Concurrent Computations of Invariant Manifolds

Jens Lorenz
Eric F. Van de Velde

CRPC-TR89002
March, 1989

Center for Research on Parallel Computation
Rice University
P.O. Box 1892
Houston, TX 77251-1892

Concurrent Computations of Invariant Manifolds *

Jens Lorenz
Applied Mathematics
217-50 Caltech
Pasadena, CA 91125

Eric F. Van de Velde
Applied Mathematics
217-50 Caltech
Pasadena, CA 91125

March 3, 1989

Abstract

It is a challenging computational problem to find manifolds M which are invariant under the flow of a dynamical system. We consider the special case where the system is partitioned in the form

$$\begin{cases} d\theta/dt &= f(\theta, r) \\ dr/dt &= g(\theta, r) \end{cases}$$

and θ is a vector of periodic variables. Then we ask for an attracting invariant manifold M parametrized by θ. A typical example is given by two coupled oscillators where one asks for an invariant torus in the four dimensional state space.

There are two different analytical approaches, both leading to interesting numerical problems. The first approach uses partial differential equations, the second approach is based on the Hadamard graph transform. We discuss the numerical analysis aspects of these two approaches, both of which lend themselves to concurrent computations.

Each step of the Hadamard graph transform requires the solution of a large number of ordinary boundary value problems. These can be solved independently of each other, but communications are necessary to compute a new global interpolant after each step of the graph transform. The new interpolant gives an improved approximation to the invariant manifold.

The partial differential equations approach leads to somewhat more standard problems: After discretization and linearization one has to solve large linear systems; these have a special sparsity structure. Concurrency is exploited in the solution of these systems.

For an efficient treatment of the corresponding systems on sequential machines we refer to [3].

1 Statement of the Problem

In this paper we present two different approaches for the numerical computation of an invariant manifold of a dynamical system. For simplicity we restrict ourselves to the following special situation: The dynamical system is given in the partitioned form

$$\begin{cases} \dot{\theta} &= f(\theta, r) \\ \dot{r} &= g(\theta, r) \end{cases} \tag{1}$$

where

$$\theta = (\theta_1, \cdots, \theta_p)^T \in T^p$$

and

$$r = (r_1, \cdots, r_q)^T \in \mathbb{R}^q.$$

We ask for an invariant manifold which can be parametrized by the θ-variables. Here T^p is the p dimensional standard torus, i.e., each variable θ_j is a 2π-periodic angle variable. The functions

$$f : T^p \times \mathbb{R}^q \to \mathbb{R}^p$$
$$g : T^p \times \mathbb{R}^q \to \mathbb{R}^q$$

on the right-hand side of Equation 1 are assumed to be smooth. Let [1]

$$S^t(\theta^0, r^0),$$

where $t \in \mathbb{R}$, denote the solution of Equation 1 with initial data

$$S^0(\theta^0, r^0) = (\theta^0, r^0).$$

For each fixed $t \in \mathbb{R}$ the operator S^t is a one-to-one map of the state space $T^p \times \mathbb{R}^q$ onto itself. We can formalize the analytical problem as follows: Determine a smooth function

$$R : T^p \to \mathbb{R}^q$$

so that the manifold

$$M = \{(\theta, R(\theta)) : \theta \in T^p\} \tag{2}$$

is invariant under each operator S^t. In other words, if (θ^0, r^0) are initial data in M, then the whole trajectory $S^t(\theta^0, r^0)$, where $t \in \mathbb{R}$, lies in M.

2 The PDE Approach

One can show that the manifold defined in Equation 2 is invariant if and only if the function $R : T^p \to \mathbb{R}^q$ solves

*This research is supported in part by the Department of Energy Grants Nos. DE-FG03-85ER25009, DE-AS03-76ER72012 and by the NSF Center for Research on Parallel Computing at Caltech.

