Tanzania Journal of Science 49(3): 597-604, 2023
ISSN 0856-1761, e-ISSN 2507-7961
© College of Natural and Applied Sciences, University of Dar es Salaam, 2023

An Efficient Derivative Free Iterative Method and Condition Number of Nonlinear Equations Arising from Real World Phenomena

Alatuhigha Nguni[*], **Chacha S. Chacha and Adeline P. Mtunya**
Department of Mathematics, Physics and Informatics, Mkwawa University College of Education, University of Dar es Salaam, P. O. Box 2513, Iringa, Tanzania.
E-mails: alatunguni6@gmail.com; chacha.chacha@muce.ac.tz; adeline.mtunya@muce.ac.tz
[*]*Corresponding author*

Received 14 Oct 2022, Revised 30 Jul 2023, Accepted 16 Aug 2023 Published Sep 2023
DOI: https://dx.doi.org/10.4314/tjs.v49i3.4

Abstract

Recent researches have shown a great interest in developing novel numerical methods for solving nonlinear equations of the form $f(y) = 0$ arising from real world phenomena. However, very little attention has been given to the study of condition numbers which is an important aspect in measuring the sensitivity of the problem in response to slight perturbations in the input data. In this article, we present an efficient free derivative iterative scheme constructed by refining Newton-Raphson method standard form in which the derivative term is approximated by using finite difference scheme; hence making it derivative free. We also conducted an in-depth analysis of the condition numbers to explore sensitivity and efficiency comparisons between the proposed algorithm and existing methods for the given problems. Our investigation focused on iteration numbers, residuals, and convergence under mild error tolerances. Based on five numerical case studies, results revealed that the proposed Algorithm 2 outperforms the existing Algorithm 1 in terms of accuracy of the approximate solution. The results for the condition numbers indicated that all problems considered were ill-conditioned, highlighting the significance of studying condition numbers in the context of solving nonlinear equations.

Keywords: Derivative free, Iterative method, Condition number, Nonlinear equations.

Introduction

An iterative method (ITM) refers to a mathematical technique that starts with an initial value and generates a series of improved approximations for certain problems, each of which is derived from the one before. Even with the most powerful processing power available, iterative methods are usually the only options in dealing with nonlinear equations (NLE) because direct methods would be too expensive or even impossible to apply (Amritkar et al. 2015). Condition numbers (CN) of NLE measure the sensitivity of the solution to slight perturbation (σ) in the input data. A problem is said to be well-structured or well-conditioned when the CN is small and ill-structured when the CN is large (Rice 1966). The expression's input should be continuous in order to prevent differing results from minor changes in the input and increase of confidence in the solution. The problem is well-posed when the CN is less than 1, and it is ill-posed if the CN is arbitrarily large, which implies that it is challenging to discover the right answer. The solution to NLE is one of the most significant and difficult problems in real world phenomena. Newton's method (NM) is one of the most predominant methods in numerical analysis (Abbasbandy 2003) for obtaining α of the function $f(\alpha) = 0$. NM for finding α is given by

$y_{m+1} = y_m - \frac{f(y_m)}{f'(y_m)}$, where y_{m+1} converges to α as m tends to infinity and $f'(y_m) \neq 0$. NM is the most prevalent and elegant algorithm, which uses the derivative of the function. On the other hand, Steffensen's method (Jain 2007, Jain and Chand 2020)

$$y_{m+1} = y_m - \left(f(y_m)\right)^2/(f\left(y_m + f(y_m)\right) - f\left(y_m + f(x_m)\right), m = 0, 1, 2, 3, \cdots,$$

is the NM variant which does not employ derivative of the function. Steffensen's strategy has the same order of convergence as NM based on the approximation of the first derivative. Solaiman and Hashim (2019) constructed some novel optimal ITM with higher convergence and validated the applicability of the proposed ITM by finding the root of problems from engineering perspective. Recently, Chu et al. (2020) suggested a new family of ITM and explored the dynamics of the presented methods. Nevertheless, most of the works in the literature such as Chen (1990), Amat et al. (2003) and Azam and Rhoades (2012), among others have concentrated much in finding a better technique to solve NLE. However, to the best of our knowledge, none of the works highlighted above have studied the stability of the problems arising from real world phenomena.

