Applied Mathematical and Computational Sciences
Volume 6, Issue 1, August 2014, Pages 21-30

A DIAGONAL QUASI-NEWTON METHOD FOR SYSTEM OF NONLINEAR EQUATIONS

M. Y. WAZIRI and H. A. AISHA

Department of Mathematical Sciences
Faculty of Science
Bayero University Kano
Kano, Nigeria
E-mail: mywaziri@gmail.com

Abstract

We present a diagonal quasi-Newton update for solving large scale systems of nonlinear equations via fixed point two-step approach. In this approach, instead of using updating matrix in every iteration to construct the interpolation curves, we proposed to use identity matrix throughout. The anticipation has been to reduce the computational cost, floating point operations and CPU time respectively. The effectiveness of our proposed scheme is appraised through numerical comparison with some well known Newton's like methods.

1. Introduction

Let us consider the systems of nonlinear equations

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

where $F : R^n \to R^n$ is a nonlinear mapping. Often, the mapping, F is assumed to satisfying the following assumptions:

A1. There exists an $x^* \in R^n$ s.t $F(x^*) = 0$;

A2. F is a continuously differentiable mapping in a neighborhood of x^*;

A3. $F'(x^*)$ is invertible.

2010 Mathematics Subject Classification: 65H11, 65K05.
Keywords: Step, Secant, System, equations approximation.
Received April 3, 2014
Accepted July 3, 2014 Academic Editor: Jose C. Valverde

The prominent method for finding the solution to (1), is the Newton's method which generates a sequence of iterates $\{x_k\}$ from a given initial guess x_0 via

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

where $k = 0, 1, 2, \ldots$. Nevertheless, Newton's method requires the computation of the matrix entails the first-order derivatives of the systems. In practice, computations of some functions derivatives are quite costly and sometime they are not available or could not be done precisely. In this case Newton's method cannot be applied directly [2].

Moreover, some significant efforts have been made in order to eliminate the well known shortcomings of Newton's method for solving systems of nonlinear equations, particularly large-scale systems. For example Chord's Newton's method, inexact Newton's method, quasi-Newton's method, etc. (see for e.g. [1], [3], [6], [13], [15]). On the other hand, most of these variants of Newton's method still have some shortcomings as Newton's counterpart. For example Broyden's method and Chord's Newton's method need to store an $n \times n$ matrix and their floating points operations are $O(n^2)$.

To do away with these disadvantages, a single step diagonally Newton's method has been suggested by Leong et al. [12] and showed that their approach is significantly cheaper than Newton's method and some of its variants. To incorporate the higher order accuracy in approximating the Jacobian matrix, Waziri et al. [16] proposed a two step approach to develop the diagonal updating scheme D_{k+1} that will satisfy the weak secant equation

$$\rho_k^T D_{k+1} \rho_k = \rho_k^T \mu_k, \tag{3}$$

where $\rho_k = s_k - \alpha_k s_{k-1}$ and $\mu_k = y_k - \alpha_k y_{k-1}$. For this purpose, equation (3) is obtained by means of constructing the interpolation quadratic curves $x(\nu)$ and $y(\nu)$, where $x(\nu)$ interpolates the iterates $x_{k-1} x_k$ and x_{k-1}, and $y(\nu)$ interpolates the functions F_{k-1}, F_k and F_{k+1}.

A diagonal updating matrix D_{k+1} that satisfying (3) is obtained via the following relation

$$D_{k+1} = D_k + \frac{(\rho_k^T \mu_k - \rho_k^T D_k \rho_k)}{Tr(H_k^2)} H_k, \tag{4}$$

where $H_k = \text{diag}((\rho_k^{(1)})^2, (\rho_k^{(2)})^2, \ldots, (\rho_k^{(n)})^2)$, $\sum_{i=1}^{n} (\rho_k^{(i)})^4 = Tr(H_k^2)$ and Tr is the trace operation. By letting $\nu_2 = 0$, ν_0 and ν_1 can be by computed as follows:

$$\begin{aligned} -\nu_1 &= \nu_2 - \nu_1 \\ &= \| x(\nu_2) - x(\nu_1) \|_{D_k} \\ &= \| x_{k+1} - x_k \|_{D_k} \\ &= \| s_k \|_{D_k} \\ &= (s_k^T D_k s_k)^{\frac{1}{2}}, \end{aligned} \tag{5}$$

and

$$\begin{aligned} -\nu_0 &= \nu_2 - \nu_0 \\ &= \| x(\nu_2) - x(\nu_0) \|_{D_k} \\ &= \| x_{k+1} - x_{k-1} \|_{D_k} \\ &= \| s_k + s_{k-1} \|_{D_k} \\ &= ((s_k + s_{k-1})^T D_k (s_k + s_{k-1}))^{\frac{1}{2}}. \end{aligned} \tag{6}$$

Waziri et al. [16] incorporated the attribute of the multi-step approach to improve over single step method. Nevertheless, they use diagonal updating matrix D_k to represent the weighted matrix that is use to parameterize the interpolating polynomials in every iteration. The numerical performance of the multi-step approach is highly promising. Based on this fact, it is pleasing to present an approach which will improve further the Jacobian approximation, as well as reducing the computational cost and floating points operations of constructing the interpolating curves. This is made possible by setting identity matrix I_n as the weighted matrix to be used in computing the metric in every iteration. This is what lead to the idea of this paper which is organized as follows: In the next section, we present the details of the proposed method.

Some numerical results are reported in Section 3. Finally, conclusions are made in Section 4.

2. Derivation Process

This section presents a new diagonal quasi-Newton like method for solving large-scale systems of nonlinear equations. Our main concerns is to employ a new strategy of obtaining the weighted matrix than the one proposed by Waziri et al. [16]. The anticipation has been to reduce the computational cost, matrix storage requirement and floating points operations of constructing the interpolation curves, as well as reducing overall CPU time in general. In parameterizing the interpolation polynomials, Waziri et al. [16] choice to use the updating matrix D_k as the weighted matrix. In this paper, we proposed to use a less storage, low computation and weighed matrix free approach in contracting the interpolation curves. This is made possible by choosing the weighted matrix to be an identity matrix in every iteration so that ρ_k, μ_k and β can be obtaining more cheaply than Waziri et al. [16] and still maintaining a higher order approximation of the Jacobian matrix in diagonal form. To this end, we define the weighting matrix in our own approach as I_n. (In Waziri et al. [16] the weighting matrix is set as D_k). Hence, (5) and (6) can be transformed to

$$-\nu_1 = (s_k^T I_n s_k)^{\frac{1}{2}}, \tag{7}$$

and

$$-\nu_0 = ((s_k + s_{k-1})^T I_n (s_k + s_{k-1}))^{\frac{1}{2}}. \tag{8}$$

Based on the properties of identity matrix, it follows from (7) and (8) that

$$-\nu_1 = (s_k^T s_k)^{\frac{1}{2}}, \tag{9}$$

$$-\nu_0 = ((s_k + s_{k-1})^T (s_k + s_{k-1}))^{\frac{1}{2}}, \tag{10}$$

$$\rho_k = s_k - \frac{\beta^2}{1 + 2\beta} s_{k-1}, \tag{11}$$

$$\mu_k = y_k - \frac{\beta^2}{1+2\beta} y_{k-1}, \tag{12}$$

where β denoted by

$$\beta = \frac{\nu_2 - \nu_0}{\nu_1 - \nu_0}, \tag{13}$$

and $-v_2 = 0$ respectively. One can see that, it is very cheaply to obtain equations (9)-(13) than to use the approach proposed by Waziri et al. [16]. To this end, the updating formula for D is given as follows :

$$D_{k+1} = D_k + \frac{(\rho_k^T \mu_k - \rho_k^T D_k \rho_k)}{Tr(G_k^2)} G_k, \tag{14}$$

where $G_k = \text{diag}((\rho_k^{(1)})^2, (\rho_k^{(2)})^2, \ldots, (\rho_k^{(n)})^2)$, $\sum_{i=1}^n (\rho_k^{(i)})^4 = Tr(G_k^2)$ and Tr is the trace operation.

