A Locally Parameterized Continuation Process

WERNER C. RHEINBOLDT and JOHN V. BURKARDT
University of Pittsburgh

The design of a package of continuation procedures called PITCON to handle the following tasks is described· (1) follow numerically any a priori specified curve on an equilibrium manifold; (2) on such a curve determine the exact location of target points where a given variable has a specified value; and (3) on such a curve identify and compute exactly the (simple) limit points where stability may be lost. The process is based on the local parameterization which uses an estimate of the curvature to control the choice of parameter variable.

Categories and Subject Descriptors. G 1.5 [**Numerical Analysis**]· Roots of Nonlinear Equations— *iterative methods, systems of equations*

General Terms· Algorithms, Design

Additional Key Words and Phrases Equilibrium problems, underdetermined systems of equations, solution manifolds of parameterized equations, continuation methods, local parameterization, limit point computation

The Algorithm A Program for a Locally Parameterized Continuation Process. *ACM Trans. Softw. 9*, 2 (June 1983), 236–241

1. INTRODUCTION

The study of many equilibrium phenomena leads to nonlinear equations which involve a number of intrinsic parameters. Interest then centers rarely on the determination of a few specific solutions of the equations for fixed parameter values, but rather on an assessment of the behavior of these solutions under general variations of the parameters. For example, in structural analysis the parameters may characterize load points and load directions, material properties, or geometrical data. The set of all solutions and associated parameter values has been called the equilibrium surface of the structure (see, e.g., [33]). This equilibrium surface provides considerable insight into the behavior of the structure and the stability properties (see, e.g., [25] and [34] for further discussions and various examples). From a numerical viewpoint the question then is to analyze computationally the shape and characterize particular features of this equilibrium surface.

In nonlinear mechanics the principal tools for such a computational analysis are the so-called incremental methods. These procedures were developed more or less independently in the engineering literature. But they are now also recognized to be closely related to the continuation methods used for some time in mathematics in general and in numerical analysis in particular. The literature in this area is extensive; we refer only to [23] for a discussion about the connection between incremental approaches for structural problems and continuation methods, to [9] for a historical overview of uses of continuation techniques in mathematics, and to [2] and [37] for some literature survey of numerical aspects of continuation methods.

Not surprisingly there are differences between the methods used in structural engineering and numerical analysis, and neither is directly suited to the analysis of an equilibrium surface. In the numerical analysis literature continuation methods are usually considered only as tools for determining a specific solution y^* of a given nonlinear operator equation $Gy = 0$. For this the equation is embedded into a one-parameter family $H(y, t) = 0$, which has a solution $y = y(t)$ for each fixed t in some interval, say, $0 \leq t \leq 1$. (See, e.g., [16] for a survey of such embeddings.) If $y(t)$ depends continuously on t and satisfies $y(0) = y^0$ and $y(1) = y^*$, where y^0 is a known point, then the numerical process constructs a sequence of points in the proximity of the path $y(t)$, $0 \leq t \leq 1$, starting at y^0 and ending at the desired point y^*. On the other hand, in structural mechanics incremental methods usually are designed to follow numerically a specific load curve parameterized by a load intensity. Hence, while in the embedding approach the parameter is essentially artificial, in the incremental procedures it has an intrinsic meaning for the application, and, even more important, there is no longer a fixed endpoint which is the aim of the computation, but the load curve itself is of interest.

For a numerical analysis of a given equilibrium surface we need to consider continuation-methods in a broader sense as a collection of numerical procedures for completing at least the following three basic tasks:

(i)   Follow numerically any curve on the surface specified by a particular combination of parameter values with one degree of freedom.

(ii) On any such curve determine the exact location of target points where a given state variable has a specified value.

(iii) On such a curve identify and compute exactly the critical points where stability may be lost.

Beyond this various more special tasks may arise as, for example, the following ones:

(iv) From any one of the critical points determined under (iii) follow a path in the critical boundary.

(v)   On any one of the curves (i) determine the location of bifurcation points and the paths intersecting at that point.

Methods which are either directly applicable or can be readily adapted to completing these various tasks have been proposed by various authors. In particular, for (i) the literature is very large and we refer here only to the mentioned surveys [2, 37]. Methods relating to (iii) were described, for instance,

in [1], [22], [24], [35], and [36], and for (iv) and (v) we refer to [30], and to [11] and [27], respectively, where further references are also given.

So far only a few library programs for performing these various tasks have been published. Without claim for completeness we mention here [15] and [40]. Each one of these programs has the objective of computing a specific solution of a nonlinear equation by a homotopy-continuation approach along the lines sketched above. In this paper, we describe the design of a new library program, called PITCON, specifically written with the objective of completing the three basic tasks (i)–(iii). The program itself is presented in a companion paper (A program for a locally parameterized continuation process, this issue, pp. 236–241). It is written in the form of a subroutine which can be incorporated into various main programs. For instance, if little is known about the problem, an interactive driver may be desirable. However, since such main programs are very problem dependent they will not be discussed here any further. PITCON may be expanded to incorporate facilities for (iv) and (v), but since these tasks are of a more specified nature, it was decided not to burden the programs with the added complexity. The algorithms are based on the continuation approaches introduced in [28] and [29] and incorporate some of the concepts of step-length determination discussed in [8]. At the same time, new techniques of parameter adaptation are utilized here based on a prediction of changes in the curvature of the continuation path.

2. BASIC FORMULATION

Generally, after suitable discretizations, the equilibrium problems mentioned in the introduction lead to a finite-dimensional, nonlinear equation of the form

$$G(y, p) = 0, \tag{2.1}$$

where $y \in R^m$ is a vector of state variables, $p \in R^r$ a vector of parameters, and G: $R^m \times R^r \to R^m$ a given function. Then we are interested in the features of the set

$$\mathcal{E}(G) = \{(y, p) \in R^m \times R^r; G(y, p) = 0\} \tag{2.2}$$

of all solutions of (2.1). Under well-known conditions $\mathcal{E}(G)$ represents an r-dimensional manifold in $R^m \times R^r$.

In most applications, interest centers on tracing paths on $\mathcal{E}(G)$ which are characterized by $r - 1$ relations between the parameters. In order words, we are given a suitable mapping $K: R^r \to R^{r-1}$ and wish to compute the subset of $\mathcal{E}(G)$ defined by the augmented equations

$$G(y, p) = 0,$$
$$Kp = 0. \tag{2.3}$$

In this formulation we should include the parameters in the list of variables, in which case (2.3) represents a system with one more variable than equations. Then, for ease of notation, it is reasonable to combine the vectors y and p into one vector x of dimension $n = m + r$. Moreover, from the viewpoint of our package of library programs it is natural to assume that both mappings G and K of (2.3) are provided for by the user. In other words, we may write (2.3) as one equation,

$$Fx = 0, \tag{2.4}$$

with a user-specified mapping $F: R^n \to R^{n-1}$. Note, however, that in this underdetermined equation (2.4) no one variable is explicitly identified as continuation variable as is typical in the incremental and continuation methods mentioned in the introduction.

We assume here that the given mapping F has the following properties:

(i) F is continuously differentiable on R^n.
(ii) The derivative $DF(x)$ of F is locally Lipschitzian on R^n.
(iii) The regularity set $\mathscr{R}(F) = \{x \in R^n; \operatorname{rank} DF(x) = n - 1\}$ is nonempty and therefore an open subset of R^n.

From (i)–(iii) it follows (see [28]) that the tangent map specified by

$$T: \mathscr{R}(F) \to R^n, \qquad DF(x)Tx = 0, \qquad \|Tx\|_2 = 1, \qquad \det\!\left(\frac{DF(x)}{(Tx)^{\mathrm{T}}}\right) > 0 \quad (2.5)$$

is uniquely determined and locally Lipschitzian on $\mathscr{R}(F)$. Furthermore, (i)–(iii) imply that the regular solution set $\mathscr{E}(F) \cap \mathscr{R}(F)$ of F is either empty or a one-dimensional C^1-manifold in the open set $\mathscr{R}(F)$. Our objective is to determine numerically a nonempty connected component $\mathscr{E}^*$ of $\mathscr{E}(F) \cap \mathscr{R}(F)$. It is well-known (see, e.g., [18]) that such a component $\mathscr{E}^*$ is diffeomorphic either to the circle or to some interval (that is, some connected subset) of R^1. Hence, $\mathscr{E}^*$ is uniquely determined by any one of its points $x^0 \in \mathscr{E}(F) \cap \mathscr{R}(F)$, and we denote this by writing $\mathscr{E}^*(F, x^0)$. Note that for any $x^1 \in \mathscr{E}^*(F, x^0)$ we have $\mathscr{E}^*(F, x^1) = \mathscr{E}^*(F, x^0)$.

