STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **6**, March 2018, pp 130–138.
Published online in International Academic Press (www.IAPress.org)

Quadrature based Broyden-like method for systems of nonlinear equations

Idowu Ademola Osinuga[1,*], Saidat Olaide Yusuff[1]

[1] *Department of Mathematics, Federal University of Agriculture, P.M.B. 2240, Abeokuta, Ogun State, Nigeria.*

Abstract A new iterative method based on the quasi-Newton approach for solving systems of nonlinear equations, especially large scale is proposed. We used the weighted combination of the Trapezoidal and Simpson quadrature rules. Our goal is to enhance the efficiency of the well known Broyden method by reducing the number of iterations it takes to reach a solution. Local convergence analysis and computational results are given.

Keywords Broyden method, quadrature rules, predictor-corrector, nonlinear systems, convergence, numerical examples

AMS 2010 subject classifications 65K05, 65D32, 90C53

DOI: 10.19139/soic.v6i1.471

1. Introduction

In vector notation form, consider the system of nonlinear equations

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

where the function $F : R^n \to R^n$, is continuously differentiable in an open neighbourhood $\Omega \subset R^n$. One of the prominent methods used to solve (1) is Newton's method as a result of its convergence property. However, the method has some challenges such as computation and storage of Jacobian matrix at every iteration as well as inefficiency in handling large scale systems. In an attempt to overcome these challenges, quasi-Newton methods have been introduced. Among the successful variants of quasi-Newton method is the Broyden's method that generate a reasonable approximation of the Jacobian matrix with no additional evaluation of the function, (see [15] and references therein). Broyden's method is given by

$$x_{p+1} = x_p - B_p^{-1} F(x_p), \tag{2}$$

where matrix B_p is the approximation of $F(x_p)$, such that the quasi-Newton equation

$$B_{p+1}(x_{p+1} - x_p) = F(x_{p+1}) - F(x_p) \tag{3}$$

is satisfied for each p. On the other hand, Broyden method needs n^2 (n is the length of vector x) storage locations, therefore, for large scale systems, this might lead to severe memory constraints [14].

Over the years, development of algorithms cheaper than Newton's method and some of its variants has been an area of intense research. Among others, several modifications of Newton's method and its variants were proposed to reduce computational cost, accelerate the convergence and reduce evaluations of functions in each step of the

*Correspondence to: Idowu Ademola Osinuga (Email: osinuga08@gmail.com). Department of Mathematics, Federal University of Agriculture, P.M.B. 2240, Abeokuta, Ogun State, Nigeria.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

iterations (see [3–5,7,17,22] and references therein). Just like quadrature rules have been used to propose Newton-like methods [17, 19–22], some efforts at using quadrature formulas to propose Broyden-like methods includes those of [13,14], all of which are directed towards the development of cost effective approaches. Based on this fact and frequent occurrence of systems of nonlinear equations in science and engineering, it is interesting to present an approach which will improve further the Jacobian approximation using weighted combination of Trapezoidal and Simpson (TS) quadrature formulas.

The paper is organized as follows: Section 2 is for the development of the proposed method; in Section 3, the convergence result is discussed; the numerical results are shown in Section 4; and Section 5 is for the conclusion.

2. Development of the Method

Let x^* be a root of the nonlinear equation $F(x) = 0$, where F is sufficiently differentiable. Newton's method originates from the Taylor's series expansion of the function (of a single variable) $f(x)$ about the point x_1:

$$f(x) = f(x_1) + (x - x_1)f^{'}(x_1) + \frac{1}{2!}(x + x_1)^2 f^{''}(x_1) + \cdots,$$

where f and its first and second derivatives, $f^{'}$ and $f^{''}$, are calculated at x_1. For a multiple variable function F, from the above equation, it is obvious that

$$F(x) = F(x_p) + \int_{x_p}^{x} F'(t)dt. \tag{4}$$

The matrix of partial derivatives appearing in (4) is the Jacobian J, where $\int_{x_p}^{x} F'(t)dt$ is multiple integrals as follows:

$$\int_{x_p}^{x} F'(t)dt = \int_{x_{p,1}}^{1}\int_{x_{p,2}}^{2}\cdots\int_{x_{p,n}}^{n} F'(x_1, x_2, \cdots, x_n)dx_n dx_{n-1}\cdots dx_1.$$