This research work, therefore, aimed to make the following contributions to the field. Firstly, it proposes Algorithm 2, which is a non-derivative ITM, as one of the most efficient method for solving NLE when compared to its variant method presented by Bahgat (2021). Secondly, the notion of CN of NLE arising from real world phenomena is introduced. Finally, dynamical comparison of the Algorithm 1 and Algorithm 2 and computation of CN are presented.

Methods and Materials

In this section, two algorithms are presented. Algorithm 1 is briefly presented and Algorithm 2 is presented in details. Both algorithms are constructed by refining Newton-Raphson method standard form in which the derivative term is approximated by using finite difference schemes; hence making those derivative free.

Algorithm 1: Derivative Free Iterative Method

This is a derivative free iterative scheme for obtaining the solution of the NLE by using the approximation version of the first derivative of $f'(y_i)$ given by:

$$f'(y_i) \approx (f\left(y_i + \beta f(y_i)\right) - f\left(y_i - \beta f(y_i)\right))/2\beta f(y_i), \qquad (1)$$

where $\beta \epsilon \mathbb{R}$ and not equal to zero. Consider the following method:

$$\begin{aligned} h_i &= y_i - \frac{f(y_i)}{f'(y_i)}, \\ g_i &= y_i - \left(1 + \frac{f(h_i)}{f(y_i) - 2f(h_i)}\right)\frac{f(y_i)}{f'(y_i)}, \\ y_{i+1} &= g_i - \left(1 + \frac{2f(h_i)}{f(y_i) - 2f(h_i)}\right)\frac{f(g_i)}{f'(y_i)}. \end{aligned} \qquad (2)$$

Substituting the approximation of the derivative $f'(y_i)$ in equation (2) by equation (1) and obtain Algorithm 1, non-derivative iterative method work as follows:

$$\begin{aligned} h_i &= y_i - \frac{2\beta f^2(y_i)}{f\left(y_i + \beta f(y_i)\right) - f\left(y_i - \beta f(y_i)\right)}, \\ g_i &= y_i - \left(\frac{f^2(y_i) - f(y_i)f(h_i) + f^2(h_i)}{f^2(y_i) - 2f(y_i)f(h_i) + f^2(h_i)}\right)\frac{2\beta f^2(y_i)}{f\left(y_i + \beta f(y_i)\right) - f\left(y_i - \beta f(y_i)\right)}, \\ y_{i+1} &= g_i - \frac{2\beta f^2(g_i)}{f\left(g_i + \beta f(y_i)\right) - f\left(g_i - \beta f(g_i)\right)}. \end{aligned} \qquad (3)$$

The convergence analysis of Algorithm 1 closely aligns with the findings of Bahgat (2021). Numerical experiments conducted by Bahgat (2021) demonstrate that Algorithm 1 yields commendable performance, exhibiting minimal error when $\beta = 1$. This level of accuracy is well within our prescribed tolerance and comparable to the results obtained with $\beta = -1$ or $\beta = -0.5$. As a result, we focus exclusively on Algorithm 1 with $\beta = 1$ in this article, streamlining our investigation.