A parameterization by arc length of $\mathscr{E}^*(F, x^0)$ is a solution of the initial-value problem

$$\dot{x} = Tx, \qquad x(0) = x^0. \tag{2.6}$$

Note that, since T is locally Lipschitzian, (2.6) has a unique solution which cannot terminate inside $\mathscr{R}(F)$. Evidently standard ODE-solvers may be applied to solve (2.6) numerically. This has been pursued for some time in the literature (see, e.g., [4], [7], [14], [39]). Independent of this, the choice of the arc length for the parameterization of $\mathscr{E}^*(F, x^0)$ has been proposed by many authors. Notably H. B. Keller and his co-workers (see, e.g., [11] and [12]) have advocated this choice for some time. It is also the basis of incremental procedures given in [6] and [31] and has been more or less implicit in various papers in the field.

Our algorithms are based more generally on the structure of $\mathscr{E}^*(F, x^0)$ as a one-dimensional manifold and use a local parameterization at each point computed along $\mathscr{E}^*(F, x^0)$. A natural class of such local parameters are the n components of the vector x. We call a process based on this choice of parameterization a locally parameterized continuation method.

3. OUTLINE OF THE PROCESS AND BASIC STEPS

As noted before, our objective is to determine numerically a nonempty component $\mathscr{E}^*(F, x^0)$ of the regular solution set $\mathscr{E}(F) \cap \mathscr{R}(F)$. For the discussion it is useful to consider a parameterization by an arc length of $\mathscr{E}^*(F, x^0)$, that is, a function x: $J \to \mathscr{E}^*(F, x^0)$ which maps some interval $J \subset R^1$ diffeomorphically onto some open subset of $\mathscr{E}^*(F, x^0)$, such that $\|\dot{x}(s)\|_2 = 1$ for $s \in J$. We may assume also that $x(0) = x^0$, $0 \in J$.

The process described here belongs to the class of predictor-corrector continuation methods. Starting from x^0 it produces a sequence of approximations $x^k \doteq x(s_k)$, $k = 0, 1, \ldots$, corresponding to some sequence $0 = s_0 < s_1 < s_2 < \cdots$ of arc length values. Note, however, that in general the values s_1, s_2, ... are only approximately computable and are of limited interest in most applications.

In our algorithm the principal steps performed during one continuation step are as follows:

(1) Initialization.
(2) Check for and computation of target point, if desired.
(3) Calculation of tangent vector and determination of new local continuation parameter.
(4) Check for and computation of limit point, if desired.
(5) Step-length computation.
(6) Computation of predicted point and corrector iteration.
(7) Storage of data and return.

The sequencing of these steps is dictated by the data flow. For the description of the details it will be advantageous not to adhere to this sequence. Instead, in the remainder of this section, we discuss the basic steps (3) and (6). Then the next section introduces the new step-length computation used in step (5), and Section 5 covers steps (2) and (4). The data-handling steps (1) and (7) should be self-explanatory from the documentation of the program itself.

Let $e^1, \ldots, e^n$ be the natural basis vectors of R^n. Then it is readily verified that (see, e.g., [28])

$$\det\begin{pmatrix} DF(x) \\ (e^i)^T \end{pmatrix} = [(e^i)^T Tx]\det\begin{pmatrix} DF(x) \\ (Tx)^T \end{pmatrix}, \qquad \forall x \in \mathcal{R}(F), \quad i = 1, \ldots, n, \quad (3.1)$$

where the matrix occurring on the right is nonsingular. Hence, for any index i, $1 \le i \le n$, such that $(e^i)^T Tx \ne 0$, the solution $v \in R^n$ of the linear system

$$\begin{pmatrix} DF(x) \\ (e^i)^T \end{pmatrix} v = e^n \qquad (3.2)$$

is uniquely defined. Evidently, then

$$Tx = \sigma \frac{v}{\|v\|_2}, \qquad (3.3)$$

and, in line with (2.5), we should set

$$\sigma = \text{sgn}(v^T e^i)\text{sgn} \det\begin{pmatrix} DF(x) \\ (e^i)^T \end{pmatrix}. \qquad (3.4)$$

As long as the solution path remains completely in $\mathcal{R}(F)$, this is satisfactory. But frequently, in applications, we may encounter a bifurcation point $x^* \notin \mathcal{R}(F)$ which appears to be the intersection of two solution paths (Figure 1). More specifically, using arc length representations, we have solution branches $x^j: J_j \subset R^1 \to \mathcal{R}(F), j = 1, \ldots, 4$, for which $x^j(s)$ tends to x^* when s tends to one of the endpoints of J_j, such that—except for their direction—the derivatives are continuous when passing through x^* from x^1 to x^2 or from x^3 to x^4. In such a case,
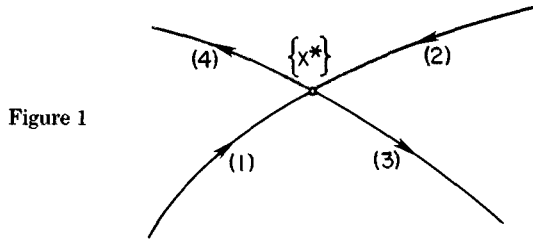
Figure 1

when the process moves along x^1 toward x^*, it usually "jumps" over x^* onto x^2. Then, unless we reverse the sign of σ in (3.3), the tangent will again point toward x^* and the process reverses direction.

In order to avoid this problem, suppose that the point x in (3.3) is the kth approximation computed along the curve. Then σ is determined as follows:

$$\sigma = \begin{cases} \text{dir} & \text{if } k = 0, \\ +1 & \text{if } \operatorname{sgn} v^\mathrm{T} e^i = \operatorname{sgn}(Tx^{k-1})^\mathrm{T} e^i, \\ -1 & \text{otherwise,} \end{cases} \tag{3.5}$$

where dir is a user-specified direction at the starting point. By comparing this value of σ with that of (3.4), we can detect if the process did jump over a bifurcation point of odd multiplicity. Obviously, bifurcation points of even multiplicity cannot be found this way.

Once the tangent Tx^k has been obtained we determine the indices j_1 and j_2 of the largest and second largest component of Tx^k in modulus, respectively. The relation (3.1) certainly suggests that the index i_k, $1 \leq i_k \leq n$, of the new local continuation variable be set equal to j_1. However, if we are approaching a limit point in the j_1-th variable, then this choice may be disadvantageous. Accordingly, if the following three conditions are simultaneously satisfied

$$\begin{array}{ll} \text{(i)} & |(e^{j_1})^\mathrm{T} Tx^k| < |(e^{j_1})^\mathrm{T} Tx^{k-1}|, \\ \text{(ii)} & |(e^{j_2})^\mathrm{T} Tx^k| > |(e^{j_2})^\mathrm{T} Tx^{k-1}|, \\ \text{(iii)} & |(e^{j_2})^\mathrm{T} Tx^k| \geq \mu |(e^{j_1})^\mathrm{T} Tx^k|, \end{array} \tag{3.6}$$

with a fixed μ, $0 < \mu < 1$, then we set $i_k = j_2$. Of course, if we do not have a previous tangent vector, this check has to be bypassed. The new continuation index i_k will be used for the computation of the next point x^{k+1} and its tangent Tx^{k+1}. For the tangent computation at x^0 a continuation index i_0 is assumed to be given by the user. This choice of i_0 has to be such that the i_0th component of Tx^0 is nonzero. Moreover, in some cases x^0 is only known approximately and the corrector, described later, has to be applied to improve the initial point. During this process the i_0th component of the input point is kept fixed and hence the choice of i_0 has a strong influence on the convergence properties of the initial correction process.