The most obvious approach is to treat the multiple integral as a nested sequence of one-dimensional integrals, and to use one-dimensional quadrature with respect to each argument in turn [11]. So we can approximate $\int_{x_p}^{x} F'(t)dt$ with the weight combination of the Trapezoidal and Simpson quadrature rules. The authors ([9], [10], [14], etc.) and references therein have proposed various methods by the approximation of the indefinite integral in (4) using Newton Cotes formulae of order zero and one. Approximating the integral in (4) by the average of Trapezoidal and Simpson (TS) quadrature rules yields:

$$x_{p+1} = x_p - 3\left[B(x_p) + B(z_p) + B(m_p)\right]^{-1} F(x_p),$$

where $z_p = \left(\frac{x_p + m_p}{2}\right)$ and $m_p = x_p - B_p^{-1}F(x_p)$, $p = 0, 1, \cdots$ Suppose we set $B_p = [B(x_p) + B(z_p) + B(m_p)]$. Then we have

$$x_{p+1} = x_p - 3B_p^{-1}F(x_p).$$

With the above formulation, we have the following two-step iterative schemes for solving the nonlinear system (1).

TS METHOD

For a given x_0 using initial matrix $B_0 = I$, compute the approximate solution x_{p+1} by the iterative schemes

$$m_p = x_p - B_p^{-1}F(x_p),$$

$$x_{p+1} = x_p - 3[B(x_p) + B(z_p) + B(m_p)]^{-1}F(x_p)$$

for $z_p = \left(\frac{x_p + m_p}{2}\right)$, $p = 0, 1, \cdots$

$$x_{p+1} = x_p - 3B_p^{-1}F(x_p). \tag{5}$$

In the following, we recall the most common notions and results about the convergence of an iterative method needed for subsequent development.

Definition 2.1 (q-superlinear convergence [18])
Let $x_p \subset R^n$ and $x^* \in R^n$. Then $x_p \to x^*$ is q-superlinearly convergent if

$$lim_{p\to\infty} \frac{\|x_{p+1} - x^*\|}{\|x_p - x^*\|} = 0.$$

Lemma 2.1 [12]
If $F'(x)$ is Lipschitz continuous with Lipschitz constant λ, then for any $u, v \in D$

$$\|F(v) - F(u) - F'(x)(v-u)\| \leq \lambda \max\{\|v-x\|, \|u-x\|\}\|v-u\|.$$

Moreover, if $F'(x)$ is invertible, then there exists ϵ and $\rho > 0$ such that

$$\frac{1}{\rho}\|v-u\| \leq \|F(v) - F(u)\| \leq \rho\|v-u\|$$

for all $u, v \in \Omega$ for which $\max\{\|u-x\|, \|v-x\|\} \leq \epsilon$.

Lemma 2.2 [18]
Let $x_p \in R^n$, $p \geq 0$. If x_p converges q-superlinearly to $x^* \in R^n$, then

$$\lim_{p\to\infty} \frac{\|x_{p+1} - x_p\|}{\|x_p - x^*\|} = 1.$$

3. Local Convergence Result

Our aim is to prove that the proposed TS method converges superlinearly. To achieve this, we consider the following standard assumptions on the nonlinear function F:
(i) F is differentiable in an open convex set $\Omega \in R^n$;
(ii) There exists $x^* \in \Omega$ such that $F(x^*) = 0$ and $F(x^*)$ is nonsingular and continuous for every x;
(iii) $F^{'}(x)$ is Lipschitz continuous and hence satisfies the Lipschitz condition of order 1 such that there exists a positive constant λ such that

$$\|F(x) - F(w)\| \leq \lambda\|x - w\|, \forall x, w \in R^n.$$

The following is the main result, which is a modified result of [14].

Theorem 1
Let $F : R^n \to R^n$ satisfy the hypothesis of *Lemma 2.1* on the set Ω. Let B_k be a sequence of nonsingular matrices in $L(R^n)$, the space of real matrices of order n. Suppose for some x_0 the sequence x_k generated by (5) remains in Ω and $\lim_{p\to\infty} x_p = x^*$ where for each p, $x_p \neq x^*$. Then $\{x_p\}$ converges q-superlinearly to x^* and $F(x^*) = 0$ if and only if

$$\lim_{p\to\infty} \frac{\|(\frac{1}{3}B_p - F'(x^*))s_p\|}{\|s_p\|} = 0, \tag{6}$$

where $s_p = x_{p+1} - x_p$ and $B_p = B(x_p) + B(z_p) + B(m_p)$.

