Mathematical Statistician and Engineering Applications
ISSN: 2094-0343
2326-9865

Solution of Non Linear Equation Using Newton Raphson and Quasi Newton Method and Application in Engineering Field

Ankush Walia[1]

[1]Post-Graduate Student, UIS Chandigarh University, Gharuan, Mohali, Punjab 140413, India

[1]Waliaankush999@gmail.com

Article Info
Page Number: 2164-2181
Publication Issue:
Vol. 71 No. 4 (2022)

Article History
Article Received: 25 March 2022
Revised: 30 April 2022
Accepted: 15 June 2022
Publication: 19 August 2022

Abstract
In this review paper, a study is done on the linearisation of the existing algorithm of the Newton-Raphson method for solving systems of non-linear equations of higher dimensions and improving the converging behaviour of the Newton-Raphson method, which differs in the case of finite elemental systems. Newton's approach is the first method to be reviewed in this paper, elucidating non-linear systems of equations with each possibility of solution of equations. This paper introduces the notion of Jacobian, discussing the inverse function theorem and define locally unique solutions along with the order of convergence of Newton Raphson. Furthermore, this paper includes the two best criteria for knowing when to end iterations, with some numerical examples to explain the algorithm and utilitarian of this method for solving non-linear systems, mentioning the shortcoming of the method and introducing Broyden's approach (also known as Broyden's method for finding an iterative formula for Jacobian inverse), which is evolved from Newton's method, and using the fixed difference approach to find ways for Jacobian computing, and lastly, including the gravity of this method in the engineering field.

Keywords: Numerical Solutions, Newton - Raphson, Non - linear, Linearisation, Jacobian, Quasi-Newton, Approximations.

Introduction

Early school taught us how to solve problems using various algebraic approaches. Replacement and elimination techniques are two of these strategies. The quadratic formula and factorization are two more algebraic approaches that can be used. In linear algebra, row reduction may be used as a method to solve systems of linear equations. Non-linear systems with real-world significance, such as triangulation of GPS signals, conversion of different types of motion in daily life, and manufacturing-crafting infrastructures, which are modelled in the form of ODEs and PDEs, which are complex in nature lacks the analytic methods for solutions. Analytical methods also have limitations in the case of non-linear problems, In such cases, numerical methods works very well. In terms of non-linear equations, knowing how many possible solutions the system has is not practical, so to amplify the numerical errors and use a single value decomposition approach, iterative algorithms are formulated, in which initial guesses are refined for solutions until the value of function at those guesses converges to something. Also, numerical methods can only reliably find solutions which are locally unique (explained later). There are many iterative methods such as Bisection, Regula-falsi, but methods like these requires two initial guesses to find the approximations, so to avoid this problem we use the Newton's method, which is an open method (it only requires one initial guess), for solving system of equation of the form $f(x) = 0$[1]. Frequently, it is

known as the Newton Raphson method.The Newton-Raphson method is an efficient iterative algorithm as it uses the concept of successive approximations and linearisation. Furthermore, having a derivative in its former formula benefits the solution as it leads the solution to the accurate direction. The pre-existing method is only prolonged for finding the solution of non-linear equations in one dimension [2], but the major query that arises in solving is how to know the final iteration is the solution and when to end the algorithm which leads in the formulation of the two criterion to check whether the solution is accurate or not, which are the Function Norm and Step-Norm criteria, elaborated later in the paper. Newton-Raphson being an open method is beneficial with many uncertainties as convergence in open methods is not guaranteed, but if the method does converge, it does so much faster than the bracketing methods. [3], The existing formula for Newton Raphson is

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \text{ for } n = (0,1,2,3..)$$

The modified Newton-Raphson method's algorithm helps in solving non-linear systems with n-different variables and for locating roots in n variables, Broyden"s approach is a quasi-Newton method in numerical analysis which was first introduced by C. G. Broyden in 1965. The Jacobian matrix, J, is used at each iteration of Newton's technique. However, calculating this Jacobian is a time-consuming and costly process. The goal of Broyden's method is to compute the entire Jacobian just once, and to perform rank-one modifications at subsequent iterations. Numerical methods can solve real world problems which are relativistic in nature. However, analytical solutions solve ideal problems which in many cases do not exist in reality. That is why the engineering field is mostly influenced by Newton-Raphson, as in chemical engineering, non-linear systems always come in front, such as, Equations of State, Energy Balance, and Mass Balance with non-linear reaction. Moreover, the Van Der Waal's equation also requires linearisation and iterative method to solve.

PRELIMINAIRIES

NON-LINEAR SYSTEM IN N -VARAIBLE

DEFINITION: A function or mapping: $\mathbb{R}^n \to \mathbb{R}$ is said to be non linear if it does not satisfy the linearity property (i.e. Superposition Principle)

$$f(x_1 + x_2 + \cdots) \neq f(x_1) + f(x_2) + \cdots$$

Defining non-linear systems on behalf of the above definition, the system of equations which has more than one equation in multi-variable and at least one of these equations has a degree greater than one is described as a system of non-linear equations, also the extension to these type of system $f(x) = 0$ in multi-dimension forms the system of equation in n-variables:

$$f_1(x_1, x_2, x_3, \ldots x_n) = 0$$
$$f_n(x_1, x_2, x_3, \ldots x_n) = 0$$

A recognizable example of non-linear system:

$$3x_1 - \cos(x_2 x_3) - \frac{1}{2} = 0$$
$$x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 = 0$$
$$e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3} = 0$$

Moreover in terms of non-linear system finding the exact number of solutions become irrelevant as there can be three generalised possibilities:

1. There could be no Solution e the system is Singular, $J(x) = 0$.
2. There is locally unique solution in between $1 \leq n \leq \infty$.
3. There are infinitely many solution.