It should be noted also that the conditions (3.6) are not invariant under any scaling of the variables. In the formulation of the original equations care should be taken to ensure that the variation of the different variables are about of the same order of magnitude. Otherwise the continuation index i_k may remain fixed for long periods and failures may arise.

With the tangent Tx^k and the step length $h_k > 0$ determined by the step length algorithm of Section 4, we now compute the predicted point $\hat{x}^k = x^k + h_k Tx^k$. Then any appropriate iterative method for the solution of the augmented equation

$$\hat{F}x \equiv \begin{pmatrix} Fx \\ (e^{i_k})^T(x - \hat{x}^k) \end{pmatrix} = 0 \tag{3.7}$$

starting from $\hat{x}^k$ may be used as a corrector process. We use either the regular Newton method or its modified form in which the Jacobian at the starting point is held fixed.

Let $y^0 = \hat{x}^k, y^1, y^2, \ldots$ be the iterates produced in this way. The process has to incorporate provisions for monitoring the convergence and for aborting the iteration as soon as divergence is suspected. In the program nonconvergence is declared if any one of the following three conditions is true:

$$
\begin{array}{lll}
\text{(i)} & \|\hat{F}y^j\| \geq \theta \|\hat{F}y^{j-1}\| & \text{for some } j \geq 1, \\
\text{(ii)} & \|y^j - y^{j-1}\| \geq \theta \|y^{j-1} - y^{j-2}\| & \text{for some } j \geq 2, \\
\text{(iii)} & j \geq j_{\max}.
\end{array}
\tag{3.8}
$$

For the constant θ we use $\theta = 1.05$, except in the first check of (3.8) (i), where $\theta = 2$ is chosen. The maximal iteration count $j_{\max}$ depends on the method. For the regular Newton process we set $j_{\max} = 10$ and double this for the modified method. In the case of nonconvergence the predictor step is reduced by a given factor, for example $\frac{1}{3}$, unless the resulting step is below a given minimal step length.

Convergence is declared if either one of two types of criteria are satisfied which—for want of a better name—we call the strong and the weak acceptance criteria. These tests use the smallest floating-point number ϵ_{mach} defined by $1.0 = 1.0 + \epsilon_{\text{mach}}$, as well as two user-specified tolerances $\epsilon_{\text{abs}}, \epsilon_{\text{rel}}$, and the relative error measure

$$\epsilon(y) = \epsilon_{\text{abs}} + \epsilon_{\text{rel}}\|y\|. \tag{3.9}$$

The strong acceptance criterion then requires that

$$(\|\hat{F}y^j\| \leq \epsilon_{\text{abs}}) \quad \text{and} \quad (\|y^j - y^{j-1}\| \leq \epsilon(y^j)) \quad \text{for some } j \geq 1. \tag{3.10}$$

This is a standard condition; but in certain cases, an iterative process may behave erratically near a solution due to ill-conditioning or unrealistic error tolerances. Typically, this manifests itself in one of the conditions (3.10) being repeatedly satisfied, while the other one holds only with a slightly larger tolerance. Accordingly, under the weak acceptance criterion, convergence is declared if either one of the following three conditions holds:

$$
\begin{array}{ll}
\text{(i)} & \|\hat{F}y^j\| \leq 8\epsilon_{\text{mach}} \\
& \quad \text{for some } j \geq 0, \\
\text{(ii)} & (\|\hat{F}y^j\| + \|\hat{F}y^{j-1}\| \leq \epsilon_{\text{abs}}) \quad \text{and} \quad (\|y^j - y^{j-1}\| \leq 8\epsilon(y^j)) \\
& \quad \text{for some } j \geq 1, \\
\text{(iii)} & (\|\hat{F}y^j\| \leq 8\epsilon_{\text{abs}}) \quad \text{and} \quad (\|y^j - y^{j-1}\| + \|y^{j-1} - y^{j-2}\| \leq \epsilon(y^j)) \\
& \quad \text{for some } j \geq 2.
\end{array}
\tag{3.11}
$$

Weak acceptance is always indicated by a special flag to allow the user to take different actions. Throughout (3.8)–(3.11) the maximum norm is used.

It should be noted that appropriate choices for the error tolerances are not only problem dependent but also path dependent. Thus, a user may have to experiment with different tolerances for a specific problem. If the equations under consideration are obtained by some discretization of an infinite-dimensional problem, then it should be taken into account that the discretization error also changes along the path (see, e.g., [3]).

4. THE STEP-LENGTH ALGORITHM

For the points x^k, $k = 0, 1, \ldots$, approximating the continuation curve $x: J \to \mathscr{E}^*(F, x^0)$ the achievable error $\| x^k - x(s_k) \|$ is solely determined by the termination criterion (3.9) of the corrector process. In contrast to this the standard ODE-solvers involve a corrector equation obtained by extrapolation for which the solutions are not, in general, on the exact curve. As a consequence the available error for the ODE-solvers depends on the history of the process up to that point, and this in turn has a strong influence on the step selection. On the other hand, for our continuation process any step $h_k > 0$ along the Euler line is acceptable in principle if only the corrector converges from the predicted point $\hat{x}^k$. Moreover, in [28] it was shown that any compact segment of the continuation curve in $\mathscr{R}(F)$ has an ϵ-neighborhood for some $\epsilon > 0$ in which Newton's method will converge to the curve.

This suggests that we estimate the radius of convergence of the corrector process at the computed points and extrapolate these radii to the next point about to be determined. In practice, the estimate of a convergence radius at some continuation point would have to be based on the corrector iterates which led to that point. Unfortunately, as was proved in [8], this represents insufficient information for obtaining such an estimate. On the other hand, an approach was presented in [8] which allows for an assessment of the convergence quality of the particular sequence of corrector iterates.

For details of this approach we refer to the cited article. In brief, let $\{y^i\}$ be a given sequence with limit y^* generated by an iterative process and denote the errors by $e_i = \| y^i - y^* \|$, $i = 0, 1, \ldots$. The definition of any convergence measure is based on a hypothetical model of the behavior of the errors. For example, if $\{y^i\}$ converges linearly it is reasonable to assume that

$$0 \le e_{i+1} \le \lambda e_i, \qquad i = 0, 1, \ldots \tag{4.1}$$

with some constant λ, $0 < \lambda < 1$, depending on $\{y^i\}$. Suppose now that the process was terminated with the iterate y^{i^*}. Then

$$\tilde{\lambda} = \tilde{\omega}^{1/(i^*-1)}, \qquad \tilde{\omega} = \frac{\| y^{i^*} - y^{i^*-1} \|}{\| y^{i^*} - y^0 \|}, \qquad i^* \ge 2, \tag{4.2}$$

represents a computable estimate of λ.

In the setting of our continuation process suppose now that the y^i, $i = 0, 1, \ldots$, are the corrector iterates leading from the current predicted point $\hat{x}^k = y^0$ to the new continuation point $x^{k+1} = y^{i^*}$. Then

$$\delta_k = \| \hat{x}^k - x^{k+1} \| = \| y^0 - y^{i^*} \| \tag{4.3}$$

is the correction distance. For the modified Newton method the convergence is indeed linear, and a reasonable aim in the construction of the steps along the curve is to ensure that the number of corrector iterates remains about constant. In other words, we aim at taking always, say, m^* corrector steps. Hence, under the heuristic assumption that the error model (4.1) remains valid for some interval of starting errors e_0 around δ_k, we should have begun with an "ideal starting error" $\delta_k^* = \theta_k \cdot \delta_k$ such that

$$\tilde{\lambda}^{m^*} \delta_k^* \doteq \tilde{\lambda}^{*} \delta_k \tag{4.4}$$

and therefore

$$\theta_k = \tilde{\lambda}^{i^* - m^*} \equiv \tilde{\omega}^{(i^* - m^*)/(i^* 1)}. \tag{4.5}$$

In the algorithm we use $m^* = 10$ for the modified Newton method and enforce always that $0.125 \le \theta_k \le 8$.

