Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational and Applied Mathematics 181 (2005) 252–265

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

A modified Brown algorithm for solving singular nonlinear systems with rank defects

Ren-Dong Ge[a,b,*], Zun-Quan Xia[a], Jian-guo Liu[a]

[a]*Department of Applied Mathematics, Dalian University of Technology, China*
[b]*Department of Applied Mathematics and Physics, Dalin Nationalities University, China*

Received 28 June 2003

Abstract

A modified Brown algorithm for solving a class of singular nonlinear systems, $F(x)=0$, where $x,\ F\in\mathbb{R}^n$, is presented. This method is constructed by combining the discreted Brown algorithm with the space transforming method. The second-order information of $F(x)$ at a point is not required calculating, which is different from the tensor method and the Hoy's method. The Q-quadratic convergence of this algorithm and some numerical examples are given as well.

MSC: 65H10; 65F10.

Keywords: System of nonlinear equation; Singular system of nonlinear equation; Brown algorithm; Jacobian matrix; Q-quadratic convergence

1. Introduction

Consider the following nonlinear system:

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

* Corresponding author. Department of Applied Mathematics and Physics, Dalin Nationalities University, China.
E-mail address: bgrbgg@163.com (R.-D. Ge).

0377-0427/$ - see front matter
doi:10.1016/j.cam.2004.12.001

where $x \in \mathbb{R}^n$, $F(x) = (f_1(x),\ f_2(x), \ldots,\ f_n(x))^{\mathrm{T}}$. We assume, throughout this paper, that there exists a solution x^* of (1.1), F' is Lipschitzian around x^* and

$$\begin{cases} \operatorname{rank}(F'(x^*)) = n - r, & 1 \leqslant r \ll n, \\ \nabla f_i(x^*) \neq 0 & i = 1, 2, \ldots, n. \end{cases} \tag{1.2}$$

There have been many publications dealing with nonlinear systems with rank defects of F' since the 1980s, see for instance, [1,2,5–7,9–12,14]. Some algorithms were designed for the case in which F' has rank $n-1$, see for instance, [4,9,11,12,14,16]. Hoy and Schwetlick [12] introduced an auxiliary function

$$T(x) = \begin{pmatrix} F(x) \\ \det(F'(x)) \end{pmatrix}$$

leading to construction of an algorithm for solving (1.1) with rank defect one, reading as follows.

Algorithm 1.1. *Step* 0: Choose $x_0 \approx x^*$, $q \approx v$, $p \approx u$, set $k = 0$.
Step 1: Set $B_k = F'(x_k) + pq^{\mathrm{T}}$.
Step 2: Determine d_k from $B_k d_k = F(x_k)$, v_k from $B_k^{\mathrm{T}} v_k = p$, u_k from $B_k u_k = q$.
Step 3: Set

$$x_{k+1} = x_k - d_k - \frac{1 - q^{\mathrm{T}} v_k - u_k^{\mathrm{T}} F''(x_k) v_k d_k}{u_k^{\mathrm{T}} F''(x_k) v_k v_k}.$$

End of Algorithm 1.1.

The sequence $\{x_k\}$ generated by this algorithm converges locally Q-quadratically to a solution of (1.1). Werber and Werner [18], and Yamamoto [19] already proposed some methods constructing extended equations before Hoy and Schwetlick, but their auxiliary functions are more complex and their corresponding algorithms are not easy to perform. Kanzawa and Oishi [13] proposed methods of interval iteration, enlightened by the method due to Werber and Werner [18], Yamamoto [19] for dealing with the auxiliary functions. The tensor model introduced by Schnabel and Frank [17] is a quadratic model of $F(x)$ formed by adding a second-order term to the linear model given by

$$M_T(x_c + d) = F(x_c) + J_c d + \tfrac{1}{2} T_c dd,$$

where $T_c \in \mathbb{R}^{N \times N \times N}$ is used to give the second-order information about $F(x_c)$ around x_c. $T_c = \arg\min\{\|\widehat{T}_c\|_F \mid \widehat{T}_c s_k s_k = z_k,\ k = 1, 2, \ldots, p\}$, where s_k, z_k are defined by Dan et al.[4]. The goal of the tensor model is to find $d \in \mathbb{R}^N$ such that d is a solution of $\min_{d \in \mathbb{R}^N} \|M_T(x_c + d)\|_2$. Under the assumptions that $F'(x^*)$ is singular with only one zero singular value and $u^{\mathrm{T}} F''(x^*) vv \neq 0$, the sequence of iterations generated by the tensor method based on an ideal tensor model converges locally and two-step Q-superlinearly to the solution with Q-order $\frac{3}{2}$, and the sequence of iterates generated by the tensor method based on a practical tensor model converges locally and three-step Q-superlinearly to the solution with Q-order $\frac{3}{2}$. Tensor method is considered as a very good algorithm in solving singular nonlinear equations since 1984. It is extended to solving unconstrained optimization (1991, 1997) and equality constrained optimization (1996). For the system with rank defects, Ge and Xia [7,8] constructed a modified ABS algorithm for solving problem (1.1) under the same conditions by combining the discreted ABS algorithm with the idea of Hoy and Schwetlick [12]. In this paper, a modified Brown method is proposed for solving systems of nonlinear equations with rank $(F'(x^*)) = n - r$, $1 < r \ll n$ and the second-order information of $F(x)$ is not required, see Algorithm 2.1 in Section 2. The new algorithm converges Q-quadratically to x^*. We now recall the discrete Brown method, due to Brent [3], for solving nonlinear systems of F with full rank.

Algorithm 1.2. *Step* 0: Choose $h_0,\ x_0$ close enough to x^*, set $k = 0,\ j = 1$.

Step 1: Let $y_1^{(k)} = x_k$. Take orthogonal matrices $Q_j^{(k)}$ for $j = 1, 2, \ldots, n$. Do steps 2–4.

Step 2: Compute

$$\overline{a}_j^{(k)} = \frac{1}{h_k}\begin{pmatrix} 0 \\ \vdots \\ 0 \\ f_j(y_j^{(k)} + h_k Q_j^{(k)} e_j) - f_j(y_j^{(k)}) \\ \vdots \\ f_j(y_j^{(k)} + h_k Q_j^{(k)} e_n) - f_j(y_j^{(k)}) \end{pmatrix}.$$

Step 3: Find an orthogonal matrix $P_j^{(k)}$ being of the form

$$P_j^{(k)} = \begin{pmatrix} I_{(j-1)\times(j-1)} & 0 \\ 0 & \widehat{P}_j^{(k)} \end{pmatrix}.$$

such that $P_j^{(k)}\overline{a}_j^{(k)} = s_j^{(k)} e_j$, where $s_j^{(k)} = \pm\|\overline{a}_j^{(k)}\|$. (For example, $\widehat{P}_j^{(k)}$ may be an elementary Hermition.)

