Mit freundlichen Gruessen des Autors

Efficient Computation of Singular Points*

P. Kunkel

Abstract

For a class of augmented systems for the computation of singular points, a Gauß-Newton like iteration is developed which only needs a represention of the Moore-Penrose pseudoinverse of a matrix which only has the size of the Jacobian of the original system. A technique based on deflated block elimination is presented to determine this pseudoinverse even in the case of large systems. To illustrate the applicability and efficiency of the methods, several numerical examples are included.

* to appear in IMA J. Numer. Anal.

1 Introduction

In order to compute singular solutions of systems of nonlinear equations

$$f(y) = 0\,, \qquad f\colon \mathbb{R}^{n+1} \to \mathbb{R}^n \quad \text{smooth}, \tag{1.1}$$

a possible way is to extend (1.1) to a so-called augmented system for which the desired singular point then is a regular solution (see e.g. [1,10,15]). The main disadvantage of this procedure, however, is that by increasing the size N of the system, the necessary amount of computational effort is $\mathcal{O}(N^3)$ if one applies standard algorithms.

In [8], on which the present paper is directly based and of which it requires some knowledge, a class of augmented systems for the computation of singular points was given for which the Gauß-Newton method converges quadratically. But in each iteration step, one has to compute a representation of the Moore-Penrose pseudoinverse of a $(N - k^2, N)$-matrix where $N = 2(k+1)(n+k+1)$ and

$$k = n - \operatorname{rank} L\,, \tag{1.2}$$

for $L = f_y(y^*)$, y^* a perfect singular point, with k small (say $k \in \{1,2,3\}$). This procedure is practicable only for small values of n. For larger n, more sophisticated methods must be applied to reduce the computational effort.

This paper wants to present a modification of the Gauß-Newton method based on an appropriate perturbation of the Jacobian in each iteration step utilizing the special structure of the equations which shall be solved, see section 2. It will be shown there that on one hand this leads to a slightly worse convergence behaviour compared with the Gauß-Newton method itself but that on the other hand the essential computational part only consists of a representation of the Moore-Penrose pseudoinverse of an $(n, n+1)$-matrix L with rank $n-k$ which is an approximation to $f_y(y)$ at the current iteration point y. The next point then is how one can get an efficient representation of L^+. Often, especially in the case of large n, there is a natural splitting of $y = (x, \tau)$ into state variables $x \in \mathbb{R}^n$ and embedding parameter $\tau \in \mathbb{R}$, where now $A = f_x(x, \tau)$ has special properties like symmetry, bandedness, sparseness. In this case, one would like to apply special techniques having already developed for the treatment of a regular A. This is discussed in section 3. Some numerical examples then conclude the present paper.

2 A Gauß-Newton like method

In [8], the following class of augmented systems was proposed for the computation of unfolded singular points

$$\begin{aligned}
\operatorname{diag}(f_y(y))\operatorname{col}(t_i) - \operatorname{diag}(z)\operatorname{col}(\beta_i) &= 0\,, \\
\operatorname{diag}(f_y^T(y))\operatorname{col}(z_j) - \operatorname{diag}(t)\operatorname{col}(\gamma_j) &= 0\,, \\
N_t(\operatorname{col}(t_i)) &= 0\,, \qquad i = 1,\dots,k+1\,, \\
N_z(\operatorname{col}(z_j)) &= 0\,, \qquad j = 1,\dots,k\,, \\
f(y) - z\alpha &= 0\,, \\
b(y, \operatorname{col}(t_i), \operatorname{col}(z_j)) &= 0\,,
\end{aligned} \tag{2.1}$$

where the following short hand notation was used.

$$\begin{aligned} \operatorname{col}(t_i)^T &= (t_1^T, \dots, t_{k+1}^T)\,, & \operatorname{col}(z_j)^T &= (z_1^T, \dots, z_k^T)\,, \\ \operatorname{col}(\beta_i)^T &= (\beta_1^T, \dots, \beta_{k+1}^T)\,, & \operatorname{col}(\gamma_j)^T &= (\gamma_1^T, \dots, \gamma_k^T)\,, \\ t &= (t_1, \dots, t_{k+1})\,, & z &= (z_1, \dots, z_k) \end{aligned}$$

The terms $\operatorname{diag}(*)$ stand for block diagonal matrices with entries $*$. Their size is determined by the vectors on which they act. Further, the functions N_t and N_z gather the orthonormalization conditions for the kernel vectors t_i and corange vectors z_j, respectively, and b is a $\mathbb{R}^{k+1}$-valued function depending on the kind of singular point one wants to compute.

