PCR - a Visualization Tool for Multi-Point Boundary Value Problems

Gábor Domokos*and Randy C. Paffenroth
Institute of Physical Science and Technology and
Advanced Visualization Laboratory
University of Maryland,
College Park, MD 20742

Abstract

In this paper we present a method, and a software based on this method, making highly inter-active visualization possible for computational results on nonlinear BVPs associated with ODEs. The program relies partly on computer graphics tools and partly on real-time computations, the combination of which not only helps the understanding of complex problems, it also permits the reduction of stored data by orders of magnitude. The method has been implemented on PCs (running on DOS) and on the Application Visualization System (AVS) for UNIX machines, this paper provides a brief introduction to the latter version besides describing the mathematical background of the method.

1 Introduction

This paper is intended to describe a method, and a software based on this method, which may be very helpful when presenting, storing or trying to understand computational results on nonlinear boundary-value problems (BVPs) associated with ordinary differential equations (ODEs). There are standard codes available (for example AUTO[Doedel 1986]) solving such BVPs, however, the problem is often that the huge data files created by such a software are hard to store and even much harder to understand. Our method offers help in such cases, minimizing the amount of data and making highly interactive, intelligent visualization available. Assume that the solution of a BVP requires time T and the output data file has size S. Roughly speaking, our method is based on the observation that there exists a very small subset of this data (say, of size $\approx 0.01S$) from which the whole data file can be recomputed within negligible time (say, $\approx 0.00001T$). This computation can be carried out in real-time, and might be in some cases even faster than the retrieval time needed in the case of the complete data set. We outline briefly the underlying deep, but well-known and non-technical ideas from the theory of ODEs and dynamical systems in Section 2.

Athough these ideas might seem sim-

*Permanent address: Technical University of Budapest, H-1521 Hungary ple, it is non-trivial to apply them to particular problems and even less so to build a code which realizes them in an inter-active way. The code PCR has been implemented on Personal Computers running DOS, using Microsoft Quick-Basic 4.5 and on UNIX workstations, using the Application Visualization System (AVS)[1992]. Section 3 gives an overview of the the backbone of both implemetations, section 4 outlines the AVS network, section 5 provides some examples and summarizes the results.

2 Mathematical background

Results in bifurcation problems are usually presented in the form of bifurcation diagrams. We are not aware of any general definition for "bifurcation diagram", in fact, depending on what purpose we have in mind, we may design rather different kinds of bifurcation diagrams. As examples we mention the widely applied method of using Fourier coefficients as coordinates, or, the bifurcation diagrams of Maddocks [1988] for direct stability analysis. Our present goal when designing bifurcation diagrams is threefold:

1./ To create diagrams which are topologically correct, i.e. *one* point of the diagram corresponds to *one* solution of the BVP and vice versa, also that the mapping is continuous.
2./ To create diagrams from which the physical configurations of the BVP can be easily reconstructed.
3./ To minimize the amount of stored data.
All three of the above criteria are met by the set of coordinates ("global coordinates") to be described below.

Global coordinates can be created for explicit ODEs satisfying the Lifschitz-condition, i.e. for ODEs which meet the requirements of the Uniqueness Theorem [Bieberbach, 1923]. Nearly all applications in mechanics are "well-behaved" and fall into this category.(For some rare exceptions see [Domokos 1993a, 1993b].) For such well-behaved ODEs a semi-infinite trajectory is completely and uniquely determined by the initial conditions. For example, consider a linearly elastic cantilever beam, clamped at the left end and free at the right end subjected to the compressive force P (Fig.1).

