Numer. Math. 53, 165–181 (1988)

Numerische Mathematik

On the Computation of Multi-Dimensional Solution Manifolds of Parametrized Equations*

Werner C. Rheinboldt
Institute for Computational Mathematics and Applications, Department of Mathematics and Statistics, University of Pittsburgh, PA 15260, USA

Dedicated to Professor Ivo Babuška on the occasion of his sixtieth birthday

Summary. A new algorithm is presented for computing vertices of a simplicial triangulation of the p-dimensional solution manifold of a parametrized equation $F(x)=0$, where F is a nonlinear mapping from R^n to R^m, $p=n-m>1$. An essential part of the method is a constructive algorithm for computing moving frames on the manifold; that is, of orthonormal bases of the tangent spaces that vary smoothly with their points of contact. The triangulation algorithm uses these bases, together with a chord form of the Gauss-Newton process as corrector, to compute the desired vertices. The Jacobian matrix of the mapping is not required at all the vertices but only at the centers of certain local "triangulation patches". Several numerical examples show that the method is very efficient in computing triangulations, even around singularities such as limit points and bifurcation points. This opens up new possibilities for determining the form and special features of such solution manifolds.

Subject Classifications: AMS(MOS): 65H10; CR: G1.5.

1. Introduction

Parameter-dependent nonlinear equations

$$F(z, \lambda)=0, \tag{1.1}$$

involving a state variable z and a parameter vector λ, arise in many applications. Under natural conditions on F and the relevant spaces the set of solutions (z, λ) of (1.1) constitutes a differentiable manifold in the product of the state

* This work was supported in part by the National Science Foundation under Grant DCR-8309926, the Office of Naval Research under contract N-00014-80-C-9455, and the Air Force Office of Scientific Research under Grant 84-0131

and parameter space, and the dimension of this manifold equals the parameter dimension.

In most practical applications interest centers not so much on computing a few solutions of (1.1), but rather on determining the form and special features of the solution manifold. For instance, if (1.1) represents an equilibrium problem, then we may wish to determine the bifurcation diagram or the boundaries of the stability regions on the manifold. But, as it turns out, all standard computational methods for such an analysis require us to construct a picture of a p-dimensional manifold from information along one-dimensional paths. In fact, all these methods belong to the family of continuation processes for which the dimension of the parameter space always has to equal one. Thus, before such a process can be applied, any problem with a larger parameter dimension must be reduced to some form involving only a scalar valued parameter and, geometrically, such a reduction is equivalent with a restriction to some path on the solution manifold of the original equation. A continuation method then computes a sequence of points along such a path. For example, in structural engineering the parameter λ often characterizes a vector of load components in which case it has become customary to fix a linear combination of these components specifying a particular load direction. The resulting reduced equation then involves only the load intensity as a one-dimensional parameter variable and the standard 'incremental' methods generate points along this load path.

In general, it is not easy to develop a good picture of a p-dimensional manifold from information along one-dimensional paths. Thus it is not surprising that there is growing interest in computational methods which generate multi-dimensional grids of solution points covering an entire segment of the manifold. Up to now the only method for computing such multi-dimensional grids appears to be that of Allgower and Schmidt [2]. It utilizes a simplicial continuation algorithm to cover, by means of n-simplices, a portion of a p-dimensional manifold defined by an equation of the form (1.1).

Here we present a different method for computing vertices of a simplicial triangulation of segments of the p-dimensional solution manifold of an Eq. (1.1). An essential part is a constructive algorithm for computing orthonormal moving frames on the manifold in the sense first considered by E. Cartan; that is, of orthonormal bases of the tangent manifolds that vary continuously with their points of contact (see e.g., [17]). The resulting triangulation algorithm uses these bases and a predictor-corrector approach to compute the desired grid points. It has many similarities with the continuation methods including a comparable computational complexity. In particular, the Jacobian matrix of the mapping is not required at all the points. For example, on a two-dimensional manifold the computation of a typical triangulation pattern with 114 triangles involves only 19 Jacobian decompositions.

After summarizing some basic concepts in Sect. 2 we introduce the moving frame algorithm in Sect. 3. Then Sect. 4 outlines the general triangulation method and Sect. 5 present several numerical examples. Finally, we end with an outlook on the utilization of the computed triangulations for the determination of specific features of the manifold.

2. Basic Concepts

Throughout this article, let

$$F: S \to R^m, \quad S \text{ open in } R^n, \quad p = n - m > 1, \tag{2.1}$$

be a differentiable mapping of class C^r, $r > 1$, on the subset S. As usual, a point $x \in S$ is called regular if the first derivative, $DF(x)$, of F has full rank m and hence maps R^n onto R^m. We consider the equation

$$F(x) = 0, \quad x \in S, \tag{2.2}$$

and assume that its regular solution set

$$M = \{x \in S;\ F(x) = 0,\ x \text{ regular}\} \tag{2.3}$$

is non-empty. It is well-known that M is a p-dimensional C^r-manifold in R^n without boundary (see, e.g., [17] or [15]).

The tangent space $T_x M$ at any point $x \in M$ may be identified with the kernel of the Jacobian $DF(x)$; that is,

$$T_x M = \ker DF(x) = \{u \in R^n;\ DF(x)\, u = 0\}. \tag{2.4}$$

The normal space $N_x M$ at $x \in M$ is the orthogonal complement of the tangent space under the natural Euclidean inner product on R^n; that is,

$$N_x M = (T_x M)^{\perp} = (\ker DF(x))^{\perp} = \operatorname{rge} DF(x)^T. \tag{2.5}$$

Since $DF(x)$ has maximal rank in some open subset S_0 of S containing M, the mapping

$$x \in S_0 \to DF(x)^T [DF(x)\, DF(x)^T]^{-1}\, DF(x) \in L(R^n) \tag{2.6}$$

from S_0 into the space $L(R^n)$ of all linear mappings on R^n is of class C^{r-1} on S_0. Hence, the orthogonal projection

$$P: M \to L(R^n);\ P(x) = I_n - DF(x)^T [DF(x)\, DF(x)^T]^{-1}\, DF(x), \quad x \in M \tag{2.7}$$

of R^n onto $T_x M$ is a mapping of class C^{r-1} on the manifold M, (here I_n denotes the identity on R^n).

