SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V11I11P118

Volume 11 Issue 11, 172-181, November 2024

Short Communication

Newton-Type Iterative Solvers for Power Flows

Ruben Villafuerte Diaz[1*], Victorino Juarez Rivera[1], Ruben Abiud Villafuerte Salcedo[2], Jesus Medina Cervantes[1]

[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.*

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

Received: 10 September 2024 Revised: 11 October 2024 Accepted: 09 November 2024 Published: 30 November 2024

Abstract *- Starting from a Newton-type method of third order and two steps, a five-step and a seven-step method is adapted to solve nonlinear equations and the power function f(V1,V2,...V3) is applied to calculate the voltage at each node of an electrical power system and the load flow in the transmission lines. The procedure involves iteratively calculating the nonlinear function and its first and second derivatives. At each node i of the system under study, the original formula of Newton's method, or an adaptation of it, is applied, and the approximations of the voltage at each node are calculated. To reduce the execution time, an acceleration function that depends on the first and second derivatives of the power function was used. The power function is generated at each node and iteratively solved for each node separately, thus avoiding the formation of the Jacobian matrix. Test systems from 9 to 118 nodes were analyzed; the maximum errors found were 0.333% for the 30-node system in the Voltage magnitude, 5.62% in the angle one node of the same system, For the 118-node system, the maximum error in the voltage magnitude was 2.254%, in one node of the system. The applied methods reduce the execution time from 1048.875 to 78.125 milliseconds with the accelerated seven-step method and with a tolerance of 1.0e-04. We foresee the possibility of applying methods with more steps and higher acceleration factors from the results obtained.*

Keywords *- Load flows, Nonlinear equations, Numerical Methods, Newton-type methods, Power systems.*

1. Introduction

Electric power is fundamental to performing many tasks in the modern world. To meet the energy demands of industrial and residential users, power plants must operate continuously, monitoring important variables at all times, including, among many others, the magnitude and angle of voltage at each node, the frequency, and the flow of real and reactive power on transmission lines. Different studies in an electrical system are carried out in steady state, dynamic state or transient state. Among the steady-state studies is the calculation of power flows, which is fundamental to determining the operating conditions of the network. With the study of load or power flows, the voltage of each node under different load conditions is known, the power flow in transmission lines and its corresponding losses are calculated, and it is also possible to investigate the operating conditions when changes are made in the network configuration and plan it properly. A power system consists

of several nodes or buses where transmission lines, generators and loads are connected. Each bus has two power flow equations associated with it, which makes the analysis nonlinear. Many researchers have studied power flow analysis and digital programs have been obtained to perform the study of different electrical power systems. In [1], Newton's method is used for calculating power flows where the maximum number of iterations for any system is five. They also use ordering techniques of the generated matrices. In [2], the authors, given the characteristics of the system to be solved, use ordering and factorization techniques of the Jacobian matrix to efficiently calculate the system voltages and the power flow in transmission lines. In [3], the authors propose a decoupled version of Newton's method, taking advantage of the weak dependence of the real power on the voltage magnitude and of the reactive power on the voltage angle, with this method and independently of the system, the number of iterations reported was three. Matrix factorization methods are indispensable when solving large systems; in [4], they use solvers and preconditioners included in Matlab in the LU factorization to reduce in flpos, the solution of the test systems used. The LU factorization of the Jacobian matrix becomes important when analyzing large systems. In [5-8], they use the inexact Newton-Krylov method to reduce the factorization runtime. In [9], the authors use a hybrid method consisting of Newton's and the downward gradient methods to compute power flows in IEEE test systems. The growth of power systems and distribution systems has generated great challenges. The size of the admittance matrix for the calculation of flows has also grown considerably, which implies the use of appropriate methods and techniques for the storage and solution of the equations generated. The ordering and storage of large and sparse matrices have been studied by

different authors among them [2, 10]. Newton's method has been applied in different versions to the solution of power flows [11, 12]. In its decoupled and fast decoupled versions, the polar form has been more widely used than the rectangular form. The different versions of Newton's methods are somewhat complex. From the use of partial derivatives to find the Jacobian matrix. The solution of the generated linear system requires the ordering of the equations that model the system to avoid the generation of non-zero elements in the factorization of the matrix. All this requires computer equipment with special features to analyze power systems and current distribution systems. The methods proposed in this work attempt to simplify a complex study, starting from the basic equations used in the analysis of electrical networks and establishing a nonlinear equation at each node of the system. The proposed methods solve the power flow problem with elementary operations and represent a further alternative to its solution.

