SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V10I11P114

Volume 10 Issue 11, 148-157, November 2023

Original Article

Analysis of Electrical Power Systems with Newton-Type Accelerated Numerical Methods

Ruben Villafuerte Diaz[1*], Jesus Medina Cervantes[1], Ruben Abiud Villafuerte Salcedo[2], Manuel Gonzalez Perez[3]

[1]*Department of Mechanical and Electrical Engineering, School of Engineering, Ixtaczoquitlán Campus, Veracruz, Mexico.*
[2]*Department of Electrical Engineering, Technological Institute of Mexico Campus Orizaba, Veracruz, Mexico.*
[3]*Department of Electronic Engineering, Technological University of Tecamachalco, Puebla, Mexico.*

[1]*Corresponding Author : rubenv46@yahoo.com.mx*

Received: 25 September 2023 Revised: 28 October 2023 Accepted: 22 November 2023 Published: 02 December 2023

Abstract *- The non-linear algebraic equations that model the load flow in an electrical network are non-linear and depend on the voltage and the complex power. In this paper, two three-step Newton-type numerical methods were reformulated to solve the non-linear equations modeling the load flow in an electrical network; the objective is to accelerate their convergence and reduce the execution time. Utilizing the Taylor series expansion of the original Newton-Raphson formula, an over-relaxation function was calculated and applied to two and three-step Newton-type numerical methods, and with the selection of an over-relaxation factor, the simulation time of the developed FORTRAN programs was reduced. With the modified techniques, the IEEE test systems of 30 and 118 nodes were solved, finding differences in the magnitude of the voltages of 0.32 percent and reducing, with a two-step method, the execution time from 1817 to 202 milliseconds for the 118-node system. For the test systems used, the low-order methods with over-relaxation present better characteristics from the point of view of execution time. From the results obtained, the load flow problem was successfully solved by applying the over-relaxation function to the reformulated Newton-type methods.*

Keywords *- Load flow, Non-linear equations, Numerical methods, Power system, Taylor series.*

1. Introduction

In the real world, the behavior of most natural phenomena behaves in a non-linear way. Many are considered linear for convenience, and their solution is obtained. Others must be linearized to find a solution, which is generally an approximation that is obtained iteratively. Several solution methods have been used to solve non-linear equations, and the Newton-Raphson method stands out among them due to its quadratic convergence and efficiency index [1].

Many ways have been developed to raise the order of convergence and the efficiency index of Newton's method [2-6]. Derivative-free and higher-order methods have been proposed for solving non-linear equations [7, 8]. Mohamed [9] offers a technique of eighteenth order of three steps and an efficiency of 1.435, which is greater than 1.4142, which is the efficiency of the Newton method; however, it requires the evaluation of eight functions and the improvement to the efficiency index is marginal; however, it is beneficial when solving a non-linear equation.

Likewise, recent and past methods have been developed to solve non-linear equations or systems of equations faster and with a higher order of convergence [10-13]. In this paper, two and three-step methods are accelerated, and the non-linear algebraic equations are solved to determine the load flow in electrical power systems. The matrix formulation of the Newton-Raphson method has been used in [14] to determine the elastic deformation gradient; again, as in the traditional methods for calculating load flows, the Jacobian matrix is used.

In [15], the authors propose an algorithm for calculating load flows in large-scale systems; however, they use the Jacobian matrix. The power flow equations can be expressed as non-linear equations: $f(x) = 0$, where x represents the state variables of equations. In this paper, the state variables are bus voltage magnitudes and angles when there is no limit violation. By establishing a function $f(x)$ at each node of an electrical network and applying Newton-type methods, the state vector x is iteratively calculated.

In this paper, only the admittance matrix of the system and one non-linear equation $f(x)$ per node, its first $f'(x)$ and its second derivative $f''(x)$, for the acceleration function are generated. An application of Taylor's series development is given, and an over-relaxation position is determined to reduce the execution time when solving non-linear equations for calculating load flow in electrical systems. For each node of a power system, the function *f(x)* is established, the state vector being the voltage in its phasor form. The function *f(x)* is a non-linear equation of second order, and with it, Newton-type methods were established to calculate the voltage at each node iteratively.

In solving non-linear equations, other authors have proposed methods applying different techniques. In [17], the authors offer a family of two- and three-step numerical methods of varying order where series are used to calculate the root of a non-linear equation. In [18], the author proposes different high-order numerical methods to calculate the *m* roots of a non-linear equation; likewise, in [19], the authors estimate the origins of a non-linear equation with third and fourth-order methods using decomposition techniques and Simpson's one-third rule.

In [20], the authors apply the modified Newton-Raphson method to calculate power flows in independent microgrids using the Jacobian matrix. There are many programs to calculate power flows; some require licenses, but others are free to use [21-23]. They use Gauss-Seidel or the classical versions of Newton-Raphson. Due to the convergence characteristics of Newton's method, it has been applied to the study of optimal flows [24, 25], and analysis tools are used, among which Matlab [26].

Considering Newton's classical formula, this work has developed methods for calculating power flows from a different perspective. This work aims to find numerical methods that solve non-linear equations in the shortest possible time. Here, we solve the equations that model the load flow in electrical power systems. For this purpose, an over-relaxation function is obtained by applying the Taylor series widely used in the convergence analysis of numerical methods. The main novelties of this paper can be summarized as follows.

- In the iterative calculation of voltages of the test systems, the Jacobian matrix is not required.
- It is unnecessary to use factoring techniques to arrive at the solution.
- The formulation is simple, and the results can be easily reproduced.
- The over-relaxation of Newton-type methods considerably reduces the execution time of the developed programs.
- The proposed objective was successfully achieved, and alternative methods for calculating power flows are available.

The paper is organized as follows: in section 2, with the Taylor series expansion, the order of Newton's method is determined and the over-relaxation function is obtained to accelerate the convergence of the Newton-type methods proposed in this paper; in section 3, two-step methods without overrelaxation are shown; in section 4, the numerical techniques without over-relaxation are shown; in section 5, the digital implementation is presented, section 6 contains the numerical methods with overrelaxation, section 7 includes results, and section 8 presents the differences found between the M2A and M4A methods and the discussion. At the end, the conclusions of the paper are given.