Locally Unique Solution: For a given system of functions having solution x^* if there exists a ball of finite radius such that the x^* is the only solution within that ball than the solution is locally unique.

JACOBIAN

The Jacobian matrix describes the differential of a function at every point where it is differentiable, also the Jacobian describes the rate of change of a vector function with respect to independent variables. The Jacobian may be interpreted as a kind of first-order derivative of a vector-valued function of n-variables and can also be understood as displaying the extent of stretching, rotating, or transforming, that the function imposes in the neighbourhood of points. Moreover, a function need not be differentiable for the validation of its Jacobian matrix, as only its first-order partial derivative is necessary to exist. The Jacobian matrix can be written in the form:

$$\mathbb{J} = \begin{bmatrix} \frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}(\mathbf{x})}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \nabla^T f_1(\mathbf{x}) \\ \vdots \\ \nabla^T f_m(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_1(\mathbf{x})}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial f_m(\mathbf{x})}{\partial x_n} \end{bmatrix}$$

Example : Taking the non-linear system from equation:

$$\mathbb{J} = \begin{bmatrix} 3 & x_3 \sin(x_2 x_3) & x_2 \sin(x_2 x_3) \\ 2x_1 & -162(x_2 + 0.1) & \cos(x_3) \\ -x_2 e^{-x_1 x_2} & -x_1 e^{-x_1 x_2} & 20 \end{bmatrix}$$

Hessisan Matrix: The Hessian matrix is similar to Jacobian matrix having partial derivative of second order $\mathbf{H} = \left[\frac{\partial^2 f}{\partial x_i \, \partial x_j}\right]_{ij}$ such that

$$H = \begin{bmatrix} \frac{\partial^2 f_1(\mathbf{x})}{\partial^2 x_1} & \cdots & \frac{\partial^2 f_1(\mathbf{x})}{\partial^2 x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f_m(\mathbf{x})}{\partial^2 x_1} & \cdots & \frac{\partial^2 f_m(\mathbf{x})}{\partial^2 x_n} \end{bmatrix}$$

Inverse- Function Theorem: In calculus, for a function $f(x)$, if it is continuous and differentiable at a point a than the function has an inverse in the neighbourhood of a. In terms of Jacobian, the inverse function theorem suggests, for a function at x if the Jacobian at x is not equal to zero i.e. $J(x) \neq 0$ than there exist a locally unique solution for that system. [4]

NEWTON-RAPHSON METHOD

Newton's method is one of the most iconic numerical method and is even referred to by Burden and Fairies. This method can be formulated from the Taylor series as the Taylor series commute with several calculus topics, its result being elegant and its accumulation to many other important results in calculus, including the derivation of many formulae. The derivation of Newton-Raphson from the Taylor series requires the conversion of the existing series into linear form. Considering the Taylor Series:

$$f(x_o + h) = f(x_o) + hf'(x_o) + \frac{h^2 f(x_o)}{2!} + \cdots$$

Taking $x_0 + h = x_i + h = x_{i+1}$ and terminating all the higher power terms in the expansion making it linear results in

$$f(x_{i+1}) = f(x_i) + (x_{i+1} - x_i) * f'(x_i)$$

In Linear Algebra, a system of equations can be expressed by using matrices and vectors. As a result, the nonlinear system may be expressed as a matrix with matching vectors. The linearisation of the Newton-Raphson method requires the replacement of the derivative of the function with the Jacobian inverse, as in higher dimensions the terms involving quadratic and higher powers are ignored. This leads to a good approximation of the non-linear function as linear. [5]

$$x_{n+1} = x_n - J^{-1}(x_n) * f(x_n)$$

Finding inverse of the Jacobian matrix being a complex task is avoided by pre multiplying Jacobian matrix both side of above equation.

$$J(x_n) * (x_{n+1} - x_n) = -J(x_n) * J^{-1}(x_n) * f(x_n)$$

The resulting and extended version of the Newton Raphson Method is

$$J(x_n) * \partial x = -f(x_n) \; where \; \partial x = (x_{n+1} - x_n)$$

ALGORITHM

The Ladder of the Newton Raphson method for finding the solution of non-linear equation are in order:

Step 1: Let $x^{(0)} = \left(x_1^{(0)}, x_2(0), \dots, x_n(0)\right)$ be a given initial vector.

Step 2: Calculate $J\left(x^{(0)}\right)$ and $f\left(x^{(0)}\right)$.

Step 3: calculate the vector $\partial x^{(0)}$, where

$$\partial x = \begin{bmatrix} \partial x_1 \\ \partial x_2 \\ \vdots \\ \partial x_n \end{bmatrix}$$

In order to find $\partial x^{(0)}$, the linear system $J\left(x^{(0)}\right)\partial x^{(0)} = -f\left(x^{(0)}\right)$, is solved using Gaussian Elimination, this can be simplified using the $J^{-1}(x)$.

