Numerical Methods for Solving Nonlinear Equations

Narges Mohammadi, Shahram Mehdipour-Ataei, and Maryam Mohammadi

Abstract The nonlinear description has continuously been crucial in a wide range of disciplines to provide an accurate prediction of a natural phenomenon. Thus, finding a reliable solution method for these nonlinear models is of significant importance since, in most real-life applications, direct solution methods are not feasible, even in linear cases. Moreover, an inefficient method is likely to take additional computational cost and effort. This chapter attempts to provide a fundamental description of various iterative methods for solving nonlinear discretized equations. In the first part, a theoretical account of nonlinear systems with different types of iterative methods are depicted. The second part deals with both one-point and multi-point iterative methods; this includes a description of the method, mathematical formulations, and the weak and strong points. Different iterative methods to solve a system of nonlinear equations are then described. Some discussed methods include the family of conjugate gradient, multi-step, and Newton-like. This part also identifies intricacies regarding a system of nonlinear equations, offering different remedies to solve these issues. Finally, a comparative study of the discussed methods and their applications in solving conventional equations are outlined in brief. The iterative methods mentioned in this chapter can be useful not only in solving nonlinear problems but also in linear problems and optimization.

Keyword Nonlinear equations · Iterative methods · Root-finding · Computational efficiency · Jacobian matrix

N. Mohammadi
Faculty of Mechanical Engineering, Amirkabir University of Technology, Tehran, Iran
e-mail: nargesmohammadi@aut.ac.ir

S. Mehdipour-Ataei (✉) · M. Mohammadi
Faculty of Polymer Science, Iran Polymer and Petrochemical Institute, Tehran, Iran
e-mail: s.mehdipour@ippi.ac.ir

M. Mohammadi
e-mail: ma.mohammadi@ippi.ac.ir

N. Mahdavi Tabatabaei and N. Bizon (eds.), *Numerical Methods for Energy Applications*, Power Systems,
https://doi.org/10.1007/978-3-030-62191-9_5

Nomenclatures

A. Acronyms

IVP	Initial Value Problems
BVP	Boundary Value Problems
ODEs	Ordinary Differential Equations
PDEs	Partial Differential Equations
FE	Finite Element
FV	Finite Volume
FD	Finite Difference
CO	Convergence Order
LU	Lower–Upper

B. Symbols/Parameters

x	Unknown value(s)
i	Iteration index
x^*	Actual root
$E.I.$	Efficiency index
ρ	Convergence order
n	Total number of function evaluation
$[a, b]$	Specific interval
$f(x)$	Nonlinear function
f'	First derivative of the nonlinear function
f''	Second derivative of the nonlinear function
$x^{(0)}$	Initial approximation of the zero of the function
C	Computacional cost
$J^{(i)}$	n-dimensional Jacobian matrix
$F(x)$	Column vector of nonlinear functions
$\frac{\partial}{\partial x}$	Partial derivative with respect to the unknown x
J	Jacobian matrix
h	Incriminate value
∇F	Gradient of $F(x)$
T	Transpose sign
$H(x)$	Hessian matrix
λ	Damping parameter
$\eta^{(i)}$	Forcing term
$S^{(i)}$	Inexact newton step

1 Introduction

There is a wide range of natural phenomena, as well as numerous practical applications that can be accurately simulated through mathematical analysis. The formulation of a real phenomenon in a format of the system of equations endows many benefits, such as validating the results of physical experiments and demonstrating a reliable relationship among variables explaining the features of a phenomenon or a system. A problem modeled in a frame of mathematical equations can be either linear or nonlinear. Linear problems are more straightforward to solve than nonlinear problems, with respect to computational cost and implementation. The linear analysis may be applied for nonlinear problems when errors are too small, or they can be tolerated; but in some problems, according to the required accuracy for a problem and the target of analysis, employing the nonlinear analysis is necessary and unavoidable. Some examples demonstrating the importance of a nonlinear description include the design of components for some special usage such as aerospace and nuclear engineering, obtaining an accurate understanding of a phenomenon, or simulating the behavior of some materials. Hence, complex nonlinear equations, in spite of intricacies stemming from solving them, should be employed to provide a perception of the behavior of these phenomena as well as present a realistic approximation of the response [1, 2].

Nonlinear equations can be either one-variable equations (scalar equations) or multi-variable equations (the system of nonlinear equations). A nonlinear equation can be represented as an equation that does not follow the superposition principle, and the output and input of the system are not directly proportional to each other. Regarding the system of nonlinear equations, it is a set of n simultaneous equations with n unknowns that consists of only one or more nonlinear equations [3]. In contrast to scalar nonlinear equations, the system of nonlinear equations due to their complexity has received less attention.

Solving a nonlinear equation is an intractable task; besides, the uniqueness and existence of the nonlinear solution may also be challenging. To solve these nonlinear problems, there are two numerical and analytical methods. Albeit the analytical method provides high accuracy without much computational cost and effort, in most cases, finding a closed-form solution for a nonlinear system is not feasible; these solutions are restricted to some simplified and exceptional cases and are not suitable for real applications. On the other hand, the growing knowledge and improvements of computer technology have made the use of complicated numerical methods an easy task for analyzing physical systems; by writing new programs or using the existing computer packages. Therefore, during the last decades, researchers have been stimulated to develop new computationally efficient methods to find a numerical method approximating a solution for nonlinear initial value problems (IVP) and boundary value problems (BVP) [1, 4, 5].

Nonlinear equations governing a system can take different forms of algebraic, differential– ordinary differential equations (ODEs), partial differential equations (PDEs)—and integral equations. Regarding differential or integral equations,

initially, one of the well-known numerical methods such as Finite Element (FE), Finite Volume (FV), Finite Difference (FD), or other discretization methods may be employed to convert nonlinear equations to nonlinear algebraic equations. These discretization techniques provide a weakly nonlinear system of equations. Then, an iteration method should be adopted to solve this weakly system of nonlinear equations [1, 6].

In iteration methods, by an initial guess, a particular process like a generalized rule should be followed for every new estimation of the final response until the termination criterion meets. Termination criterion is a condition determining an acceptability level for the final allowable error that should be satisfied to terminate to an iteration process. The termination criterion for a small value ε, and the approximated values x with the iteration index i can be defined in different ways. Three commonly used of these criteria are as:

(1) Two last responses are very close together:

$$\left|x^{(i+1)} - x^{(i)}\right| < \varepsilon \tag{1}$$

(2) The relative difference of two last responses are small:

$$\left|\frac{x^{(i+1)} - x^{(i)}}{x^{(i)}}\right| < \varepsilon \tag{2}$$

(3) The final response is sufficiently small [2, 7, 8]:

$$\left|f(x^{(i)})\right| < \varepsilon \tag{3}$$

The value of ε can be varied according to each problem and the final goal of that problem. As a numerical method provides an estimate of the exact response, the termination criterion is an important factor defining the accuracy and reliability of results. Moreover, this criterion can significantly affect the time of the process.