For the computation we require local coordinate systems on M. Any p-dimensional subspace T of R^n induces a local coordinate system of M at any point $x \in M$ where

$$T \cap N_x M = \{0\}, \tag{2.8}$$

In fact, if (2.8) holds for $x \in M$ then there exist open neighborhoods V_1 and V_2 of the origins of T and R^n, respectively, as well as a unique C^{r-1} function $w: V_1 \to T^{\perp}$ with $w(0) = 0$, such that

$$M \cap V_2 = \{y \in R^n;\ y = x + t + w(t),\ t \in V_1\}, \tag{2.9}$$

(see, e.g., [9] or [15]). In other words, in the local coordinate system induced by T the point $y=x+t+w(t)$ of M has the coordinate $t\in T$.

A point $x\in M$ where (2.8) holds is a non-singular point with respect to the given coordinate space T, else we call x a singular point. Clearly, at any point $x\in M$ the tangent space $T_x M$ can be chosen as coordinate space and x is non-singular with respect to it. In most applications, a "natural" parameter space A is given, as indicated by the form of the Eq. (1.1), and the orthogonal subspace $Z=A^{\perp}$ is the state space. Then interest centers on determining the singular points with respect to the space A. These are the so-called foldpoints on M where the tangent space has a non-zero intersection with the state space Z and the parameter space A can no longer be used as a local coordinate space. These are also the points where, for example, in equilibrium problems a change in the stability behavior of the physical system under study may be expected.

Numerically, the mapping w of (2.9) can be implemented in various ways. A simple approach is based on a chord form of the Gauss-Newton method. At the given point $x\in M$ we compute the QR-factorization

$$DF(x)^T=Q\begin{bmatrix}R\\0\end{bmatrix} \tag{2.10}$$

of the transposed Jacobian $DF(x)^T$ involving the $n\times n$ orthogonal matrix Q and $m\times m$ nonsingular, upper triangular matrix R. Then, starting from any point y in a suitable neighborhood of x in $x+T_x M$, we may apply the process

1) Set $y^0=y$;

2) for $k=0, 1, \ldots$ until convergence

2a) solve $R^T z=F(y^k)$ for $z\in R^m$;

2b) compute the next iterate $y^{k+1}=y^k-Q(z, 0)^T$. (2.11)

With (2.10) this is readily rewritten in the form

$$DF(x)(y^{k+1}-y^k)+F(y^k)=0,\quad y^{k+1}-y^k\in[\ker DF(x)]^{\perp},\quad k=0, 1, \ldots \tag{2.12}$$

which shows that $y^{k+1}-y^k\in N_x M$. Thus we have $y^k\in y+N_x M$ for all $k>0$, whence the limit point y^* – if it exists – is the unique point in the intersection of M and $y+N_x M$ which, in the notation of (2.9), can be written as $y^*=x+t+w(t)$, $t=y-x$.

The convergence theory of Gauss-Newton processes is well understood. Earlier studies of these methods considered applications to least squares problems and hence assumed that F maps R^n into R^m where $n<m$. A local convergence result which covers our case $n>m$ may be found in [8, Theorem 4]. Another simple proof for the method in the form (2.12) also follows along the lines of the convergence proof for singular chord methods given in [14]. These results guarantee the validity of the following theorem:

Theorem 1. *Under the stated assumptions about the mapping F there exists for any point x of M a neighborhood $V(x)$ of x in $x+T_x M$ such that for any starting*

point y in $V(x)$ the Gauss-Newton process (2.11) *converges to the unique point y^* in the intersection of M and $y+N_x M$ that has the coordinate $t=y-x$ in the local coordinate system induced by the tangent space $T_x M$.*

3. A Moving Frame Algorithm

As usual, a vector field of class C^s, $s \leqq r$, on an open subset M_0 of our manifold M is a C^s function $u: M_0 \to TM$ from M_0 into the tangent bundle TM such that $u(x) \in T_x M$ for all $x \in M_0$. A moving frame of class C^s on M_0 is a mapping which associates with each $x \in M_0$ a frame (i.e., ordered basis) $\{u^1, \ldots, u^p\}$ of $T_x M$ such the functions $u^i: M_0 \to TM$, $i=1, \ldots, p$, form p vector fields of class C^s on M_0. When such a moving frame exists on M_0 then the sub-manifold M_0 is said to be parallelizable. We will consider only orthonormal moving frames; that is, frames for which the basis vectors are orthonormal. (For a discussion of these concepts see, e.g., [17].)

Clearly, the problem of computing an orthonormal basis of the tangent space $T_x M$ of M at a given point $x \in M$ is equivalent with the construction of an $n \times p$ matrix U with orthonormal columns for which

$$DF(x)\, U = 0. \tag{3.1}$$

There are many techniques for computing such a matrix. A well-known procedure is provided by the QR-decomposition (2.10). In fact, if the matrix Q is partioned in the form $Q=(Q_1, Q_2)$ where Q_1 has m columns then we may use $U=Q_2$ as the desired basis. Various other techniques for computing U have been proposed. In particular, for practical application to large problems the methods in [5] and [6] for producing sparse bases U are specially important.

