Journal of Numerical Mathematics and Stochastics, **4** (1) : 70-78, 2012 © ***JNM@S***
http://www.jnmas.org/jnmas4-7.pdf Euclidean Press, LLC
Online: ISSN 2151-2302

A New Higher Order Diagonal-Type Newton's Method for Higher Dimensional Systems of Nonlinear Equations

M. Y. WAZIRI[1,2], **W. J. LEONG**[2], **and A. U. MOYI**[2]

[1]Department of Mathematical Sciences, Bayero University Kano, Kano, Nigeria;
[2]Department of Mathematics, Universiti Putra Malaysia, 43400, Serdang, Selangor Malaysia,
E-mail: mywaziri@gmail.com

Abstract. *In this paper, an efficient diagonal updating scheme has been developed and used to solve large scale systems of nonlinear equations. The algorithm is based on the technique of Waziri et al [M.Y. WAZIRI, W.J. LEONG, M.A. HASSAN, and M. MONSI, Jacobian computation-free Newton method for systems of non-linear equations, Journal of Numerical Mathematics and Stochastics* ***2****(1), (2010), 54-63].Unlike the one step approach in most Newton's-like methods, in our new approach we make use of data from two previous steps. The rationale behind this approach is to improve the current Jacobian inverse approximation by a diagonal matrix. To illustrate the efficiency of our diagonal updating scheme, some numerical experiments are presented, and the results are compared with some other Newton's -like methods.*

Key words : Newton's Method, Systems, Nonlinear Equations, Multi-Points, Diagonal-Type.

AMS Subject Classifications : 65H05

1. Introduction

Consider the problem

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

with $F : R^n \to R^n$, where the mapping F is assumed to satisfy the following standard assumptions:
1. F is continuously differentiable in an open convex set Φ ;
2. There exist a solution x^* of (1) in Φ such that $F(x^*) = 0$ and $F'^*) \neq 0$;

3. The Jacobian $F'(x)$ is local Lipschitz continuous at x^*.
The best approach for finding the solution to (1) is the Newton's method. The method is simple to implements and it produces an iterative sequence $\{x_k\}$ from any given initial guess x_0 in the neighborhood of solution, through the following steps.

The NM Algorithm (Newton's method)
For $k = 0, 1, 2, \ldots$ where $F'(x_k)$ is the Jacobian matrix of F:
Step 1: solve $F'(x_k)s_k = -F(x_k)$
Step 2: Update $x_{k+1} = x_k + s_k$,
where s_k is the Newton correction, and

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

for some h.

Among its attractive feature is that the convergence rate is quadratic from any initial point x_0 in the neighborhood of a solution whenever the Jacobian matrix $F'(x^*)$ is nonsingular at that solution [2, 1],

Notwithstanding, Newton's method have some numerous disadvantages which have attracted the attention of many researchers over time. These include the computation and storage of Jacobian matrix as well as solving n linear equations in each iteration. It is well known that the Jacobian computation entails first-order derivatives of the systems. And some functions derivatives are quite costly and sometime not available or could not be precisely evaluated. In this case Newton's method cannot be directly used. Therefore, to tackle this crucial issue some effort need to be applied.

It is worth mentioning that, the use of quasi-Newton's method has reduced the evaluation cost of an expensive Jacobian matrix. Here the cost of a solution by quasi-Newton's methods could be much less than with inexact Newton methods [3]. Nonetheless it requires to store the full elements of the Jacobian in each iteration in which for large scale will be a very rigorous task to handle. Another interesting variant of Newton's method is the fixed Newton method. This method overcomes both the computation and storage of the Jacobian (except for the first iteration), as well as avoiding solving a system of n linear equations in each iteration, [5]. Still it requires to store full elements of the Jacobian matrix. Numerous approaches exist to overcome the widely known drawbacks. Many efforts have recently been made by a number of authors, see e.g. [4, 10, 6, 13, 11], to overcome the shortcomings of various Newton methods. Among the successful ones are the diagonal Newton's like methods proposed in [6, 13, 11]. These methods are so appealing because, they neither require Jacobian computation and storage, nor solving Newtonian equations (linear) in each iteration. In addition, the methods have floating points operations of $O(n)$. Apart from these achievements, they make use of a standard one-step two-point approach in Jacobian or its inverse approximation by a nonsingular diagonal matrix, which is commonly used by most Newton's-like methods. In contrast, this paper uses a two-step multi points approach to extend the scheme of Waziri et al [11].