According to Traub [9], iterative methods to solve nonlinear equations can be divided into two categories viz one-point and multi-point iterative methods, regarding the fact that whether they employ new data from different points or not. These two methods are also divided into multi-point methods with memory or without memory and one-point methods with memory or without memory based on whether old information is reused or not. In one-point methods without memory, the value of a new estimation, say x_i, is computed only based on the information at x_i. In this method, the only way to enhance the convergence of the problem is by increasing the derivative order. For example, to achieve a method of convergence order i, one should employ the $(i-1)$th derivative of a function. Some examples of this method are Newton, Halley, and Cauchy methods. A method is named a one-point method with memory when the next approximation is obtained based on the information of older points such as $x^{(i-1)}, x^{(i-2)}, \ldots, x^{(1)}$ as well as $x^{(i)}$ to estimate the value of $x^{(i)}$.

One well-known example of this case is the Secant method. This type of iterative methods mostly includes derivative-free algorithms.

Regarding multi-point iterative methods, they use the new information at different points, which results in a more computationally efficient method in comparison to one-point methods. These methods can increase the order of convergence without any need to employ a higher-order derivative as well as provide a wider region of convergence [10].

To select an iterative numerical method for a nonlinear problem, there are some criteria that should be considered to ensure the efficient performance of a solver in providing an accurate result. In the first step, the convergence order (CO) should be measured to identify the speed of a numerical method to obtain the final response. According to Eq. (4), for $\rho \geq 1$ and $K > 0$, if a function with the actual root of x^* satisfies this relationship, it is said that the method converges to x^* with order ρ. This relationship can be calculated based on either n-dimensional Taylor expansion or the matrix approach.

$$\lim_{n\to\infty} \left|x^{(i+1)} - x^*\right| \leq K \left|x^{(i)} - x^*\right|^{\rho} \tag{4}$$

In addition, if $\rho = 1$ and $K = 1$, it is said that the method is super-linearly convergent, which is faster than linearly convergent [11].

With respect to the convergence, iterative methods can be either locally convergent or globally convergent methods. In the locally convergent method, the order of convergence is greater than 1. Moreover, to ensure the convergence to the accurate root, this method requires an initial guess sufficiently close to the root. In contrast, a globally convergent method is not restricted to a good initial approximation to converge, but the convergence order of these methods is often lower than locally convergent methods. For example, the Newton method is a second-order locally convergent method highly sensitive to the initial guess; whereas the bisection method linearly converges even for a poor initial approximation [12].

The number of call functions or function evaluations required at each step can also determine the efficiency and the CPU time (running time of an algorithm) required for a process; this can be determined by the number of times that goal function and its supporting functions computed during the process. In a system of nonlinear equations along with the number of function evaluations, other operations such as matrix-vector multiplications, matrix-matrix multiplications, Jacobian evaluations, and Jacobian inverses calculation are deemed factors determining the efficiency of a method. These factors are important since they may cause a very time-consuming process, in particular, in temporal problems in engineering disciplines, which require the repetition of a lengthy iterative procedure to solve the system of nonlinear equations per time step. Another factor is the radius of convergence that is often ignored

since it is difficult to calculate. Lastly, the CPU time used for a process can again determine the efficiency and speed of a method, in particular, for a large nonlinear system. Obviously, a process with less CPU time is a better method. All factors mentioned above can affect each other; for example, to obtain more accuracy, a method should use more function evaluations [7, 13, 14]. On the other hand, considering all these factors for each problem is intractable; therefore, regarding each problem, some of these criteria may be ignored; for example, to reach a higher accuracy, one may have to employ more function evaluations. Hence, it is of great importance to identify various numerical methods and their characters so that the most suitable one can be opted according to the circumstances.

To this aim, the rest of the chapter depicts a fundamental description of various iterative methods for solving nonlinear algebraic equations originating from discretized partial and ordinary derivative or integral equations. In the following parts, initially, both on-point and multi-point iterative methods to solve scalar equations are described; they are compared from a computational viewpoint. Then, different commonly-used methods to solve a system of nonlinear equations are described. Advantages and drawbacks of each method are mentioned, and some alternative methods to overcome their weaknesses are represented. Finally, the application of the methods together with a comparative study are provided, in the last part.

2 One-Variable Nonlinear Equations

In this part, a description of different methods to solve one-variable nonlinear equations is presented. Although nonlinear scalar equations are only employed to some limited cases, a knowledge of them can assist researchers in comprehending possible solutions for a system of nonlinear equations more straightforward. Furthermore, basic ideas of some methods to solve one-variable problems can also be used for nonlinear multi-variable problems. Methods depicted in this part consist of one-point and multi-point methods.

In order to evaluate the efficiency of various scalar iterative methods and compare different offered methods with each other, the efficiency index $E.I.$ can be defined as a function of the convergence order ρ and the total number of function evaluation n as $E.I. = \rho^{\frac{1}{n}}$. The higher efficiency index means that a method is of higher speed with lower computationally cost.

2.1 One-Point Methods

One-point iterative methods are the simplest methods for solving scalar nonlinear equations that are not often adopted for complicated problems. These methods can be used either as a base for more complicated methods, such as the Newton and

Steffensen methods, or as a starting method for locally convergent methods like bisection method.

I. **Bisection Method**

By assuming f as a continuous scalar function at a specific interval, say $[a, b]$, in which the sign of $f(a)$ and $f(b)$ are opposite, the bisection method determines that there is at least one root in this interval based on the intermediate value theorem. This method computes the value of f at the midpoint of the domain $c = \frac{a+b}{2}$. Then, according to the sign of the $f(c)$, the value of c is replaced with either a or b. As a result, a new smaller subinterval around the root of the equation can be established for the next step. This procedure recurs until the termination criterion meets [15].

One of the advantages of the halving method is that it only uses one function evaluation per iteration, except for the first step, it employs two functions that can be overlooked by the final number of functions used at the end of the process. This method also unconditionally converges to the actual root provided that the nonlinear function is continuous in the whole of the domain. Moreover, the number of iterations for a specified accuracy can be obtained in advance; for example, the error after n iteration will be less than $\left|\frac{b-a}{2^n}\right|$. Nevertheless, this method is not widely used for complicated nonlinear equations due to the low rate of convergence (linear convergence) for the same accuracy in comparison with other methods. Additionally, for a function with multiple roots in an interval, this method may not work accurately, since the sign of endpoints may be the same. Therefore, it is proposed to employ this method to find a rough estimate of a function for other faster methods requiring a proper approximate initial value; that is, for the first iterations, this method can be applied to obtain an approximation sufficiently close to the root, and for the next iterations, other faster methods can be used to accelerate the speed of the calculation [8, 15].

II. **Fixed-Point Method**

This method is a simple and stable method that can be employed to solve continuous functions. This method initially converts the function of $f(x) = 0$ to $g(x) - x = 0$. Then, to find the root of $f(x)$, it addresses the iteration equation as $x^{(i+1)} = g(x^{(i)})$; in each iterate, the response of the function is set as the new point. As the bisection method, this method is also linearly convergent.

The rate of convergence of this method may change for a different arrangement, and even for some arrangement, it either may not converge or converge to a false root. There is not a general rule to determine what arrangement is the best; nevertheless, based on the previous research, it can seem that the slope of the curve of the function $g(x)$ can affect the speed of the algorithm [8, 15].

III. **Muller Method**

In contrast with previous methods, this method approximates the function $f(x)$ by a quadratic polynomial as $ax^2 + bx + c$, which can increase the speed of the method.

By using three points of x_0 and x_2 as endpoints and x_1 between two endpoints, near the zero of the function, the coefficients of this quadratic polynomial can be obtained. In the next step, the nearest root to the middle point, x_1, is set as the new point. The resulting value is assigned as a new point for the next approximation along with the two points that are closer to the root.

