Malaysian Journal of Mathematical Sciences 5(2): 241-255 (2011)

Jacobian-Free Diagonal Newton's Method for Solving Nonlinear Systems with Singular Jacobian

[1]Mohammed Waziri Yusuf, [1,2]Leong Wah June and [1,2]Malik Abu Hassan

[1]*Faculty of Science, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia*

[2]*Institute for Mathematical Research, Universiti Putra Malaysia, 43400 UPM Serdang, Malaysia*

E-mail: waziri@math.upm.edu.my

ABSTRACT

The basic requirement of Newton's method in solving systems of nonlinear equations is, the Jacobian must be non-singular. This condition restricts to some extent the application of Newton method. In this paper we present a modification of Newton's method for systems of nonlinear equations where the Jacobian is singular. This is made possible by approximating the Jacobian inverse into a diagonal matrix by means of variational techniques. The anticipation of our approach is to bypass the point in which the Jacobian is singular. The local convergence of the proposed method has been proven under suitable assumptions. Numerical experiments are carried out which show that, the proposed method is very encouraging.

Keywords**:** Nonlinear equations, diagonally update, Jacobian, singular.

INTRODUCTION

Let us consider the problem of finding the solution of nonlinear equations

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

where $F = (f_1, f_2 ... f_n): R^n \to R^n$ is assumed to satisfy the following assumptions:

A1. F is continuously differentiable in on open neighbourhood $E \subset R^n$.

A2. There exists a solution x^* in E such that $F(x^*) = 0$ and $F'(x^*)$ is non-singular at a solution.

A3. The Jacobian $F'(x_k)$ is Lipschitz continuous at x^*.

The best-know method for finding the solution to (1), is the Newton's method, in which the Newtonian iterations are given by:

$$x_{k+1} = x_k - (F'(x_k))^{-1} F(x_k), \tag{2}$$

where $k = 0,1,2\ldots$. If $F'(x^*)$ is non-singular at a solution of (1) , the convergence is guaranteed and the rate is quadratic from any initial point x_0 in the neighbourhood of x^* (Dennis and Wolkowicz (1993); Shen and Ypma (2005)), i.e.

$$\|x_{k+1} - x^*\| \le h \|x_k - x^*\|^2 \tag{3}$$

for some *h*. When the Jacobian is singular the convergence rate may be unsatisfactory, i.e. it slows down when approaching a singular root and the convergence to x^* may even be lost (Leong and Hassan (2009); Hassan *et al.* (2009); Dennis and Schnabel (1983); Waziri *et al.* (2010)). This condition (non-singular Jacobian $F'(x^*) \neq 0$) restricts to certain extent the application of Newton's method for solving nonlinear equations (Kelly (1995)). Some approaches have been developed to overcome this shortcoming. The simplest approach is incorporated in the fixed Newton's method. The method generates an iterative sequence $\{x_k\}$ from a given initial guess x_0 (Ortega and Rheinboldt (1970); Waziri *et al.* (2010); Farid *et al.* (2011))

$$x_{k+1} = x_k - (F'(x_0))^{-1} F(x_k), \quad k = 0,1,2,\ldots. \tag{4}$$

This method overcomes both the disadvantages of Newton's method (computing and storing the Jacobian in each iteration) but the method is significantly slow. From the computational complexity point of view, the fixed Newton is cheaper than Newton's method (Ortega and Rheinboldt (1970)), however, it still requires computing and storing the Jacobian at initial guess x_0.

The idea of diagonal updating has been used by (Modarres *et al.* (2011); Grapsay and Malihoutsakit (2007); Leong *et al.* (2010)) on unconstrained optimization. Leong *et al.* (Modarres *et al.* (2011)) approximates the Hessian into a diagonal matrix and Hassan *et al.* (2003) focuses on the Hessian inverses diagonal updating. In this paper we proposed a modification of Newton's method for nonlinear systems with singular Jacobian at the solution x^*, by approximating the Jacobian inverse into a diagonal matrix. The anticipation has been to bypass the point at which the Jacobian is singular, since we do not need to compute the Jacobian at all. The method proposed in this work is computationally cheaper than Newton's method and fixed Newton's method in term of CPU time, due to the fact that our strategy is derivatives free and the approximation is only on the diagonal elements.

