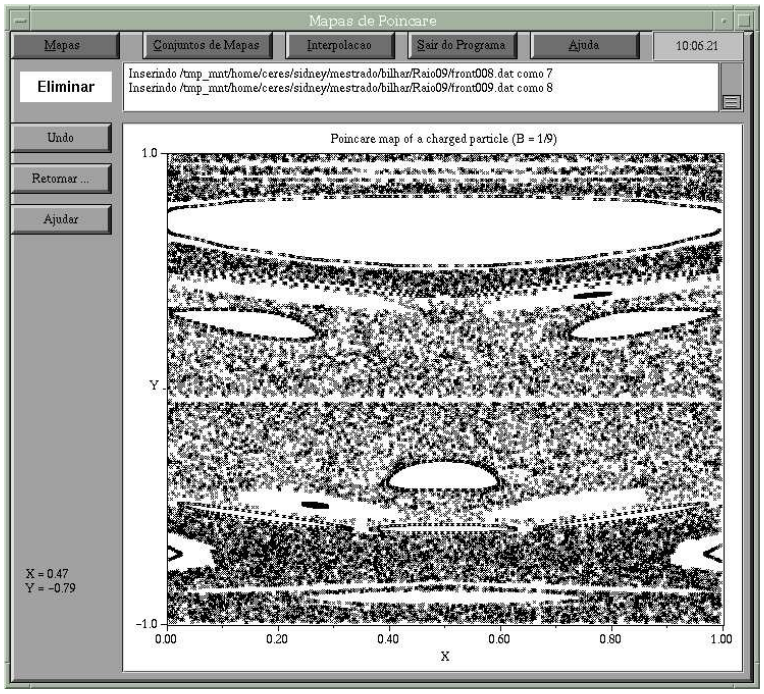



Fig. 1. A Poincaré map

Poincaré maps is given. In Section 3 we show how to deal the problem as an image processing one and to achieve the aiming results.

Section 4 gathers some experimental results obtained. Finally, in Section 5 some conclusions are drawn.

## 2 Poincaré Maps

Poincaré Maps have become a very important tool in the study of dynamical systems in several areas of research. Although its precise definition depends on the particular problem under consideration, the basic idea of a Poincaré Map is to reduce the study of a continuous flow to the study of a discrete map. The idea, due to Poincaré [3,4], originated in his studies of conservative systems with two degrees of freedom.

Consider for instance a particle moving it the x-y plane under the action of forces. Given an *initial condition* $x_0, y_0, \dot{x}_0, \dot{y}_0$, a unique solution of Newton's equation is determined. To describe the particle's trajectory completely we need to show $x(t)$, $y(t)$, $\dot{x}(t)$ and $\dot{y}(t)$. This is usually not very practical, since we cannot visualize this four dimensional *phase space*. However, if the motion is bounded, i.e., it remains in a finite region of the phase space (suppose the particle is attached to hard springs) a 2-dimensional representation of the motion is possible : fix for instance $y_0 = 0$ and $\dot{y}_0 > 0$. Then, propagate the trajectory until the time $t = t_1$ where $y(t_1) = 0$ once again with $\dot{y}(t_1) > 0$. At this instance we save the values of $x(t)$ and $\dot{x}(t)$. The Poincaré Map is exactly this mapping $(x_0,\dot{x}_0) \rightarrow (x(t_1),\dot{x}(t_1))$.

It can be shown that all the basic properties of the flow are reflected in the map. In particular it is easy to see that periodic trajectories of the flow correspond to periodic points of the map. The stability character of the trajectories is also preserved: stable (unstable) trajectories of the flow correspond to stable (unstable) orbits of the map. Unstable trajectories are usually refered to as *chaotic*, since they appear in the Poincaré Map as a sequence of apparently random points covering uniformly certain areas of the map plane. Stable, or *regular*, trajectories, on the other hand, draw simple curves when mapped successively.