Muller's method can be used to obtain complex root if the starting point is a complex number. In addition, except for the first iteration that requires three function evaluations, this method only employs one function evaluation per iteration [8, 15].

IV. Newton Method

The well-known Newton method approximates the roots of a nonlinear functions on the basis of the first two terms of the Taylor polynomial expansion of the nonlinear function $f(x)$ expanded about the actual root of the equation x^* as:

$$f(x^*) = f(x^{(0)}) + f(x^* - x^{(0)})f'(x^{(0)}) + \frac{(x^* - x^{(0)})^2}{2} f''(x^{(0)}) \tag{5}$$

where f' and f'' are the first and second derivative of the $f(x)$, respectively, and $x^{(0)}$ is the initial approximation of the zero of the function, which is sufficiently close to it. Since $f(x^*) = 0$ and $\left|x^* - x^{(0)}\right|$ is small enough, the terms with higher order can be ignored, and the iterative method for $n > 1$ iteration can be shown as:

$$x^* \approx x^{(i+1)} = x^{(i)} - (x^{(i+1)} - x^{(i)}) \frac{f(x^{(i)})}{f'(x^{(i)})} \tag{6}$$

According to the above equation, the new point, $x^{(i+1)}$, is considered as the x-intercept of the tangent line to the graph of $f(x)$ at the current approximation x_n.

This method is a highly efficient method that converges quadratically. Additionally, it can work even for complex roots, providing that the initial guess be a complex value. It should be pointed out that this initial root should be sufficiently close to the actual root to avoid this method from being trapped in an endless loop. In comparison with previous methods, this method employs two function evaluations at each step. Additionally, this method requires the second derivative of the nonlinear equation, which may be either unavailable for some problems or burdensome to calculate. Therefore, concerning these potential hurdles, some alternative methods have existed to overcome some. In the following, a number of these methods are presented [16, 17].

V. Secant Method

The secant method is a quasi-Newton method in which the derivative of the function is estimated with the finite-difference approximation. The iteration algorithm of this

method can be shown as:

$$x^{(i+1)} = x^{(i)} - f(x^{(i)}) \frac{(x^{(i+1)} - x^{(i)})}{f(x^{(i+1)}) - f(x^{(i)})} \tag{7}$$

This formulation offers a derivative-free method at the expense of decreasing the convergence order to super linearly convergence. As the Newton method, this method requires to combine with other methods such as the bisection method since this method still requires two good starting points of the zero of the function. The method of False Position can be suggested to prevent this algorithm from diverging. Although this method is of slow convergence and requires more calculations, it always ensures that during the iteration process, the new estimation is always bracketed; that is, the function $f(x)$ changes signs at two x-values [8, 15].

VI. **Steffensen Method**

Unlike previous method, Steffensen method offers a convergence identical to the Newton method; whereas, it does not involve the derivatives of the function. Using divided difference, this derivative-free method offers a proper one-point algorithm for non-differentiable problems as below [18]:

$$x^{(i+1)} = x^{(i)} - \frac{f^2(x^{(i)})}{f(x^{(i)} + f(x^{(i)})) - f(x^{(i)})} \tag{8}$$

VII. **Chebyshev–Halley Methods**

The well-known Chebyshev–Halley methods are improved versions of the Newton method, offering a higher convergence order for the iterative method using the second derivative of the function. This method can be shown as:

$$x^{(i+1)} = x^{(i)} - \left(1 + \frac{L_f(x^{(i)})}{2(1 - \beta L_f(x^{(i)}))}\right) \frac{f(x^{(i)})}{f'(x^{(i)})} \tag{9}$$

$$L_f(x^{(i)}) = \frac{f''(x^{(i)}) f(x^{(i)})}{(f'(x^{(i)}))^2} \tag{10}$$

For different values of β this formulation includes three well-known methods as: Chebyshev method for $\beta = 0$, Halley method for $\beta = \frac{1}{2}$, and Super-Halley method for $\beta = 1$.

As the Newton method, these methods still require a good initial guess in the neighborhood of the root. These well-known methods provide cubic convergence order; however, they are not suggested to use since they require second-order derivative. Generally speaking, in most problems, it is attempted to avoid higher-order derivative because of some potential challenges concerning them. As stated before, methods in which higher-order derivatives are required may be cumbersome to deal. This issue can even be more problematic about a system of nonlinear equations with

multi variables. In these problems, some additional problems such as the calculation of the Jacobian matrix and the difficulty thereof may arise. Hence, the enhancement of the convergence order at the expense of the employing higher derivative is often deemed to be impracticable. As an alternative solution to obtaining higher convergence, multi-point methods can be offered. These methods can improve the convergence order without any need to employ a higher derivative. In the next part, this method is represented [19, 20].

2.2 Multi-point Methods

All methods discussed so far were based on the one-point iterative methods. As stated, these methods are of a lower convergence order; hence, to improve their convergence, multi-point methods are presented. These methods are mainly proposed to enhance the convergence order of existing methods, such as the Newton and Steffensen methods, by adding one or more steps along with employing more variables, without using higher-order derivative.

It should be noted that these methods require more function evaluations; therefore, although every added step can increase the convergence, it may not be computationally efficient. In other words, it can be said that every multi-step method is not suitable to use since some of them result in the same efficiency as other existing methods with lower computational efficiency. Therefore, as stated in previous part, the efficiency index $E.I.$ is used to determine the performance of a method; that is, the number of function evaluations used is whether or not proportional to the corresponding accuracy.

As an example of a multi-point method, consider the following algorithm based on the Newton method in three steps.

$$\begin{cases} y^{(i)} = x^{(i)} - \frac{f(x^{(i)})}{f'(x^{(i)})} \\ z^{(i)} = y^{(i)} - \frac{f(y^{(i)})}{f'(y^{(i)})} \\ x^{(i+1)} = z^{(i)} - \frac{f(z^{(i)})}{f'(z^{(i)})} \end{cases} \tag{11}$$

At first sight, it may seem an efficient method since it increases the convergence order of the Newton method to eight; but, this algorithm requires six function evaluations—three functions and three derivatives of the function—that results in $8^{1/6} = 1.414$efficiency index, which is as equal as the Newton method [10, 21]. Therefore, it should be attempted to employ or offer multi-step methods with optimum convergence order; for example, the Ostrowski method is as an improved Newton method in two steps by adding more variable as:

$$\begin{cases} y^{(i)} = x^{(i)} - \frac{f(x^{(i)})}{f'(x^{(i)})} \\ x^{(i+1)} = y^{(i)} - \frac{f(y^{(i)})(x^{(i)} - y^{(i)})}{f(x^{(i)}) - 2f(y^{(i)})} \end{cases} \tag{12}$$

This method increases the convergence order of the Newton method to four with employing only three function evaluations at each iteration—two functions and one first derivative evaluation. There are many other methods providing higher optimum accuracy in more steps, such as three-point methods with eighth-order convergence [22, 23] or four-point method with sixteenth-order convergence [24]. To obtain such optimum convergence order, appropriate weight functions can be employed. These weight functions can not only effectively increase the convergence order but also can be used for non-smooth functions. A through discussion of the different weight functions are beyond the scope of this text. Nevertheless, avid readers can find more detailed description in [25, 26].

There are also a number of multi-point iterative methods using the Steffensen method as the base method that may be appropriate for non-differentiable functions. Some of these derivative-free multi-point methods are given in studies [27, 28].

