NUMERICAL ANALYSIS AND MATHEMATICAL MODELLING
BANACH CENTER PUBLICATIONS, VOLUME 24
PWN-POLISH SCIENTIFIC PUBLISHERS
WARSAW 1990

A TWO-STAGE NEWTON-LIKE METHOD FOR COMPUTING SIMPLE BIFURCATION POINTS OF NONLINEAR EQUATIONS DEPENDING ON TWO PARAMETERS

GERD PÖNISCH

Section of Mathematics, Technical University of Dresden, Dresden, German Democratic Republic

An efficient method is described for the numerical determination of a simple bifurcation point of a nonlinear operator equation depending on two parameters. Two augmented systems containing the original equation are discussed, for which the simple bifurcation point is an isolated solution. An efficient implementation of Newton's method is presented.

1. Introduction

In this paper we consider a finite dimensional system

$$G(x, t, s) = 0, \quad G\colon \boldsymbol{R}^n \times \boldsymbol{R}^1 \times \boldsymbol{R}^1 \to \boldsymbol{R}^n, \tag{1.1}$$

of nonlinear, parameter dependent equations where G is twice continuously differentiable on an open subset D_G of $\boldsymbol{R}^n \times \boldsymbol{R}^1 \times \boldsymbol{R}^1$. The aim is to compute simple bifurcation points of G with regard to the parameter t.

Let us assume that the point $(x^*, t^*, s^*) \in D_G$ satisfies the following assumptions

$$G(x^*, t^*, s^*) = 0, \tag{A1}$$

$$\dim N\big(G_x(x^*, t^*, s^*)\big) = 1, \tag{A2}$$

$$\dim N\big([G_x(x^*, t^*, s^*) \vdots G_t(x^*, t^*, s^*)]\big) = 2, \tag{A3}$$

$$\dim N\big(G'(x^*, t^*, s^*)\big) = 2, \tag{A4}$$

where N denotes the nullspace of a linear operator. $G_x(x, t, s)$ and $G_t(x, t, s)$ symbolize the partial Fréchet derivatives of G with respect to $x \in \boldsymbol{R}^n$ and $t \in \boldsymbol{R}^1$,

respectively. The Fréchet derivative $G'(x, t, s)$ at (x, t, s) is represented by

$$G'(x, t, s) = [A(x, t, s) \vdots G_s(x, t, s)],$$

where

$$A(x, t, s) = [G_x(x, t, s) \vdots G_t(x, t, s)]. \tag{1.2}$$

Both (A2) and (A3) imply that there exist $v_1^* \in \boldsymbol{R}^{n+1}$ and $v_2^* \in \boldsymbol{R}^{n+1}$ such that

$$N(A(x^*, t^*, s^*)) = \operatorname{span}\{v_1^*, v_2^*\}, \tag{1.3}$$

$$(e^{n+1})^T v_1^* = 0, \quad (e^{n+1})^T v_2^* = 1,$$

where e^i denotes the i-th coordinate vector $(0, \ldots, 0, 1, 0, \ldots, 0)^T \in \boldsymbol{R}^{n+1}$. Moreover, there is a nontrivial vector $\psi^* \in \boldsymbol{R}^n$ with

$$N(G_x(x^*, t^*, s^*)^T) = N(A(x^*, t^*, s^*)^T) = \operatorname{span}\{\psi^*\}. \tag{1.4}$$

In addition to (A1)–(A4) let us assume that

$$\alpha_{11}\alpha_{22} - (\alpha_{12})^2 < 0, \tag{A5}$$

where

$$\alpha_{ij} = (\psi^*)^T \left[\begin{array}{c|c} G_{xx}(x^*, t^*, s^*) & G_{xt}(x^*, t^*, s^*) \\ \hline G_{tx}(x^*, t^*, s^*) & G_{tt}(x^*, t^*, s^*) \end{array}\right] v_i^* v_j^*$$

for $i = 1, 2$ and $j = 1, 2$.

Then (x^*, t^*, s^*) is called a *simple bifurcation point* of (1.1) with regard to t, i.e., for fixed s^* there is a neighbourhood of (x^*, t^*, s^*) in D_G such that the solution set of (1.1) exactly consists of two smooth curves intersecting at (x^*, t^*, s^*), see [5].

In the case of one parameter some numerical approaches to the computation of simple bifurcation points have been developed. We refer to [3] and [6] for a survey.