An algorithm for constructing a moving frame of class C^s on some open subset M_0 of M has to generate a basis matrix $U=U(x)$ for each $x \in M_0$ in such a way that the mapping $U: M_0 \to L(R^p, R^n)$ is of class C^s. As Coleman and Sorensen [7] have noted, the approach based on the QR-decomposition does not give continuously varying matrices $U(x)$. This observation extends to other algorithms of a similar nature; in fact, it relates directly to the corresponding problems of computing eigenvectors associated with a multiple eigenvalue. Three remedies are proposed in [7], but they concern only the construction of a limit U_0 of a sequence of bases $U(x)$ when x tends to x_0.

For our construction of a moving frame we restrict attention to open subsets M_0 of M where a given p-dimensional subspace T of R^n induces a local coordinate system; that is, where (2.8) holds for all x of M_0. With a mild abuse of language, let T_0 be an $n \times p$ matrix with orthonormal columns which span the coordinate space T. Moreover, assume that we have picked some method for computing at any point x of M_0 an $n \times p$ matrix $U(x)$ with orthonormal columns that span the tangent space $T_x M$ at x. Of course, U is not expected to depend continuously on x. For instance, we may use the matrices produced by the QR-decomposition technique. For any orthogonal $p \times p$ matrix $Q=Q(x)$ the matrix $U(x)Q$ is another orthonormal basis of $T_x M$ and our aim is to

construct matrices $Q(x)$ such that the "rotated" bases $U(x)Q(x)$ depend continuously on x for all x in M_0.

The normalization $(T_0)^T T_0 = I_p$ suggests that we choose the orthogonal matrix Q so that $(U(x)Q)^T T_0$ approximates the identity I_p. Various norms may be used for this; an advantageous choice is the Frobenius norm $\|A\|_F = [\mathrm{tr}(A^T A)]^{1/2}$. The resulting optimization problem

$$\|(U(x)Q)^T T_0 - I_p\|_F = \min, \quad \text{subject to } Q^T Q = I_p \tag{3.2}$$

is a case of the orthogonal Procustes problem. As discussed in [12], the following algorithm solves (3.2):

(1) $U_0 := U(x)^T T_0$;

(2) compute the singular value decomposition $U_0 = A\Sigma B^T$

(3) $Q := AB^T$. (3.3)

For our purposes the essential fact is now the content of the following theorem:

Theorem 2. *Let M_0 be an open subset of M on which the given p-dimensional subspace T of R^n induces a local coordinate system. For any $x \in M_0$, let $U(x)$ be an orthonormal basis matrix of $T_x M$ and compute the orthogonal matrix $Q = Q(x)$ of* (3.3). *Then the mapping $x \in M_0 \to U(x)Q(x) \in L(R^p, R^n)$ is of class C^{r-1} on M_0 and defines an orthonormal moving frame on M_0.*

Proof. Evidently, $z^T U_0 = 0$ implies that the tangent vector $U(x)z \in T_x M$ must be orthogonal to the subspace T of R^n spanned by the columns of T_0, and, and hence that $U(x)z \in T^{\perp}$. By construction of M_0 this cannot happen for $x \in M_0$ unless $z = 0$. In other words, for $x \in M_0$ the matrices U_0 and Σ arising in (3.3) are non-singular. Now

$$U_0 = A\Sigma B^T = AB^T(B\Sigma B^T) = QH, \quad H = B\Sigma B^T,$$

is the polar decomposition of U_0 and it follows that

$$H = [(U_0)^T U_0]^{1/2} = [(T_0)^T U(x) U(x)^T T_0]^{1/2}$$

is non-singular, whence

$$Q = U(x)^T T_0 [(T_0)^T U(x) U(x)^T T_0]^{-1/2}.$$

Evidently, $U(x)U(x)^T = P(x)$ is the orthogonal projection (2.7) from R^n onto $T_x M$. Thus we see that

$$U(x)Q = P(x) T_0 [(T_0)^T P(x) T_0]^{-1/2} \tag{3.4}$$

and, since P was already shown to be of class C^{r-1} on M, the result follows.

Our overall moving frame algorithm on M_0 now consists of the following three steps:

(1) Given $x \in M_0$, compute the basis matrix $U(x)$ of $T_x M$;

(2) compute the orthogonal matrix Q by (3.3); (3.5)

(3) form the desired basis matrix $U(x)Q$.

If the QR-factorization is used in step (1), then the order of the required number of floating point operations is as follows:

Computation of $U(x)$	$O(nm^2)$
Multiplication $U(x)^T T_0$	$O(np^2)$
Singular value decomposition	$O(p^3)$
Formation of the product $U(x)Q$	$O((n+p)p^2)$.

Thus, when the dimension p of the manifold is small in comparison with the space-dimension n, as is typical in applications, then the principal cost is related to the QR-factorization of $DF(x)^T$ and involves about $(2/3)\,n^3$ operations. This is indeed analogous to the complexity of a standard continuation process. It should be noted also that in practice the "rotated" basis $U(x)Q$ need not be computed explicitly. This is especially important when $U(x)$ is sparse.

For the practical implementation it is certainly desirable to choose the basis matrix T_0 of our reference coordinate space T in R^n as simply as possible. In particular, it is very advantageous to define T as a subspace spanned by p appropriate natural basis vectors $e^1, \ldots, e^n$ of R^n. Then T_0 can be taken as a matrix with columns e^i with certain distinct indices $i=i_j$, $1 \leqq i_j \leqq n$, $j=1, \ldots, p$. For the choice of these indices, recall that for any x of M and given vector $a \in R^n$, $\|a\|_2 = 1$, the principal angle $\alpha \in [0, \pi/2]$ between $T_x M$ and span $\{a\}$ is defined by

$$\cos(\alpha) = \max\{u^T a;\ u \in T_x M,\ \|u\|_2 = 1\}.$$

Evidently, if a is one of the global basis vectors of R^n, then

$$\cos(\alpha_i) = \|U(x)^T e^i\|_2, \qquad i=1, \ldots, n; \tag{3.6}$$

that is, the Euclidean norm of the i-th row of $U(x)$ is the cosine of the principal angle between the tangent space $T_x M$ and the i-th coordinate line span $\{e^i\}$ of R^n. Since the Euclidean norm is invariant under orthogonal transformations, it is obvious that the principal angle does not depend on the particular basis $U(x)$ of $T_x M$.