As a second example, consider the *Duffing Oscillator* [5], $\ddot{x} + k\dot{x} + cx + \epsilon x^3 = F \cos \omega t$, which represents a dampped nonlinear oscillator subjected to an external periodical force. Instead of plotting a solution in the $(x, \dot{x})$ plane continuously in time, one may generate a map by plotting $(x(nT), \dot{x}(nT))$ for $n = 1, 2, ...$ and $T = 2\pi/\omega$. In this case we use the *natural* period of the external force to construct the map. Once again the basic properties of the flow are reflected in the map.

Further applications on Engineering and on Astronomy can be found in [19] and [20,21], respectively.

## 3 Algorithm

This section presents our algorithm thoroughly. Our proposal intends to aid in the analysis of Poincaré maps regarding the chaotic regions: their presence at each Poincaré map and their evolution along a set of maps.

The algorithm consists basically of the following steps:

- converting each map from object-space to the image-space,
- filling the regular regions on each map,
- detecting the border of the regular regions and
- interpolating the borders of successive maps.

Through this algorithm we transform a set of Poincaré maps into a geometric volume which can be visualized with the aid of several 3D graphics packages. In our case we used OpenGL [15]/Mesa [13].

### *3.1 Conversion*

Up to know, a typical way to analyze Poincaré maps regarding to their chaotic regions is based on their visual appearance. Since our objective is to develop techniques to facilitate this visual analysis, we decide to input Poincaré maps to our program as images consisting of filled and unfilled pixels. Therefore, the first task is to convert the set of points represented in object-space to image-space.

As a main consequence of this conversion, we can use image processing algorithms to segment Poincaré maps. Another advantage of this approach is that, although the Poincaré maps can involve thousand million of points, the complexity of the subsequent processing algorithms is only dependent on the pre-defined image size.

### *3.2 Region Filling*

The main problem of this step and the step presented in Section 3.3 is to find out the location of the *boundary* between regular and chaotic regions. After the representation conversion given in Section 3.1, we may consider the partition of Poincaré maps into chaotic and regular regions as segmentation of their corresponding images.

However, as we already observed, depending on the number of sampling data and the number of initial conditions considered, the patterns of chaotic and regular regions on the Poincaré maps may differ strongly. Although it is visually very easy to distinguish between a regular and a chaotic set of points, it is computationally difficult to devise a generic segmentation algorithm to satisfy all situation. We tried to rectify this deficiency allowing user interactions through a user-friendly interface. It is worth remarking that for the applications we have in mind, the number of Poincaré maps necessary to study a typical parameter interval is of the order of ten and each map has about 10 sets of curves. Therefore, by providing a pleasing look and feel interface, these interactions should not be a tedious task.

The basic idea for dealing with *pattern diversity* is to reduce any *pattern* of regular regions into an automatically identifiable one and, as exclusion, the remaining regions are chaotic ones.

In the regular regions the sampling points are *regularly distributed* and always build curve shape. Therefore, if we remove every well-defined curve on a map, we may consider that the regular regions correspond to the blank ones, which are computationally identifiable.

But, unfortunately, not every blank region on a map corresponds to a regular region, such as the blank line in the middle of the map in Fig. 1. According to experts, this line corresponds to a special family of periodic orbits which is only weakly unstable. As more data are sampled, this blank region tends to disappear. For dealing with this ambiguity, we prefer to let experts select interactively the correct regular (blank) regions.

Finally, in order to separate unambiguously the regular regions from chaotic points, all we have to do is to fill the blank ones. This procedure relies on the fact that the chaotic trajectories are ergodic [6], i.e. they visit all the accessible regions uniformly, up to the boundary. Although there are lacunes due to the incompleteness of the data, the lacunes between the outermost regular curve in each region and the *cloud* of chaotic points (Fig. 1) are also part of the regular regions and the boundary that we seek is close to the chaotic cloud. Of course there will be problems on a finer scale. If there is not enough data we will not be able e.g. to decide whether or not two close regular regions are connected. We refer to this an *experimental* error, that the user could minimize producing more data.