JACOBIAN-FREE DIAGONAL NEWTON'S METHOD FOR SOLVING NONLINEAR SYSTEMS WITH SINGULAR JACOBIAN (JFSJ)

In this section, we derive Jacobian-Free diagonal Newton's method for solving nonlinear systems with singular Jacobian. We start by the mean value theorem, to obtain the secant's equation

$$\bar{F}'(x_k)\Delta x_k = \Delta F_k, \tag{5}$$

where $\bar{F}'(x_k)$ is the Jacobian matrix, $\Delta x_k = x_{k+1} - x_k$ and $\Delta F_k = F(x_{k+1}) - F(x_k)$.

On the other hand, (5) can be rearranged to give

$$\Delta x_k = (\bar{F}'(x_k))^{-1}\Delta F_k. \tag{6}$$

Let G be an approximation of the Jacobian inverse into a diagonal matrix i.e.

$$(\bar{F}'(x_k))^{-1} \approx G, \tag{7}$$

We propose to update G by adding a correction U (diagonal matrix) in each iteration i.e.

$$G_{k+1} = G_k + U_k. \tag{8}$$

To obtain the accurate information of the Jacobian inverse in relation to the updating matrix, G_{k+1}, we entail G_{k+1} to satisfy the secant's equation (5), i.e.

$$\Delta x_k = (G_k + U_k)\Delta F_k. \quad (9)$$

Now, we can write using the weak secant condition Farid *et al.* (2010), Leong and Hassan (2011) and Natasa and Zorna (2001) to obtain

$$\Delta F_k^T \Delta x_k = \Delta F_k^T (G_k + U_k)\Delta F_k. \quad (10)$$

In order to ensure superior condition number and numerical steadiness in the approximation, we effort to control the error of the U (correction) through the following problem.

$$\begin{aligned} &\min \frac{1}{2}\|U_k\|_F^2 \\ &\text{s.t } \Delta F_k^T (G_k + U_k)\Delta F_k = \Delta F_k^T \Delta x_k, \end{aligned} \quad (11)$$

where $\|.\|_F$ is the Frobenius norm. Denoting $U = diag(\tau_1, \tau_2, \ldots, \tau_n)$ and $\Delta F_k = (\Delta F_k^{(1)}, \Delta F_k^{(2)}, \ldots \Delta F_k^{(n)})$, therefore we can rewrite (11) as follows

$$\begin{aligned} &\min \frac{1}{2}(\tau_1^2 + \tau_2^2 + \ldots + \tau_n^2) \\ &\text{s.t } \sum_{i=1}^{n} \Delta F_k^{(i)^2} \tau_i = \Delta F_k^T \Delta x_k - \Delta F_k^T G_k \Delta F_k. \end{aligned} \quad (12)$$

The solution of (12) can be found through taking into consideration its Lagrange function as follows

$$L(\tau_i, \lambda) = \frac{1}{2}(\tau_1^2 + \tau_2^2 + \ldots + \tau_n^2) + \beta(\sum_{i=1}^{n} \Delta F_k^{(i)^2} \tau_i - \Delta F_k^T \Delta x_k + \Delta F_k^T G_k \Delta F_k). \quad (13)$$

where β is the corresponding Lagrange multiplier.

Differentiating (13) with respect to each τ_i and equating them to zero, we get

$$\frac{\partial L}{\partial \tau_i} = \tau_i + \beta \Delta F_k^{(i)^2} = 0, \quad i = 0, 1, 2, \ldots, n. \quad (14)$$

It follows from (13) that

$$\tau_i = -\beta \Delta F_k^{(i)^2}, \quad i = 0,1,2,...,n. \tag{15}$$

Multiplying (15) by $\Delta F_k^{(i)^2}$ and summing over n, we have

$$\sum_{i=1}^{n} \Delta F_k^{(i)^2} \tau_i = -\sum_{i=1}^{n} \Delta F_k^{(i)^4} \beta. \tag{16}$$

In order to invoking the constraint, we differentiate (13) with respect to β which yields

$$\sum_{i=1}^{n} \Delta F_k^{(i)^2} \tau_i = \Delta F_k^T \Delta x_k - \Delta F_k^T G_k \Delta F_k. \tag{17}$$

Using (16) and (17), it follows that

$$\beta = -\frac{\Delta F_k^T \Delta x_k - \Delta F_k^T G_k \Delta F_k}{\sum_{i=1}^{n} \Delta F_k^{(i)^4}} \tag{18}$$

Substituting (18) into (15), followed by some algebraic manipulation, we obtain:

$$\tau_i = \frac{(\Delta F_k^T \Delta x_k - \Delta F_k^T G_k \Delta F_k)}{\sum_{i=1}^{n} \Delta F_k^{(i)^4}} \Delta F_k^{(i)^2}, \quad i = 1,2,\ldots,n. \tag{19}$$