2. Over-Relaxation Function

The Newton-Raphson method (1) is widely used for solving absolute non-linear equations. Its convergence is ensured when the derivative of the function *f(x)* exists and when the initial value is close to the solution α.

$$x_{n+1} = x_n - \left[\frac{f(x_n)}{f'(x_n)}\right] \tag{1}$$

For the convergence analysis of (1), assume that α is a root of *f(x)*, so $f(\alpha)=0$ and $f'(\alpha)\neq 0$ and that the error $e_n = x_n - \alpha$, developing in Taylor series with Derive [29], the function *f(xn)*, about α we obtain Equation (2).

$$f(x_n) = f(\alpha) + e_n f'(\alpha) + \frac{1}{2} e_n^2 f''(\alpha) + \frac{1}{3!} e_n^3 f'''(\alpha) + O(e_n)^4 \tag{2}$$

Because: $f(\alpha)=0$, one has,

$$f(x_n) = f'(\alpha)\left(e_n + \frac{1}{2f'(\alpha)} e_n^2 f''(\alpha) + \frac{1}{3!f'(\alpha)} e_n^3 f'''(\alpha) + O(e_n)^4\right) \tag{3}$$

Defining:

$$C_k = \frac{f^k(\alpha)}{k!f'(\alpha)} \qquad \text{For; k = 2, 3, 4, 5,..}$$

Thus, the function *f(xn)* is defined by (4).

$$f(x_n) = f'(\alpha)(e_n + C_2 e_n^2 + C_3 e_n^3 + C_4 e_n^4 + C_5 e_n^5 + C_6 e_n^6 + O(e_n)^7) \tag{4}$$

Derived successively with respect to e_n, the first and second derivatives of $f(x_n)$ are obtained.

$$f'(x_n) = f'(\alpha)\left(1 + 2C_2 e_n + 3C_3 e_n^2 + 4C_4 e_n^3 + O(e_n)^4\right) \tag{5}$$

$$f''(x_n) = f'(\alpha)\left(2C_2 + 6C_3 e_n + 12C_4 e_n^2 + 20C_5 e_n^3 + O(e_n)^4\right) \tag{6}$$

Developing in the Taylor series with the program derive, the relation $\frac{f(x_n)}{f'(x_n)}$ in (7).

$$\frac{f(x_n)}{f'(x_n)} = e_n - C_2 e_n^2 + (2C_3 - 2C_2^2)e_n^3 + e_n^4(4C_2^3 - 7C_2C_3 + 3C_4) + (O)e_n^5 \quad (7)$$

From Equation (1) and Equation (7), the error is obtained.

$$e_{n+1} = C_2 e_n^2 - (O)e_n^3 \quad (8)$$

Where; $(O)e_n^3$ terms of order greater than three.

Newton's method, as shown in (8), has quadratic convergence and requires the evaluation of $f(x_n)$ and $f'(x_n)$. The method's efficiency is calculated with the order of convergence and the number of assessments.

$$EI = \rho^{1/\beta} \quad (9)$$

Where β is the number of evaluations and ρ the order of the method. The Newton-Raphson method has an efficiency index; EI = $2^{1/2}$ = 1.414213, and different papers focused on solving non-linear equations have been tasked with raising this index to improve its efficiency. In the solution of linear equations, acceleration factors are used to reduce the number of iterations [28]. The acceleration factors can be selected empirically or by calculating the minimum and maximum eigenvalue. This is usually time-consuming, and the empirical selection of the acceleration factor is chosen. In the Gauss-Seidel method, when applied in the analysis of load flows, an acceleration factor ranging between 1 and 2 is also usually used [29]. To accelerate the convergence of the Newton-Raphson method in this paper and give a practical application to the Taylor series development, Equation (1) is modified to the form shown in (10).

$$x_{n+1} = x_n - \left[\frac{f(x_n)}{f'(x_n)}\right]\Phi \quad (10)$$

The coefficient Φ is an over-relaxation factor [30] that is used to speed up the convergence of Newton's method. Generally, its value oscillates between 1 and 2. This paper considers it a relaxation function because it depends on the first and second derivatives of the function $f(x)$, as shown later.

From (7) and (9), we have (11)

$$\frac{f(x_n)\Phi}{f'(x_n)} = (e_n - C_2 e_n^2 + (2C_3 - 2C_2^2)e_n^3 + e_n^4(4C_2^3 - 7C_2C_3 + 3C_4) + (O)e_n^5)\Phi \quad (11)$$

The error e_{n+1}, is written by Equation (12).

$$e_{n+1} = e_n - (e_n - C_2 e_n^2 + (2C_3 - 2C_2^2)e_n^3 + (O)e_n^4) \quad (12)$$

Simplifying, it has (13).

$$e_{n+1} = e_n(1-\Phi) + \Phi C_2 e_n^2 + 2\Phi e_n^3(C_3 - C_2^2) + (O)e_n^4 \quad (13)$$

Where; $(O)e_n^5$ are terms of fifth order and higher.

It was found that Φ is a function that depends on the first and second derivatives of $f(xn)$, as shown by the error analysis e_{n+1} (13). Taking two terms of Equation (13) and considering that $e_{n+1} = 0$, $e_n = 1$, substituting C_2 and simplifying, we establish the over-relaxation function $\Phi(f(x_n), f''(x_n))$ (14).

$$\Phi\big(f'(x_n), f''(x_n)\big) = \frac{2f'(x_n)}{2f'(x_n) - f''(x_n)} = \frac{\alpha_1 f'(x_n)}{2f'(x_n) - f''(x_n)} \quad (14)$$

In Equation (14), Φ is an over-relaxation function that depends on the first and second derivatives of $f(x)$. Considering the possibility of having two acceleration functions and observing their behaviour, we obtain the inverse of (14).

$$\Phi(f'(x_n), f''(x_n)) = \frac{\alpha_1 f'(x_n) - f''(x_n)}{2f'(x_n)} \quad (15)$$

The over-relaxation function Φ depends on the factor α_1 and a number varying between 2 and 13, depending on the system.