In order to solve the underdetermined system (2.1), which we can write as

$$F(u) = 0\,, \qquad u^T = (y^T, \operatorname{col}(t_i)^T, \operatorname{col}(z_j)^T, \alpha^T, \operatorname{col}(\beta_i)^T, \operatorname{col}(\gamma_j)^T)\,, \tag{2.1$'$}$$

we can apply the Gauß-Newton method

$$\begin{aligned} \Delta u_l &= -F_u(u_l)^+ F(u_l)\,, \\ u_{l+1} &= u_l + \Delta u_l\,, \end{aligned} \qquad l = 0, 1, \dots\,, \qquad u_0 \text{ given.} \tag{2.2}$$

Then, the main result in [8] concerning the convergence behaviour of (2.2) is given in the following theorem.

Theorem 1

Let y^* be a perfect solution of (2.1), i.e.

$$\alpha = 0\,, \qquad \operatorname{col}(\beta_i) = 0\,, \qquad \operatorname{col}(\gamma_j) = 0\,. \tag{2.3}$$

For the Jacobian J at the solution y^*, we have

$$J = \begin{pmatrix} T & \operatorname{diag}(L) & 0 & 0 & -\operatorname{diag}(z) & 0 \\ Z & 0 & \operatorname{diag}(L^T) & 0 & 0 & -\operatorname{diag}(t) \\ 0 & M_t & 0 & 0 & 0 & 0 \\ 0 & 0 & M_z & 0 & 0 & 0 \\ L & 0 & 0 & -z & 0 & 0 \\ s^T & p^T & q^T & 0 & 0 & 0 \end{pmatrix}, \tag{2.4}$$

where the ordering of equations and unknowns is according to (2.1) and (2.1$'$) so that the definition of the entries is evident. Then, under

$$p^T \operatorname{diag}(t) = 0\,, \qquad q^T \operatorname{diag}(z) = 0\,, \tag{2.5}$$

which is called separation condition in the following, full rank of J is equivalent to the nonsingularity of the $(k+1, k+1)$-matrix

$$B = s^T t - p^T \operatorname{diag}(L^+) T t - q^T \operatorname{diag}(L^{+T}) Z t\,. \tag{2.6}$$

∎

Because of the full rank of J, the iteration (2.2) is known to converge locally and quadratically to a solution of (2.1).

Since the system (2.1) is underdetermined, it does not possess a unique solution. But for the following, we need a detailed knowledge of its solution manifold. Because the equations (2.1) do not distinguish between different orthonormal bases of kernel L and corange L, one may conjecture that the kernel of J represents changes between such bases. To investigate this, we must look at the special structure of the matrices

$$M_t = \left(\begin{array}{cccc} 2t_1^T & & & \\ t_2^T & t_1^T & & \\ \vdots & & \ddots & \\ t_{k+1}^T & & & t_1^T \\ \hline & 2t_2^T & & \\ & \vdots & \ddots & \\ & t_{k+1}^T & & t_2^T \\ \hline & & \cdots & \cdots \\ \hline & & & 2t_{k+1}^T \end{array}\right), \tag{2.7}$$

M_z similarly. In connection with them, consider the matrices

$$K_t = \left(\begin{array}{cccc|ccc|c|c} t_2 & t_3 & \cdots & t_{k+1} & & & & & \\ -t_1 & & & & t_3 & \cdots & t_{k+1} & & \\ & -t_1 & & & -t_2 & & & \vdots & \\ & & \ddots & & & \ddots & & \vdots & t_{k+1} \\ & & & -t_1 & & & -t_2 & \vdots & -t_k \end{array}\right), \tag{2.8}$$

K_z similarly. One immediately realizes $M_t K_t = 0$ and $M_z K_z = 0$. Observing the separation condition (2.5), this carries over to the whole matrix J. Our conjecture then follows from the fact that J has rank deficiency k^2 which is equal to the number of columns in the matrices K_t and K_z. Moreover, we see that the location of the singular point is not affected by the kernel vectors such that this part of the solution vector is unique.