Our new approach aims at increasing the accuracy of the Jacobian inverse approximation by a diagonal matrix. We organized the rest of this paper as follows. Section 2 presents the proposed method, Numerical experiments are reported in Section 3, and finally some conclusions are made in Section 4.

2. Derivation Process of I-2DN

This section presents our new modification of Newton's method using a two-step multi-point approach. This method generates a sequence of points $\{x_k\}$ via

$$x_{k+1} = x_k - Q_k F(x_k), \tag{3}$$

where Q_k is a diagonal approximation of the Jacobian inverse matrix. To achieve this, we require to construct a matrix Q_k via a diagonal updating scheme which is a good approximation to the Jacobian inverse matrix. Waziri et al. [11] utilize data from one preceding step to improve the current approximate Jacobian into a diagonal matrix, meaning that $y_k = F(x_{k+1}) - F(x_k)$ and $s_k = x_{k+1} - x_k$. This approach is known as a one-step approach. We continue in the spirit of diagonal updating; using an interpolating curve in the variable-space, we develop an incomplete Taylor series expansion of $F(x)$ at x_k. This is anticipated to be a more accurate approximation of the Jacobian inverse matrix. To this end, we consider some of the most successful of the two-step methods (see [7]-[9] for more detail). By using this two-step approach, we can present the improved incomplete Taylor series expansion of $F(x)$ viz.

$$(s_k - \lambda_k s_{k-1}) \approx Q_k(y_k - \lambda_k y_{k-1}). \tag{4}$$

If $\mu_k = s_k - \lambda_k s_{k-1}$ and $\psi_k = y_k - \lambda_k y_{k-1}$, then it follows from (1) that

$$\mu_k \approx Q_k \psi_k. \tag{5}$$

Since we used information from the last two previous points instead of one step in (4) and (5), the improved incomplete Taylor series expansion of $F(x)$, derived from this scheme, will also improve the precision of the Jacobian inverse approximation. Therefore, we require the building of interpolating quadratic curves $x(\epsilon)$ and $y(\epsilon)$, where by $x(\epsilon)$ interpolates the last two earlier iterates x_{k-1}, x_k by x_{k+1}, and $y(\epsilon)$ interpolates the last two earlier function evaluation F_{k-1}, F_k and F_{k+1} (which are assumed to be available). We follow the approach proposed by [7] to obtain the λ_k in [4] using the value of γ_0, γ_1 and γ_2 respectively. If we assume that $\gamma_2 = 0$ and $\{\gamma_j\}_j^2 = 0$, then γ_k is defined by

$$\begin{aligned} -\gamma_1 &= \gamma_2 - \gamma_1 = \|x(\gamma_2) - x(\gamma_1)\|_{Q_k} \\ &= \|x_{k+1} - x_k\|_{Q_k} = \|s_k\|_{Q_k} \\ &= \sqrt{(s_k^T Q_k s_k)}, \end{aligned} \tag{6}$$

and

$$\begin{aligned} -\gamma_0 &= \gamma_2 - \gamma_0 = \|x(\gamma_2) - x(\gamma_0)\|_{Q_k} \\ &= \|x_{k+1} - x_{k-1}\|_{Q_k} = \|s_k + s_{k-1}\|_{Q_k} \\ &= \sqrt{((s_k + s_{k-1})^T Q_k (s_k + s_{k-1}))}. \end{aligned} \tag{7}$$