Proof
Suppose (6) holds. Then (5) becomes

$$0 = \frac{1}{3}B_p s_p + F(x_p),$$

$$0 = \frac{1}{3}B_p s_p + F(x_p) - F'(x^*)s_p + F'(x^*)s_p,$$

$$0 = \frac{1}{3}B_p s_p - F'(x^*)s_p + F(x_p) + F'(x^*)s_p,$$

$$-F(x_{p+1}) + F(x_{p+1}) = (\frac{1}{3}B_p - F'(x^*))s_p + F(x_p) + F(x^*)s_p,$$

$$-F(x_{k+1}) = (\frac{1}{3}B_p - F'(x^*))s_p + (-F(x_{p+1})) + F(x_p) + F(x^*)s_p. \tag{7}$$

Take the norm of both sides to have:

$$\|-F(x_{k+1})\| = \|(\frac{1}{3}B_p - F'(x^*))s_p + (-F(x_{p+1})) + F(x_p) + F(x^*)s_p\|.$$

Using vector norm properties, we have

$$\|-F(x_{k+1})\| \leq \|(\frac{1}{3}B_3 - F'(x^*))s_p\| + \|(-F(x_{p+1})) + F(x_p) + F(x^*)s_p\|.$$

Divide through by $\|s_p\|$ to have

$$\frac{\|-F(x_{p+1})\|}{\|s_p\|} \leq \frac{\|(\frac{1}{3}B_p - F'(x^*))s_p\|}{\|s_p\|} + \frac{\|(-F(x_{p+1})) + F(x_p) + F(x^*)s_p\|}{\|s_p\|}.$$

Using *Lemma 2.1*,

$$\|-F(x_{p+1})\| \leq \frac{\|(\frac{1}{3}B_k - F'(x^*))s_p\|}{\|s_p\|} + \lambda \max\{\|x_{p+1} - x^*\|, \|x_p - x^*\|\}.$$

Since $x_p \to x^* \; \forall \; p$, then, from (6), we have

$$\lim_{p\to\infty} \frac{\|(F(x_{p+1}))\|}{\|s_p\|} \leq \lim_{p\to\infty} \frac{\|(\frac{1}{3}B_p - F'(x^*))s_p\|}{\|s_p\|} + \lambda \lim_{p\to\infty} \max\{\|x_{p+1} - x^*\|, \|x_p - x^*\|\} = 0. \tag{8}$$

Therefore,

$$F(x^*) = F(\lim_{p\to\infty} x_p) = \lim_{p\to\infty} F(x_p) = 0.$$

But $F'(x^*)$ is nonsingular. Thus, by *Lemma 2.1*, $\exists\ \rho > 0, p_0 \geq 0$ such that we have

$$\|F(x_{p+1})\| = \|F(x_{p+1}) - F(x^*)\| \geq \frac{1}{\rho}\|x_{p+1} - x^*\|. \tag{9}$$

For all $p \geq p_0$, (8) and Equation (9) gives

$$\begin{aligned} 0 &= \lim_{p\to\infty} \frac{\|F(x_{p+1})\|}{\|s_p\|} \\ &\geq \lim_{p\to\infty} \frac{1}{\rho}\frac{\|x_{p+1} - x^*\|}{\|s_p\|} \\ &\geq \lim_{p\to\infty} \frac{1}{\rho}\frac{\|x_{p+1} - x^*\|}{\|x_p - x^*\| + \|x_p - x^*\|} \\ &= \lim_{p\to\infty} \frac{\frac{1}{\rho}t_p}{1 + t_p} \end{aligned}$$

and $t_p = \frac{\|x_{p+1}-x^*\|}{\|x_p-x^*\|}$. This implies that

$$\lim_{p\to\infty} t_p = 0.$$

Therefore, x_p converges q-superlinearly to x^*. Conversely, suppose that x_p converges q-superlinearly to x^* and $F(x^*) = 0$. Then, by *Lemma 2.1*, there exists $\rho > 0$ such that we have