This technique is also readily applicable for Newton's method. In [8] two different hypothetical error models for the Newton process are discussed. Here we use only one of these models, namely, the one arising in the attraction theorem formulated in [26]. In essence, under certain conditions about the equation and the desired limit y^* of the Newton process, there exists a radius $r^* > 0$ such that for any starting point y^0 in the ball $B(y^*, r^*)$ the relative errors $\epsilon_i = e_i/r^*$, $i = 0$, $1, \ldots$, satisfy

$$0 \le \epsilon_{i+1} \le \phi(\epsilon_i), \qquad i = 0, 1, \ldots, \qquad \phi(t) = \frac{t^2}{3 - 2t}, \qquad 0 \le t \le 1. \tag{4.6}$$

The radius r^* depends on global information about the equation and is not accessible. If $0 < \epsilon_0 < 1$ and the $\{\epsilon_i\}$ satisfy (4.6), then we have

$$\epsilon_i \le \eta_i \equiv \phi^i(\eta_0) \equiv \frac{3}{1 + 2 \cosh 2^i \alpha}, \qquad i = 0, 1, \ldots, \eta_0 = \epsilon_0, \tag{4.7}$$

where α is the unique positive solution of

$$\psi(\alpha) = \eta_0, \qquad \psi(\alpha) = \frac{3}{1 + 2 \cosh \alpha}. \tag{4.8}$$

Moreover, for any ω, $0 < \omega < 1$, and $i^* \ge 2$ the equation

$$\frac{1}{\psi(\alpha)} \phi^{i^*-1}(\psi(\alpha)) \equiv \frac{1 + 2 \cosh \alpha}{1 + 2 \cosh 2^{i^*-1} \alpha} = \omega \tag{4.9}$$

has a unique solution $\alpha > 0$.

Now suppose that $\{y^i\}$ denotes the sequence of Newton iterates and that the process was terminated at y^{k^*}. As in the linear case we use the approximation

$$\tilde{\omega} = \frac{\|y^{i^*} - y^{i^*-1}\|}{\|y^{i^*} - y^0\|} \doteq \frac{e_{i^*-1}}{e_0} = \frac{\epsilon_{i^*-1}}{\epsilon_0} \le \frac{\eta_{i^*-1}}{\eta_0} \tag{4.10}$$

and compute with this $\tilde{\omega}$ the solution $\tilde{\alpha}$ of (4.9) which gives the estimate $\tilde{\eta}_0 = \psi(\tilde{\alpha})$ of ϵ_0. Now we proceed as before and obtain the factor

$$\theta_k^* = \frac{\delta_k^*}{\delta_k} = \frac{\tilde{\eta}_0'}{\tilde{\eta}_0} \tag{4.11}$$

Table I

	$\tilde{\omega} \in [a, b]$		
i^*	a	b	θ_k
2	0.8735115	1	1
	0.1531947	0.8735115	$0.9043128 - 0.7075675 \ln \omega$
	0.03191815	0.1531947	$-4.667383 - 3.677482 \ln \omega$
	0	0.03191815	8
3	0.4677788	1	1
	0.6970123(−3)	0 4677788	$0 8516099 - 0.1953119 \ln \omega$
	0.1980863(−5)	0 6970123(−3)	$-4 830636 - 0.9770528 \ln \omega$
	0	0 1980863(−5)	8
4	0	1	1
5	0.3339946(−10)	1	$1.040061 + 0.03793395 \ln \omega$
	0	0 3339946(−10)	0.125
6	0.1122789(−8)	1	$1.042177 + 0.04450706 \ln \omega$
	0	0 1122789(−8)	0.125
≥7	0	1	0.125

for the ideal starting error by determining the unique solution $\tilde{\eta}'_0$, $0 < \tilde{\eta}'_0 < 1$, of

$$\phi^{m^*}(\tilde{\eta}'_0) = \phi^{i^*}(\tilde{\eta}_0). \tag{4.12}$$

Since the iterates ϕ^i are explicitly known, the various equations are not difficult to solve numerically. However, for the computation it is more advantageous to introduce a least squares fit of θ_k as a function of $\tilde{\omega}$ for all relevant values of i^*. In the program we use $m^* = 4$ and the approximations for θ_k given in Table I. Note that as before we restrict θ_k to the interval $0.125 \leq \theta_k \leq 8$.

We turn now to the algorithm for the determination of the step length $h_k > 0$ along the Euler line $\pi(t) = x^k + t\,Tx^k$ used for the prediction. In order to estimate the distance between $\pi(t)$ and the exact curve $x = x(s)$ we introduce the quadratic Hermite-Birkhoff interpolation polynomial

$$q(t) = x^k + t\,Tx^k + \tfrac{1}{2}t^2 w^k, \tag{4.13}$$

$$w^k = \frac{1}{\Delta s_k}(Tx^k - Tx^{k-1}), \qquad \Delta s_k = \|x^k - x^{k-1}\|_2,$$

for which

$$q(0) = x^k, \qquad q'(0) = Tx^k, \qquad q'(-\Delta s_k) = Tx^{k-1}. \tag{4.14}$$

Since

$$w^k \doteq \int_0^1 x''(s_k - \sigma\Delta s_k)\, d\sigma = x''(s_k - \tilde{\sigma}\Delta s_k), \qquad 0 < \tilde{\sigma} < 1, \tag{4.15}$$

the quantity

$$\|w^k\|_2 = \frac{2}{\Delta s_k}\left|\sin\frac{1}{2}\alpha_k\right|, \qquad \alpha_k = \arccos((Tx^k)^T Tx^{k-1}) \tag{4.16}$$
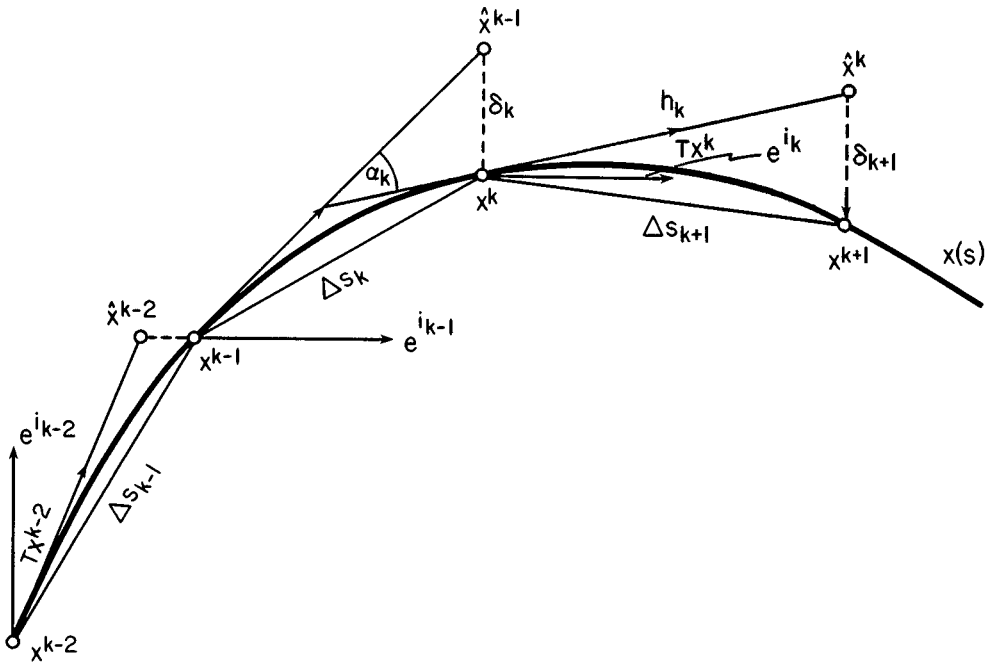
Figure 2

represents an approximation of the curvature of the exact point at some point between $x(s_{k-1})$ and $x(s_k)$.

It is tempting to derive from q a prediction of the curvature to be expected during the next continuation step. However, a closer computation shows that the value of the curvature of q assumes its maximum $\|w^k\|_2/\cos^2 \frac{1}{2}\alpha_k$ at $t = -\frac{1}{2}\Delta s_k$ and that, for increasing t, this value decreases rapidly. For example, at $t = 0$ the curvature of q equals only $\|w^k\|_2|\cos \frac{1}{2}\alpha_k|$ and for positive t no reasonable predictive information can be gained this way.