For the characterization of a simple bifurcation point by a nonsingular system of nonlinear equations in n state variables we have to consider, at least, $n+2$ conditions. Consequently, from (1.1) we can get such a system by adding two suitable equations. Then we have a system of $n+2$ equations in n state variables and two control parameters. In this way we do not need an imperfection parameter like in [3].

The next section presents two characterizations of a simple bifurcation point by means of an auxiliary system of dimension $n+2$. The general principle for the construction of adapted methods is described in Section 3. Section 4 contains an efficient implementation of one numerical method.

2. Characterization of a simple bifurcation point

In order to give numerical characterizations of a simple bifurcation point we have to consider $n+2$ conditions. As (A1) can used directly the assumptions (A2)–(A4) have to be reformulated. For this reason we define the square matrix

$$B(x, t, s, r) := \begin{bmatrix} G_x(x, t, s) & \vdots & G_s(x, t, s) \\ - - - - - - - & - & - - - - \\ & r^T & \end{bmatrix} \tag{2.1}$$

having the dimension $n+1$. For it the following lemma holds.

LEMMA 2.1. *Let* $(x^*, t^*, s^*) \in D_G$ *be a simple bifurcation point of* (1.1). *Then, for any* $r \in \boldsymbol{R}^{n+1}$ *with*

$$r^T v_1^* \neq 0 \tag{2.2}$$

a neighbourhood U *of* (x^*, t^*, s^*) *exists such that* $B(x, t, s, r)$ *are nonsingular for all* $(x, t, s) \in U$.

The proof of Lemma 2.1 is easy done using Lemma 7.2 of [3] e.g. Note that (A3) and (A4) imply

$$(\psi^*)^T G_s(x^*, t^*, s^*) \neq 0. \tag{2.3}$$

Considering (1.3), the assumption (A2) implies

$$v_1^* = v^* B(x^*, t^*, s^*, r)^{-1} e^{n+1} \tag{2.4}$$

with $v^* = r^T v_1^*$ and any $r \in \boldsymbol{R}^{n+1}$ according to (2.2). Hence, one additional equation is suggested in the form

$$g(x, t, s, r) := (e^{n+1})^T B(x, t, s, r)^{-1} e^{n+1} = 0. \tag{2.5}$$

By (1.3) we obtain $g(x^*, t^*, s^*, r) = 0$.

Both (A2) and (A3) imply that

$$G_t(x^*, t^*, s^*) \in R\big(G_x(x^*, t^*, s^*)\big), \tag{2.6}$$

where

$$R\big(G_x(x^*, t^*, s^*)\big) := \{y \in \boldsymbol{R}^n : (\psi^*)^T y = 0\}$$

denotes the range of $G_x(x^*, t^*, s^*)$. Consequently

$$(\psi^*)^T G_t(x^*, t^*, s^*) = 0. \tag{2.7}$$

Using (1.4) and (2.3), the vector $\psi^* \in \boldsymbol{R}^n$ can be expressed by

$$G_x(x^*, t^*, s^*)^T \psi^* = 0,$$

$$G_s(x^*, t^*, s^*)^T \psi^* = \varkappa^* \neq 0.$$

Hence we can write

$$\begin{pmatrix} \psi^* \\ 0 \end{pmatrix} = \varkappa^* [B(x^*, t^*, s^*, r)^T]^{-1} e^{n+1}, \tag{2.8}$$

where we have to consider that $Pr \notin R(G_x(x^*, t^*, s^*)^T)$ for any $r \in \boldsymbol{R}^{n+1}$ according to (2.2) and for $P = [I \,\vdots\, 0]$. I denotes the unit matrix of dimension n.

Then, the condition (2.7) yields the second additionally equation

$$f(x, t, s, r) := (e^{n+1})^T B(x, t, s, r)^{-1} \begin{bmatrix} G_t(x, t, s) \\ - - - - - \\ 0 \end{bmatrix} = 0. \tag{2.9}$$

Finally, a characterization of a simple bifurcation point is given by

$$F(z) = 0, \quad F\colon \boldsymbol{R}^{n+2} \to \boldsymbol{R}^{n+2}, \tag{2.10}$$

defined by

$$F(z) := \begin{bmatrix} G(x, t, s) \\ g(x, t, s, r) \\ f(x, t, s, r) \end{bmatrix}, \quad z := \begin{bmatrix} x \\ s \\ t \end{bmatrix}, \tag{2.11}$$

where $r \in \boldsymbol{R}^{n+1}$ is chosen fixed according to (2.2).

