A COMPARATIVE STUDY OF DIAGONAL UPDATING NEWTON METHODS FOR SYSTEMS OF NONLINEAR EQUATIONS WITH SINGULAR JACOBIAN

Mohammed waziri Yusuf[1] and Ibrahim Saidu[2]
[1]Faculty of Science, Department of Mathematics, University Putra Malaysia, Malaysia
[2]Faculty of Computer Science and Information Technology University Putra, Malaysia
E-Mail: waziri@math.upm.edu.my

ABSTRACT

We have compared the efficiency of two diagonal updating Newton methods for solving systems of nonlinear equations with singular Jacobian. Due to the fact that, the quadratic rate of convergence of Newton method for solving nonlinear systems of equations depends on when the Jacobian is nonsingular in the neighborhood of the solution. Contrary to this condition, i.e. the Jacobian to be singular the convergence is very slow and may even vanished. The two approaches are simple and straight forward to implement. We report on numerous numerical experiments which show that, the proposed algorithms are very promising.

Keywords: diagonal, Newton method, and Jacobian.

1. INTRODUCTION

Consider the system of nonlinear equations

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

Where $F:\mathbb{R}^n \to \mathbb{R}^n$ with the following properties:

1. There exist x^* with $F(x^*)=0$
2. F is continuously differentiable in a neighborhood f x^*.
3. $F'(x^*)=J_F\left(x^*\right)\neq 0$

The most renowned method for solving (1.1) is Newton's method. Moreover, the Newton's method for nonlinear equations it has the following general form: Given an initial point, say $x_0\in\Re^n$ it generates a sequence of correction $\{s_k\}$ and iterates $\{x_k\}$ [1], according to following stages:

Algorithm NM *(*Newton's method*)*

Step 1: Solve $J_F\left(x_k\right)s_k=-F(x_k)$

Step 2: Update $x_{k+1}=x_k+s_k$

Step 3: Repeat 1-2 until converges.

Where $k=0,1,2,...$ and $J_F\left(x_k\right)$ is the Jacobian matrix of F. When the Jacobian matrix is nonsingular at the neighborhood of the solution, the convergence is guaranteed and the rate is quadratic [Yun and Tijalling, 2005]. Newton method i.e.

$$\left\|x^{k+1}-x^*\right\|\leq h\left\|x^k-x^*\right\|^2 \tag{1.2}$$

For some h, $k=0,1,2......$

Violating this condition, i.e. when the Jacobian is invertible at the solution the convergence is unsatisfactory and may even diminish [Yun and Tijalling, 2005]. In addition Newton method slows down when approaching a singular root, this may vanishes the possibility of convergence to the solution x^* [Eulalia and Juan, 2009]. This requirement of non singularity of the Jacobian restricts to some level the application of Newton method for solving nonlinear systems of any category [Yun and Tijalling, 2005]

There are a number of modifications to circumvent the point in which the Jacobian is singular, but there converges rate turn to be slow, resulting from less Jacobian information at each iteration [Waziri *et al.*, 2010]. The trouble-free modification is fixed Newton method. The modification is simply to implement as by setting $J_F\left(x_k\right)\equiv J_F\left(x_0\right)$ for $k>0$. Fixed Newton method generates iterative points via the following stages:

Algorithm FN (fixed Newton)

Given x_0

Step1: Solve $J_F\left(x_0\right)s_k=-F(x_k)$

Step2: Set $x_{k+1}=x_k+s_k$, for $k=0,1,2......$.

FN method has eliminated both the computation and storage of the Jacobian (except at k=0) as well as avoiding solving n linear system in each iteration but is significantly slower [Natasa and Zoma, 2001]. Inexact Newton method is also another Newton-type method for eliminating the shortcomings of Newton method for solving nonlinear equations. This method avoids solving Newton equation (stage1 of algorithm NM) by taking the correction $\{s_k\}$ satisfying $r_k=J_F\left(x_0\right)s_k+F(x_k)$ [3]. Inexact Newton method is given by the following stages:

Algorithm INM (Inexact Newton)

Let x_0 be given

Step1: Find some s_k which satisfies

$$J_F(x_k)s_k = -F(x_k) + r_k$$

Where

$$\|r_k\| \le \eta_k \|F(x_k)\|.$$

Step2: set $x_{k+1} = x_k + s_k$.