This suggests the desired selection of T_0. We initialize our moving frame algorithm at some reference point x^* of M and compute a basis matrix $U(x^*)$ of the tangent space of M at this point. This allows for a straightforward calculation of the principal direction cosines $\tau_i = \cos(\alpha_i)$ of (3.6) which we order in descending order of size. Let $i_1, \ldots, i_p$ be the indices of the p largest of these τ_i (with equality broken, say, by lexicographic ordering), then the corresponding natural basis vectors of R^n span our reference coordinate space T and form the columns of the basis matrix T_0. Since $U(x^*)$ has rank p, none of the selected coordinate directions can be orthogonal to the tangent space. Hence, as required, x^* is a non-singular point with respect to T, and the subset M_0 of Theorem 2 contains an open neighborhood of x^* on M. Geometrically the constructed

subspace T is close to the tangent space of M at x^* in the sense of the above maximization of the direction cosines (3.6). In fact, our construction is analogous to the local coordinate selection used in the continuation program PITCON (see [15] or [16]). Obviously, this choice of T_0 also has the advantage that the computation of the matrix U_0 in step (1) of the algorithm (3.3) simply becomes an extraction of p of the rows of $U(x)$.

We end this section with two observations. The restriction to the subset M_0 of Theorem 2 opens up other possibilities for computing a moving frame. In particular, the widely used augmenting methods for parametrized problems suggest consideration of the matrix $A(x)=(DF(x)^T, T_0)$, which, under our assumptions about T_0, is non-singular for all $x\in M_0$. Therefore, the QR-factorization $A(x)=Q(x)R(x)$ is unique provided only that all diagonal elements of $R(x)$ are enforced to be positive. From this it follows readily that $Q(x)$ is of class C^{r-1} on M_0 and, since the last p columns of $Q(x)$ form an orthonormal basis of $T_x M$, we have obtained another moving frame algorithm on M_0. When p is again small in comparison to n, its complexity is slightly higher than that of (3.5); but, more importantly, (3.5) has the advantage not to depend on the use of the QR-factorization. In fact, as was stressed before, the tangent basis $U(x)$ may be computed by any available method, and for high dimensional problems sparse basis methods of the type in [5, 6] are especially desirable. There are also other situations where the QR-factorization is not readily applicable for computing tangent bases. An example for this occurs in the triangulation of sub-manifolds of foldpoints of the manifold M. Then, as will be shown elsewhere, different methods for computing tangent bases are needed, but the "rotation" technique of Theorem 2 remains nevertheless applicable.

As noted already, in the algorithm (3.5) the formation of the product $U_0 = U(x)^T T_0$ can be reduced to the extraction of p rows of $U(x)$, and, in practice, the product $U(x)Q$ need not be computed either. Hence it may be of interest that, in the frequently occurring case of manifolds of dimension $p=2$, we can even avoid the singular value decomposition of U_0. In fact, in that case, there exists a direct representation, up to signs, of the orthogonal matrix Q of the algorithm (3.3). More specifically, if

$$U_0=\begin{bmatrix} a & b \\ c & d \end{bmatrix},$$

then a straightforward calculation shows that

$$Q=\begin{bmatrix} \tau & -\sigma \\ \sigma & \tau \end{bmatrix}$$

where

$$\tau=\pm(a+d)/\delta, \quad \sigma=\pm(b-c)/\delta, \quad \delta=[(a+d)^2+(b-c)^2]^{1/2}.$$

With the normalization $ad-bc>0$, the signs are readily chosen by comparing the directions of the computed frame with the directions of the natural basis vectors of T_0. This leads also to a direct proof of the theorem in this special case.

4. A Triangulation Algorithm

The results of the previous section are now used to generate the desired triangulation on a subset of our manifold M. The basic idea is as follows: We introduce a reference triangulation on R^p and use the bases produced by the moving frame algorithm to map segments of it onto the spaces $x+T_xM$ corresponding to appropriate points x on M. Then the Gauss-Newton process (2.11) is applied to "project" these triangulations from $x+T_xM$ onto M.

The reference triangulation, of course, is any covering of R^p by a locally finite collection of p-simplices such that any two of these simplices intersect either in a common face or not at all. The literature on this topic is large and we refer here only to the discussion of various numerically efficient triangulations in [18]. Our algorithm does not place any particular restrictions on the choice of this triangulation except that we should be guided by considerations of computational simplicity. Let Σ be the collection of simplices of this triangulation.

Most triangulations used in simplicial continuation studies are generated by pivoting rules. A simple such rule is pivoting by reflection. For any index $j\in\{1, 2, \ldots, p\}$ set $j_+=j+1$ and $j_-=j-1$ with the provision that $j_+=1$ if $j=p$ and $j_-=p$ if $j=1$. Then, for a given p-simplex $\sigma=[y^0, y^1, \ldots, y^p]$ in R^p, pivoting by reflection of the vertex y^j is defined as the replacement of σ by the simplex $[y^0, \ldots, y^{j-1}, y^{j+}+y^{j-}-y^j, y^{j+1}, \ldots, y^p]$. If σ_0 is a given reference simplex in R^p, then by repeated application of this procedure a triangulation of R^p can be generated (see, e.g. [1]).