3 System of Nonlinear Equations

In previous parts, iterative methods for solving one-variable problems were described, but, in most real problems, one encounters more complex problems with more than one variable; in these cases, a system of nonlinear equations should be solved, which requires a different approach to one-variable problems. For example, in one-variable problems, attempts are made to provide a higher convergence order with lower function evaluations; whereas, in multi-variable problems, the goal is preventing the Jacobian matrix from being singular or ill-condition, or decreasing the computational evaluation concerning Jacobian matrix. Hence, regarding a system of nonlinear equations, to define the optimal convergence order, besides the number of function evaluations and convergence order, other factors such as Jacobian inversions, vector-vector, matrix–matrix, and matrix-vector multiplications should be considered as options affecting computational cost. Table 1 demonstrates the effect of these parameters on the total cost.

Therefore, the efficiency index of a system of nonlinear equations with n iteration can be defined in terms of the computational cost, C, as $E.I. = \rho^{\frac{1}{C}}$ [29].

Of previous methods to solve scalar nonlinear equations, only Newton and fixed-point methods can be extended for multi-variable nonlinear equations since the notations of other one-variable methods cannot be defined for this type of problems. In this part, other commonly-used methods such as Newton-like and multi-step methods are also proposed.

Table 1 Computational cost attributed to various operations used for solving a system of nonlinear equations [29]

Operations	Computational cost
LU-factorization	
Multiplications	$\frac{n(n-1)(2n-1)}{6}$
Divisions	$\frac{n(n-1)}{2}$
Total cost	$\frac{n(n-1)(2n-1)}{6} + 3\frac{n(n-1)}{2}$
Solution of lower and upper triangular systems	
Multiplications	$n(n-1)$
Divisions	n
Total cost	$n(n-1)+3n$
Scalar-vector multiplication	n
Point-wise vector-vector multiplication	n
Matrix-vector multiplication	n

3.1 Gauss–Seidel Method

As the simplest technique for solving nonlinear equations, in this method, instead of solving n nonlinear equations simultaneously, each nonlinear equation is assigned to one of the unknowns and solved to find that unknown separately. In other words, using an initial guess, say $x^{(0)} = \left[x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}\right]^T$, each equation $f_j(x^{(i)})$ is considered as a one-variable equation for each variable $x_j^{(i+1)}$, then it is solved using one of the on-variable methods. The procedure of this method can be shown as:

$$\begin{aligned}
&f_1(x_1^{(i+1)}, x_2^{(i)}, x_3^{(i)}, \ldots, x_{n-1}^{(i)}, x_n^{(i)}) = 0\\
&f_1(x_1^{(i+1)}, x_2^{(i+1)}, x_3^{(i)}, \ldots, x_{n-1}^{(i)}, x_n^{(i)}) = 0\\
&\qquad\qquad\vdots\\
&f_1(x_1^{(i+1)}, x_2^{(i+1)}, x_3^{(i+1)}, \ldots, x_{n-1}^{(i+1)}, x_n^{(i+1)}) = 0
\end{aligned} \tag{13}$$

If the new variables are not updated for the successive equations, the method converts to Gauss-Jacobi algorithm.

Generally, there are no scheme explaining which sequence should be considered for each equation and variable; nonetheless, it is preferable that if an equation contains only one variable, that equation and variable be considered as the first one. Both methods converge considerably slower than other methods. Additionally, this method may not work efficiently if the system is not diagonally dominant or positive-definite [30].

3.2 Fixed-Point Method

As the one-dimensional fixed-point method, initially, each equation is assigned to one variable; then, every equation is split and solve for that assigned variable.

$$\begin{aligned} x_1^{(i+1)} &= g_1(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, \ldots, x_n^{(i)}) \\ x_2^{(i+1)} &= g_2(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, \ldots, x_n^{(i)}) \\ &\vdots \\ x_3^{(i+1)} &= g_3(x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, \ldots, x_n^{(i)}) \end{aligned} \tag{14}$$

To simply accelerate the convergence order of this method, each new estimation from previous functions should be replaced in the next functions [8, 30].

It should be stressed that both fixed-point and Guess-Seidel methods neither require the derivative of a function nor involve high-computational complexities; besides, these methods can provide a good rough estimation for complicated problems. However, these methods can be applied for simple problems since these methods give a low convergence order. In other words, if one encounters problems demanding lengthy calculation of functions per iteration, the following faster methods are suggested in order to obtain the final result in fewer iteration [8, 30].

3.3 Newton Method

As the one-dimensional Newton method, by using multi-variable Taylor series expansion, the well-known Newton method for a system of nonlinear equations can be gained in a matrix form as:

$$F(x^{(i+1)}) = F(x^{(i)}) + J(x)(x^{(i+1)} - x^{(i)}) \tag{15}$$

in which $J^{(i)}$ is the n-dimensional Jacobian matrix for n-dimensional column vector of $F(x)$ and x is the column vector of the unknowns as:

$$x^{(i)} = [x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)}]^T \tag{16}$$

$$F(x) = [f_1(x), f_2(x), \ldots, f_n(x)]^T \tag{17}$$

$$J^{(i)} = \begin{bmatrix} \frac{\partial}{\partial x_1} f_1^{(i)} & \cdots & \frac{\partial}{\partial x_n} f_1^{(i)} \\ \vdots & \ddots & \vdots \\ \frac{\partial}{\partial x_1} f_n^{(i)} & \cdots & \frac{\partial}{\partial x_n} f_n^{(i)} \end{bmatrix} \tag{18}$$

where $\frac{\partial}{\partial x}$ is the partial derivative with respect to the unknowns of the problem.

As stated previously, despite being quadratically convergent, this method has many drawbacks, even more than the one-dimensional case. This method is too costly since numerous function evaluations, and the derivative thereof should be calculated at each iteration, in particular, for nonlinear systems with a large matrix. More importantly, the evaluation of the Jacobian matrix J per iteration can be a challenging task with an excess amount of time for each process; during the process, the Jacobian matrix may become ill-conditioned or singular, which cause the iteration procedure to diverge and influence the stability of this method. Another problem concerning this method is that the iteration process is highly sensitive to an accurate initial approximation to ensure convergence and control progress path towards the root of the function, even more than a one-variable case. Hence, this method may deem to be an expensive method to employ. Keeping this in mind, there are many remedies proposed to overcome some obstacles regarding these issues and enhance the efficiency of this method. In the next parts, the most important ones are discussed [31, 32].

3.4 Quasi-Newton Method

To release difficulties regarding the calculation of the Jacobian matrix for each iteration, quasi-Newton methods have been developed to estimate the Jacobian matrix at each iteration based on the previous steps.

The Broyden technique as a generalized secant method for multi-variable problems is a quasi-Newton method that can be used as an alternative method with a less computational cost. This method employs the Jacobian matrix for only the initial guess. Then, using the current approximation of the Jacobian matrix J_{n-1}, this method updates it at each iteration by the Eq. (19):

$$J^{(i+1)} = J^{(i)} + \frac{F^{(i)} - J^{(i)}x^{(i)}}{\left|x^{(i)}\right|^2}\left|x^{(i)}\right|^T \tag{19}$$

where $\Delta x^{(i)} = x^{(i+1)} - x^{(i)}$ and $\Delta F^{(i)} = F^{(i+1)} - F^{(i)}$. In the next step, the successive approximation of the solution can be obtained by substituting the new estimate of the Jacobian matrix in the Newton method. This approximation significantly decreases the computational cost and provides a superlinear convergence [33]. Moreover, unlike the Newton method, these methods are not self-correcting; that is, round-off errors are accumulated during successive iterations.