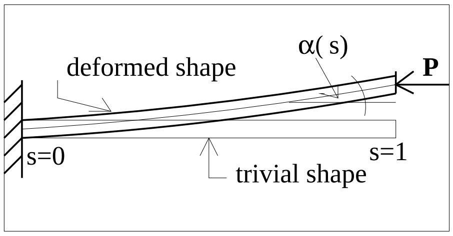

Figure 1: The cantilever beam

The ODE describing the shape of the beam in terms of the slope α as a function of the arclength s was first described by Euler:

$$\alpha'' + P\sin(\alpha) = 0. \qquad (1)$$

The trajectories of this equation are uniquely determined by the three scalars $\alpha(0), \alpha'(0)$ and P (the former being "true" initial conditions, the latter one a parameter). However, not all trajectories are of interest for us, only those which meet the boundary conditions

$$\begin{array}{rrcl} a/ & \alpha(0) & = & 0 \\ b/ & \alpha'(1) & = & 0 \end{array} \qquad (2)$$

expressing zero slope and zero curvature at the left and right end, respectively. For the time being, we ignore the far-end condition (**??**/b) and concentrate on (**??**/a). This condition eliminates one of the "variable" initial conditions as a constant, so all trajectories which *might* meet the boundary conditions can be uniquely represented in the $[\alpha'(0), P]$ plane. This plane has property 1/, i.e. one point of the plane corresponds to one trajectory and vice versa. The scalars $\alpha'(0)$ and P are the *global coordinates* for this BVP, the plane (space) spanned by them will be called the *global representation space* of the BVP. We assume that the BVP has been solved previously by some numerical package and now we are interested in the correct representation of the results. (However, we remark that the idea of the global representation space leads to the construction of a numerical method see [Domokos and Gáspár 1993], [Domokos, 1994].) A finite segment of the bifurcation diagram for the BVP (**??**-**??**) is illustrated in Fig.2 in the global representation space.

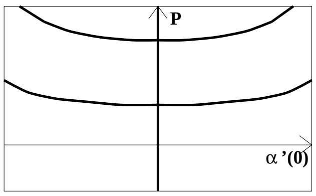

Figure 2: Segment of the global bifurcation diagram for the cantilever beam.

We remark that, although this might seem a trivial example, the strengh of the global coordinates is already displayed by the fact that all (countably infinite) buckling modes of the cantilever can be represented in the same plane in an unambigous way.

Based on the example of the cantilever we can now define the *global coordinates* of a two-point BVP as *those (Cauchy-type) initial conditions and parameters which are not fixed by the boundary conditions at the origin. The global representation space of the BVP is spanned by the global coordinates.*

Although the title of the paper promised "multi-point" BVPs, up to now we discussed only two-point problems. The generalization of these ideas to multipoint problems (including the modeling of elastic trusses and frameworks) is straightforwrad and has been carried out in [Domokos and Gáspár 1993]. In this paper we do not discuss this case, however, we choose this title because the software described here is capable of handling multi-point BVPs.

The bifurcation diagram in the global representation space not only has the advantage that it is topologically correct (Property 1/), it also makes the reconstruction of physical shapes extremely simple (Property 2/). Take any point from the bifurcation diagram and add the remaining constant initial condition (in our cantilever example: $\alpha(0) = 0$) to obtain a complete set of initial conditions. Running an IVP solver with these conditions regenerates the physical shape instantly (even on a small PC the computation time is negligible). This feature of the global bifurcation diagram dictates the basic idea of PCR: Have constantly two windows open on the screen.

One window shows the bifurcation diagram (or a projection of it) and by some device we can locate points on this diagram. The other window instantly displays the corresponding physical shape.

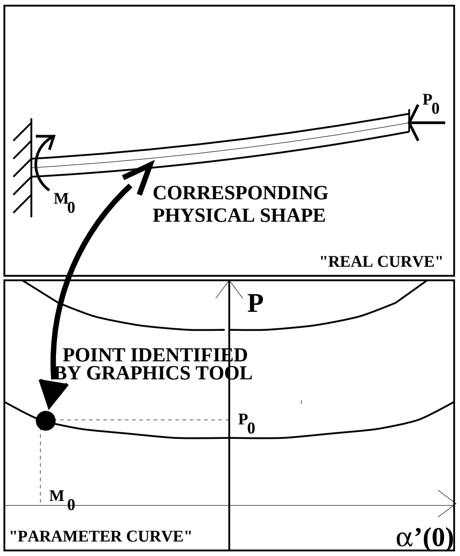

Figure 3: The double-window system of PCR

Such a configuration is shown in Fig.3. Branches of the bifurcation diagram can be illustrated by series of pictures. For two examples showing 3D representation space but 2D physical space we refer to [Domokos 1994].

