Comparison of Derivative-free Method and Finite-difference Method for Singular Systems

Sandra Buhmiler[1], Sanja Rapajić[2], Slavica Medić[1], Nataša Duraković[1], Tatjana Grbić[1]

[1]University of Novi Sad, Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia

[2] University of Novi Sad, Faculty of Sciences, Trg Dositeja Obradovića 3, 21000 Novi Sad, Serbia

e-mails: sandrabu@uns.ac.rs, sanja@dmi.uns.ac.rs, slavicam@uns.ac.rs, natasa.durakovic@uns.ac.rs, tatjana@uns.ac.rs

Abstract: In this paper, the derivative-free method for solving singular systems is compared with the classical finite-difference method for nonlinear systems. Since the convergence rate of an iterative method to singular solution drops down, the convergence can be accelerated by forming the bordered system. Left and right singular vectors of the finite-difference approximation of the Jacobian are used for the construction of the bordered system. The local algorithm for finding a solution is tested on several examples and compared with the finite-difference method. The obtained numerical results, which are promising, indicate fast local convergence of the proposed derivative-free method and point out that it has better performances than the finite-difference method.

Keywords: nonlinear system; singular system; bordered system; finite-difference approximation

1 Introduction

Consider the nonlinear system

$$F(x) = 0, \tag{1}$$

where $F: \mathbb{R}^n \to \mathbb{R}^n$, $F(x) = (F_1(x), F_2(x), \dots, F_n(x))^T$ and let x^* be the solution of (1). Nonsingularity of the Jacobian matrix $F'(x^*)$, namely nonsingularity of the problem, allows one to use Newton's method for finding the solution of the system (1). Newton's method is one of the most popular iterative methods because of its local quadratic convergence. However, Newton's method is very expensive since in every iteration the Jacobian matrix should be calculated. That is why

quasi-Newton (QN) methods have been developed. In every iteration, QN methods use the Jacobian approximation, which is obtained by updating the previous one, using basic matrix operations.

When the Jacobian matrix $F'(x^*)$ is singular, the problem is singular and the convergence rate of an iterative method decreases. One of the most important results about the convergence of Newton's method in the singular case, [1], shows that the region of the acceptable initial points has changed, compared to the regular case. Convergence rate of the component that corresponds to the null space of the Jacobian $F'(x^*)$ is slower and it affects the convergence of the entire iterative sequence. These results were extended to singular problems having higher-order singularities, [2], [3]. A more detailed convergence analysis of Newton's method can be found in [4], while similar results for the convergence of QN methods are given in [5], [6], [7].

To accelerate convergence to the singular solution, one can modify the method or modify the problem. The acceleration of Newton's method in several ways can be found in [8]. Modifications of the method such as including mid-iteration or including second derivative, Hessian, can speed up the convergence. The modifications with mid-iteration are given for Newton's method, [9], inexact Newton's method, [7], and Shamanski's method, [10]. Based on these modifications, in [11], two modifications of the arbitrary QN method are suggested and tested for Thomas', [12], Broyden's, [13], and Martinez's, [14], methods. The other methods which use the second derivative are proposed in [15], [16], [17].

Constructing the bordered system, the initial singular system is transformed into the regular one by inserting additional variables. For problems having isolated solutions and Jacobian rank deficiency $q = 1$, the construction of the bordered system is proposed in [18], although the method given in [8] can be used to accelerate the convergence to the solution that is not singular. Under certain conditions, imposing additional variables to the singular problem leads to rapid convergence to the solution. The construction of the bordered system for the problems with Jacobian $F'(x^*)$ having rank deficiency $q \geq 1$ can be found in [8], [19], and [20].

In [21], the bordered system was formed using left and right singular vectors of the Jacobian $F'(\bar{x})$, where $\bar{x}$ is close enough to the solution x^*. The fast local algorithm was proposed and its convergence to the solution of the initial problem was theoretically proven and applied on several numerical examples. Since in many applications, function $F(x)$ is not available analytically and direct evaluation of the Jacobian and Hessian is not possible, the bordered system and fast local and global algorithm for finding its solution were presented in [22], using finite-difference approximations. Both algorithms have very good characteristics, like fast local convergence around the solution.

In order to construct the derivative-free method to define a bordered system, the algorithm that uses left and right singular vectors of the finite-difference approximation of the Jacobian close to the solution is proposed in [23]. The algorithm was tested on two singular examples and numerical results show that all good properties are preserved. In this paper, we want to compare the features of this algorithm with the classical finite-difference method for nonlinear systems since both algorithms are completely derivative-free and can be applied to problems when the function is not given analytically.

The paper is organized in the following way. Some preliminaries are given in the second section. The third section presents the construction procedure of the bordered system using singular vectors of the Jacobian's finite-difference approximation. The derivative-free algorithm that uses only approximations of the Jacobian and Hessian both in defining and solving the bordered system is also presented. Numerical examples are given in the fourth section, where the local character of the algorithm and its very fast convergence in the neighbourhood of the solution are illustrated and compared with the finite-difference method.

2 Preliminaries

This section contains some basic definitions which are necessary.

Definition 1: [15]

Tensor $T \in \mathbb{R}^{n\times n\times n}$ is composed of n horizontal faces $H_k \in \mathbb{R}^{n\times n}$, $k = 1, \dots, n$, where $[H_k]_{i,j} = [\mathrm{T}]_{i,j,k}$.

Products between tensor $T \in \mathbb{R}^{n\times n\times n}$ and vector $v \in \mathbb{R}^n$ are defined in the following way

$Tv = ((H_1 v)^T, (H_2 v)^T, \dots, (H_n v)^T)^T \in \mathbb{R}^{n\times n}$,

$v^T T = (v^T H_1, v^T H_2, \dots, v^T H_n)^T \in \mathbb{R}^{n\times n}$.

For symmetric matrices $H_1, H_2, \dots, H_n$, the product is commutative, i.e., $Tv = v^T T$.

Let $F''(x) \in \mathbb{R}^{n\times n\times n}$ be the second G-derivative of $F(x)$ composed by n horizontal faces $H_k(x)$, where $H_k(x)$ is Hessian matrix of component $F_k(x)$, $k = 1,2, \dots, n$. Then,

$F''(x)v = ((H_1(x)v)^T, (H_2(x)v)^T, \dots, (H_n(x)\, v)^T)^T \in \mathbb{R}^{n\times n}$.