3. Two-Step Methods

3.1. M1 Method

For now, consider the two-step frozen slope method (M1) represented by Equations (16) and (17).

$$y_n = x_n - \left(\frac{f(x_n)}{f`(x_n)}\right) \quad (16)$$

$$x_{n+1} = y_n - \left(\frac{f(y_n)}{f`(x_n)}\right) \quad (17)$$

The convergence analysis shows that the method is of third order and that it requires the evaluation of three functions; $f(x)$, $f'(x)$ and $f(y)$, Equation (18) shows their order and their efficiency index is EI=$3^{1/3}$=1.4422.

$$e_{n+1} = 2C_2^2 e_n^3 - (9C_2^3 - 7C_3C_2 - C_4)e_n^4 + (O)e_n^5 \quad (18)$$

3.2. M2 Method

The two-step method M2 represented by Equations (19) and (20) is of third order.

$$y_n = x_n - \left(\frac{f(x_n)}{f`(x_n)}\right) \quad (19)$$

$$x_{n+1} = x_n - \frac{f(y_n)}{f'(x_n) + f'(y_n)} \quad (20)$$

His convergence analysis demonstrates the order of the method (21).

$$e_{n+1} = e_n^3(4C_2^2 - C_3) + O(e_n^4) \quad (21)$$

Its efficiency index depends on the functions *f(x), f'(x)* and *f'(y),* which is $EI=3^{1/3}=1.4422$.

4. Three-Step Methods

Two three-step methods are used in this paper to solve the non-linear equations.

4.1. M3 Method

The three-step frozen slope method (M4) is represented by Equations (22-24).

$$y_n = x_n - \left(\frac{f(x_n)}{f'(x_n)}\right) \tag{22}$$

$$z_n = y_n - \left(\frac{f(y_n)}{f'(x_n)}\right) \tag{23}$$

$$x_{n+1} = z_n - \left(\frac{f(z_n)}{f'(x_n)}\right) \tag{24}$$

The convergence analysis of the M4 method shows that it is fourth order (25).

$$e_{n+1} = 14e_n^4(C_3 - C_2^2) + e_n^5(20C_2^4 - 22C_2^2C_3 + 2C_2C_4 - 6C_3^2 + C_5) + O(e_n^6) \tag{25}$$

The numerical method represented by Equations (22-24) is fourth order and depends on the evaluation of the functions *f(x), f'(x), f(y),* and *f(z)* with an efficiency index; $EI=4^{1/4}=1.414213$. It should be noted that this efficiency index is the same as that of Newton's method.

4.2. M4 Method

Mohamed's method [9], with three steps represented by Equations (26-28), is eighteenth order and has a high-efficiency index.

$$w_n = x_n - \frac{f(x_n)}{f'(x_n)} \tag{26}$$

$$y_n = w_n - \frac{2f(w_n)f'(w_n)}{2f'^2(w_n) - f(w_n)f''(w_n)} \tag{27}$$

$$x_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{f'(y_n)^2 f''(y_n)}{2(f'(y_n))^3} \tag{28}$$

Its convergence analysis shows that it is in the eighteenth order.

$$e_{n+1} = C_2^9(2C_2^2 - C_3)(C_2^2 - C_3)^3 e_n^{18} + O({e_n}^{19}) \tag{29}$$

The functions that are evaluated in this method when solving a non-linear equation are f(x), f'(x), f(w), f'(w), f''(w), f(y), f'(y), f''(y), which gives rise to your efficiency rating of; $EI=18^{1/8}=1.4351$.

5. Digital Implementation

Considering the system of five nodes shown in Figure 1 and applying the laws of electrical networks, the method of Equations (30) is obtained [28-32].

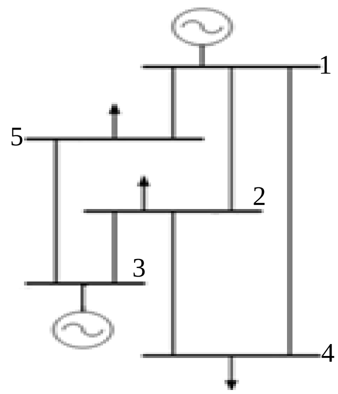

Fig. 1 Five node system [28]

$$\begin{aligned}
V_1Y_{11} + Y_{12}V_2 + Y_{13}V_3 + Y_{14}V_4 + Y_{15}V_5 &= \frac{S_1^*}{V_1^*} \\
Y_{21}V_1 + Y_{22}V_2 + Y_{23}V_3 + Y_{24}V_4 + Y_{25}V_5 &= \frac{S_2^*}{V_2^*} \\
Y_{31}V_1 + Y_{32}V_2 + Y_{33}V_3 + Y_{34}V_4 + Y_{35}V_5 &= \frac{S_3^*}{V_3^*} \\
Y_{41}V_1 + Y_{42}V_2 + Y_{43}V_3 + Y_{44}V_4 + Y_{45}V_5 &= \frac{S_4^*}{V_4^*} \\
Y_{51}V_1 + Y_{52}V_2 + Y_{53}V_3 + Y_{54}V_4 + Y_{55}V_5 &= \frac{S_5^*}{V_5^*}
\end{aligned} \tag{30}$$

Where:
V_i is the voltage in values per unit of node *i*,
$S_i^* = Sgi^* - Sci^*$ is the complex power in values per unit of node *i*,
Y_{ij} is the admittance per unit between node *i* and node *j*.

The methods to solve the Equations (30) imply laborious desk work when the study is carried out manually; the simplest ways, such as Gauss and Gauss-Seidel, practically require elementary operations with complex numbers; however, the versions of Newton's method require knowledge of partial derivatives and matrix factorization methods. The authors have proposed and adapted Newton-type methods; with them, it is intended to avoid the formation of the Jacobian matrix and apply factorization methods to find the solution required by all versions of the classical Newton method. Neglecting the Slack node, and for each node *i*, of Equations (30), we have (31).

$$f_i(V_1^k, V_2^k,,, V_n^k) = V_i^{*k}\sum_{j=1}^{N} Y_{ij}V_j^k - [(Sg_i - Sc_i)]^* \quad i = 1, N-1 \tag{31}$$

Where N is the number of nodes in the electrical network, Y_{ij} is the admittance between node i and node j, Vi is the voltage of node i, Sgi is the complex power generated, and Sci is the complex power demanded at the node i, respectively. Different methods have been applied to solve linear N equations; however, the execution times have been extended. With the proposed over-relaxed methods in this paper, execution times have been reduced considerably, as shown in the results obtained and presented below.