If the dimension of the global representation space is more than 2 then we seemingly loose the topological correctness since projection to the 2D screen may result in spurious intersection points. However, such points may be eliminated by coloring or rotations, as described in the next section.

We also required that the amount of data stored should be minimal (Property 3/). This is true for the bifurcation diagrams in the global representation space. No matter how we computed the BVP's solutions, if we store the points in the global coordinates then no further information can be omitted. On the other hand, this data is sufficient to re-create the configurations in the physical space, so the large data files created by the BVP solver can be deleted (*after* we extracted the global coordinates of the solution points).

In this section we introduced the global coordinates which are suited optimally to meet the requirements 1-3 listed at the beginning of the section. We illustrated the global coordinates on the example of the cantilever beam and showed how the basic idea of PCR follows from the idea of global representation.

3 Software implementation

The software PCR was developed based on the above described method. PCR has two implementations: a PC/DOS version, requiring 386DX/33 processor and VGA graphic screen (or up). A more sophisticated version is running on the Application Visualization System (AVS) on UNIX workstations, this latter version offers better graphics and more flexibility. In this section we describe the main features characteristic for both versions.

Our software is called PCR, which stands for: $Parameter \rightarrow Curve \leftarrow Real$. (It also represents the initials of one of the authors, in reversed order. It is left to the reader to find out which author we mean.) The input for the software is an $n \times m$ matrix, n (number of rows) denoting the number of data points in the global representation space, m (number of columns) denoting the dimensions of this space. As described in section 2, each of the rows contains the set of initial values which are not set constant by the boundary-conditions at the origin. The data points might be grouped according to branches, this is the case if the solution of the BVP was carried out by a continuation code, or, they might describe arbitrary segments of the bifurcation diagram. The integers m and n can be arbitrary.

PCR also needs an IVP solver routine for the specific BVP, there is an empty "slot" in the AVS network for that routine. After reading in the data file, PCR opens up two graphic windows, one of them showing the bifurcation diagram in the global representation space, one point of this diagram is identified by a small sphere. The physical configuration corresponding to the point identified by the sphere is displayed in the second window. This is the setup we illustrated already in Figure 3.

Standard AVS handles 3D objects, so both figures (the parameter curve and the physical curve) can be instantaneously rotated and shifted in 3D by using the mouse buttons.(In the PC version this can be done by typing in rotation angles.) A point on the parameter curve (bifurcation diagram) can be selected in two ways: Either simply clicking with the mouse on the point (PCR will take the data point *closest*to the pixel you clicked on) or, by selecting a branch identification number and a point identification number on that branch. This latter choice is possible only if the data has been produced by a continuation code, i.e. the data points are arranged according to branches. The selection of branch/point numbers is carried out on small dials on the screen, also operated by the mouse. After having picked a point (either way) the second window immediately displays the corresponding physical shape.

The manipulation of the bifurcation diagram requires more than standard AVS features if $m > 3$. In this case PCR produces a 3D projection of the diagram, and this projection is handled by AVS as a normal 3D object, i.e. it can be moved by the mouse.