Where η_k is a forcing sequence. Letting $\eta_k \equiv 0$ it gives back to Newton method. Another modification is Quasi-Newton's method, the method is the famous method that replaces derivatives computation with direct function computation and also replaces Jacobian or its inverse with an approximation which can be updated at each iteration [Broyden, 1965]

In this paper we compared the efficiency of two diagonally updating Newton method for solving systems of nonlinear equations. These methods are presented by [Waziri *et al*., 2010a] and [Waziri *et al.*, 2010b]. In [Waziri *et al*., 2010a] the approximation is on the Jacobian into nonsingular diagonal matrix. Whereby in [Waziri *et al*., 2010b] the approximation is on Jacobian inverse without the cost of computing and storing the Jacobian in every iteration. Our main emphasis in this paper is to compare the CPU time, number of iterations, matrix storage requirement and robust index of the two newly modified Newton methods for solving nonlinear equations when the Jacobian is singular at the solution with some variant of Newton methods. The diagonal updating methods study in this work has a very simple form, which will be favorable to making code and is significantly cheaper than Newton method as well as faster than both Newton and fixed Newton methods in term of CPU time and number of iterations, [Waziri *et al*., 2010b]. The rest of this work is organized as follows: we present the background of the two diagonally updating scheme in Section 2 and 3. Some numerical results and analysis are reported in section 4. Finally Discussion are presented in section 5.

2. DIAGONAL JACOBIAN APPROXIMATION

In this section we shall consider the approximation to the Jacobian into nonsingular matrix proposed by [Waziri *et al.*, 2010a]. The authors presented the approximation via Taylor series expansion of $F(x)$ about x_k , i.e.

$$F(x) = F(x_k) + F'(x_k)(x - x_k) + O(\|x - x_k\|^2). \quad (1.3)$$

Through imposing some conditions on the incomplete Taylor series expansion of $F(x)$ they were able to approximate $D_k \approx F'(x_k)$ as:

$$D = diag(d_1, d_2, \ldots, d_n) \quad (2.1)$$

Where d_i is

$$d_i = \frac{F_i(x_{k+1}) - F_i(x_k)}{(x^i_{k+1} - x^i_k)} \text{ for } i = 1, 2, \ldots ,$$

If only the denominator is not equal to zero (see [Waziri *et al*., 2010a] for details).

The algorithms and the update formula for the new approximation to the Jacobian into nonsingular diagonal matrix, is given as:

The update formula and the algorithm are given as (DJAN) [Waziri *et al*., 2010a]:

$$x_{k+1} = x_k - D^{-1}F(x_k) \quad (2.2)$$

Where D_k defined by (2.1), provided $|x^i_{k+1} - x^i_k| > 10^{-8}$ else set $d^i = d^{i-1}$ for $k = 0, 1, 2, \ldots, n.$

Algorithm DJAN [Waziri *et al*., 2010a]

Consider $F : \mathbb{R}^n \to \mathbb{R}^n$ with the same property as (1.1)

Step 1: Given x_0, ε *and* $D_0 = I_n$, set k=0

Step 2: Compute $x_{k+1} = x_k - D_k^{-1}F(x_k)$ *and* $F(x_k)$

Where D_k defined by (2.1),

Step 3: If $\|x_{k+1} - x_k\| + \|F(x)\| \le 10^{-8}$ stop. Else go to step 4.

Step4: If $\|x_{k+1} - x_k\| + \|F(x)\| \le 10^{-8}$ compute D_{k+1} , else $D_{k+1} = D_k$. Set $K = k + 1$ and go to step 2.

The above algorithm is faster than Newton method and fixed Newton method so also cheaper than Newton methods.