Step 4: As $\partial x^{(0)}$, to finish the first iteration process and finding $x^{(1)}$. Thus using the output from Step 3,

$$x^{(1)} = \partial x^{(0)} + x^{(0)} = \begin{bmatrix} \partial x_1^{(0)} \\ \partial x_2^{(0)} \\ \vdots \\ \partial x_n^{(0)} \end{bmatrix} + \begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \\ \vdots \\ x_n^{(0)} \end{bmatrix}$$

Step 5:Calculating the absolute relative approximate error ϕa, Where $|\phi a| = \left|\left(x_{n+1} - \frac{x_n}{x_{n+1}}\right)\right| * 100$

CONVERGENCE RATE

The rate of the convergence of numerical methods are examined by limit of ratio of the difference between the successive approximations to the final iterations:

$$\lim_{x\to\infty} \frac{\|(x_{n+1} - x_0)\|^p}{\|(x_n - x_0)\|_q^p} = M$$

If the above limit exist and is not equal to zero than, for $q = 1$, the value of $M < 0$, implies the convergence is linear, $q > 1$, the convergence is super linear andf for $q = 2$, the convergence is said to be quadratic, i.e. The number of accurate digits doubles after each iteration. More generalised [6] determination of the convergence rate is done by:

$$|x - f(x)/f'(x)| < 1$$

$$|\, f(x) * f''(x)/[f(x)^2] < 1$$
$$|f(x) * f''(x)| < [f(x)^2]$$

For the existence of the root of the function, the above expression must be satisfied for any interval taken, which implies the standard convergence rate of the NR-method is quadratic. [3] Newton Raphson is only locally convergent [7] and not globally which means the initial approximation is already sufficiently close to the solution, if the method's subsequent approximations are assured to converge to a solution. This method converges quadratically only when the Jacobian is not singular. i.e. $detJ(x) \neq 0$, and when the initial guesses are good and the iterations are sufficiently closer to the root of the equation whereas bad initial guess leads to a chaotic analysis.

STOPPING CRITERION

Non-linear systems are solved iteratively, which implies making an algorithm map for solving, which takes a value x_i and generates x_{i+1} which is a better iteration than the previous one, and the algorithm stops when the map is sufficiently converged. Theoretically, to check if the iterations are reliable or accurate, stopping criterion are used. The two major stopping criterion are:

Function-norm criteria: Examining and solving the system of non-linear function to find the point of convergence and reach the best possible iteration, the size of the function at that iteration in the norm space is observed, if the size is smaller than some absolute tolerance ε i.e.

$$\|f(x_{i+1})\|_p \leq \varepsilon$$

Step-norm criteria: Observing two consecutive successive approximations and taking the difference between them, and checking the norm of the difference is smaller than some absolute tolerance εA or to product of some relative error and norm of the current solution, i.e. :

$$||x_{i+1} - x_i\|_p \leq \varepsilon_R ||x_{i+1}\|_p + \varepsilon_A$$

The expression suggests, two approximations can have large spacing between them but the relative spacing may be small, in that case relative error is necessary, implies if the norm is smaller than tolerance than the function sufficiently converge.

SHORTCOMING IN PERFORMANCE

The most major drawback of Newton's approach is that for each iteration, J(x) and its inversion must be calculated. Depending on the scale of considered system, calculating the Jacobian matrix and its inverse might take a long time. Another difficulty arises, is while using Newton's method is that it may fail to converge. An oscillation between points will occur if Newton's method fails to converge. Newton Raphson Method fails in certain cases like Maxima/Minima, or if iteration follows an asymptote they blow up.

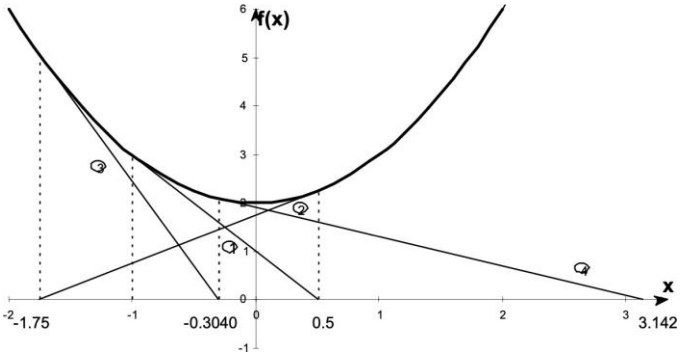

Figure 1: Oscillation around local minima around f(x) = $\boldsymbol{x^2 + 2}$

Newton Raphson not being globally convergent faces these problem, in case of asymptote if we take a linearisation the iteration along the way of asymptote, iterations blows up at a high rate.