The methods considered in this paper apply Equation (31) with its first and second derivatives. The two- and three-step practices described above have been adapted to solve Equation (31) generated for each node, as can be seen, depending on N voltages. Calculating the voltages in each node of the power system with Equation (31) omits the matrix formulation, which is traditional in Newton's methods. The first derivative of the function (31) with respect to the voltage at node i is Equation (32).

$$f'_i(V_1^k, V_2^k,,, V_n^k) = \sum_{j=1}^{N} Y_{ij} V_j^k \tag{32}$$

The second derivative of the function (31) with respect to the voltage node i is Equation (33).

$$f''_i(V_1^k, V_2^k,,, V_n^k) = 2Y_{ii} \tag{33}$$

Functions (31), (32), and (33) are applied to each of the above methods to solve the *(N-1)* non-linear equations of an electrical power system and iteratively calculate the voltage at each node. Considering Equation (1) and substituting Equation (31), (32), and (33) in the corresponding term, Equation (34) is obtained.

$$v_{n+1} = v_n - \left[\frac{f(v_n)}{f'(v_n)}\right] \Phi$$

$$v_{n+1} = v_n - \left[\frac{V_i^{*k} \sum_{j=1}^{N} Y_{ij} V_j^k - [(Sg_i - Sc_i)]^*}{\sum_{j=1}^{N} Y_{ij} V_j^k}\right] \left[\frac{\alpha_1 \sum_{j=1}^{N} Y_{ij} V_j^k - 2Y_{ii}}{2Y_{ii}}\right] \tag{34}$$

The procedure applied to Equation (34) is used in each of the methods described in this paper.

6. Numerical Methods with Over-Relaxation

When Equation (14) was used to speed up a numerical method, convergence problems occurred in the solution or were found after many iterations. It is then chosen to work with the inverse, given by Equation (15). Where, α_1 is the scalar that is used as the over-relaxation factor. If Equation (15) is applied to Equation (1), we have (35).

$$x_{n+1} = x_n - \left[\frac{f(x_n)}{f'(x_n)}\right] \frac{\alpha_1 f'(x_n) - f''(x_n)}{2f'(x_n)} \tag{35}$$

Developing Equation (35) in series, we have the error Equation (36).

$$e_{n+1} = C_2 e_n - e_n^2(3C_2^2 - C_2 - 3C_3) + (O)e_n^3 \tag{36}$$

In (36), it is observed that the quadratic convergence of Newton's method is lost when affected by the over-relaxation function (14) or (15). Then, to directly affect the order of convergence of a process, an acceleration factor α_1 is randomly selected, and the value of the function Φ is calculated in each iteration to accelerate convergence.

This paper uses function (15) to accelerate the methods M1, M2, M3 and M4 and obtain over-relaxed methods. The M1A method, Equation (37) and (38).

$$y_n = x_n - \left(\frac{f(x_n)}{f`(x_n)}\right) \tag{37}$$

$$x_{n+1} = y_n - \left(\frac{f(y_n)}{f`(x_n)}\right) |\Phi(f'(x_n), f''(x_n)| \tag{38}$$

The M2A method, Equation (39) and (40).

$$y_n = x_n - \left(\frac{f(x_n)}{f`(x_n)}\right) \tag{39}$$

$$x_{n+1} = x_n - \frac{f(y_n)}{f'(x_n) + f'(y_n)} |\Phi(f'(x_n), f''(x_n)| \tag{40}$$

The M3A method, Equation (41), (42) and (43).

$$y_n = x_n - \left(\frac{f(x_n)}{f`(x_n)}\right) \tag{41}$$

$$z_n = y_n - \frac{f(y_n)}{f'(x_n)} |\Phi(f'(x_n), f''(x_n)| \tag{42}$$

$$x_{n+1} = z_n - \left(\frac{f(z_n)}{f`(x_n)}\right) \tag{43}$$

M4A Method, Equation (44), (45) and (46).

$$w_n = x_n - \frac{f(x_n)}{f'(x_n)} \tag{44}$$

$$y_n = w_n - \frac{2f(w_n)f'(w_n)}{2f'^2(w_n) - f(w_n)f''(w_n)} \tag{45}$$

$$x_{n+1} = y_n - \frac{f(y_n)}{f'(y_n)} - \frac{(f(y_n))^2 f''(y_n)}{2(f'(y_n))^3} |\Phi(f'(x_n), f''(x_n)| \tag{46}$$

7. Results

7.1. Numerical Methods with and without Over-Relaxation

The over-relaxation function aims to reduce the number of iterations and, mainly, the execution time when analyzing larger systems**.**

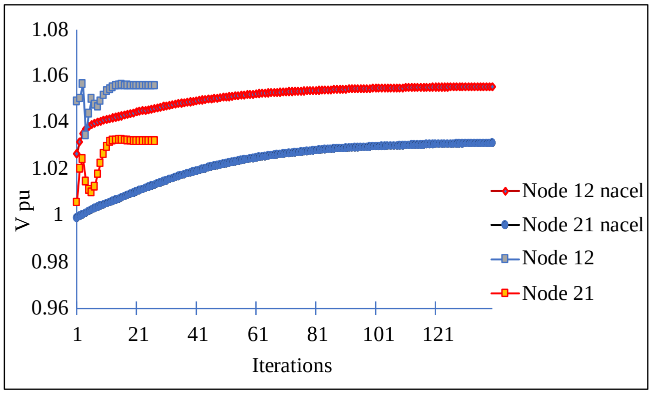

Fig. 2 Normal solution (node 12 nacel and node 21 nacel) not accelerated, and accelerated solution (node 12 and node 21)

The acceleration function Φ(f'(xn), f''(xn)) that is obtained in this paper is to accelerate the iterative process in the solution of the non-linear Equations (31). If Newton's method is used to solve the non-linear Equations (31) and is applied to the IEEE test system of 30 nodes [34], Figure 2 presents the behaviour of nodes 12 and 21 for the over-relaxed case (Node 12 and Node 21) and the non- over-relaxed case (Node 12 nacel and Node 21 nacel)

By speeding up the method, we see that case (Node 12 and Node 21) is a faster approximation to the solution. The solution of the equations resolved in this paper is non-linear. Because the analysis is carried out steadily, the variables are phasor or complex numbers, the network's admittance matrix, and the complex power demanded and generated in each node.