For $\xi > 0$, the finite-difference approximations of Jacobian $F'(x)$ and Hessian $H_k(x)$, denoted by $D(x, \xi)$ and $H_k(x, \xi)$, are defined by

$$[D(x,\xi)]_{i,j} = \frac{F_i\left(x+\xi e_j\right) - F_i\left(x-\xi e_j\right)}{2\xi}, \tag{2}$$

$$[H_k(x,\xi)]_{i,j} = \frac{F_k\left(x+\xi e_i+\xi e_j\right) - F_k\left(x+\xi e_i-\xi e_j\right) - F_k\left(x-\xi e_i+\xi e_j\right) + F_k\left(x-\xi e_i-\xi e_j\right)}{4\xi^2}, \tag{3}$$

for $i,j = 1,2,\dots,n$ and e_j, e_i being the $j-$th and $i-$th unit vector, respectively.

Thus, the finite-difference approximation of $F''(x) \in \mathbb{R}^{n\times n\times n}$, denoted by $F''(x,\xi) \in \mathbb{R}^{n\times n\times n}$, is composed by n horizontal faces $H_k(x,\xi), k = 1,2,\dots,n$, defined by (3).

Throughout the paper, the Euclidean norm denoted by $\|.\|$ will be used. Besides, the infinity norm, denoted by $\|.\|_{\infty,}$ will also be used in the fourth section.

3 Bordered System and Algorithm

We will assume that function F satisfies the following conditions:

A1: F is twice continuously differentiable and $F''(x)$ is Lipschitz continuous in some neighbourhood of the solution, $\mathcal{B}(x^*,\varepsilon)$.

A2: Jacobian $F'(x^*)$ has rank $r = rank\ F'(x^*) = n-q$, for $1 \le\ q \le\ n$.

A3: There exists a nonzero vector $\mu^* \in N((F'(x^*))^T)$ and a basis $\{\eta_1^*, \eta_2^*, \dots, \eta_q^*\}$ of the null space $N(F'(x^*))$ such that $q\ \times q$ matrix

$$[\eta^*]^T[(\mu^*)^T F''(x^*)]\eta^* \equiv [\eta_1^*, \eta_2^*, \dots, \eta_q^*]^T[(\mu^*)^T F''(x^*)][\eta_1^*, \eta_2^*, \dots, \eta_q^*]$$

is nonsingular.

SVD factorization gives

$$D(\bar{x},\xi) = \bar{U}(\bar{x})\bar{\Sigma}(\bar{x})\bar{V}^T(\bar{x})$$

$$= [\bar{u}_1(\bar{x}), \dots, \bar{u}_n(\bar{x})]diag\left(\bar{\sigma}_1(\bar{x}), \bar{\sigma}_2(\bar{x}), \dots, \bar{\sigma}_n(\bar{x})\right)[\bar{v}_1(\bar{x}), \dots, \bar{v}_n(\bar{x})]^T,$$

where $\bar{u}_1(\bar{x}), \dots, \bar{u}_n(\bar{x})$, $\bar{v}_1(\bar{x}), \dots, \bar{v}_n(\bar{x})$ are left and right singular vectors and $\bar{\sigma}_1(\bar{x}), \bar{\sigma}_2(\bar{x}), \dots, \bar{\sigma}_n(\bar{x})$ are corresponding singular values.

Since the singular values are evaluated in descending order

$$\bar{\sigma}_1(\bar{x}) \ge\ \bar{\sigma}_2(\bar{x}) \ge \cdots \ge\ \bar{\sigma}_n(\bar{x}) \ge\ 0$$

and singular values are well conditioned, for $\bar{x} \in\ \mathcal{B}\ (x^*,\varepsilon)$ there are q small but nonzero singular values $\bar{\sigma}_{n-q+1}(\bar{x}), \dots, \bar{\sigma}_n(\bar{x})$ of $D(\bar{x},\xi)$.

From singular vectors that correspond to small singular values $\bar{\sigma}_{r+1}(\bar{x}) \ge \cdots \ge \bar{\sigma}_n(\bar{x}) \ge 0$, matrices $\bar{R}(\bar{x}), \bar{L}(\bar{x}) \in \mathbb{R}^{n\times q}$ are formed using

$\bar{R}(\bar{x}) = [\bar{u}_{r+1}(\bar{x}), \dots, \bar{u}_n(\bar{x})]$

and

$\bar{L}(\bar{x}) = [\bar{v}_{r+1}(\bar{x}), \dots, \bar{v}_n(\bar{x})]$.

Matrix $\bar{A}_\xi(x, \bar{x}) \in \mathbb{R}^{(n+q)\times(n+q)}$ is defined using matrices $\bar{R}(\bar{x})$ and $\bar{L}(\bar{x})$

$$\bar{A}_\xi(x, \bar{x}) = \begin{bmatrix} D(x,\xi) & \bar{R}(\bar{x}) \\ \bar{L}^T(\bar{x}) & 0 \end{bmatrix}, \tag{4}$$

where $D(x, \xi)$ is an approximation of $F'(x)$ given by (2).

For $x \in \mathcal{B}(x^*, \varepsilon)$ let $\bar{\eta}(x,\xi) \in \mathbb{R}^{n\times q}$, $\bar{h}(x,\xi) \in \mathbb{R}^{q\times q}$, $\bar{\mu}(x,\xi) \in \mathbb{R}^n$ and $\bar{g}(x,\xi) \in \mathbb{R}^q$ be the solutions of the following systems

$$\bar{A}_\xi(x, \bar{x}) \begin{bmatrix} \bar{\eta}(x,\xi) \\ \bar{h}(x,\xi) \end{bmatrix} = \begin{bmatrix} 0 \\ E_q \end{bmatrix}, \tag{5}$$

$$\bar{A}_\xi{}^T(x, \bar{x}) \begin{bmatrix} \bar{\mu}(x,\xi) \\ \bar{g}(x,\xi) \end{bmatrix} = \begin{bmatrix} 0 \\ \alpha \end{bmatrix}, \tag{6}$$

where $E_q \in \mathbb{R}^{q\times q}$ is an identity matrix and $\alpha \in \mathbb{R}^q$ is a randomly chosen vector.