A frequently used example is the so-called Kuhn triangulation which is generated by repeated pivoting by reflection starting with the simplex

$$\sigma_0=[0, e^1, e^1+e^2, \ldots, e^1+e^2+\ldots+e^p].$$

In the case $p=2$, we can also use triangulations of R^2 by equilateral triangles generated by pivoting by reflection beginning with the 2-simplex

$$\sigma_0=[0, e^1, 0.5(e^1+\sqrt{3}e^2)]. \tag{4.1}$$

Let x denote a given vertex of the triangulation in R^p and $\delta>0$ a fixed steplength. Then for any point $x\in M$ where a basis matrix U of T_xM is known, the mapping

$$A: R^p\to x+T_xM,\ A\eta=x+\delta U(\eta-\xi), \qquad \eta\in R^p \tag{4.2}$$

transfers Σ onto $x+T_xM$. As in Theorem 1, let $V(x)$ denote the local convergence domain in $x+T_xM$ of the Gauss-Newton process (2.11). If η is a vertex of Σ for which $A\eta$ belongs to $V(x)$, then (2.11) can be used to map $A\eta$ into a point $y\in M$. The set $\Gamma(\xi, x, U, \delta)$ of vertices of Σ that can be mapped onto M in this way shall be called the "patch" corresponding to the information ξ, x, U, δ.

An idealized version of our algorithm can now be formulated as follows:

(1) Select a reference vertex $\xi=\xi^*$ of Σ;
(2) select a reference point $x=x^*$ of M;
(3) initialize the moving frame algorithm at x^* and let M_0 be the subset where, by Theorem 2, this algorithm applies;

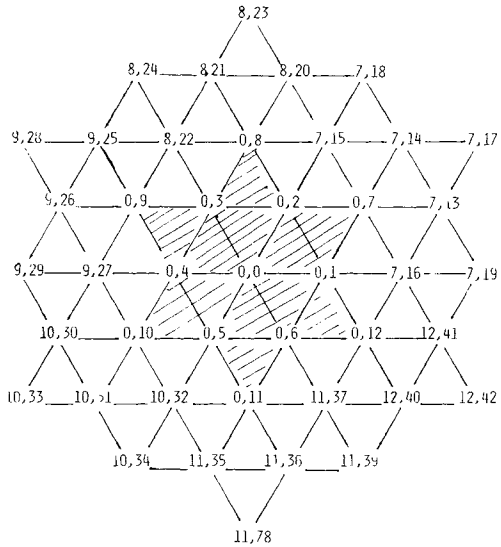


Fig. 1

(4) mark the vertex ξ as used;
(5) while x belongs to M_0
- (5a) mark ξ as a "center"; (4.3)
- (5b) compute the frame $U = U(x)$ by the moving frame algorithm;
- (5c) select all vertices of the patch $\Gamma(\xi, xU, \delta)$ which have not yet been marked "used";
- (5d) use (4.2) to map these vertices onto $x + T_x M$ and mark them "used";
- (5e) use the Gauss-Newton process to project the resulting points from $x + T_x M$ onto M;
- (5f) choose a "used" vertex ξ of Σ not marked a "center" and let x be its computed image on M;

The points computed on M inherit the connectivity pattern of the original simplices of Σ which in turn induces a simplicial approximation M_Σ of a segment of M in R^n. The algorithm is still "ideal" in nature since, in practice, the sets M_0 and $\Gamma(\xi, x, U, \delta)$ are not known explicitly. Without this knowledge the computation may halt when the iteration in step (5e) fails to converge; that is, when we encounter a point in the affine space $x + T_x M$ of one of the centers ξ which does not belong to the neighborhood $V(x)$ specified in Theorem 2. A second possibility for failure arises in the execution of the moving frame algorithm in step (5b) when the selected point x does not belong to M_0.

In order to make the algorithm practical, we replace the "ideal" patches $\Gamma(\xi, xU, \delta)$ in step (5c) by "standardized" patches $\Gamma_0(\xi)$. The definition of these patches depends on the specific reference triangulation in R^p. As an example, consider the earlier mentioned triangulation on R^2 consisting of equilateral triangles produced by pivoting by reflection from the triangle (4.1). Then the standarized patch for the center point (0, 0) in Fig. 1 is the hatched, star-shaped region

and for any other vertex it is obtained by obvious translation. With this standard patch the progress of the algorithm is easily followed in Fig. 1. There, at each vertex, the second of the two integers is a counter and the first one identifies the "center" ξ that is used in mapping that vertex onto M. Thus after the initial vertex 0, the nodes 7, ..., 12 become centers which serve to map the nodes 13, ..., 42 onto M. Then the process continues with nodes 17, 18, 19, 23, 24, 28, 29, 33, 34, 38, 39, 42 as centers. This is no longer shown in the figure, but, in practice, we have always continued through this further stage. It results in the earlier mentioned total of 114 triangles involving 19 centers and hence as many Jacobian evaluations.

Once a standardized patch $\Gamma_0(\xi)$ is used in step (5c), a suitable divergence check has to be built into the Gauss-Newton process. If in step (5d) this check is triggered, then the corresponding vertex ξ of Σ is flagged as unusable. Such unusable vertices are excluded from the further computation. A similar procedure may be followed when in step (5b) the moving frame algorithm fails. However, in the latter case it is often advantageous to re-initialize the moving frame algorithm at one of the successfully computed points x on M. Of course, then the computed basis $U(x)$ has to be used as the reference matrix T_0.