Unfortunately, we do not have the special structure of J away from a perfect solution. But near a perfect or slightly unfolded singular point, the actual Jacobian is only a slight perturbation of a matrix with the same structure as the Jacobian at a perfect solution, i.e., of a matrix with an entry L which has rank deficiency k, with entries p and q satisfying the separation condition, and with vanishing entries corresponding to (2.3). The idea now is to compute a modified Gauß-Newton correction under the additional assumptions (1.2), (2.3), and (2.5) which yields a Gauß-Newton like procedure where we work with a perturbed Jacobian. Because these additional assumptions are in general not satisfied away from a perfect solution, the convergence rate reduces to superlinear

for computing a perfect singular point and to linear for computing an unfolded singular point with a rate factor proportional to the error introduced by the additional assumptions (cf. [12]). The loss in the convergence rate even in the case of slightly perturbed singular points then is not very essential because numerically superlinear convergence is hardly distinguishable from quadratic convergence and in the unfolded case only small imperfections are of interest since for large imperfections one rather prefers the separation of the underlying higher singularity into simpler singularities. Thus, the rate factor of linear convergence normally is small in applications and linear convergence can be observed only if one requires high accuracy. Together with the decrease in the necessary computational effort per iteration step, the perturbed method may therefore be expected to exhibit a drastic speed-up compared with (2.2) especially for larger values of n (cp. section 4).

The question now is, how to compute a correction

$$\Delta u = J^+ r \tag{2.9}$$

with

$$\Delta u^T = (\Delta y^T, \Delta t^T, \Delta z^T, \Delta\alpha^T, \Delta\beta^T, \Delta\gamma^T)\,, \qquad r^T = (r_t^T, r_z^T, n_t^T, n_z^T, r_f^T, r_b^T)$$

in an efficient way if J has the form (2.4) with the additional assumption that it exactly has the properties of a Jacobian at a perfect solution. To do this, we first need an explicit form of the Moore-Penrose pseudoinverses of M_t and M_z.

Lemma 2

The Moore-Penrose pseudoinverses of M_t and M_z are given by

$$M_t^+ = \frac{1}{2}\left(\begin{array}{cccc|ccc|c|c} t_1 & t_2 & \cdots & t_{k+1} & & & & & \\ & t_1 & & & t_2 & \cdots & t_{k+1} & & \\ & & \ddots & & & \ddots & & \vdots & \\ & & & t_1 & & & t_2 & \vdots & t_{k+1} \end{array}\right)\,, \tag{2.10}$$

M_z^+ similarly.

Proof

Verification of the four axioms, which here simplifies because of full rank of M_t and M_z yielding $M_t M_t^+ = I$ and $M_z M_z^+ = I$.

∎

Note that lemma 2 has at least two consequences. First, there is no need to compute M_t^+ and M_z^+ numerically since we know them explicitly. Second, we immediately recognize that $\mathrm{diag}\,(L) M_t^+ = 0$ and $\mathrm{diag}\,(L^T) M_z^+ = 0$.

Now we can state our main result.

Theorem 3

Let J be as in (2.4) and have full rank. Further, let L be an $(n, n+1)$-matrix with rank $L = n - k$. Finally, let the columns of t and z form orthonormal bases of kernel L and corange L satisfying (2.5). Then the least squares solution Δu of (2.9) is given by

$$
\begin{aligned}
\Delta y &= \overline{L} r_f + t B^{-1} r_b - \overline{p}^T r_t - \overline{q}^T r_z \,, \\
\Delta t &= M_t^+ n_t + \mathrm{diag}\,(L^+)(r_t - T\Delta y) \,, \\
\Delta z &= M_z^+ n_z + \mathrm{diag}\,(L^{+T})(r_z - Z\Delta y) \,, \\
\Delta \alpha &= -z^T r_f \,, \\
\Delta \beta &= -\mathrm{diag}\,(z^T)(r_t - T\Delta y) \,, \\
\Delta \gamma &= -\mathrm{diag}\,(t^T)(r_z - Z\Delta y) \,,
\end{aligned}
\tag{2.11}
$$

where

$$
\begin{aligned}
\overline{L} &= L^+ - t B^{-1} \overline{B} L^+ \,, \\
\overline{B} &= s^T - p^T \mathrm{diag}\,(L^+) T - q^T \mathrm{diag}\,(L^{+T}) Z \,, \\
\overline{p}^T &= t B^{-1} p^T \mathrm{diag}\,(L^+) \,, \\
\overline{q}^T &= t B^{-1} q^T \mathrm{diag}\,(L^{+T}) \,,
\end{aligned}
\tag{2.12}
$$

and $B = \overline{B} t$ as given in (2.6).