Step 4: Compute $Q_{j+1}^{(k)} = Q_j^{(k)} P_j^{(k)}$ and

$$y_{j+1}^{(k)} = y_j^{(k)} - s_j^{(k)-1} f_j(y_j^{(k)}) Q_{j+1}^{(k)} e_j.$$

Step 5: Let $x_{k+1} = y_{n+1}^{(k)}$, $k \leftarrow k + 1$ and go to step 1.

End of Algorithm 1.2.

Note that Algorithm 1.2 is different from the (quasi-) Newton's methods solving systems of nonlinear equations directly. The process of its iteration contains a sub-iterate of an auxiliary variable y in which one equation in y is solved by approximating a projection of the gradient of f_j, each time for j, see steps 2–4.

In this paper, a new algorithm constructed by combining the discrete Brown algorithm with a space transformation method is presented in Section 2. The modified Brown Algorithm avoids the case, which $\overline{a}_j^{(k)} = 0$, by using the space transformation method and still ensures its Q-quadratic rate convergence, see Section 3. In the last section, some numerical experiments are given.

2. The modified Brown algorithm

In this section, we assume that Null $(F'(x^*)) = \text{span}\{u_1, u_2, \ldots, u_r\}$, Null $(F'(x^*)^{\mathrm{T}}) = \text{span}\,\{e_1, e_2, \ldots, e_r\}$, $N=(u_1, u_2, \ldots, u_r)$ and $E=(e_1, e_2, \ldots, e_r)$, where $\{u_1, u_2, \ldots, u_r\}$ and $\{e_1, e_2, \ldots, e_r\}$ both are norm orthogonal vector sets. Let P, Q be the $n \times r$ matrices with columns full rank and $P^{\mathrm{T}}N$, $Q^{\mathrm{T}}E$ are nonsingular. It is can be obtained easily that $F'(x^*) + QP^{\mathrm{T}}$ also is nonsingular and the solution of the matrix equations $(F'(x^*) + QP^{\mathrm{T}})X = Q$ is $N(P^{\mathrm{T}}N)^{-1}$. Moreover, let

$$N(P^{\mathrm{T}}N)^{-1} = (v_1, v_2, \ldots, v_r).$$

We assume that the set of vectors consisting of the rows of $F'(x^*)$, indexed by i_k, $k=1, \ldots, n-r$, is one of the largest linear independent sets of $F'(x^*)$, where $i_1 < i_2 < \cdots < i_{n-r}$. The other rows of $F'(x^*)$ are indexed by $j_1 < j_2 < \cdots < j_r$. We can construct an auxiliary function as follows:

$$T(x) = (f_{i_1}(x), f_{i_2}(x), \ldots, f_{i_{n-r}}(x), \widetilde{f}_{j_1}(x), \widetilde{f}_{j_2}(x), \ldots, \widetilde{f}_{j_r}(x))^{\mathrm{T}},$$

where $\widetilde{f}_{j_s}(x) = f_{j_s}(G_s^* x + b_s^*)$, $s = 1, 2, \ldots, r$,

$$(G_s^*)^{\mathrm{T}} = \frac{v_s \nabla f_{j_s}^{\mathrm{T}}(x^*)}{\|\nabla f_{j_s}(x^*)\|^2}, \quad b_s^* = (I - G_s^*)x^*. \tag{2.1}$$

It is easy to see that $T(x^*) = 0$ and $T'(x^*)$ is nonsingular. A basic algorithm for solving (1.1) with (1.2) can be given by applying the Brown method to $T(x)$.

Let $a_i \in \mathbb{R}^n$, $i=1, \ldots, j$, $A_j=(a_1, \ldots, a_j)$ and $H(a_1, \ldots, a_j)=A_j^{\mathrm{T}}A_j$. Clearly, $H(a_1, a_2, \ldots, a_j)$ is positive semi-definite and $\{a_1, a_2, \ldots, a_j\}$ is linearly independent if and only if $\det H(a_1, a_2, \ldots, a_j) \neq 0$. Let $H_j^* = H(\nabla f_{i_1}(x^*), \nabla f_{i_2}(x^*), \ldots, \nabla f_{i_j}(x^*))$ denote the matrix consisting of j column vectors taken arbitrarily from $F'(x^*)$ and $H_j^{(k)} = H(\nabla f_{i_1}(y_1^{(k)}), \nabla f_{i_2}(y_2^{(k)}), \ldots, \nabla f_{i_j}(y_j^{(k)}))$, where $y_j^{(k)}$ $(j = 1, 2, \ldots, n)$ is computed by the Algorithm 2.1. Let $r^* = \min_{1 \leqslant j \leqslant n}\{\det H_j^* \neq 0,\ 1 \leqslant i_1 < i_2 < \cdots < i_j \leqslant n\}$ and $\varepsilon^* < r^*/2$. One has that if $y_j^{(k)}$ closes enough to x^*, then $\det(H_j^{(k)}) \leqslant \varepsilon^*$ iff $\det(H_j^*) = 0$. Also let $A_j^{(k)} = (\nabla f_{i_1}(y_1^{(k)}),\ \nabla f_{i_2}(y_2^{(k)}), \ldots, \nabla f_{i_j}(y_j^{(k)}))$, if $A_j^{(k)}$ is of full rank, we adopt a set of the Householder elementary matrices $\overline{P}_1^{(k)}, \overline{P}_2^{(k)}, \ldots, \overline{P}_j^{(k)}$ to transform $A_j^{(k)}$ into the upper triangular matrix. There is an orthogonal matrix $\overline{Q}_{j+1}^{(k)} = \overline{P}_1^{(k)} * \cdots * \overline{P}_j^{(k)}$, such that

$$\overline{Q}_{j+1}^{(k)\mathrm{T}} A_j^{(k)} = \begin{pmatrix} \bar{s}_1^{(k)} & * & * & \cdots & * \\ 0 & \bar{s}_2^{(k)} & * & \cdots & * \\ 0 & 0 & \bar{s}_3^{(k)} & \cdots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \bar{s}_j^{(k)} \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}, \tag{2.2}$$

where $\overline{P}_i^{(k)}(1\leqslant i\leqslant j)$ is the ith Householder transformation in the transformation process of matrix $A_j^{(k)}$ and is of the form

$$\overline{P}_i^{(k)}=\begin{pmatrix} I_{(i-1)\times(i-1)} & 0\\ 0 & \widehat{P}_i^{(k)}\end{pmatrix}.$$

Therefore, $\det(H_j^{(k)})=\det(A_j^{(k)\mathrm{T}}A_j^{(k)})=\prod_{m=1}^{j}\overline{s}_m^{(k)2}=\det(H_{j-1}^{(k)})\overline{s}_j^{(k)2}$. In consequence, if $A_{j-1}^{(k)}$ is of full rank, then $A_j^{(k)}$ is of full rank iff $\overline{s}_j^{(k)}\neq 0$.