The methods proposed in this paper make use of equation (15) of [13], and with them, the voltages of an electrical power system are calculated in a faster way. The methods applied here use the original Newton-Raphson formula and methods known as Newton-type [14, 15]. There are several methods for solving nonlinear equations; some can be found in [16-19]. The objective of this work was the development of simplified numerical methods here, the formation of the Jacobian matrix and the application of matrix factorization schemes are dispensed with; instead, the linearized equations are solved with elementary operations for the calculation of the network voltages and their consequence, the power flows in transmission lines. Based on the existence of methods for the solution of nonlinear equations that are often applicable to a single equation, in this paper, a five-step method and a seven-step method were adapted to solve the complex nonlinear equations of electrical power systems. The M2A method [13] with the acceleration function (15) is the reference for the proposed methods. The Newton-type methods were applied to 9, 30, 39, and 118 node systems. The main novelties of this paper can be summarized as follows.

- A two-step Newton-type numerical method with acceleration in each step is adapted for the analysis of interconnected power systems
- A five-step Newton-type numerical method with acceleration in steps one, three and five is adapted to analyse interconnected electrical systems.
- A seven-step Newton-type numerical method with acceleration in steps one, three, five and seven is adapted to analyse interconnected electrical systems.
- The characteristic of the three methods is that the Jacobian matrix is not formed to solve the system of equations.
- With proper selection of the acceleration factors, the run time is reduced by up to 1300%.

2. Equations for the Calculation of Power Flows, A Review

In the study of power flows, nonlinear equations are established to calculate the magnitude of the voltage Vi and its angle δ_i at each node. The active power P_i and reactive power Q_i, net at each node, and the admittances $Y_{ij}=G_{ij}+jB_{ij}$ are important. At node *i* of an N-node power system, with nodal analysis, Equation (1) [20, 21] is established.

$$Y_{i1}V_1V_i^* + Y_{i2}V_2V_i^* + .. + Y_{ii}V_iV_i^* + \cdots + Y_{iN}V_NV_i^* = P_i - jQ_i$$
For; i =1, N, i ≠ Slack (1)

In a steady state, the voltages and powers are phasors, and the admittances Y_{ij} of the transmission lines are complex. From Equation (1), representing in polar form the voltages V_i and the admittances Y_{ij}, Equation (2) [21-23] is obtained for the real power P_i [21-23].

$$P_i = |Y_{ii}||V_i|^2 cos(\theta_{ii}) + \sum_{j=1, j\neq i}^{N} \left|Y_{ij}V_jV_i^*\right| \cos(\theta_{ij} + \delta_j - \delta_i)$$
For; i =1, N, i ≠ Slack (2)

For reactive power Qi, Equation (3).

$$Q_i = -|Y_{ii}||V_i|^2 sin(\theta_{ii}) - \sum_{j=1, j\neq i}^{N} \left|Y_{ij}V_jV_i^*\right| \sin(\theta_{ij} + \delta_j - \delta_i)$$
For; i=1, N, i ≠ Slack (3)

From Equation (1), we define functions (4) and (5).

$$f_{di} = |Y_{ii}||V_i|^2 cos(\theta_{ii}) + \sum_{j=1, j\neq i}^{N} \left|Y_{ij}V_jV_i^*\right| \cos(\theta_{ij} + \delta_j - \delta_i) - P_{inet} \quad (4)$$

$$f_{vi} = |Y_{ii}||V_i|^2 sin(\theta_{ii}) + \sum_{j=1, j\neq i}^{N} \left|Y_{ij}V_jV_i^*\right| \sin(\theta_{ij} + \delta_j - \delta_i) + Q_{inet} \quad (5)$$

Equations (4) and (5), except Slack node.