[1] For simplicity we assume all solutions to exist for all time. This is true if f and g are bounded. Using a simple cut-off technique, one can achieve boundedness without affecting the solution in a fixed bounded region of the state space $T^p \times \mathbb{R}^q$.

the following first-order system of partial differential equations:

$$f_1(\theta, R)\frac{\partial R}{\partial \theta_1} + \cdots + f_p(\theta, R)\frac{\partial R}{\partial \theta_p} = g(\theta, R) \qquad (3)$$

with $\theta \in T^p$. We refer to [7] for an account of analytical aspects. In the numerical treatment of Equation 3 one can distinguish at least four aspects:

- The discretization scheme; we used leap-frog.
- The nonlinear solver; we used Newton's method.
- The solution algorithm for linear systems; we used a concurrent sparse LU-factorization algorithm described in more detail in Section 4 and in [8].
- The starting process for the nonlinear solver; in our example (see Section 5) Equation 3 depends on a parameter $\lambda \geq 0$, and we use arc length continuation. For $\lambda = 0$ the problem is linear.

On sequential machines the largest amount of computing time is usually required for the solution of the linear systems. At this stage we have used software for concurrent computations. Aspects of interest are

- a concurrent sparse LU-factorization;
- a corresponding concurrent back solve;
- a suitable distribution of the matrix elements;
- extensions to treat the bordered matrices which typically occur in continuation steps.

3 The Hadamard Graph Transform

We outline here an alternative analytical approach to obtain an invariant manifold of Equation 1; it is not based on the PDE formulated in Equation 3. (A perturbation theory of invariant manifolds, which uses the graph transform technique described below, is developed in [4].) We anticipate that this approach is attractive for concurrent computations, but so far only limited computational experience has been obtained [2] on sequential machines. For simplicity we assume that the manifold M is locally attracting in positive time, i.e., if $(\theta^0, r^0) \in T^p \times \mathbb{R}^q$ is sufficiently close to M , then the distance of $S^t(\theta^0, r^0)$ to M tends to zero as $t \to \infty$. We use the simple projection operators P and Q defined by

$$P(\theta, r) = \theta \text{ and } Q(\theta, r) = r.$$

Clearly, P projects the state space $T^p \times \mathbb{R}^q$ onto T^p and Q projects it onto $\mathbb{R}^q$. Let

$$R^{(0)} : T^p \to \mathbb{R}^q$$

denote a known approximation for the unknown function $R = R(\theta)$ which determines the invariant manifold (see Equation 2), and let $\tau > 0$ denote a chosen time increment. Hadamard's operator F^τ applied to $R^{(0)}$ is defined as follows:

1. For any $\bar{\theta} \in T^p$ determine $\alpha \in T^p$ such that
$$P(S^\tau(\alpha, R^{(0)}(\alpha))) = \bar{\theta}.$$
2. Set
$$(F^\tau R^{(0)})(\bar{\theta}) = Q(S^\tau(\alpha, R^{(0)}(\alpha))).$$

The function $R^{(1)} := F^\tau R^{(0)}$ is, again, an approximation to R, and, under appropriate assumptions, the iteration $R^{(n+1)} = F^\tau R^{(n)}$ converges linearly to R; the function R determining the invariant manifold M is a fixed point of F^τ. (Under appropriate assumptions, the function R is independent of the chosen time increment τ though the iteration sequence and the convergence do depend on τ.)

For any fixed $\bar{\theta} \in T^p$ the above process of determining $R^{(1)}(\bar{\theta}) = (F^\tau R^{(0)})(\bar{\theta})$ can be reformulated as an ordinary boundary value problem BVP$(\bar{\theta}, R^{(0)})$:

Solve the Equation 1 for $(\theta(t), r(t))$ in $0 \leq t \leq \tau$ under the boundary conditions

$$r(0) = R^{(0)}(\theta(0)), \quad \theta(\tau) = \bar{\theta};$$

then set

$$R^{(1)}(\bar{\theta}) := r(\tau).$$

In practice, we solve BVP$(\bar{\theta}, R^{(0)})$ for $\bar{\theta}$-values in a discrete mesh T_h^p,

$$\bar{\theta} \in T_h^p \subset T^p.$$

It is important to note that all problems BVP$(\bar{\theta}, R^{(0)})$ are independent of each other and differ only in the $\bar{\theta}$-value of the right boundary condition.

