VOL. 7, NO. 4, APRIL 2012 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences


www.arpnjournals.com

A MODIFIED CHORD NEWTON'S METHOD FOR HIGH-DIMENSIONAL ALGEBRAIC EQUATIONS

M. Y. Waziri[1], H. Aisha[1] and I. Saidu[2]
[1]Department of Mathematics, Faculty of Science, Bayero University Kano (BUK), Kano, Nigeria
[2]Faculty of Computer Science, Universiti Putra Malaysia, Malaysia
E-Mail: mywaziri@gmail.com

ABSTRACT

There is a great deal of interest on reducing overall computational budget of classical Newton's method for solving nonlinear systems of equations. The appealing approach is based on chord Newton's but the method mostly requires high number of iteration as the dimension of the systems increases. In this paper, we introduce a new procedure that will reduce the well known shortcoming of Newton method. Our approach was implemented on some benchmarks nonlinear systems which show that, the proposed method is very encouraging.

Keywords: chord Newton's method, systems, algebraic.

1. INTRODUCTION

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

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

is continuously differentiable in an open neighborhood S of a solution $x^* \in S$ of the system (1.1). Assume that, there exists a solution x^* where $F(x^*) = 0$ and $F'(x^*) \neq 0$. The promising method for handling such a problem is the classical Newton's method [1], in which the Newtonian iterates are generated by:

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

where $F'(x_k)$ is the Jacobian of F. The attractive feature of this method is the convergence properties of it, i.e., whenever the Jacobian matrix $F'(x_k)$ is non-singular at a solution of (1), the method exhibits quadratic rate of converges from any initial point x_0 in the neighborhood of x^* [1], i.e.

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

for some *h*.

The computational budget of Newton's method mostly becomes more expensive, particularly as the dimension of the nonlinear systems increases due to computation and storage of Jacobian matrix in every iteration. In practice some derivatives are highly costly to obtain or cannot be done precisely, in this case Newton's method could not be a good candidate [1, 2, 3, 4]. Many efforts have been made, by a substantial number of researchers to overcome the well know disadvantage of Newton's method. The simplified and easiest variant of Newton method is Newton's chord method. Newton's chord method eliminates some well-know shortcomings of Newton's method, i.e., computation and storage of Jacobian matrix in every iteration. The method computes only one Jacobian at all. Moreover, from any starting point x_0 in the neighborhood of the solution x, Newton's Chord method generates an iterative point via

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

Despite its good quality, Newton's chord method mostly required more number of iteration and the convergence may even lost because of less Jacobian information in each iteration [3, 5, 6, 7]. In fact, most of the variants of Newton's chord method do not work perfectly. In this paper we propose a simple modification of Chord Newton's method for nonlinear systems, by computing subsequent Jacobian when the number of iteration is beyond ten (10). The main idea behind this task is that, we aim at reducing the number of iteration and to correct the convergence property of Chord Newton's method. The method proposed in this paper is significantly cheaper than classical Newton's method and faster than Newton's Chord method with respect to CPU time in general. We organize the paper as follows: Section 2 presents the new variant of Chord Newton's method. Some numerical results and discussion are given in section 3, and finally conclusion is reported in section 4.

2. DERIVATION PROCESS

In this section, we shall present our new modification of Newton's chord method. The proposed method generates a sequence of points $\{x_k\}$ using:

$$x_{k+1} = x_k - \Omega_k^{-1}F(x_k), \tag{5}$$

where Ω_k is a Jacobian matrix which can be updated at a certain point. In [7] and [6] it has been shown that the undesirable performance behaviors of Newton's chord method especially when solving high dimensional systems of nonlinear equations is associated with the insufficient Jacobian information in each iteration. The validation

associated to our procedure is to improve the Jacobian information and to the convergence properties as well. This is made possible by computing subsequent Jacobian whenever the number of iteration is more than 10, unlike the Newton's chord method which computes Jacobian once. With this scheme, we expect our method will yield a significant reduction in CPU time consumption and number of iteration compared to Newton's Chord method. To establish a substantial number of iterations and Jacobian information, as well as for good numerical stability we propose to compute the subsequent Jacobian when $n_m > 10$ where n_m is a number of iteration. The scheme terminates at:

$$\|F(x_k)\| \leq 10^{-4} \tag{6}$$

Finally, we present the algorithm and the general safeguard condition of the proposed method as follows:

Algorithm MCNM (A Modified Chord Newton's Method)

For k = 0, 1, 2….. where $F'(x_k)$ is the Jacobian matrix of F.