To fill the blank regions a *seed-fill algorithm* is applied. In our case, because of the data incompleteness the size of discontinuities in the chaotic regions may differ widely from each other. Therefore, the success of a seed-fill algorithm depends on the seed size. It should be enough large to not penetrate into the chaotic region and sufficiently small to catch the details of the border shape. Once several attempts for seed size estimation were done without success, we opt for the interactive adjustment with *undo*. There is a menu from which a user can adjust the size of the seed in x- and y-direction.

Knowing that the sampling points are organized in files such that each file contains data belonging to the region of the same type (regular or chaotic one), we implement a graphical user interface on the basis of Xforms [12] that allows the experts through a pointer device to obtain regular regions in two steps:

- to remove the curves in the regular regions; and
- to select the correct associated blank regions.

The device inputs not only a point which is used for searching the file it belongs to, but also the point which is used as the seed position. From the identified files, it is possible to remove all points belonging to regular regions; and from the seed each regular region will automatically be filled.

Fig. 2 was obtained after removing all well-defined curves and selecting three blank regions of the Poincaré map presented in Fig. 1.

As a by-product of the filling procedure the user obtains immediately the total percents of regular and chaotic zones in the map, by simply counting the filled and unfilled pixels. This result is of considerable importance in many areas.

### *3.3 Border Detection*

Once the regular regions are filled, it is trivial to detect their borders and, hence, the borders of the chaotic regions.

For the border detection we applied the morphological filter [18] available in the toolbox Morph [14]. In this case, the borders of an image X is defined as $X$ - $(X \ominus B)$, where $\ominus$ is the erosion operation, - the difference operation and $B$ the *structuring element*. As a result of this step there is a set of borders. Each of these borders correspond to a selected region of a Poincaré map. Moreover, each border is represented by a clockwise, ordered and randomly distributed sequence of points on the border. The

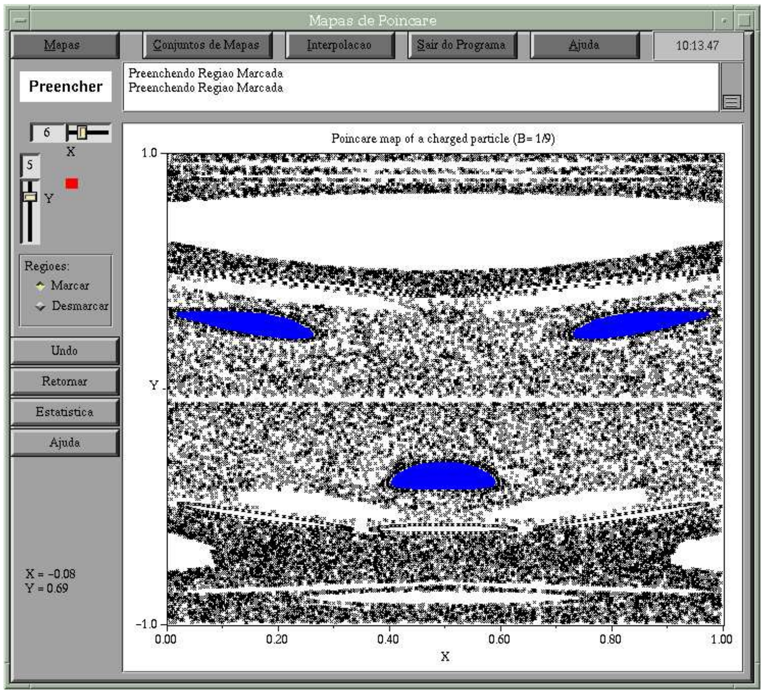



Fig. 2. Poincaré map with 3 selected regular regions

total number of points depends on the concavity of the border.

Repeating the steps described in Sections 3.1, 3.2 and 3.3 on the successive Poincaré maps, sets of borders are obtained. We may interpolate them to investigate the behavior of the system regarding to the variation of the external parameter.