Letting $E_k = diag(\Delta F_k^{(1)^2}, \Delta F_k^{(2)^2}, \ldots, \Delta F_k^{(n)^2})$ and $\sum_{i=1}^{n} \Delta F_k^{(i)^4} = Tr(E_k^2)$ where Tr is the trace operation, yields

$$U_k = \frac{(\Delta F_k^T \Delta x_k - \Delta F_k^T G_k \Delta F_k)}{Tr(E_k^2)} E_k. \tag{20}$$

Finally, we present the proposed updating scheme as follows:

$$G_{k+1} = G_k + \frac{(\Delta F_k^T \Delta x_k - \Delta F_k^T G_k \Delta F_k)}{Tr(E^2)} E_k. \tag{21}$$

We safeguard the possibly very small ΔF_k and $Tr(E^2)$, where $\|\Delta F_k\| \geq 10^{-4}$. If not set $G_{k+1} = G_k$. Based on the above explanation, we have the following algorithm.

Algorithm JFSJ:

Step 1: Given x_0 and $G_0 = I_n$, set $k = 0$
Step 2: Compute $F(x_k)$
Step 3: Compute $x_{k+1} = x_k - G_k F(x_k)$
Step 4: Check if $\|\Delta x_k\| + \|F(x_k)\|_2 \leq 10^{-4}$ stop. If not go to step 5.
Set $k := k+1$
Step 5: If $\|\Delta F_k\|_2 \geq 10^{-4}$, compute G_{k+1} using formula (21), else set $G_{k+1} = G_k$ and go to 2.

CONVERGENCE ANALYSIS

To analyze the convergence of the proposed method, we will make the following standard assumptions on F .

Assumption 1

(i) F is differentiable in an open convex set E in $\Re^n$
(ii) There exists $x^* \in E$ such that $F(x^*) = 0$, $F'(x)$ is continuous for all x.
(iii) $F'(x)$ Satisfies Lipschitz condition of order one i.e. there exists a positive constant μ such that

$$\|F'(x) - F'(y)\| \leq \mu \|x - y\| \tag{22}$$

for all $x, y \in \Re^n$
(iv) There exists constants $c_1 \leq c_2$ such that $c_1 \|\omega\|^2 \leq \omega^T F'(x)\omega \leq c_2 \|\omega\|^2$ for all $x \in E$ and $\omega \in \Re^n$. To prove the convergence of JFSJ method, we need the following result.

Theorem 3.1.

Let Assumption 1 holds. There are $K_B > 0,\ \delta > 0$ and $\delta_1 > 0,$, such that if $x_0 \in B(\delta)$ and the matrix-valued function $B(x)$ satisfies $\left\| I - B(x)F'(x^*) \right\| = \rho(x) < \delta_1$ for all $x \in B(\delta)$ then the iteration

$$x_{k+1} = x_k - B(x_k)F(x_k), \tag{23}$$

converges linearly to x^*.

For the proving of Theorem 3.1 see Kelly (1995).

Based on the Theorem 3.1 together with our explanation in the previous section, we have the following results.

Theorem 3.2

With that the assumption 1 holds, there are constant $\beta > 0,\ \delta > 0,\ \alpha > 0$ and $\gamma > 0,$ such that if $x_0 \in E$ and D_0 satisfies $\left\| I - G_0 F'(x^*) \right\|_F < \delta,$ where $\|.\|_F$ denotes the Frobenius norm for all $x \in E$ then the iteration

$$x_{k+1} = x_k - G_k F(x_k)$$

where G_k defined by (21), converges linearly to x^*.

Proof.

We need to show that the updating formula D_k satisfied $\left\| I - G_k F'(x^*) \right\|_F < \delta_k,$ for some constant $\delta_k > 0$ and all k. $\|G_{k+1}\|_F = \|G_k + U_k\|_F$, it follows that

$$\|G_{k+1}\|_F \leq \|G_k\|_F + \|U_k\|_F\,. \tag{24}$$

For $k=0$ and assuming $D_0 = I$, we have

$$\|G_1\|_F \le \|G_0\|_F + \|U_0\|_F . \tag{25}$$

Since $G_0 = I_n$ hence $\|G_0\|_F = \sqrt{n}$. From (20) when $k=0$ we have

$$|U_0^{(i)}| = \left| \frac{\Delta F_0^T \Delta x_0 - \Delta F_0^T G_0 \Delta F_0}{Tr(E_0^{\ 2})} \Delta F_0^{(i)^2} \right|$$