let us define ξ as

$$\xi = \frac{\gamma_2 - \gamma_0}{\gamma_1 - \gamma_0}, \tag{8}$$

then μ_k and ψ_k are give as

$$\mu_k = s_k - \frac{\xi^2}{1+2\xi} s_{k-1}, \tag{9}$$

$$\psi_k = y_k - \frac{\xi^2}{1+2\xi} y_{k-1}. \tag{10}$$

Consider (9) and (10) in (5) and the fact that $Q_k \approx F'^{-1}$ to obtain

$$Q_{k+1}^{(i)} = \frac{\mu_k^{(i)}}{\psi_k^{(i)}}. \tag{11}$$

Hence

$$Q_{k+1} = diag(q_{k+1}^{(i)}), \tag{12}$$

for $1 \leq i \leq n$ and $k = 0, 1, 2, \ldots, n$. Here $\mu_k^{(i)}$ is the i^{th} component of the vector μ_k , $\psi_k^{(i)}$ is the i^{th} component of the vector ψ_k and $q_{k+1}^{(i)}$ is the i^{th} diagonal element of a diagonal matrix Q_{k+1} respectively. To safeguard on the possibly of generating singular Q_{k+1}, we proposed the usage of
the following updating scheme for Q_{k+1}.

$$Q_{k+1} = \begin{cases} \frac{\mu_k}{\psi_k} & ; \psi_k \neq 0, \\ Q_k & ; \text{otherwise.} \end{cases}$$

Now we are able to present the following main result of this paper.

The I-2DN Algorithm
Step 1 : Choose an initial guess x_0 and $Q_0 = I$, and let $k := 0$
Step 2 : Compute $F(x_k)$. If $\|F(x_k)\| \leq 10^{-4}$, stop
Step 3 : If $k := 0$ define $x_1 = x_0 - Q_0 F(x_0)$. Else if $k := 1$ set $\mu_k = s_k$ and $\psi_k = y_k$, and goto 5
Step 4 : If $k \geq 2$ compute γ_1 ,γ_0 and ξ via (6)-(8), respectively and find ψ_k and μ_k using (9)and (10), respectively. If $\psi_k^T \mu_k \leq 10^{-4} \|\psi_k\|_2 \|\mu_k\|_2$, set $\mu_k = s_k$ and $\psi_k = y_k$
Step 5 : Let $x_{k+1} = x_k - Q_k F(x_k)$ and update Q_{k+1} as define by (12)
Step 6 : Check if $\|\psi_k\|_2 \geq \epsilon_1$ where $\epsilon_1 = 10^{-4}$, if yes retain Q_{k+1} that is computed by step 5. Else set, $Q_{k+1} = Q_k$
Step 7 : Set k:=k+1 and goto 2.

3. Numerical Results

In order to evaluate the performance of the proposed method, we apply this updating scheme to solve five benchmark problems using dimensions ranging from 25 to 50,000 variables. Four Newton's-like methods are compared and the comparison involves the number of iterations, CPU time in seconds,storage requirement and floating points operations. The methods considered namely are:
(1) I-2DN stands for method proposed in this paper.
(2) The Newton's method (NM).
(3) The Fixed Newton method (FN).
(4) Broyden method (BM).
All the computational experiments were carried out in double precision. The termination point

is defind viz.

$$\|s_k\|+\|F(x_k)\|\leq 10^{-4}, \tag{13}$$

and we have designed the program to terminate whenever:
(i) The number of iterations is at least 500, but no point of x_k satisfies (13),
(ii) CPU time in seconds reaches 500,
(iii) Insufficient memory to initiate the run.
"–" in the tabulated results represents a failure due to any of (i)-(iii).
In the following we present some details of the solution of the benchmark test systems of nonlinear equations.