Let us assume a multicomputer computation with N processes. Each process solves problems of the type BVP$(\bar{\theta}, R^{(0)})$. We choose a mesh $T_h^p \subset T^p$ of kN mesh points. Then we can assign k BVPs to each process. When solving the BVPs no communication is necessary provided the functions f, g, and $R^{(0)}$ are available locally. Solution of the BVPs provides values

$$R^{(1)}(\bar{\theta}), \quad \bar{\theta} \in T_h^p.$$

For the next iteration we need an interpolant of these discrete values, again denoted by $R^{(1)}$. Also, if the BVPs are to be solved by Newton's method, we need the first partial derivatives of $R^{(1)}$. To obtain this interpolant and its derivatives we can us FFTs. At this stage interprocess communication is necessary.

4 Computational Aspects of the PDE Approach

We attack the problem of solving Equation 3 in the more general context of solving coupled systems of nonlinear partial differential equations of the form:

$$A_1\frac{\partial R}{\partial \theta_1} + A_2\frac{\partial R}{\partial \theta_2} + \ldots + A_p\frac{\partial R}{\partial \theta_p} = g, \qquad (4)$$

where g is a q-dimensional vector and $A_1, \ldots, A_p$ are $q \times q$ matrices that depend smoothly on $(\theta, r) \in T^p \times \mathbb{R}^q$ and on a parameter λ, i.e.,

$$A_1 = A_1(\theta, r; \lambda), \ldots, A_p = A_p(\theta, r; \lambda), \text{ and } g = g(\theta, r; \lambda).$$

Equation 4 is 2π-periodic in all components of θ. The PDE is discretized on a $K_1 \times K_2 \times \ldots \times K_p$ grid. At each grid point one has q scalar unknowns. If D_j denotes the centered divided difference operator in direction θ_j, then the leap-frog discretization leads to a nonlinear algebraic system of difference equations

$$A_1 D_1 R + A_2 D_2 R + \ldots + A_p D_p R = g. \qquad (5)$$

Here $R = R^h(\theta, \lambda)$ is the unknown grid function. (The index h indicates the dependence on the mesh-size of the grid. We do not discuss the convergence as $h \to 0$.)

We wish to find solutions of Equation 5 as a function of the parameter λ. This computation is done with the *pseudo-arc-length continuation* method of Keller, see [5]. To find the solution at a particular, given value of λ, one could apply a straightforward Newton iteration process. To follow solution *paths*, more advanced tools are needed because the paths may have vertical tangents with respect to the parameter λ at *folds*, or, they may *bifurcate* into two or more solution paths. In the neighborhood of such critical points, straight Newton iterations usually diverge or converge very slowly.

Given a solution $R(\lambda_0)$ at a particular value λ_0, an initial approximation to another solution of the branch is found through linear extrapolation: a tangent to the solution path is constructed at $(R(\lambda_0), \lambda_0)$ and a step is taken along the tangent. This leads to a first approximation of a new solution; the approximation is subsequently improved by a Newton iteration for (R, λ) in a hyperplane orthogonal to the tangent direction at $(R(\lambda_0), \lambda_0)$. The constraint of orthogonality to the tangent adds an extra equation to the system 5; of course, one has the extra unknown λ. Thus, in the continuation procedure systems must be solved which have the form

$$\begin{bmatrix} S & b \\ c^T & \delta \end{bmatrix} \begin{bmatrix} x \\ \xi \end{bmatrix} = \begin{bmatrix} y \\ \eta \end{bmatrix}.$$

The matrix S is sparse; it results from a straight linearization of Equation 5 in R. The vector b results from linearization in λ. The last equation ensures orthogonality of the Newton corrections to the previous tangent. The vectors b and c are usually dense. To avoid excessive fill, it is not desirable to perform the LU-factorization of the extended matrix, unless the pivoting strategy is restricted. Using the bordering algorithm of Keller, see [6], a solution to the extended system can be found if the LU-factorization of S is known. This algorithm is applicable at folds, where the matrix S is singular. The only requirement is that the extended matrix has full rank.

