Derivative Free Levenberg-Marquardt Method for Solving Fuzzy Nonlinear Equation

A U Omesa\textsuperscript{1}, I M Sulaiman\textsuperscript{2,*}, M Mamat\textsuperscript{2}, M Y Waziri\textsuperscript{3}, A Shadi\textsuperscript{2}, M A Zaini\textsuperscript{4}, Kalfin\textsuperscript{5}, and I Sumiati\textsuperscript{6}

\textsuperscript{1}Department of Mathematics, College of Agriculture Zuru, Kebbi, Nigeria.
\textsuperscript{2}Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Campus Kuala Nerus, 22200, Terengganu, Malaysia.
\textsuperscript{3}Department of Mathematics, Bayero University, Kano, Nigeria.
\textsuperscript{4}Institute of Engineering Mathematics, Universiti Malaysia Perlis, Kangar, Malaysia
\textsuperscript{5}Doctor Program of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran, Indonesia
\textsuperscript{6}Master Program in Mathematics, Faculty of Mathematics and Natural Science, Universitas Padjadjaran, Indonesia

Corresponding author: sulaimanib@unisza.edu.my

Abstract. In this paper, we present a derivative-free computational method for solving fuzzy nonlinear equation. Derivative-free technique avoids computing the derivative by generating an estimate to the derivative. This is made possible by inserting the estimate of $F'(x_k)$ in Levenberg-Marquardt’s method. Numerical experiments are carried out which shows that the method is efficient.

1. Introduction
In recent time, most real-life problems are modelled in form of nonlinear system equations as

$$F(x) = 0$$

$F: \mathbb{R}^n \to \mathbb{R}^n$ and that it is required to find $x^* \in \mathbb{R}^n$ such that $F(x^*) = 0$. When the coefficients of (1) are written in crisp number which can be conveniently represented by fuzzy numbers, Zadeh [13] introduced and investigated the idea of numbers having fuzzy variables with their arithmetic operations. Among the widely used application of arithmetic of number with fuzzy variables is systems of nonlinear equations with a parametric fuzzy number [4,5,8]. The numerical solution to fuzzy nonlinear equation with fuzzy coefficient involving fuzzy variable is on when the Jacobian is non-singular near exact root( $x^*$), in particular [1] solved the parameterized fuzzy equations via Newton’s method. [2] obtained the solution of nonlinear least squares approximation via a derivative free scheme of Levenberg-Marquardt and Gauss algorithms. [9] employ Levenberg-Marquardt method for the solution of parameterized fuzzy equations and [10] used a modification of Shamanskii’s steps to obtained the solutions of Dual fuzzy nonlinear equations. [11] employ a Quasi Newton algorithm to obtain the solution of parameterized fuzzy equation. [12] employ Chord approach for the same problems. Most of the above discussed methods are variants of Newton’s method that require computing for the Jacobian or approximate Jacobian at every iteration. To overcome these shortcomings, numerous studies suggested evaluating the Jacobian matrix once throughout the iterations or once after every few iterations [3].
In this paper, we consider forward difference and central difference approach applied to Levenberg-Marquardt that does not compute the Jacobian for solving systems of fuzzy nonlinear equations. The Levenberg-Marquardt method is an iterative method whose iterative function is generated either by computing the Jacobian or by a derivative estimation. We introduce the finite difference approach to the method that avoids computing the derivative of the function \( f \). This is made possible by inserting the estimate \( g(x) \) of \( f'(x) \) in Levenberg-Marquardt method. The expectation is to reduce the burden of computing for the Jacobian matrix at every iteration. Section 2 presents a brief overview and some fundamental of fuzzy nonlinear equations. The Levenberg-Marquardt method is presented in Section 3 followed by the proposed method in Section 4. And finally, we report our numerical results and conclusion in Sections 5 and 6 respectively.

2. Preliminaries

Some basic definition numbers with fuzzy variables are presented in this section.

**Definition 1.** A fuzzy number can be defined as a set like \( u: R \rightarrow I = [0,1] \) satisfying the criteria below [13,15],

1. \( u \) is upper semicontinuous,
2. outside some interval say \([c, d]\), then, \( u(x) = 0 \)
3. \( \exists a, b \) such that \( c \leq a \leq b \leq d \), where \( a, b, c, d \) are real numbers and
   (3.1) on the interval \([c, a]\), we say \( u(x) \) is monotonic increasing
   (3.2) on the interval \([b, d]\), we say \( u(x) \) is monotonic decreasing
   (3.3) for \( a \leq x \leq b \), \( u(x) = 1 \).