Proof

Before going into the details of proof, we state some relations between L, L^+ and t, z under the given assumptions, which are relevant in the following.

$$
\begin{aligned}
Lt = 0 \,, \qquad & t^T L^+ = 0 \,, \qquad & t^T t = I \,, \qquad & L^+ L + t t^T = I \\
z^T L = 0 \,, \qquad & L^+ z = 0 \,, \qquad & z^T z = I \,, \qquad & L L^+ + z z^T = I
\end{aligned}
$$

First we show that (2.11) actually forms a solution of (2.9).

$$
\begin{aligned}
& T\Delta y + \mathrm{diag}\,(L)\Delta t - \mathrm{diag}\,(z)\Delta\beta = \\
& \qquad = T\Delta y + \mathrm{diag}\,(LL^+ + zz^T)(r_t - T\Delta y) = r_t \\
& Z\Delta y + \mathrm{diag}\,(L^T)\Delta z - \mathrm{diag}\,(t)\Delta\gamma = \\
& \qquad = Z\Delta y + \mathrm{diag}\,(L^+L + tt^T)(r_z - Z\Delta y) = r_z \\
& M_t \Delta t = M_t M_t^+ n_t = n_t \\
& M_z \Delta z = M_z M_z^+ n_z = n_z \\
& L\Delta y - z\Delta\alpha = \\
& \qquad = L\overline{L} r_f + zz^T r_f = (LL^+ + zz^T) r_f = r_f \\
& s^T\Delta y + p^T\Delta t + q^T\Delta z = \\
& \qquad = s^T\Delta y + p^T \mathrm{diag}\,(L^+)(r_t - T\Delta y) + q^T \mathrm{diag}\,(L^{+T})(r_z - Z\Delta y) = \\
& \qquad = \overline{B}(\overline{L} r_f + t B^{-1} r_b - \overline{p}^T r_t - \overline{q}^T r_z) + \\
& \qquad\qquad + p^T \mathrm{diag}\,(L^+) r_t + q^T \mathrm{diag}\,(L^{+T}) r_z = \\
& \qquad = \overline{B} L^+ r_f - \overline{B} t B^{-1} \overline{B} L^+ r_f + r_b = r_b
\end{aligned}
$$

Second we show that the solution is orthogonal to kernel J.

$$K_t^T \Delta t = K_t^T M_t^+ n_t + K_t^T \mathrm{diag}\,(L^+)(r_t - T\Delta y) = 0$$
$$K_z^T \Delta z = K_z^T M_z^+ n_z + K_z^T \mathrm{diag}\,(L^{+T})(r_z - Z\Delta y) = 0$$

Together with full rank of J, the proof is finished.

∎

Thus, working with a perturbed Jacobian as described above, the essential part to compute a correction Δu is the calculation of a representation of L^+. For a more detailed discussion of this point, see section 3. All other computations only consist of backsubstitutions and scalar products. Moreover, we do not need to compute the total number of derivatives T, Z, s^T, p^T, and q^T, but only derivatives with respect to certain directions (cf. table 1).

TABLE 1

entry	necessary derivatives in direction of	number of derivatives
T	t, $L^+ r_f$, Δy	$k+3$
Z	t, $L^+ r_f$, Δy	$k+3$
s^T	t, $L^+ r_f$	$k+2$
p^T	$\mathrm{diag}\,(L^+)Tt$, $\mathrm{diag}\,(L^+)TL^+ r_f$, $\mathrm{diag}\,(L^+) r_t$	$k+3$
q^T	$\mathrm{diag}\,(L^{+T})Zt$, $\mathrm{diag}\,(L^{+T})ZL^+ r_f$, $\mathrm{diag}\,(L^{+T}) r_z$	$k+3$

3 Representations of the Moore-Penrose pseudoinverse

The main costs to carry out the procedure described in section 2 arise in computing a representation for L^+. Simultaneously, the computation of kernel and corange would be of interest, especially for gaining initial values for t and z. Note that in the present application the rank of L is given to be $n-k$ due to an appropriate perturbation of the Jacobian for the modified Gauß-Newton method of the previous section.

3.1 QR-Cholesky decomposition

In the case of a full matrix L with small n, a common method is QR-Cholesky decomposition (see [4]). By QR-decomposition of L with column pivoting under the condition (1.2), one obtains

$$(3.1)\quad L = Q \begin{pmatrix} R & S \\ 0 & 0 \end{pmatrix} \Pi\,,$$

Q orthogonal (n,n)-matrix,
R nonsingular $(n-k, n-k)$-upper triangular matrix,
S $(n-k, k+1)$-matrix,
Π $(n+1, n+1)$-permutation matrix.