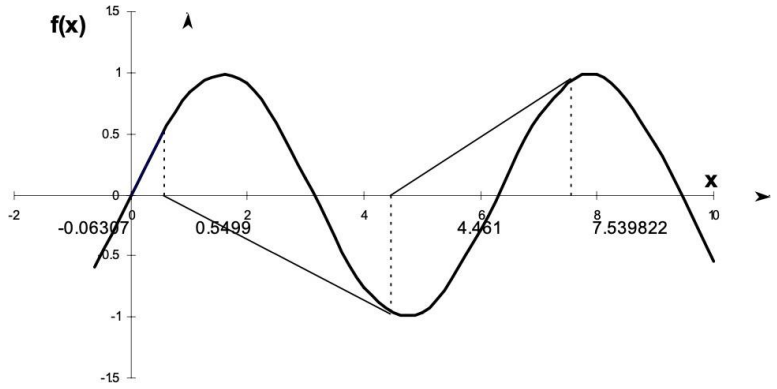

Figure 2: Root Jumping for f(x) = sinx = 0

One may choose an initial guess that is near to a root in some situations where the function f(x) oscillates and has a number of roots. The estimations could diverge at a different root though. For instance, if you select $x_0 = 2.4\pi = (7.539822)$ as an initial guess to solve the equation $f(x) = \sin x = 0$, it converges to the root of $x = 0$ as shown in fig 2 and table 3, whereas to converge to $x = 2\pi = 6.2831853$, one may have picked this starting guess.

TABLE I: Iteration Table For Overshoot Function

Iteration Number	x_i	$f(x_i)$	$\lvert\varepsilon_a\rvert\%$
0	7.539822	0.951	68.973
1	4.462	-0.969	711.44
2	0.5499	0.5226	971.91
3	-0.06307	-0.06303	7.54×10^4
4	8.376×10^4	8.375×10^5	4.28×10^{10}
5	-1.95861×10^{13}	-1.95861×10^{13}	

In other cases of powered functions having overshoot and leading to convergence/divergence :

$F(x) \Rightarrow |x|^S$

Given general function for, $0 < s < 1/2$, it diverge and for $\frac{1}{2} < s < 1$ it converge.Taking a linearisation in case of function having power loss scaling near the root, leads to non-existence of derivative of the function which occur in back and forth path of the iteration causing an overshoot and convergence doesn't happen.

QUASI NEWTON METHOD

There are many failures of the method, the shortcomings and the analytically calculation of the Jacobian and its inverse which becomes very difficult in case of N-dimensional system, requiring adjustment in some the existing algorithm. Newton raphson method is based upon the linear approximation of the function near the root, altering the linearisation costs in the reduction of the quadratic convergence of the method to a super- linear factor convergence. The conversion of the Newton Raphson into more efficient ways are known as Quasi - Newton methods. The concept of Finite Difference (FD) approximations of Jacobian, Broyden's method for approximating inverse Jacobian comprises Quasi- Newton.

Broyden's method: In place of the Jacobian matrix, this approach uses an approximation matrix that is updated at each iteration. This means that Broyden's method's iterative technique is nearly identical to Newton's method's iterative procedure. The main difference is that instead of J(x), an approximation matrix *Ji* is used. The formulation for this is deduced as:

$$x^{(i+1)} = x^{(i)} - J_i^{-1} * f\left(x^{(i)}\right)$$

The secant method is a special case of the Newton Raphson method that uses a coarse approximation of the derivative in one dimension. 8

$$x_{i+1} = x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})}$$

Now this method is used to form an iterative formula for calculating the Jacobian inverse, but first this can be extended to N-dimensional by using undetermined Secant approximation of Jacobian, which is also the

$$J(x_i)(x_i - x_{i-1}) = f(x_i) - f(x_{i-1})$$

Newton's method for : x_i

$$J(x_{i-1})(x_i - x_{i-1}) = -f(x_{i-1})$$

Taking the difference of secant approximation and newton's approach:

$$\big(J(x_i) - J(x_{i-1})\big)(x_i - x_{i-1}) = f(x_i)$$

The iterative form for approximation of Jacobian, which is one of the possible solution for Jacobian: known as the (Rank - I Update approximation)

$$J(x_i) = J(x_{i-1}) + \frac{f(x_i)(x_i - x_{i-1})^T}{\|x_i - x_{i-1}\|_2^2}$$

Using the Sherman- Morrison Formula which makes it possible to write inverse in terms of $J(x_{i-1})$ i.e.

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^TA^{-1}}{1 + v^TA^{-1}u}$$

Applying above formula to rank - I update :

$$J(x_i)^{-1} = J(x_{i-1})^{-1} - \frac{J(x_{i-1})^{-1}f(x_i)(x_i - x_{i-1})^TJ(x_{i-1})^{-1}}{\|x_i - x_{i-1}\|_2^2 + (x_i - x_{i-1})^TJ(x_{i-1})^{-1}f(x_i)}$$

Above equation represents an iterative formula for the Jacobian Inverse, which is major outcome of Broyden's method

Algorithm for Broyden's method:

Step 1: $x^{(0)} = \left(x_1^{(0)}, x_2(0), \dots, x_n(0)\right)$ be a given initial vector, considering it calculate $f\big(x^{(0)}$.

Step 2: Broyden's method allow us to let $A_0 = J\big(x^{(0)}\big)$, which implies that $A^{-1} = J\big(x^{(0)}\big)^{-1}$, calculate $J\big(x^{(0)}\big)^{-1}$.