Problem 1. System of n nonlinear equations :

$$f_i(x) = 3n - \left(\sum_{i=1}^{n}(\cos x_i - 2)\right) - \left(\sum_{i=1}^{n} x_i + \sin(x_i - 2)\right)$$

$i = 1,2,\ldots,n, \quad and \quad x_0 = (0,0,\ldots 0).$

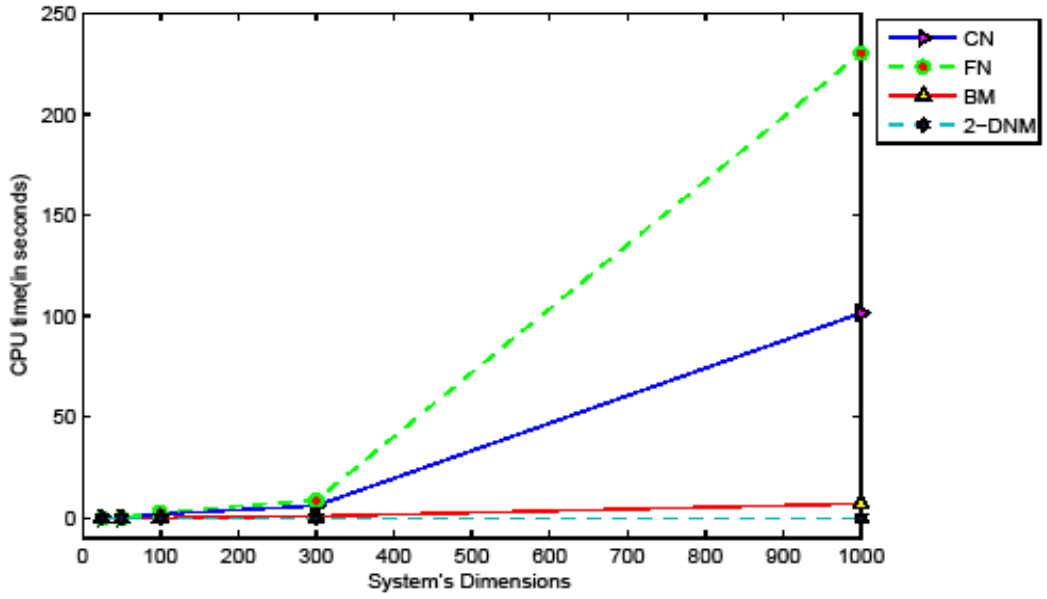

Figure 1: Comparison of CPU times in the NM, FN, BM and I-2DN methods for Problem 1

Problem 2. The system:

$$f_1(x) = x_1^2 - \cos(x_1 - x_2)$$

$$f_i(x) = (4 - 3x_i)(1 + \sin(x_i - 1)) + \sum_{j=1}^{n} x_j - \exp(\sin(x_i - 1)) - n$$

$$f_n = \sec(x_n - x_{n-1}) + \exp(\sin(x_n - 1)) - 2x_n$$

$i = 2,\ldots,n-1, \quad and \quad x_0 = (4,4,4,\ldots,4).$

Problem 3. The System :

$$f_i(x) = \left(\sum_{i=1}^{n} x_i^2 + i\right)(x_i - 1) + \exp(x_i - 1) - 1$$

$i = 1,2,\ldots,n, \quad and \quad x_0 = (3,3,3,3,\ldots,3).$

Problem 4. Extended system of Byeong [12] :

$$f_i(x) = \cos(x_i^2 - 1) - 1$$

$i = 1,2,\ldots,n, \quad and \quad x_0 = (0.5,0.5,..,0.5) \times \frac{\pi}{180} = (0.0087,0.0087,\ldots,0.0087).$

Problem 5. System of n nonlinear equations :