Given an n dimensional bifurcation diagram, PCR allows the user to choose any 3 dimensions and display these. Also, as the projection changes, PCR will change the placement of the pointer so that it always sits on the same data point in the diagram. Thus, corresponding points in the various projections can be explored and studied, all the while having the physical configuration, corresponding to the selected point, in view. One of the important features of the global coordinates was that the bifurcation diagram in these coordinates is topologically correct, i.e. if two lines intersect in the diagram, this means a bifurcation point. However, if we project the diagram to some lower dimensional space, spurious intersection points may arise. Figure 4 shows a collection of curves, which are part of a 4D global bifurcation diagram. We can observe 6 intersection points on the 3D image of this curve, they are numerated 1 through 6. (We do not count the intersection points arising from projecting from 3D to 2D, because the human mind is capable of eliminating those immediately.) PCR offers two basic tools to get rid of these points: If the diagram is in 4D then there is a coloring feature according to the 4th coordinate.

If $m > 3$ then the coloring gets complicated, it can be maintained up to $m = 6$ (with a 3D color-space) but for higher dimensions we need a different tool. PCR provides an interface where the user can pick any two integers $i, j < m$ and a real number $0 < \alpha < 2\pi$. These three numbers define a rotation in the $[i, j]$ coordinate plane by angle α, this is carried out by PCR and the new 3D image dispalyed. PCR also offers an option to couple these higher dimensional rotations with mouse buttons. In the previous example we picked a rotation in the $[2, 4]$ plane by $\alpha = 0.05$.

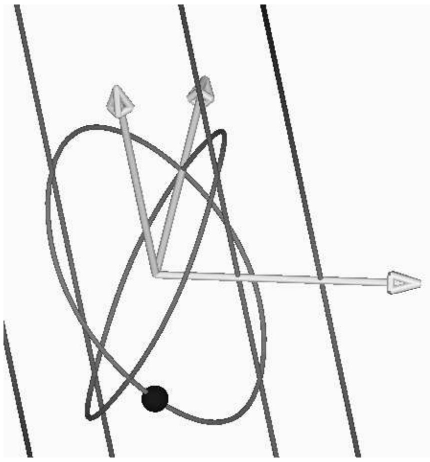

Figure 4: Part of a 4D bifurcation diagram

As Fig.5 shows, the spurious intersection points 5 and 6 have been dissociated by this rotation.

Another noteworthy feature is that, beyond the computation module, there is nothing in our software or our example that restricts the usefulness of the software to the study of ODEs. Any problem which has some type of "parameter space" and an associated "real space", and which has the property that the "interesting" parts of the parameter space happen to be 1D could probably be fruitfully studied in this manner. On the other hand, there is nothing to stop

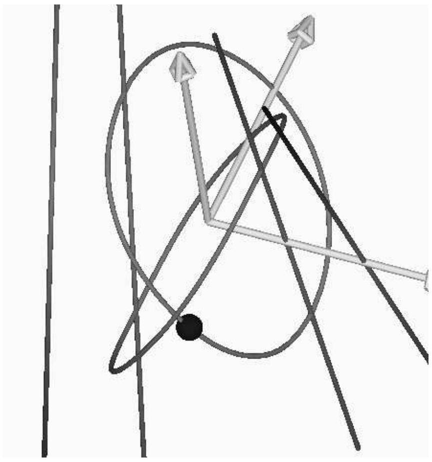

Figure 5: Part of a 4D bifurcation diagram after small 4D rotation.

one from wanting to study 2D data sets. Many of the features of this software have been implemented, in a slightly different setting, in a piece of software called MC^2. It also runs under AVS and is used in the study of 2D solution surfaces.

4 Brief description of the AVS network

AVS is a software package designed for the visualization of data. It provides programming tools and graphic renders that, while having a broad range of flexibility, allow the scientific programmer to ignore some of "dirtier" aspects of graphics programming. In AVS, one uses program segments called "modules" which can be hooked together into a "network", using a graphical interface, to create the desired application. These modules can either be user written or drawn from pre-existing libraries. This section is intended for an advanced AVS user. The network of PCR is displayed in Fig.6 "Geometry viewer", a 3 dimensional renderer, "tube", a line thickener, and "setview", a geometry viewer accessory, are all supplied by the standard AVS library of modules. The rest of the modules were constructed here for this application. The left half of the network is mainly concerned with the production of the solution curve in the real space, using a user supplied computation module such as our "dna data". The right half of the network is concerned with the production of the bifurcation diagram and the control of the pointer on the diagram. Below we describe some of the functions described by the custom modules.