This technique, using the approximation of the Jacobian matrix instead of the actual value of it, has been developed by other researchers; the correlation method, Davidon-Fletcher-Powell formula, and Broyden-Fletcher-Goldfarb-Shanno method for example, are another method to estimate the Jacobian matrix instead of directly

calculate it. To obtain more insight into these methods, interested readers are referred to [34–36].

3.5 Steepest Descent Method

An effective way to solve a system of nonlinear equations is by transforming a root-finding problem to an optimization problem so that the global optimization methods can be used to solve the problem [37]. One of the well-known methods based on this idea is the first-order steepest descent method or gradient ascent method that ensures convergence even with a poor initial guess for a system of nonlinear equations. This method converts the system of nonlinear equations to a problem of minimizing the sum of the square of all nonlinear functions, $G(x)$, as:

$$G(x) = \sum_{n=1}^{m} f_n^2(x) \tag{20}$$

To calculate the minimum of the function $G(x)$, or zeros of nonlinear functions, this method moves in a direction in which the value of $G(x)$ decreases. To determine this direction, the property of the gradient of a function and the extreme value theorem—the maximum of a function occurs when is parallel to ∇G are employed. Accordingly, this method computes $-\nabla G$ as the direction in which the most decrease in the value of nonlinear functions occurs. Additionally, per Eq. (21), to increase the efficiency of the method, it is attempted to add and select the best value for α, which can improve the speed of the method towards the root.

$$x^{(i+1)} = x^{(i)} - \alpha\left(\nabla G\left(x^{(i)}\right)\right) \tag{21}$$

To find the best value for α resulting in faster convergence, readers are referred to the Ref. [8].

The speed of this algorithm is still slow. On the other hand, the calculation of the gradient of the sum of the square of nonlinear functions is a costly process for each iteration. As a consequence, this method may not be served as a computationally efficient method since it does not reduce the amount of calculation. Nevertheless, this method is suggested to use in combination with other methods, which are sensitive to the starting point, since it does not depend on the initial guess and can provide a good rough initial guess. As in the one-dimensional case, this method is similar to the bisection method used for a scalar equation [38].

3.6 Leven-Marquardt Method

Like the previous method, the Leven-Marquardt method is used to optimize the sum of the square of nonlinear functions, to find zeros of the system of nonlinear equations; except, it estimates the functions in terms of their Taylor expansion. In this method, firstly, the function of S in terms of the sum of the square of nonlinear functions is defined as:

$$S^2(x) = \sum_{n=1}^{m} [f(x)]^2 \tag{22}$$

By using the Taylor expansion (Eq. (16)) of the function $f(x)$, the value of S^2 can be yielded as:

$$S^2(x) = F(x)^T F(x) - 2F^T(x)J(x)h + J^T(x)J(x)h^2 \tag{23}$$

where h is the incriminate value equals $h = x^{(i+1)} - x^{(i)}$. Therefore, the minimum value of S^2 can be obtained by setting the derivative of the square of the nonlinear functions with respect to h equal to zero as:

$$\frac{\partial}{\partial h} S^2(x) = -2F^T(x)J(x) + 2h^T J^T(x)J(x) = 0 \tag{24}$$

where $J^T(x)J(x)$ and $J^T f(x)$ is the Hessian matrix, $H(x)$, and the gradient of $F(x)$, ∇F, respectively. Thus, by rewriting the Eq. (24), the Gauss–Newton iteration algorithm can be gained as:

$$x^{(i+1)} = x^{(i)} + H^{-1}(X)\nabla F \tag{25}$$

The above-mentioned equation, which is called Gauss-Newton method, can only be used when the initial guess is in the vicinity of the roots of the system of nonlinear equations. Therefore, to resolve this issue, the Levenberg–Marquardt method can be replaced. According to Eq. (26), by adding λ as a damping parameter varying between 0 and 1, this method can interpolate between two methods of steepest descent method and Gauss–Newton method. In other words, if the current solution is far from the actual response, the algorithm becomes slow like the steepest descent method and the value of λ is set to be large; while, when the approximated solution is near the accurate response, the speed of the algorithm increases, approaching a Gauss–Newton method. In this case, the value of λ is set to be small.

$$x^{(i+1)} = x^{(i)} - [H(x) + \lambda I]^{-1}\nabla F \tag{26}$$

Note that, in the Levenberg algorithm, if the value of the damping factor becomes large, the Hessian matrix is never used. To avoid this problem, the process of scaling can be done based on the curvature of equations so that where the gradient is small (low curvature), a small value for the damping parameter should be selected to provide a larger movement. The improved Levenberg algorithm, Eq. (27), connects the movement to curvature as [39–41]:

$$x^{(i+1)} = x^{(i)} - [H(x) + \lambda diag(H(x))]^{-1} \nabla F \tag{27}$$

The great advantage of this method over the Newton method is finding all potential roots of a nonlinear system without any requirement for a good initial guess.

3.7 Multi-step Method

This method is an efficient and rapid technique with wide applications in curve-fitting, approximating the derivative and integral of a function, and solving nonlinear equations. Generally speaking, in multi-step methods, the information of previous methods is retained to approximate the value of the next step. Regarding solving nonlinear equations, multi-step methods are of a more prominent place since it can lead to a significant decrease in computational time and effort emanating from computing Jacobian matrix per iteration.

The basic idea behind this method is using the Jacobian matrix obtained in previous steps for the successive steps. As a consequence, the time-consuming process of calculation of the Jacobian matrix and the lower-upper (LU) factorization in each iteration is simply removed by following a method without updating the Jacobian matrix at every single step, which is called the frozen Jacobian method. It can be stated that this technique can convert an existing one-point iterative method without memory, like the Newton method, to a multi-step method with memory.

Multi-step iterative techniques comprise two parts, namely base part and multi-step part. In the base part, the Jacobian matrix and its LU-factorization are calculated and kept for the next part. By reusing the information of the base-part, multi methods solve a system of linear equations. For example, the multi-step Newton method is shown as [42–44]:

$$\text{Base Newton Method} \begin{cases} y^0 \text{ initial guess} \\ F'\left(y^0\right)x^{(i+1)} = F\left(y^{(i)}\right) \\ y^{(i+1)} = y^{(i)} - x^{(i+1)} \end{cases}$$

$$\text{Multistep Newton Method} \begin{cases} \text{for } i = 1 : m-1 \\ F'\left(y^0\right)x^{(i+1)} = F\left(y^{(i)}\right) \\ y^{(i+1)} = y^{(i)} - x^{(i+1)} \\ \text{end} \\ y^{(0)} = y^{(m)} \end{cases} \tag{28}$$

In the multi-step Newton method, for $m \geq 1$ steps, the convergence order of the Newton method is increased to $m + 1$, which means that every additional step increases the convergence order by one. This algorithm may not seem cost-efficient, but, only one evaluation of the Jacobian matrix for one cycle can justify the wide-application of this method, in particular, for a large nonlinear system with sparse matrix. In contrast, for scalar nonlinear equations, this method is not recommended since it does not have the optimal convergence order [6, 45].