The relation (4.15) suggests the use of the simple linear extrapolation

$$\gamma_k^{\text{tent}} = \|w^k\|_2 + \frac{\Delta s_k}{\Delta s_k + \Delta s_{k-1}} (\|w^k\|_2 - \|w^{k-1}\|_2) \qquad (4.17a)$$

for a prediction of the curvature during the next continuation step. However, this value may become negative and accordingly we use instead

$$\gamma_k = \max(\gamma_{\min}, \gamma_k^{\text{tent}}) \qquad (4.17b)$$

with a given small $\gamma_{\min} > 0$.

Most of the data discussed so far are sketched in Figure 2. In order to derive a formula for the desired predictor step h_k, we note that

$$\|q(t) - \pi(t)\|_2 = \frac{1}{2}t^2\|w^k\|_2 \qquad (4.18)$$

represents an estimate of the distance between the Euler line and the exact curve. In fact, for smooth curves the error of this estimate is asymptotically of order

three in $\max(|t|, \Delta s_k)$ as this quantity tends to zero. Hence, if we want this distance to be at most equal to a tolerance $\epsilon > 0$, then we should choose the next step as

$$t = \sqrt{\frac{2\epsilon}{\|w^k\|_2}}. \tag{4.19}$$

It is natural to replace the curvature $\|w^k\|_2$ by the predicted value γ_k of (4.17) and to relate the tolerance ϵ to the "ideal starting error" δ_k^* obtained earlier. As Figure 2 indicates it is unreasonable to expect $\epsilon > \Delta s_k$. Hence, we use instead

$$\epsilon_k = \begin{cases} \epsilon_{min}\Delta s_k & \text{if} \quad \delta_k^* \leq \epsilon_{min}\Delta s_k, \\ \Delta s_k & \text{if} \quad \delta_k^* \geq \Delta s_k, \\ \delta_k^* & \text{otherwise,} \end{cases} \tag{4.20}$$

with a small $\epsilon_{min} > 0$, for example, $\epsilon_{min} = 0.01$. Then a tentative predicted step is given by

$$h_k^{(1)} = \sqrt{\frac{2\epsilon_k}{\gamma_k}}. \tag{4.21}$$

From the form (3.7) of the augmented equation we see that the corrector iterates remain in a hyperplane perpendicular to the basis vector e^{i_k} through the predicted point. Then Figure 2 suggests that we adjust the predicted step length h_k so as to ensure that h_k will be approximately equal to Δs_{k+1}. There is no need to enforce this too rigidly. It suffices to define a new tentative step by the requirement

$$(e^i)^T \pi(h_k^{(2)}) = (e^i)^T q(h_k^{(1)})$$

whence

$$h_k^{(2)} = h_k^{(1)}\left[1 + \frac{h_k^{(1)}}{2\Delta s_k}\left(1 - \frac{(e^i)^T Tx^{k-1}}{(e^i)^T Tx^k}\right)\right]. \tag{4.22}$$

This formula may involve substractive cancellation and has to be evaluated in double precision.

The final value h_k of the step length is now obtained from $h_k^{(2)}$ by enforcing three different bounding requirements. First of all, if the previous continuation step from x^{k-1} to x^k was obtained only after a failure of the correction process and a corresponding reduction of the predicted step, then we should not allow h_k to exceed Δs_k. Second, as in the case of the ODE solvers, we need to control both the relative growth and the absolute size of the predictor step. Thus, we require that

$$\frac{1}{\kappa}\Delta s_k \leq h_k \leq \kappa \Delta s_k, \qquad h_{min} \leq h_k \leq h_{max}, \tag{4.23}$$

where κ is some factor, say, $\kappa = 3$, and h_{min}, h_{max} depend on the machine as well as the requirements of the problem. It should be obvious how the final step h_k is obtained from $h_k^{(2)}$ on the basis of these restrictions.

5. THE COMPUTATION OF TARGET AND LIMIT POINTS

By generating a sequence of solution points on a given curve, the continuation process reveals the shape of the curve, but there are often other items of interest that need to be studied as well. Our program is designed to pause during the continuation steps in order to seek out special points that the user has requested, namely, target and limit points.

A target point $x \in \mathscr{E}^*(F, x^0)$ is a point on the solution curve for which the component $x_\iota = (e^\iota)^T x$ with given index $i = IT$ has a prescribed value $\bar{x}_\iota = XIT$. Limit or turning points with respect to a given index $i = LIM$ are points $x \in \mathscr{E}^*(F, x^0)$ where the ith component $(e^\iota)^T Tx$ is zero. More specifically, since it is computationally unreasonable to attempt to compute zeros of even order, we are concerned only with limit points on the continuation curve where $(e^\iota)^T Tx(s)$ changes sign.

It might be mentioned that bifurcation points represent another interesting, special class of points. But in that case we are not only interested in the specific location of the point but also in the solution curves that branch off from it. This is exactly the task (v) listed in Section 1. The corresponding procedures (loc. cit.) would add considerably to the complexity of our program, and, since their utility tends to be of a more specialized nature, it was decided not to cover task (v) (nor (iv)).

As indicated before, the determination of a target or limit point represents an interruption in the normal flow of the continuation program. After at least one step has been taken, the program has available an old point x^{k-1}, a new point x^k, and the tangent vector Tx^{k-1}. Normally, then we turn to the computation of Tx^k, of the new step length, and finally of the next point x^{k+1}. But if the index IT or LIM is nonzero, then these computations are postponed for the search of a target or limit point, respectively. We discuss these cases separately.

Target Points. Suppose that a nonzero value of $i = IT$ and associated value $\bar{x}_\iota = XIT$ have been given. If $\bar{x}_\iota$ lies between $(e^\iota)^T x^{k-1}$ and $(e^\iota)^T x^k$, then it is assumed that a solution point $x \in \mathscr{E}^*(F, x^0)$ with $(e^\iota)^T x = \bar{x}_\iota$ is nearby. In this case a point

$$y(t) = (1 - t)x^{k-1} + tx^k, \qquad 0 \le t \le 1, \tag{5.1}$$

on the secant between x^{k-1} and x^k is determined such that $(e^\iota)^T y(t) = \bar{x}_\iota$. Now with the augmenting equation $(e^\iota)^T x = \bar{x}_\iota$ the corrector process is applied, and, if it terminates successfully, the resulting point is taken as the desired target. Otherwise, a failure is indicated for the target routine. In either case, the routine returns and on the next call the continuation loop will pick up from where it was interrupted. Note that in effect the target routine uses the IT-th variable for the local parameterization of the curve between x^{k-1} and x^k. This may be an inferior choice of parameter for the corrector, but it allows us to enforce that the resulting target point $x \in \mathscr{E}^*(F, x^0)$ indeed satisfies $(e^\iota)^T x = \bar{x}_\iota$. Clearly, for very large continuation steps we have no guarantee that all target points will be detected or that a target computation will succeed. Thus, the utility of the target routine will depend on the maximal allowed step size that has been chosen.

Limit Points. If the limit point index $i = \text{LIM}$ is nonzero, then a limit point determination is carried out after a target point search has been successfully or unsuccessfully completed, provided it was called for at all. Recall that we still have as current information the vectors x^{k-1}, x^k, and Tx^{k-1}. Now the new tangent Tx^k is evaluated and if $\text{sgn}(e^i)^T Tx^{k-1} \neq \text{sgn}(e^i)^T Tx^k$ for $i = \text{LIM}(\neq 0)$, then a limit point search is begun. For this the index $\hat{\imath}$ of the largest component in modulus of the secant direction $x^k - x^{k-1}$ is chosen as a local parameterization of the curve between x^{k-1} and x^k. However, if $\hat{\imath} = \text{LIM}$, then the index of the second largest component is taken. If $x: [s_{k-1}, s_k] \to \mathcal{E}^*(F, x^0)$ represents the segment of the curve between x^{k-1} and x^k, then $\hat{\imath}$ may be assumed to constitute the index of a local coordinate for which there exists a bijective parameter transformation $\phi: [0, 1] \to [s_{k-1}, s_k]$, such that $(e^{\hat{\imath}})^T y(t) = (e(e^{\hat{\imath}})^T x(\phi(t)))$, $0 \leq t \leq 1$, where $y(t)$ is defined by (5.1).