Another characterization for the numerical determination of a simple bifurcation point is pointed out in [3]. Following this idea, we have to substitute only the second additional equation (2.9). By (1.3) there exists an integer $m \in \{1, 2, \ldots, n\}$ such that

$$(e^m)^T v_1^* \neq 0.$$

Without loss of generality we can assume that

$$(e^m)^T v_2^* = 0; \tag{2.12}$$

see [3]. Analogous to (2.2), let $d \in \boldsymbol{R}^{n+1}$ be a given vector with

$$d^T v_2^* \neq 0. \tag{2.13}$$

Further, we form the $(n+1)$-dimensional matrix

$$\hat{B}(x, t, s, d) := \begin{bmatrix} A(x, t, s)\big(I - e^m (e^m)^T\big) + G_s(x, t, s)(e^m)^T \\ - - - - - - - - - - - - - - - - - - - \\ d^T \end{bmatrix}$$

from $A(x, t, s)$ substituting the m-th column by $G_s(x, t, s)$ and bordering d^T. Lemma 2.1 of [3] shows that $B(x, t, s, d)$ is nonsingular in a neighbourhood of a simple bifurcation point (x^*, t^*, s^*). Particularly, considering (1.3) we obtain

$$v_2^* = \mu^* \hat{B}(x^*, t^*, s^*, d)^{-1} e^{n+1} \tag{2.14}$$

with $\mu^* = d^T v_2^*$ and any $d \in \boldsymbol{R}^{n+1}$ according to (2.13). According to (2.12), we can take

$$\hat{f}(x, t, s, d) := (e^m)^T \hat{B}(x, t, s, d)^{-1} e^{n+1} = 0 \tag{2.15}$$

instead of (2.9). If we choose

$$r := e^m, \quad d := e^{n+1} \tag{2.16}$$

the functions f and $\hat{f}$ are equivalent, see Janovsky [1].

3. A two-stage method

For solving systems like (2.10) we point out an efficient implementation of Newton's method starting from $(x^0, t_0, s_0) \in \boldsymbol{R}^n \times \boldsymbol{R}^1 \times \boldsymbol{R}^1$ where (x^0, t_0, s_0) is a point near the simple bifurcation point (x^*, t^*, s^*). In the following we describe an iterative step of a two-stage method, which is also used for solving similar systems defining a turning point and a hysteresis point, respectively; see [4], [2].

For simplicity of notation we define the vector

$$u = \begin{bmatrix} x \\ s \end{bmatrix} \in \boldsymbol{R}^{n+1}$$

and write the system (2.10) in the form

$$G(u, t) = 0, \quad g(u, t, r) = 0, \quad f(u, t, r) = 0. \tag{3.1}$$

Applying one step of Newton's method to (3.1), from $(u, t) \in \boldsymbol{R}^{n+1} \times \boldsymbol{R}^1$ we obtain (u^+, t^+) as a solution of the linear system

$$G(u, t) + G_u(u, t)(u^+ - u) + G_t(u, t)(t^+ - t) = 0, \tag{3.2a}$$

$$g(u, t, r) + g_u(u, t, r)(u^+ - u) + g_t(u, t, r)(t^+ - t) = 0, \tag{3.2b}$$

$$f(u, t, r) + f_u(u, t, r)(u^+ - u) + f_t(u, t, r)(t^+ - t) = 0. \tag{3.2c}$$

At first, we compute the general solution of (3.2a). Due to the regularity of $B(x, t, s, r)$ the underdetermined system (3.2a) has the general solution

$$u^+ - u = \gamma v + \delta w + y, \quad t^+ - t = \delta, \quad \gamma, \delta \in \boldsymbol{R}^1, \tag{3.3}$$

with

$$v := B(x, t, s, r)^{-1} e^{n+1},$$

$$w := -B(x, t, s, r)^{-1} \begin{bmatrix} G_t(x, t, s) \\ - - - - \\ 0 \end{bmatrix}, \quad y := -B(x, t, s, r)^{-1} \begin{bmatrix} G(x, t, s) \\ - - - - - \\ 0 \end{bmatrix}.$$