Where P_{inet} and Q_{inet} are the real and net reactive power of node *i*, respectively. Also, a complex function *f(Vi)* which depends on the voltages V_i and the net complex power at each node *i.*

$$f(V_1, , V_i, , V_N)_i = \sum_{j=1}^{N} Y_{ij}V_jV_i^* - (P_i - jQ_i)$$
For; i=1, N, i ≠ Slack (6)

From 1, expressing in its rectangular form the voltage at each node *i*, $V_i=e_i+jf_i$ and the admittance $Y_{ij}=G_{ij}+jB_{ij}$, it is decomposed to obtain the equations for real power, Equation (7) and for reactive power, Equation (8) [21-23].

$$P_i = G_{ii}(e_i^2 + f_i^2) + \sum_{j=1, j\neq i}^{N} G_{ij}(e_j e_i + f_j f_i) - B_{ij}(f_j e_i - e_j f_i)$$
For; i =1, N, i ≠ Slack (7)

$$P_i = G_{ii}(e_i^2 + f_i^2) + \sum_{j=1, j\neq i}^{N} G_{ij}(e_j e_i + f_j f_i) - B_{ij}(f_j e_i - e_j f_i)$$
For; i =1, N, i ≠ Slack (8)

Equations (2) to (8) generate a system of nonlinear equations, which in the general form are expressed by means of Equation (9).

$$F(x) = 0 \quad (9)$$

In the Equation (9) solution, the Newton-Raphson method is the traditional form for calculating power flows, and different versions have been derived from it. Due to the size of the linearized system of equations, the ordering and factorization techniques have been the most used. Different methods have been proposed in the solution of large systems; for example, [8] proposes the Newton-Krylov method. With Equations (2) and (3), Newton's methods were developed in their polar form, decoupled form and fast decoupling; however, either version involves the solution of a large system of equations. Similarly, with Equations (7) and (8), the rectangular form of the Newton-Raphson method was developed, which also involves the solution of a large number of linearized equations. As an alternative to the solution of power flows, this paper proposes the solution by node of Equation (6) and the adaptation of Newton-type methods that comply with (9) and avoid the formation of large matrices, except for the admittance matrix and any factorization method. An important advantage of the matrix formulation is that it gives robustness to the solution of the linearized system of equations [24], and the number of iterations reported when solving with Newton-Raphson is six [25]. In the methods proposed in this work, because each node equation is solved separately, the lowest number of iterations was 17; however, as seen in the results section, the execution time for the largest test system solved was 78.125 milliseconds.

3. Iterative Solvers

The Newton-Raphson method in its original form is represented by Equation (10).

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - (f(x_n))^{-1} f(x_n) \quad (10)$$

For the solution of Equation (10), an initial condition x_o is required, the function *f(xn)* evaluated at each iteration n, and its derivative *f'(xn)* must exist to ensure the convergence of the method [26]. In the solution of nonlinear equations, a quadratic convergence is associated with the Newton-Raphson method, which gives robustness to the method and ensures its convergence when the initial condition is close to the solution. Methods of different order have been developed to improve convergence and efficiency; among them are those proposed by Ogbereyivwe [27] and Mudassir [26]. The Newton-type methods take as reference Equation (10), and in this paper, we have adapted methods of different steps to the solution of power flows. An advantage of the methods that have been adapted in this work is that the formation of a derivative matrix is dispensed with; instead, for each node, the power Equation (6) is generated with its derivatives and iteratively solved by applying acceleration functions that reduce the number of iterations and the execution time.

3.1. Iterative Solvers Derived from Newton-Type Methods

In the Newton-type methods used in this work, equation 6 is solved with its derivatives and Equation (10) was adapted to calculate the voltage at each network node iteratively. The solution consisted of generating a complex function $f(V_1,,,,V_N)$ with its derivatives for each node i and solving iteratively to calculate the voltage Vi at each node and the power flow in transmission lines. The five-step Newton-type methods M1, a seven-step method M2, and the M2A method [13] are adapted to compute power flows in interconnected power systems.

3.2. Five-Step Method M1

Equation (3) of [28] was modified and adapted to the power flow solution; to accelerate its convergence, Equation (15) of [13] was used. Equations (11) correspond to the M1 method.

Step 1 : It is accelerated by the function $\phi(f'(x_n), f''(x_n))$.

$$y_n = x_n - \left(\frac{f(x_n)}{f'(x_n)}\right) \emptyset(f'(x_n), f''(x_n))$$

Step 2 :

$$z_n = y_n - \frac{f(y_n)}{f'(x_n)}$$

Step 3 : It is accelerated by the function $\phi(f'(z_n),f''(z_n))$.

$$v_n = z_n - \left(\frac{f(z_n)}{f'(x_n)+f'(z_n)}\right) \emptyset(f'(z_n), f''(z_n)) \quad (11)$$

Step 4 :

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

Step 5 : It is accelerated by the function $\phi(f'(w_n), f''(w_n))$.

$$x_{n+1} = w_n - \left(\frac{f(w_n)}{f'(w_n) + f'(v_n)}\right) \emptyset(f'(w_n), f''(w_n))$$

3.3. Seven-Step Method M2

A seven-step method with acceleration is adapted for calculating power flows with the Equations (12).