### *3.4 Interpolation*

Obtaining the borders of regions on each Poincaré map, we can reduce the interpolating problem to a geometric surface reconstruction problem, where each set of borders corresponds to a cross section of constant $z$ value and each border to a contour in a geometric reconstruction.

Our prototype relies on the use of a set of functions implemented by Pedrini [16]. These functions are based on the heuristic triangulation method described in [17]. To be self-contained an outline of this algorithm is presented.

The algorithm can handle multiple-branching (1:m and n:m mapping) with non-convex contours. In order to reduce the multiple-branching with non-convex contour problem into a single-branching with convex contour problem with well-known solution, Ekoule et al. suggest to preprocess the contours in two steps:

- to *merge* multiple contours into one contour; and
- to *deform* a non-convex contour into a convex one.

This method relies on the hypothesis that the convex hull of the contours of the two successive interpolated curves are similar in shape and orientation.

From two successive single-branching convex contours, it is possible to generate an ordered list of edges, and then the set of triangular patches, on the basis of the *minimum edge-length* criterion.

To merge multiple contours into a unique closed contour, the centroids of each contour are computed. By linking these centroids an unique closed contour is obtained. Notice that this contour encloses an inner region which may also be covered by triangular patches after its centroid is determined. Furthermore, observe that if the centroids are collinear the inner region is degenerated to a segment.

If the resulting outer contour is not convex, it may be deformed topologically into a convex one by *projecting* each point of the contour on its convex hull. This projection consists basically in distributing at the same rate the points in the *concavity* on the corresponding arc of the convex hull.

To illustrate the idea let us consider three contours. In Fig. 3.a the centroids $B_1$, $B_2$ and $B_3$ of these contours are shown. In Fig. 3.b these centroids are linked and the intersecting points $c$, $d$, $h$, $i$, $m$ and $n$ are computed. There are two contours: the outer ($abcdefghijklmno$) and the inner ($hdcnmi$) one. We may interpolate the inner contour by triangulated patches after computing its centroid as shown in Fig. 3.c. Once the outer contour is nonconvex we *project* the points in the *concavity* on its convex hull. Observe that the rates $bc'$:$c'd'$:$d'e$, $gh'$:$h'i'$:$i'j$ and $lm'$:$m'n'$:$n'o'$:$o'a$ are, respectively, equal to $bc$:$cd$:$de$, $gh$:$hi$:$ij$ and $lm$:$mn$:$no$:$oa$.

## 4 Experimental Results

Our prototype is written in C language and it is runnable in several Unix platforms, such as Solaris (Sun) , AIX (IBM) and FreeBSD (PC). This multi-platform characteristic is due to the use of three portable libraries:

- Xforms [12], to build a graphical user interface,
- Mesa [13], to perform geometrical transformations, clipping and rendering, and
- Morph [14], for the morphological operations (edge detection).

We have chosen the following system to demonstrate the functionalities of our prototype: a charged particle moving in the x-y plane and confined to a square box of size 1. The reason we have chosen this particular system is because it models the motion of an electron in the surface of a flat metallic object and it has been analysed both experimentally [7] and theoretically [8–11].

If the particle is not subjected to forces it moves along straight lines inside the square and is specularly reflected when it hits the boundaries. In our model system, however, the particle is also subjected to a

