A Modified Shamanskii Accelerated Scheme Via Diagonal Jacobian Approximation for Nonlinear Equations

IBRAHIM MOHAMMED SULAIMAN, MUSTAFA MAMAT *
Faculty of Informatics and Computing
Universiti Sultan Zainal Abidin
Terengganu, Malaysia.
sulaimanib@unisza.edu.my, *must@unisza.edu.my

Abstract: - This paper presents a variation of Newton's method, based on a diagonal Jacobian approximation scheme on an accelerated Shamanskii process for systems of nonlinear equations especially on problems with singular Fréchet derivative at the solution points. This method aims to reduce the computation cost and storage requirements as in Newton-type methods. Numerical results are presented to illustrate the efficiency of the proposed scheme.

Key-Words: - Shamanskii method, Diagonal Jacobian approximation, Nonlinear equations,

1 Introduction

Consider the system of nonlinear equations of the form:

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

where F is continuously differentiable from a Banach space E into itself and the point $x^* \in E$ is the solution of the function $F(x^*) = 0$. These systems (1), are often solved by Newton-type methods. The most well-known method being the classical Newton's method is computed via the following scheme.

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

Supposed the initial point x_0 is sufficiently chosen near the solution point x^*, then Newton's method would converge quadratically, i.e.,

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

for some K_c, provided $F'(x^*)$ is nonsingular [2], [11]. Some of the drawbacks of this method include; evaluating and storing the Jacobian matrix at every iteration, solving the systems of n linear equations for $s(x_k) = -F'(x_k)^{-1}F(x_k)$, and time consumption due to increase in equation dimension which makes the method very expensive. To overcome these lapses, a variation of Newton method has been proposed. This includes; the Chord Newton method, which computes the Jacobian matrix $F'(x_0)^{-1}$ only once for finite dimensional problems [3]. This method generates its sequence $\{x_k\}$ of iterate as follows;

$$x_{k+1} = x_k - F'(x_0)^{-1}F(x_k) \tag{5}$$

The strategy reduces the computational cost at each iteration. However, the convergence rate of the Chord method is reduced to linear, i.e.,

$$\|x_{k+1} - x^*\| \le K_c\|x_0 - x^*\|\|x_k - x^*\| \tag{6}$$

which continue to improve with an improvement of the initial points [1].

Motivated by the low cost of Jacobian computation of the Chord method and rapid convergence of Newton's method, Shamanskii [7], developed a multiple pseudo-Newton iteration scheme that lies between the chord and Newton's methods. Numerous researchers such as [4], [5], [6], [10] have analysed this method in details. This method is described as follows

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

$$x_{k+1} = x_{k+\frac{1}{2}} - F'(x_k)^{-1}F\left(x_{k+\frac{1}{2}}\right)$$

which can be rewritten as

$$x_{k+1} = x_k - F'(x_k)^{-1}Q \tag{7}$$

$$Q = [F(x_k) + F(x_k - F'(x_k)^{-1}F(x_k))].$$

The shamanskii method is regarded as very efficient as it overcomes the complexity encountered by other iterative methods by using only one factored Jacobian for computing more than one pseudo-

Newton's iterates. The convergence of the shamanskii method is and easy aftermath of (5) and the linear convergence (6) of the Chord method. Supposed the initial point x_0 is sufficiently chosen near the solution point x^*, then this method would converge m-step g-superlinearly with g-order of at least $m+1$. [6], i.e., there exist some $K_s > 0$ such that

$$\|x_{k+1} - x^*\| \leq K_s \|x_k - x^*\|^{m+1} \quad (8)$$

provided $F'(x^*)$ is nonsingular [2]. The convergence analysis of the considered shamanskii method has been established.

Theorem 1 [8]. Suppose $F: D \subset R^n \to R^n$ conform to hypotheses $H1(2)$, $H2$, and $H3$. Then, the solution point x^* is referred to the attraction point of the Shamanskii process defined in (7) with at least $m+1$ order of convergence.

For further reading on the Shamanskii method and its convergence, please refer to [5], [6], [12], [13].

The purpose of this paper is to consider cases of where the Jacobian $F'(x_k)$ is singular. The next section presents the method derivation and reports the numerical results in section 3. The conclusion followed by a discussion of the results is presented in section 4.

2 Method Formulation

A Shamanskii-like process with diagonal Jacobian approximation:

Given a system of nonlinear equation (1), we consider its Taylor's expansion at the point x_k, i.e.,

$$F(x) = F(x_k) + F'(x_k)(x - x_k) + (o\|x - x_k\|^2) \quad (9)$$

We further defined the incomplete expansion of the nonlinear function (1) by the Taylors series as follows:

$$\hat{F}(x) = F(x_k) + F'(x_k)(x - x_k) + (o\|x - x_k\|^2) \quad (10)$$

where $F'(x_k) = J_F(x_k)$ denotes the Jacobian matrix of (1) as the point x_k.