The set of all number with fuzzy variables are represented by \( E \) with its parameterized form defined in [8] as follows.

**Definition 2.** The parameterized fuzzy number \( u \) is a pair \((\underline{u}, \overline{u})\) of function \( \underline{u}(r), \overline{u}(r), 0 \leq r \leq 1 \), satisfying the criteria below [14,15]:

1. The function \( \underline{u}(r) \) is bounded and monotonic increasing left continuous,
2. The function \( \overline{u}(r) \) is bounded and monotonic decreasing left continuous,
3. For \( 0 \leq r \leq 1 \), \( \underline{u}(r) < \overline{u}(r) \).

The following \( u(r) = \overline{u}(r) = \alpha, 0 \leq r \leq 1 \) is used to represent the crisp number \( \alpha \). One of the well-known number with fuzzy variables is the trapezoidal number with fuzzy variable given as \( u = (x_0, y_0, \alpha, \beta) \) with \([x_0, y_0]\) as the interval defuzzifier with \( \alpha \) and \( \beta \) as left and right fuzziness respectively, where

\[
u(x) = \begin{cases} \frac{1}{\alpha} (x - x_0 + \alpha), & x_0 - \alpha \leq x \leq x_0, \\ 1, & x \in [x_0, y_0] \\ \frac{1}{\beta} (y_0 - x + \beta), & y_0 \leq x \leq y_0 + \beta, \\ 0, & \text{otherwise}. \end{cases}
\]

is the membership function whose parameterized form is

\[
\underline{u}(r) = x_0 - \alpha + \alpha r, \quad \overline{u}(r) = y_0 + \beta - \beta r.
\]
Let the set of all triangular numbers with fuzzy variables be denoted as \( TF(R) \). Then, arithmetic operations of addition and scalar multiplication of numbers with fuzzy variables are defined by the extension principle which are denoted as \([14]\).

For \( u = (u, \overline{u}) \), \( v = (v, \overline{v}) \) and \( k > 0 \) as arbitrary, then, the addition \( u + v \) and multiplication by real number \( k > 0 \) is given as

\[
(u + v)(r) = u(r) + v(r), \quad (u + v)(r) = \overline{u}(r) + \overline{v}(r),
\]

\[
(ku)(r) = ku(r), \quad (ku)(r) = k\overline{u}(r).
\]

For more reference of fuzzy nonlinear systems, researchers should refer to (see \([14,15,16,17]\)).

### 3. Levenberg-Marquardt method

The Levenberg-Marquardt method \([6,7]\) is an iterative scheme that depends on a damping parameter \( \mu_k \) and generates a sequence of approximation to the minimum

\[
x_{n+1} = x_n - (J_k^T J_k + \mu_k I)^{-1} J_k^T F(x)
\]

where \( \mu_k \) is a scalar that controls both the magnitude and direction \( d_k \). The direction which is given by

\[
d_k = -(J_k^T J_k + \mu_k I)^{-1} g_k
\]

with \( g_k = J_k^T F_k \). The Jacobian matrix \( J(x) \) will be approximated using \( J(x^k, Fx^k) \) and is given as

\[
d_k = -(D_k^T D_k + \mu_k I)^{-1} D_k^T F_k
\]

where the matrix \( D_k \) is component wise computed by two possible choices of forward difference or central difference and is defined as

\[
D_{ij}^k = \frac{f_i(x_k + h_i^k e_j) - f_i(x_k)}{h_i^k}
\]

and

\[
D_{ij}^k = \frac{f_i(x_k + h_i^k e_j) - f_i(x_k - h_i^k e_j)}{2h_i^k}
\]

with \( e_j \) the \( j - th \) unit column vector and \( \mu_k \) and \( h_i^k \) given by

\[
\mu_k = c \frac{\|J_k\|}{\|J_k\|^2}
\]

with

\[
c = \begin{cases} 
10, \quad \text{if } 10 \leq \frac{\|J_k\|}{\|J_k\|^2} \\
1, \quad \text{if } 1 < \frac{\|J_k\|}{\|J_k\|^2} < 10 \\
0.01, \quad \text{if } \frac{\|J_k\|}{\|J_k\|^2} \leq 1 
\end{cases}
\]

and

\[
h_i^k = \min \left\{ \frac{\|J_k\|}{\|J_k\|^2}, \delta_i^k \right\}
\]

Notice that \( h_k \) and \( \delta_k \) are \( n \) dimensional vectors, the index \( j \) denotes the \( j - th \) component of these vectors, and

\[
\delta_k = \begin{cases} 
10^{-9} |x_i^k| < 10^{-6} \\
0.001 |x_i^k|, \quad \text{otherwise}
\end{cases}
\]