Description of Algorithm 2: Derivative free iterative method

Let us consider the nonlinear problem $f(y) = 0$. (4)

Assume that α is a zero of equation (4) with initial value y_0 close to the actual zero α, then Taylor's series in the neighbourhood of y_0 for the equation (4) gives;

$$f(y_0) - \frac{(y-y_0)f'(y_0)}{1!} + \frac{(y-y_0)^2 f''(y_0)}{2!} = 0. \quad (5)$$

When $f'(y_0) \neq 0$, then (5) implies

$$y_{i+1} = y_i - \frac{f(y_i)}{f'(y_i)} \quad (6)$$

This refers to Newton's algorithm for obtaining the root for scalar nonlinear functions (Kincaid and Cheney 1990, Chun 2006). Two iterative steps are given by:

$$h_i = y_i - \frac{f(y_i)}{f'(y_i)}, \text{ and}$$

$$y_{i+1} = h_i - \frac{f(h_i)f(y_i)}{f'(y_i)[f(y_i) - 2f(h_i)]}. \quad (7)$$

Incorporating Newton's algorithm, the two iterative steps method is transformed into three stages as follows:

$$h_i = y_i - \frac{f(y_i)}{f'(y_i)},$$

$$g_i = h_i - \frac{f(h_i)}{f'(h_i)},$$

$$y_{i+1} = g_i - \frac{f(g_i)f(h_i)}{f'(h_i)[f(h_i) - 2f(g_i)]} \quad (8)$$

To reduce the computational cost and create it to be more effective, the first derivative is estimated and made it derivative free with the intention of simple application to solve NLE arising from real world problems whose first derivative come to be unbounded or does not exist. Then the forward difference approximation

$$f'(y_i) = \frac{f(y_i + f(y_i))}{f(y_i)} = m(y_i) \quad (9)$$

is employed to approximate $f'(h)$ and exploit the finite difference scheme as

$$f'(h_i) = \frac{f(h_i) - f(y_i)}{h_i - y_i} = v(y_i, h_i) \quad (10)$$

When given the initial guess y_0, estimated solution y_{i+1} is found as follows:

$$h_i = y_i - \frac{f(y_i)}{m(y_i)}, \quad i = 0, 1, 2,$$

$$g_i = h_i - \frac{f(h_i)}{v(y_i, h_i)},$$

$$y_{i+1} = g_i - \frac{f(g_i)f(h_i)}{v(y_i, h_i)[f(h_i) - 2f(g_i)]}.$$

The suggested algorithm's calculating cost is low resulting to a higher efficacy.

To find α of NLE of the form $f(y) = 0$, $f: \mathbb{R} \to \mathbb{R}$, the following steps are followed:

1: State the function $f(y)$

2: Suggest an initial guess y_0

3: Compute the next estimate of the root y_{i+1}, $(i = 0, 1, 2 \ldots)$

4: While $|y_{i+1} - \alpha| < \varepsilon$, where ε denotes the tolerance to the error; repeat step 3 until the intended approximate solution is found.

Convergence of Algorithm 2

Lemma 1: Suppose α is a zero of $f(y) = 0$. When $f(y)$ is adequately smooth in the vicinity of α, then, the convergence order for Algorithm 2 is at least four.

Proof: To investigate the convergence, let α be a zero of $f(y) = 0$ and e_i denote the error at i^{th} iterative step; then $e_i = y_i - \alpha$. Employing Taylor's series expansion, it follows that,

$$f(y_i) = f'(\alpha)e_i + \frac{f''(\alpha){e_i}^2}{2!} + \frac{f'''(\alpha){e_i}^3}{3!} + \frac{f^{(iv)}(\alpha){e_i}^4}{4!} + O({e_i}^5),$$
$$= f'(\alpha)[e_i + c_2{e_i}^2 + c_3{e_i}^3 + c_4{e_i}^4 + O({e_i}^5)], \quad (11)$$

where $c_i = \frac{1}{i!}\frac{f^{(i)}(\alpha)}{f'(\alpha)}$ and

$$m(y_i) = f'(\alpha)\left[1 + 3c_2e_i + [7c_3 + c_2^2]e_i^2 + (6c_2c_3 + 15c_2^4)e_i^3 + (18c_2c_4 + 31c_5 + c_3c_2^2 + 5c_i^4)e_i^4 + O\left(e_i^5\right)\right]. \quad (12)$$