$f_1(x) = \frac{1}{(1+x_1^2)} - \exp(x_1)$

$f_i(x) = \frac{1}{(1+x_i^2)} + \cos(x_{i+1}) - 2\exp(x_i)$

$i = 2,\ldots,n-1, \quad x_{n+1} = 0, \quad and \quad x_0 = (0.5,0.5,..,.5).$

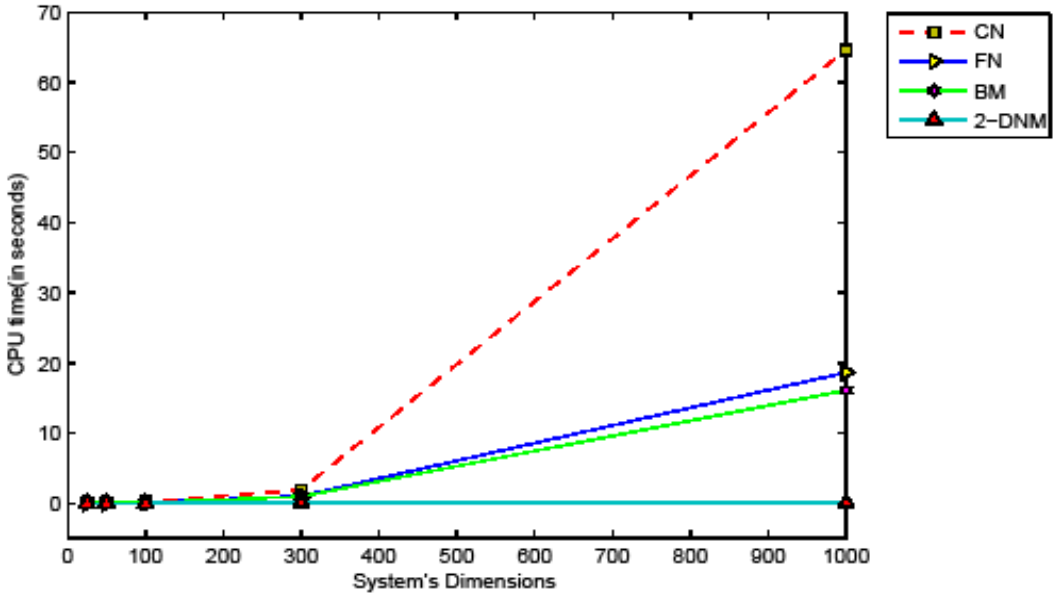

Figure 2: Comparison of CPU times in the NM, FN, BM and I-2DN methods for Problem 5

These four methods are compared in terms of CPU time, number of iterations, floating point operations and matrix storage requirements, as displayed in Table 1.

From the CPU time point of view, Figures 1 and 2 show that the I-2DN method consumes less CPU time as compared to the NM, FN and BM methods. This is due to less computational effort associated with the building of the approximate Jacobian inverse while avoiding the needs of solving Newtonian (linear) systems in each iteration.

It is remarkable how the I-2DN method significantly reduce the matrix storage requirement from n^2 location storage to only a row vector requirement. Therefore, since the storage needed by the I-2DN method is less than the one utilized by NM, FN and BM methods, I-2DN

methods appear to be the best available option for solving large-scale systems. Table 1 illustrates also that even as the dimension increases up to 50,000 variables, the I-2DN method is able to converge while the NM, FN and BM methods fail to do so. Apparently due to insufficient memory to initiate the run.

Table 1: Numerical results of the NM, FN, BM and I-2DN methods.

Problem	Dimension	NM		FN		BM		I-2DN	
		NI	CPU	NI	CPU	NI	CPU	NI	CPU
1		5	0.062	–	–	10	0.016	9	0.001
2		6	0.094	45	0.031	–	–	15	0.002
3	25	9	0.078	45	0.046	–	–	14	0.010
4		6	0.031	63	0.015	8	0.011	26	0.009
5		6	0.031	50	0.047	–	–	6	0.001
1		5	0.125	–	–	11	0.047	9	0.004
2		6	0.156	45	0.047	–	–	13	0.028
3	50	9	0.227	46	0.098	–	–	6	0.012
4		6	0.047	75	0.021	9	0.016	27	0.010
5		6	0.094	52	0.053	–	–	6	0.014
1		5	2.752	–	–	11	0.062	9	0.012
2		6	0.515	45	0.175	–	–	15	0.030
3	100	9	0.841	52	0.301	–	–	15	0.018
4		6	0.461	90	0.031	9	0.022	27	0.014
5		6	0.468	53	0.187	–	–	6	0.014

4. Conclusions

We have presented a new diagonal Newton's method for solving large-scale systems of nonlinear equations using a two-step approach, namely I-2DN. The method uses an identity matrix as initial approximation to the Jacobian inverse which can be updated in each iteration. The numerical experimentsIt indicate that the proposed method is capable of significantly reducing the execution time (CPU time), matrix storage requirement and floating points operations, as compared to NM, FN and BM methods. This happens while maintaining good accuracy of the numerical solution to some extend. Another fact that makes the I-2DN method more appealing is that throughout the tested problems it never failed to converge.