Step 1: Compute $F(x_k)$ and $F'(x_0)$

Step 2: Solve $F'(x_0)s_k = -F(x_k)$

Step 3: Update $x_{k+1} = x_k + s_k$

Step 4: If $\|F(x_k)\| \leq 10^{-4}$ stop. Else go to step 5

Step 5: Check whether k > 10

If yes set $k = k+1$, Compute New Jacobean M_k, set $F'(x_0) = M_k$ and go to step 2

Else set $k = k+1$ and go to step 2.

3. NUMERICAL EXPERIMENT

This section presents numerical result of the method proposed in this paper (MNCM) as compared with classical Newton's method (NM) and Newton's Chord method (NC) on some 4 benchmark algebraic equations. We consider the dimension of the systems from 25 to 1000 variables. The comparison was based on number of iteration and CPU time respectively. Based on comparison indices presented in [8], we reported on robustness, efficiency and combined robustness and efficiency of NM, CN, BM and MCNM methods, respectively. The codes were made using MATLAB 7.0 and implemented with a double precision computer. The stopping criterion used is:

$$\|F(x_k)\| \leq 10^{-4}.$$

In the following we illustrate some details on the benchmarks used test problems.

Problem 1: System of n nonlinear equations:

$$f_i(x) = \left(3x_i - x_{i+1}^2\right)\tfrac{1}{n+1} + \sinh x_i \sinh x_i$$

$$i = 1, 2, \ldots, \quad n, \quad x_0 = (-1.2, -1.2, \ldots)^T$$

Problem 2: System of n nonlinear equations [6]:

$$f_i(x) = n - \textstyle\sum_{j=1}^{n} \cos x_j + i(1 - \cos x_i) - \sin x_i,$$

$$i = 1, ..., n \text{ and } \quad x_0 = (\tfrac{1}{n}).$$

Problem 3: Trigonometric system:

$$f_1(x) = x_1^2 - 3x_1 + 1 + \cos(x_1 - x_2),$$

$$f_i(x) = x_i^2 - 3x_i + 1 + \cos(x_i - x_{i-1}),$$

$$f_n(x) = \exp^{(1-x_n)} - \cos(x_n - x_{n-1}),$$

$$x_i^0 = 0, i = 2, \ldots, n-1.$$

Problem 4: System of n nonlinear equations [7]:

$$f_i(x) = (1 - x_i^2) + x_i(1 + x_i x_{n-2} x_{n-1} x_n) - 2$$

$$i = 1, 2, \ldots, n, \quad x_0 = (2, 2, ...)^T$$

Our numerical results are graphically presented in Figures 1 and 2; from the figures we observe that MNCM method is highly superior to the NM and NC methods with respect to the benchmarks problems. Moreover, we see that MNCM method needs lesser CPU time than NM and NC methods, this can be explained by the substantial number of Jacobian computation and least floating points operations for our method MNCM. In fact, we observed that, MNCM method's efficiency and combined efficiency and robustness indices CPU time growth linearly whereas for NM and NC method's increases exponential as the systems increases.

Hence, we can conclude that MNCM method could be a good method for handling large-scale system of nonlinear equations, particularly extreme large systems.