In the second stage we determine the coefficients γ and δ. Instead of it, we compose a suitable system of two linear equations from (3.2b) and (3.2c). For

this reason we insert (3.3) into (3.2b) and (3.2c) and obtain

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{pmatrix} \gamma \\ \delta \end{pmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, \tag{3.4}$$

where

$$\begin{aligned} a_{11} &:= g_u(u, t, r)v, \\ a_{12} &:= g_u(u, t, r)w + g_t(u, t, r), \\ a_{21} &:= f_u(u, t, r)v, \\ a_{22} &:= f_u(u, t, r)w + f_t(u, t, r), \\ b_1 &:= -g(u, t, r) - g_u(u, t, r)y = -(e^{n+1})^T v - g_u(u, t, r)y, \\ b_2 &:= -f(u, t, r) - f_u(u, t, r)y = (e^{n+1})^T w - f_u(u, t, r)y. \end{aligned}$$

The partial derivatives of g and f are given by

$$g_u(u, t, r)q = -(e^{n+1})^T B(x, t, s, r)^{-1} \begin{bmatrix} G_{uu}(u, t)q \\ - - - - - \\ 0^T \end{bmatrix} B(x, t, s, r)^{-1} e^{n+1}$$

$$= -(e^{n+1})^T B(x, t, s, r)^{-1} \begin{bmatrix} G_{uu}(u, t)qv \\ - - - - - \\ 0 \end{bmatrix},$$

$$f_u(x, t, r)q = -(e^{n+1})^T B(x, t, s, r)^{-1} \begin{bmatrix} G_{uu}(u, t)q \\ - - - - - \\ 0^T \end{bmatrix} B(x, t, s, r)^{-1} \begin{bmatrix} G_t(u, t) \\ - - - - \\ 0 \end{bmatrix}$$

$$+ (e^{n+1})^T B(x, t, s, r)^{-1} \begin{bmatrix} G_{tu}(u, t)q \\ - - - - - \\ 0 \end{bmatrix}$$

$$= (e^{n+1})^T B(x, t, s, r)^{-1} \begin{bmatrix} G_{uu}(u, t)qw + G_{tu}(u, t)q \\ - - - - - - - - - - \\ 0 \end{bmatrix}$$

with $q \in \boldsymbol{R}^{n+1}$ and

$$g_t(u, t, r) = -(e^{n+1})^T B(x, t, s, r)^{-1} \begin{bmatrix} G_{ut}(u, t)v \\ - - - - - \\ 0 \end{bmatrix},$$

$$f_t(u, t, r) = (e^{n+1})^T B(x, t, s, r)^{-1} \begin{bmatrix} G_{ut}(u, t)w + G_{tt}(u, t) \\ - - - - - - - - - \\ 0 \end{bmatrix}.$$

Analyzing a_{12} and a_{21}, we find $a_{12} = -a_{21}$. Note further that for $(u^*, t^*) = ((x^{*T}, s^*)^T, t^*)$ the coefficient a_{ij} is proportional to α_{ij} in (A5). Consequently, the system (3.4) is nonsingular. Because of $a_{12} \neq 0$ for (u, t) near to (u^*, t^*) we can set

$$\delta := (b_1 + a_{11} b_2/a_{12})/(a_{12} + a_{11} a_{22}/a_{12}), \qquad \gamma := (\delta a_{22} - b_2)/a_{12}.$$

Thus, the kernel of our two-stage method is formulated in the following algorithm.

ALGORITHM 3.1.

1. Preelimination.

1.1. Compute a LU-decomposition of $B(x, t, s, r)$.

1.2. Use the LU-factors from 1.1 for computing v, w, $y \in \mathbf{R}^{n+1}$ from

$$B(x, t, s, r)v = e^{n+1},$$

$$B(x, t, s, r)w = \begin{bmatrix} -G_t(x, t, s) \\ - - - - - \\ 0 \end{bmatrix}, \qquad B(x, t, s, r)y = \begin{bmatrix} -G(x, t, s) \\ - - - - - \\ 0 \end{bmatrix}.$$