Step 3: Calculate $x^{(1)} = x^{(0)} - J\big(x^{(0)}\big)^{-1}f\big(x^{(0)}\big)$

Step 4: Calculate $f\big(x^{(1)}\big)$ using $x^{(1)}$ from previous step

Step 5: Compute $\big(x^{(1)} - x^{(0)}\big)$

Step 6: Compute $J\big(x^{(1)}\big)^{-1} = J\big(x^{(0)}\big)^{-1} - \frac{J\left(x^{(0)}\right)^{-1}f\left(x^{(1)}\right)\left(x^{(1)}-x^{(0)}\right)^TJ\left(x^{(0)}\right)^{-1}}{\|\left(x^{(1)}-x^{(0)}\right)\|_2^2+\left(x^{(1)}-x^{(0)}\right)J\left(x^{(0)}\right)^{-1}f\left(x^{(1)}\right)}$ using the broyden's formula and previous step.

Step 7: Take $J\big(x^{(1)}\big)^{-1}$ and calculate $x^{(2)} = x^{(1)} - J\big(x^{(1)}\big)^{-1}f\big(x^{(1)}\big)$

Step 8: Repeat the iteration process until converge occurs to $\bar{x}$, i.e. when $x^{(i)} = x^{(i+1)} = \bar{x}$.

Finite Difference approximation of Jacobian

In finite difference approach the Jacobian from the newton method is replaced by a finite difference approximation of the derivatives in the Jacobian. The Finite difference approximation of derivatives is :

$$f'(x) = \frac{f(x+\varepsilon) - f(x)}{\varepsilon}$$

where the accuracy depends upon the ε. The elements of the jacobian is defined as $J_{ij}(x) = \frac{\partial f_i}{\partial x_i}$, and these can be approximated by finite difference as:

$$\frac{\partial f_i}{\partial x_i} = \frac{f_i(x_i + \varepsilon_i e_j)}{\varepsilon_i}$$

Where e_j is the unit vector for which $x.e_j = x_j$.

Similarly, Fixed difference is used for the columns of the Jacobian:

$$J_j^C = \frac{f(x_i + \varepsilon e_j) - f(x_i)}{\varepsilon_i}$$

Whenever, the step-size in finite difference is chosen accurately, it can be shown that the quadratic convergence of the method is still retained

$$J_j^C = \frac{f(x_i + \varepsilon_i e_j) - f(x_i)}{\varepsilon_i} \text{ for } j = 1 \ldots n$$
$$x_{i+1} = x_i - J_j^C f(x_i) \text{ for } i = 1, \ldots, n$$

The above equation are very well -defined and converge sufficiently for the local unique solution x^8. If

$$\lim_{i \to 0} \varepsilon_i = 0$$

The convergence is super-linear , if there exists a constant k_1 such that

$$\|\varepsilon_i\| = k_1 \|x_i - x^*\|$$

Convergence evolve to q-quadratic if the above equation is equivalent to some other constant k_2 such that

$$\|\varepsilon_i\| = k_2 \|f(x_i)\|$$

Now, MATLAB has a non-linear equation solver, it uses Newton Raphson Method to find roots of equation provided an initial guess, so requirement of computing Jacobian efficiently requires a simulation code, moreover if there are 'n' elements of ' x ' in jacobian, a call of two function per column has to be made i.e $2n$, the simulation code for this is given is Appendices..

APPLICATION IN ENGINERRING FIELD

Almost all engineering calculations involve solutions of non-linear equations, and in a single steady state simulation, these are typically non-linear systems. Most of the time, these equations are solved using the Newton-Raphson method. In electric power engineering, the Newton-Raphson method is used to solve the power flow (sometimes called load flow) problem. This is one of the critical problems that underpins many other problems encountered in power system studies, such as fault analysis, relay coordination, security, contingency, and so on [9] .The steady- equation which is a non-linear partial differential equation for the velocity field and the pressure in fluid is solved using the quasi-newton method. In the thermal field, the EHD lubrication problem is solved using Newton-Raphson, providing a very stable numerical estimation for the solution. Another simple utilisation of this method in engineering is the Cole-Brook Equation:

$$\frac{1}{\sqrt{f}} = -2\log_{10}\left[\frac{\varepsilon}{3.71D_h} + \frac{2.51}{\mathrm{Re}\sqrt{f}}\right]$$

Where D is the dimension of the object, and Re is the Reynold's number used to find the friction coefficient f. This equation has a solution in terms of the Lambert function, but the initial guess approach leads to the Newton-Raphson method to a higher accuracy of the solution. The system of the equation formulated in the finite elemental simulation of constituents of ferroelectroelastic material is non-linear, which requires a modified version of the method to reduce the convergence complexities, which further can be used on an integration point level [10] In the triangulation of co-ordinates of GPS problems finding coordinates over a system, the Newton-Raphson Method is needed.

The equation for triangulation in any space is given as:

$$\sqrt{(x-x_i)^2 + (y-y_i)^2 + \cdots} - c.dT = d_i$$

This existing system is solved using Newton-Raphson and then transformed into a non-linear system.

$$f_1(x,y) = \sqrt{(x-x_1)^2 + (y-y_1)^2} \cdots - d_1$$
$$f_2(x,y) = \sqrt{(x-x_2)^2 + (y-y_2)^2} \cdots - d_2$$

Now for an initial guess $r_0 = (x_0, y_0)$, a new solution can be found by iterating the recurrence i.e.

$$r_{n+1} = r_n - J^{-1}(r_n)F(r_n)$$

Iterative algorithms are the only ones capable of solving non-linear electromagnetic field problems. The fixed point approach and the Newton-Raphson method are the two most popular methods. While the second technique has quadratic convergence speed, the first is renowned as a steady but extremely slow algorithm that is far from the solution of non-linear equations. The Newton-Raphson scheme can, however, diverge occasionally. The polarisation formulation can manage the non-linear properties of ferromagnetic materials.