Moreover, one has

$$\overline{Q}_{j+1}^{(k)\mathrm{T}}A_j^{(k)}=(\overline{P}_1^{(k)}\nabla f_{i_1}(y_1^{(k)}),\overline{P}_2^{(k)}\overline{Q}_2^{(k)\mathrm{T}}\nabla f_{i_2}(y_2^{(k)}),\ldots,\overline{P}_j^{(k)}\overline{Q}_j^{(k)\mathrm{T}}\nabla f_{i_j}(y_j^{(k)})). \tag{2.3}$$

By steps 2–3 of Algorithm 2.1, $\overline{a}_m^k$ $(1\leqslant m\leqslant j)$ is the discretion of $\overline{Q}_m^{(k)\mathrm{T}}\nabla f_{i_m}(y_m^{(k)})$. From (2.2) and (2.3), it is obvious that the $s_m^{(k)}$ in Step 2 of Algorithm 2.1, obtained by performing the Householder transformation to $\overline{a}_m^{(k)}$, is naturally the approximation of the $\overline{s}_m^{(k)}$ obtained by using the Householder transformation to $\overline{Q}_m^{(k)\mathrm{T}}\nabla f_{i_m}(y_m^{(k)})$. From the continuity of elementary transformation on matrix, we can get that for a fixed k, one has

$$\lim_{h_k\to 0} s_m^{(k)}=\overline{s}_m^{(k)},\quad 1\leqslant m\leqslant j.$$

Consequently, we can estimate $\det(H_j^{(k)})$ by $s_j^{(k)}$ instead of $\overline{s}_j^{(k)}$. Additionally, let the columns of matrix $A^{(k)}(x)$ be $a_j^{(k)}(x)$, $j=1,\ldots,n$, where

$$a_j^{(k)\mathrm{T}}(x)=\left(\frac{f_j(x+h_ke_1)-f_j(x)}{h_k},\ldots,\frac{f_j(x+h_ke_n)-f_j(x)}{h_k}\right).$$

A modified Brown algorithm for solving a class of singular nonlinear systems is defined by the following steps.

Algorithm 2.1. Given h_0, $\varepsilon>0$ small enough, x_0 close sufficiently to x^* and set $H=\Phi$, $j_{\min}=0$, $k=0$.

Step 0: If $\|F(x_k)\|<\varepsilon$, then stop.

Step 1: Let $Q_1^{(k)}$ be an orthogonal matrix, $y_1^{(k)}=x_k$, $s_1^{(k)}=\pm\|a_1^{(k)}\|$, $H_0^{(k)}=I$, $D^{(k)}=1$, $j=1$.

Step 2: Compute $a_j^{(k)}=a_j^{(k)}(y_j^{(k)})$,

$$\overline{a}_j^{(k)}=\frac{1}{h_k}\begin{pmatrix}0\\ \vdots\\ 0\\ f_j(y_j^{(k)}+h_kQ_j^{(k)}e_j)-f_j(y_j^{(k)})\\ \vdots\\ f_j(y_j^{(k)}+h_kQ_j^{(k)}e_n)-f_j(y_j^{(k)})\end{pmatrix} \tag{2.4}$$

and $s_j^{(k)}=\pm\|\overline{a}_j^{(k)}\|$, $E^{(k)}=D^{(k)}|s_j^{(k)}|$.

If $E^{(k)} < \varepsilon^*$ and $j < n - j_{\min} + 1$,
then $\{j_{\min} \leftarrow j_{\min} + 1$, exchange the positions
between $f_j(x)$ and $f_{n-j_{\min}+1}(x)$ in $F(x)$,
$a_j(x)$ and $a_{n-j_{\min}+1}(x)$ in $A^{(k)}(x)$.
go to the beginning of Step 2$\}$.

$D^{(k)} = E^{(k)}$.

Step 3: Find a Householder transformation matrix $P_j^{(k)}$ being of the form

$$P_j^{(k)} = I - \rho u u^{\mathrm{T}} = \begin{pmatrix} I_{(j-1)\times(j-1)} & 0 \\ 0 & \widehat{P}_j^{(k)} \end{pmatrix},$$

such that $P_j^{(k)}\overline{a}_j^{(k)} = s_j^{(k)} e_j$, where $s_j^{(k)} = \pm\|\overline{a}_j^{(k)}\|$.

Step 4: Compute $Q_{j+1}^{(k)} = Q_j^{(k)} P_j^{(k)}$ and

$$y_{j+1}^{(k)} = y_j^{(k)} - s_j^{(k)-1} f_j(y_j^{(k)}) Q_{j+1}^{(k)} e_j.$$

Set $j \leftarrow j+1$. If $j < n - j_{\min} + 1$, then go to step 2.

If $j = n+1$, then go to Step 7.

Step 5: If $j = n - j_{\min} + 1$, then let $A_k = (a_1^{(k)}, a_2^{(k)}, \ldots, a_n^{(k)})^{\mathrm{T}}$, $r = j_{\min}$, by the pivoting Gauss eliminating method, compute N_k such that

$$(A_k + QP^{\mathrm{T}})N_k = Q, \quad \text{where} \quad N_k = (v_1^{(k)}, v_2^{(k)}, \ldots, v_r^{(k)}),$$

where matrices $Q, P \in \mathbb{R}^{n\times r}$ are generated by a random function.

Step 6: Set $s = j - n + r$, let

$$G_s^{(k)\mathrm{T}} = \frac{v_s^{(k)} a_j^{(k)\mathrm{T}}}{\|a_j^{(k)}\|^2} \tag{2.5}$$

$$\text{compute } b_s^{(k)} = (1 - G_s^{(k)})x_k \text{ and set } \widetilde{f}_j^{(k)}(x) = f_j(G_s^{(k)}x + b_s^{(k)}) \to f_j(x), \tag{2.6}$$

go to Step 2.

Step 7: Let $x_{k+1} = y_{n+1}^{(k)}$, $h_{k+1} = \mathrm{O}(\|F(x_{k+1})\|)$, $k \leftarrow k+1$ and go to Step 0.

End of Algorithm 2.1.