By using equations (11) and (12) you can obtain;

$$h_i = \alpha + 2c_2{e_i}^2 + (6c_3 - 5c_2^2){e_i}^3 + (14c_4 - 26c_3c_2 + 13c_2^2)e_i^4 + O\left(e_i^5\right) \quad (13)$$

$$f(h_i) = f'(\alpha)\left[2c_2{e_i}^2 + (6c_3 - 5c_2^2){e_i}^3 + (14c_4 - 26c_3c_2 + 13c_2^2)e_i^4 + O\left(e_i^5\right)\right] \quad (14)$$

$$v(y_i, h_i) = f'(\alpha)\left[1 + c_2 e_i + (c_3 + c_2^2)e_i^2 + (8c_2c_3 - 5c_2^3 + c_4)e_i^3 + (13c_2^4 - 27c_3c_2^2 + 16c_4c_2 + c_5 + 6c_3^2)e_i^4 + O(e_i^5)\right], \quad (15)$$

$$g_i = \alpha + 2c_2e_i^3 + (8c_2c_3 - 17c_2^3)e_i^4 + O(e_i^5), \quad (16)$$

$$f(g_i) = f'(\alpha)\left[2c_2e_i^3 + (8c_2c_3 - 7c_2^3)e_i^4 + O(e_i^5)\right]. \quad (17)$$

Exploiting equations (11)−(17) in **Algorithm 2** yields equality:

$$y_{i+1} = \alpha - 2c_2^3e_i^4 + O(e^5). \quad (18)$$

This implies that

$$e_{i+1} = -2c_2^3e_i^4 + O(e^5). \quad (19)$$

Equation (19) illustrates that **Algorithm 2** has convergence of fourth-order.

Condition Number of Nonlinear Equations

Condition number of a function measures how much the output value of the function can change for a small change in the input argument. Each problem we attempt to solve is dependent on an expression in some way. To be confident in our solution, we must first be aware that the expression's inputs are continuous which prevent us from getting drastically different results from minute changes in the input, although it falls short. Furthermore, we require the knowledge of the expression's sound conditioning. If the expression is well-conditioned, slight changes in the input will result into small changes in the outcomes. We refer to a problem as being ill-conditioned if modest changes in the input cause significant changes in the result.

To derive the formula for the condition number, we consider a function $f(y)$ evaluated at a point $y = y_0$. When the input is perturbed to $y = y_0 + \sigma$, the output becomes $f(y_0 + \sigma)$. By using the Mean Value Theorem, then it follows that:

$$f(y_0+\sigma)\Big/f(y_0) = \sigma f'(\delta)\Big/f(y_0) \approx \left[y_0 f'(y_0)\Big/f(y_0)\right](\sigma/y_0), \text{ where } \delta \in (y_0,\ y_0+\sigma).$$

So, the condition number that indicates magnification of changes of f at y_0 is roughly given by $C_f(y_0) = \left[y_0 f'(y_0)\Big/f(y_0)\right]$. If $C_f(y_0) < 1$ then the equation is well conditioned.

If $C_f(y_0)$ is arbitrarily large, then the equation is ill-conditioned. More details on condition numbers can be found in the articles by Chacha and Naqvi (2018) and Chacha (2022).

Results and Discussion

This part consists of two sections. The first section shows numerical experiment that was carried out to illustrate the efficiency of **Algorithm 2** in comparison with **Algorithm 1**. The second section shows the computation of condition number. Seven real world problems in the form of nonlinear equations were tested via MATLAB R2018a. The loops exited when $|f(y_{i+1})| < 10^{-5}$.

Numerical experiment

Equation 1: (The depth of embedment y in a sheet-pile wall (Hafiz and Al-Goria 2012) $f(y) = \frac{1}{4.62}(y^3 + 2.87y^2 - 10.28) - y$, initial point $y_0 = 2.5$.