$$\|F(x_{p+1})\| \leq \rho\|x_{p+1} - x^*\|.$$

Then,

$$\begin{aligned} 0 &= \lim_{p\to\infty} \frac{\|x_{p+1} - x^*\|}{\|x_p - x^*\|} \\ &\geq \lim_{p\to\infty} \frac{\|F(x_{p+1})\|}{\rho\|x_p - x^*\|} \\ &= \lim_{p\to\infty} \frac{\|F(p_{p+1})\|}{\rho\|s_p\|} . \frac{\|s_p\|}{\|x_p - x^*\|}. \end{aligned}$$

Using *Lemma 2.2*, we have

$$\lim_{p\to\infty} \frac{\|F(x_{p+1})\|}{\|s_p\|} = 0.$$

From (7), we have

$$\begin{aligned} \frac{\|(\frac{1}{3}B_p - F'(x^*))s_p\|}{\|s_p\|} &\leq \lim_{p\to\infty} \frac{\|F(x_{p+1})\|}{\|s_p\|} + \lim_{p\to\infty} \frac{\|(-F(x_{p+1})) + F(x_p) + F(x^*)s_p\|}{\|s_p\|} \\ &\leq 0 + \lambda \lim_{p\to\infty} \max\{\|x_p - x^*\|, \|x_{p+1} - x^*\|\}. \end{aligned}$$

Since x_p converges to x^*, then

$$\lim_{p\to\infty} \|x_p - x^*\| = 0,$$

which proves that

$$\frac{\|(\frac{1}{3}B_p - F'(x^*))s_p\|}{\|s_p\|} = 0. \tag{10}$$

The proof is complete. □

4. Numerical Results

We used six test functions with eight instances of dimension $n = 5$ to $n = 1065$, which makes a total of 48 problems, in order to check the effectiveness of the proposed methods. The x_0 stands for the initial approximation to the solution in all the tested problems.

Stopping criterion
We have to stop the program if any of the following conditions are met.
(1) no x_p satisfies $\|F(x_p)\| \leq 10^{-12}$.
(2) the number of iterations reaches 500. (We do not want the program to run indefinitely)
A comparison of the numerical test results of our two-step methods is made with these well-known methods: Classical Broyden Method [2], Trapezoidal Broyden Method [13] and Midpoint-Simpson Broyden Method [14].

Table 1 shows the results on the number of iterations (NI) and the time (CPU) needed to converge to a solution. A failure is reported (denoted by '-') in the tabulated results. All the methods were implemented using MATLAB R2012b. All computations were carried out on a PC with Intel(R) Pentium(R) processor with 4GB of RAM

and CPU 2.20 GHz. The comparison of the tested methods is based on the performance profile of [8] and the comparison indices of [1].

Set of Test Problems

Problem 1 [5]
$F_i(x) = x_i x_{i+1} - 1,$
$F_n(x) = x_n x_1 - 1.$
$i = 1, 2, \cdots, n-1$ and $x_0 = (0.8, 0.8, \cdots, 0.8)^T.$

Problem 2 [16]
$F_i(x) = x_i x_{i+1} - 1,$
$F_n(x) = x_n x_1 - 1.$
$i = 1, 2, \cdots, n-1$ and $x_0 = (0.7, 0.7, \cdots, 0.7)^T.$

Problem 3 [16]
$F_i(x) = x_i x_{i+1} - 1,$
$F_n(x) = x_n x_1 - 1.$
$i = 1, 2, \cdots, n-1$ and $x_0 = (2, 2, \cdots, 2)^T.$

Problem 4 [14]
$F_i(x) = (\cos(x_i) - 1)^2 - 1,$
$i = 1, 2, \cdots, n$ and $x_0 = (1, 1, \cdots, 1)^T.$

Problem 5 [6]
$F_i(x) = x_i^2 - \cos(x_i - 1),$
$i = 1, 2, \cdots, n$ and $x_0 = (1.5, 1.5, \cdots, 1.5)^T.$

Problem 6 [6]
$F_i(x) = \exp(x_i^2 - 1) - \cos(1 - x_i^2),$
$i = 1, 2, \cdots, n$ and $x_0 = (0.5, 0.5, \cdots, 0.5)^T.$

The *Robustness* index measures the performance of the algorithm on a wide variety of problems in their class for all reasonable values of the starting point. The *Efficiency* index records the use of low amount of overall resources involved in the programming, while the *combined Robustness and Efficiency* index measures both robustness and efficiency performances. The Robustness (R), Efficiency (E) and Combined Robustness and Efficiency (C_{ER}) performance of each solver are shown as proposed by [1].