In order to apply the accurate information needed on the Jacobian to the updating matrix, from (10), we enforce some conditions as follows.

$$\hat{F}(x_{k+1}) = F(x_{k+1}) \quad (11)$$

Applying (11) to (10), we have

$$F(x_{k+1}) \approx F(x_k) + F'(x_k)(x_{k+1} - x_k) \quad (12)$$

which implies

$$F'(x_k)(x_{k+1} - x_k) \approx F(x_k) - F(x_{k+1}) \quad (13)$$

Now, we apply the above derivation to propose the Jacobian approximation of $F'(x_k)$. Let

$$D_k \approx F'(x_k) \quad (14)$$

be the diagonal matrix that would be updated during the iteration process. Substituting (14) in (13), we have:

$$D_{k+1}(x_{k+1} - x_k) \approx F(x_k) - F(x_{k+1}) \quad (15)$$

We derived the components of the diagonal matrix, i.e., $D = diag(d^1, d^2, \ldots, d^n)$ as follows. From (15), we have

$$D_{k+1} = \frac{F(x_k) - F(x_{k+1})}{(x_{k+1} - x_k)} \quad (16)$$

which implies

$$d_{k+1}^{(i)} = \frac{F_i(x_k) - F_i(x_{k+1})}{x_{k+1}^{(i)} - x_k^{(i)}} \quad (17)$$

Hence, the diagonal element of the diagonal matrix is

$$D_{k+1} = \text{diag}(d_{k+1}^{(i)}) \quad (18)$$

For $i = 1,2, \ldots, n$ and $k = 0, 1, \ldots, n$.

Diagonal components defined in (18) can be used if the denominator is negligible i.e., $\left|x_{k+1}^{(i)} - x_k^{(i)}\right| > 10^{-8}$. Otherwise, we set $d_k^{(i)} = d_{k-1}^{(i)}$.

We present the proposed diagonal update to the Shamanskii method (MSDM) and its Algorithm as follows.

$$x_{k+1} = x_k - D_k^{-1}\left[F(x_k) + F\left(x_k - D_k^{-1}F(x_k)\right)\right] \quad (19)$$

Algorithm 1 (MSDM)
Consider the systems of nonlinear equation defined in (1).
Step 1. Given an initial guess x_0 and $D_0 = I_n$, set $k = 0$.
Step 2. Compute for $F(x_k)$
Step 3. Compute for D_k by (18).
Step 4. Update new iterate using (19).
Step 5. Convergence and stopping criteria
Check if $\|x_{k+1} - x_k\| + \|F(x_k)\| \leq 10^{-8}$, stop. Else, go to step 2 and set $k = k + 1$.

3 Results

In this section, we present some numerical computations based on number of iterations, CPU time, and storage requirements to illustrate the theoretical analysis presented above. We considered the classical methods of Shamankii (SM), Chord (CNM), and Newton (NM), respectively for comparison to demonstrate the performance of the proposed (MSDM) method. All problems are taken from Waziri et al., [11], Shin et al., [14], and More et al. [15]. The codes were implemented on MATLAB (2018b) subroutine programing. The problems employed includes small scale problems and large-scale problems with either dense or sparse Jacobian. The termination criterion for this computation is $\|x_{k+1} - x_k\| + \|F(x_k)\| \leq 10^{-8}$, and x_{k+1} is the final iterate. We denote dimension with "Dim" through the paper.

Problem 1: Structured Exponential function

$$F_i(x) = x_i - 0.1x_{i+1}^2$$
$$F_n(x) = x_n - 0.1x_n^2$$
$$i = 1, 2, 3, \dots, n-1, \quad x_0 = (0.05, 0.05, \dots, 0.05)$$

Problem 2: Structured Exponential function

$$F_i(x) = x_i^2 - 1$$
$$F_n(x) = x_n - 0.1x_n^2$$
$$i = 1, 2, 3, \dots, n, \quad x_0 = (0.05, 0.05, \dots, 0.05)$$

Problem 3: Extended Trigonometric of Byeong-Chun

$$F_i(x) = \cos(x_i^2 - 1) - 1$$
$$i = 1, 2, 3, \dots, n, \quad x_0 = (0.06, 0.06, \dots, 0.06)$$