We expect matrix (4) to be nonsingular and obtained numerical results confirm that. This guarantees the uniqueness of the solutions of systems (5) and (6). Using $\bar{\eta}(x,\xi)$ from (5) and $\bar{\mu}(x,\xi)$ from (6), the $q \times q$ matrix

$\bar{B}_\xi(x, \alpha) \equiv \bar{\eta}^T(x,\xi)\,[\bar{\mu}^T(x,\xi)\,H(x,\xi)]\,\bar{\eta}(x,\xi)$

is formed and it will be used in the algorithm for finding the solution of the bordered system.

The bordered system is defined by

$$\boldsymbol{F}(x, \lambda) = \begin{bmatrix} F(x) + \bar{R}(\bar{x})\lambda \\ \bar{g}(x,\xi) \end{bmatrix} = 0, \tag{7}$$

where $\bar{g}(x,\xi)$ is the solution of the system (6). Vector $\lambda \in \mathbb{R}^q$ corresponds to the relaxation parameter that makes the problem more stable, [8].

The local derivative-free algorithm for solving (7), proposed in [23], is given below.

Algorithm:

Step 0: Choose $x_0 \in \mathbb{R}^n$ and a small parameter $\xi > 0$.

Step 1: Set $\bar{x} = x_0$ and $k = 0$.

Step 2: Compute $D(\bar{x}, \xi)$ using (2), evaluate its SVD, and determine the value of q to form matrices $\bar{R}(\bar{x})$ and $\bar{L}(\bar{x})$.

Step 3: Generate a random vector $\alpha \in \mathbb{R}^q$ and set $\lambda_0 = 0 \in \mathbb{R}^q$.

Repeat Step **4** – **Step 7** until convergence is obtained.

Step 4: Compute $D(x_k, \xi)$ defined by (2) and form $\bar{A}_\xi(x_k, \bar{x})$ defined by (4).

Step 5: Find solutions to the following systems

$$\bar{A}_\xi(x_k, \bar{x})\, Y = \begin{bmatrix} -(F(x_k) + \bar{R}(\bar{x})\lambda_k) \\ 0 \end{bmatrix},$$

$$\bar{A}_\xi(x_k, \bar{x}) \begin{bmatrix} \bar{\eta}\,(x_k, \xi) \\ \bar{h}\,(x_k, \xi) \end{bmatrix} = \begin{bmatrix} 0 \\ E_q \end{bmatrix},$$

$$\bar{A}_\xi{}^T(x_k, \bar{x}) \begin{bmatrix} \bar{\mu}\,(x_k, \xi) \\ \bar{g}\,(x_k, \xi) \end{bmatrix} = \begin{bmatrix} 0 \\ \alpha \end{bmatrix}.$$

Step 6: Compute the matrix

$$\bar{B}_\xi(x_k, \alpha) = \bar{\eta}^T\,(x_k, \xi)\,[\bar{\mu}^T\,(x_k, \xi)\,H(x_k, \xi)]\,\bar{\eta}\,(x_k, \xi)$$

and solve the linear system

$$\bar{B}_\xi(x_k, \alpha)\, W = \bar{g}(x_k, \xi) - \bar{\eta}^T\,(x_k, \xi)\,[\bar{\mu}^T\,(x_k, \xi)\,H(x_k, \xi)]\,Y_1.$$

Step 7: Update step

$$\begin{bmatrix} dx \\ d\lambda \end{bmatrix} = Y + \begin{bmatrix} \bar{\eta}\,(x_k, \xi) \\ \bar{h}\,(x_k, \xi) \end{bmatrix} W,$$

update iteration

$$\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} + \begin{bmatrix} dx \\ d\lambda \end{bmatrix}$$

and update k, i.e., $k = k + 1$.

Like in [21] and [22], it is sufficient to determine the solutions $Y = [Y_1 \quad Y_2]^T \in \mathbb{R}^{n+q}$, $\bar{\eta}\,(x, \xi) \in \mathbb{R}^{n \times q}$, $\bar{h}\,(x, \xi) \in \mathbb{R}^{q \times q}$, $\bar{\mu}\,(x, \xi) \in \mathbb{R}^n$ and $\bar{g}\,(x, \xi) \in \mathbb{R}^q$ of the corresponding linear systems in Step 5 and $W \in \mathbb{R}^q$ in Step 6 to update the iteration in Step 7.

In [21] and [22] the bordered system was formed using SVD factorization of $F'(\bar{x})$ for $\bar{x} \in \mathcal{B}(x^*, \varepsilon)$, more precisely using left and right singular vectors of $F'(\bar{x})$. Besides, using the Jacobian matrix $F'(\bar{x})$ in [21] and its finite-difference approximation $D(\bar{x}, \xi)$ in [22], the bordered system was constructed. Nonsingularity of the defined system was proven and fast algorithms for solving it were proposed in [21] and [22].

The local characteristic of the algorithm given in [21] was numerically tested. To achieve fast convergence, the initial iterate has to be very close to the solution, and since the solution is usually not known this was a big disadvantage of the algorithm.

The Jacobian and Hessian matrices were approximated by finite-differences in [22]. Thus, the calculations of derivatives were avoided and application of the algorithm for problems with function, which is not defined analytically is included. The local convergence of the algorithm constructed in that way was proven. To overcome the difficulty of selecting the initial approximation, the local algorithm was combined with a globally convergent descent algorithm with finite-difference approximations. In the beginning, while the current iterate is far away from the solution, the descent algorithm with finite-differences and Armijo rule was used. This prevents large steps. When the step is too small, it was assumed that the current iterate is close to the solution and a local algorithm was applied, taking the current iterate to be the initial approximation.

The algorithms presented in [22] are not completely derivative-free since SVD of the Jacobian $F'(\bar{x})$ and the corresponding singular vectors are used to define bordered system. The previously defined algorithm is constructed using singular vectors of the finite-difference approximation $D(\bar{x}, \xi)$ of Jacobian. This algorithm, proposed in [23], is completely derivative-free and its local characteristic is confirmed on some numerical examples. Furthermore, like in [22], one can exploit the fast local convergence of this algorithm by combining it with a globally convergent method, which will lead to the globally convergent derivative-free method.