3. DIAGONAL JACOBIAN INVERSE APPROXIMATION

In this section we consider Jacobian inverse approximation into nonsingular diagonal matrix proposed by [Waziri *et al.*, 2010b]. The advantage of this approach over the above updating scheme is that, no cost of storing or computation of the Jacobian matrix required in the implementation of the scheme. Indeed this gives it an improvement in terms of execution time (CPU time).

The authors designed this diagonal updating method in the same fashion as Jacobian approximation (see [Waziri *et al*., 2010b] for details).

They presented the approximation via Taylor series expansion of $F(x)$ about x_k , i.e.

$$F(x) = F(x_k) + F'(x_k)(x - x_k) + O(\|x - x_k\|^2).$$

By imposing some conditions on the incomplete Taylor series expansion of $F(x)$ they were able to approximate

$D_k \approx F'(x_k)^{-1}$ as:

$$D = diag(d_1, d_2, \ldots, d_n) \quad (3.1)$$

Where

$d_i = \frac{(x^i_{k+1} - x_k^{\ i})}{F_i(x_{k+1}) - F_i(x_k)}$ For $i = 1, 2, \ldots$, if only the denominator is not equal to zero.

The algorithms and the update formula for the new approximation to the Jacobian inverse into nonsingular diagonal matrix, is given as follows, [Waziri *et al*., 2010b]:

The update formula is given as (DJIAN):

$$x_{k+1} = x_k - DF(x_k) \qquad (3.2)$$

Where D_k defined by (3.1), provided $\left|F_i(x_{k+1}) - F_i(x_k)\right| > 10^{-8}$ else set $d^i = d^{i-1}$ for $k = 0, 1, 2, \ldots, n.$

Algorithm DJIAN [Waziri *et al*., 2010b]

Consider $F : \mathbb{R}^n \to \mathbb{R}^n$ with the same property as (1.1)

Step 1: Given x_0, ε *and* $D_0 = I_n$, set k=0

Step 2: Compute $x_{k+1} = x_k - D_k F(x_k)$ *and* $F(x_k)$

Where D_k defined by (2.1),

Step 3: If $\|x_{k+1} - x_k\| + \|F(x)\| \le 10^{-8}$ stop. Else go to step 4.

Step 4: If $\|x_{k+1} - x_k\| + \|F(x)\| \le 10^{-8}$ compute D_{k+1}, else $D_{k+1} = D_k$. Set $K = k+1$ and go to step 2.

The above algorithm is also considerable faster than Newton method and fixed Newton method so also cheaper than Newton methods. Despite the fact that DJAN is an approximation to the Jacobian DJIAN outperforms it, due to low computational cost and storage requirement associated to the building of the approximation matrix in DJIAN.

To this end we next compare the numerical performance of the DJAN and DJIAN methods with some variant of Newton method.

Our main aims here, is to locate between the two newly approximation which is reliable and efficient, in terms of CPU time, number of iterations and matrix storage requirements.

4. NUMERICAL RESULTS AND COMPARISON

In this section, we present numerous numerical tests to exemplify the comparative study of DJAN and DJIAN for solving nonlinear systems of equations with singular Jacobian at a solution. The comparison was based on number of iterations and CPU time in seconds and matrix storage requirements. The methods used are namely:

a) DJAN stands for Diagonal Jacobian approximation.
b) DJIAN stands for Diagonal Jacobian inverse approximation.
c) NM: Newton method.
d) FN: Fixed Newton Method.

DJAN is proposed by [Waziri *et al*., 2010a] and [Waziri *et al*., 2010b] proposed DJIAN

The numerical experiments were accomplished using MATLAB 7.0. All the calculations were carried out in double precision computer. We used the stopping criterion:

$$\|x_{k+1} - x_k\| + \|F(x)\| \le 10^{-8} \qquad (4.1)$$

We used the following notations: Cpu: Cpu time in seconds. Lastly we present some details of the used benchmarks test problems as follows:

Problem 1 [Eulalia and Juan, 2009]

$f : R^2 \to R^2$ is defined by

$$f(x) = \begin{cases} (x_1-1)^2(x_1-x_2) \\ (x_1-2)^2 \cos\left(\frac{2x_1}{x_2}\right) \end{cases}$$

$x_0 = (1.5, 2.5)$ and $(0.5, 1.5)$ are chosen the solution is $x^* = (1, 2)$.