Hence, we may consider the function

$$g: [0, 1] \to R^1, \qquad g(t) = (e^i)^T Tx(\phi(t)), \qquad 0 \leq t \leq 1, \qquad (5.2)$$

and our problem is to determine a zero of g. Since by assumption sign $g(0) \neq$ sgn $g(1)$, a root finder of the Dekker–Brent type can be applied. More specifically, for any current iterate t_j, we have either sgn $g(0) \neq$ sgn $g(t_j)$ or sgn $g(1) \neq$ sgn $g(t_j)$. Suppose, for the moment, that the latter case applies. Then the root finder produces a new value $t_{j+1} \in (t_j, 1)$. For the evaluation of $x(\phi(t_{j+1}))$ and $g(t_{j+1})$ we use the augmenting equation $(e^{\hat{\imath}})^T x = (e^{\hat{\imath}})^T y(t_j)$ and apply the modified Newton method with

$$z = \frac{1 - t_{j+1}}{1 - t_j} x(t_j) + \frac{t_{j+1} - t_j}{1 - t_j} x^k \qquad (5.3)$$

as starting point. If it terminates successfully, then the final iterate is used as $x(t_{j+1})$, the tangent $Tx(t_{j+1})$ is evaluated, and we set $g(t_{j+1}) = (e^{\hat{\imath}}{}^T Tx(t_{j+1})$. It should be clear how the formulas have to be modified for the case sgn $g(0) \neq$ sgn $g(t_j)$.

Certainly, g is costly to compute and we require an efficient root finder to speed the convergence of the limit-point routine. A specially modified version of the routine given in [5] is used here. Clearly, as in the case of target points, we may fail to detect a limit point if the continuation steps are too large, and in such a situation the root finder may also fail to converge. In addition, the evaluation of g may run into difficulties when the desired limit point is near a bifurcation point.

Some literature about other limit-point algorithms was cited in the Introduction. A comparison of the performance of the above used method with various other algorithms may be found in [20].

6. SOME NUMERICAL EXAMPLES

The programs described here have been used extensively with excellent success on problems from many different areas. We include here only a few numerical examples to illustrate the operation of the programs.

Example 1. In order to present some details of the performance of the programs, we consider first a very small problem which was originally formulated in [10] and subsequently used as a test case by many authors. The mapping F

Table II

Step	Continuation point			Continuation variable	Total correct steps	Comments
	x_1	x_2	x_3			
0	15.000	−2.00000	0.00000	x_3	—	
1	14.710	−1.9421	0.065381	x_1	2	
2	14 285	−1 7291	0.26874	x_3	3	
3	16 906	−1.2094	0.54684	x_2	1	
4	24.918	−0.59908	0.55514	x_1	3	
5	48 974	0 71803	−0.080761	x_1	3	
6	57.988	1 2846	−0.40736	x_1	4	Step reductions
7	60.052	1.5709	−0.54036	x_1	4	Step reductions
8	61.666	2.0010	−0.66684	x_2	2	
9	−5.0546	4.1503	1.3447	x_2	2	Target passed

Total   24

Computation of target   3

Table III

Step	Continuation point			Continuation variable	Total correct steps	Comments
	x_1	x_2	x_3			
0	15 000	−2.0000	0.00000	x_3	—	
1	14.710	−1.9421	0.065381	x_1	3	
2	14.285	−1.7291	0.26874	x_3	4	
3	16.906	−1.2094	0 54685	x_2	2	
4	24.918	−0.59905	0.55514	x_1	5	Weak acceptance
5	48.976	0.71811	−0.080806	x_1	5	
6	57.289	1.2847	−0 40742	x_1	5	Step reductions
7	60.053	1.5711	−0 54043	x_1	5	Step reductions
8	61.666	2.0014	−0.66689	x_2	2	
9	−4.4290	4.1408	1.3217	x_2	2	Target passed

Total   33

Computation of target   8

has here the form

$$Fx = \begin{pmatrix} x_1 - x_2^3 + 5x_2^2 - 2x_2 + 34x_3 - 47 \\ x_1 + x_2^3 + x_2^2 - 14x_2 + 10x_3 - 39 \end{pmatrix}, \quad \forall x \in R^3. \tag{6.1}$$

For the starting point $x^0 = (15, -2, 0)^T$ the solution curve passes through $x^* = (5, 4, 1)^T$, and this point is chosen as target.

Tables II and III show runs with the full Newton method and modified Newton method, respectively, as corrector process. A starting step $h_0 = 0.3$ and maximum step $h_{max} = 25.0$ were used. The performance for the two correctors is practically the same, although the step prediction exhibits certain differences owing to our assessment of the corrector distance. Clearly, the use of the modified Newton process is much less expensive and hence preferable, as shown in Table IV, which summarizes the total number of function and Jacobian calls, including those for the target calculation. Comparative performance data given in [8] for this problem involved 22 continuation steps, 15 step reductions, and 128 Jacobian evaluations.

Table IV

	Corrector process	
	Newton	Modified Newton
Function calls	39	54
Jacobian calls	36	21

The procedure discussed in [21] required 25 continuation steps, but no further details were provided in the paper.

It may be noted that the solution curve has two limit points each with respect to x_1 and x_3. The two step reductions are almost unavoidable here, since the curve has a long, straight segment followed by a very sharp bend. The target computation is relatively expensive, since the last step is extremely large due to another straight curve segment.

Example 2.   Maneuvering airplanes, especially at high angles of attack, sometimes undergo sudden jumps in their response to the pilot's control inputs. The problem has been discussed extensively in the literature (see, for example, [19], [32], [41]). Without going into further details we use here a simplified version of a system of five equilibrium equations involving the roll rate (x_1), pitch rate (x_2), yaw rate (x_3), (incremental) angle of attack (x_4), side slip angle (x_5), elevator angle (x_6), aileron angle (x_7), and rudder angle (x_8). More specifically, for the particular aircraft discussed in [19], these equations have the dimensionless form

$$Fx \equiv Ax + \phi(x) = 0, \quad \forall x \in R^8, \qquad (6.2)$$

where

$$A = \begin{pmatrix} -3.933 & 0.107 & 0.126 & 0 & -9.99 & 0 & -45.83 & -7.64 \\ 0 & -0.987 & 0 & -22.95 & 0 & -28.37 & 0 & 0 \\ 0.002 & 0 & -0.235 & 0 & 5.67 & 0 & -0.921 & -6.51 \\ 0 & 1.0 & 0 & -1.0 & 0 & -0.168 & 0 & 0 \\ 0 & 0 & -1.0 & 0 & -0.196 & 0 & -0.0071 & 0 \end{pmatrix}$$

and

$$\phi(x) = \begin{pmatrix} -0.727x_2x_3 + 8.39x_3x_4 - 684.4x_4x_5 + 63.5x_4x_7 \\ 0.949x_1x_3 + 0.173x_1x_5 \\ -0.716x_1x_2 - 1.578x_1x_4 + 1.132x_4x_7 \\ -x_1x_5 \\ x_1x_4 \end{pmatrix}.$$