In this paper, the local characteristics of the proposed derivative-free algorithm are examined in more detail and compared with the classical finite-difference method.

The rank deficiency q of the Jacobian $F'(x^*)$ is used for defining matrix (4) in Step 4 of the algorithm. The corresponding value of q is unknown and one way to determine this value, [20], is to overestimate it since it is known that in that case, any iterative sequence will not converge. The next step is to decrease the value until the convergence is obtained, but keeping in mind that convergence will occur in case the value of q is underestimated.

The SVD decomposition of $F'(\bar{x})$ implies that there are q small but nonzero singular values, so one can conclude that q is equal to the number of small singular values of finite-difference approximation $D(\bar{x}, \xi)$. Since these values are evaluated in descending order, it is easy to count them and predict the value of q.

4 Numerical Results

We tested the proposed derivative-free algorithm on some relevant examples from [21] and [22], with different starting approximations, and compared it with the classical finite-difference method. The obtained numerical results are presented in this section. We consider that the current iterate x_k generated by both algorithms is a good approximation of the solution if it satisfies $\|F(x_k)\| \leq 10^{-6}$ for the

maximum number of iterations equal to 30. The rate of convergence is determined through the quotient

$$d_k = \frac{\|x^* - x_k\|}{\|x^* - x_{k-1}\|}.$$

The Jacobian and Hessian are approximated by finite-differences with parameter $\xi = 10^{-5}$.

Example 1: Function $F: \mathbb{R}^2 \to \mathbb{R}^2$ is defined by

$$F(x_1, x_2) = \begin{bmatrix} x_1^2 - x_2 \\ x_1^2 + x_2^2 \end{bmatrix}.$$

The singular solution of equation (1) is $x^* = (0,0)^T$. The Jacobian matrix is

$$F'(x_1, x_2) = \begin{bmatrix} 2x_1 & -1 \\ 2x_1 & 2x_2 \end{bmatrix},$$

and since

$$F'(0,0) = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix},$$

the rank deficiency of the Jacobian at the solution is q=1. Moreover, the null space of the Jacobian, $N(F'(0,0))$, is generated by $\eta^* = (1,0)^T$. One characteristic of singular systems is that all components do not converge at the same rate. Slower convergence of the components that correspond to the null space also slows down the convergence of the entire iterative sequence. In this example, the first component is slower than the second one.

In order to examine a local characteristic of the derivative-free algorithm defined in the previous section, iterative sequences generated with three starting points are presented in the following tables. These iterative sequences are compared with the iterative sequences generated by the finite-difference method, using the same starting approximations.

Table 1 presents the iterative sequence generated by the finite-difference method. Since the method is applied to the singular system, slower convergence of the first component is clearly indicated. The convergence rate of the finite-difference method drops from superlinear to linear, which is indicated through the quotient d_k that converges to 0.5.

Table 1

$x_0 = (0.5,0.7)^T$

k	$[x_k]_1$	$[x_k]_2$	$\|F(x_k)\|$	d_k
1	4.541e-01	2.041e-01	2.479e-01	0.578855
2	2.596e-01	2.959e-02	7.808e-02	0.524862
3	1.314e-01	8.274e-04	2.385e-02	0.502899

4	6.572e-02	6.918e-07	6.107e-03	0.500029
5	3.285e-02	7.395e-12	1.526e-03	0.500038
6	1.642e-02	7.396e-17	3.817e-04	0.500076
7	8.219e-03	2.059e-20	9.554e-05	0.500152
8	4.112e-03	-6.512e-21	2.385e-05	0.500304
9	2.058e-03	2.636e-22	5.993e-06	0.500607
10	1.031e-03	2.635e-22	1.505e-06	0.501211
11	5.184e-04	5.187e-23	3.800e-07	0.502411

The iterative sequence generated by our algorithm is presented in Table 2. We can point out that both components of the iterative sequence converge to the solution at the same speed, which affects the convergence of the whole iterative sequence. The quotient d_k converges to 0, which means that superlinear convergence is obtained. The number of iterations needed for the convergence is smaller than in the case when the finite-difference method is applied. All of these indicate that our method is faster than the finite-difference method.

Table 2

α=8.90903, $x_0 = (0.5, 0.7)^T$

k	$[x_k]_1$	$[x_k]_2$	$\|F(x_k)\|$	d_k
1	-9.416e-02	4.026e-01	4.293e-01	0.48075
2	1.218e-02	1.450e-01	1.463e-01	0.351867
3	2.166e-03	3.244e-02	3.245e-02	0.223456
4	1.774e-04	2.473e-03	2.473e-03	0.076274
5	-1.548e-06	1.673e-05	1.673e-05	0.006775
6	8.821e-11	7.774e-10	7.774e-10	0.000046

Similar results are obtained for starting approximations being closer to the solution and generated iterative sequences are presented in Tables 3-6. Iterative sequences generated by the finite-difference method are given in Table 3 and Table 5, while Table 4 and Table 6 contain the sequences generated by our derivative-free algorithm.

Table 3

$x_0 = (0.3, 0.4)^T$

k	$[x_k]_1$	$[x_k]_2$	$\|F(x_k)\|$	d_k
1	2.981e-01	8.889e-02	9.679e-02	0.62224
2	1.603e-01	6.708e-03	3.199e-02	0.515782
3	8.030e-02	4.441e-05	9.088e-03	0.500443
4	4.015e-02	2.423e-09	2.279e-03	0.500031
5	2.008e-02	2.424e-14	5.699e-04	0.500062
6	1.004e-02	2.246e-19	1.426e-04	0.500124
7	5.023e-03	-5.733e-21	3.569e-05	0.500249

8	2.514e-03	1.042e-21	8.941e-06	0.500497
9	1.259e-03	1.958e-22	2.244e-06	0.500992
10	6.323e-04	-1.588e-23	5.654e-07	0.501977

Table 4

α=5.85264, $x_0 = (0.3, 0.4)^T$

k	$[x_k]_1$	$[x_k]_2$	$\|F(x_k)\|$	d_k
1	2.165e-02	5.342e-02	5.305e-02	0.11529
2	5.878e-04	1.651e-03	1.651e-03	0.030412
3	6.050e-07	1.714e-06	1.714e-06	0.001036
4	6.508e-13	1.852e-12	1.852e-12	0.000001