Particular care is necessary in the selection of a pivoting strategy for the LU-decomposition of a possibly singular matrix. Complete pivoting, in spite of its cost, may sometimes be the only alternative. In the majority of the LU-decompositions of the path following procedure, however, a more restricted pivoting strategy is sufficient. Our program uses the LU-decomposition of [8], which can incorporate arbitrary pivoting strategies. This allows us to change the pivoting strategy dynamically during the continuation. We have used this feature in critical Newton iterations to achieve both efficiency and numerical stability. We note that pivoting is an expensive operation on sparse matrices because of two reasons: First, the cost of sparse matrix computations is dominated by the cost of *accessing* entries. Second, because arithmetic with superfluous zeroes is avoided, the arithmetic cost is decreased relative to the cost of pivoting. Typically, we perform an expensive dynamic pivot search in the LU-decomposition of the matrix S in the first step of a Newton iteration. Currently, we use either partial or complete pivoting without taking sparsity into account. (We plan to also incorporate sparse pivoting techniques in the near future.) In subsequent steps of the Newton iteration, the pivot selections of the first step are used over again. This amortizes the cost of dynamic pivoting over several LU-decompositions. The heuristic behind this procedure is simply that the matrices S of subsequent steps in a Newton iteration are nearly equal. Hence, a good pivoting strategy for the first step is likely good for the following steps, also. In the event that the LU-decomposition detects a zero or a very small pivot, we start the factorization over with complete pivoting. A matrix is accepted as singular only if the latter strategy detects a zero or a small pivot.

Concurrent performance of sparse LU-decomposition is critically dependent on the *distribution* of the matrix over the concurrent processes. The best distribution, i.e., the distribution leading to the smallest execution time, depends on the *fill* of the matrix during the elimination. The distribution of the fill over the processes itself is not easily predictable if dynamic pivoting is used. Therefore, the best distribution can often be determined only experimentally. We shall show some results in Section 6 below which demonstrate the dependence of the execution time on the data distribution. For the calculations we used the LU-decomposition of [8], which allows a wide range of data distributions.

5 Example: A System of Two Coupled Oscillators

A simple example of a single oscillator is described by the two scalar equations

$$\begin{cases} \dot{\theta} &= \omega \\ \dot{r} &= r(1 - r^2) \end{cases}$$

where ω is a fixed constant. If $r(0) > 0$, then $r(t) \to 1$ as $t \to \infty$. The one dimensional manifold

$$\{(\theta, 1) : \theta \in T^1\}$$

is invariant. It describes an attracting limit cycle. As in [1], we consider two oscillators of the above form which are coupled. If $(\theta_j, r_j), j = 1, 2$, are polar coordinates for the two oscillators, then the equations considered read

$$\begin{aligned} \dot{r}_j &= r_j(1 - r_j^2) + \lambda C_j \\ \dot{\theta}_j &= \omega + \lambda C'_j, \end{aligned} \qquad (6)$$

where j = 1,2. Here

$$C_1 = r_2(\sin(\theta_1 + \theta_2) + \cos(\theta_1 - \theta_2)) - r_1(1 + \sin 2\theta_1)$$

$$C_1' = -\cos 2\theta_1 + \frac{r_2}{r_1}(\cos(\theta_1+\theta_2) - \sin(\theta_1-\theta_2))$$

and C_2, C_2' are obtained by interchanging the indices 1 and 2 on the right-hand sides of the above expressions for C_1, C_1' . We refer to [1] for a motivation of the specific form of the coupling.