Table 1: Results summary for Equation 1

Method	Iteration	y_{i+1}	$\lvert f(y_{i+1})\rvert$	$\varepsilon = \lvert y_{i+1} - y_i\rvert$
Algorithm 1 (β = 1)	3	2.002118	$8.829506e^{-16}$	$2.731050e^{-09}$
Algorithm 2	3	2.002118	$4.440892e^{-16}$	$2.629598e^{-09}$

Based on Table 1, both Algorithm 1 and Algorithm 2 converge at the 3rd iteration. The results demonstrate that Algorithm 2 consistently produces improved results (y_{i+1}) at each iteration stage with a minimum error ($\varepsilon = |y_{i+1} - y_i|$) compared to Algorithm 1. Furthermore, Algorithm 2 exhibits smaller residuals $|f(y_{i+1})|$ than Algorithm 1, indicating its faster approach towards zero.

Equation 2: Consider the beam scheming problem (Shams et al. 2020) concerning the embedment y of a sheep pile wall is governed by the equation:

$$f(y) = \frac{1}{4.62}(y^3 + 2.87y^2 - 4.62y - 10.28)$$

with an initial point $y_0 = 3.0$. We use Algorithms 1 and 2 to obtain the approximate solution and the results summary is recorded in Table 2.

Based on Table 2, both Algorithm 1 and Algorithm 2 converge at the 3rd iteration. However, Algorithm 2 consistently produces superior results (y_{i+1}) at each iteration with a minimum error ($\varepsilon = |y_{i+1} - y_i|$) compared to Algorithm 1. Additionally, Algorithm 2 exhibits smaller residuals $|f(y_{i+1})|$ than Algorithm 1, indicating its faster approach towards zero.

Table 2: Results summary for Equation 2

Method	Iteration	y_{i+1}	$\|f(y_{i+1})\|$	$\varepsilon = \|y_{i+1} - y_i\|$
Algorithm 1 (β = 1)	3	2.002118	8.829506e^{-16}	3.815893e^{-09}
Algorithm 2	3	2.002118	3.844928e^{-16}	1.073091e^{-09}

Equation 3: The Planck's radiance law problem appearing in Bradie (2006) and Jain (2013) is given by:

$$\phi(\mu) = \frac{8\pi hc u^{-5}}{e^{hc/\mu TK}-1}.$$

This equation computes the density of energy in an isothermal blackbody. The equation is re-written as $(y) = 1 - 0.2y - e^{-y}$. We apply **Algorithm 1** and **Algorithm 2** with an initial point $y_0 = 4.1$ for solving the approximate solution y and a summary of results is recorded in Table 3.

According to Table 3, both Algorithm 1 and Algorithm 2 converge at the 2nd iteration. However, it is noteworthy that Algorithm 2 consistently produces improved results (y_{i+1}) at each iteration stage with a minimum error ($\varepsilon = |y_{i+1} - y_i|$) when compared to Algorithm 1. Additionally, Algorithm 2 exhibits smaller residuals $|f(y_{i+1})|$ than Algorithm 1, signifying its faster approach towards zero.

Table 3: Results summary for Equation 3

Method	Iteration	y_{i+1}	$\|f(y_{i+1})\|$	$\varepsilon = \|y_{i+1} - y_i\|$
Algorithm 1 (β = 1)	2	4.965114	5.464379e^{-17}	3.525123e^{-04}
Algorithm 2	2	4.965114	3.382711e^{-17}	1.1524331e^{-04}

Equation 4: Blood rheology model (Fournier 2017). The caisson fluid model demonstrates that the flow of basic fluids in a tube is such that the centre core of the fluids will move as a plug with slight deformation and pace gradient will take place close to the wall. The blood is a non-Newtonian fluid and is assumed as caisson fluid. For example, consider the following function as a NLE to explore the plug flow of caisson fluids: $\mathrm{H} = 1 - \frac{16}{7}\sqrt{y} + \frac{4}{3}y - \frac{1}{21}y^4$, where flow pace reduction is calculated by H. Setting $\mathrm{H} = 0.40$ which yields:

$$f(y) = \frac{1}{441}y^8 - \frac{8}{63}y^5 - 0.05714285714y^4 + \frac{16}{9}y^2 - 3.624489796y + 0.3.$$

Algorithm 1 and **Algorithm 2** are employed with an initial point $y_0 = 0.9$ to solve the approximate solution y and a summary of results is recorded in Table 4.