The above provisions may result in triangulations that cover a somewhat irregular domain on M. Fortunately, in practice, this does not occur as frequently as might be expected, provided the steplength δ is not chosen too large. Accordingly, it is natural to introduce procedures for adapting δ in cases of failures and hence for producing irregular triangulations on M. The ideas entering into such procedures are similar to those used in continuation methods and will not be discussed here.

5. Numerical Experiments

In this Section we present results of some numerical experiments with the triangulation algorithm. The method produces a wealth of numerical data which cannot be reproduced in the limited space of this paper. At the same time, it is a challenging problem to invent instructive graphical representations of manifolds of dimension larger than 2. As a consequence, only some graphical results for two-dimensional manifolds are shown here. It is hoped that other experiments with higher dimensional manifolds can be given elsewhere.

An Exothermic Reaction

As a first example we consider a simple transport model for an exothermic, first-order reaction-scheme discussed in [4] which, in dimensionless form, leads to a two-point boundary value problem

$$(Du')' + k_0(\mu - u)\exp(-\lambda(1+u)^{-1}) = 0, \quad u(0) = u(L) = 0. \tag{5.1}$$

The dimensionsless parameters μ and λ involve the constant concentration and temperature on the outside of the system, and for the calculation we follow

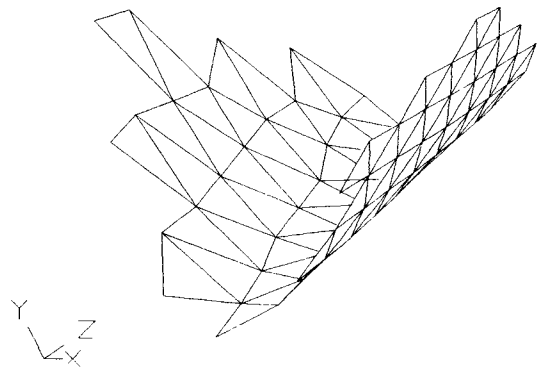


Fig. 2

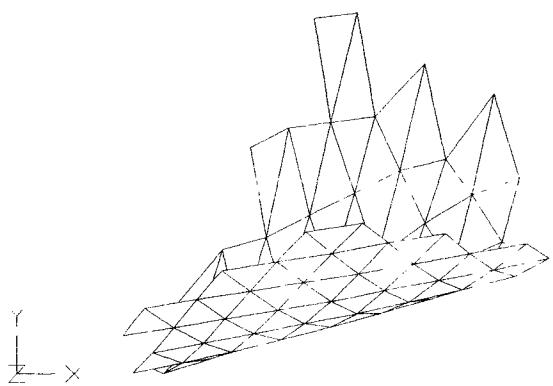


Fig. 3

[4] and set $D=1$, $L=1$, and $k_0=10^7$. The standard finite difference approximation of (5.1) on a uniform mesh $x_i=ih$, $i=0, 1, \ldots, n+1$, $h=(n+1)^{-1}$ then produces a nonlinear equation of the form

$$-x_{i-1}+2x_i-x_{i+1}=h^2 k_0(\mu-x_i)\exp(-\lambda(1+x_i)^{-1},$$
$$i=1, \ldots, n,\ x_0=x_{n+1}=0. \tag{5.2}$$

For $\mu=1$ there is a simple turning in λ near $\lambda=22$ which was calculated with the continuation code PITCON and $n=10$. This point was then used to initialize the triangulation algorithm. Here – as in the subsequent example – the reference triangulation in R^2 was generated by repeated pivoting by reflection from the equilateral triangle (4.4). The stepsize in the affine mapping (4.5) was $\delta=0.4$. Figure 2 shows the results of this triangulation. More specifically, the intersection of the computed simplicial approximation M_Σ with the (λ, μ, x_c)-space is shown, where x_c is the computed x-value at the center of the interval. The coordinate axes are marked by $x=\lambda$, $Y=\mu$, and $Z=x_c$, and are slightly rotated. One sees clearly the star-shaped pattern of the reference triangulation of Fig. 1 as it was

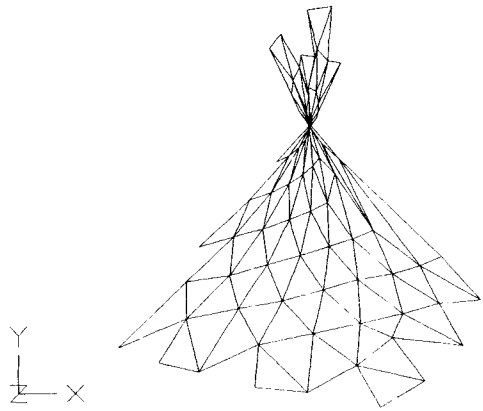


Fig. 4

mapped onto the manifold. The floor of the valley actually represents a foldline as can be seen in Fig. 3, where the same surface in (λ, μ, x_c)-space is projected onto the (λ, μ)-parameter plane and we have again $X=\lambda$, $Y=\mu$, $Z=x_c$. It shows also that the location of the turning point in λ depends approximately linearly on μ.