For the coupling constant $\lambda = 0$ the system has the attracting invariant 2-torus

$$M(\lambda = 0) = \{(\theta, 1, 1) : \theta \in T^2\}.$$

One can show that the torus persists for small coupling. More precisely, general theory [7] [4] yields the following result:

For any $k = 1, 2, \ldots$, there exists $\lambda_k > 0$ and a C^k function

$$R : T^2 \times (-\lambda_k, \lambda_k) \to \mathbb{R}^2$$

such that

$$M(\lambda) = \{(\theta, R(\theta, \lambda)) : \theta \in T^2\}, \quad -\lambda_k < \lambda < \lambda_k$$

is invariant under the flow of 6. Here $R(\theta, \lambda = 0) \equiv (1, 1)$, and the invariant manifold $M(\lambda)$ is locally unique. With increasing $|\lambda|$ the torus looses more and more derivatives. It is not known how exactly the torus disappears if the coupling gets too large.

6 Numerical and Computational Results

We treated the above example for $\omega = -0.55$ and $\lambda \geq 0.0$. One obtains a system of PDE's of the general form 3 for $R(\theta) = R(\theta, \lambda)$ with parameter λ. Here $p = q = 2$. It is convenient to introduce the scaled unknown $S(\theta) = S(\theta, \lambda)$ by

$$R = \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \lambda S$$

which leads to a meaningful linear limit problem at $\lambda = 0$.

Since the two identical oscillators are coupled in a symmetric fashion, it is not difficult to show: If $R = R(\theta)$ describes an invariant manifold, then

$$\tilde{R}(\theta) = \begin{bmatrix} R_2(\theta_2, \theta_1) \\ R_1(\theta_2, \theta_1) \end{bmatrix}$$

describes an invariant manifold, also. Thus, as long as the solution of the PDE for R (or for S) is unique, R must obey the symmetry

$$R_1(\theta_1, \theta_2) = R_2(\theta_2, \theta_1).$$

Whether one enforces the symmetry or not might, of course, influence the bifurcation diagram.

6.1 First Study

We discretize the PDE for S with leap-frog and do not enforce the symmetry on the discrete solution. In Figure 1, we display the execution time of the concurrent sparse LU-decomposition for the first system of the branch, i.e., at

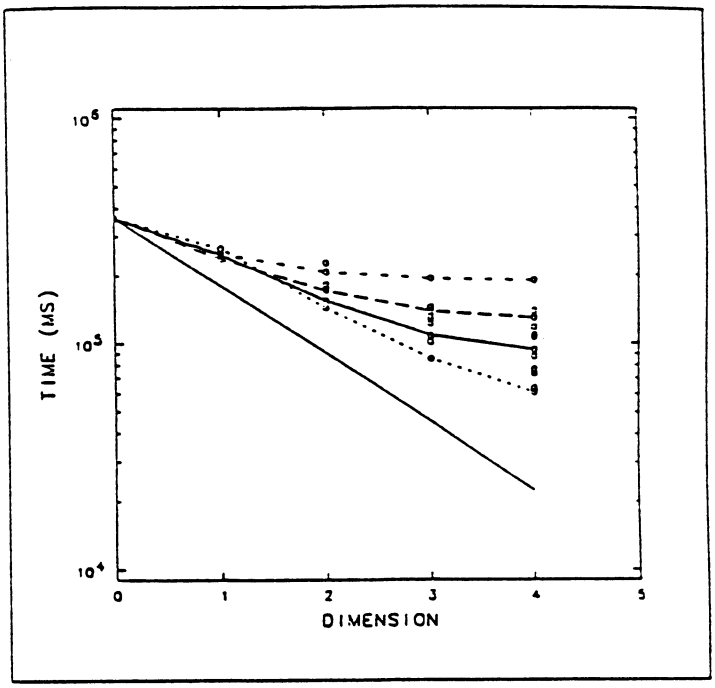

Figure 1: Sparse LU-Decomposition With Multirow Pivoting.