2. Determination of the coefficients γ and δ.

2.1. Use the LU-factors from 1.1 for computing only the last component of the vectors h_{11}, h_{12}, $h_{22} \in \mathbf{R}^{n+1}$ and k_1, $k_2 \in \mathbf{R}^{n+1}$ from

$$B(x, t, s, r)h_{11} = \begin{bmatrix} G_{uu}(u, t)v\,v \\ - - - - - \\ 0 \end{bmatrix},$$

$$B(x, t, s, r)h_{12} = \begin{bmatrix} G_{uu}(u, t)v\,w + G_{ut}(u, t)v \\ - - - - - - - - - - \\ 0 \end{bmatrix},$$

$$B(x, t, s, r)h_{22} = \begin{bmatrix} G_{uu}(u, t)w\,w + 2G_{ut}(u, t)w + G_{tt}(u, t) \\ - - - - - - - - - - - - - - - - - \\ 0 \end{bmatrix},$$

$$B(x, t, s, r)k_1 = \begin{bmatrix} G_{uu}(u, t)v\,y \\ - - - - - \\ 0 \end{bmatrix},$$

$$B(x, t, s, r)k_2 = \begin{bmatrix} G_{uu}(u, t)w\,y + G_{ut}(u, t)y \\ - - - - - - - - - - \\ 0 \end{bmatrix}.$$

2.2. Set

$$a_{11} := -(e^{n+1})^T h_{11}, \quad a_{12} := -(e^{n+1})^T h_{12}, \quad a_{22} := (e^{n+1})^T h_{22},$$
$$b_1 := -(e^{n+1})^T v + (e^{n+1})^T k_1, \quad b_2 := (e^{n+1})^T w - (e^{n+1})^T k_2.$$

2.3. Set

$$\delta := (b_1 + a_{11} b_2/a_{12})/(a_{12} + a_{11} a_{22}/a_{12}),$$
$$\gamma := (\delta a_{22} - b_2)/a_{12}.$$

3. Set

$$u^+ = \begin{pmatrix} x^+ \\ s^+ \end{pmatrix} := u + \gamma v + \delta w + y, \quad t^+ := t + \delta.$$

This algorithm requires only one LU-decomposition of the $(n+1, n+1)$-matrix $B(x, t, s, r)$, three "full" and five "half" backsubstitutions. The main computational work has to be done in step 2.1. If the partial derivatives G_{uu}, G_{ut} and G_{tt} are explicitly used the building up of the corresponding terms in step 2.1 requires $2n+8$ matrix-vector multiplications in general. In the next section we suggest efficient approximations for the quoted terms reducing the computational costs drastically.

Remark 3.2. In order to compute the values of a_{11}, a_{12}, a_{22}, b_1 and b_2 it is also possible to replace the steps 2.1 and 2.2 by:

2.1′. Use the LU-factors from 1.1 for computing $\psi \in \boldsymbol{R}^{n+1}$ from

$$B(x, t, s, r)^T \psi = e^{n+1}.$$

2.2′. Compute

$$a_{11} := -\psi^T \begin{bmatrix} G_{uu}(u, t) v\, v \\ - - - - - \\ 0 \end{bmatrix},$$

$$a_{12} := -\psi^T \begin{bmatrix} G_{uu}(u, t) v\, w + G_{ut}(u, t) v \\ - - - - - - - - - - - \\ 0 \end{bmatrix},$$

$$a_{22} := \psi^T \begin{bmatrix} G_{uu}(u, t) w\, w + 2 G_{ut}(u, t) w + G_{tt}(u, t) \\ - - - - - - - - - - - - - - - - - - - \\ 0 \end{bmatrix},$$

$$b_1 := -(e^{n+1})^T v + \psi^T \begin{bmatrix} G_{uu}(u, t) v\, y \\ - - - - - \\ 0 \end{bmatrix},$$

$$b_2 := (e^{n+1})^T w - \psi^T \begin{bmatrix} G_{uu}(u, t) w\, y + G_{ut}(u, t) v \\ - - - - - - - - - - \\ 0 \end{bmatrix}.$$

Substituting the steps 2.1 and 2.2 by 2.1′ and 2.2′ the Algorithm 3.1 requires still one LU-decomposition, four "full" backsubstitutions and five dot products now.

For analyzing the convergence behaviour we imbed Algorithm 3.1 into an iterative process:

ALGORITHM 3.3.

0. Choose $(x^0, t_0, s_0) \in \boldsymbol{R}^n \times \boldsymbol{R}^1 \times \boldsymbol{R}^1$, set $k := 0$.
1. Perform Algorithm 3.1 with $x = x^k$, $t = t_k$, $s = s_k$.
2. Set $x^{k+1} := x^+$, $t_{k+1} := t^+$, $s_{k+1} := s^+$ and go to 1.