Table 5

$x_0 = (0.02, 0.02)^T$

k	$[x_k]_1$	$[x_k]_2$	$\|F(x_k)\|$	d_k
1	1.962e-02	3.848e-04	3.851e-04	0.693812
2	9.816e-03	1.518e-07	1.361e-04	0.500228
3	4.906e-03	1.541e-12	3.405e-05	0.500255
4	2.457e-03	1.540e-17	8.543e-06	0.500509
5	1.231e-03	3.353e-22	2.144e-06	0.500101
6	6.182e-04	-8.818e-23	5.404e-07	0.502022

Table 6

α=6.99077, $x_0 = (0.02, 0.02)^T$

k	$[x_k]_1$	$[x_k]_2$	$\|F(x_k)\|$	d_k
1	7.179e-04	-3.413e-04	3.418e-04	0.028106
2	-5.090e-07	-4.919e-07	4.919e-07	0.000890

All iterative sequences were generated with starting points close enough to the solution, so the convergence of both methods is obtained. Iterative sequences generated by the finite-difference method need more iterations for the convergence than our method since the system is singular and slower convergence of the first component slows down the convergence of the whole sequence. The convergence rate of sequences presented in Table 1, Table 3, and Table 5 is linear since d_k converges to 0.5.

The sequences generated by our derivative-free algorithm, given in Table 2, Table 4, and Table 6, indicate that both components converge to the solution at the same speed. The rate of the convergence is superlinear since the quotient d_k converges to 0. Faster convergence of our method is a consequence of the fact that the singular system is transformed into the regular one using the bordered system.

To illustrate the local characteristics of the algorithm in detail, we tested it with much more starting points close to the solution and compared the number of

iterations obtained by our derivative-free method with the one required for the convergence when the finite-difference method is applied. The algorithms were tested with starting approximations of the form

$x_0 = (\pm i \cdot 0.1, \pm j \cdot 0.1)^T,$

for $i, j = 0, 1, \ldots, 10$. Since starting points with $i, j = 0, 1, \ldots, 5$ are close enough to the solution, both algorithms have fast convergence for these initial points with $i, j = 0, 1, \ldots, 5$, so only these results are presented in Table 7.

The first column of

Table *7* contains the first component of the initial approximation, $[x_0]_1$, while the first row contains the second component, $[x_0]_2$. Each table field presents the number of iterations for our derivative-free algorithm and the number of iterations for the finite-difference method (given in brackets), generated with the starting point x_0.

Table 7

The number of iterations for derivative-free algorithm and (finite-difference method) with initial point x_0

	-0.5	-0.4	-0.3	-0.2	-0.1	0	0.1	0.2	0.3	0.4	0.5
-0,5	30(28)	9(13)	6(12)	6(11)	6(11)	7(11)	9(11)	16(11)	30(11)	6(11)	6(11)
-0,4	27(28)	8(13)	7(12)	7(11)	6(11)	5(10)	6(10)	6(11)	6(11)	5(11)	5(11)
-0,3	16(28)	7(14)	7(12)	6(11)	6(10)	5(10)	5(10)	5(10)	4(11)	5(11)	5(11)
-0,2	20(29)	7(14)	6(12)	6(11)	5(10)	5(9)	5(10)	4(10)	4(11)	5(11)	5(12)
-0,1	18(30)	6(15)	6(13)	5(11)	5(10)	4(8)	4(9)	4(24)	4(11)	5(11)	5(12)
0	7(4)	5(29)	4(27)	4(25)	3(23)	1(1)	3(22)	4(24)	4(25)	5(26)	5(26)
0,1	18(30)	6(15)	6(13)	5(11)	5(10)	4(8)	4(9)	4(10)	4(11)	5(12)	5(12)
0,2	21(29)	7(14)	6(12)	5(11)	5(10)	5(9)	5(10)	4(10)	4(11)	5(11)	5(12)
0,3	16(28)	7(13)	7(11)	6(11)	6(10)	5(10)	5(10)	5(10)	4(11)	5(11)	5(11)
0,4	30(28)	8(13)	7(12)	6(11)	6(11)	5(10)	6(10)	6(11)	6(11)	5(11)	5(11)
0.5	30(28)	9(13)	6(12)	6(11)	6(11)	7(11)	9(11)	15(11)	30(11)	14(11)	14(11)

Table *7* shows that our derivative-free method is much faster than the finite-difference method for almost all starting points.

Example 2: Function $F: \mathbb{R}^3 \to \mathbb{R}^3$ is defined by

$$F(x_1, x_2, x_3) = \begin{bmatrix} x_1^3 + x_1 x_2 \\ x_2 + x_2^2 \\ x_1^2 + x_3^2 \end{bmatrix}$$

with $x^* = (0,0,0)^T$ being a singular solution of equation (1) and Jacobian matrix

$$F'(x_1, x_2, x_3) = \begin{bmatrix} 3x_1^2 + x_2 & x_1 & 0 \\ 0 & 1 + 2x_2 & 0 \\ 2x_1 & 0 & 2x_3 \end{bmatrix}.$$

Null space of the Jacobian $F'(0,0,0)$ is generated by vectors $\{(1,0,0)^T, (0,0,1)^T\}$. Since

$$F'(0,0,0) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

the rank deficiency is $q = 2$, so we tested our algorithm for $q = 2$ and $q = 1$.

Iterative sequences generated by our method with starting approximation $x_0 = (0.2, 0.5, 0.7)^T$ and two values of q are presented in Table 8 and Table 9.

Table 8

q=2, α=$(9.59492, 6.55741)^T$, $x_0 = (0.2, 0.5, 0.7)^T$

k	$[x_k]_1$	$[x_k]_2$	$[x_k]_3$	$\|F(x_k)\|$	d_k
1	6.066e-03	1.335e-01	2.370e-04	1.513e-01	0.151379
2	3.544e-03	1.409e-02	-1.192e-09	1.429e-02	0.108739
3	4.194e-05	1.987e-04	-1.036e-08	1.988e-04	0.013975
4	9.338e-09	4.045e-08	-1.242e-10	4.045e-08	2.043e-04

The predicted value $q = 2$ is given in Table 8. The first three columns indicate that all components of generated sequence converge to the solution at the same speed and the convergence is fast. The quotient d_k converges to 0 indicating that the convergence rate is superlinear, as it is the rate of convergence of the finite-difference method in regular case.