Multirow Pivoting		
	Minimum	Maximum
Speed Up	1.9	6.0
Efficiency	11.9%	37.2%
Process Grid	16 x 1	16 x 1
Row Distribution	linear	scatter
Column Distribution		

Table 1: Least and the Most Effective 16 Node sparse LU-Decomposition with Multirow Pivoting.

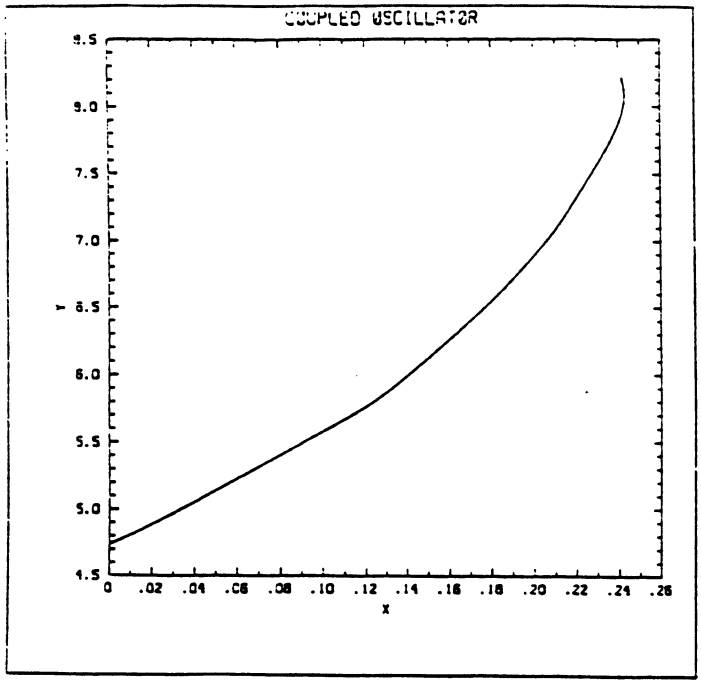

Figure 2: Norm of Solution as a Function of the Continuation Parameter λ.

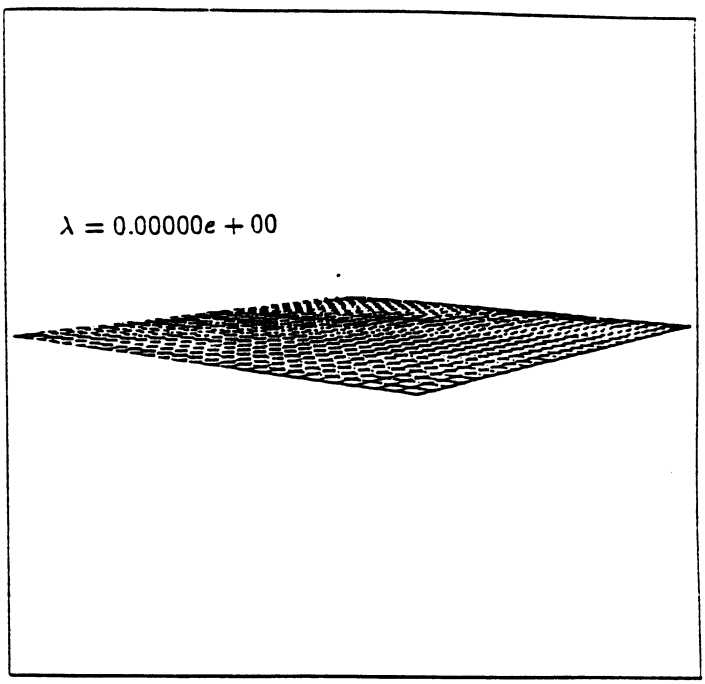

Figure 3: The Trivial Solution $R_1 = 1$ for Zero Coupling.