Due to [11], the Moore-Penrose pseudoinverse of L then is given by

$$L^+ = \Pi^T \begin{pmatrix} (I - VM^{-1}V^T)R^{-1} & 0 \\ M^{-1}V^TR^{-1} & 0 \end{pmatrix} Q^T , \qquad \begin{aligned} V &= R^{-1}S , \\ M &= I + V^TV . \end{aligned} \tag{3.2}$$

Here one needs a representation of the inverse of the positive definite $(k+1, k+1)$-matrix M, which is obtained by Cholesky decomposition

$$M = XX^T , \qquad X \text{ nonsingular } (k+1, k+1)\text{-lower triangular matrix.} \tag{3.3}$$

Finally, orthonormal bases for kernel and corange can be computed by

$$t = \Pi^T \begin{pmatrix} -V \\ I \end{pmatrix} X^{-T} , \qquad z = Q \begin{pmatrix} 0 \\ I \end{pmatrix} . \tag{3.4}$$

But this method is known to disturb special structures of the matrix L, so that it is not applicable for large values of n.

3.2 Deflated block elimination

Especially for large values of n, one is interested in methods where special properties of L can be used particularly in the part $A = f_x(x, \tau)$. In most cases, one has a special adapted solver for the matrix A but under the assumption that A is regular. Here, however, we have rank deficiency according to

$$n - k - 1 \leq \operatorname{rank} A \leq n - k . \tag{3.5}$$

For this case, a method called deflated block elimination is developed in [3] which aims to use an arbitrary solver for the rank deficient matrix A but nevertheless to have a stable and efficient algorithm for solving a well-conditioned system with a matrix of the form

$$M = \begin{pmatrix} A & B \\ C^T & D \end{pmatrix} \tag{3.6}$$

where M is some small extension of A. For a regular M, its size has to be at least $2n - \operatorname{rank} A$. Because of (3.5), we therefore have to take an M with a size of at least $n + k + 1$. The basic method of the deflated block elimination then is to solve all equations with matrix A with the corresponding right hand sides projected onto a subset of range A by an algorithm adapted to the structure of A and to solve the remaining orthogonal part together with the stabilizing additional equations and unknowns by standard techniques. Note that in our case the first part is the dominating one consisting of essentially $n - k - 1$ equations while the second part treats the remaining $2(k+1)$ equations. For a detailed discussion of the deflated block elimination, we must refer to [3].

We now want to apply this method to get a representation of the Moore-Penrose pseudoinverse L^+. For this, we change the partitioning of the block matrix M and consider systems of the form

$$\begin{pmatrix} L & \tilde{z} \\ \tilde{t}^T & 0 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} = \begin{pmatrix} b \\ c \end{pmatrix} , \qquad \begin{array}{l} L \ (n, n+1)\text{-matrix, rank } L = n-k, \\ \tilde{t} \ (n+1, k+1)\text{-matrix}, \\ \tilde{z} \ (n, k)\text{-matrix}. \end{array} \tag{3.7}$$

Then the following regularity result holds (cp. [3]).

Theorem 4

Let L be an $(n, n+1)$-matrix with rank $L = n-k$, let the columns of the matrices t and z form orthonormal bases of kernel L and corange L, respectively. Then, under the transversality condition

$$\tilde{t}^T t \text{ and } \tilde{z}^T z \text{ regular}, \tag{3.8}$$

system (3.7) is regular and its solution is given by

$$\begin{aligned} v &= \big(I - t(\tilde{t}^T t)^{-1}\tilde{t}^T\big) L^+ \big(I - \tilde{z}(z^T\tilde{z})^{-1} z^T\big) b + t(\tilde{t}^T t)^{-1} c \, , \\ w &= (z^T\tilde{z})^{-1} z^T b \, . \end{aligned} \tag{3.9}$$