Remark. (1) Note that if $E^{(k)} < \varepsilon^*$ in Step 2, then $\nabla f_{i_j}(y_j^{(k)})$ is regarded as a linear combination of $\nabla f_{i_m}(y_m^{(k)})$, $m = 1, 2, \ldots, j-1$.

(2) We will prove the $A_k + QP^{\mathrm{T}}$ in Step 5 is nonsingular later.

3. Convergence analysis

In this section, the local convergence is investigated and Q-quadratic convergence is demonstrated. In the following, $\|x\|$ denotes the Euclidean norm for $x \in \mathbb{R}^n$ and $\|A\|$ denotes the Frobenius norm for $A \in \mathbb{R}^{n,n}$. To begin with, the following assumptions are needed.

Basic assumptions:

(A) There exist two positive constants $r_0 > 0$ and $K_0 > 0$, such that for any $x,\ y \in B(x^*,\ r_0),\ j = 1,\ 2, \ldots, n$,

$$\|F'(x) - F'(y)\| \leqslant K_0 \|x - y\|. \tag{3.1}$$

(B)

$$\begin{aligned} &\operatorname{rank}(F'(x^*)) = n - r, \quad && 1 \leqslant r \leqslant n, \\ &\nabla f_i(x^*) \neq 0, && i = 1, 2, \ldots, n. \end{aligned}$$

(C) The rows of $J(x^*) = F'(x^*)$, indexed by $i_1 < i_2 < \cdots < i_{n-r}$, form a largest linearly independent set, and the subscripts of the rest are denoted by $j_1,\ j_2, \ldots, j_r$.

Therefore, when k is large enough, the nonlinear equations and its approximation A_k to Jacobian matrix $J(x^*) = F'(x^*)$ are rearranged as follows: $F(x) = (f_{i_1}(x),\ f_{i_2}(x), \ldots, f_{i_{n-r}}(x), f_{j_1}(x), f_{j_2}(x), \ldots, f_{j_r}(x))^{\mathrm{T}},\ A_k = (a_{i_1}^{(k)}(y_1^{(k)}), a_{i_2}^{(k)}(y_2^{(k)}), \ldots, a_{i_{n-r}}^{(k)}(y_{n-r}^{(k)}), a_{j_1}^{(k)}(y_{t_1}^{(k)}), a_{j_2}^{(k)}(y_{t_2}^{(k)}), \ \ldots, a_{j_r}^{(k)}(y_{t_r}^{(k)}))^{\mathrm{T}}$, where $y_{t_1}^{(k)}, y_{t_2}^{(k)}, \ldots, y_{t_r}^{(k)}$ are some elements among $y_{i_1}^{(k)}, y_{i_2}^{(k)}, \ldots, y_{i_{n-r}}^{(k)}$.

Lemma 3.1 (*Ortego and Rheinboldt*). *Assume that* (3.1) *holds, then for any* $x, y \in B(x^*, r_0)$, *one has* $\|F(y) - F(x) - J(x)(y - x)\| \leqslant K_0 \|y - x\|^2$.

Lemma 3.2. *Let* $T(x) = (f_{i_1}(x),\ f_{i_2}(x), \ldots, f_{i_{n-r}}(x),\ \widetilde{f}_{j_1}(x),\ \widetilde{f}_{j_2}(x), \ldots, \widetilde{f}_{j_r}(x))^{\mathrm{T}}$. *Then* $T(x^*) = 0$ *and* $T'(x^*)$ *is of full rank.*

The lemma given above can be proved from (B) and (C) of the basic assumptions at the beginning of this section and Lemma 3.1 can be obtained directly from (3.1).

Lemma 3.3. *If* $r_2 (r_2 < r_0)$ *is small enough, then there exist* $\delta > 0$ *and* $L > 0$ *such that if* $x,\ y \in B(x^*, r_2)$, *one has that*

1. $\|T'(x) - T'(x^*)\| \leqslant L\|x - x^*\|,\ \ \|T(y) - T(x) - T'(x^*)(y - x)\| \leqslant L\|y - x\| \max\{\|x - x^*\|, \|y - x^*\|\}$;
2. $\|x - x^*\| \leqslant \delta \|T'(x^*)^{-1}\| \|T(x)\|$.

Proof. From (2.1) and (3.1), it is easy to see that conclusion (1) of this Lemma holds. By virtue of $T(\cdot)$ in Lemma 3.2, one has

$$\begin{aligned} T'(x^*)^{-1} T(z) &= T'(x^*)^{-1}(T(z) - T(x^*)) \\ &= T'(x^*)^{-1} \int_0^1 T'(x^* + t(z - x^*))(z - x^*) \mathrm{d}t \\ &= (z - x^*) + T'(x^*)^{-1} \int_0^1 (T'(x^* + t(z - x^*)) - T'(x^*))(z - x^*)\, \mathrm{d}t. \end{aligned}$$

There exists L such that

$$\|z - x^*\| - L/2 \|T'(x^*)^{-1}\| \|z - x^*\|^2 \leqslant \|T'(x^*)^{-1} T(z)\|.$$

according to conclusion (1) of this Lemma. Let $\delta^{-1} = 1 - (L/2)r_0\|T'(x^*)^{-1}\|(\delta^{-1} < 1)$. The value on the right-hand side of the formula above, i.e. δ^{-1}, is greater than zero when r_2 is small enough. It leads to the second conclusion. □

Lemma 3.4. *If r_2 is small enough, then $\|T(x)\| \leqslant K_1\|x - x^*\|,\ x \in B(x^*, r_2),\ K_1 = 0.5Lr_2 + \|T'(x^*)\|$.*

From Step 7 of Algorithm 2.1, it follows that there exists a real number $K_2 > 0$ such that $|h_k| \leqslant K_2\|T(x_k)\|$.

Lemma 3.5. *Under the basic assumptions and Algorithm 2.1, there exists a real number r_3 satisfying $0 < r_3 < r_2$, and a constant c_1 such that for any $1 \leqslant j \leqslant n - r$, we have the conclusion that if $|h_k| < r_3$ and $\|y_1^{(k)} - x^*\| \leqslant r_3$ hold, then $\|y_{j+1}^{(k)} - x^*\| \leqslant c_1\|y_1^{(k)} - x^*\|$ and $|s_j^{(k)}| \geqslant 1/(2\|\widetilde{J}^{-1}(x^*)\|)$.*

Remark. The difference between Lemma 3.5 and Lemmas 6 and 7 of [3] lies in that the latter is given based on the condition that matrix is of full rank, see [3].