Step 1 : It is accelerated by the function $\phi(f'(x_n), f''(x_n))$.

$$y_n = x_n - \left(\frac{f(x_n)}{f\prime(x_n)}\right) \emptyset(f'(x_n), f''(x_n))$$

Step 2 :

$$z_n = y_n - \frac{f(y_n)}{f\prime(x_n)}$$

Step 3 : It is accelerated by the function $\phi(f'(z_n), f''(z_n))$.

$$v_n = z_n - \left(\frac{f(y_n)}{f'(x_n)+f'(z_n)}\right) \emptyset(f'(z_n), f''(z_n))$$

Step 4 :

$$t_n = v_n - \frac{f(v_n)}{f\prime(x_n)} \quad (12)$$

Step 5 : It is accelerated by the function $\phi(f'(t_n), f''(t_n))$.

$$s_n = t_n - \left(\frac{f(t_n)}{f'(t_n)+f'(v_n)}\right) \emptyset(f'(t_n), f''(t_n))$$

Step 6 :

$$w_n = s_n - \frac{f(s_n)}{f\prime(x_n)}$$

Step 7 : It is accelerated by the function $\phi(f'(w_n), f''(w_n))$.

$$x_{n+1} = w_n - \left(\frac{f(w_n)}{f'(w_n)+f'(y_n)}\right) \emptyset(f'(w_n), f''(w_n))$$

3.4. Two-Step Method M2A

The M2A method of [13], Equations (39) and (40), is modified by additionally accelerating step (13),

$$y_n = x_n - \left(\frac{f(x_n)}{f\prime(x_n)}\right) \emptyset(f'(x_n), f''(x_n)) \quad (13)$$

$$x_{n+1} = y_n - \left(\frac{f(y_n)}{f'(y_n)+f'(x_n)}\right) \emptyset(f'(y_n), f''(y_n))$$

The acceleration function depends on the first and second derivative of the function that is solved at each step and is calculated with Equation (14).

$$\emptyset(f'(x_n), f''(x_n)) = \frac{\alpha_i f'(x_n) - f''(x_n)}{2f'(x_n)} \quad (14)$$

Being α_i a randomly chosen acceleration factor, which is between (19) and (13) and depends on the size of the system. Taylor series are used to find the acceleration function, and Appendix 2 shows how to obtain it.

4. Solution Algorithm

The programs comprise similar subprograms; a flowchart is shown in Figure 1, and each block is described below.

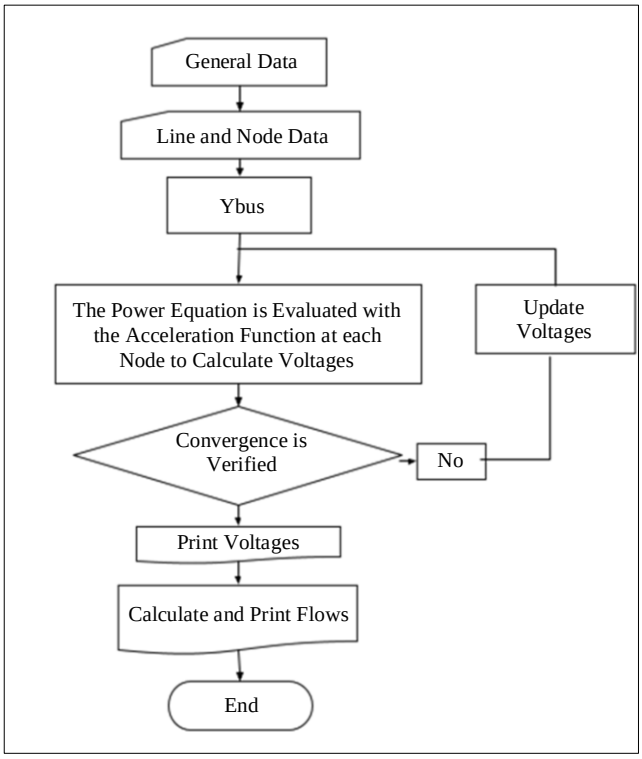

Fig. 1 Flowchart

Block 1 : Reading of general data, number of nodes, number of transmission lines, compensator node, tolerance, transformers, base power.