To safeguard on the possibly of generating undefined D_{k+1}, the proposed updating scheme for D_{k+1} is used whenever :

$$D_{k+1} = \begin{cases} D_k + \dfrac{(\rho_k^T \mu_k - \rho_k^T Q_k \rho_k)}{Tr(G_k^2)} G_k; & \| \rho_k \| > 10^{-4}, \\ D_k; & \text{otherwise.} \end{cases}$$

Hence, the stages of the proposed method are presented as follows:

Algorithm (2-DQNM).

Step 1. Choose an initial guess x_0 and $D_0 = I$, let $k := 0$.

Step 2. Compute $F(x_k)$. If $\| F(x_k) \| \leq \epsilon_1$ stop, where $\epsilon_1 = 10^{-4}$.

Step 3. If $k := 0$ define $x_1 = x_0 - D_0^{-1} F(x_0)$. Else if $k := 1$ set $\rho_k = s_k$ and $\mu_k = y_k$ and go to 5.

Step 4. If $k \geq 2$ compute ν_1, ν_0 and β via (9), (10) and (13), respectively and find ρ_k and μ_k using (11) and (12), respectively. If $\rho_k^T \mu_k \leq 10^{-4} \| \rho_k \|_2 \| \mu_k \|_2$ set $\rho_k = s_k$ and $\mu_k = y_k$.

Else retains the ρ_k and μ_k computed using (11) and (12),

Step 5. Let $x_{k+1} = x_k - D_k^{-1} F(x_k)$ and update D_{k+1} as define by (14)

Step 6. Check if $\| \rho_k \|_2 \geq \epsilon_1$

if yes retain D_{k+1} that is computed by step 5. Else set, $D_{k+1} = D_k$

Step 7. Set $k := k + 1$ and go to 2.

3. Numerical Results

In this section, the performance of 2-DQNM has been illustrated, when compared with MFDN method proposed by Leong et al. [12], 2-MFDN presented by Waziri et al. [16], Broyden Method (BM) and Newton's method (NM) respectively. The codes are written in MATLAB 7.0 with a double precision computer, the stopping criterion used is:

$$\| \rho_k \| + \| F(x_k) \| \leq 10^{-4}. \tag{15}$$

The identity matrix has been chosen as an initial approximate Jacobian. Six (8) different dimensions are performed on the benchmark problems ranging from 50 to 500000. The codes also terminates whenever one of the following happens;

(i) The number of iteration is at least 600 but no point of x_k that satisfies (15) is obtained;

(ii) CPU time in second reaches 600,

(iii) Insufficient memory to initial the run.

The performance of these methods will be compared in terms of number of iterations and CPU time in seconds. We present all the results using performance profiles indices in terms of robustness, efficiency and combined robustness and efficiency as proposed in [4]. In the following, some details on the benchmarks test problems are presented.

Problem 1. Trigonometric System of Byeong [11] :

$$f_i(x) = \cos(x_i) - 1 \quad i = 1, 2, \ldots, n \quad \text{and} \quad x_0 = (0.87, 0.87, \ldots, 0.87)$$

Problem 2 [16]**:**

$$f_i(x) = \ln(x_i)\cos((1-(1+(x^T x)^2)^{-1}))\exp((1-(1+(x^T x)^2)^{-1})) \quad i = 1, 2, \ldots, n,$$

and $x_0 = (2.5, 2.5, \ldots, 2.5)$.