Lemma 3.6. *Under Assumption (A), if r_3 is small enough, then there exists $L_1 > 0$ such that if $y_1^{(k)} = x_k \in B(x^*,\ r_3)$, then one has*

1. $\|a_j^{(k)} - \nabla f_j(x^*)\| \leqslant L_1\|x_k - x^*\|,\ 1 \leqslant j \leqslant n,$
2. $\|A_k - J(x^*)\| \leqslant \sqrt{n}L_1\|x_k - x^*\|,$
3. $2\|\nabla f_j(x^*)\| \geqslant \|a_j^{(k)}\| \geqslant \|\nabla f_j(x^*)\|/2.$

Proof. Let $L_1 = \sqrt{n}K_0(c_1 + K_1K_2)$, where $r_3 < (c_1 + K_1K_2)^{-1}r_1 \leqslant r_2$. It can be verified that for any m, $m \in \{1, 2, \ldots, n - r\}$, one has $y_m^{(k)} + h_ke_i \in B(x^*, r_2) \subseteq B(x^*, r_0)$. From Lemmas 3.4 and 3.5, one has

$$
\begin{aligned}
\|a_{i_m}^{(k)} - \nabla f_{i_m}(x^*)\|^2 &= \sum_{i=1}^{n} |(f_{i_m}(y_m^{(k)} + h_ke_i) - f_{i_m}(y_m^{(k)}))/h_k - \nabla f_{i_m}(x^*)^{\mathrm{T}}e_i|^2 \\
&\leqslant nK_0^2(\|y_m^{(k)} - x^*\| + |h_k|)^2 \\
&\leqslant nK_0^2(c_1 + K_1K_2)^2\|(x_k - x^*)\|^2.
\end{aligned}
$$

Similarly, we can prove that for $j \neq i_1, i_2, \ldots, i_{n-r}$,

$$\|a_j^{(k)} - \nabla f_j(x^*)\|^2 \leqslant nK_0^2(c_1 + K_1K_2)^2\|(x_k - x^*)\|^2.$$

The argument given above implies that the three conclusions are valid if r_3 is small enough. □

Lemma 3.7. *If r_3 is small enough, then there exist $K_3 > 0$ and $K_4 > 0$ such that if $\|x_k - x^*\| < r_3$, one has $A_k + QP^{\mathrm{T}}$ is nonsingular and*

(1) $\|N_k - N(P^{\mathrm{T}}N)^{-1}\| \leqslant K_3\|x_k - x^*\|$, *where N_k is defined in Algorithm* 2.1,
(2) $\|G_s^* - G_s^{(k)}\| \leqslant K_4\|x_k - x^*\|$.

Proof. Conclusion (1) can be obtained by Step 5 of Algorithm 2.1 and Lemma 3.6. By Lemma 3.6, we have

$$\begin{aligned}\|G_s^{(k)} - G_s^*\| &= \left\| \frac{v_s^{(k)} a_{j_s}^{(k)\mathrm{T}}}{\|a_{j_s}^{(k)}\|^2} - \frac{v_s^* \nabla f_{j_s}^{\mathrm{T}}(x^*)}{\|\nabla f_{j_s}^{\mathrm{T}}(x^*)\|^2} \right\| \\ &\leqslant \frac{1}{\|a_{j_s}^{(k)}\| \|\nabla f_{j_s}^{\mathrm{T}}(x^*)\|^2} (\|a_{j_s}^{(k)}\|^2 \|\nabla f_{j_s}^{\mathrm{T}}(x^*)\| \|v_s^* - v_s^{(k)}\| \\ &\quad + \|\nabla f_{j_s}^{\mathrm{T}}(x^*)\|^2 \|v^*\| \|\nabla f_{j_s}^{\mathrm{T}}(x^*) - a_{j_s}^{(k)\mathrm{T}}\| \\ &\quad + \|v^* \nabla f_{j_s}^{\mathrm{T}}(x^*)\| | \|a_{j_s}^{(k)\mathrm{T}}\|^2 - \|\nabla f_{j_s}^{\mathrm{T}}(x^*)\|^2 |) \\ &\leqslant \frac{4}{\|\nabla f_{j_s}^{\mathrm{T}}(x^*)\|^2} (2K_3 \|\nabla f_{j_s}^{\mathrm{T}}(x^*)\| + 4\|v_s^*\|) \|x_k - x^*\| \\ &= K_4 \|x_k - x^*\| \qquad \square.\end{aligned}$$

Lemma 3.8. *Under the basic assumptions given at the beginning of this section, it follows from Lemmas* 3.6 and 3.7 *that there exist* $r_4 (r_4 < r_3)$, L_3, L_4, *such that for any* $1 \leqslant s \leqslant r$ *and* $x, y \in B(x^*, r_4)$, *one has that*

(1) $\|\nabla \widetilde{f}_{j_s}^{(k)}(x^*) - \nabla \widetilde{f}_{j_s}(x^*)\| \leqslant L_3 \|x^{(k)} - x^*\|$,
(2) $|\widetilde{f}_{j_s}^{(k)}(x) - \widetilde{f}_{j_s}^{(k)}(y) - \nabla \widetilde{f}_{j_s}(x^*)(x - y)| \leqslant L_4 \|x - y\| (\max\{\|y - x^*\|, \|x - x^*\|\} + \|x_k - x^*\|)$.

Proof. Firstly, one has

$$\begin{aligned}\|G_s^{(k)} x + b_s^{(k)} - x^*\| &= \|G_s^{(k)} x + b_s^{(k)} - G_s^* x^* - b_s^*\| \\ &\leqslant \|x_k - x^*\| + \|G_s^{(k)}\| \|x - x^*\| + \|G_s^{(k)}\| \|x_k - x^*\| \\ &\leqslant (1 + \|G_s^*\| + K_4 r_0) \max\{\|x_k - x^*\|, \|x - x^*\|\}. \end{aligned} \tag{3.2}$$

Evidently, if r_4 is small enough and $x, x_k \in B(x^*, r_4)$, then we have $G_s^{(k)} x + b_s^{(k)} \in B(x^*, r_2)$. Consequently, from (3.2) we have

$$\begin{aligned}\|\nabla \widetilde{f}_{j_s}^{(k)}(x^*) - \nabla \widetilde{f}_{j_s}(x^*)\| &= \|(G_s^{(k)})^{\mathrm{T}} \nabla f_{j_s}(G_s^{(k)} x^* + b_s^{(k)}) - (G_s^*)^{\mathrm{T}} \nabla f_{j_s}(x^*)\| \\ &\leqslant \|G_s^{(k)} - G_s^*\| (\|\nabla f_{j_s}(x^*)\| + K_0 \|G_s^{(k)}\| \|x_k - x^*\|) \\ &\quad + K_0 \|G_s^{(k)}\| \|G_s^{(k)}\| \|x_k - x^*\| \leqslant L_3 \|x_k - x^*\|.\end{aligned}$$