The methods M1, M2, M3, and M4 are applied to solve the (N-1) non-linear algebraic equations that model the load flow in electric power systems. The IEEE 118-node network [30] is used as the test system. For each method, a program was developed in the FORTRAN language. The Force-2 compiler was used, and with laptops with 8 and 16 GB of RAM memory, a tolerance of 0.001 was considered. In all cases, the criterion for ending the iterative process is calculated with Equation (47).

$$\varepsilon = \sum_{i=1}^{N-1} \left(\frac{f(i)}{df(i)}\right)^2 \tag{47}$$

Data on transmission lines generated complex power *Sgi* and demanded complex power Sci at each node. The number of transformers can be found in the cited IEEE reference. Table 1 summarizes the number of iterations and the execution time in milliseconds of the methods M1, M2, M3 and M4, when solving the 118-node system [34]. Table 2 summarizes the number of iterations and the execution time in milliseconds of each of the methods M1A, M2A, M3A and M4A, accelerating the solution; the over-relaxation factor α1 used is shown in the fourth column.

Table 1. Iterations and execution time of methods without over-relaxation

Method	Iterations	CPU (ms.)
M4	239	1126
M1	358	1475
M2	475	1817
M3	239	1466

Table 2. Iterations and execution time of methods with acceleration

Method	Iterations	CPU (ms.)	α_1	E2
M4A	43	349.4	7.5	71.875
M1A	45	277.6	7.0	50.0
M2A	31	196.56	12.5	53.125
M3A	19	177.84	7.5	46.875
M5	5	845.5	-----	134.375

The decoupled Newton method M5 is implemented for comparative purposes; the number of iterations and the execution time are also found in Table 2. In Table 2, column E2 corresponds to the simulation with 16 GB of RAM equipment. The other values were obtained with equipment that has 8 GB of RAM. The M1A and M3A methods, as shown in Table 2, require the least number of iterations and the shortest average execution time. The execution time may vary depending on the processor's activity when the program is executed. The average time was calculated considering five simulations with each method. Figure 3 shows the behaviour of nodes 39 and 41 when solving the 118-node system with program M3.

Figure 4 shows the behaviour of nodes 39 and 41 at each step when solving the 118-node system with the M3A program. Figure 4 shows the effect of the acceleration function; the observed changes may be related to the derivative's variation in that node in each step, thus allowing the searched solution to be reached in less time.

A widely used method for calculating load flows is the decoupled version of Newton. This method requires the establishment of the Jacobian matrix to calculate the magnitudes and angles of the voltages. Simulating with the decoupled Newton method, the number of iterations (5 iterations) and the behaviour of nodes 39 and 41 is shown in Figure 5.

The over-relaxation function changes at each iteration, Figure 6 shows its behaviour at each node for iterations 1, 5, 15, 20, 25 and 31 when the M2A method is applied and an over-relaxation factor α1 of 12.5 is used.