Proof

Let $Lv + \tilde{z}w = 0$, $\tilde{t}^T v = 0$. Multiplying the first equation with z^T and using (3.8), one immediately obtains $w = 0$. Then $v \in$ kernel L, thus we can write $v = t\alpha$. Substituting into the second equation and again using (3.8), one obtains $\alpha = 0$ and therefore $v = 0$. Hence, (3.7) is shown to be regular. Now putting (3.9) into (3.7), we get

$$\begin{aligned} Lv + \tilde{z}w &= LL^+\big(I - \tilde{z}(z^T\tilde{z})^{-1}z^T\big)b + \tilde{z}(z^T\tilde{z})^{-1}z^T b = \\ &= LL^+ b + (I - LL^+)\tilde{z}(z^T\tilde{z})^{-1}z^T b = \\ &= LL^+ b + zz^T\tilde{z}(z^T\tilde{z})^{-1}z^T b = b \, , \\ \tilde{t}^T v &= \big(\tilde{t}^T - \tilde{t}^T t(\tilde{t}^T t)^{-1}\tilde{t}^T\big) L^+ \big(I - \tilde{z}(z^T\tilde{z})^{-1}z^T\big) b + \tilde{t}^T t(\tilde{t}^T t)^{-1} c = c \, , \end{aligned}$$

which completes the proof. ∎

Note that a similar result is valid for matrices L of arbitrary size, hence for L^T.

Now we are able to compute bases $\overline{t}$ and $\overline{z}$ of kernel L and corange L because of

$$\begin{aligned} \begin{pmatrix} L & \tilde{z} \\ \tilde{t}^T & 0 \end{pmatrix} \begin{pmatrix} \overline{t} \\ 0 \end{pmatrix} &= \begin{pmatrix} 0 \\ I \end{pmatrix} , \qquad \overline{t} = t(\tilde{t}^T t)^{-1} \, , \\ \begin{pmatrix} L & \tilde{z} \\ \tilde{t}^T & 0 \end{pmatrix}^T \begin{pmatrix} \overline{z} \\ 0 \end{pmatrix} &= \begin{pmatrix} 0 \\ I \end{pmatrix} , \qquad \overline{z} = z(\tilde{z}^T z)^{-1} \, , \end{aligned} \tag{3.10}$$

which we can orthonormalize and use for example as initial values for the iterative process of section 2. But furthermore, substituting $\tilde{t}$ and $\tilde{z}$ by $\overline{t}$ and $\overline{z}$ yields the desired representation of L^+.

Corollary 5

The solution of

$$\begin{pmatrix} L & \overline{z} \\ \overline{t}^T & 0 \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix} , \qquad \overline{t} = t\alpha_t , \qquad \overline{z} = z\alpha_z , \tag{3.11}$$

with α_t and α_z regular matrices is given by

$$\begin{aligned} v &= L^+ b , \\ w &= \alpha_z^{-1} z^T b . \end{aligned} \tag{3.12}$$

Proof

Because $t^T\overline{t} = \alpha_t$ and $z^T\overline{z} = \alpha_z$ are regular, the transversality condition (3.8) is satisfied. Thus, the solution of (3.11) reads

$$\begin{aligned} v &= \big(I - t(\alpha_t^T t^T t)^{-1}\alpha_t^T t^T\big)L^+\big(I - z\alpha_z(z^T z\alpha_z)^{-1}z^T\big)b = L^+ b , \\ w &= (z^T z\alpha_z)^{-1} z^T b = \alpha_z^{-1} z^T b . \end{aligned}$$

∎

Summerizing, we have to do the following to compute L^+b. First, we have to choose matrices $\tilde{t}$ and $\tilde{z}$ satisfying the transversality condition (3.8). This is not essential due to Sard's Theorem which says that random choice of $\tilde{t}$ and $\tilde{z}$ satisfies conditions like (3.8) with probability one (see e.g. [6]). Then using the deflated block elimination procedure, we have to compute $\overline{t}$ and $\overline{z}$ by (3.10) and finally to solve (3.11).

4 Numerical examples

In order to illustrate the applicability and the efficiency of the methods presented in the previous sections, we report on some results including comparisons with related methods. All computations were performed on an IBM3090 with 56 bit mantissa. The common termination criterion was $\|\Delta u\|_\infty \leq 10^{-5}$ for the last correction Δu.

In our test, we used the following methods.

method 1 – Gauß-Newton method applied to (2.1), representation of L^+ by QR-decomposition as described in section 3

method 2 – Gauß-Newton like method as described in section 2, representation of L^+ by QR-decomposition as described in section 3

method 3 – as method 2, additionally using $p = 0$, $q = 0$ (for singular points which require additional equations b on first derivatives of f only, see [8])

method 4 – as method 3, additionally using sparse techniques, i.e. representation of L^+ by deflated block elimination as described in section 3 basing on the MA28 sparse solver from Harwell library [5]

method 5 – Newton's method applied to Moore's system for simple bifurcation points which reads

$$\begin{aligned} f_y^T(y)z &= 0\,, \\ z^T z &= 1\,, \\ f(y) - z\alpha &= 0\,, \end{aligned}$$

see [10], representation of J^{-1} by LU-decomposition

All higher order derivatives needed for J are approximated by finite differences. At this point, we should in addition mention that the techniques of section 2 also apply to Moore's system. The result is a method which coincides in the essential parts with method 3 for simple bifurcation points.