Since

$$\begin{aligned}|\widetilde{f}_{j_s}^{(k)}(x) - \widetilde{f}_{j_s}^{(k)}(y) - \nabla \widetilde{f}_{j_s}^{(k)}(x^*)(x - y)| &\leqslant K_0 \|G^{(k)}\|^2 \|x - y\| \max\{\|y - x^*\|, \|x - x^*\|\} \\ &\leqslant L_2 \|x - y\| \max\{\|y - x^*\|, \|x - x^*\|\},\end{aligned}$$

we have

$$\begin{aligned}|\widetilde{f}_{j_s}^{(k)}(x) - \widetilde{f}_{j_s}^{(k)}(y) - \nabla \widetilde{f}_{j_s}(x^*)(x - y)| &\leqslant L_2 \|x - y\| \max\{\|y - x^*\| + \|x - x^*\|\} \\ &\quad + L_3 \|x_k - x^*\| \|x - y\| \\ &\leqslant L_4 \|x - y\| (\max\{\|y - x^*\|, \|x - x^*\|\} + \|x_k - x^*\|),\end{aligned}$$

where $L_4 = \max\{L_2, L_3\}$. The demonstration is completed. $\square$

In order to simplify the following discussion, we give the notations below

$$\begin{aligned}\overline{f}_m(x) &= f_{i_m}(x), \quad m = 1, 2, \ldots, n-r,\\ \overline{f}_m(x) &= \widetilde{f}_{j_s}(x), \quad m = n-r+1, \ldots, n, \ \ s = m-n+r.\end{aligned} \tag{3.3}$$

Lemma 3.9. *Under the assumption of Lemma* 3.8, *if* r_4 $(r_4 < r_3)$ *is small enough, there exist* c_3, c_4, *such that for any* $1 \leqslant j \leqslant n$ *and* $h_k < r_5$, *one has*

(1) *If* $\|y_1^{(k)} - x^*\| \leqslant r_4, \ldots, \|y_j^{(k)} - x^*\| \leqslant r_4$, *then* $\|y_{j+1}^{(k)} - x^*\| \leqslant c_3 \|x^{(k)} - x^*\|$,
(2) $\|y_j^{(k)} - x^*\| \leqslant c_4 \|x_k - x^*\|$ *and* $|s_j^{(k)}| \geqslant 1/(2\|\widetilde{J}^{-1}(x^*)\|)$, *where* $\widetilde{J}(x^*) = T'(x^*)$.

Proof. Firstly, by lemma 3.6, if $1 \leqslant j \leqslant n-r$, conclusions (1) and (2) obviously hold. If $n-r < j \leqslant n$, denote $\widetilde{J} = \widetilde{J}(x^*)$. Simplicity, keep i and j fixed, and let $L = (l_{p,q})$, where

$$l_{p,q} = \begin{cases} e_p^{\mathrm{T}} \widetilde{J} Q_{j+1}^{(k)} e_q, & 1 \leqslant q < p \leqslant n, \quad j < p \leqslant q \leqslant n,\\ s_p^{(k)}, & 1 \leqslant p = q \leqslant j,\\ 0 & \text{otherwise.}\end{cases}$$

Suppose $p \leqslant j$. By Lemma 3.1 and the structure of $\overline{a}_p^{(k)}$ (Step 2 of Algorithm 2.1), one has

$$|(\overline{a}_p^{(k)\mathrm{T}} - e^{\mathrm{T}} \widetilde{J}(y_p^{(k)}) Q_p^{(k)}) e_q| \leqslant K_0 h_k / 2, \ \ q = p, p+1, \ldots, n-r. \tag{3.4}$$

If $q > n-r$, then let $s = q-n+r$. Firstly, by the assumption of induction and (3.2), one has that if $r_4(r_4 < r_3)$ is small enough and $y_p^{(k)} \in B(x^*, r_4)$, then $G_s^{(k)} y_p^{(k)} + b_s^{(k)} \in B(x^*, r_2)$. Furthermore, if any $x \in B(x^*, r_4)$, we also have $G_s^{(k)} x + b_s^{(k)} \in B(x^*, r_2)$. Hence, from Lemma 3.3, Lemma 3.9 (2) and the assumptions, we have that for any q, $n-r+1 \leqslant q \leqslant n$,

$$\begin{aligned}|(\overline{a}_p^{(k)\mathrm{T}} - e^{\mathrm{T}} \widetilde{J}(y_p^{(k)}) Q_p^{(k)}) e_q| &= |\overline{a}_p^{(k)\mathrm{T}} e_q - \nabla \widetilde{f}_{j_s}(y_p^{(k)})^{\mathrm{T}} Q_p^{(k)} e_q|\\ &\leqslant \frac{1}{h_k} |\widetilde{f}_{j_s}^{(k)}(y_p^{(k)} + h_k Q_p^{(k)} e_q) - \widetilde{f}_{j_s}^{(k)}(y_p^{(k)}) - \nabla \widetilde{f}_{j_s}(x^*)^{\mathrm{T}} Q_p^{(k)} e_q h_k|\\ &\quad + |(\nabla \widetilde{f}_{j_s}(x^*) - \nabla \widetilde{f}_{j_s}(y_p^{(k)}))^{\mathrm{T}} Q_p^{(k)} e_q|\\ &\leqslant L_4(\|y_p^{(k)} - x^*\| + |h_k| + \|x_k - x^*\|) + L\|y_p^{(k)} - x^*\|\\ &\leqslant L_5 \max\{\|y_p^{(k)} - x^*\|, \|x^{(k)} - x^*\|\}.\end{aligned} \tag{3.5}$$

Therefore, from (3.4) and (3.5), for any $p \leqslant j$, $q = p, p+1, \ldots, n$, we have

$$|(\overline{a}_p^{(k)\mathrm{T}} - e^{\mathrm{T}} \widetilde{J}(y_p^{(k)}) Q_p^{(k)}) e_q| \leqslant L_6 \max\{\|y_p^{(k)} - x^*\|, \|x^{(k)} - x^*\|\}. \tag{3.6}$$

We can prove that there exists a $M' > 0$ such that $\|L - \widetilde{J} Q_{j+1}^{(k)}\| \leqslant M' r_4$, which is similar to the proof of [3, Lemma 6]. Therefore, if r_4 is small enough, we have

$$|s_j^{(k)}| \geqslant \frac{1}{2\|\widetilde{J}^{-1}\|}. \qquad \square$$

By Algorithm 2.1, one has $\|y_{j+1}^{(k)} - x^*\| \leqslant c_3\|x^{(k)} - x^*\|$. Similarly, by [3, Lemma 3.7], one has that conclusion (2) holds. Now we establish the main result, the convergence theorem.