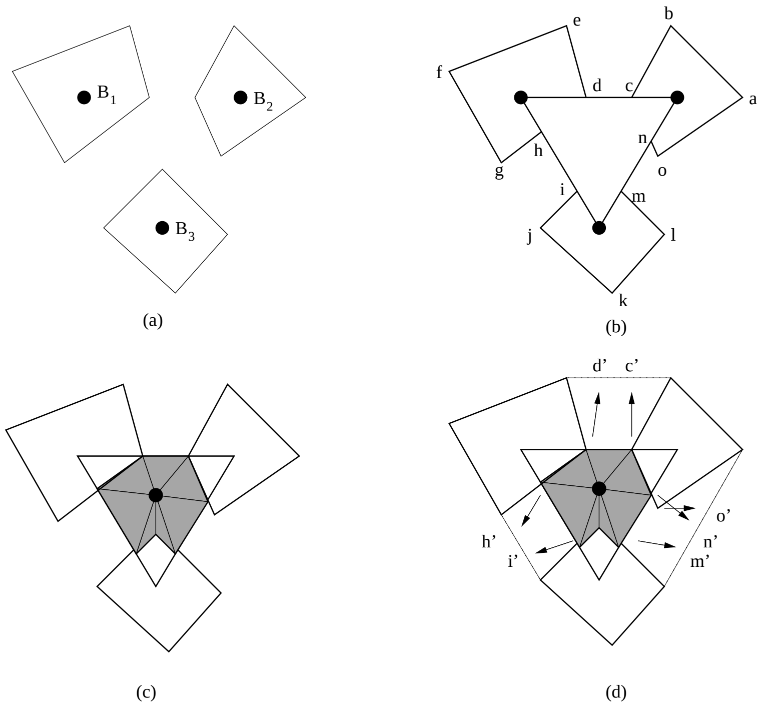



Fig. 3. From multiple-branching to single-branching

constant magnetic field in the z-direction. The effect of the magnetic field is to replace the straight motion of the particle by motion along circles whose radius is given by R $= \sqrt{2E}/B$ where the energy E has been fixed to 1/2 and B is the magnetic field. Since the motion of the particle between two consecutive collisions is trivial, the Poincaré Map is defined by giving information at the collision points only: we plot the arc-length $x$ of the collision point, measured from one of the corners, and $\cos\theta$, the angle with which the particle leaves towards the next collision. These parameters are shown in Fig. 4. Moreover, once the system is invariant under rotations by $\pi/2$, it is enough to show the Poincaré Map for $0 \leq x \leq 1$.

Fig. 5.a, 5.b, 5.c and 5.d present four Poincaré maps corresponding,

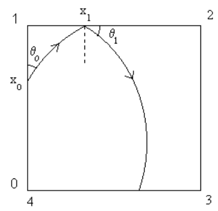



Fig. 4. Charged particle moving inside the square-box

respectively, to the following particular values of the magnetic field: 1/9, 1/10, 1/11 and 1/12 units. Fig 5.e and Fig 5.f show two different views of the interpolation obtained from the three selected regions presented in Fig. 2.

As a second example we consider a more complex type of Poincaré maps generated from the motion of a particle in the x-y plane and subjected to a conservative force $\mathbf{F} = -\nabla V$ where the potential energy is $V(x,y) = \lambda(x^4 + y^4) + x^2Y^2$. This problem has also been extensively studied both in classical and in quantum mechanics [22–24]. The Poincaré maps are obtained by plotting $x(t)$ and $\dot{x}(t)$ every time $y(t) = 0$ with $\dot{y}(t) \geq 0$, as discussed in section 2.

Fig. 6.a Fig. 6.b, Fig. 6.c and Fig. 6.d shows four maps corresponding to $\lambda = 0.09$, 0.10, 0.11 and 0.12 units, respectively. Fig. 6.e and Fig. 6.f present two different views of the interpolation obtained from these four maps. Notice that the distribution of stable and chaotic regions is now much more complex than in the previous example, but the interpolation is still very good.

## 5 Conclusions

This paper is a result of a cooperative work between a research group at the Physics Institute and a research group at the Computer Engineering Department at Unicamp. The main purpose of this work is to use computer graphics tools as alternative mechanism to visualize their data. Although the program has not been extensively used by physicists yet, we believe that it will be a useful tool and we hope that the it can be readily used by the physicists in their analysis in order to evaluate its performance.