Problem 2 [Eulalia and Juan, 2009]

$f : R^3 \to R^3$ is defined by

$$f(x) = \begin{cases} (x_1-1)^4 e^{x_2} \\ (x_2-2)^5 (x_1x_2-1) \\ (x_3+4)^6 \end{cases}$$

$x_0 = (2, 1, -2)$ is chosen and $x^* = (1, 2, -4)$

Problem 3 [Waziri *et al*., 2010a]

$f : R^2 \to R^2$ is defined by

$$f(x) = \begin{cases} e^{x_1} - 1 \\ e^{x_2} - 1 \end{cases}$$

$x_0 = (0.5, 0.5), (-1.5, -1.5)$ Are chosen the solution is $x^* = (0, 0)$.

Problem 4 [Waziri *et al*., 2010a]

$f : R^2 \to R^2$ is defined by:

$$f(x) = \begin{cases} 5x_1^2 + \cos(x_1)x_2^2 \\ x_1^2 \cos(x_1 e^{x_2}) + 3x_2 \end{cases}$$

$x_0 = (0.2, -0.1)$ *and* $x^* = (0, 0)$

Problem 5 [Eulalia and Juan, 2009]

$f : R^2 \to R^2$ is defined by

$$f(x) = \begin{cases} (6x_1 - x_2)^4 \\ \cos(x_1) - 1 + x_2 \end{cases}$$

$x_0 = (-0.5, 0.5)$ and $x^* = (0,0)$

Problem 6 [Yun and Tijalling, 2005]

$f : R^2 \rightarrow R^2$ is defined by

$$f(x) = \begin{cases} x_1^2 + x_2^2 \\ x_1^2 + 3x_2 \end{cases}$$

$x_0 = (0.5, -0.3)$, $(0.5, 0.5)$ are chose and $x^* = (0,0)$

Problem 7 [Waziri *et al.*, 2010b]

$f : R^2 \rightarrow R^2$ is defined by:

$$f(x) = \begin{cases} e^{x_1} - x_2 - 1 \\ x_1 - x_2 \end{cases}$$

$x_0 = (0.7, 0.7)$ is chosen the solution is $x^* = (0,0)$.

Table-1. The numerical results of problems 1-7: Number of iterations.

Problems	x_0	NM	FN	DJAN	DJIAN
1	1.5,2.5) (0.5,1.5)	108 33	133 71	9 11	9 9
2	(2,1,-2)	143	_	37	35
3	(0.5,0.5) (-1.5,-1.5)	4 7	9 _	4 7	4 6
4	0.2,-0.1	12	85	10	7
5	(-0.5, 0.5) (0.5,0.5)	27 26	_ _	25 25	25 23
6	(0.5,-0.3) (-0.5,-0.4)	13 _	144 _	8 5	7 5
7	(0.7,0.7)	28	128	16	13

Table-2. The numerical results of problems 1-7: CPU time in seconds.

Pm	x_0	NM	FN	DJAN	DJIAN
1	(1.5,2.5) (0.5,1.5)	90.4503 28.6261	68.1734 47.0925	6.5312 5.9521	5.2416 4.6956
2	(2,1,-2)	267.8698	_	41.046	30.4045
3	(0.5,0.5) (-1.5,-1.5)	3.8064 5.5224	5.1012 _	2.5625 4.7145	1.9812 3.6831
4	(0.2,-0.1)	19.0298	46.0203	9.0793	4.900
5	(-0.5, 0.5) (0.5,0.5)	21.3877 20.4517	_ _	18.2053 19.1845	13.0573 14.2585
6	(0.5,-0.3) (-0.5,-0.4)	10.0152 _	59.1244 _	6.1409 4.0248	3.6504 3.5384
7	(0.7,0.7)	20.7562	61.0040	4.9184	3.4792