In this paper we consider the Derivative free approach to solving Levenberg-Marquardt method which we present via the following Algorithm.
4. Iterative Approach for solving fuzzy nonlinear equations

This section presents the proposed scheme for solving nonlinear equation

\[ F(x) = 0 \]

whose parameterized form is:

\[ \bar{F}(\bar{x}, r) = 0 \quad \forall r \in [0,1]. \tag{8} \]

Assume \( \alpha = (\alpha, \bar{\alpha}) \) represents the solution to (5), that is

\[ \begin{align*}
F(\alpha, \bar{\alpha}; r) &= 0, \\
\bar{F}(\alpha, \bar{\alpha}; r) &= 0, \\
\forall r &\in [0,1]
\end{align*} \]

Now, suppose \( x_0 = (\bar{x}_0, \bar{x}_0) \) is the approximate solution of the above nonlinear system, at that point, \( \forall r \in [0,1] \), there are \( h(r), k(r) \) with

\[ \alpha(r) = x_0(r) + h(r), \]
\[ \bar{\alpha}(r) = \bar{x}_0(r) + k(r). \]

Applying the Taylor series of \( F, \bar{F} \) of \( (\bar{x}_0, \bar{x}_0) \), at that point \( \forall r \in [0,1], \)

\[ \begin{align*}
\frac{\partial}{\partial r} F\left(\alpha, \bar{\alpha}; r\right) &= F\left(\bar{x}_0, \bar{x}_0, r\right) + h F_x\left(\bar{x}_0, \bar{x}_0, r\right) + g F_x\left(\bar{x}_0, \bar{x}_0, r\right) + 0(h^2 + hk + h^2) = 0 \\
\frac{\partial}{\partial r} \bar{F}\left(\alpha, \bar{\alpha}; r\right) &= \bar{F}\left(\bar{x}_0, \bar{x}_0, r\right) + h \bar{F}_x\left(\bar{x}_0, \bar{x}_0, r\right) + g \bar{F}_x\left(\bar{x}_0, \bar{x}_0, r\right) + 0(h^2 + hk + h^2) = 0
\end{align*} \]

Also, suppose \( \bar{x}_0 \) and \( \bar{x}_0 \) are defined close to \( \alpha \) and \( \bar{\alpha} \), at that point, \( h(r) \) and \( k(r) \) are sufficiently small. Let suppose all partial derivatives that are needed exist and bounded. Then, for sufficiently small \( h(r) \) and \( k(r) \), \( \forall r \in [0,1] \), we get,

\[ \begin{align*}
F\left(\bar{x}_0, \bar{x}_0, r\right) + h F_x\left(\bar{x}_0, \bar{x}_0, r\right) + g F_x\left(\bar{x}_0, \bar{x}_0, r\right) = 0 \\
\bar{F}\left(\bar{x}_0, \bar{x}_0, r\right) + h \bar{F}_x\left(\bar{x}_0, \bar{x}_0, r\right) + g \bar{F}_x\left(\bar{x}_0, \bar{x}_0, r\right) = 0
\end{align*} \]

and hence the unknown quantities \( h(r) \) and \( k(r) \) are obtained using the equation that follows, \( \forall r \in [0,1], \)

\[ \begin{bmatrix} h(r) \\ g(r) \end{bmatrix} = \begin{bmatrix} F_x\left(\bar{x}_0, \bar{x}_0, r\right) \\ F_x\left(\bar{x}_0, \bar{x}_0, r\right) \end{bmatrix}^{-1} \begin{bmatrix} F\left(\bar{x}_0, \bar{x}_0, r\right) \\ \bar{F}\left(\bar{x}_0, \bar{x}_0, r\right) \end{bmatrix} \tag{9} \]

where

\[ J(\bar{x}_0, \bar{x}_0, r) = \begin{bmatrix} F_x\left(\bar{x}_0, \bar{x}_0, r\right) & F_x\left(\bar{x}_0, \bar{x}_0, r\right) \\ \bar{F}_x\left(\bar{x}_0, \bar{x}_0, r\right) & \bar{F}_x\left(\bar{x}_0, \bar{x}_0, r\right) \end{bmatrix} \]

is the Jacobian of \( F = (F, \bar{F}) \) computed at \( x_0 = (\bar{x}_0, \bar{x}_0) \). However, \( J(x_0, \bar{x}_0, r) \) in (6) is derived by a derivative estimation \( J(x_k, \bar{x}_k, F(x_k), r) \) for \( k = 0, 1, 2, \ldots \) and for all \( r \in [0,1] \)

Thus, the subsequent approximations for \( \bar{x}(r) \) and \( \bar{x}(r) \) are

\[ \bar{x}_i(r) = x_0(r) + h(r), \]
\[ \bar{x}_i(r) = \bar{x}_0(r) + k(r), \]

for all \( r \in [0,1] \).