Problem 4: Extended Spares System of Byeong

$$F_i(x) = x_i - \sum_{i=1}^{n} \frac{x_i^2}{n^2} + \sum_{i=1}^{n} x_i - n$$
$$i = 1, 2, 3, \dots, n, \quad x_0 = (1.1, 11.1, \dots, 1.1)$$

Problem 5: System of n Nonlinear equations

$$F_i(x) = (1 - x_i^2) + x_i(1 + x_i x_{n-2} x_{n-1} x_n) - 2$$
$$i = 1, 2, 3, \dots, n, \quad x_0 = (0.3, 0, 3, \dots, 0.3)$$

Table 1: Results of Problem 1 base on iteration number/CPU time

Dim	NM	CNM	SM	DSM
25	3/0.0313	5/0.0497	4/0.0156	4/0.0313
50	3/0.0313	6/0.0552	4/0.0313	4/0.0456
100	3/0.0625	7/0.0313	4/0.0469	4/0.0313
500	3/0.2344	8/0.3125	4/0.2969	4/0.4844
1000	3/0.7344	8/1.1875	4/0.9375	4/4.5781
5000	*	8/64.4219	4/43.6250	4/46.371

Table 2: Results of Problem 2 base on iteration number/CPU time

Dim	NM	CNM	SM	MDSM
25	10/ 0.0156	4/ 0.0156	11/ 0.0156	8/ 0.0062
50	10/ 0.0313	4/ 0.0109	11/ 0.0313	8/ 0.0079
100	10/ 0.1250	4/ 0.0469	11/ 0.1250	8/ 0.0210
500	10/ 0.6250	4/ 0.1719	11/ 0.6406	8/ 0.9063
1000	10/ 2.2188	4/ 0.6875	11/ 2.2813	8/ 7.7813
5000	*	4/ 32.328	11/ 98.2031	8/87.1525

Table 3: Results of Problem 3 base on iteration number/CPU time

Dimension	NM	CNM	SM	DSM
25	12/0.0255	6/0.0313	12/0.0313	4/0.0025
50	12/0.0469	6/0.0156	12/0.0469	4/0.0050
100	12/0.1094	6/0.0156	12/0.1250	4/0.0313
500	12/1.1563	6/0.2188	12/1.0469	4/0.4063
1000	12/1.1563	6/0.9688	12/3.9219	4/4.2031
5000	*	6/50.7031	12/149.343	4/92.7452

Table 4: Results of Problem 4 base on iteration number/CPU time

Dim	NM	CNM	SM	DSM
25	17/0.0262	52/0.0103	18/0.0239	11/0.0058
50	21/0.0781	78/0.0156	22/0.0625	10/0.0123
100	26/0.3750	142/0.1719	27/0.3594	9/0.0237
500	43/2.8125	*	44/2.8906	9/1.0156
1000	49/10.4219	*	50/10.6875	9/8.5156
5000	*	*	*	8/480.8347

Table 5: Results of Problem 5 base on iteration number/CPU time

Dim	NM	CNM	SM	DSM
25	12/0.0469	4/0.0088	13/0.0222	9/0.0048
50	12/0.0313	4/0.0156	13/0.0469	9/0.0117
100	12/0.1875	4/0.0313	13/0.1563	9/0.0469
500	12/0.8438	4/0.1719	13/0.8594	9/0.9531
1000	12/2.6563	4/0.6250	13/2.8750	9/8.6719
5000	*	4/33.1094	13/122.625	9/71.2349

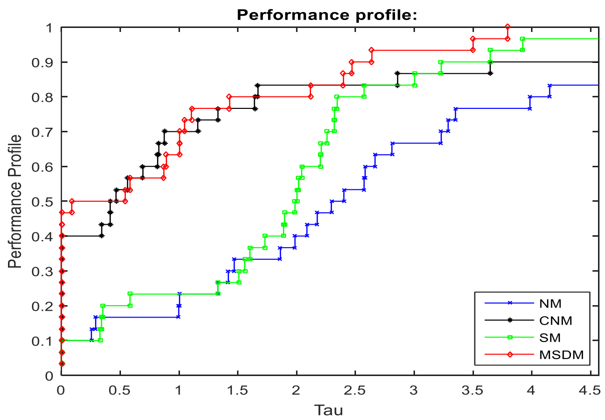

Figure 1: Performance based on number of Iterations

The performance of Newton-type methods frequently degrades with the Jacobian matrix been singular. Tables 1 to Table 5 above presents the obtained results used to estimate the zeros of the given nonlinear functions using the classical Newton's method, the Chord Newton method, classical Shamanskii method, and the proposed Modified Shamanskii method. We observed the proposed MSDM uses less computational cost than the classical methods employed. This is due to the diagonal approximation scheme derived which makes the method cheaper compared to other methods.