From the point-of-view of computer graphics, we faced a challenge to visualize a huge volume of incomplete points in three-dimensions. We presented a relatively simple approach which works well for our purpose. We showed that we can apply complementarily the computer graphics and image processing techniques to obtain an elegant solution.

As our goal is to devise a way to improve the computational aids in the visual analysis of Poincaré maps, there is still need for improvement in several functional modulus of our prototype, especially the rendering flexibilities.

## References

[1] B. Eckhardt and D.M.Yao, Physica **D 65**, 100 (1993) and references therein.

[2] Wu, S.T., et al. Scientific Visualization of Poincaré Maps. *Proceedings of the 5th International Conference on Computer Graphics and Visualization*, Plzen, 1997, pp. 592-601.

[3] H. Poincaré *Les Methodes Nouvelle de la Mechanique Céleste* (Gauthier-Villars, Paris, 1899)

[4] D.W. Jordan and P. Smith *Nonlinear Ordinary Differential Equations* (Clarendon Press, Oxford, 1987, second edition) section 12.5.

[5] D.W. Jordan and P. Smith *i*bidd and references therein.

[6] A.J. Lichtenberg and M.A. Lieberman *R*egular and Stochastic Motion (Springer-Verlag, NY, 1983).

[7] L.P. Lévy, D.H. Reich, L. Pfeiffer and K. West, Physica **B 189** (1993) 204

[8] D. Ullmo, K. Richter and R. Jalabert, Phys. Rev. Lett. **74** (1995) 383

[9] N. Berglund and H. Kunz, J. Stat. Phys. 83, 81 (1996)

[10] M.A.M. de Aguiar, Phys. Rev. E53, 4555 (1996)

[11] M.L. Tiago, T.O. de Carvalho e M.A.M. de Aguiar, Phys. Rev. E55, 65 (1997)

[12] Forms Library for X, *T.C.Zhao and Mark Overmann*, http://laue.phys.uwm.edu/ pub/xforms.

[13] Mesa Library, *Brian Paul*, http://www.ssec.wisc.edu/ brianp/homepage1.html.

[14] Morph Library, *Richard Alan Peters II*, ftp://129.59.100.16/pub/ morph.tar.Z.

[15] J. Neider et al., *OpenGL Programming Guide*, Addison-Wesley Publishing, New York, 1993.

[16] H. Pedrini,*3D Reconstruction from transversal sections of objects (in Portuguese)*, M.Sc. Dissertation, FEE/UNICAMP, Brazil, 1994.

[17] Ekoule, A.B. et al., A triangulation algorithm from arbitrary shaped multiple planar contours. ACM Transactions on Graphics, 1991, **10**(2), 182-199.

[18] Gonzalez, R. and Wintz, P., *D*igital Image Processing, Addison Wesley Publishing Company, 1992.

[19] Zhong, G.-Q. et al., Experimental Poincaré Maps from the Twist-and-Flip Circuit. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, 1996, **43**(10), 874-879.

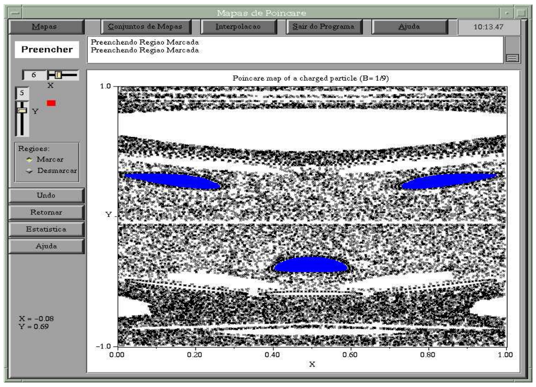
[20] Wisdom, J., ICARUS, 1983, **56**, 51-74

[21] Levison, H.F. Shoemaker, E. M. Shoemaker, C. S., NATURE, 1997, **385(6611)**, 42-44