This formulation is used to reformulate the non-linear constitutive relations between the magnetic field intensity and the magnetic flux density as

$$\mathbb{B} = \mu\mathbb{H} + R \text{ and } \mathbb{H} = \varphi + \mathbb{I}$$

where $\mathbb{R}$ and $\mathbb{I}$ are iteratively calculated residual terms and μ and φ are permeability quantities splitting the hysterias model into linear and non-linear parts.

NUMERICAL EXAMPLES

In this section, several numerical examples are given to show the efficiency of our proposed method for approximating the solution of non-linear system of equation using Newton-Raphson and quasi newton method.

Example 1 Solve the Non-linear system in

$$\begin{aligned} &3x_1 - \cos(x_2x_3) - \tfrac{1}{2} = 0 \\ &x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 = 0 \\ &e^{-x_1x_2} + 20x_3 + \tfrac{10\pi - 3}{3} = 0 \end{aligned}$$

when $x^{(0)} = \begin{bmatrix} 0.1 \\ 0.1 \\ -0.1 \end{bmatrix}$.

First evaluation of the function and its jacobian must be done:

- Step 1:Define $f(x)$ and $J(x)$:

$$f(x) = \begin{bmatrix} 3x_1 - \cos(x_2x_3) - \dfrac{1}{2} \\ x_1^2 - 81(x_2 + 0.1)^2 + \sin x_3 + 1.06 \\ e^{-x_1x_2} + 20x_3 + \dfrac{10\pi - 3}{3} \end{bmatrix}$$

$$J(x) = \begin{bmatrix} 3 & x_3\sin(x_2x_3) & x_2\sin(x_2x_3) \\ 2x_1 & -162(x_2 + 0.1) & \cos(x_3) \\ -x_2e^{-x_1x_2} & -x_1e^{-x_1x_2} & 20 \end{bmatrix}$$

- Step 2: Finding the Values of $f(x)$ and $J(x)$ at $x^{(0)} = (0.1, 0.1, -0.1)^T$

$$f(x) = \begin{bmatrix} -1.19995 \\ -2.269833417 \\ 8.462025346 \end{bmatrix}$$

$$J(x) = \begin{bmatrix} 3 & x_3\sin(x_2x_3) & x_2\sin(x_2x_3) \\ 2x_1 & -162(x_2 + 0.1) & \cos(x_3) \\ -x_2e^{-x_1x_2} & -x_1e^{-x_1x_2} & 20 \end{bmatrix}$$

- Step 3: Solve the system $J\left(x^{(0)}\right)\partial x^{(0)} = -f\left(x^{(0)}\right)$ for finding $\partial x^{(0)}$ using Gauss Elimination:

$$\begin{bmatrix} 3 & x_3\sin(x_2x_3) & x_2\sin(x_2x_3) \\ 2x_1 & -162(x_2+0.1) & \cos(x_3) \\ -x_2e^{-x_1x_2} & -x_1e^{-x_1x_2} & 20 \end{bmatrix} \begin{bmatrix} \partial x_1^{(0)} \\ \partial x_2^{(0)} \\ \vdots \\ \partial x_n^{(0)} \end{bmatrix} = -\begin{bmatrix} -1.19995 \\ -2.269833417 \\ 8.462025346 \end{bmatrix}$$

such that the yield after solving is

$$\partial x^{(0)} = \begin{bmatrix} 0.40003702 \\ -0.08053314 \\ -0.42152047 \end{bmatrix}$$

- step 4: Using the yield of Step 3, $x^{(1)} = \partial x^{(0)} + x^{(0)}$:

$$x^{(1)} = \begin{bmatrix} 0.1 \\ 0.1 \\ -0.1 \end{bmatrix} + \begin{bmatrix} 0.40003702 \\ -0.08053314 \\ -0.42152047 \end{bmatrix} = \begin{bmatrix} 0.50003702 \\ 0.01946686 \\ -0.52152047 \end{bmatrix}$$

Using the results of $x^{(1)}$ to find our next iteration x(2) by using the same algorithm.

- Step 5: Repeating the above process till one of the stopping criteria meets (i.e. $\|x^{(k)} - x^{(k-1)}\|$) the following iterations are obtained:

TABLE II: Iteration table for Example 1

Value of k	$x_1^{(k)}$	$(x_2^{(k)})$	$x_3^{(k)}$	$\|\|x_n - x_{n-1}\|\|_2$
0	0.10000000	0.10000000	-0.10000000	-
1	0.50003702	0.01946686	-0.52152047	0.422
2	0.50004593	0.00158859	-0.52355711	0.0179
3	0.50000034	0.00001244	-0.52359845	0.00158
4	0.50000000	0.00000000	-0.52359877	0.0000124
5	0.50000000	0.00000000	-0.52359877	0

Hence, From the Iteration table final solution is as:

$$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0.50000000 \\ 0.00000000 \\ -0.52359877 \end{bmatrix}$$

This example shows the use of step-norm criteria to determine the final iteration of the non-linear system.