Now, using the iterative scheme, the approximated solution is obtained as follows where \( r \in [0,1], \)

\[ \begin{align*}
\bar{x}_{n+1}(r) &= \bar{x}_n(r) + h_n(r), \\
\bar{x}_{n+1}(r) &= \bar{x}_n(r) + k_n(r), \tag{10}
\end{align*} \]
when \( n = 1,2, \ldots \) corresponding to (5)

\[
J(x_n, \bar{x}_n, r) \frac{h(r)}{g(r)} = \left( -F(x_n, \bar{x}_n, r) \right)
\]

Now, if \( J(x_n, \bar{x}_n, r) \) is non-singular, at this point, applying (6), we can obtain the iterative scheme of Newton’s method as,

\[
\begin{bmatrix}
x_{n+1}(r) \\
\bar{x}_{n+1}(r)
\end{bmatrix} = \begin{bmatrix} x_n(r) \\
\bar{x}_n(r)
\end{bmatrix} - J(x_n, \bar{x}_n, r)^{-1} \frac{F(x_n, \bar{x}_n, r)}{F(x_n, \bar{x}_n, r)}
\]

which is the proposed method whose algorithm is defined as follows

**Algorithm 2:** Derivative free method

*Step 1.* Initialization: Given a parameterized nonlinear equation with fuzzy variables

*Step 2.* Compute for \( x_0 \) by solving the parameterized equations for \( r = 0,1 \) and \( k = 0,1,2 \ldots 

*Step 3.* Evaluate \( F(x_k) \)

*Step 4.* Compute \( \left[ f(x_k, h_k)^T f(x_k, h_k) + \mu_k I \right]^{-1} \) via (4) or (5) above

*Step 5.* Compute \( x_{k+1} = x_k - \left[ f(x_k, h_k)^T f(x_k, h_k) + \mu_k I \right]^{-1} f(x_k, h_k)^T F(x_k) \)

*Step 6.* Continue the process with step 3 to 5 using the next \( k \) until \( \epsilon \leq 10^{-4} \) are satisfied.

5. **Numerical Results**

This section considered some nonlinear equations with fuzzy variables from [1] and presented the solutions to the parameterized equations illustrate the performances of forward difference and central difference methods applied to Levenberg-Marquardt iterative method. The experiments are performed on MATLAB (R2015a) programming software using double precision computer.

**Example 1 [1]:** Given a nonlinear equation with fuzzy variables as

\[
(3,3,4,5)x^2 + (1,2,3)x = (1,1,2,3)
\]

Let \( x \) be positive, then, without any loss of generality, we parameterized the equation as follows:

\[
(3 + r)x^2(r) + (1 + r)x(r) = (1 + r)
\]
\[
(5 - r)x^2(r) + (3 - r)x(r) = (3 - r)
\]

The initial guess is obtained by letting \( r = 0 \) and \( r = 1 \) in the parameterized system,

\[
r = 1
\]
\[
4x^2(1) + 2x(1) = 2
\]
\[
4x^2(1) + 2x(1) = 2
\]

\[
r = 0
\]
\[
3x^2(0) + x(0) = 1
\]
\[
5x^2(0) + 3x(0) = 3
\]

when \( r = 0 \), then, \( x(0) = 0.4343, \bar{x}(0) = 0.5307 \) and when \( r = 1 \), we have \( x(1) = \bar{x}(1) = 0.5000 \). We consider \( x_0 = (0.4,0.4,0.5,0.6) \), as our initial guess. Via Algorithm 2 with \( x_0 = (0.4,0.4,0.5,0.6) \) and approximate Jacobian \( J(x_k, F\bar{x}) \) the number of iteration and execution time for forward difference and central difference are 2(0.6443) and 6(0.6676) respectively with an error > 10^{-5}. Please, refer to figure 1 for details of the solution for problem 1 \( \forall r \in [0,1] \).
Example 2 [12]: Given a nonlinear equation with fuzzy variables as

\[(2,2,1,1)\, x^3 + (3,3,1,1)\, x^2 + (4,1,1) = (8,8,3,5)\]