In Table 9 the predicted value is $q = 1$ and it does not coincide with the real value. The bordered system is formed but it is still singular. Since the first and the third components correspond to null space, the convergence of these components is visibly slower than the convergence of the second component, and this is the characteristic of a singular system. Linear convergence indicated with quotient d_k is the rate of convergence of the finite-difference method in a singular case.

Table 9

q=1, α=0.357117, $x_0 = (0.2, 0.5, 0.7)^T$

k	$[x_k]_1$	$[x_k]_2$	$[x_k]_3$	$\|F(x_k)\|$	d_k
1	-6.366e-02	1.320e-01	4.096e-01	2.279e-01	0.492652
2	-5.903e-03	1.317e-02	2.076e-01	4.515e-02	0.478336
3	-1.196e-04	1.609e-04	1.038e-01	1.078e-02	0.499062
4	-4.087e-05	2.450e-08	5.193e-02	2.697e-03	0.499999
5	-2.241e-05	-5.175e-14	2.596e-02	6.742e-04	0.5
6	-1.274e-05	-4.681e-16	1.298e-02	1.685e-04	0.5
7	-7.322e-06	-1.222e-16	6.491e-03	4.214e-05	0.5
8	-3.877e-06	-5.539e-17	3.245e-03	1.053e-05	0.5
9	-1.241e-06	-3.548e-17	1.622e-03	1.633e-06	0.5
10	1.424e-06	-3.032e-17	8.114e-04	6.584e-07	0.5

Similar results are obtained when our algorithm is applied with starting points $x_0 = (0.1, 0.3, 0.5)^T$ and $x_0 = (0.05, 0.05, 0.05)^T$. Generated sequences with $q = 2$ are shown in Table 10 and Table 12, while Table 11 and Table 13 present sequences with $q = 1$.

Table 10

q=2, α=$(7.43132, 3.92227)^T$, $x_0 = (0.1, 0.3, 0.5)^T$

k	$[x_k]_1$	$[x_k]_2$	$[x_k]_3$	$\|F(x_k)\|$	d_k
1	7.221e-03	5.768e-02	1.747e-05	6.100e-02	0.0982596
2	6.399e-04	3.004e-03	-1.095e-08	3.013e-03	0.0528371
3	1.772e-06	9.095e-06	-1.073e-09	9.095e-06	0.0030170
4	4.958e-11	8.379e-11	-2.982e-12	8.379e-11	1.051e-05

Table 11

q=1, α=6.55478, $x_0 = (0.1, 0.3, 0.5)^T$

k	$[x_k]_1$	$[x_k]_2$	$[x_k]_3$	$\|F(x_k)\|$	d_k
1	-8.594e-02	5.847e-02	2.810e-01	1.064e-01	0.506488
2	-3.820e-02	2.887e-03	1.416e-01	2.170e-02	0.489566
3	-1.912e-02	2.929e-06	7.077e-02	5.374e-03	0.499767
4	-4.790e-03	-2.292e-07	3.538e-02	1.343e-03	0.499977
5	-4.790e-03	-2.843e-08	1.768e-02	3.358e-04	0.499989
6	-2.398e-03	-3.563e-09	8.844e-03	8.396e-05	0.499994
7	-1.201e-03	-4.460e-10	4.421e-03	2.099e-05	0.499997
8	-6.035e-04	-5.579e-11	2.209e-03	5.247e-06	0.499999
9	-3.043e-04	-6.978e-12	1.104e-03	1.311e-06	0.500001
10	-1.547e-04	-8.734e-13	5.514e-05	3.280e-07	0.500005

Table 12

q=2, α=$(1.71187, 7.06046)^T$, $x_0 = (0.05, 0.05, 0.05)^T$

k	$[x_k]_1$	$[x_k]_2$	$[x_k]_3$	$\|F(x_k)\|$	d_k
1	-5.553e-03	2.396e-03	-5.114e-06	2.402e-03	0.0698435
2	-7.163e-05	5.112e-06	9.571e-08	5.112e-06	0.0118738
3	-1.248e-08	1.054e-11	1.250e-09	1.054e-11	0.0001747

Table 13

q=1, α=0.318328, $x_0 = (0.05, 0.05, 0.05)^T$

k	$[x_k]_1$	$[x_k]_2$	$[x_k]_3$	$\|F(x_k)\|$	d_k
1	-3.161e-02	2.306e-03	2.092e-02	2.724e-03	0.438628
2	-1.649e-02	7.491e-06	9.792e-03	3.679e-04	0.504906
3	8.403e-03	9.893e-08	4.665e-03	9.238e-05	0.50113
4	4.232e-03	1.321e-08	2.286e-03	2.314e-05	0.500522
5	2.125e-03	1.705e-09	1.129e-03	5.791e-06	0.500263

6	1.066e-03	2.161e-10	5.586e-04	1.448e-06	0.500136
7	5.354e-04	2.722e-11	2.750e-04	3.623e-07	0.500081

When the rank deficiency is underestimated, the bordered system is still singular and its characteristics can be recognized in Table 11 and Table 13. The quotient d_k converges to 0.5, which guarantees linear convergence and convergence of the first and the third components are slower than the second one.

Faster convergence is obtained when the predicted value is equal to the real value of rank deficiency, i.e., $q = 2$, which can be seen from Table 10 and Table 12. All components converge to the solution at the same speed and the convergence rate of the whole sequence is superlinear.

Table 14 indicates linear convergence of the finite-difference method and presents the obtained results with starting approximations x_0 and the number of iterations, k, needed to achieve a good approximation of the solution. The norm of the function value in the last generated iteration, denoted by $\|F(x_k)\|$, and the quotient d_k are also presented in **Hiba! A hivatkozási forrás nem található.**. Compared to the results presented in Table 8, Table 10 and Table 12, the finite-difference method requires more iterations to achieve convergence than our derivative-free algorithm. That was expected since for $q = 2$ the iterative sequences in Table 8, Table 10 and Table 12 converge to the regular solution of the bordered system. Besides, compared to the results given in Table 9, Table 11 and Table 13, the finite-difference method is also slower than our derivative-free method, although for $q = 1$ both systems are singular.