Discussion

The Tables show the results obtained for the benchmark problems. The Figures show the performance profile of each of the methods for small value of τ, the best possible ratio. We assume that we have n_s solvers and n_p problems, the performance profile $P : R \to [0; 1]$ is defined as follows: let P and S be the set of problems and set of solvers respectively. For each problem $p \in P$ and for each solver $s \in S$, we define $t_{p;s} :=$ (number of iterations required to solve problem p by solver s) and $t_{p;s} :=$ (computing time required to solve problem p by solver s) for Figures 1 and 2 respectively. The performance ratio is given by $r_{p;s} = \frac{t_{p;s}}{\min t_{p;s}, s \in S}$. Then the performance profile is defined by $P(\tau) := \frac{1}{n_p} sizep \in P : \log_2(r_{p,s}) \leq \tau, \forall \tau \in R$, where $P(\tau)$ is the probability for solver $s \in S$ that a performance ratio $r_{p,s}$ is within a factor $\tau \in R$ of the best possible ratio.

In both Figures, TS climb off the graph and this indicates that it solves all of the test problems successfully. MSB has the highest probability of 0.83 and 0.57 in Figures 1 and 2 respectively followed by the proposed method with 0.27 and 0.22, TB with 0.17 and 0.18 and CB with 0.02 and 0.04. In Figure 1, TS competes with MSB for $\tau \geq 0.4$. For factor $\tau \geq 0.3$, MSB and TS have the best probabilities in Figure 2. On the set of benchmark problems, TS is

Table 1. Comparison of Numerical Results

Prob	n	1		2		3		4	
		CB		TB		MSB		TS	
		NI	CPU	NI	CPU	NI	CPU	NI	CPU
1	5	6	0.051	4	0.048	3	0.039	4	0.051
	15	6	0.065	4	0.084	3	0.062	4	0.099
	35	5	0.155	5	0.164	3	0.103	4	0.164
	65	5	0.215	5	0.389	3	0.247	4	0.222
	165	5	0.515	5	0.574	4	0.437	4	0.484
	365	6	1.521	5	1.424	4	1.648	4	1.136
	665	6	2.794	5	2.469	4	2.351	4	2.067
	1065	6	5.131	5	4.463	4	4.225	4	3.666
2	5	6	0.038	5	0.049	4	0.032	4	0.037
	15	6	0.083	5	0.278	4	0.096	4	0.089
	35	6	0.152	5	0.165	4	0.114	4	0.149
	65	6	0.245	5	0.249	4	0.189	4	0.254
	165	6	0.700	5	0.784	4	0.449	4	0.525
	365	6	1.791	5	1.236	4	1.752	4	1.206
	665	6	3.097	5	2.422	4	2.571	4	2.252
	1065	6	5.253	5	4.369	4	4.017	4	3.711
3	5	-	-	-	-	5	0.058	6	0.046
	15	-	-	-	-	5	0.063	6	0.094
	35	-	-	-	-	5	0.131	6	0.191
	65	-	-	-	-	5	0.241	6	0.439
	165	-	-	-	-	5	0.510	6	0.642
	365	-	-	-	-	5	2.895	6	1.547
	665	-	-	-	-	5	2.663	6	2.939
	1065	-	-	-	-	5	5.648	6	5.174
4	5	6	0.045	5	0.086	4	0.034	4	0.051
	15	6	0.081	5	0.118	4	0.069	5	0.082
	35	6	0.157	5	0.183	4	0.137	5	0.204
	65	6	0.294	5	0.301	4	0.218	5	0.499
	165	6	0.795	5	0.668	4	0.525	5	1.018
	365	6	2.117	5	1.527	4	1.819	5	1.475
	665	6	3.359	5	2.781	4	2.557	5	2.812
	1065	6	5.571	5	4.841	4	4.195	5	5.169
5	5	11	0.068	6	0.062	4	0.035	5	0.059
	15	11	0.131	6	0.123	4	0.066	5	0.122
	35	11	0.290	6	0.279	4	0.136	5	0.201
	65	11	0.495	6	0.520	4	0.258	5	0.317
	165	11	1.334	6	0.982	4	0.817	5	0.836
	365	13	3.807	6	1.788	4	1.418	5	1.569
	665	13	6.799	6	3.754	4	2.656	5	5.203
	1065	13	11.412	6	8.567	4	4.639	5	6.756
6	5	-	-	5	0.075	-	-	7	0.078
	15	-	-	5	0.089	-	-	7	0.194
	35	-	-	5	0.280	-	-	7	0.335
	65	-	-	5	0.453	-	-	7	0.656
	165	-	-	5	0.982	-	-	7	1.365
	365	-	-	5	3.596	-	-	7	5.606
	665	-	-	5	6.012	-	-	7	7.632
	1065	-	-	5	8.088	-	-	7	9.729