Some improvements concerning increasing the convergence of multi-step methods have been made. In these methods, it has become feasible to obtain higher convergence order than $m + 1$ by using the idea of multi-variable iterative methods. More details on these algorithms are given in studies [46, 47].

It has been demonstrated that this method, in spite of being efficient from a computational viewpoint, still requires the derivative of the function, which is not available for every problem. Hence, in such cases, derivative-free methods, which are represented below, can be adopted.

3.8 Picard Method

The main idea of the Picard or direct iteration method is splitting the nonlinear equations into two linear and nonlinear parts to reduce the size of computational work. In this case, the expensive operator of LU decomposition is only applied to the linear part. Picard method is a derivative-free technique of the decomposition technique for solving a system of nonlinear algebraic equations in engineering disciplines containing large scale problems. This method arranges the system of nonlinear equations as:

$$K(x)\mathrm{x} = \mathrm{F} \tag{29}$$

In which x is the vector column of unknowns, K and F are nonlinear coefficient matrix and the column vector of constants of nonlinear equations, respectively. The simple iterative method to solve this equation is as:

$$\{x\}^i = \left\{K(\{x\}^{(i-1)})\right\}^{-1}\{F\} \tag{30}$$

However, according to the above formulation, as all nonlinear equations are solved simultaneously, the coefficient matrix containing nonlinear part should be inversed at each iteration, which is computationally expensive. Therefore, to solve this issue, the decomposition technique can be effectively used by splitting the coefficient matrix into two parts of linear K_L and nonlinear $K_N(x)$ as:

$$K(x) = K_L + K_N(x) \tag{31}$$

Instead of inverting of $K(x)$ in Eq. (30), the alternative Eq. (32) can be followed. In this equation, the operation of inverse is only applied to the linear part. As a consequence, the computational time can considerably decrease.

$$\{x\}^{(i)} = (K_L)^{-1}\left[F - K_N(\{x\}^{(i-1)}).\{x\}^{(i-1)}\right] \tag{32}$$

To accelerate the convergence of this method, an auxiliary relaxation parameter γ, which varies between 0 and 1, can be added. By using this parameter, the new estimation can be substituted by a weighted average of the last two approximations as [2]:

$$\{\overline{x}\}^{(i)} = \gamma\{x\}^{(i-2)} + (1-\gamma)\{x\}^{(i-1)} \tag{33}$$

It should be pointed out that in this method, this method is of a slow speed; besides, it is likely that the decomposition process results in the non-singularity of the coefficient matrix.

3.9 Newton–Krylov Method

Inexact Newton methods are practical methods to approximate the resulting equation of Newton method, instead of exactly solving it. The algorithm of this method can be shown as:

$$\left\| J(x^{(i)})s^{(i)} + F(\mathrm{x}^{(i)}) \right\| \leq \eta^{(i)} \left\| F(x^{(i)}) \right\| \tag{34}$$

where $\eta^{(i)}$ is the forcing term of the i-th iteration varying between [0, 1) and $S^{(i)}$ is the inexact Newton step ($x^{(i+1)} = x^{(i)} + S^{(i)}$). In this method, initially, a suitable value for $\eta^{(i)}$ is selected; then, the linear equation is approximately solved for $S^{(i)}$. Some conventional methods to solve the resulting linear equations are classical splitting methods or the modern Krylov subspace methods. If widely-used Krylov subspace iteration methods are employed for solving the inexact Newton step, this method is called the Newton–Krylov subspace method [48].

The determination of forcing term in this method is of great importance that directly affects the accuracy and convergence of the method as the right-hand side expression is both a linear model of the system and the residual of the Newton method. Therefore, a suitable choice of $\eta^{(i)}$ results in a reduction in the norm of the model along with an increase in the accuracy of the model. For example, for $i = 0$ inexact Newton method converts to the quadratically convergent Newton method [48–50]. To explore some strategies for finding the best value for $\eta^{(i)}$, readers can be referred to [51, 52].

In comparison with the previous method, this method is deemed the most efficient tool for solving a system of nonlinear equations with a large sparse matrix. This

method only involves matrix–vector products instead of expensive operations like the inverse Jacobian matrix and LU decomposition thereof. Hence, this method decreases the computational cost regarding these operations. In various studies, it is shown that for small size problems, although this method cannot compete with other higher order Newton methods; for problems with large sparse matrix, this method considerably decreases the computational cost. Note that, as the Newton method, this method is locally convergent and require a globalization method. One effective method to enhance the global convergence of this method is using the backtracking technique—shortening the interval for an unsatisfactory step.

A downside of this method can be attributed to the calculation of the Jacobian matrix. Hence, for problems in which constructing the Jacobian matrix is difficult, the Jacobian free Newton–Krylov method can be replaced. By contrast with other methods mentioned in previous parts, this method does not calculate or store the Jacobian matrix, even for the initial approximation; instead, an approximation of Jacobian-vector product is constructed using the finite difference method. The step of this finite difference method can significantly affect the accuracy of the Jacobian-vector product [53, 54].

4 Discussion

In this part, an overall comparative explanation of the described methods, as well as some examples of the practical usage of these methods in different disciplines, are presented. As stated previously, the numerical methods for solving scalar equations can be described in two separate parts: one-point methods and multi-point methods. Accordingly, one-point methods are the most straightforward methods offering lower convergence together with ease of implementation. Among different one-point methods, Newton and Steffensen methods are two one-point methods, which are the basis of more advanced approaches, with the optimum convergence order. As these methods require a good initial guess, it is proposed to combine these methods with other globally convergent methods such as the bisection method. Chebyshev and Halley methods are other one-variable methods offering the cubic convergence order by employing the second derivative of a function; however, owing to some difficulties for computing a higher-order derivative, multi-point methods may be better alternatives increasing the convergence order. Using auxiliary variables in more steps, multi-point techniques present higher accuracy for the existing methods. It should be pointed out that every multi-point method with a higher convergence rate cannot deem an efficient method since need more function evaluations with the same efficiency as the previous methods.

Regarding the nonlinear system of equations, Gauss–Seidel, and fixed-point methods as derivative-free algorithms are just suitable for simple problems since they have slow convergence speed. The well-known Newton method is another method with quadratic rate of convergence. This method has some limitations, such

as needing a good initial guess, the time-consuming process of matrix–matrix evaluation, and the evaluation of the Jacobian matrix and its inverse. Nevertheless, this method has widely incorporated with a lot of mathematical software and computational simulation. One of the main problems in using Newton methods is the costly evaluation of the Jacobian matrix per iteration. To overcome such an issue, the estimation of this matrix, instead of directly computing, is suggested by Quasi-Newton methods for problems in which the derivatives of the function are not available, or the calculation of them is time-consuming. The quasi-Newton methods have also been used to solve Maxell's equations and Navier–Stokes equations [5].