Problem 3. Spares System of Byeong [5] :

$$f_i(x) = x_i x_{i+1} - 1$$

$$f_n(x) = x_n x_1 - 1, \quad i = 1, 2, \ldots, n-1 \text{ and } x_0 = (0.5, 0.5, \ldots, 0.5).$$

Problem 4 [16]**:**

$$f_i(x) = n(x_i - 3)^2 + \frac{\cos(x_i - 3)}{2} - \frac{x_i - 2}{\exp(x_i - 3) + \log(x_i^2 + 1)}$$

$$i = 1, 2, \ldots, n \text{ and } x_0 = (-3, -3, -3, \ldots, -3)$$

Problem 5 [14]**:**

$$f_1(x) = x_1$$

$$f_i(x) = \cos x_{i+1} + x_i - 1, \; i = 2, 3, \ldots, n \text{ and } x_0 = (0.5, 0.5, \ldots, 0.5).$$

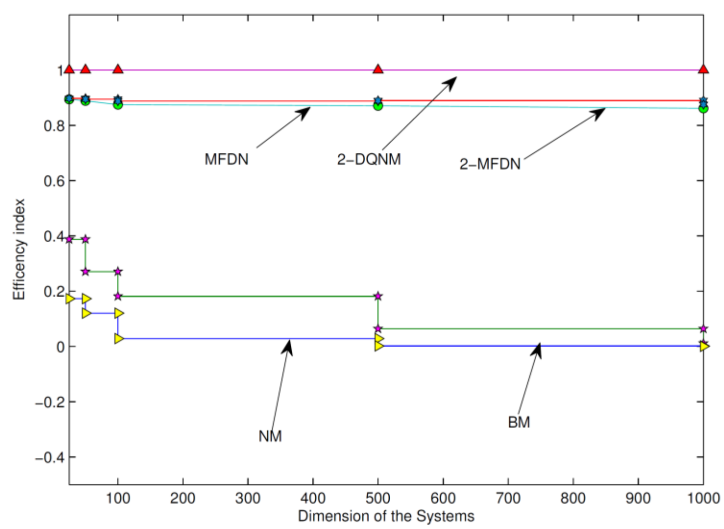

Figure 1. Efficiency Profile of NM, BM, MFDN, 2-MFDN and 2-QDNM methods as the dimensions increase (in term of CPU time).

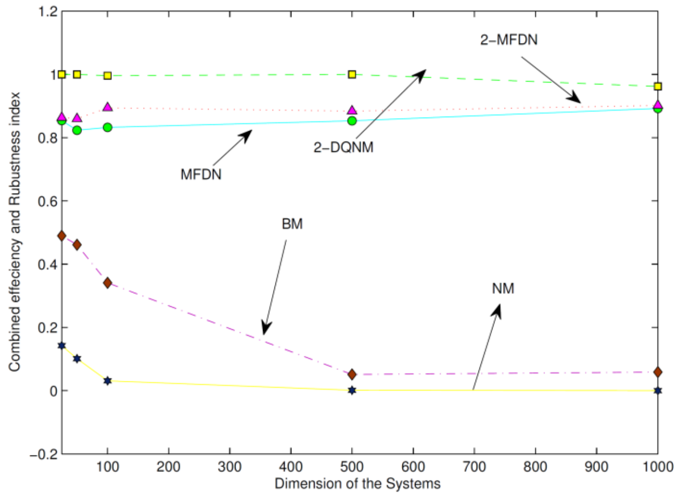

Figure 2. Combined Efficiency and Robustness profile of NM, BM, MFDN, 2-MFDN and 2-DQNM methods as the dimensions increase (in term of CPU time).

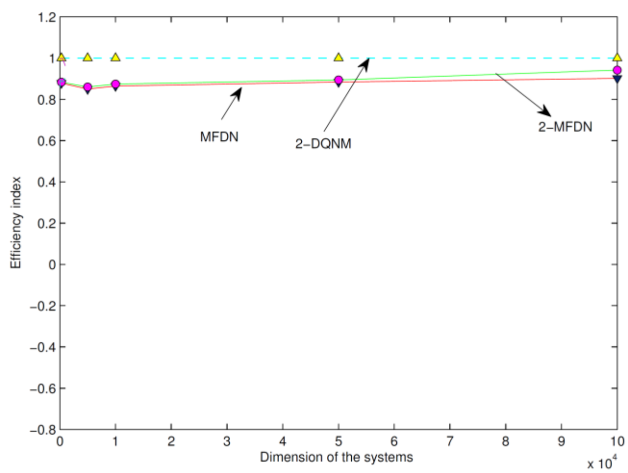

Figure 3. Efficiency profile of MFDN, 2-MFDN and 2-DQNM methods as the Dimension > 1000 (in term of CPU time).