Theorem 3.1 (*Convergence theorem*). *Suppose assumptions* (A), (B) and (C) *are valid. Then there exists a constant* $\tau > 0$ *such that for any* $x_0 \in B(x^*, \tau)$ *the sequence* $\{x_k\}$ *generated by the algorithm converges* Q*-quadratically to* x^*.

Proof. Firstly, take $y_1^{(k)} = x_k \in B(x^*,\ r_5)$, where $r_5 < r_4$. If $1 \leqslant j \leqslant n$, then from Lemma 3.4, Lemma 3.9 and (3.3) one has

$$\|\overline{f}_j(y_j^{(k)})\| \leqslant c_4\|x_k - x^*\|. \tag{3.7}$$

According to the proof of [3, Lemma 3.7], there exists a constant c_5 such that for any j, $1 \leqslant j \leqslant n$, we have

$$\|s_j^{(k)} - e_j^{\mathrm{T}}\widetilde{J}(x^*)Q_{n+1}^{(k)}e_j\| \leqslant c_5\|x_k - x^*\|.$$

So there exists a constant c_6, such that $\|s_j^{(k)} - e_j^{\mathrm{T}}\widetilde{J}(y_j^{(k)})Q_{n+1}^{(k)}e_j\| \leqslant c_6\|x_k - x^*\|$. Therefore,

$$|\overline{f}_j(y_j^{(k)}) - e_j^{\mathrm{T}}\widetilde{J}(y_j^{(k)})s_j^{(k)-1}\overline{f}_j(y_j^{(k)})Q_{n+1}^{(k)}e_j| \leqslant 2c_4c_6\|\widetilde{J}(x^*)^{-1}\|\|x_k - x^*\|^2.$$

By the definition of $y_{j+1}^{(k)}$, one has

$$|\overline{f}_j(y_j^{(k)}) - e_j^{\mathrm{T}}\widetilde{J}(y_j^{(k)})(y_j^{(k)} - y_{j+1}^{(k)})| \leqslant 2c_4c_6\|\widetilde{J}(x^*)^{-1}\|\|x_k - x^*\|^2. \tag{3.8}$$

Thus, from (3.8) and Lemma 3.4, we have

$$|\overline{f}_j(y_{j+1}^{(k)})| \leqslant \overline{c}_8\|x_k - x^*\|^2 + L\|y_{j+1}^{(k)} - y_j^{(k)}\|^2. \tag{3.9}$$

$\|y_{j+1}^{(k)} - y_j^{(k)}\|^2$ is of the order $\mathrm{O}(\|x_k - x^*\|^2)$, we have

$$|\overline{f}_j(y_{j+1}^{(k)})| \leqslant c_8\|x_k - x^*\|^2. \tag{3.10}$$

Now we estimate $\overline{f}_j(y_{n+1}^{(k)}) - \overline{f}_j(y_{j+1}^{(k)})$. By virtue of Steps 2–4 of Algorithm 2.1 and the constitution of $Q_j^{(k)}$, we have

$$\begin{aligned} |\overline{f}_j(y_{n+1}^{(k)}) - \overline{f}_j(y_{j+1}^{(k)})| &= |\nabla\overline{f}_j(u_{n+1}^{(k)})^{\mathrm{T}}(y_{n+1}^{(k)} - y_{j+1}^{(k)})| \\ &= |\nabla\overline{f}_j(u_{n+1}^{(k)})^{\mathrm{T}}\Sigma_{m=j}^{n}s_m^{(k)-1}\overline{f}_m(y_m^{(k)})Q_{m+1}^{(k)}e_m| \\ &= |(\nabla\overline{f}_j(u_{n+1}^{(k)}) - Q_j^{(k)}\overline{a}_j^{(k)})^{\mathrm{T}}\Sigma_{m=j}^{n}s_m^{(k)-1}\overline{f}_m(y_m^{(k)})Q_{m+1}^{(k)}e_m|. \end{aligned} \tag{3.11}$$

From the proof of Lemma 3.9, there exists a constant c_9 such that

$$|\overline{f}_j(y_{n+1}^{(k)}) - \overline{f}_j(y_{j+1}^{(k)})| \leqslant c_9\|x_k - x^*\|^2.$$

Also, from (3.10) and (3.11), we have

$$|\overline{f}_j(y_{n+1}^{(k)})| \leqslant (c_8 + c_9)\|x_k - x^*\|^2. \tag{3.12}$$

By virtue of (3.12), there exists a constant $N>0$ such that $\|T(x_{k+1})\| \leqslant N\|x_k - x^*\|^2$. It follows from (2) in Lemma 3.4 that

$$\|x_{k+1} - x^*\| \leqslant \delta N \|T'(x^*)^{-1}\| \|x_k - x^*\|^2. \tag{3.13}$$

If τ is small enough, $\tau < \min\{r_6, (\delta\|T'(x^*)^{-1}\|N)^{-1}\}$ and $\|x_0 - x^*\| < \tau$, then by induction, it can be proved from (3.13) that for all $k>0$ and $x_0 \in B(x^*, \tau)$ one has

$$y_1^{(k)} \in B(x^*, r_6)$$

$$\|x_{k+1} - x^*\| \leqslant \delta\tau N \|T(x^*)^{-1}\| \|x_k - x^*\|. \tag{3.14}$$

where $B(\cdot, \cdot)$ denotes an open ball. Taking the limit of the second line of (3.14), one has

$$\lim_{k\to\infty} x_k = x^*.$$

Summarizing the statement given above, it follows from (3.13) that $\{x_k\}$ generated by the algorithm converges Q-quadratically to x^*. □

4. Numerical experiments

We take some examples from [15], satisfying (B) of the basic assumptions in Section 3 (see Table 1). Using the formulation

$$\widehat{F}(x) = F(x) - F'(x_*)A(A^{\mathrm{T}}A)^{-1}A^{\mathrm{T}}(x - x^*) \tag{4.1}$$

due to Dan et al. (1993), one has that (4.1) is of the rank one defect if taking $A^{\mathrm{T}} = (1, 1, \ldots, 1)$, of the rank two defect if taking

$$A^{\mathrm{T}} = \begin{pmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & -1 & 1 & -1 & \cdots & (-1)^n \end{pmatrix}.$$

In what follows, some computation results are given via the algorithm presented in Section 2, and related comparison of the results obtained by Algorithm 2.1 with the ones given by other authors, in the case that the same choices of matrices A mentioned above are used, are listed by Tables 2–4.