$\lambda = 0$. The system has $2 \times 25 \times 25 = 1250$ unknowns. The calculations were performed on the Ametek 2010 multicomputer using different numbers of nodes: a multicomputer of dimension d has 2^d nodes. For different numbers of nodes and different matrix distributions we display the execution times in a log-log plot. The straight line corresponds to the ideal speed up, which is not actually reached. We also connected execution times (for different values of d) that correspond to comparable distributions of the matrix. For example, the largest execution times were obtained for distributions where blocks of consecutive rows were allocated to the processes. The smallest execution times were obtained when consecutive rows were allocated to different processes, i.e., the rows were scattered. Rectangular distributions, i.e., the rows themselves are also distributed columnwise, performed at a level in between these two extremes. Table 1 summarizes the best and worst performance on the 16 node system. Further performance gains for the solver are to be expected with incorporation of fine tuned pivoting strategies. The pivoting strategy used for this computation was multirow pivoting (see [8] for details of this strategy).

6.2 Second Study

We discretize the PDE for S with leap-frog and enforce the symmetry on the discrete solution. With a grid spacing $h = \frac{2\pi}{25}$ we obtain a solution branch of discrete solutions. In Figure 2, we display the norm of these solutions as a function of the coupling parameter λ. In Figures 3 through 6, we display the first component $R_1(\theta)$ for a representative set of λ values along the branch. The solution above the fold in Figure 6 displays nonsmooth behavior.

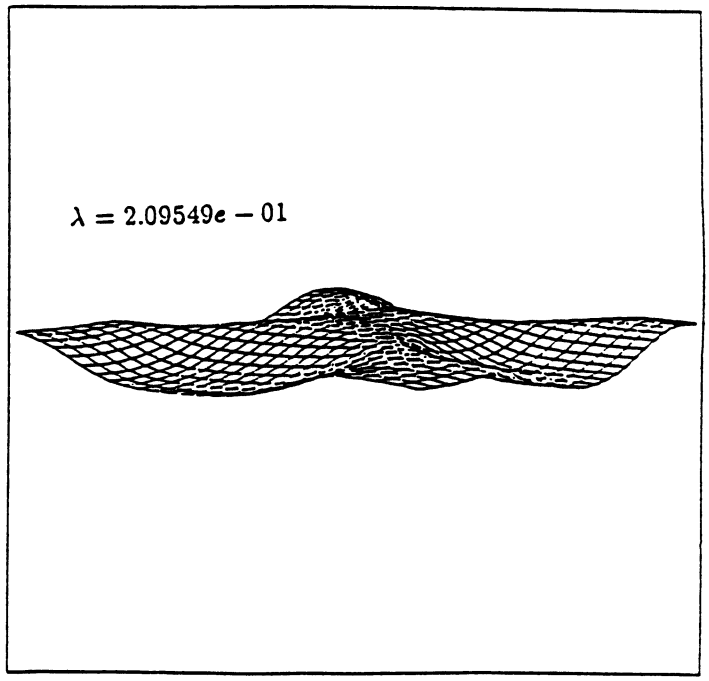

Figure 4: A Solution of the Discrete System below the Fold.

Probably, it does not correspond to a solution of the continuous problem.