The convergence behaviour of the generated sequence $\{z^k\}$ is described by the following theorem, where

$$z := \begin{bmatrix} x \\ s \\ t \end{bmatrix} \in \boldsymbol{R}^{n+2}.$$

THEOREM 3.4. *Let* G: $D_G \subset \boldsymbol{R}^n \times \boldsymbol{R}^1 \times \boldsymbol{R}^1 \to \boldsymbol{R}^n$ *be twice Lipschitz-continuous differentiable and let* $z^* \in D_G$ *be a simple bifurcation point of* (1.1).

Then, there is an $\varepsilon > 0$ *such that for all starting points* $z^0 \in \boldsymbol{R}^{n+2}$ *with* $\|z^0 - z^*\| \leqslant \varepsilon$ *and all vectors* $r \in \boldsymbol{R}^{n+1}$ *satisfying* (2.2) *Algorithm* 3.3 *is well defined and the sequence* $\{z^k\}$ *converges Q-quadratically to the simple bifurcation point* z^*.

Since Algorithm 3.3 is an implementation of Newton's method for the system (2.10), the well-known assertions about Newton's method imply Theorem 3.4. Note that z^* is an isolated solution of (2.10); cf. Lemma 2.2 in [3] and Theorem 2.2 in [1].

If we use $\hat{f}$ according to (2.15) instead of f from (2.9) the suggested way for solving the modified system (2.10) is similar. The corresponding Newton-like method is essentially outlined in [3] in the special case

$$G(x, t, s) = H(x, t) + sq,$$

where $H := \boldsymbol{R}^n \times \boldsymbol{R}^1 \to \boldsymbol{R}^n$ and $q \in \boldsymbol{R}^n$ according to $q^T \psi^* \neq 0$.

The description of this procedure is some more expensive. For this reason we refer to [3]. In the quoted implementation at one step it requires the solution of nine linear systems where for two weakly different coefficient matrices $B(x, t, s, r)$ and $\hat{B}(x, t, s, d)$ the LU-decompositions have to be done. Theorem 3.4 holds in this case analogously.

4. An efficient implementation

The need of only one LU-decomposition per iteration step is an advantage of the proposed two-stage method described in Algorithm 3.3 and Algorithm 3.1. Furthermore the procedure given in [3] requires a weakly expensive data handling. For these reasons we consider Algorithm 3.1 in the following.

For reducing the computational work of Algorithm 3.1 we want to

substitute the right-hand sides of the linear systems in step 2.1 by direction approximations. Using the notations

$$z = \begin{pmatrix} x \\ s \\ t \end{pmatrix} = \begin{pmatrix} u \\ t \end{pmatrix}, \quad \underline{v} = \begin{pmatrix} v \\ 0 \end{pmatrix} \in \mathbf{R}^{n+2}, \quad \underline{w} = \begin{pmatrix} w \\ 1 \end{pmatrix} \in \mathbf{R}^{n+2}, \quad \underline{y} = \begin{pmatrix} y \\ 0 \end{pmatrix} \in \mathbf{R}^{n+2},$$

$$G(z) = G(u, t) = G(x, t, s), \tag{4.1}$$

$$G_z(z) = [G_u(u, t) \vdots G_t(u, t)] = [G_x(x, t, s) \vdots G_s(x, t, s) \vdots G_t(x, t, s)]$$

we obtain

$$\begin{aligned} G_{uu}(u, t)vv &= G_{zz}(z)\underline{v}\,\underline{v}, \\ G_{uu}(u, t)vw + G_{ut}(u, t)v &= G_{zz}(z)\underline{v}\,\underline{w}, \\ G_{uu}(u, t)ww + 2G_{ut}(u, t)w + G_{tt}(u, t) &= G_{zz}(z)\underline{w}\,\underline{w}, \\ G_{uu}(u, t)vy &= G_{zz}(z)\underline{v}\,\underline{y}, \\ G_{uu}(u, t)wy + G_{ut}(u, t)y &= G_{zz}(z)\underline{w}\,\underline{y}. \end{aligned} \tag{4.2}$$