Referring to the data in Table 4, it is evident that Algorithm 2 achieves convergence at the 3rd iteration, while Algorithm 1 diverges. This clearly demonstrates the faster convergence of Algorithm 2 compared to Algorithm 1. The results vividly illustrate the considerable improvement at each iteration step in Algorithm 2 when compared to Algorithm 1.

Table 4: Results summary for Equation 4

Method	Iteration	y_{i+1}	$\|f(y_{i+1})\|$	$\varepsilon = \|y_{i+1} - y_i\|$
Algorithm 1 (β = 1)	-	Diverged	-	-
Algorithm 2	3	8.643355e^{-02}	5.551115e^{-17}	3.210544e^{-06}

Equation 5: Hydraulic permeability, which is a problem with fluid permeability, is essentially a measurement of flow resistance. It can be expressed as;
$k = \frac{r_e y^3}{20(1-y)^2}$, $r_e y^3 - 20(1-y)^2 = 0$, where k stands for specific hydraulic permeability r_e denoting the radius of the tube, and $0 \leq y \leq 1$ is the porosity. For $k = 0.4655$ and $r_e = 100$, we get the NLE: $f(y) = 100y^3 - 9.31(1-y)^2$. **Algorithm 1** and **Algorithm 2** are employed with an initial point $y_0 = 2.0$ to solve the approximate solution y and a summary of results is recorded in Table 5.

Table 5: Results summary for Equation 5

Method	Iteration	y_{i+1}	$\|f(y_{i+1})\|$	$\varepsilon = \|y_{i+1} - y_i\|$
Algorithm 1 (β = 1)	-	Diverged	-	-
Algorithm 2	4	3.426482e^{-01}	0	1.131404e^{-06}

Based on the data presented in Table 5, it is evident that Algorithm 2 achieves convergence at the 4th iteration, whereas Algorithm 1 diverges. This highlights the faster convergence of Algorithm 2 in comparison to Algorithm 1. These results clearly demonstrate the significant improvement at each iteration step of Algorithm 2 when compared to Algorithm 1.

Numerical results for the considered five real world problems demonstrate the efficiency of Algorithm 2. This is to say that Algorithm 2 is also appropriate in solving NLE arising from real world phenomena when compared with Algorithm 1. Algorithm 1 becomes unstable at some points for certain problems as it can be seen in Table 1 to Table 5.

Computation of condition number $C_f(y_i)$

Equation 1: Consider $f(y) = \frac{1}{4.62}(y^3 + 2.87y^2 - 10.28) - y$ with an input data $y = 2.002118778953827$. The summary of results is presented in Table 6.

From Table 6, condition number of Equation 1 is arbitrarily large ($C_f(y_i) > 1$). This result shows that the problem is ill-conditioned which means is very sensitive to any small perturbation in the input data.

Table 6: Condition number for Equation 1

$y_i = y + \sigma$	$C_f(y_i)$	$f(y_i)$
2.002118778953827+10^{-7}	2.002118979674711e+07	4.090389920108351e-07
2.002118778953827+10^{-9}	2.002119185724736e+09	4.090388916466736e-09
2.002118778953827+10^{-11}	2.002148962727241e+11	4.090328076244987e-11
2.002118778953827+10^{-16}	1.844099323583018e+16	4.440892098500626e-16

Equation 2: Consider $f(y) = \frac{1}{4.62}(y^3 + 2.87y^2 - 4.62y - 10.28)$ with an input data $y = 2.002118778953827$. The summary of results is presented in Table 7.

From Table 7, condition number of Equation 2 is arbitrarily large ($C_f(y_i) > 1$). This result shows that the problem is ill-conditioned which means is very sensitive to any small perturbation in the input data.