Table 2. Summary of Robustness, Efficiency and Combined Robustness and Efficiency Measures

	CB	TB	MSB	TS
Successes	32	40	40	48
R	67%	83%	83%	100%
E	50%	80%	100%	84%
C_{ER}	38%	67%	83%	84%

superior in terms of robustness over other methods. The number of iterations obtained is promising and competitive, the CPU time is also encouraging. The results of the proposed method are better than one of CB and TB. However, MSB is more efficient than the proposed method. In conclusion, the numerical tests have shown that our new method is comparable with the well known existing Broyden-like methods.

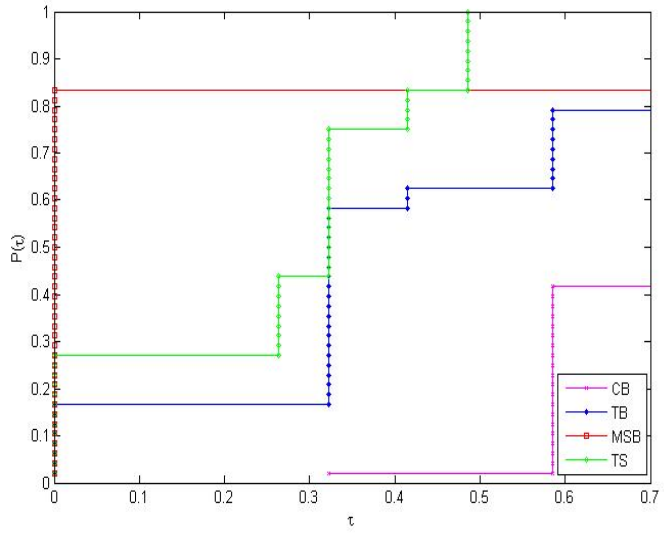

Figure 1. Performance Profile by number of iterations

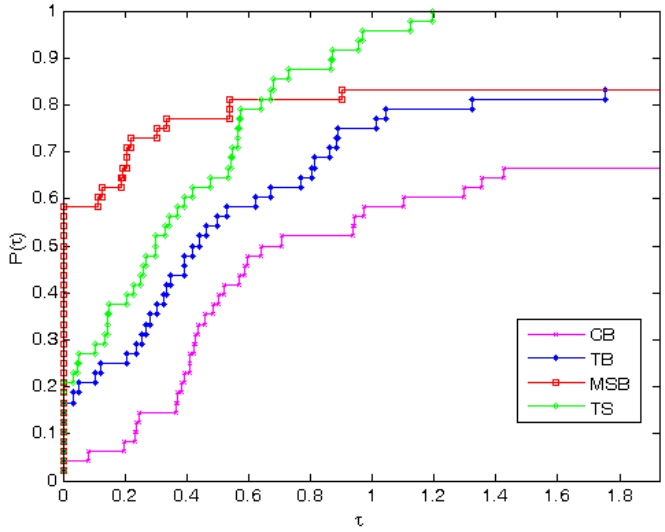

Figure 2. Performance Profile by CPU

5. Conclusion

The Broyden method was introduced to overcome the shortcomings of the Newton's method but still requires a higher number of iterations than the Newton's method. The Broyden-like method proposed is a promising and competitive alternative to quasi-Newton and Broyden method, especially for large scale systems.