From Figures 1 and 2, it is clear that 2-DQNM method outperform the other four methods in terms of CPU time in seconds. However, the efficiency and combined efficiency and Robustness indices demonstrate that, as the dimensions of the systems increase say 300,000, 2-DQNM method still maintains the best indices. To present the improvement of our approach over 2-MFDN method as dimensions increase, Figure 3 illustrates that 2-DQNM method is the best. Note that, the efficiency of 2-DQNM method is 26% more than that of MFDN method.

In addition, it is worth mentioning that, 2-DQNM method has improved the numerical outcomes of the 2-MFDN method and still keeping memory requirement and CPU time in seconds to only linear order.

4. Conclusions

This paper presents a new diagonal quasi-Newton update for solving large-scale system of nonlinear equations (2-DQNM). The method employs a two-step 3-points scheme to update the Jacobian approximation into a nonsingular diagonal matrix and uses identity matrix as the weighted matrix in each iterations, unlike the Waziri et al. [16] that required to compute and store the diagonal matrix in every iterations. The anticipation behind this approach is to reduce the computational burdens, storage requirement and floating points operations of constructing the interpolation curves. Hence, from the numerical results reported, we can conclude that, 2-DQNM is very encouraging compared to NM, BM, MFDN and 2-MFDN methods in handling large-scale systems of nonlinear equations.

References

[1] I. K. Argyros, S. K. Khattri and S. Hilout, Expanding the applicability of inexact Newton methods under Smale's (α, γ)- theory, Applied Mathematics and Computation 224 (2013), 224-237.

[2] I. K. Argyros and S. K. Khattri, On the Secant method, J. Complexity 29(6) (2013), 454-471.

[3] I. K. Argyros and S. K. Khattri, A new convergence analysis for the two-step Newton method of order three, Proyecciones J. Math. 32(1) (2013), 73-90.

[4] I. Bogle and J. D. Perkins, A new sparsity preserving quasi-Newton update for solving nonlinear equations, SIAM J. Sci. Statist. 11 (1990), 621-630.

[5] 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, Appl. Math. Comput. 217 (2010), 3190-3198.

[6] J. E. Dennis, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[7] M. Farid, W. J. Leong and M. A. Hassan, A new two-step gradient-type method for large scale unconstrained optimization, J. Comput. Math. Appl. 59 (2010), 3301-3307.

[8] J. A. Ford and L. A. Moghrabi, Multi-step quasi-Newton methods for optimization, J. Comput. Appl. Math. 50 (1994), 305-323.

[9] J. A. Ford and L. A. Moghrabi, Alternating multi-step quasi-Newton methods for unconstrained optimization, J. Comput. Appl. Math. 82 (1997), 105-116.

[10] J. A. Ford and S. Thrmlikit, New implicite updates in multi-step quasi-Newton methods for unconstrained optimization, J. Comput. Appl. Math. 152 (2003), 133-146.

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

[12] W. J. Leong, M. A Hassan and M. Y. Waziri, A matrix-free quasi-Newton method for solving large-scale nonlinear systems, Comput. Math. App. 62(5) (2011), 2354-2363.

[13] K. Natasa and L. Zorna, Newton-like method with modification of the right-hand vector, J. Maths. Compt. 71 (2001), 237-250.

[14] A. Roose, V. L. M. Kulla and T. Meressoo, Test examples of systems of nonlinear equations, Tallin: Estonian Software and Computer Service Company, Estonia 1990.

[15] M. Y. Waziri, W. J. Leong, M. A. Hassan and M. Monsi, Mathematical Problems in Engineering, Article ID 467017(2011), 12 pages.

[16] M. Y. Waziri, W. J. Leong and M. Mamat, A two-step matrix-free secant method for solving large-scale systems of nonlinear equations, Journal of Applied Mathematics (2012), 9 pages, doi:10.1155/2012/348654.