Finally, we conclude that our I-2DN method is a good alternative to Newton's-type methods for solving large-scale nonlinear equations or systems with a nearly singular Jacobian.

Table 1: Numerical results of the NM, FN, BM and I-2DN methods (Continuation).

Problem	Dimension	NM		FN		BM		I-2DN	
		NI	CPU	NI	CPU	NI	CPU	NI	CPU
1		5	10.104	–	–	11	2.667	9	0.030
2		6	5.912	45	1.768	–	–	13	0.049
3	300	9	10.154	58	4.410	–	–	15	0.039
4		6	8.731	–	–	10	2.093	27	0.031
5		6	4.803	56	1.602	–	–	7	0.022
1		5	114.567	–	–	11	15.401	9	0.034
2		6	108.530	–	–	–	–	16	0.072
3	1000	9	165.814	74	18.171	–	–	16	0.045
4		6	102.712	–	–	10	11.161	31	0.048
5		6	110.371	59	25.319	–	–	7	0.031
1		–	–	–	–	–	–	9	1.001
2		–	–	–	–	–	–	14	1.669
3	50000	–	–	–	–	–	–	20	4.627
4		–	–	–	–	–	–	35	0.589
5		–	–	–	–	–	–	7	0.743

Acknowledgements
We thank an anonymous referee for his critical reading of the typescript.

References

[1] L. H. Jose, M. Eulalia, and R. M. Juan, Modified Newton's method for systems of nonlinear equations with singular Jacobian, *Computational and Applied Mathematics* **224** (2009), 77-83.

[2] J. E. Dennis, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prince-Hall, Englewood Cliffs, New Jersey, 1983.

[3] L. Binh, On the convergence of a quasi-Newton's method for sparse nonlinear systems, *Mathematics of Computation* **32** (1978), 447-451.

[4] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, Inexact Newton methods, *SIAM Journal on Numerical Analysis* **19** (1982), 400-408.

[5] K. Natasa, and L. Zorna, Newton-like method with modification of the right-hand vector, *Journal of Computational Mathematics* **71** (2001), 237-250.

[6] M. Y. Waziri, W. J. Leong, M. A. Hassan, and M. Monsi, A New Newton method with diagonal Jacobian approximation for systems of Non-Linear equations. *Journal of Mathematics and Statistics* **6** (3) (2010), 246-252.

[7] J. A. Ford, and I. A. Moghrabi, Alternating multi-step quasi-Newton methods for unconstrained optimization, *Journal of Computational and Applied Mathematics* **82** (1997), 105-116.

[8] J. A. Ford, and I. A. Moghrabi, Multi-step quasi-Newton methods for optimization, *Journal of Computational and Applied Mathematics* **50** (1994), 305-323.

[9] J. A. Ford, and S. Thrmlikit, New implicit updates in multi-step quasi-Newton methods for unconstrained optimization, *Journal of Computational and Applied Mathematics* **152** (2003), 133-146.

[10] L. Lukšan, S. C. Eisenstat, and T. Steihaug, Inexact trust region method for large sparse systems of nonlinear equations, *Journal of Optimization Theory and Applications* **81** (1994), 569-590.

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

[12] C. S. Byeong, M. T. Darvishi, and H. K. Chang, A comparison of the Newton Krylov method with high order Newton-like methods to solve nonlinear systems, *Applied Mathematics and Computation* **217** (2010), 3190-3198.

[13] W. J. Leong, M. A. Hassan, and M. Y. Waziri, A matrix-free quasi- Newton method for solving large-scale nonlinear systems, *Computers and Mathematics With Applications* **62** (5) (2011), 2354-2363.

Article history: Submitted December, 31, 2011; Accepted February, 17, 2012.