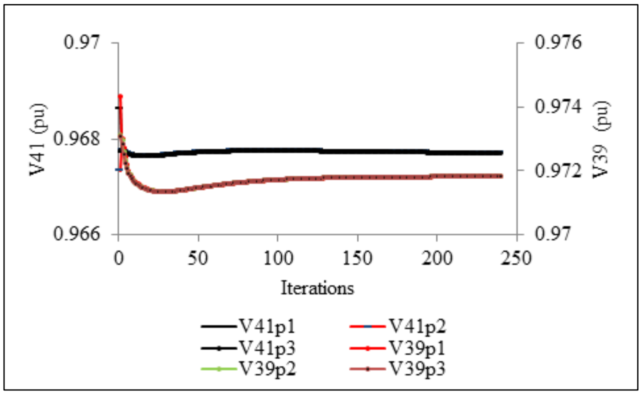

Fig. 3 Behavior of nodes 39 and 41 without over-relaxation

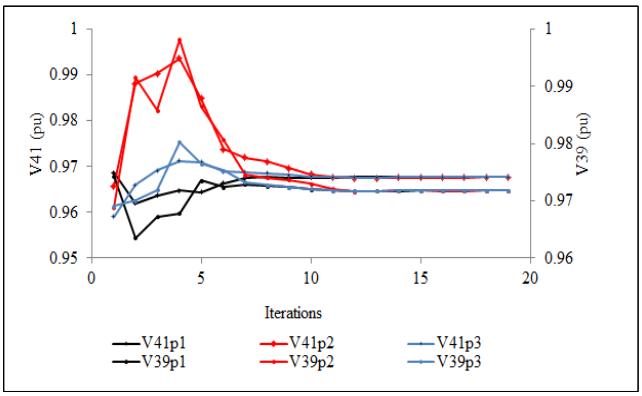

Fig. 4 Behavior of nodes 39 and 41 accelerating the second step

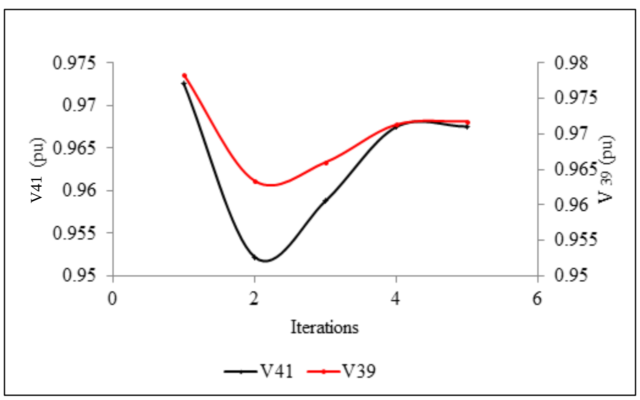

Fig. 5 Behavior of nodes 39 and 41 in the iterative cycle

The M3A method takes 19 iterations to solve the 118-node system; the behaviour of the absolute value of the over-relaxation function at each node, when the over-relaxation factor α_1 is 7.5 in iterations 1, 5, 10, 15 and 19, is shown in Figure 7. The over-relaxation function given by Equation (15) has been used instead of the one represented by Equation (14) because there is a notable difference in the simulation times. Figure 8 shows these functions' behaviour when the value of α_1 is selected when solving the equations of a 30-node IEEE test system with the M2A method. Values of $\alpha_1 = 5$, Equation (14) and $\alpha_1 = 11$, Equation (15). The graphs show the behavior of node 8.

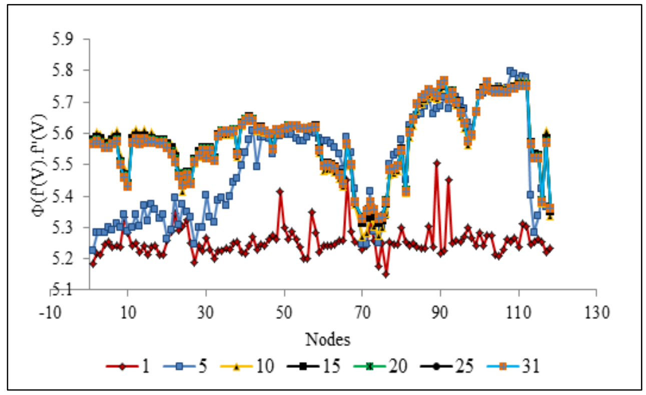

Fig. 6 Behavior of the absolute value of the acceleration function with the M2A method

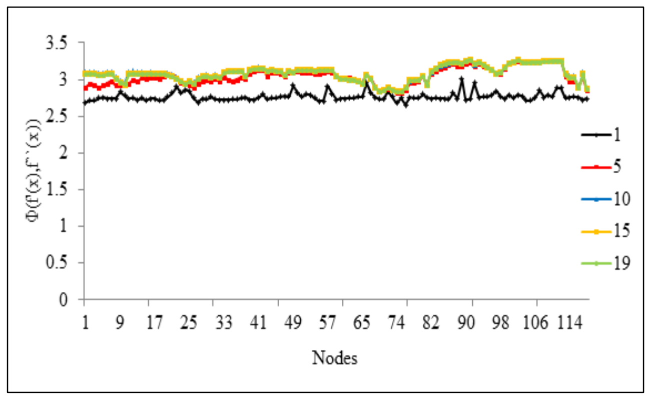

Fig. 7 Behavior of the absolute value of the acceleration function with the M3A method

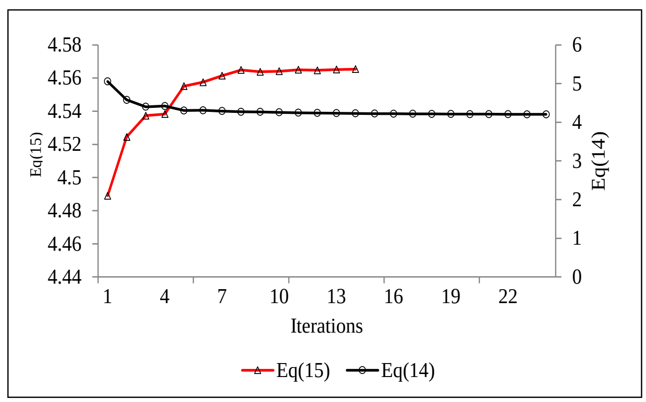

Fig. 8 Functions (14) and (15) in the 30-node system

Figure 9 shows the behaviour of Equations (14) and (15) when the 118-node IEEE test system equations are solved with the M2A method. Values of $\alpha_1 = 7$, eq. (14) and $\alpha_1 = 11$, eq. (15). The graphs show the behavior of node 95. Table 3 shows the average execution time and the over-relaxation factor α_1 when using the over-relaxation functions in the M3A method. Figure 10 shows the magnitude of voltages obtained with the M2A and M3A methods when applied to the solution of the 30-node system, compared with the values reported [30].

Table 3. Over-relaxation factors and execution time

System	Eq	α_1	CPU time (ms)
30-nodes	(14)	5	46.8
	(15)	11	31.2
118-node	(14)	7	343
	(15)	11	196.96