$$\le \frac{|\Delta F_0^T \Delta x_0 - \Delta F_0^T G_0 \Delta F_0|}{Tr(E_0^{\ 2})} \Delta F_0^{(\max)^2}$$

$$= \frac{|\Delta F_0^T \Delta x_0 - \Delta F_0^T D_0 \Delta F_0|}{\Delta F_0^{(\max)^2} \sum_{i=1}^n \Delta F^{(i)^4}} \Delta F_0^{(\max)^4} . \tag{26}$$

Since $\frac{\Delta F_0^{(\max)^4}}{\sum_{i=1}^n \Delta F_0^{(i)^4}} \le 1$, then (26) turns into

$$|U_0^{(i)}| \le \frac{|\Delta F_0^T F'(x) \Delta F_0 - \Delta F_0^T G_0 \Delta F_0|}{\Delta F_0^{(\max)^2}}. \tag{27}$$

From condition (iv) and since $c_1 \le c_2$, but c_1 and c_2 can be negative, hence we may not have $\|c_1\| \le \|c_2\|$. Therefore, we choose the largest among $\|c_1\|$ and $\|c_2\|$ (i.e. $c = \max\{|c_1|, |c_2|\}$), then (27) becomes

$$|U_0^{(i)}| \le \frac{|c - \sqrt{n}|(\Delta F_0^T \Delta F_0)}{\Delta F_0^{(\max)^2}}. \tag{28}$$

Since $\Delta F_0^{(i)^2} \le \Delta F_0^{(\max)^2}$ for $i = 1,\ldots,n,$ it follows that

$$|U_0^{(i)}| \le \frac{|c - \sqrt{n}| \Delta F_0^{(\max)^2}}{\Delta F_0^{(\max)^2}}. \tag{29}$$

Hence we obtain

$$\|U_0\|_F \le n^{\frac{3}{2}} \,|\, c-\sqrt{n}\,|. \tag{30}$$

Suppose $\alpha = n^{\frac{3}{2}} \,|\, c-\sqrt{n}\,|$, then

$$\|U_0\|_F \le \alpha. \tag{31}$$

Substituting (31) into (25) and let $\beta = \sqrt{n}+\alpha$, it follows that

$$\|G_1\|_F \le \beta. \tag{32}$$

At $k=0$, it's assumed that $\|I - G_0 F'(x^*)\|_F < \delta$, then we have

$$\begin{aligned} \|I - G_1 F'(x^*)\|_F &= \|I-(G_0+U_0)F'(x^*)\|_F \quad , \\ &\le \|I - G_0F'(x^*)\|_F + \|U_0 F'(x^*)\|_F \quad , \\ &\le \|I - G_0F'(x^*)\|_F + \|U_0\|_F \, \|F'(x^*)\|_F \quad , \end{aligned} \tag{33}$$

hence $\|I - G_1 F'(x^*)\|_F < \delta + \alpha\varphi = \delta_1$. (Even when $\|F'(x^*)\|_F = 0$). Thus, by induction, $\|I - G_k F'(x^*)\|_F < \delta_k$ for all k. Therefore from Theorem 3.1, the sequence generated by Algorithm JFSJ converges linearly to x^*.

NUMERICAL RESULTS

In order to demonstrate the performance of our new proposed method (JFSJ) for solving nonlinear systems with singular Jacobian, it has been applied to some popular problems. We implemented the method (JFSJ) using variable precision arithmetic in Matlab 7.0 . All the calculations were carried out in double precision computer. The stopping criterion used is

$$\|\Delta x_k\| + \|F(x_k)\| \le 10^{-4}. \tag{34}$$

We present and describe the used test problems as follows:

Problem 1. (Jose *et al.* (2009)). $f : R^2 \to R^2$ is defined by

$$f(x) = \begin{cases} (x_1 - 1)^2 (x_1 - x_2) \\ (x_1 - 2)^5 \cos\left(\dfrac{2x_1}{x_2}\right) \end{cases}$$

$x_0 = (0,3)$,$(0.5, 2)$ are chosen and $x^* = (1,2)$.