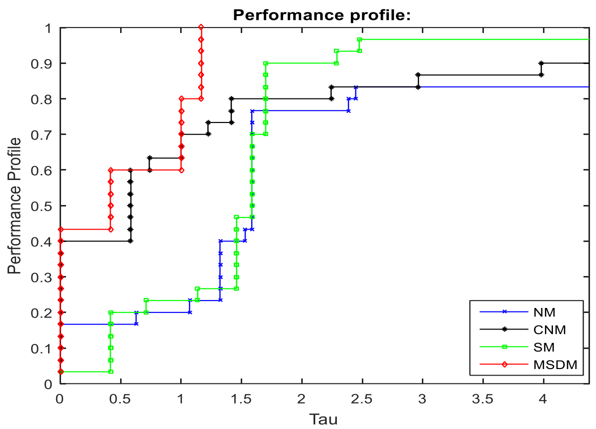

Figure 2: Performance based on CPU

Also, from Figures 1 and 2, it can be seen that the performance of the proposed method is better in terms of number of iterations and CPU time than the other existing methods employed. Another advantage of the proposed method is the ability to bypass the point at which the Jacobian is singular as in the case of problem 4. Results have shown that the proposed method is an improvement with respect to matrix storage, computational cost and time.

4 Conclusion

The main aim of this paper is proposing a `new technique using the shamanskii algorithm accelerated by a diagonal Jacobian approximation procedure. The goal of the new method is to improve the complexity and convergent rate of existing methods, while reducing the computational cost and storage requirement at each iteration. With a diagonal updating scheme to the shamanskii method, we avoid the high and perilous computation of the Jacobian inverse during the iteration process. Numerical results reported has shown that the proposed method is promising.

References:

[1] Ezquerro, J., Grau-Sánchez, M., Grau, A., Hernández, M., Noguera, M., Romero, N.: On iterative methods with accelerated convergence for solving systems of nonlinear equations. *J. Optim. Theory Appl.* Vol 151, No. 1, 2011, pp. 163–174.

[2] Kelley C.T. A Shamanskii-Like Acceleration Scheme for Nonlinear Equations at Singular Roots. *Mathematics of Computation*. Vol 47, 1986, 609-623.

[3] Kelley C.T. *Iterative Methods for Linear and Nonlinear Equations",* SIAM, Philadelphia, PA, 1995.

[4] Brent R.P. Some efficient algorithms for solving systems of nonlinear equations. *Journal of Nucleic Acids* Vol. 10, No.2, 1973, pp. 327–344.

[5] Traub J.F. *Iterative Methods for the Solution of Equations*. Prentice-Hall, Englewood Cliffs 1964.

[6] Kchouk B, Dussault J. The Chebyshev–Shamanskii Method for Solving Systems of Nonlinear Equations. *J Optim Theory Appl* Vol. 157, 2013, pp. 148–167.

[7] Shamanskii V.E."A modification of Newton's method" *Ukrain.Mat.Zh*. Vol 19, 1967. pp.133-138.

[8] Ortega J.M, Rheinboldt, W.C. *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[9] Dennis Jr J.E, Schnabel R.B. *Numerical method for unconstrained optimization and nonlinear equations*. SIAM. 1996.

[10] Mathews J. H, Fink K. D. *Numerical method using MATLAB*. Prentice Hull, Upper saddle river, NJ 07458. 1999.

[11] Waziri M.Y, Leong W.J, Mamat M, Moyi, A.U. Two-Step Derivative-Free Diagonally Newton's Method for Large-Scale Nonlinear Equations. *World Applied Sciences Journal* Vol 21, 2013, pp. 86-94.

[12] Sulaiman I.M, Mamat M, Nurnadiah Z, Puspa, L.G. Solving Dual Fuzzy Nonlinear Equations via Shamanskii method. *International Journal of Engineering Technology*. Vol.7, No. 3.28, 2018, pp. 89-91.

[13] Sulaiman I.M, Mamat M, Afendee M.M, Waziri M.Y. Diagonal Updating Shamanskii-like method for solving singular fuzzy nonlinear equation. Far East Journal of Mathematical Sciences. Vol. 103, No. 10, 2017, pp. 1619-1629.

[14] Shin B,-C., Darvishi M., and Kim C.-H. A comparison of the newton-krylov method with high order newton-like methods to solve nonlinear systems, *Appld maths. and comptn*, Vol. 217, No. 7, 2010, pp. 3190-3198.

[15] More J. J., Grabow, B. S., and Hillstrom K. E. Testing unconstrained optimization software. *ACM trans. Math. Software*, Vol7, 1981, pp.17-41.