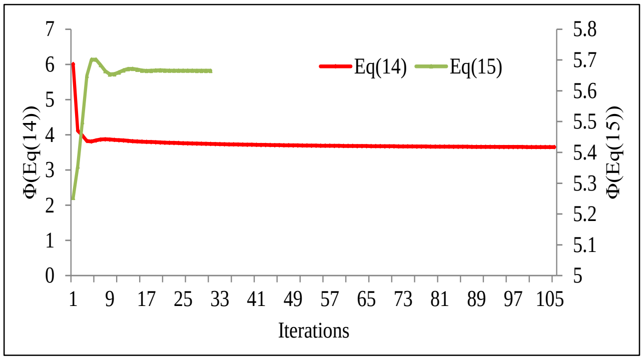

Fig. 9 Functions (14) and (15) in the 118-node system

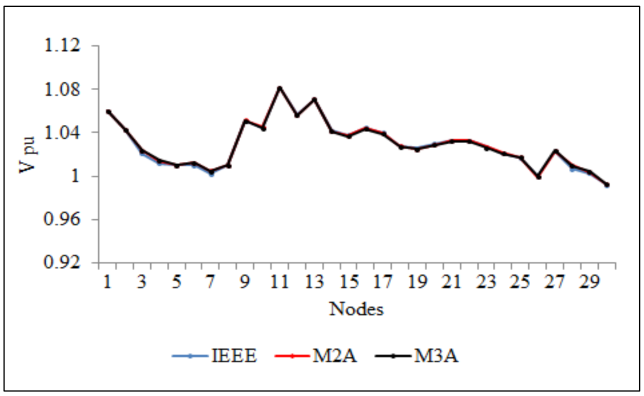

Fig. 10 Comparison between the magnitude of voltages

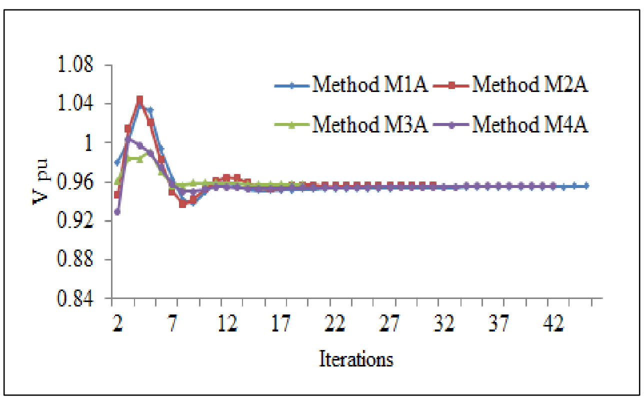

Fig. 11 Table 2 data expressed graphically

The value of the voltage angle, as well as its magnitude, are essential in the calculation of load flows. Figure 11 summarizes the behaviour of numerical methods with over-relaxation corresponding to the data shown in Table 2. Figure 11 shows that the iterative process could practically end at iteration 17; however, the programs end until the previously established convergence criterion has been met.

8. Differences between M2A and M4A Methods and Discussion

As reported by its author, the numerical method represented by method M4 has been applied to the solution of non-linear fundamental algebraic Equations (10). Due to its order of convergence and efficiency, it was adapted in this paper to solve complex non-linear equations in electrical engineering represented by Equation (31).

The numerical method described by method M2A, which has a lower order of convergence and its over-relaxation version, modified with Equation (15), has been used to solve the 118-node system. Table 4 shows the results obtained by applying the method represented by method M4 and its over-relaxed version M4A in the third step, the M2 method and its over-relaxed version M2A.

Table 4. Comparison between methods M4, M4A, M2 and M2A

Method	α_1	Iterations	CPU (ms)
M4		239	1388.4
M4A	7.5	40	327.3
M2		475	1817.0
M2A	12.5	31	202.8

Table 4 shows a slower convergence in the M4A method in relation to the M2A method, which requires 61 percent less execution time to reach the solution. Newton-type numerical methods were adapted to solve each of the non-linear equations that model the load flow of an electrical power system. Equation (31) was generated at each of the nodes of the test systems, except for the compensator node; they were solved iteratively with accelerated or over-relaxed methods, as you want to call them. The formulation used does not require, as in the traditional methods, the generation of the Jacobian matrix; therefore, applying factorization methods to reach the solution was unnecessary.

Because of the formulation used, the method may have numerical weaknesses; however, from the authors' point of view, the formulation is coded and very easy to reproduce. The results obtained show that the two-step method requires a shorter execution time, which makes it suitable for treating larger systems than those used in this paper. Another essential fact found in this paper's development is that the over-relaxation factor depends on the size of the system and could be higher for systems with more nodes. Figure 5 and Figure 6 show that the over-relaxation function is dynamic, accelerating the convergence of the methods used in this paper.

9. Conclusion

Two-step and three-step methods have been used to solve the (N-1) non-linear equations of a 30-node and 118-node electrical power system. An α_1 number that varies between 2 and 12.5 was used for the M2A method; their over-relaxation factor is between 2 and 7.5 for the other methods. The characteristic of the over-relaxation function is variable, as shown in Figure 3, Figure 5, and Figure 6.

This allows an acceleration per node that translates into a shorter execution time. Table 2 shows that the accelerated methods M1A, M2A and M3A are more convenient, and the execution time is significantly reduced when using simulation equipment with better characteristics (more RAM and faster processor). The same Table 2 shows that the over-relaxed methods require 2 to 3 times less execution time compared to the decoupled version of Newton's method. The use of Equation (15) in solving the non-linear equations of the test methods reduces the execution time, as shown in Table 2. It is essential to mention that by affecting any step of the process by the acceleration function, the convergence order is lost; however, the execution time is considerably reduced.

The order of convergence of a numerical method is an essential theoretical reference when solving absolute algebraic equations; concerning the solution of complex non-linear equations, the concept, at least for the systems studied in this paper, does not seem to have a significant effect. Table 4 shows this: the M2A method is of low order; however, speeding it up achieves the objective sought in this paper. From the results obtained, the over-relaxation can be applied to multi-step methods and analyze larger power systems with shorter run times and three-phase systems. Likewise, the methodology can be used multi-steply, reducing the execution time.