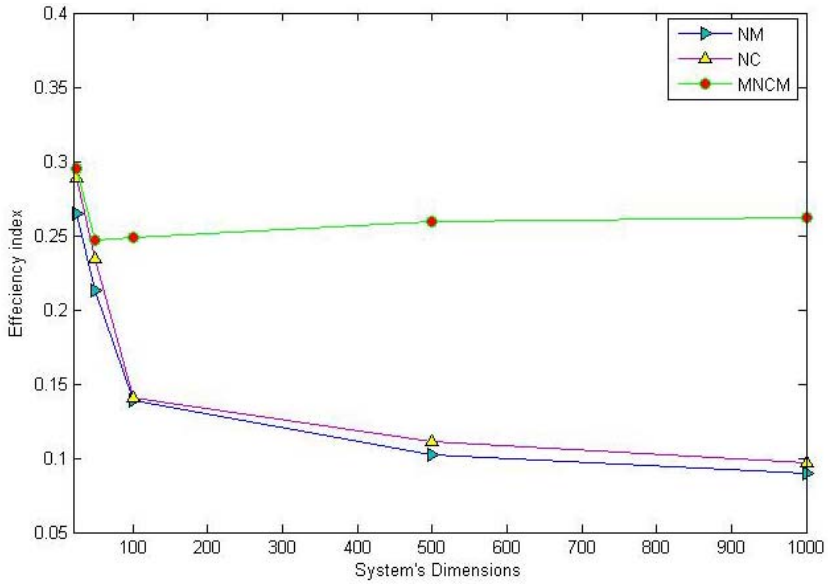

Figure-1. Efficiency profile of NM, CN, MNCM methods as the dimensions increases for solving the problems (in terms of CPU time).

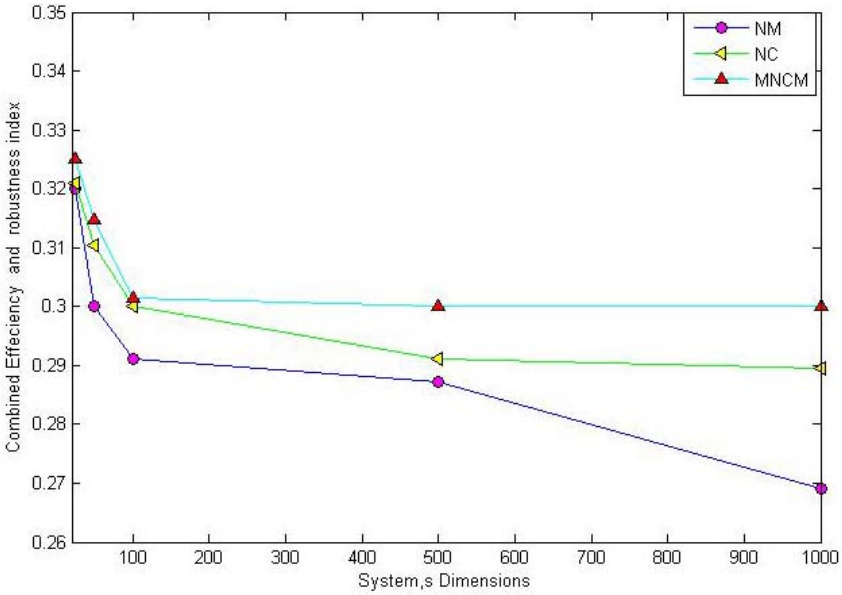

Figure-2. Combined efficiency and robustness profile of NM, CN, MNCM methods as the dimensions increases for solving the problems (in term of CPU time).

4. CONCLUSIONS

This paper presents a new approach on Chord Newton's method for system of nonlinear equations (MNCM). The numerical results profile presented shown obviously that the MNCM method is significant efficient than Chord Newton's and Newton's method respectively. It is worth mentioning that the method has reduced the CPU time compared to some variants of Newton's method as having the highest efficiency and combined efficiency and robustness profile respectively, especially when dimension increases.

REFERENCES

[1] Dennis J. E. 1983. Numerical methods for unconstrained optimization and nonlinear equations. Prince-Hall, Inc., Englewood Cli s, New Jersey, USA.

[2] Waziri M. Y., Leong W.J., Hassan M.A. and Monsi M. 2010. An efficient solver for systems of Non-Linear equations with singular Jacobian via diagonal updating. Applied mathematical sciences. 4(69): 3403-3412.

[3] 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.

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

[5] Waziri M.Y., W.J. Leong and M. A. Hassan. 2011. Jacobian-free diagonal Newton's method for solving nonlinear systems with singular Jacobian. Malaysian Journal of Mathematical Sciences. 5(2): 241-255.

[6] Spedicato E. 1975. Computational experience with quasi-Newton algorithms for minimization problems of moderately large size. Rep. CISE-N-175. 3: 10-41.

[7] Waziri M.Y., W.J. Leong and M. A. Hassan. 2012. Diagonal Broyden- like Method for Large-scale Systems of Nonlinear Equations. Malaysian Journal of Mathematical Sciences. 6(1): 59-73.

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