These terms can be considered as a limit of a second order divided difference. Let, without loss of generality, be $p, q \in \mathbf{R}^{n+2}$ with $\|p\| = \|q\| = 1$ and approximate

$$G_{zz}(z)pq \tag{4.3}$$

by

$$l(z, p, q, \mu) = [G(z+\mu p) - G(z+\mu p-\mu q) + G(z-\mu q) - G(z)]/\mu^2 \tag{4.4}$$

where $\mu \in \mathbf{R}^1 \setminus \{0\}$ is the discretization stepsize. From [3] we take over Lemma 3.1:

LEMMA 4.1. *Let G: $\mathbf{R}^{n+2} \to \mathbf{R}^n$ be twice Lipschitz-continuous differentiable in a neighbourhood of z^* with a constant $L_2 > 0$. Then, there is a $\xi^* > 0$ and a $\mu^* > 0$ such that for any $z \in S(z^*, \xi^*) = \{z \in \mathbf{R}^{n+2}: \|z - z^*\| \leqslant \xi^*\}$ and any $\mu \in \mathbf{R}^1$ with $0 < |\mu| \leqslant \mu^*$ the direction approximation (4.4) is well defined any fulfils*

$$\|l(z, p, q, \mu) - G_{zz}(z)pq\| \leqslant L_2|\mu|. \tag{4.5}$$

Thus we replace step 2.1 of Algorithm 3.1 by

2.1″. Use the LU-factors from 1.1 for computing only the last component of the vectors h_{11}, h_{12}, $h_{22} \in \mathbf{R}^{n+1}$ and k_1, $k_2 \in \mathbf{R}^{n+1}$ from

$$B(x, t, s, r)h_{11} = \|v\|^2 \begin{bmatrix} l(z, \underline{v}/\|v\|, \underline{v}/\|v\|, \mu) \\ - - - - - - - \quad - - \\ 0 \end{bmatrix},$$

$$B(x, t, s, r)h_{12} = \|v\|\,\|\underline{w}\| \begin{bmatrix} l(z, \underline{v}/\|v\|, \underline{w}/\|\underline{w}\|, \mu) \\ - - - - - - - - - - \\ 0 \end{bmatrix},$$

$$B(x, t, s, r)h_{22} = \|\underline{w}\|^2 \begin{bmatrix} l(z, \underline{w}/\|\underline{w}\|, \underline{w}/\|\underline{w}\|, \mu) \\ - - - - - - - - - - \\ 0 \end{bmatrix},$$

$$B(x, t, s, r)k_1 = \|v\| \, \|y\| \begin{bmatrix} l(z, \underline{v}/\|v\|, \underline{y}/\|y\|; \mu) \\ - - - - - - - - - - \\ 0 \end{bmatrix},$$

$$B(x, t, s, r)k_2 = \|\underline{w}\| \, \|y\| \begin{bmatrix} l(z, \underline{w}/\|\underline{w}\|, \underline{y}/\|y\|, \mu) \\ - - - - - - - - - - \\ 0 \end{bmatrix},$$

where $\mu \in \boldsymbol{R}^1 \setminus \{0\}$.

By keeping the costs for solving the linear systems we economize the $2n+8$ matrix-vector multiplications and the explicit use of the second partial derivatives of G. We only need 8 additional function values of G, namely $G(u-\mu v, t)$, $G(u+\mu v, t)$, $G(u-\mu w, t-\mu)$, $G(u+\mu v-\mu w, t-\mu)$, $G(u+\mu w, t+\mu)$, $G(u+\mu v-\mu y, t)$, $G(u-\mu y, t)$, $G(u+\mu w-\mu y, t+\mu)$.

Of course, there are other modifications of step 2.1 in order to reduce the computational costs. One consists in the approximation of the bilinear form (4.3) by means of additional first partial derivatives. In this way we need the two matrices $G_u(u+\mu v, t)$, $G_u(u+\mu w, t+s)$ and four vectors $G_t(u+\mu v, s)$, $G_t(u+\mu w, t+\mu)$, $G_t(u, t+\mu)$, $G_t(u+\mu y, t)$. Furthermore the building up of the right-hand sides of step 2.1 requires 6 matrix-vector multiplications, additionally. Therefore the given modification in step 2.1″ seems to be very efficient.

Consequently, we suggest for computing of a simple bifurcation point the following algorithm.

ALGORITHM 4.2.