Figure 3 shows some solution curves on the three-dimensional equilibrium surface in R^8. More specifically, in all cases we fixed a value of x_6 (elevator deflection) and chose the rudder deflection $x_8 = 0$. The paths $x_6 > \omega_1$, $x_8 = 0$ with $\omega_1 \approx -0.0061771$ have two limit points; for $\omega_1 > x_6 > \omega_2$, $x_8 = 0$, with $\omega_2 \approx -0.012498$ a third limit point appears; and for $\omega_2 > x_6$, $x_8 = 0$ only one limit point remains. A similar picture arises for negative roll rates.
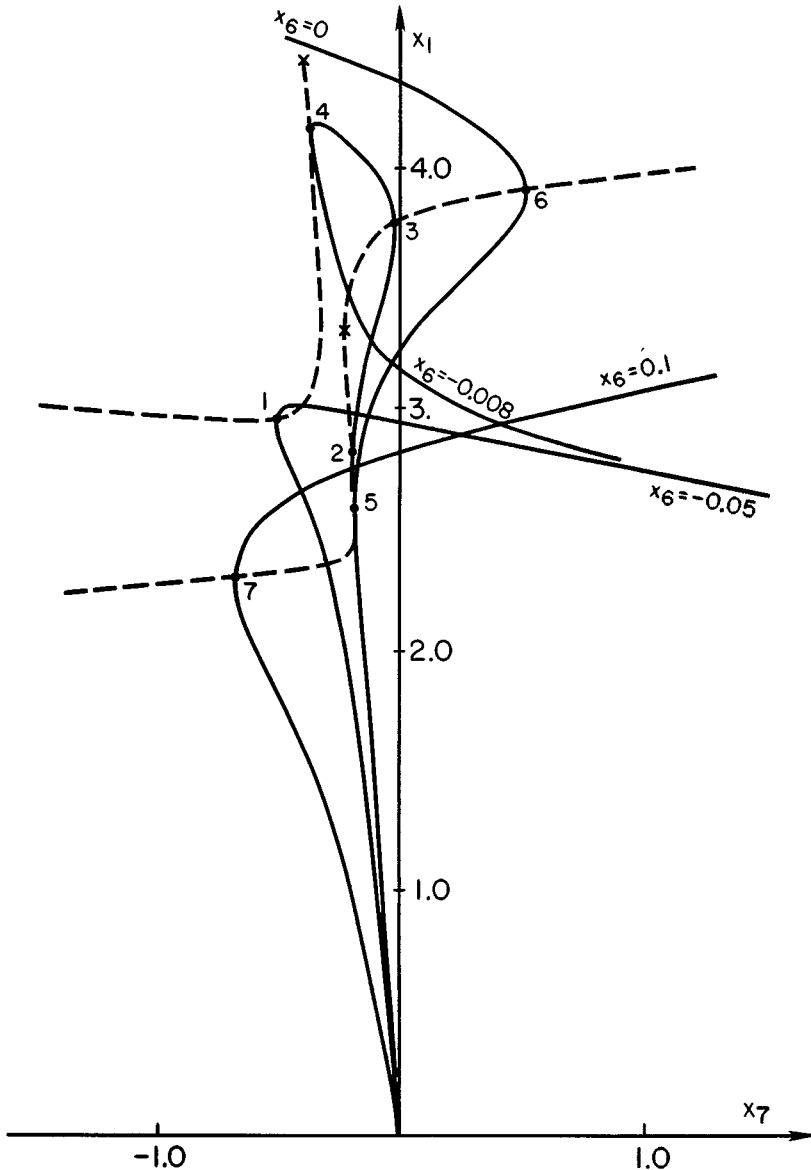
Figure 3

In all cases the programs easily detected and computed the various limit points (see Table V). But the example also shows that even with a large number of search paths it is difficult to provide a full picture of the location of the critical boundary, that is, of the curves of limit points with respect to x_1 for $x_8 = 0$ and varying x_6, x_7. In Figure 3 the corresponding branches of limit-point curves are shown as dotted lines. They were obtained with a code for the earlier mentioned task (iv) in Section 1 (see [28]).

Table V

	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	2.9649	0.8255	0.073661	0.0413	0.26735	−0.05	0.50481
2	2.8174	−0.17629	0.089926	0.026429	−0.071476	−0.008	−0.20497
3	3.7579	−0.65541	0.38658	0.092520	−0.19867	−0.008	0.006201
4	4.1638	0 089133	0.094805	0.022888	0.016232	−0.008	−0.37766
5	2.5873	−0.22355	0 054683	0.013676	−0.091687	0.0	−0 18691
6	3.9005	−1.1482	0.58156	0.13352	−0.32859	0.0	0.51016
7	2.2992	−1.4102	−0 061849	−0.079009	−0.58630	0.1	−0.68972
8	4.4565	−4.4909	1.6164	0.33091	−1 0857	0.1	10.0212

Example 3.   As an example for the numerical investigation of the equilibrium surface of a mechanical structure, we consider a clamped, thin, shallow, circular arch which has been used as a test case by various authors (see, e.g., [13], [17], [38]). Let U and W be the radial and axial displacements, R the arch radius, A the cross-sectional area, H the thickness, and E Young's modulus. With the dimensional displacements $u = U/H$, $w = W/H$, the total potential energy—nondimensionalized by dividing by $EAR(H/R)^2$—is given by

$$\int_{-\theta_0}^{\theta_0} \left\{ \left[\left(\frac{dw}{d\theta} - u \right) + \frac{1}{2} \frac{H}{R} \left(\frac{du}{d\theta} \right)^2 \right]^2 + \alpha_1 \left(\frac{d^2 u}{d\theta^2} \right)^2 - \alpha_2 p u \right\} d\theta. \qquad (6.3)$$

Here $p = p(\theta)$ is the dimensionless radial load, and α_1, α_2 are dimensionless constants. Each end is assumed to be pinned; that is, we have the boundary conditions

$$u(\pm \theta_0) = 0, \qquad w(\pm \theta_0) = 0, \qquad \frac{d^2 u}{d\theta^2} (\pm \theta_0) = 0. \qquad (6.4)$$

The finite-element approximation introduced in [38] was applied. More specifically, we used a uniform mesh with eight elements, $\theta_0 = 15$ and the constants $\alpha_1 = 3.8716 \times 10^{-6}$, $\alpha_2 = 1.65504 \times 10^{-1}$ corresponding to the data in [17]. Moreover, the following load function $p = p(\mu, \nu)$ was chosen

$$p(\mu, \nu) = \begin{cases} \mu(1 + 7\nu) & \text{for element 4,} \\ \mu(1 - \nu) & \text{otherwise,} \end{cases} \qquad (6.5)$$

corresponding to a base load $\beta = \mu(1 - \nu)$ and an excess load $8\mu\nu$ in element 4, such that the average load is always μ.

Several curves on the equilibrium surface corresponding to constant values of μ or ν were computed. Figure 4 shows the projection of these curves into the (β, δ)-plane, where δ represents the radial displacement of the center point. For uniform loads, that is, $\nu = 0$, we encounter two bifurcation points on the primary curve which are connected by two "buckling" curves that have the same projection in the (β, δ)-plane.
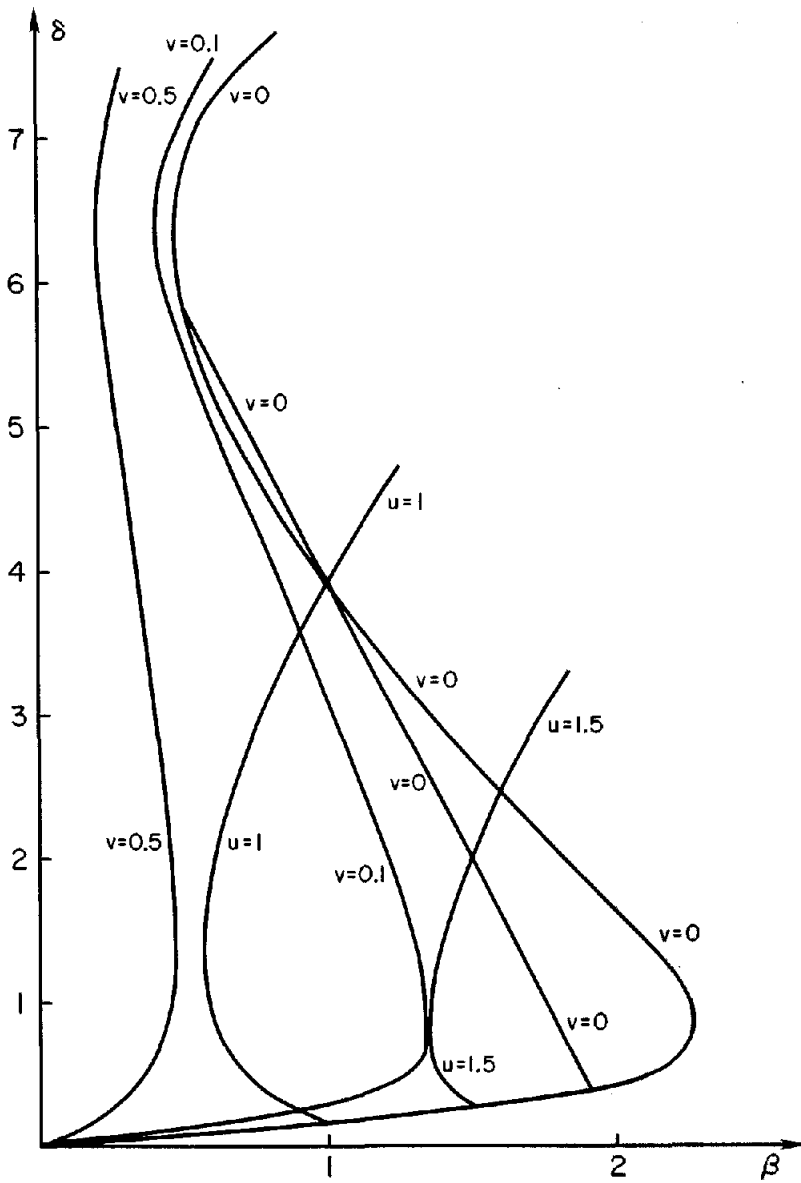
Figure 4