Example 1

The following equations describe a linear array of three diffusively coupled well-stirred cells each with two reacting chemical species, see e.g. [13] or [14].

$$\begin{aligned} R_1(x_1, x_2) + d_1(x_3 - x_1) &= 0 \\ R_2(x_1, x_2) + d_2(x_4 - x_2) &= 0 \\ R_1(x_3, x_4) + d_1(x_5 - 2x_3 + x_1) &= 0 \\ R_2(x_3, x_4) + d_2(x_6 - 2x_4 + x_2) &= 0 \\ R_1(x_5, x_6) + d_1(x_3 - x_5) &= 0 \\ R_2(x_5, x_6) + d_2(x_4 - x_6) &= 0 \end{aligned}$$

$$\begin{aligned} R_1(c_1, c_2) &= 2 - 7c_1 + c_1^2 c_2 \\ R_2(c_1, c_2) &= 6c_1 - c_1^2 c_2 \end{aligned}$$

$$\begin{aligned} d_1 &= \tau \\ d_2 &= 10\tau \end{aligned}$$

Starting from

$$x = (2.8, 2.3, 0.5, 3.0, 2.8, 2.3)^T, \qquad \tau = 0.2\,,$$

a simple bifurcation point was to be computed. The corresponding results are listed in table 2. For methods which are based on the decomposition of J itself, the percentage portion of CPU-time for the linear algebra part is given. Note that the rest part then is required for setting-up J by finite differences.

TABLE 2

method	number of iterations	CPU-time per iteration in msec	portion of linear algebra
1	4	24.5	55%
2	4	5.5	—
3	4	2.4	—
5	4	3.0	35%

For this example, there is almost no slow-down in the convergence rate using a perturbed Jacobian in the methods 2 and 3 compared with method 1. On the other side, we have a speed-up by a factor of five for the general method of section 2 and even by a factor of ten using the special structure of the additional equations b. Looking at the CPU-time, method 3 is comparable with method 5.

Example 2

Let us consider the partial differential equation

$$-\Delta u + 10u = \tau e^u$$

on the unit square with Neumann boundary conditions (see e.g. [9] or [7]). Standard discretization on a 10×10 equidistant grid leads to a system of 100 nonlinear equations $f(y) = 0$ in 101 unknowns $y = (x, \tau)$. This problem has a perfect simple bifurcation point near $y = (3.0, \ldots, 3.0, 1.5)^T$. Here the matrix $A = f_x(x, \tau)$ is large but sparse. Note that in general the Jacobian J is sparse too (apart from some almost full rows and columns), but properties of A like bandedness or symmetry do not carry over to J. See table 3 for the results obtained.

TABLE 3

method	number of iterations	CPU-time per iteration in msec	portion of linear algebra
1	3	17,392.2	63%
2	3	419.6	—
3	3	271.7	—
4	3	62.4	—
5	3	1,521.0	31%

Besides the profit from being able to use sparse techniques for A, we can recognize another advantage of the Gauß-Newton like procedure of the methods 3 and 4. For method 5, we have to compute 202 derivatives by finite differences to set up the corresponding Jacobian J. Observing the special structure of J, this can be reduced to the 101 derivatives with respect to y to get the second order terms, which would still cost about 500 msec or eight iterations of method 4 (cp. table 1).

Example 3

The problem given by

$$\begin{aligned} x_1^2 - x_2 + 1 &= 0 \\ (x_1 - 1)x_2 - 2(x_2 - x_1 - 1)\cosh\tau &= 0 \end{aligned}$$

exhibits a triple point bifurcation at $y^* = (1, 2, 0)^T$. Starting with $y = (0.8, 2.2, -0.3)^T$, the results were as given in table 4.

TABLE 4

method	number of iterations	CPU-time per iteration in msec	portion of linear algebra
1	5	5.9	56%
2	6	1.8	—

Here for the first time, we have an increase in the number of iterations turning to a perturbed Jacobian. On the other hand, we can perform three iterations of method 2 during one iteration of method 1. This ratio becomes more drastic if one treats problems with larger n.