The "plot bifur" module reads in a file containing the global coordinates. It requires that the first entry in the file be the number of dimensions in the global coordinate space. It then reads in an arbitrary number of bifurcation diagram branches, which have been computed by continuation software. The module requires that the first entry of each branch be the number of points in the branch followed by three numbers between 0 and 1, which can be used as RGB values for the colorization of the branch. This module provides 4 output ports. "Diagram" gives an AVS geometry which represents the bifurcation diagram and its colorization information. "Points" provides an integer which is the number of points in the selected branch. "Branch" gives an AVS 1-D float field which contains the selected branch in full untransformed detail, for use by the computation modules. "Plot bifur" provides the bulk of the custom visualization features, such as the colorization and rotations.

5 summary

In this paper we presented the theoretical basis and description of the software PCR, intended for inter-active visualization of results on BVPs. This usage of the "global coordinates" defined in section 2, offers several advantages: It minimizes the size of the stored data, it provides a topologically correct bifurcation diagram and makes the real-time reproduction of the physical configurations possible. PCR offers a variety of inter-active tools to deal with the dataset in the global representation space. Its main feature is the simultaneous viewing of the bifurcation diagram and of the physical configuration corresponding to a user-defined point on this diagram. PCR also offers several tools to understand higher-dimensional bifurcation diagrams, including colorization and rotations in the higher dimensional space. In Fig. 8 we give an interesting example of application: torsional buckling of a uniform, circular rod. PCR was running on data obtained by Li and Maddocks [1994]. The global representation space is 4 dimensional in this problem and is spanned by the initial moments M_1, M_2, M_3 and the force F. Fig 7a shows the M_1, M_2, M_3 projection of this diagram, Fig. 7b shows one physical configuration. We feel that problems of this complexity might be very hard to understand without a tool like PCR.

Acknowledgement

The authors are happy to thank Professor John H. Maddocks for continuous encouragement and support during their work. This research was supported by the Dr Imre Korányi Fellowship and the Hungarian National Science Foundation grant No. F7690 (G.D.) and the ASSERT award from the Air Force Office of Scientific Research (R.C.P.). The software PCR has been developed on hardware supplied from Digital Equipment Corporation under a grant from the Scientific Innovators Program.

References

AVS, Inc. (1992): *AVS User's Guide* Waltham, Massechusetts

Bieberbach, L. (1923): *Differentialgleichungen* Springer, Berlin

Doedel, E. (1986): *AUTO 86 user manual* Caltech,Dept. Applied Mathematics, Pasadena, California

Domokos, G., (1993a): *Can strings buckle?* AMD -Vol 167 (Recent development in stability, vibration and control of structural systems) ASME 1993, Applied Mechanics Division, ISBN 0-7918-1146-8, pp 167-174

Domokos, G., (1993b): *Kinks and strings* AMD -Vol 167 (Recent development in stability, vibration and control of structural systems) ASME 1993, Applied Mechanics Division, ISBN 0-7918-1146-8, pp 191 -196

Domokos, G., (1994): *Global Description of Elastic Bars* ZAMM **74** (4) T289-T291

Li, Y and Maddocks, J.H. (1994): *On The Computation of Spatial Equilibria of Elastic Rods Including Effects of Self-Contact* submitted for publication

Maddocks, J.H. (1988): *Stability and folds* Arch. Rat. Mech. Anal **99** 4 pp 301-308