Example 2 Solve for the roots of equation of non-linear system of intersection of circles -

$$F_1(x_1, x_2) = (x_1 - 2)^2 + (x_2 - 2)^2 - 9 = 0$$
$$F_2(x_1, x_2) = (x_1 + 3)^2 + (x_2 + 1)^2 - 9 = 0$$

First for the given system, find the Jacobian:

$$\begin{bmatrix} 2(x_1 - 2) & 2(x_2 - 2) \\ 3(x_1 + 3) & 2(x_2 + 1) \end{bmatrix}$$

Taking Initial guess as x_0 as per table, and finding the value of the function and the value of the Jacobian at the initial guess. The numerical approximation is in the table below

Value of n	x_n	$f(x_n)$	$\|x_n - x_{n-1}\|_2$	
0	$(-1.00,3.00)$	$(1.00,11.0)$	0.00	0.556
1	$(-1.25,1.75)$	$(1.63,1.63)$	0.173	0.020
2	$(-0.963,1.27)$	$(0.310,0.310)$	0.439	
3	$(-0.875,1.124)$	$(0.030,0.030)$	0.42	
4	$(-0.864,1.101)$	$(0.004,0.004)$	0.006	

TABLE III: Iteration table for Example 2

This example shows the use of function-norm criteria to determine the final iteration of the non-linear system. Applying the Iterative algorithm of newton - raphson and moving along the approximations, a sudden downfall in the magnitude of values can be seen, at fourth iteration $(n = 4)$ the value of Step norm $\|x_n - x_{n-1}\|_2$ is smaller than tolerance so as the function's norm $\|f(x_n)\|_2$ corresponding to which the function's value $f(x)$ is closer to zero, hence it converge sufficiently and the fourth approximation is the numerical solution of the given non-linear system. This example shows the use of function-norm criteria to determine the final iteration of the non-linear system.

Example 3 Let

$$F(x) = \begin{bmatrix} x_1 + x_2 - 3 \\ x_1^2 + x_2^2 - 9 \end{bmatrix}$$

which has roots $(0,3)^T$ and $(3,0)^T$. Let $x_0 = (1,5)^T$, use Broyden's algorithm with

$$A_o = J(x_o) = \begin{bmatrix} 1 & 1 \\ 2 & 10 \end{bmatrix}$$

Then

$$F(x_o) = \begin{bmatrix} 3 \\ 17 \end{bmatrix},\ s_o = -A_o^{-1}F(x_o) = \begin{bmatrix} -1.625 \\ 1.375 \end{bmatrix}$$
$$x_1 = x_o + s_o = \begin{bmatrix} -0.625 \\ 3.625 \end{bmatrix},\ F(x_1) = \begin{bmatrix} 0 \\ 4.53125 \end{bmatrix}$$

The iteration for A_1 is given as

$$A_1 = A_o + \begin{bmatrix} 0 & 0 \\ -1.625 & -1.375 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0.375 & 8.625 \end{bmatrix}$$

From reading it can be confirmed $A_1 s_o = y_o$.such that

$$J(x_1) = \begin{bmatrix} 1 & 1 \\ -1.25 & 7.25 \end{bmatrix}$$

so that A_1 is not close to $J(x_1)$. At the next iteration:

$$s_1 = -A_1^{-1}F(x_1) = \begin{bmatrix} 0.549 \\ -0.549 \end{bmatrix},\ x_2 = x_1 + s_1 = \begin{bmatrix} -0.076 \\ 3.076 \end{bmatrix}$$
$$F(x_2) \equiv \begin{bmatrix} 0 \\ 0.466 \end{bmatrix},\ A_2 \equiv \begin{bmatrix} 1 & 1 \\ -0.799 & 8.201 \end{bmatrix}$$

Again A_2 is not close to

$$J(x_2) = \begin{bmatrix} 1 & 1 \\ -0.152 & 6.152 \end{bmatrix}$$

Below are given the entire iteration sequences for Broyden's approach and, for comparison, Newton's method's iteration are provided. For $k \geq 1, (x_k)_1 + (x_k)_2 = 3$ for both methods; so only $(x_k)_2$ is listed below:

TABLE IV: Comparison of Newton's with Broyden's Method in Example 3

Broyden's Method	x_i	Newton's Method
$(1,5)^T$	x_0	3.625
3.625	x_1	3.0919117647059
3.0757575757575	x_2	3.0026533419372
3.0127942681679	x_3	3.0000023425973
3.0003138243387	x_4	3.0000000000018
3.0000013325618	x_5	3.0
3.0000000001394	x_6	–
3.0	x_7	