Table 14

x_0	k	$[x_k]_1$	$[x_k]_2$	$[x_k]_3$	$\|F(x_k)\|$	d_k
$(0.2,0.5,0.7)^T$	15	6.973e-04	2.486e-63	-1.709e-04	5.155e-07	0.632681
$(0.1,0.3,0.5)^T$	15	3.993e-04	1.459e-65	-4.328e-04	3.468e-07	0.509358
$(0.05,0.05,0.05)^T$	11	9.101e-04	7.834e-51	-3.55e-04	9.543e-07	0.533782

Example 3: Function $F: \mathbb{R}^4 \to \mathbb{R}^4$ is defined by

$$F(x_1, x_2, x_3, x_4) = \begin{bmatrix} x_1 + x_1x_2 + x_2^2 \\ x_1^2 - 2x_1 + x_2^2 \\ x_1 + x_3^2 \\ x_1^2 + x_4^2 \end{bmatrix}.$$

The solution of equation (1) is $x^* = (0,0,0,0)^T$ and the Jacobian matrix is

$$F'(x_1, x_2, x_3, x_4) = \begin{bmatrix} 1 + x_2 & x_1 + 2x_2 & 0 & 0 \\ 2x_1 - 2 & 2x_2 & 0 & 0 \\ 1 & 0 & 2x_3 & 0 \\ 2x_1 & 0 & 0 & 2x_4 \end{bmatrix}.$$

Null space of the Jacobian

$$F'(0,0,0,0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

is generated by the set of vectors $\{(0,1,0,0)^T, (0,0,1,0)^T, (0,0,0,1)^T\}$. The rank deficiency is $q = 3$, so we tested our algorithm with $q = 3, q = 2$, and $q = 1$. The obtained results are given in the following tables.

Table 15show the components of the solution approximation that satisfies the convergence criterion generated by our algorithm with three different starting approximations x_0. The number of iterations required to obtain this approximation is denoted by *k*, while q is the predicted value of rank deficiency.

Table 15

$x_0 = (0.4, 0.6, 0.6, 0.6)^T$

q	k	$[x_k]_1$	$[x_k]_2$	$[x_k]_3$	$[x_k]_4$	$\|F(x_k)\|$	d_k
1	10	1.219e-13	1.710e-03	2.480e-12	-1.604e-03	4.874e-06	0.5
2	10	-6.392e-23	8.271e-04	8.538e-19	-1.802e-18	9.675e-07	0.5
3	4	1.892e-11	-1.545e-10	3.738e-11	6.029e-10	4.635e-11	1.315e-05

Table 16

$x_0 = (0.3, 0.2, 0.2, 0.2)^T$

q	k	$[x_k]_1$	$[x_k]_2$	$[x_k]_3$	$[x_k]_4$	$\|F(x_k)\|$	d_k
1	9	1.237e-21	7.211e-04	-1.897e-18	6.353e-04	8.389e-07	0.5
2	9	-9.578e-23	7.160e-04	-8.334e-19	8.653e-18	8.389e-07	0.5
3	3	3.685e-11	1.865e-10	2.176e-10	-2.361e-11	9.026e-11	6.221e-06

Table 17

$x_0 = (0.2, 0.05, 0.05, 0.05)^T$

q	k	$[x_k]_1$	$[x_k]_2$	$[x_k]_3$	$[x_k]_4$	$\|F(x_k)\|$	d_k
1	10	-3.066e-24	1.781e-04	-3.252e-19	7.493e-04	5.633e-07	0.5
2	8	3.008e-23	7.152e-04	-2.302e-16	1.790e-16	7.235e-07	0.5
3	3	-1.435e-10	1.282e-10	1.424e-10	-1.173e-12	3.516e-10	8.405e-06

It is clear from the tables above that underestimating the value of q results in the singularity of the bordered system, so the rate of convergence is not the same for all components of the iterative sequence. If the predicted value is $q = 1$ then the second component $[x_k]_2$ and the fourth component $[x_k]_4$ converge slower than the first and third components. Assuming $q = 2$, the convergence of the second component, $[x_k]_2$, is slower than others. In both cases, the slower convergence of mentioned components affects the linear convergence of the whole iterative sequence, indicated by the quotient d_k. When the predicted value is equal to the real value of rank deficiency, i.e., $q = 3$, the superlinear convergence is obtained and all components converge to the solution with the same speed.

The results obtained using the finite-difference method for three starting approximations x_0 are presented in Table 18. The linear convergence resulting from the singularity of the problem in all three cases is indicated by the quotient d_k, as well as the slower convergence of the second, third and fourth components of the iterative sequence. All of these point out that our method is faster than the finite-difference method.

Table 18

x_0	k	$[x_k]_1$	$[x_k]_2$	$[x_k]_3$	$[x_k]_4$	$\|F(x_k)\|$	d_k
$(0.4, 0.6, 0.6, 0.6)^T$	11	-2.671e-23	3.9e-04	3.28e-04	4.48e-04	3.142e-07	0.50316
$(0.3, 0.2, 0.2, 0.2)^T$	10	3.090e-23	3.8e-04	3.87e-04	7.47e-04	6.144e-07	0.502216
$(0.2, 0.05, 0.05, 0.05)^T$	11	1.718e-24	1.07e-04	2.23e-04	4.97e-04	2.530e-07	0.503339

Furthermore, we compared residuals $\|x_k - x^*\|_\infty$ gained with our algorithm and $\|x_k - x^*\|_\infty^{fd}$ when the finite-difference method is applied on a singular system, where x_k is the approximation of the solution that satisfies the stopping criterion. These results are presented in the following table.

Table 19

The residuals

ex	x_0	q	$\|x_k - x^*\|_\infty$	$\|x_k - x^*\|_\infty^{fd}$
1	$(0.5,0.7)^T$	1	7.774e-10	5.183e-04
	$(0.3,0.4)^T$	1	1.852e-12	6.323e-04
	$(0.02,0.02)^T$	1	5.090e-07	6.182e-04
2	$(0.2, 0.5,0.7)^T$	1	8.114e-04	6.973e-04
		2	4.045e-08	
	$(0.1, 0.3,0.5)^T$	1	5.514e-04	4.328e-04
		2	8.379e-11	
	$(0.05, 0.05,0.05)^T$	1	5.354e-04	9.101e-04
		2	1.248e-08	
3	$(0.4, 0.6, 0.6, 0.6)^T$	1	4.277e-04	4.487e-04
		2	8.271e-04	
		3	6.029e-10	
	$(0.3, 0.2, 0.2, 0.2)^T$	1	7.211e-04	7.479e-04
		2	7.160e-04	
		3	2.176e-10	
	$(0.2, 0.05, 0.05, 0.05$	1	7.493e-04	4.975e-04
		2	7.152e-04	
		3	1.435e-10	

Table 19 points out that our derivative-free algorithm is more accurate since the residuals are much smaller than the residuals generated by the finite-difference method. This is the consequence of the regularity of the bordered system.