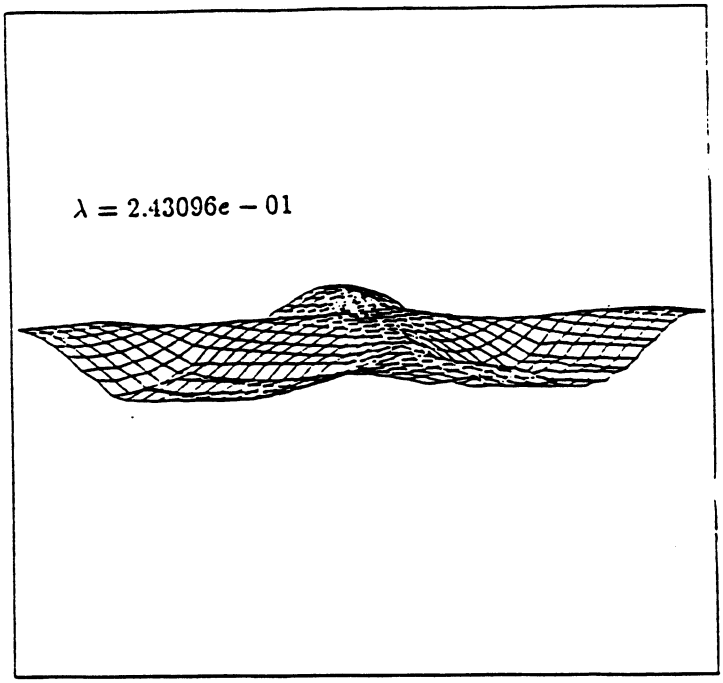

Figure 5: A Solution of the Discrete System near the Fold.

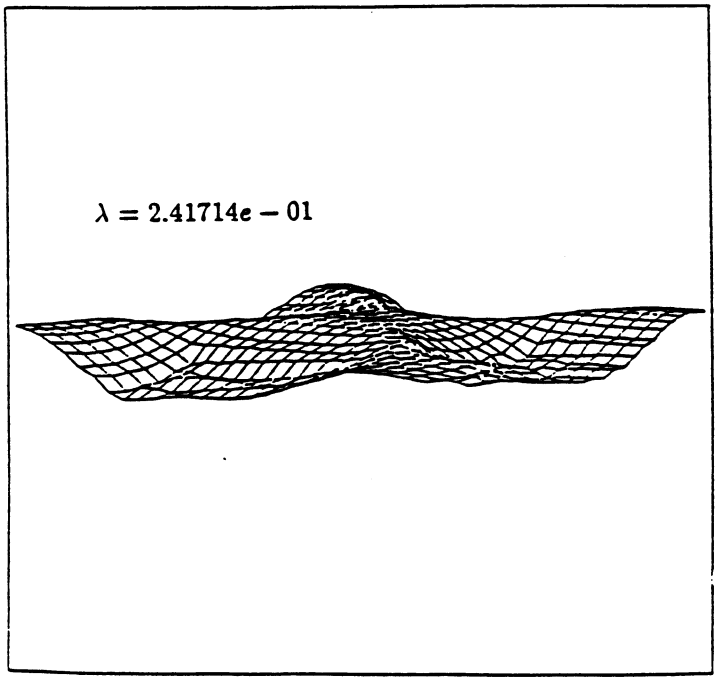

Figure 6: A Solution of the Discrete System above the Fold.

References

[1] D. G. Aronson, E. J. Doedel, and H. G. Othmer. An analytical and numerical study of the bifurcations in a system of linearly-coupled oscillators. *Physica*, 25D:20–104, 1987.

[2] L. Dieci, J. Lorenz, and R. D. Russell. *Decoupling of dynamical systems using boundary value techniques.* report LCCR TR 88-9, Simon Fraser University, 1988.

[3] L. Dieci, J. Lorenz, and R. D. Russell. Numerical calculation of invariant manifolds. In Preparation.

[4] N. Fenichel. Persistence and smoothness of invariant manifolds for flows. *Ind. Univ. Math. Journal*, 21:193–226, 1971.

[5] H.B. Keller. Numerical solution of bifurcation and nonlinear eigenvalue problems. In P. Rabinowitz, editor, *Applications of Bifurcation Theory*, pages 359–384, Academic Press, New York, N.Y., 1977.

[6] H.B. Keller. Practical procedures in path following near limit points. In R. Glowinski and J.L. Lions, editors, *Computing Methods in Applied Sciences and Engineering*, North-Holland, 1982.

[7] R. Sacker. A perturbation theorem for invariant manifolds and Hölder continuity. *J. Math. Mech.*, 18:705–762, 1969.

[8] E. F. Van de Velde. Multicomputer matrix computations: theory and practice. March 1989. Proceedings of the Fourth Conference on Hypercubes, Concurrent Computers and Applications.