Example 4

Finally, let us consider the following system of equations (see e.g. [2]).

$$\begin{aligned} x_1 + d(x_1 - x_2) - R(4 - x_1) &= 0 \\ d(2x_2 - x_1 - x_3) - R(4 - x_2) &= 0 \\ d(2x_3 - x_2 - x_4) - R(4 - x_3) &= 0 \\ 2x_4 + d(2x_4 - x_3 - x_5) - 2R(4 - x_4) &= 0 \\ d(2x_5 - x_4 - x_6) - R(4 - x_5) &= 0 \\ d(2x_6 - x_5 - x_7) - R(4 - x_6) &= 0 \\ x_7 + d(x_7 - x_6) - R(4 - x_7) &= 0 \end{aligned} \qquad \begin{aligned} R(c) &= \frac{10^{1.4}c}{1 + c + \tau c^2} \\ d &= \frac{3}{10} \end{aligned}$$

Near $y = (3.6, 4.0, 4.0, 3.6, 4.0, 4.0, 3.6, 9.3)^T$, this problem has an unfolded simple bifurcation point from a double eigenvalue. So we have to set $k = 2$. For more details, see [8]. The results of the different methods are given in table 5.

TABLE 5

method	number of iterations	CPU-time per iteration in msec	portion of linear algebra
1	5	102.2	56%
2	6	11.0	—
3	5	4.4	—

Again we have a drastic decrease of a factor of up to about twenty in the CPU-time necessary to perform one iteration while the loss in the convergence rate is nearly neglegible. The size of imperfection is about 10^{-4} leading to a difference to superlinear or quadratic convergence only for a more stringent termination criterion.

Summerizing, we want to state that we have an evident increase in efficiency applying the techniques presented in this paper. Moreover, by the concepts of the deflated block elimination, we can utilize the full structure of the problem as we can do in the regular case.

Acknowledgement

The author wants to thank G. Bader for many helpful discussions. This work has been supported by the *Deutsche Forschungsgemeinschaft*.

References

[1] E. ALLGOWER, K. BÖHMER,
Highly Singular Equations and Bifurcations,
Preprint (1986).

[2] W.-J. BEYN,
Defining Equations for Singular Solutions and Numerical Applications,
Inter.Ser.Num.Math. 70, 42-56 (1984).

[3] T.F. CHAN, D.C. RESASCO,
Generalized Deflated Block-Elimination,
SIAM J.Numer.Anal. 23, 913-924 (1986).

[4] P. DEUFLHARD, W. SAUTTER,
On Rank-Deficient Pseudo-Inverses,
Lin.Alg.Appl. 14, 257-272 (1976).

[5] I.S. DUFF,
MA28 — A Set of Fortran Subroutines for Sparse Unsymmetric Linear Equations,
Tech.Rep. AERE-R 8730, Harwell (1980).

[6] C.G. GIBSON,
Singular Points of Smooth Mappings,
Research Notes in Mathematics 25, Pitman (1979).

[7] H. Jarausch, W. Mackens,
Computing Bifurcation Diagrams for Large Nonlinear Variational Problems,
Preprint, University of Aachen (1986).

[8] P. Kunkel,
Quadratically Convergent Methods for the Computation of Unfolded Singularities,
to appear in SIAM J.Numer.Anal. (1988).

[9] H.D. Mittelmann,
Multi-level Continuation Techniques for Nonlinear Boundary Value Problems with Parameter Dependence,
Preprint, Arizona State University (1985).

[10] G. Moore,
The Application of Newton's Method to Simple Bifurcation and Turning Point Problems,
Ph.D.Thesis, University of Bath (1979).

[11] B. Noble,
A Method for Computing the Generalized Inverse of a Matrix,
SIAM J.Numer.Anal. 3, 582-584 (1966).

[12] J.M. Ortega, W.C. Rheinboldt,
Iterative Solution of Nonlinear Equations in Several Variables,
Academic Press, New York (1970).

[13] I. Prigogine, R. Lefever,
Symmetry Breaking Instabilities in Dissipative Systems II,
J.Chem.Phys. 48, 1695-1701 (1968).

[14] R. Seydel,
Numerical Computation of Branch Points in Nonlinear Equations,
Numer.Math. 33, 339-352 (1979).

[15] A. Spence, A.D. Jepson,
The Numerical Calculation of Cusps, Bifurcation Points and Isola Formation Points in Two Parameter Problems,
Inter.Ser.Num.Math. 70, 502-514 (1984).