Block 2 : Reading of series impedance, shunt admittance, complex power generated, complex power demanded, initial voltage and characteristics of each node.

Block 3 : Formation of the admittance matrix.

Block 4 : In each step, depending on the method, the voltage correction is calculated and multiplied by the acceleration function, if applicable.

Convergence is checked using the relative value of the voltage or another convenient way. If the convergence criterion is met, the iterative process is terminated; otherwise, it is continued. If the convergence criterion is met, voltages are printed, and the power flow is calculated. The iterative process ends.

5. Results and Discussion

The M1, M2, and M2A methods were applied to IEEE test systems of 9, 30, 39 and 118 nodes. The programs were developed in FORTRAN language with Force 2.0 version on an Intel(R) Core (TM) i7-7500U CPU @ 2.70GHz with 16 Gb in RAM.

5.1. Case 1: Nine-Node Systems

Figure 2 shows the nine-node system; it contains all the data needed for a power flow study [29].

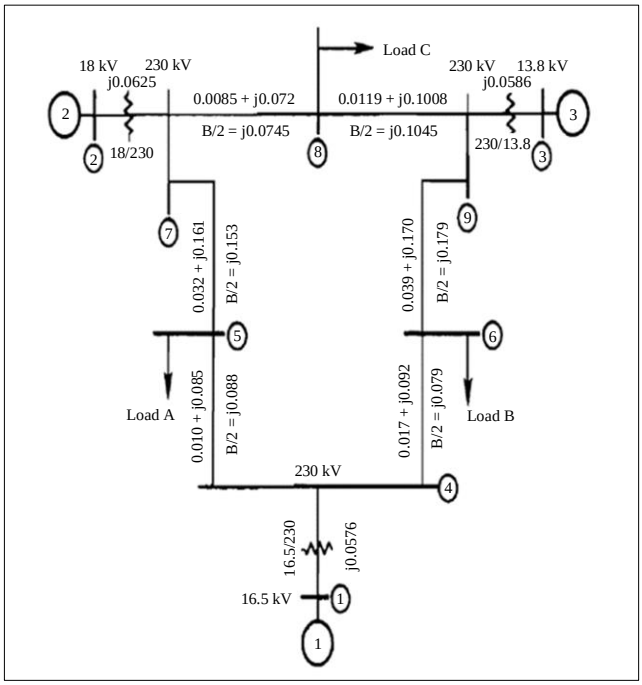

Fig. 2 Nine-node system

In the system of Figure 2, node 1 is the compensator, nodes 2 and 3 are voltage-controlled, and the other nodes are loaded. Simulating with a student version of the Powerworld program, with a tolerance of 0.1, the voltage of each node and the power flow in each generator are calculated by the program and shown in Figure 3 [30].

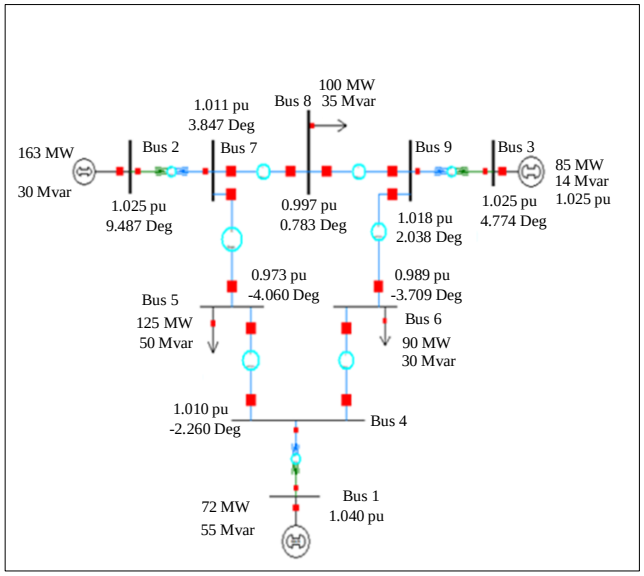

Fig. 3 Voltages and power flows calculated with Powerworld

The programs developed with the M1, M2 and M2A methods provide the magnitude and angle of the voltage at each system node in Figure 2 and are shown in Table 1 of Appendix 1. The acronyms Pw correspond to the values obtained with the Powerworld program. The execution times used with programs M1, M2 and M2A are 15.625 milliseconds, and the number of iterations is 16, 6, and 5, respectively, being the convergence used at 1.0e-06 in all programs. Table 2 of Appendix 1 shows the power flow in transmission lines calculated with the M2 program.