Multi-step methods with frozen Jacobian are another technique that can be a remedy to the costly evaluation of the Jacobian matrix per iteration with increasing the convergence order. As scalar equations, the multi-point techniques can also enhance the convergence and effectiveness of these methods. Multi-step methods have also successfully been adopted for solving nonlinear Poisson, heat conduction, and wave equations [29, 42, 43]. Another way to solve multi-variable nonlinear equations is using global optimization methods to find all zeros of a nonlinear system, even with a poor initial guess. Despite being efficient, most of these methods require the computation of derivatives of the functions. These methods are also popular methods that have extensively incorporated in mathematical software. Picard method based on the decomposition technique is also defined as a derivative-free and straightforward method; it, though, may show instability for complex problems. As one of the commonly-used methods, the Newton-like methods can provide lower computational time for problems with a large sparse matrix. Of various techniques, Newton–Krylov method, as a Newton-like method, may be the best option in solving a complicated nonlinear system. The Jacobian-free Newton–Krylov method has also been employed to solve a wide range of problems, including radiation diffusion, Bratu, Navier–Stokes, and Maxwell's equations [49, 50]. Moreover, this method has been incorporated in many computational and simulation software. For the sake of comparison, the numerical characteristics of these numerical methods are also given in the Table 2. To the best of our knowledge, this chapter reviewed some general iterative techniques to solve nonlinear equations. However, there are other numerous developed techniques that are an improved version of the existing methods or a combination of them.

There are also some tangible cases displaying the application and importance of the discussed methods in real-world applications. As some notable examples, the fixed-point method has been extensively employed for solving the magnetic hysteresis field problems [56, 57]. The gradient-family methods have also been used in microwave imaging applications [55, 58, 59]. In the application of piezoelectric material as an energy harvester, the Newton and Picard method is also adopted to address the nonlinear behavior of structures with piezoelectric material actuators [60, 61]. In computational plasma physics, the Newton–Krylov methods are also employed to solve many diverse cases; these applications are all given in a study by Knoll and Keyes [49]. It should be pointed out that the aforementioned examples are only a limited number of cases demonstrating broad applications of iterative methods in energy applications. Obviously, given the potential capability of each technique,

Table 2 A comparative study of numerical methods for solving a nonlinear system of equations

Methods	Derivative free	Convergence speed	Globally convergent	LU factorization
Gauss–Seidel	Yes	Slow	No	0
Fixed point	Yes	Slow	No	0
Newton	No	Fast	No	Number of iterations
Quasi-Newton	No[a]	Slow	No	0
Steepest descent	No	Slow	Yes	Number of iterations
Leven-Marquardt	No	Fast	Yes	Number of iterations
Multi-Step	No[a]	Fast	Yes/No[b]	≥ 1
Picard	Yes	Slow	No	1
Newton–Krylov	No[a]	Fast	No	0

[a]The derivative is only needed for the initial guess
[b]It depends on the algorithm

these iterative methods can be adopted for solving other types of applications and equations.

5 Conclusion

Nonlinear analysis is an essential part of every discipline. In some applications, nonlinear solutions should be unavoidably employed to address some nonlinear phenomena, such as the dynamic behavior of wind turbines, the analysis of the damage within the structures, or the behavior of some novel materials. Keeping the leading role of the nonlinear solution methods in mind, this chapter provides an overview of some fundamental numerical methods and their attributes for solving the nonlinear discretized equations. Having dealt with some iterative numerical methods, in this part, this chapter is enclosed with concise results gained through this investigation, as well as some comments on future and ongoing directions in the development of iteration methods for solving nonlinear discretized problems.

To sum up, it may be concluded that for selecting a proper method for solving nonlinear equations, the equations themselves and the final accuracy always should be examined in selecting a method since employing some complex solution methods may seem unnecessary. On the other hand, there are some general factors specifying the best efficient technique for a problem. In dealing with scalar nonlinear equations, the number of call functions, convergence order, initial guess, and smoothness are determining factors in selecting a method. Regarding a system of nonlinear equations together with the above factors, other criteria should be considered, in particular, if

one is dealing with a complex system of nonlinear equations. For example, matrix–matrix operation, vector–matrix operation, the evaluation of Jacobian matrix and its inverse are other factors that are critical in choosing the best method.

Further development of more efficient and robust methods for nonlinear equations can be directed in the following way. Regarding scalar iterative methods, more efficient methods can be developed by establishing multi-point iterative methods with optimum convergence. Many ongoing researches are focusing on finding a suitable weight function or involved free parameter for the existing method that provides optimum convergence order.

With respect to iterative methods for solving the nonlinear system of equations, these solution methods have a prominent place in the foreseeable future due to the extensive application of them in advanced design and research as well as many intricacies regarding these methods. For example, concerning multi-step methods, there are still numerous continuing researches offering the optimal and higher convergence order. Moreover, developing some techniques to smooth or globalize the existing method and prevent them from diverging is the area of interest in this field. Another promising place can be related to Newton–Krylov methods. According to the merits and potential applications of these methods, a considerable number of researches during the very recent years have been focused on this topic; this area is still in progress. As the last remark, the combination of methods can assist researchers in enjoying the positive attributes of each method, as many techniques have been developed only by combining different existing methods.

References

1. Wang J (2018) A new iterative method for solving nonlinear equation. Dissertation, University of Saskatchewan
2. Reddy JN (2014) An introduction to nonlinear finite element analysis: with applications to heat transfer, fluid mechanics, and solid mechanics. OUP, Oxford
3. Wong EC (2014) Numerical methods for nonlinear systems of equations. Dissertation, Universiti Teknologi Malaysia
4. Mohammadi N, Asadi H, Aghdam MM (2019) An efficient solver for fully coupled solution of interaction between incompressible fluid flow and nanocomposite truncated conical shells. Comput Methods Appl Mech Eng 351:478–500
5. Zienkiewicz OC, Taylor RL, Zhu JZ (2005) The finite element method: its basis and fundamentals. Elsevier
6. Malik ZU (2015) Numerical iterative methods for nonlinear problems. Dissertation, Università degli Studi dell'Insubria
7. Maheshwari AK (2009) A fourth order iterative method for solving nonlinear equations. Appl Math Comput 211(2):383–391
8. Burden RL, Faires JD (2005) Numerical analysis, 8th ed. Thompson Brooks/Cole
9. Traub JF (1982) Iterative methods for the solution of equations, vol 312. American Mathematical Soc
10. Petković MS, Neta B, Petković LD et al (2014) Multipoint methods for solving nonlinear equations: a survey. Appl Math Comput 226:635–660
11. Soleymani F, Lotfi T, Bakhtiari P (2014) A multi-step class of iterative methods for nonlinear systems. Optim Lett 8(3):1001–1015