Let \( x \) be positive, then, without any loss of generality, we parameterized the equation as follows:

\[(1 + r)x^3(r) + (2 + r)x^2(r) + (3 + r) = (5 + 3r)\]
\[(3 - r)x^3(r) + (4 - r)x^2(r) + (5 - r) = (13 - 5r)\]

or equality

\[(1 + r)x^3(r) + (2 + r)x^2(r) = (2 + 2r)\]
\[(3 - r)x^3(r) + (4 - r)x^2(r) = (8 - 4r)\]

let \( r = 0 \) and \( r = 1 \). Then, the initial guess is obtained as follows

\[x^3(0) + 2x^2(0) = 2\]
\[3x^3(0) + 4x^2(0) = 8\]

and

\[2x^3(1) + 3x(1) = 4\]
\[2x^3(1) + 3x(1) = 4\]

The initial guess \( x_0 = (0.8,0.8,0.9,0.9) \) is considered. Applying Algorithm 2, the number of iteration and execution time for forward difference and central difference are 2(0.0636) and 7(0.0604) respectively with an error \( > 10^{-5} \). Please, refer to figure 2 for details of the solution for problem 2 \( \forall r \in [0,1] \).
6. Conclusion

This paper studied the forward difference and central difference methods applied to the Levenberg-Marquardt iterative method for solution of nonlinear equation having fuzzy variables. We were mainly interested in reducing the experiment cost of the Jacobian by computing the approximation to the Jacobian matrix throughout the iteration process. This was achieved by parameterizing the nonlinear equation with fuzzy variables and then solved via Levenberg-Marquardt method. The numerical result presented in section five illustrates that forward difference approach is very promising in all the tested problems used.

References

[1] Abbasbandy S and Asady B 2004 Newton’s method for solving fuzzy nonlinear equations Applied Mathematics and Computation 159 (2) 349-356
[2] Brown K M and Dennis J E 1971 Derivative free analogues of the Levenberg-Marquardt and Gauss algorithms for nonlinear least squares approximation Numerische Mathematik 18 (4) 289-297
[3] Broyden C G 1965 A class of methods for solving nonlinear simultaneous equations Mathematics of computation 19 (92) 577-593
[4] Buckley J J and Qu 1990 Y Solving linear and quadratic fuzzy equations Fuzzy sets and systems 38 (1) 43-59
[5] Fang J X 2002 On nonlinear equations for fuzzy mappings in probabilistic normed spaces Fuzzy Sets and Systems 131 (3) 357-364
[6] Levenberg K A 1944 method for the solution of certain non-linear problems in least squares Quarterly of applied mathematics 2 (2) 164-168
[7] Marquardt D W 1963 An algorithm for least-squares estimation of nonlinear parameters Journal of the society for Industrial and Applied Mathematics 11 (2) 431-441
[8] Peeva K 1992 Fuzzy linear systems Fuzzy Sets and Systems 49 (3) 339-355
[9] Sulaiman I M, Mamat M, Waziri M Y, Mohamed M A and Mohamad F S 2018 Solving Fuzzy Nonlinear Equation via Levenberg-Marquardt Method Far East Journal of Math Sci 103 (10) 1547-1558
[10] Umar A O, Waziri M Y and Sulaiman I M 2018 Solving Dual Fuzzy Nonlinear Equations via a Modification of Shamanskii Steps Malaysian Journal of Computing and Applied Mathematics 1 (2) 1-9
[11] Omesa U A, Mamat M, Sulaiman I M and Sukono 2020 On Quasi Newton Method for Solving Fuzzy Nonlinear Equations International Journal of Quantitative Research and Modelling 1 (1) 1-10
[12] Waziri M Y and Moyi A U 2016 An alternative approach for solving dual fuzzy nonlinear equations International Journal of Fuzzy Systems 18 (1) 103-107
[13] Zadeh L A 1965 Fuzzy sets Information and control 8 (3) 338-353
[14] Mamat M, Sulaiman I M and Ghazali P L 2020 An accelerated scheme for solving parameterized fuzzy nonlinear equations International Journal of Advanced Science and Technology 29 (5) 248-255
[15] Sulaiman I M and Mamat M 2021 Shamanskii Method for Solving Parameterized Fuzzy Nonlinear Equations An International Journal of Optimization and Control: Theories & Applications. In press.
[16] Sulaiman I M, Mamat M, Zamri N and Ghazali P L 2018 Solving dual fuzzy nonlinear equations via Shamanskii method International Journal of Engineering and Technology 7 (28) 89-91
[17] Sulaiman I M, Mamat M, Waziri M Y, Fadhilah and Kamfa K 2016 Regula Falsi method for solving fuzzy nonlinear equation Far East Journal of Mathematical Sciences 100 (6) 873-88