REFERENCES

1. I. Bogle, and J. Perkins, *A new sparsity preserving quasi-Newton update for solving nonlinear equations*, SIAM J Sci Stat Comput 11(4), 621-630, 1990.
2. C. G. Broyden, *A class of methods for solving nonlinear simultaneous equations*, Math Comput 19, 577-593, 1965.
3. A. Cordero and J. R. Torregrosa, *Variants of Newton's method for functions of several variables*, Appl Math Comput 183, 199-208, 2006.
4. A. Cordero and J. R. Torregrosa, *Variants of Newton's method using fifth-order quadrature formulas*, Appl Math Comput 190, 686-698, 2007.
5. A. Cordero, J. R. Torregrosa, and M. P. Vassileva, *Pseudocomposition: a technique to design predictor-corrector methods for systems of nonlinear equations*, Appl Math Comput 218, 11496-11504, 2012.
6. M. T. Darvishi and B. Shin, *High-Order Newton-Krylov Methods to Solve Systems of Nonlinear Equations*, J. KSIAM, 15, 19-30, 2011.
7. A. Dhamacharoen, *An efficient hybrid method for solving systems of nonlinear equations*, J Comput Appl Math 263, 59-68, 2014.
8. E. D. Dolan and J. J. More, *Benchmarking optimization software with performance profiles*, Math Program Ser. A 91, 201-213, 2002.
9. M. Frontini and E. Sormani, *Some variants of Newton's method with third order convergence and multiple roots*, J Comput Appl Math 156, 345-354, 2003.
10. M. Frontini and E. Sormani, *Third order methods for quadrature formulae for solving system of nonlinear equations*, Appl Math Comput 149, 771-782, 2004.
11. M. A. Hafiz and S. M. Bahgat, *An efficient two-step iterative method for solving system of nonlinear equations*, J Math Res 4, 28-34, 2012.
12. C. T. Kelly, *Iterative Methods for Linear and Nonlinear Equations*, Philadelphia, PA, USA: SIAM, 1995.
13. M. Mamat, K. Muhammad and M.Y. Waziri, *Trapezoidal Broyden Method for Solving Systems of Nonlinear Equations*, Appl Math Sc 8, 251-260, 2014.
14. H. Mohammad and M. Y. Waziri, *On Broyden-like update via some quadratures for solving nonlinear systems of equations*, Turk J Math 39, 335-345, 2015.
15. K. Muhammad, M. Mamat and M. Y. Waziri, *A Broyden's-like Method for Solving Systems of Nonlinear Equations*, World Appl Sc J 21, 168-173, 2013.
16. B. C. Shin, M. T. Darvishi and C. H. Kim, *A comparison of the Newton-Krylov method with high order Newton-like methods to solve nonlinear systems*, Appl Math Comput 217(7), 3190-3198, 2010.
17. A. R. Soheili, S. A. Ahmadian and J. Naghipoor, *A Family of Predictor-Corrector Methods Based on Weight Combination of Quadratures for Solving Nonlinear Equations*, Int J Nonlinear Sc 6, 29-33, 2008.
18. B. Van De Rotten and S. V. Lunel, *A Limited Memory Broyden Method to Solve High-Dimensional Systems of Nonlinear Equations*, Technical Report MI 2003-2006. Leiden, the Netherlands: University of Leiden, 2003.
19. M. Y. Waziri, H. A. Aisha and M. Mamat, *A Structural Broyden's Like Method for Solving Nonlinear Equations*. Appl Math Sc 8(141), 7039-7046, 2014.
20. M. Y. Waziri, H. A. Aisha and M. Mamat, *A Newton's Like Method with Extra Updating Strategy for Singular Fuzzy Nonlinear Equations*. Appl Math Sc 8(142), 7047-7057, 2014.
21. M.Y. Waziri, W.J. Leong and M. Mamat, *A Two-Step Matrix-Free Secant Method for Solving Large-Scale Systems of Nonlinear Equations*, J Appl Math, Art ID 348654, 2012.
22. S. Weerakoon and T. G. I. Fernando, *A variant of Newton's method with accelerated third order convergence*, Appl Math Lett 13, 87-93, 2000.