5.2. Case 2: 30-Node System

The thirty-node IEEE test system was analyzed using the developed programs. The magnitude of the voltages in unit values are plotted and shown in Figure 4 and their angles are plotted in Figure 5.

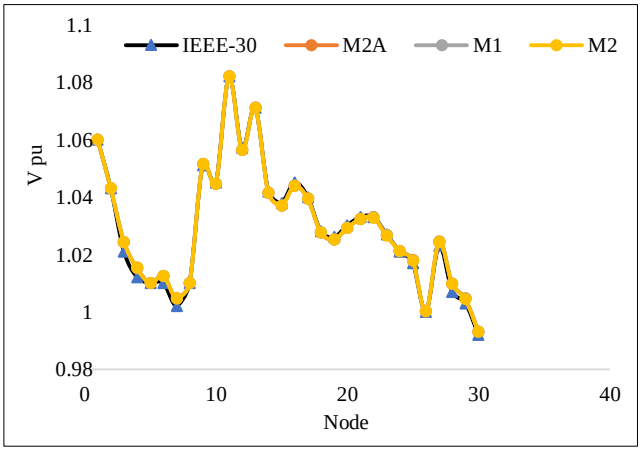

Fig. 4 Voltages calculated with programs M1, M2, M2A and reported in [31]

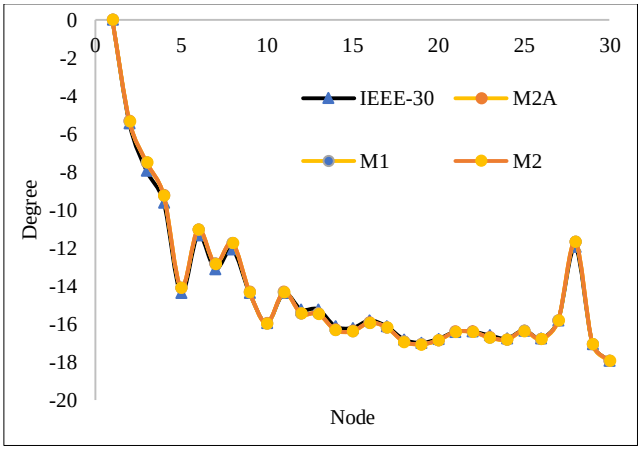

Fig. 5 Angle of the voltages in Figure 4

Figure 4 shows errors in the voltage magnitude, and those errors are plotted in Figure 6, where the error formed by the IEEE-30-M2A difference, calculated between the values reported for the IEEE-30 and those obtained with the M2A method, is shown.

In Figure 6, the maximum relative error occurs at nodes 3 and 4, being 0.3330% for node 3 and 0.326087% for node 4. In the angles, the maximum relative value is also present in

nodes 3 and 4, being for node three at 5.62% and node four at 3.38%. The errors may be caused by slight differences in the data. The tolerance used in methods M1, M2 and M2A was 1.0e-03.

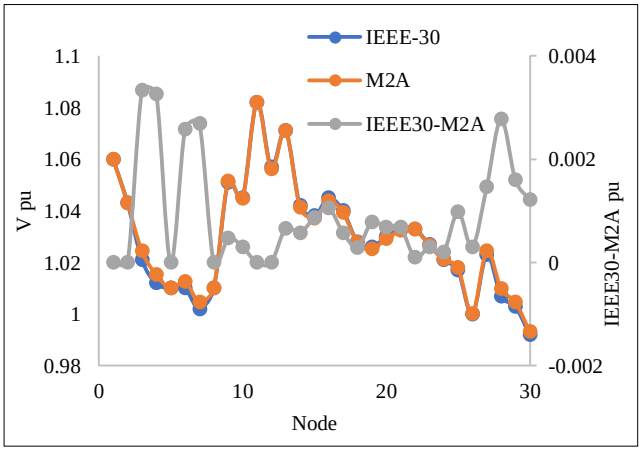

Fig. 6 IEEE-30-M2A relative error in Voltage magnitude between IEEE-30 and M2A results

5.3. Case 3: 39-Node System

The 39-node system was simulated with the programs M1, M2 and M2A; the tolerances used in the three programs were 1.0e-05, and the data were taken from [32]. Figure 7 shows the magnitude of the voltage at each of the nodes in unit values.