A Shallow Circular Arch

As the second example we consider a finite-element model for the deformation of a thin, shallow, circular arch which has been used as a test case by many authors. It appears to go back to Walker [19] and we employ here the same formulation as in [13]. In particular, in a (r, Θ)-polar coordinate system with the vertical direction as the r-axis, the unloaded configuration of the arch is represented by the circular segment $\{(r, \Theta); r=10, -\Theta_0 \leqq \Theta \leqq \Theta_0 = 15°\}$ and, for pinned ends, the dimensionless total potential energy and associated boundary conditions are given by

$$\int_{-\Theta_0}^{\Theta_0} [[(w'-u)+\alpha_0(u')^2]^2+\alpha_1(u'')^2-\alpha_2 pu]\, d\Theta,$$

$$u(\Theta)=w(\Theta)=u''(\Theta)=0, \quad \Theta=\pm\Theta$$

where primes denote derivatives with respect to Θ. For the asymmetrical load

$$p(\Theta)=(1-\nu)\mu, \quad \text{if } -\Theta_0 \leqq \Theta<0, \quad \text{and} \quad p(\Theta)=(1+\nu)\mu, \quad \text{if } 0 \leqq \Theta \leqq \Theta_0,$$

involving the two parameters ν and μ, the load path for $\nu=0$ has a bifurcation point near $\mu=1.9$. This point was computed with PITCON and used to initialize the triangulation algorithm. The stepsize of the mapped triangles was $\delta=0.5$. The results are shown in Fig. 4. More specifically, let x_c denote the (dimensionless) radial deformation at the center, then the figure shows the intersection

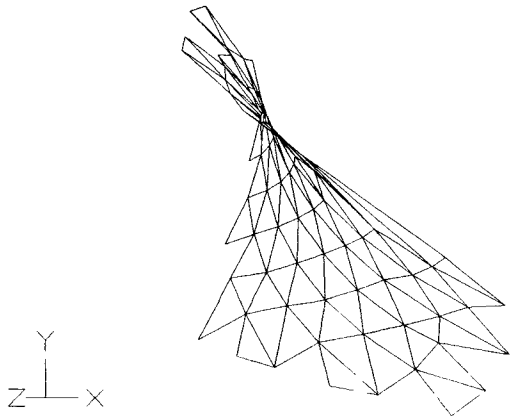
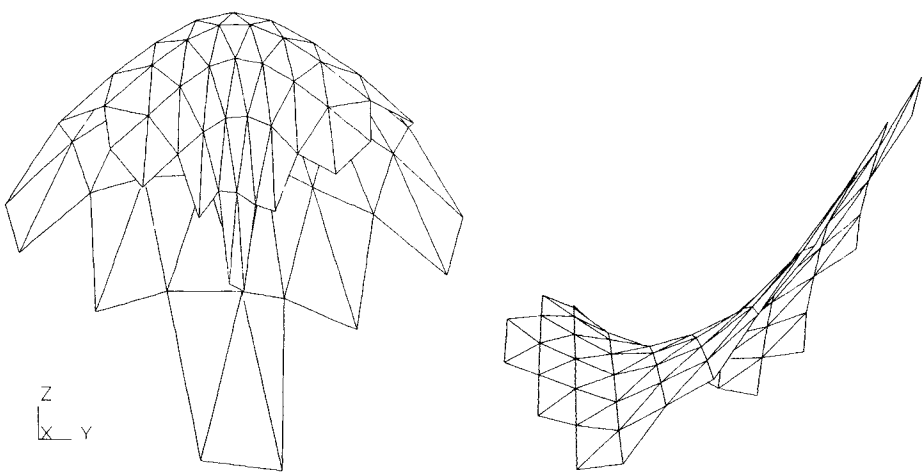


Fig. 5



Figs. 6 and 7

of the manifold with the (μ, v, x_c)-space projected onto the (μ, v)-plane. The coordinate axes are $X=\mu$, $Y=v$, $Z=x_c$. The cusp-bifurcation is clearly visible and the saddle shape of the surface can be seen even better in the slightly rotated Fig. 5.

The problem was also run with the load function

$$p(\Theta)=\mu[1-4(v-\Theta)(\Theta_0)^{-1}], \quad \text{if } \max(-\Theta_0, v-0.25\Theta_0)\leqq\Theta\leqq v,$$
$$p(\Theta)=\mu[1+4(v-\Theta)(\Theta_0)^{-1}], \quad \text{if } v\leqq\Theta\leqq\min(\Theta_0, v+0.25\Theta_0),$$

considered already in [3]. In other words, the load is a piecewise-linear hat function which has the value μ at $\Theta=v$ and is zero outside the interval centered at v of width $0.5\Theta_0$.

Figures 6 and 7 give results with this load obtained at two initialization points. The coordinates are $Y=v$, $Z=\mu$, $X=x_c$. More specifically, the triangula-

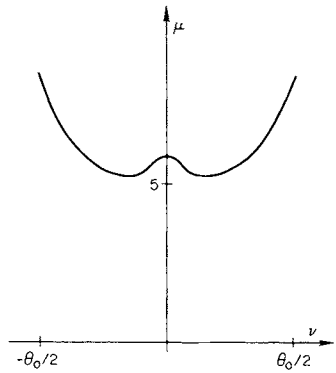


Fig. 8

tions were centered at the limit points with respect to μ when v is fixed at $v=0$ or $v=0.5$, respectively. Once again these limit points were computed with PITCON. The foldline in the (v, μ)-plane has the shape shown in Fig. 8 and Fig. 6 and 7 clearly show segments of this foldline. In particular, Fig. 7 contains one of the points where the most dangerous load occurs, namely at about $v = 0.16\Theta_0$.

6. Outlook

The numerical examples indicate that the triangulation algorithm works very efficiently even around singularities. Thus – as intended – it does indeed offer a new tool for deriving information about the shape and features of the manifold. Of course, as mentioned before, we are able to present here only some graphical information and none of the extensive numerical output of the algorithm. This output is available as input to various post-processes for extracting further information. Several such processes have already been developed and will be described elsewhere in more detail. Here we give only a brief overview of some of the possibilities.