The numerical results of the four (4) methods are reported in Tables 1 and 2, which includes number of iterations and CPU time in seconds respectively. Through examination of these Tables, we can easily observe that all the four methods have shown a good attempt to solve the systems, except FN method. The performance of DJAN method is considerable better than NM and FN in terms of number of iterations and CPU time, as can easily be seen that, it has least number of iterations and less CPU time.

More so the result of DJIAN revealed that the method is quite cheaper than DJAN, this is due to the facts that, the method do not require computation and storage of the Jacobian in each iteration In particular, the DJAN and DJIAN methods outperformance the NM and FN methods for all the tested benchmarks problems with their respective initial guesses. In addition, we observe that DJAN and DJIAN methods are the best with 100% of successes when compared with NM method having 90.9% and FN method with 54.54% respectively. It is worth mentioning that the DJIAN has total eliminates the need of Jacobian matrix storage, whereby DJAN method has reduces to vector storage, respectively.

5. CONCLUSIONS

We compared the numerical performance of two newly diagonal Newton method for solving nonlinear equations with singular Jacobian proposed by [Waziri *et al*., 2010a] and [Waziri *et al*., 2010b].

It is well acknowledged that the convergence of Newton method in solving nonlinear equations with singular Jacobian at the solution is unsatisfactory and may even fail. Due to the fact that, DJAN and DJIAN methods have a less computational expenditure and low storage requirements related with the building the approximation scheme. It is worth mentioning that the DJAN and DJIAN methods are capable of reducing the numerical execution time (CPU time) and number of iterations, while maintaining the good precision of the numerical result. Another assertion that makes the DJAN and DJIAN methods attractive is that throughout the numerical test they shown a promising performance. Hence we can recommend that DJIAN method is the good alternative to NM and FN methods, especially when the Jacobian is singular at a solution.

ACKNOWLEDGEMENTS

This is to acknowledge Dr Leong of Department of Mathematics, University Putra Malaysia for his valuable advice and suggestions by improving the quality of this work.

REFERENCES

Broyden C.G. 1965. A class of methods for solving nonlinear simultaneous equations. Math. Comp. 19: 577-593.

Dembo R.S, Eisenstat S.C and Steihaug T. 1982. Inexact Newton method. SIAM J. Numer. Ana. 19: 400-408.

Dennis J E. 1983. Numerical methods for unconstrained optimization and nonlinear equations, Prince-Hall, Inc., Englewood Cliffs, New Jersey.

José L.H, Eulalia M. and Juan R.M. 2009. Modified Newton's method for systems of nonlinear equations with singular Jacobian. Comput. Appl. Math. 224: 77-83.

M.Y waziri, W.J Leong, M.A Hassan and M. Monsi. 2010a. Newton method for systems of nonlinear equations with singular Jacobian using diagonal updating. Journal of mathematics and computations. 8. S10, pp. 1-7.

M.Y waziri, W.J Leong, M.A Hassan and M. Monsi. 2010b. An efficient solver for systems of nonlinear equations with singular Jacobian via diagonal updating. Journal of applied mathematical sciences Vol. 4, 2010, no. 69, 3403 - 3412.

Natasa K. and Zorna L. 2001. Newton-like method with modification of the right- hand vector. J. maths. compt. 71: 237-250.

S. Yun-Qiu and J. Y. Tijalling. 2005. Newton's method for singular nonlinear equations using approximate left and right null space of the Jacobian. App. Num. Math. 54: 256-265.

T.N. Grapsay and E.N. Malihoutsakit. 2007. Newton's method without direct function evaluation. In: Proceedings of 8th Hellenic European Conference on Computer Mathematics and its Applications (HERCMA 2007), Athens, Hellas.