REFERENCES

1. ABBOTT, J.P.   An efficient algorithm for the determination of certain bifurcation points. *J. Comput. Appl. Math. 4* (1978), 19–27.
2. ALLGOWER, E. AND GEORG, K.   Simplicial and continuation methods for approximating fixed points and solutions to systems of equations. *SIAM Rev. 22* (1980), 28–85.
3. BABUSKA, I., AND RHEINBOLDT, W.C.   Computational error estimates and adaptive processes for some nonlinear structural problems. *J. Comput. Math. Appl. Mech. Eng. 34* (1982), 895–937.

4. BOGGS, P.T.   The solution of nonlinear systems of equations by A-stable integration techniques. *SIAM J. Numer. Anal. 8* (1971), 767–785.

5. BRENT, R.P.   *Algorithms for Minimization Without Derivatives.* Prentice-Hall, Englewood Cliffs, N.J., 1973

6 CRISFIELD, M.A.   A fast incremental/iterative solution procedure that handles snap-through *Comput Struct. 13* (1981), 55–62.

7 DAVIDENKO, D F.   On a new method of numerical solution of systems of nonlinear equations. *Dokl. Akad. Nauk SSSR 88* (1953), 601–602.

8. DEN HEIJER, C., AND RHEINBOLDT, W.C.   On steplength algorithms for a class of continuation methods. *SIAM J Numer. Anal 18* (1981), 925–947.

9 FICKEN, F.   The continuation method for functional equations. *Commun Pure Appl. Math. 4* (1951), 435–456.

10. FREUDENSTEIN, F., AND ROTH, B   Numerical solution of systems of nonlinear equations. *J. ACM 10,* 4(Oct 1963), 550–556.

11. KELLER, H.B   Numerical solution of bifurcation and nonlinear eigenvalue problems. In *Applications of Bifurcation Theory,* P. Rabinowitz (Ed.), Academic Press, New York, 1977, pp. 359–384

12 KELLER, H B   Global homotopies and Newton methods. In *Recent Advances in Numerical Analysis,* C. deBoor and G. H. Golub (Ed.), Academic Press, New York, 1978, pp. 73–94.

13. KERR, A D., AND SOIFER, M T   The linearization of the prebuckling state and its effect on the determined instability load. *Trans ASME, J. Appl. Mech. 36* (1969), 775–783.

14 KIZNER, W.   A numerical method for finding solutions of nonlinear equations. *SIAM J. Appl. Math 12* (1964), 424–428.

15 KUBICEK, M   Dependence of solution of nonlinear systems on a parameter, *ACM Trans. Math. Softw. 2,* 1 (March 1976), 98–107.

16. KUBICEK, M , HOLODNIOK, M., AND MAREK, I.   Numerical solution of nonlinear equations by one-parameter imbedding methods. *Numer. Funct. Anal. Optim. 3* (1981), 223–264.

17. MAU, S.T., AND GALLAGHER, R.H.   A finite element procedure for nonlinear prebuckling and initial postbuckling analysis. NASA Contractor Rep., NASA-CR 1936, January 1972.

18 MILNOR, J W.   *Topology from the Differential Viewpoint.* University of Virginia Press, Charlottesville, 1965

19. MEHRA, R K., KESSEL, W.C , AND CARROLL, J.V.   Global stability and control analysis of aircraft at high angles of attack. ONR Report-CR-215-248 1, 2, 3, June 1977–1979.

20. MELHEM, R., AND RHEINBOLDT, W C   A comparison of methods for determining turning points of nonlinear equations. *Computing 29* (1982), 201–226.

21 MENZEL, R., AND SCHWETLICK, H.   Zur Losung parameter-abhangiger nichtlinearer Gleichungen mit singularen Jacobi-Matrizen. *Numer. Math. 30* (1978), 65–79.

22 MOORE, G., AND SPENCE, A.   The calculation of turning points of nonlinear equations. *SIAM J. Numer Anal 17* (1980), 567–576.

23 ODEN, J.T.   *Finite Elements of Nonlinear Continua.* McGraw-Hill, New York, 1972.

24 PONISCH, G., AND SCHWETLICK, H.   Computing turning points of curves implicitly defined by nonlinear equations depending on a parameter. *Computing 26* (1981), 107–121.

25. POSTON, T., AND STEWART, I   *Catastrophe Theory and Its Applications.* Pitman, London, 1978.

26. RHEINBOLDT, W.C.   An adaptive continuation process for solving systems of nonlinear equations. Banach Ctr. Publ., Vol. 3, Polish Academy of Science, Warsaw, 1977, 129–142.

27 RHEINBOLDT, W.C.   Numerical methods for a class of finite dimensional bifurcation problems. *SIAM J. Numer Anal 15* (1978), 1–11

28. RHEINBOLDT, W C   Solution fields of nonlinear equations and continuation methods. *SIAM J. Numer Anal. 17* (1980) 221–237.

29. RHEINBOLDT, W.C.   Numerical analysis of continuation methods for nonlinear structural problems. *Comput. Struct. 13* (1981), 103–114

30. RHEINBOLDT, W.C   Computation of critical boundaries on equilibrium manifolds. *SIAM J. Numer. Anal 19* (1982), 653–669

31. RIKS, E   An incremental approach to the solution of snapping and buckling problems. *Int. J. Solids Struct.* 15 (1979), 524–551.

32. SCHY, A A , AND HANNAH, M.E.   Prediction of jump phenomena in roll-coupled maneuvers of airplanes. *J Aircraft 14* (1977), 375–382.

33 SEWELL, M J.   On the connection between stability and the shape of the equilibrium surface. *J. Mech. Phys Solids 14* (1966), 203–230

34 SEWELL, M.J.   Some global equilibrium surfaces. *Int. J. Mech Eng Educ. 6* (1978), 163–174

35 SEYDEL, R.   Numerical computation of branch points in nonlinear equations. *Numer. Math. 33* (1979), 339–352.

36 SIMPSON, R.B   A method for the numerical determination of bifurcation states of nonlinear systems of equations. *SIAM J Numer Anal. 12* (1975), 439–451.

37. WACKER, H.J. (Ed.)   *Continuation Methods.* Academic Press, New York, 1978.

38 WALKER, A.C   A non-linear finite element analysis of shallow circular arches. *Int. J. Solids Struct 5* (1969), 97–107.

39. WATSON, L.T.   A globally convergent algorithm for computing fixed points of C^2-maps. *Appl Math. Comput. 5* (1979), 297–311

40 WATSON, L.T., AND FENNER, D.   Chow–Yorke algorithm for fixed points or zeros of C^2 maps. *ACM Trans. Math. Softw. 6,* 2 (June 1980), 252–259

41 YOUNG, J W., SCHY, A.A., AND JOHNSON, K.G.   Prediction of jump phenomena in aircraft maneuvers, including nonlinear aerodynamic effects *J Guidance Contr. 1* (1978), 26–31.