CONCLUSION

It is reasonable to conclude from this work that numerical techniques are an important aspect of mathematics. Furthermore, non-linear equation systems account for the majority of real-world problems. Apart from Linear Algebra for solving linear systems numerical methods are used to solve non-linear algebraic equations in one variable as well as multiple variables. The paper's key findings may be divided into few categories: elaboration of the method

convergence the role of Newton's technique its modification and the construction of quasi - newton methods such as Broyden's method and fixed-difference. In terms of convergence, we may say that a numerical approach with a faster rate of convergence may be able to solve a problem in fewer iterations than a method with a slower rate of convergence. Newton's approach, for example, converges quadratically and is efficient in several complicated problems, Newton's approach would reach the solution of $F = 0$ in fewer iterations than Broyden's method. The importance of Newton's technique in numerical approaches is the paper's second major finding. Newton's approach is integrated into both Broyden's method and the Finite-Difference method's algorithms. This highlights Newton's method's versatility, as it may be used to resolve a wide range of issues. This leads us to believe that Newton's technique is a significant step in the field of numerical methods. Over time, many simulation codes have been formulated for different types of models, especially finite-element models. The finite elemental non-linear equations on ferroelectroelasticity and other finite elemental systems are solved using evolved quasi-newton methods, covering concepts like Finite Difference approximation and computational Jacobian and its inverse in an iterative way. The engineering field is mostly influenced by this method, in which most of the major problems (work load) are solved using this method and set a foundation for other major problems.

ACKNOWLEDGMENTS

This review paper was partially supported by Chandigarh University. I thank my supervisor from the institution [Dr. Bhagat Singh], who provided insight and expertise that efficiently assisted this review paper. I thank my fellow classmates for the comments that greatly improved the manuscript. I am also immensely grateful for the faculty's comments on my previous manuscript, although any errors in the paper should not tarnish the reputations of these people.

REFERNCES

[1] J. Ypma, "Historical development of the newton–raphson method," SIAM review **37**, 531–551 (1995).

[2] A. Yah, N. Yaakob, M. E. Elobaid, O. B. Lynn, R. Badlishah, and W. A. N. W. Abdullah, "Newton-raphson method to solve systems of non-linear equations in vanet performance optimization," Bulletin of Electrical Engineering and Informatics **8**, 336–342 (2019).

[3] Akram and Q. U. Ann, "Newton raphson method," International Journal of Scientific & Engineering Research **6**, 1748–1752 (2015).

[4] Xinghua, "Convergence of newton's method and inverse function theorem in banach space," Mathematics of Computation **68**, 169–186 (1999).

[5] Ben-Israel *et al.*, "A newton-raphson method for the solution of systems of equations," J. Math. Anal. Appl **15**, 243–252 (1966).

[6] Soram, S. Roy, S. R. Singh, M. Khomdram, S. Yaikhom, and S. Takhellambam, "On the rate of convergence of newton-raphson method," The International Journal of Engineering and Science (IJES) **2**, 05–12 (2013).

[7] Altman and P. Boulos, "Convergence of newton method in nonlinear network analysis," Mathematical and computer modelling **21**, 35–41 (1995).

[8] Al-Baali, E. Spedicato, and F. Maggioni, "Broyden's quasi-newton methods for a nonlinear system of equations and unconstrained opti- mization: a review and open problems," Optimization Methods and Software **29**, 937–954 (2014).

[9] B. Kisabo, N. C. Uchenna, and F. A. Adebimpe, "Newton's method for solving non-linear system of algebraic equations (nlsaes) with matlab/simulink® and maple®," American Journal of Mathematical and Computer Modelling **2**, 117–131 (2017).

[10] Stark, S. Roth, P. Neumeister, and H. Balke, "Modifications of the newton–raphson method for finite element simulations in ferroelectroe- lasticity," International Journal of Solids and Structures **50**, 773–780 (2013).

[11] E. Dennis Jr and R. B. Schnabel, Numerical methods for unconstrained optimization and nonlinear equations (SIAM, 1996).

[12] Kuczmann, "Using the newton–raphson method in the polarization technique to solve nonlinear static magnetic field problems," IEEE Transactions on Magnetics **46**, 875–879 (2010).

APPENDICES

SIMULATION CODES

• MATLAB helps in Improving Fixed Difference approximation method and increasing the accuracy of finding the Jacobian of a non- linear system. [9]

1. A MATLAB function that does function evaluation

```
function f = my_func(x)

f = %The nature of the function;
```

2. A MATLAB function that calculates the Jacobian:

```
 function J = my_jacobian (x)

j= zeros( length(x), length(x) );

for i = 1:length (x)

dx = x; eps = 10^-8 * x(i);

dx(i) = dx(i) + eps;

J( :, i) = ( my_func(dx)– my_func(x) )  / eps;

end;
```

• MATLAB Simulation code for solving the Colebrook's equation using Newton-Raphson for major parameters including the roughness and Reynolds number

```
e=input('Enter the value of the roughness, epsilon: ');
D=input('Enter the value of the dimension, D: ');
R=input('Enter the value of the Reynolds Number, Re: ');
f=@(x) 1/sqrt(x);
g=@(x) -2*log(e/(3.7*D) + 2.51/(R*sqrt(x)));
h=@(x) g(x)-f(x);
n=@(x) diff(h(x));
x0 = 0.01; %Guess for initial x
x=x0;
dx = 1e-3; % Step size for the centered finite difference
for i=0:100000
   y=x;
   m = (h(x+dx) - h(x-dx)) / (2*dx); % centered finite difference
   x = y-(h(x)/m);
   if x==y
break end
end
fprintf('The total number of iterations are: ');
i;
x;
```