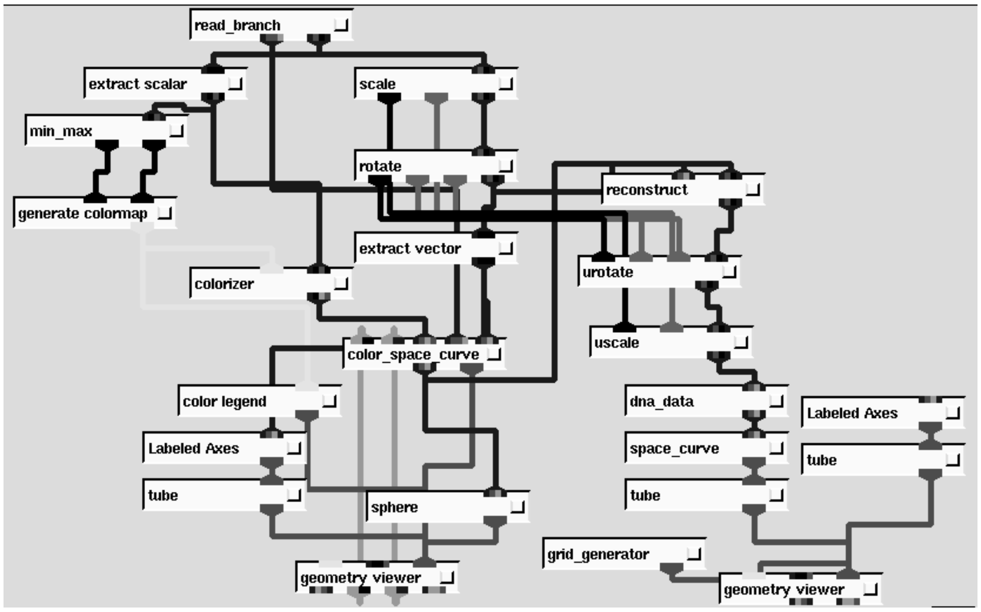

Figure 6: The AVS network for PCR

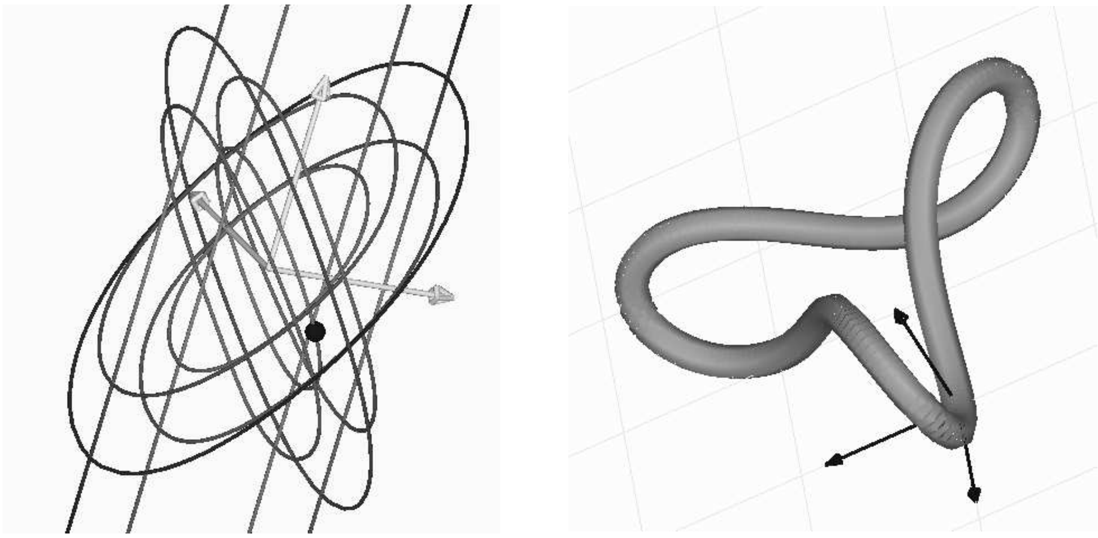

Figure 7: Application of PCR to the problem of torsional ring buckling 7a: Bifurcation diagram 7b: Physical configuration