Problem 2. (Ishihara, K. (2001)). $f : R^3 \to R^3$ is defined by

$$f(x) = \begin{cases} 4x_1 - 2x_2 + x_1^2 - 3 \\ -x_1 + 4x_2 - x_3 + x_2^2 - 3 \\ -2x_2 + 4x_3 + x_3^2 - 3 \end{cases}$$

$x_0 = (-1.5, 0, -1.5), (4, 0, 4), (-1, 5, -1), (4, 4, 4), (-10, 0, -10)$ are chosen and $x^* = (1,1,1)$.

Problem 3. $f : R^2 \to R^2$ is defined by

$$f(x) = \begin{cases} \dfrac{2}{1 + x_1^2} + \sin(x_2 - 1) - 1 \\ \sin(x_2 - 1) + \dfrac{2}{1 + x_2^2} + -1 \end{cases}$$

$x_0 = (0.5, 0.5), (2, 2), (0.1, 0.1)$ are chosen and $x^* = (1,1)$.

Problem 4. $f : R^2 \to R^2$ is defined by

$$f(x) = \begin{cases} 1 + \tan(2 - 2\cos x_1) - \exp(\sin x_1) \\ 1 + \tan(2 - 2\cos x_2) - \exp(\sin x_2) \end{cases}$$

$x_0 = (3, 0), (0, 0.5), (-0.5, -0.5)$ are chosen and $x^* = (0,0)$.

Problem 5. $f : R^2 \to R^2$ is defined by

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

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

Problem 6. $f : R^3 \to R^3$ is defined by

$$f(x) = \begin{cases} \cos x_1 - 9 + 3x_1 + 8\exp(x_2) \\ \cos x_2 - 9 + 3x_2 + 8\exp(x_1) \\ \cos x_3 - x_3 - 1 \end{cases}$$

$x_0 = (-1,-1,-1), (3,3,3)(0.5,0.5,0.5), (-3,-3,-3)$ and $x^* = (0,0,0)$.

Problem 7. (Ishihara, K. (2001)). $f : R^2 \to R^2$ is defined by

$$f(x) = \begin{cases} 4x_1 - 2x_2 + x_1^2 - 3 \\ -2x_1 + 4x_2 + x_1^2 - 3 \end{cases}$$

$x_0 = (3,3), (0,-1.5), (-2,3), (0,2)$ are chosen and $x^* = (1,1)$.

Problem 8. $f : R^2 \to R^2$ is defined by

$$f(x) = \begin{cases} \sqrt{3}x_1^2 - x_2^2 \\ \cos(x_1) - 1/(1 + x_2^2) \end{cases}$$

$x_0 = (0.5,1)$ is chosen and $x^* = (0,0)$.

Problem 9. (Shen and Ypma (2005)). $f: R^2 \to R^2$ is defined by

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

$x_0 = (0.5, 0.4), (-0.5, -0.4), (-0.3, -0.5)$ and (0.4,0.5) are chosen and $x^* = (0,0)$.

TABLE 1: The Numerical Results of JFSJ method on Problems 1-9

Problems	x_0	**Number of iteration**	**CPU time**
1	(0,3)	9	0.0002
	(0.5, 2)	7	0.0001
2	(-1.5, 0, -1.5)	17	0.0156
	(4,0,4)	38	0.0311
	(-1, 0.5, -1)	15	0.0009
	(-10,0,-10)	51	0.0312
	(4,4,4)	10	0.0311
3	(0.5,0.5)	18	0.0281
	(1.5, 1.5)	18	0.0252
	(2, 2)	27	0.0310
	(0.1,0.1)	8	0.0013
4	(3,0)	12	0.0006
	(0, 0.5)	8	0.0004
	(-0.5, -0.5)	6	0.0004
5	(-0.5,-0.5)	5	0.0003
6	(-1,-1,-1)	10	0.0156
	(3,3,3)	15	0.0321
	(-3,-3,-3)	19	0.0388
	(0.5,0.5,0.5)	8	0.0018
7	(-2,3)	14	0.0156
	(3,3)	11	0.0107
	(0,-1.5)	18	0.0168
	(0,2)	10	0.0003
8	(0.5,1)	39	0.0312
9	(0.5,0.4)	11	0.0010
	(-0.5,-0.4)	21	0.0156
	(-0.3,-0.5)	9	0.0006
	(0.4,0.5)	12	0.0012

The numerical results of our proposed method (JFSJ) is reported in Table 1, which include number of iteration and CPU time in seconds. From Table 1 it appears that, proposed method of this paper JFSJ has solved all the tested problems with their respective initial guesses. Moreover the results are very encouraging for our method. Indeed we observe that JFSJ method requires little CPU time to converge to the solution, due to the fact that Newton's method may likely fails to converge if the Jacobian is singular.