Table 7: Condition number for Equation 2

$y_i = y + \sigma$	$C_f(y_i)$	$f(y_i)$
$2.002118778953827+10^{-7}$	2.002118980672156e+07	4.090389918070539e-07
$2.002118778953827+10^{-9}$	2.002119229951120e+09	4.090388826110923e-09
$2.002118778953827+10^{-11}$	2.002157055487802e+11	4.090311543053624e-11
$2.002118778953827+10^{-16}$	1.064967359369193e+16	7.689856447620132e-16

Equation 3: Consider $f(y) = 1 - 0.2y - e^{-y}$ with an input data $y = 4.965114231744276$. The summary of results is presented in Table 8. From Table 8 above, condition number of Equation 3 is arbitrarily large ($C_f(y_i) > 1$). This result shows that the problem is ill-conditioned which means is very sensitive to any small perturbation in the input data.

Table 8: Condition number for Equation 3

$y_i = y + \sigma$	$C_f(y_i)$	$f(y_i)$
$4.965114231744276 + 10^{-7}$	4.965114335323633e+07	1.930228469074202e-08
$4.965114231744276 + 10^{-9}$	4.965112948608904e+09	1.930228962776504e-10
$4.965114231744276 + 10^{-11}$	4.965140791823373e+11	1.930218138102013e-12
$4.965114231744276 + 10^{-16}$	2.833172991458863e+16	3.382710778154774e-17

Equation 4: Consider $f(y) = \frac{1}{441}y^8 - \frac{8}{63}y^5 - 0.05714285714y^4 + \frac{16}{9}y^2 - 3.624489796y + 0.3$ with an input data $y = 8.643355805246679e^{-02}$. The summary of results is presented in Table 9. From Table 9, condition number of Equation 4 is arbitrarily large ($C_f(y_i) > 1$). This result shows that the problem is ill-conditioned which means is very sensitive to any small perturbation in the input data.

Table 9: Condition number for Equation 4

$y_i = y + \sigma$	$C_f(y_i)$	$f(y_i)$
$8.64335580524667\mathrm{e}^{-02} + 10^{-7}$	8.643365341367542e+05	3.317353331477868e-07
$8.64335580524667\mathrm{e}^{-02} + 10^{-9}$	8.643355909839228e+07	3.317353503007325e-09
$8.643355805246679\mathrm{e}^{-02} + 10^{-11}$	8.643345406346273e+09	3.317357499810214e-11
$8.643355805246679\mathrm{e}^{-02} + 10^{-16}$	8.608800834968345e+14	3.330669073875470e-16

Equation 5: Consider $f(y) = 100y^3 - 9.31(1-y)^2$ with an input data $= 3.426482058114499e^{-01}$. The summary of results is recorded in Table 10. From Table 10, condition number of Equation 5 is arbitrarily large ($C_f(y_i) > 1$). This result shows that the problem is ill-conditioned which means is very sensitive to any small perturbation in the input data. The computed condition numbers $C_f(y_i)$ for the nonlinear equations in Equations 1–5 indicate extreme ill-conditioning, with arbitrarily large values. Consequently, even a minor perturbation in the input data leads to significant changes in the solutions (see Table 6–Table 10).

Table 10: Condition number for Equation 5

$y_i = y + \sigma$	$C_f(y_i)$	$f(y_i)$
$3.426482058114499\mathrm{e}^{-01} + 10^{-7}$	3.426484×10^{6}	$4.746223762452928\mathrm{e}^{-06}$
$3.426482058114499\mathrm{e}^{-01} + 10^{-9}$	3.426482×10^{8}	$4.746222881379936\mathrm{e}^{-08}$
$3.426482058114499\mathrm{e}^{-01} + 10^{-11}$	3.426483×10^{10}	$4.746221193840938\mathrm{e}^{-10}$
$3.426482058114499\mathrm{e}^{-01} + 10^{-16}$	4.577585×10^{15}	$3.552713678800501\mathrm{e}^{-15}$

Conclusion

Analysis on the efficiency based on numerical computation revealed that the proposed Algorithm 2 was more effective when compared with Algorithm 1. On the other hand, condition numbers computed for each problem indicated that most of problems were ill-conditioned. This infers that nonlinear equations arising from real world phenomena are quite sensitive and may generate approximations with huge error. To enhance accuracy and applicability, it is crucial to prioritize meticulous data input and simulation in the model, thereby minimizing errors and ensuring the attainment of correct solutions. In our future research, we intend to study intensively the stability of these algorithms and their accuracy under a certain tolerance.