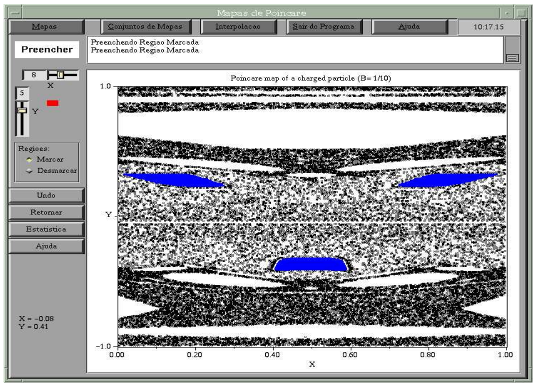
[22] Hose, G., Eckhardt, B. and Pollak, E., Phys. Rev. **A38**, 1989, 3776

[23] Carnegie, A. and Percival, J. C., J. Phys. **A17**, 1984, 801

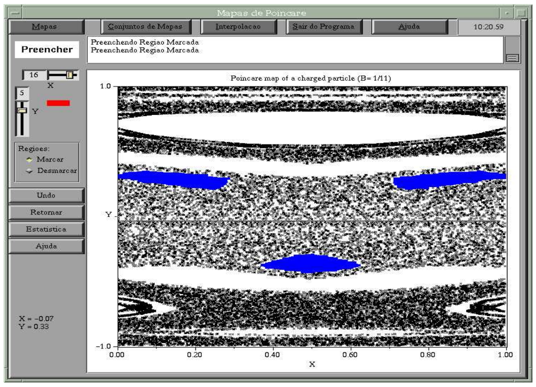
[24] Bajay, F.A., PhD. Thesis, UNICAMP, 1996 (in Portuguese)



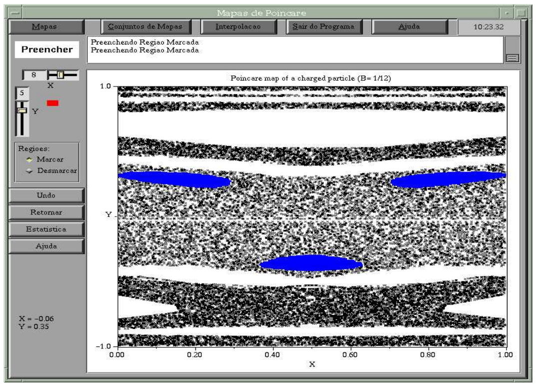

(a)



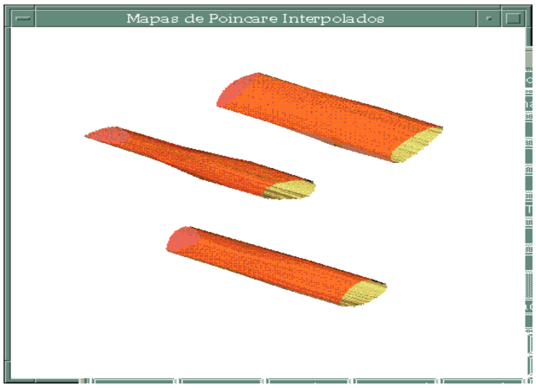

(b)



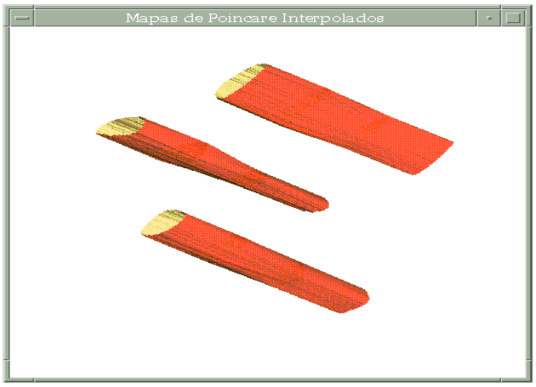

(c)




(d)




(e)




(f)