As noted earlier, linear interpolation between the vertices of the computed triangulation defines a simplicial approximation M_Σ of the corresponding part of the manifold M. The points of M_Σ can be used to compute further points of M. For example, we may project any such point onto M by applying the corrector iteration (2.11). Alternately, we can augment the system (2.2) with p appropriately chosen equations and then apply, say, a chord Newton method to calculate a corresponding point on M. This approach is useful when points on M with specific properties are desired. For example, we may be interested in certain target points where the parameters have prescribed values. In that case, these target conditions become the augmenting equations and we may start the iterative process from a point on M_Σ where the parameters have the specified values. Augmenting equations are also essential when we are interested in determining the specific location of certain types of singularities. For the

computation of limit points a comparison of various such augmentations was given in [13], and for higher order singularities the literature on suitable augmentations is fairly extensive. In our present setting, the "minimal" augmentations discussed in [11], and thereafter in [10], appear to be of particular interest.

For any given functional the computed data allow us to generate contour plots on the intersection of the simplicial approximation M_Σ with various subspaces. For instance, in some structural problem we might be interested in seeing lines, where some stress component is constant, plotted in dependence of certain other variables. A special example of such a contour plot involves the graphical representation of lines of foldpoints. As the figures of the previous section already show, our triangulations provide information for approximating segments of such foldlines. One approach for detecting foldpoints is to monitor the orientation of the projection of the tangent basis onto the parameter space. If there is a change in this orientation then we have passed through a foldpoint, but the converse is not necessarily true; that is, not every foldpoint can be detected this way. The orientation is characterized by the determinant of the projected basis in the parameter space. Thus, if we plot lines of constant determinant values, then lines of zero determinant are approximations of the desired foldlines. Of course, for this we need the tangent basis at each vertex of the triangulation and that increases the cost of the overall algorithm. However, there are also other possible techniques for approximating foldlines from data obtained by our triangulation algorithm. This will not be pursued here.

Even though these contour plots only provide lines of constant values on the simplicial approximation M_Σ rather than on M itself, they tend to offer already good insight into the shape of the manifold. Of course, as discussed earlier, we can always call on various local corrector methods to project these lines onto M itself.

So far we mentioned only the need for appropriate post-processing techniques for analyzing the output of our triangulation method. There is also considerable room for improving the algorithm itself. In particular, for large sparse problems the QR-factorization may be computationally expensive. As noted earlier, there exist results for computing sparse bases of the null space of a matrix (cf. [5, 6]). The rotation required for the moving frame algorithm is likely to destroy this sparsity, and hence the rotated basis should not be stored but computed only as needed. In the case of low dimensional manifolds, this is highly desirable when the computation of the original basis of the tangent space takes account of the sparsity structure of the Jacobian.

References

1. Allgower, E.L., Georg, K.: Generation of Triangulations by Reflections. Util. Math. **16**, 123–129 (1979)
2. Allgower, E.L., Schmidt, P.H.: An Algorithm for Piecewise-linear Approximation of an Implicitly Defined Manifold. SIAM J. Numer. Anal. **22**, 322–346 (1985)
3. Babuska, I., Rheinboldt, W.C.: Adaptive Finite Element Processes in Structural Mechanics. In: Elliptic Problem Solvers II, Birkhoff, G., Schoenstadt, A., eds.), pp. 345–378. New York: Academic Press

4. Bohl, E.: Finite Modelle gewöhnlicher Randwertaufgaben. Stuttgart: Teubner 1981
5. Berry, M.W., Heath, M.T., Kaneko, I., Law, M., Plemmons, R.J., Ward, R.C.: An Algorithm to Compute a Sparse Basis of the Null Space. Numer. Math. **47**, 483–504 (1985)
6. Coleman, T.F., Pothen, A.: The Null Space Problem II: Algorithms, Cornell University, Department of Computer Science, Technical Report TR 86-747, April 1986
7. Coleman, T.F., Sorensen, D.C.: A Note on the Computation of an Orthonormal Basis for the Null Space of a Matrix. Math. Program. **29**, 234–242 (1984)
8. Deuflhard, P., Heindl, G.: Affine Invariant Convergence Theorems for Newton's Method and Extensions to Related Methods. SIAM J. Numer. Anal. **16**, 1–10 (1979)
9. Fink, J.P., Rheinboldt, W.C.: Solution Manifolds of Parametrized Equations and their Discretization Error. Numer. Math. **45**, 323–343 (1984)
10. Fink, J.P., Rheinboldt, W.C.: A Geometric Framework for the Numerical Study of Singular Points. University of Pittsburgh, Institute for Computer Mathematics and Application, Technical Report ICMA-86-90, January 1986. SIAM J. Numer. Anal. (in press)
11. Griewank, A., Reddien, G.W.: Characterization and Computation of Generalized Turning Points. SIAM J. Numer. Anal. **21**, 176–185 (1984)
12. Golub, G.H., VanLoan, C.F.: Matrix Computations. Baltimore: The Johns Hopkins University Press 1983
13. Melhem, R.G., Rheinboldt, W.C.: A Comparison of Methods for Determining Turning Points of Nonlinear Equations. Computing **29**, 201–226 (1982)
14. Rheinboldt, W.C.: Numerical Methods for a Class of Finite-dimensional Bifurcation Problems. SIAM J. Numer. Anal. **17**, 221–237 (1980)
15. Rheinboldt, W.C.: Numerical Analysis of Parametrized Nonlinear Equations. New York: Wiley
16. Rheinboldt, W.C., Burkardt, J.V.: A Locally Parametrized Continuation Process. ACM Trans. Math. Software **9**, 236–246 (1983)
17. Spivak, M.: A Comprehensive Introduction to Differential Geometry. Five Volumes, Second Ed. Berkeley: Publish or Perish 1979
18. Todd, M.J.: The Computation of Fixed Points and Applications. New York, Berlin, Heidelberg: Springer 1976
19. Walker, A.C.: A Non-Linear Finite Element Analysis of Shallow Circular Arches. Int. J. Solids Struct. **5**, 97–107 (1969)

Received December 30, 1986