Acknowledgements: Authors thank all those who provided constructive contributions which led to the improvement of this paper.

References

Abbasbandy S 2003 Improving Newton–Raphson method for nonlinear equations by modified Adomian decomposition method. *Appl. Math. Comput.* 145(2-3): 887-893.

Amat S, Busquier S and Gutiérrez JM 2003 Geometric constructions of iterative functions to solve nonlinear equations. *J. Comput. Appl. Math.* 157(1): 197-205.

Amritkar A, Sturler ED, Świrydowicz K, Tafti D and Ahuja K 2015 Recycling Krylov subspaces for CFD applications and a new hybrid recycling solver. *J. Comput. Phys*. 303: 222-237.

Azam A and Rhoades BE 2012 Some fixed point theorems for a pair of self-maps of a cone metric space. *IAENG Int. J. Appl. Math*. 42(3): 193-197.

Bahgat MS 2021 Three-point iterative algorithm in the absence of the derivative for solving nonlinear equations and their basins of attraction. *J. Egypt. Math. Soc*. 29(1): 1-17.

Bradie B 2006 A friendly introduction to numerical analysis, 1st ed, Pearson Education Inc., London.

Chacha CS 2022 On iterative algorithm and perturbation analysis for the nonlinear matrix equation. *Commun. Appl. Math. Comput*. 4: 1158–1174.

Chacha CS and Naqvi SMRS 2018 Condition numbers of the nonlinear matrix equation. *J. Function Spaces* 2018: 1-8.

Chen D 1990 On the convergence of a class of generalized Steffensen's iterative procedures and error analysis. *Int. J. Comput. Math.* 31(3-4): 195-203.

Chu Y, Rafiq, N, Shams M, Akram S, Mir NA and Kalsoom H 2020 Computer methodologies for the comparison of some efficient derivative free simultaneous iterative methods for finding roots of non-linear equations. *Comput. Mater. Contin*. 66(1): 275-290.

Chun C 2006 Construction of Newton-like iteration methods for solving nonlinear equations. *Numer. Math.* 104: 297-315.

Fournier RL 2017 Basic Transport Phenomena in Biomedical Engineering, CRC Press.

Hafiz MA and Al-Goria SMH 2012 New ninth and seventh order methods for solving nonlinear equations. *Eur. Sci. J.* 8(27): 83-95.

Jain P 2007 Steffensen type methods for solving non-linear equations. *Appl. Math. Comput.* 194(2): 527-533.

Jain D 2013 Families of Newton-like methods with fourth-order convergence. *Int. J. Comput. Math*. 90(5): 1072-1082.

Jain P and Chand P 2020 Derivative free iterative methods with memory having higher R-order of convergence. *Int. J. Nonlin. Sci. Numer. Simul*. 21: 641-648.

Kincaid DE and Cheney EW 1990 Numerical Analysis, Brooks/Cole Publishing Company, Pacific Grove, CA, USA.

Rice JR 1966 A theory of condition. *SIAM J. Numer. Anal*. 3(2): 287-310.

Shams M, Mir NA, Rafiq N, Almatroud AO and Akram S 2020 On dynamics of iterative techniques for nonlinear equation with applications in engineering. *Math. Probl. Eng.* 2020: 1-17.

Solaiman OS and Hashim I 2019 Efficacy of optimal methods for nonlinear equations with chemical engineering applications. *Math. Probl. Eng*. 2019: 1-11.