Fig. 5. Poincaré maps of a charged particle

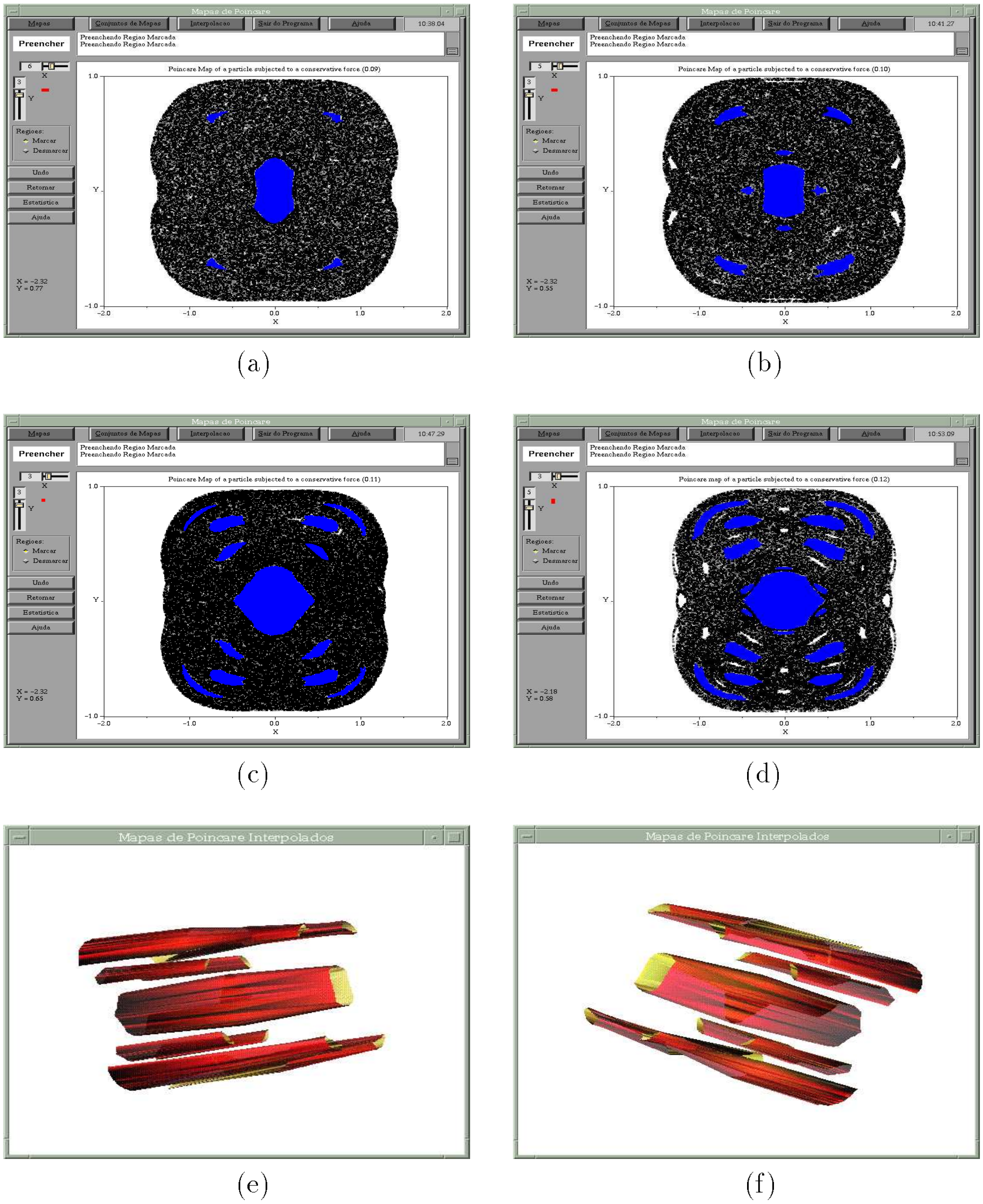



Fig. 6. Poincaré maps of a particle subjected to a conservative force