Another advantage of our method over Newton's method and Fixed Newton method is the storage requirement; this is more noticeable as the dimension increases.

CONCLUSION

In this paper, we have presented a modification of Newton's method for solving systems of nonlinear equations with singular Jacobian. Our scheme is based on approximating the Jacobian inverse to a non-singular diagonal matrix without computing the Jacobian at all. The anticipation has been to bypass the point at which the Jacobian is singular. Among its desirable feature is that it requires very low memory requirement in building the approximation of the Jacobian inverse. In fact, the size of the update matrix increases in $O(n)$, as oppose to Newton's and Fixed Newton methods that increase in $O(n^2)$. This is more noticeable as the dimension of the system increases.

Finally, we conclude that, to the best of our knowledge there are not many alternatives when the Jacobian matrix of a nonlinear system is singular. As such, this result confirms that our method (JFSJ) is a good alternative to Newton method and fixed Newton, especially when the Jacobian is singular at any point x^k.

REFERENCES

Decker, D.W. and Kelly, C.T. 1985. Broyden's methods for a class of problems having singular Jacobian at root. *SIAM J. of Num. Anal.* **23:** 566-574.

Dennis Jr, J.E. and Schnabel, R.B. 1983. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, NJ: Prentice-Hall.

Dennis, J.E. and Wolkowicz, H. 1993. Sizing and least change secant methods. *SIAM J. Numer. Anal.* **30**: 1291–1313.

Farid, M., Leong, W.J. and Hassan, M.A. 2010. A new two-step gradient-type method for large scale unconstrained optimization. *Computers and Mathematics with Applications.* **59**: 3301-3307.

Farid, M., Leong, W.J. and Hassan, M.A. 2011. An improved multi-step gradient-type method for large scale optimization. Computers and Mathematics with Applications. **61:** 3312-3318.

Grapsay, T.N. and Malihoutsakit, E.N. 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.

Griewank, A. and Osborne, M.R. 1983. Analysis of Newton's method at irregular singularities. *SIAM J. Num. Anal.* **20:** 747-773.

Hassan, M.A., Leong, W.J. and Farid, M. A new gradient method via quasi-Cauchy relation which guarantees descent. *J. Comput. Appl. Math.* **230**: 300-305.

Ishihara, K. 2001. Iterative methods for solving Nonlinear equations with singular Jacobian matrix at the solution. In the *Proceedings of the International Conference on Recent Advances in Computational Mathematics (ICRACM 2001)*, Japan.

Joseé, 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.

Kelley, C.T. 1995. Iterative Methods for Linear and Nonlinear Equations. Philadelphia, PA: SIAM.

Leong, W.J and Hassan, M.A. 2009. A restarting approach for the symmetric rank one update for unconstrained optimization. *Computational Optimization and Applications*. **43**: 327-334.

Leong, W.J and Hassan, M.A. 2011. A new gradient method via least change secant update. *International Journal of Computer Mathematics*. **88**: 816-828.

Leong, W.J., Hassan, M.A. and Farid, M. 2010. A monotone gradient method via weak secant equation for unconstrained optimization. *Taiwanese J. Math.* **14**(2): 413-423.

Modarres, F., Hassan, M.A. and Leong, W.J. 2011. Improved Hessian approximation with modified secant equations for symmetric rank-one method. *J. Comput. Appl. Math.* **235**: 2423-2431.

Natasa, K. and Zorna, L. 2001. Newton-like method with modification of the right-hand vector. *J. Math. Compt.* **71:** 237-250.

Ortega, J.M. and Rheinboldt, W.C. 1970. *Iterative Solution of Nonlinear equations in several variables*. New York: Academic Press.

Shen, Y.Q and Ypma, T.J. 2005. Newton's method for singular nonlinear equations using approximate left and right nullspaces of the Jacobian. *App. Num. Math*. **54:** 256-265.

Waziri, M.Y., Leong, W.J., Hassan, M.A. and Monsi, M. 2010. A new Newton's method with diagonal jacobian approximation for systems of nonlinear equations. *J. Mathematics and Statistics*. **6**: 246-252.

Waziri, M.Y., Leong, W.J., Hassan, M.A. and Monsi, M. 2010 Jacobian computation-free Newton method for systems of Non-Linear equations. *Journal of numerical Mathematics and stochastic*. **2**(1): 54-63.