Moreover, the slower convergence of the components that correspond to the null space affects the slower convergence of the whole iterative sequence obtained by the finite-difference method.

Considering the obtained numerical results, we can conclude that the proposed derivative-free algorithm is faster than the classical finite-difference method. It has better characteristics both in the case when the predicted value of q corresponds to the real value and in the case when the value of q is underestimated. The number of iterations needed for the convergence is smaller when our method is applied and the obtained approximation of the solution is more accurate than the one obtained by the finite-difference method.

Conclusions

A local derivative-free algorithm for solving bordered system and finding a singular solution is compared with the classical finite-difference method for nonlinear systems. It is numerically shown in several examples that left and right singular vectors of the finite-difference approximation of the Jacobian can be used to define the bordered system. Numerical experiments indicate fast local convergence of the proposed method. Moreover, the obtained numerical results point out that our derivative-free method is promising, successful, and has better performances than the classical finite-difference method. Theoretical convergence of the proposed algorithm will be considered in further research.

Acknowledgement

The authors are partially supported by Chair of Mathematics, Department of Fundamental Disciplines, Faculty of Technical Sciences, through the Project ”Teorijska i primenjena matematika u tehničkim i informatičkim naukama” and Project 142-451-2102/2019 of the Provincial Secretariat for Higher Education and Scientific Research.

References

[1] G. Reddien, “On Newton's method for singular problems,” *SIAM Journal on Numerical Analysis,* Vol. 15, No. 5, pp. 993-996, 1978

[2] D. Decker and C. Kelley, “Newton's method at singular points. I,” *SIAM Journal on Numerical Analysis,* Vol. 17, No. 1, pp. 66-70, 1980

[3] D. Decker and C. Kelley, “Newton's method at singular points. II,” *SIAM Journal on Numerical Analysis,* Vol. 17, No. 3, pp. 465-471, 1980

[4] D. Decker, H. Keller and C. Kelley, “Convergence rates for Newton's method at singular points,” *SIAM Journal on Numerical Analysis,* Vol. 20, No. 2, pp. 296-314, 1983

[5] D. Decker and C. Kelley, “Sublinear convergence of the chord method at singular point,” *Numerische Mathematik,* Vol. 42, pp. 147-154, 1983

[6] D. Decker and C. Kelley, "Broyden's method for a class of problems having singular Jacobian at the root," *SIAM Journal on Numerical Analysis,* Vol. 22, No. 3, pp. 566-573, 1985

[7] C. Kelley and Z. Xue, "Inexact Newton methods for singular problems," *Optimization Methods and Software,* Vol. 2, No. 3-4, pp. 249-267, 1993

[8] A. Griewank, "On solving nonlinear equations with simple singularities or nearly singular solutions," *SIAM Review,* Vol. 27, No. 4, pp. 537-563, 1985

[9] C. Kelley and R. Suresh, "A new acceleration method for Newton's method at singular points," *SIAM Journal on Numerical Analysis,* Vol. 20, No. 5, pp. 1001-1009, 1983

[10] C. Kelley, "A Shamanskii-like acceleration scheme for nonlinear equations at singular point," *Mathematics of Computation,* Vol. 47, pp. 609-623, 1986

[11] S. Buhmiler, N. Krejić and Z. Lužanin, "Practical Quasi-Newton algorithms for singular nonlinear systems," *Numerical Algorithms,* Vol. 55, pp. 481-502, 2010

[12] S. Thomas, "Sequential Estimation Techniques for Quasi-Newton Algorithms," Cornell University, Ithaca, 1975

[13] G. Broyden, "A class of methods for solving nonlinear simultaneous equations," *Mathematics of Computation,* Vol. 19, pp. 577-593, 1965

[14] J. Martínez, "A quasi-Newton method with modification of one column per iteration," *Computing,* Vol. 3, pp. 353-362, 1984

[15] R. Schnabel and P. Frank, "Tensor methods for nonlinear equations," *SIAM Journal on Numerical Analysis,* Vol. 21, No. 5, pp. 815-843, 1984

[16] A. Bouaricha and R. Schnabel, "Tensor methods for large, sparse nonlinear least squares problems," *SIAM Journal on Scientific Computing,* Vol. 21, pp. 1199-1221, 1999

[17] D. Feng, P. Frank and R. Schnabel, "Local convergence analysis off tensor methods for nonlinear equations," *Mathematical Programming,* Vol. 62, pp. 427-459, 1993

[18] H. Weber and W. Werner, "On the Accurate Determination of Nonisolated Solutions of Nonlinear Equations," *Computing,* Vol. 26, pp. 315-326, 1981

[19] A. Griewank and G. Reddien, "Characterization and computation of generalized turning points," *SIAM Journal on Numerical Analysis,* Vol. 21, No. 1, pp. 176-185, 1984

[20] P. Rabier and G. Reddien, "Characterization and computation of singular points with maximum rank deficiency," *SIAM Journal on Numerical Analysis,* Vol. 23, No. 5, pp. 1040-1051, 1986

[21] Y. Shen and T. Ypma, "Newton's method for singular nonlinear equations using approximate left and right nullspaces of the Jacobian," *Applied Numerical Mathematics,* Vol. 54, pp. 256-265, 2005

[22] S. Buhmiler, S. Rapajić, S. Medić and T. Grbić, "Finite-difference method for singular nonlinear systems," *Numerical algorithms,* Vol. 79, No. 1, pp. 65-86, 2018

[23] S. Buhmiler, S. Rapajić, M. Lukić, S. Medić, N. Duraković and T. Grbić, "Derivative-free method for singular systems," in *SISY 2019*, Subotica, 2019