Table 1
The start points of the test functions

Functions	Start points
Bigg exp 6	(1, 10, 1)
Box 3D	(1.5, 10.5, 1.5)
Broyden banded	$(-1, -1, \ldots, -1)$
Rosenbrock	$(-1.2, 1)$
Powell singular	$(3, -1, 0, 1)$
Brown alm	$(\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2})$

Table 2
Results on the nonsingular cases by the tensor method and the modified Brown method

Function	n	k	TM	x^*?	k	MBM	x^*?
Bigg exp 6	6	70	0.13–12	Y	19	0.86–17	Y
Box 3D	3	3	0.10–11	Y	3	0.14–21	Y
Broyden banded	30	4	0.12–11	Y	5	0.29–17	Y
Rosenbrock	4	7	0.14–20	Y	3	0.23–24	Y
Powell singular	4	3	0.25–15	Y	16	0.72–15	Y
Brown alm	10	7	0.38–11	Y	7	0.92–25	Y

Table 3
Results on the first singular test set with rank $(F'(x_*) = n-1)$

Function	n	k	TM	x^*?	k	MBM	x^*?
Bigg exp 6	6	150	∞	N	7	0.19–16	Y
Box 3D	3	5	0.57–15	N	6	0.38–22	Y
Broyden banded	30	4	0.12–11	Y	6	0.98–26	Y
Rosenbrock	4	3	0.47–14	Y	3	0.37–24	Y
Powell singular	4	3	0.25–15	Y	16	0.72–15	Y
Brown alm	10	4	0.41–7	Y	4	0.21–20	Y

Table 4
Results on the second singular test set with rank $(F'(x_*) = n-2)$

Function	n	k	TM	x^*?	k	MBM	x^*?
Bigg exp 6	6	150	∞	N	5	0.18–09	Y
Brown alm	10	4	0.9–13	N	3	0.27–23	Y
Box 3*D*	10	11	0.2–12	N	19	0.39–15	Y

In Tables 2–4, the two columns labelled x^*?, contain “Y(yes)” if the method converged to the singular points, “N(no)” otherwise; the results in the two columns labelled TM and MBM are the values of $\frac{1}{2}\|F(x_k)\|_2^2$; n denotes the number of variables; k denotes the number of iterations; “0.13–12” means 0.13×10^{-12}; TM denotes the tensor method and MBM denotes the modified Brown method.

Remark. Matrices Q and P are generated by a random function using MATLAB 6.1 language. In the process of iteration, we select $h_k = \min\{c\|F(x_k)\|, 10^{-8}\}$, where $0 < c < 0.0001$.

4.1. Comparison

(1) *The comparison of calculating amount*: Tensor method is needed to calculate accurate Jacobian Matrix per iterative step, but it is difficult and more complicated than the nonlinear equations. The total cost of solving the tensor model is about $\frac{2}{3}n^3 + n^2p + O(n^2)$ multiplications and additions in the dense

case. My method need not calculate accurate Jacobian Matrix $F'(x_k)$. We use an approximate matrix A_k to substitute $F'(x_k)$, hence $N^2/2 + \mathrm{O}(n)$ function evaluation is necessary. The total cost of the method proposed in this paper is $n^3 + n^2/2r + \mathrm{O}(n)$.

(2) *Evaluation from the numerical experiments*: We can derive the conclusion that the approximate solution obtained by using MBM is far more accurate than ones of TM from Table 3. In Table 4, we find that the approximate solution obtained by using MBM convergence to the singular point while the ones of TM do not converge to the singular point at all.

It can be seen from the comparisons given above that the modified Brown method is highly efficient and locally *Q*-quadratic convergent under the rank defect conditions of $F(x)$.

References

[1] E.L. Allgower, K. Böhmer, Resolving singular nonlinear equations, Rocky Mountain Math. J. 18 (1987) 225–268.

[2] W.J. Beyn, Defining Equations for Singular Solutions and Numerical Applications, vol. ISNM 70, Birkhäuser, Basel, 1984, pp. 42–56.

[3] R.P. Brent, Some efficient algorithm for solving systems to nonlinear equations, SIAM J. Numer. Anal. 10 (2) (1973) 327–344.

[4] F. Dan, P.D. Frank, R.B. Schnabel, Local convergence analysis of tensor methods for non-linear equations, Math. Program. 62 (1993) 427–459.

[5] D.W. Decker, H.B. Keller, C.T. Kelley, Newton's method at singular points I, II, SIAM J. Numer. Anal. 17 (1980) 66–70.

[6] D.W. Decker, C.T. Kelley, Convergence rates for Newton's method at singular points, SIAM J. Numer. Anal. 20 (1983) 296–314.

[7] R.-D. Ge, Z.-Q. Xia, An ABS algorithm for solving singular nonlinear systems with rank one defect, Korea. J. Comput. Appl. Math. 9 (1) (2002) 167–184.

[8] R.-D. Ge, Z.-Q. Xia, An algorithm for solving singular nonlinear systems with rank defects, J. Appl. Math. Comput. 12 (2003) 1–17.

[9] A. Griewank, On solving nonlinear equations with simple singularities or nearly singular solutions, SIAM Rev. 27 (1985) 537–563.

[10] A. Hoy, A relation between Newton and Gauss–Newton steps for singular nonlinear equations, Computing 40 (1988) 19–27.

[11] A. Hoy, An efficiently implementable Gauss–Newton-like method for solving singular nonlinear equations, Computing 41 (1989) 107–122.

[12] A. Hoy, H. Schwetlick, Some superlinearly convergent methods for solving singular nonlinear equations, Lectures in Applied Mathematics, vol. 26, 1990, pp. 285–299.

[13] Y. Kanzawa, S. Oishi, Imperfect singular solutions of nonlinear equations and numerical method of proving their existence, IEICE Trans. E82-A 6 (1999) 1062–1069.

[14] R. Menzel, G. Pönisch, A quadratically convergent method for computing simple singular roots and application to determining simple bifurcation points, Computing 32 (1984) 127–138.

[15] J.J. Moré, B.S. Garbow, K.E. Hillstrom, Testing unconstrained optimization software, ACM Trans. Math. Software 7 (1981) 17–41.

[16] R.B. Schnabel, T. Chow, Tensor methods for unconstrained optimization using second derivatives, SIAM J. Optim. 1 (1991) 293–315.

[17] R.B. Schnabel, P.D. Frank, Tensor method for nonlinear equations, SIAM J. Numer. Anal. 21 (1984) 815–843.

[18] H. Werber, H. Werner, On the accurate determination of nonisolated solutions of nonlinear equations, Computing 26 (1981) 315–326.

[19] N. Yamamoto, Regularization of solutions of nonlinear with singular Jacobian matrices, J. Inform. Process. 17 (1) (1984) 16–21.