References

[1] Alicia Cordero et al., "Increasing the Convergence Order of an Iterative Method for Nonlinear Systems," *Applied Mathematics Letters*, vol. 25, no. 12, pp. 2369-2374, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[2] YoonMee Ham, and Changbum Chun, "A Fifth-Order Iterative Method for Solving Nonlinear Equations," *Applied Mathematics and Computation*, vol. 194, no. 1, pp. 287–290, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[3] Hameer Akhtar Abro, and Muhammad Mujtaba Shaikh, "A New Time-Efficient and Convergent Nonlinear Solver," *Applied Mathematics and Computation*, vol. 355, pp. 516–536, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[4] Alicia Cordero, Juan R. Torregrosa, and Maria P. Vassileva, *Design, Analysis, and Applications of Iterative Methods for Solving Nonlinear Systems*, In Nonlinear Systems-Design, Analysis, Estimation and Control, IntechOpen, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[5] M.A. Hafiz, and Mohamed S.M. Bahgat, "An Efficient Two-Step Iterative Method for Solving System of Nonlinear Equations," *Journal of Mathematics Research*, vol. 4, no. 4, pp. 28-34, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[6] Changbum Chun, "Some Fourth-Order Iterative Methods for Solving Nonlinear Equations," *Applied Mathematics and Computation*, vol. 195, no. 2, pp. 454–459, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[7] Janak Raj Sharma, and Himani Arora, "Some Novel Optimal Eighth Order Derivative-Free Root Solvers and their Basins of Attraction," *Applied Mathematics and Computation*, vol. 284, pp. 149-161, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[8] Farahnaz Soleimani, Fazlollah Soleymani, and Stanford Shateyi, "Some Iterative Methods Free from Derivatives and their Basins of Attraction for Nonlinear Equations," *Deterministic Discrete Dynamical Systems: Advances in Regular and Chaotic Behavior with Applications*, vol. 2013, pp. 1-10, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[9] Mohamed S.M. Bahgat, and M.A. Hafiz, "Three-Step Iterative Method with Eighteenth Order Convergence for Solving Nonlinear Equations," *International Journal of Pure and Applied Mathematics*, vol. 93, no. 1, pp. 85-94, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[10] Rajni Sharma, and Ashu Bahl, "An Optimal Fourth Order Iterative Method for Solving Nonlinear Equations and Its Dynamics," *Journal of Complex Analysis*, vol. 2015, pp. 1-9, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[11] Young Hee Geum, Young Ik Kim, and Beny Neta, "Constructing a Family of Optimal Eighth-Order Modified Newton-Type Multiple-Zero Finders along with the Dynamics behind their Purely Imaginary Extraneous Fixed Points," *Journal of Computational and Applied Mathematics*, vol. 333, pp. 131–156, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[12] Sania Qureshi, Higinio Ramos, and Abdul Karim Soomro, "A New Nonlinear Ninth-Order Root-Finding Method with Error Analysis and Basins of Attraction," *Mathematics*, vol. 9, no. 16, pp. 1-18, 1996. [CrossRef] [Google Scholar] [Publisher Link]

[13] Higinio Ramos, and M.T.T. Monteiro, "A New Approach Based on the Newton's Method to Solve Systems of Nonlinear Equations," *Journal of Computational and Applied Mathematics*, vol. 318, pp. 3–13, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[14] Miguel Vieira de Carvalho, Rui Pedro Cardoso Coelho, and Francisco M. Andrade Pires, "On the Computational Treatment of Fully Coupled Crystal Plasticity Slip and Martensitic Transformation Constitutive Models at Finite Strains," *International Journal for Numerical Methods in Engineering*, vol. 123, no. 21, pp. 5155–5200, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[15] Kunjie Tang et al., “A Robust and Efficient Two-Stage Algorithm for Power Flow Calculation of Large-Scale Systems,” *IEEE Transactions on Power Systems*, vol. 34, no. 6, pp. 5012-5022, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[16] Kadir Abaci, and Volkan Yamaçli, “Hybrid Artificial Neural Network by Using Differential Search Algorithm for Solving Power Flow Problem,” *Advances in Electrical and Computer Engineering*, vol. 19, no. 4, pp. 57-64, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[17] Faisal Ali et al., “New Family of Iterative Methods for Solving Nonlinear Models,” *Discrete Dynamics in Nature and Society*, vol. 2018, pp. 1-12, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[18] Tekle Gemechu, “Some Multiple and Simple Real Root Finding Methods,” *Mathematical Theory and Modeling*, vol. 7, no. 10, pp. 1-12, 2017. [Google Scholar] [Publisher Link]

[19] Awais Gul Khan et al., “Some New Numerical Schemes for Finding the Solutions to Nonlinear Equations,” *AIMS Mathematics*, vol. 7, no. 10, pp. 18616–18631, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] Ashiq Hussain Lone, and Neeraj Gupta, “A Novel Modified Decoupled Newton-Raphson Load Flow with Distributed Slack Bus for Islanded Microgrids Considering Frequency Variations,” *Electric Components and Systems*, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[21] ETAP Software. [Online]. Available: https://etap.com/es

[22] DIgSILENT, Power System Solutions. [Online]. Available: http://www.digsilent.de/en/

[23] PowerWorld, Software for Students. [Online]. Available: https://www.powerworld.com/solutions/students

[24] Dhanushka Kularatne, Hadi Hajieghrary, and M. Ani Hsieh, “Optimal Path Planning in Time-Varying Flows with Forecasting Uncertainties,” *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4857-4864, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[25] Adrián Esteban-Pérez, and Juan M. Morales, “Distributionally Robust Optimal Power Flow with Contextual Information,” *European Journal of Operational Research*, vol. 306, no. 3, pp. 1047-1058, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[26] Mathworks, Seleccione un país/idioma, Matlab. [Online]. Available: https://la.mathworks.com/products/matlab.html

[27] Software Derive for Windows V 6.1, Microsoft®Update Catalog. [Online]. Available: https://www.catalog.update.microsoft.com/Search.aspx?q=6.1.7600.16385

[28] William D. Stevenson, *Sistemas Eléctricos de Potencia*, McGraw-Hill Interamericana, 1970. [Publisher Link]

[29] John J. Grainger, and William D. Stevenson, *Análisis de Sistemas de Potencia*, McGraw-Hill Interamericana, 1995. [Publisher Link]

[30] Hans Petter Langtangen, “*Solving Nonlinear ODE and PDE Problems*,” Center for Biomedical Computing, Simula Research Laboratory, Department of Informatics, University of Oslo, 2016. [Google Scholar] [Publisher Link]

[31] William. F. Tinney, and Clifford E. Hart, “Power Flow Solution by Newton’s Method,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-86, no. 11, pp. 1449-1460, 1967. [CrossRef] [Google Scholar] [Publisher Link]

[32] B. Stott, and O. Alsac, “Fast Decoupled Load Flow,” *IEEE Transactions on Power Apparatus and Systems*, vol. PAS-93, no. 3, pp. 859-869, 1974. [CrossRef] [Google Scholar] [Publisher Link]

[33] Timothy A. Davis, Sivasankaran Rajamanickam, and Wissam M. Sid-Lakhdar, “A Survey of Direct Methods for Sparse Linear Systems,” *Acta Numerica*, vol. 25, pp. 383-566, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[34] Power Systems Test Case Archive. [Online]. Available: http://www.ee.washington.edu/research/pstca/