12. Yamamoto T (2000) Historical developments in convergence analysis for Newton's and Newton-like methods. J Comput Appl Math 124(1–2):1–23
13. Pooley D (2003) Numerical methods for nonlinear equations in option pricing. Dissertation, University of Waterloo
14. Janem GA (2015) Solving nonlinear boundary value problems using the homotopy analysis method. Dissertation, United Arab Emirates University
15. Gerald CF (2004) Applied numerical analysis. Pearson Education India
16. Alefeld G (1981) On the convergence of Halley's method. Am Math Mon 88(7):530–536
17. Argyros IK, Szidarovszky F (2018) The theory and applications of iteration methods. CRC Press
18. Cordero A, Torregrosa JR (2011) A class of Steffensen type methods with optimal order of convergence. Appl Math Comput 217(19):7653–7659
19. Kou J, Li Y (2007) The improvements of Chebyshev-Halley methods with fifth-order convergence. Appl Math Comput 188(1):143–147
20. Chun C (2007) Some second-derivative-free variants of Chebyshev-Halley methods. Appl Math Comput 191(2):410–414
21. Wang X, Liu L (2010) Modified Ostrowski's method with eighth-order convergence and high efficiency index. Appl Math Lett 23(5):549–554
22. Soleymani F, Sharifi M, Mousavi BS (2012) An improvement of Ostrowski's and King's techniques with optimal convergence order eight. J Optim Theory Appl 153(1):225–236
23. Thukral R, Petković MS (2010) A family of three-point methods of optimal order for solving nonlinear equations. J Comput Appl Math 233(9):2278–2284
24. Sharifi S, Salimi M, Siegmund S et al (2016) A new class of optimal four-point methods with convergence order 16 for solving nonlinear equations. Math Comput Simul 119:69–90
25. Zafar F, Cordero A, Torregrosa JR (2019) Stability analysis of a family of optimal fourth-order methods for multiple roots. Numer Algorithms 81(3):947–981
26. Artidiello S, Cordero A, Torregrosa JR et al (2015) Design of high-order iterative methods for nonlinear systems by using weight function procedure. Abstr Appl Anal. https://doi.org/10.1155/2015/289029
27. Soleymani F (2012) Optimized Steffensen-type methods with eighth-order convergence and high efficiency index. Int J Math Math Sci. https://doi.org/10.1155/2012/932420
28. Zheng Q, Li J, Huang F (2011) An optimal Steffensen-type family for solving nonlinear equations. Appl Math Comput 217(23):9592–9597
29. Qasim S, Ali Z, Ahmad F et al (2016) Solving systems of nonlinear equations when the nonlinearity is expensive. Comput Math Appl 71(7):1464–1478
30. Sadiku MN (2000) Numerical techniques in electromagnetics. CRC press
31. Ahmad F, Tohidi E, Carrasco JA (2016) A parameterized multi-step Newton method for solving systems of nonlinear equations. Numer Algorithms 71(3):631–653
32. Nili Ahmadabadi M, Ahmad F, Yuan G et al (2016) Solving systems of nonlinear equations using decomposition technique. J Linear Topol Algebra 5(03):187–198
33. Kelley CT (1995) Iterative methods for linear and nonlinear equations. Siam
34. Shademan A, Farahmand AM, Jägersand M (2010) Robust Jacobian estimation for uncalibrated visual servoing. In: 2010 IEEE International conference on robotics and automation, Anchorage, AK, USA, May 2010. IEEE, p 5564
35. Lapresté JT, Jurie F, Dhome M et al (2004) An efficient method to compute the inverse Jacobian matrix in visual servoing. In: 2004 IEEE International conference on robotics and automation, New Orleans, LA, USA, USA, April 2004. ICRA'04, vol 1. IEEE, p 727
36. Mansard N, Lopes M, Santos-Victor J et al (2006) Jacobian learning methods for tasks sequencing in visual servoing. In: 2006 IEEE/RSJ International conference on intelligent robots and systems, Beijing, China, October 2006. IEEE, p 4284
37. Tsoulos IG, Stavrakoudis A (2010) On locating all roots of systems of nonlinear equations inside bounded domain using global optimization methods. Nonlinear Anal Real World Appl 11(4):2465–2471

38. Johnson R, Zhang T (2013) Accelerating stochastic gradient descent using predictive variance reduction. In: Advances in neural information processing systems, vol 26. Neural information processing systems, Lake Tahoe, Nevada, December 2013, p 315
39. Ranganathan A (2004) The Levenberg-Marquardt algorithm. Tutoral LM Algorithm 11(1):101–110
40. Shukla PK (2010) Levenberg-Marquardt algorithms for nonlinear equations, multi-objective optimization, and complementarity problems. Dissertation, Dresden University of Technology
41. Moré JJ (1978) The Levenberg-Marquardt algorithm: implementation and theory. Numerical analysis. Springer, Heidelberg
42. Ilyas I, Ali Z, Ahmad F et al (2017) Multi-step frozen Jacobian iterative scheme for solving IVPs and BVPs based on higher order Fréchet derivatives. J Math 49(1):125–137
43. Kouser S, Rehman SU, Ahmad F et al (2018) Generalized newton multi-step iterative methods GMN p, m for solving system of nonlinear equations. Int J Comput Math 95(5):881–897
44. Shah FA, Noor MA (2015) Some numerical methods for solving nonlinear equations by using decomposition technique. Appl Math Comput 251:378–386
45. Kung HT, Traub JF (1974) Optimal order of one-point and multipoint iteration. J ACM (JACM) 21(4):643–651
46. Alzahrania EO, Al-Aidarousa ES, Younasa AM et al (2016) A higher order frozen Jacobian iterative method for solving Hamilton-Jacobi equations. Convergence 1000:1
47. Montazeri H, Soleymani F, Shateyi S et al (2012) On a new method for computing the numerical solution of systems of nonlinear equations. J Appl Math. https://doi.org/https://doi.org/10.1155/2012/751975
48. Shin BC, Darvishi MT, Kim CH (2010) A comparison of the Newton-Krylov method with high order Newton-like methods to solve nonlinear systems. Appl Math Comput 217(7):3190–3198
49. Knoll DA, Keyes DE (2004) Jacobian-free Newton-Krylov methods: a survey of approaches and applications. J Comput Phys 193(2):357–397
50. An HB, Wen J, Feng T (2011) On finite difference approximation of a matrix-vector product in the Jacobian-free Newton-Krylov method. J Comput Appl Math 236(6):1399–1409
51. An HB, Mo ZY, Liu XP (2007) A choice of forcing terms in inexact Newton method. J Comput Appl Math 200(1):47–60
52. Gomes-Ruggiero MA, Lopes VLR, Toledo-Benavides JV (2008) A globally convergent inexact Newton method with a new choice for the forcing term. Ann Oper Res 157(1):193–205
53. Eisenstat SC, Walker HF (1994) Globally convergent inexact Newton methods. SIAM J Optim 4(2):393–422
54. Chen Y, Shen C (2006) A Jacobian-free Newton-GMRES (m) method with adaptive preconditioner and its application for power flow calculations. IEEE Trans Power Syst 21(3):1096–1103
55. Mojabi P, LoVetri J (2009) Overview and classification of some regularization techniques for the Gauss-Newton inversion method applied to inverse scattering problems. IEEE Trans Antennas Propag 57(9):2658–2665
56. Dlala E, Arkkio A (2008) Analysis of the convergence of the fixed-point method used for solving nonlinear rotational magnetic field problems. IEEE Trans Magn 44(4):473–478
57. Dlala E, Belahcen A, Arkkio A (2007) Locally convergent fixed-point method for solving time-stepping nonlinear field problems. IEEE Trans Magn 43(11):3969–3975
58. Rubæk T, Meaney PM, Meincke P et al (2007) Nonlinear microwave imaging for breast-cancer screening using Gauss–Newton's method and the CGLS inversion algorithm. IEEE Trans Antennas Propag 55(8):2320–2331
59. De Zaeytijd J, Franchois A, Eyraud C et al (2007) Full-wave three-dimensional microwave imaging with a regularized Gauss-Newton method—theory and experiment. IEEE Trans Antennas Propag 55(11):3279–3292
60. Komijani M, Reddy JN, Eslami MR (2014) Nonlinear analysis of microstructure-dependent functionally graded piezoelectric material actuators. J Mech Phys Solids 63:214–227
61. Yang J, Kitipornchai S, Liew KM (2004) Non-linear analysis of the thermo-electro-mechanical behaviour of shear deformable FGM plates with piezoelectric actuators. Int J Numer Methods Eng 59(12):1605–1632