0. Choose $(x^0, t_0, s_0) \in \boldsymbol{R}^n \times \boldsymbol{R}^1 \times \boldsymbol{R}^1$, $\bar{\mu} > 0 \in \boldsymbol{R}^1$, set $k := 0$.
1. Choose μ_k according to $0 < |\mu_k| \leqslant \bar{\mu}$.
2. Perform Algorithm 3.1 with the modified step 2.1″ using

$$x := x^k, \quad t := t_k, \quad s := s_k, \quad \mu := \mu_k.$$

3. Set $x^{k+1} := x^+$, $t_{k+1} := t^+$, $s_{k+1} := s^+$ and go to 1.

THEOREM 4.3. *Let* $G\colon D_G \subset \boldsymbol{R}^n \times \boldsymbol{R}^1 \times \boldsymbol{R}^1 \to \boldsymbol{R}^n$ *be twice Lipschitz-continuous differentiable and let* $z^* \in D_G$ *be a simple bifurcation point of* (1.1). *Moreover, let* $r \in \boldsymbol{R}^{n+1}$ *be such that* (2.2) *holds. Then, there exist* $\varepsilon^* > 0$ *and* $\mu^* > 0$ *such that for all starting values* $z^0 \in S(z^*, \varepsilon^*)$ *and all* $\bar{\mu} \in (0, \mu^*]$ *Algorithm* 4.2 *is well defined and the sequence* $\{z^k\}$ *satisfies*

$$\|z^{k+1} - z^*\| \leqslant c(\|z^k - z^*\| + |\mu_k|) \, \|z^k - z^*\| \tag{4.6}$$

with some $c > 0$. *If, additionally,*

$$\lim_{k \to \infty} \mu_k = 0 \tag{4.7}$$

the sequence $\{z^k\}$ *converges Q-superlinearly to the simple bifurcation point* z^*. *Finally, if* $\{\mu_k\}$ *is chosen according to*

$$|\mu_k| \leqslant \varkappa \|G(z^k)\|, \quad \varkappa > 0,$$

then $\{z^k\}$ *converges at least Q-quadratically.*

Theorem 4.3 can be proved like Theorem 5.3 in [3].

Remark 4.4.

(i) Note that $\{z^k\}$ generated by Algorithm 4.2 converges at least Q-quadratically because all components of z^k are problem variables. If we have to use an imperfection parameter like in [3] under similar assumptions the corresponding subsequence $\{\binom{x^l}{t_l}\}$ is convergent with R-order $\geqslant 2$.

(ii) If we use a stepsize bound

$$|\mu_k| \leqslant \varkappa \|z^k - z^{k-1}\|, \quad \varkappa > 0,$$

the sequence $\{z^k\}$ converges Q-superlinearly with R-order $(1+\sqrt{5})/2$.

(iii) It is also possible to work with variable bordering directions $r = r^k$ such that

$$|(v_1^*)^T r^k| \geqslant \varrho, \quad \varrho > 0.$$

Theorem 4.3 holds in this case, too. One implementation consists in the choice of

$$r^k := v^k = B(x^k, t_k, s_k, r^{k-1})^{-1} e^{n+1}.$$

Another version is $r^k := e^j$ where j is defined by

$$|(e^j)^T v^k| = \max\{|(e^i)^T v^k| : i = 1, 2, \ldots, n+1\}.$$

References

[1] V. Janovský, *A note on computing simple bifurcation points*, Computing, submitted.

[2] G. Pönisch, *Computing hysteresis points of nonlinear equations depending on two parameters*, Computing 39 (1987), 1–17.

[3] –, *Computing simple bifurcation points using a minimally extended system of nonlinear equations*, Computing 35 (1985), 277–294.

[4] G. Pönisch, H. Schwetlick, *Computing turning points of curves implicitly defined by nonlinear equations depending on a parameter*, Computing 26 (1981), 107–121.

[5] H. Weber, *On the numerical approximation of secondary bifurcation problems*, in E. L. Allgower, K. Glashoff, H. O. Peitgen (eds.), Numerical Solution of Nonlinear Equations 407–425, Lecture Notes in Math. 878, Springer, Heidelberg–Berlin–New York 1981.

[6] H. D. Mittelmann, H. Weber, *Numerical methods for bifurcation problems — a survey and classification*, in H. D. Mittelmann, H. Weber (eds.), *Bifurcation Problems and Their Numerical Solution* 1–45, ISNM 54, Basel–Birkhäuser 1980.

Presented to the Semester
Numerical Analysis and Mathematical Modelling
February 25 — May 29, 1987