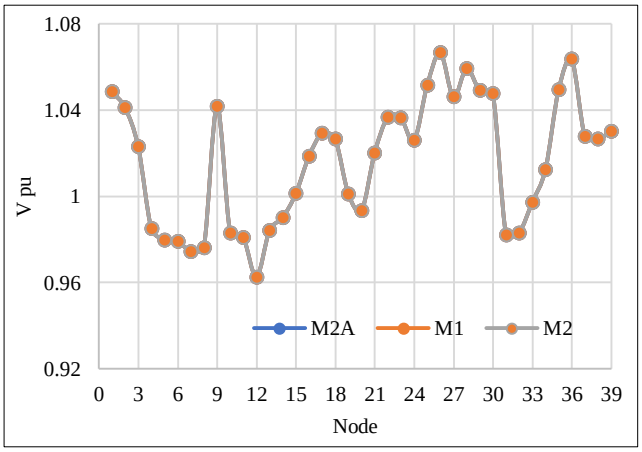

Fig. 7 Magnitude of the 39-node system voltages calculated with methods M1, M2 and M2A

In Figure 7, the results of the three methods are identical; the comparison with the source values is not shown because the tolerance value used in the reference is unknown.

5.4. Case 4: 118-Node System

The 118-node IEEE test system was simulated with programs M1, M2 and M2A; the tolerances used in the three programs were 1.0e-04 [31]. Figure 8 shows the voltage magnitude at each node in unit values.

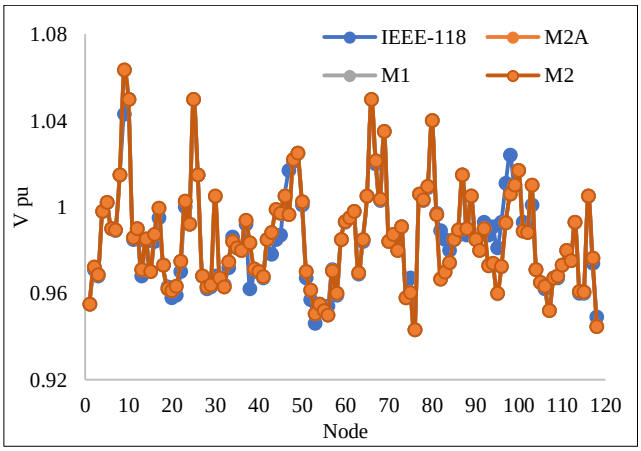

Fig. 8 Magnitude of the voltages of the 118-node system calculated with M1, M2 and M2A

There are differences between the values calculated with the programs of this work and the reference; the largest of them are shown in Figure 9, where it has been plotted from nodes 38 to 53 and from 82 to 98; it is observed that the maximum relative error in percent is presented at node 82 with a value of 2.254%.

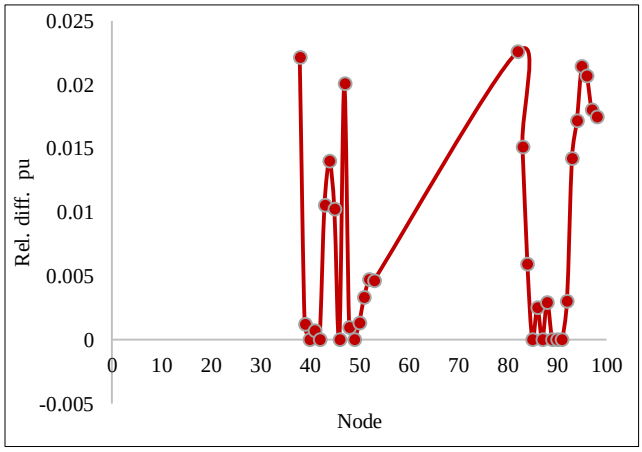

Fig. 9 Differences between calculated and reported values [31]

Table 3 of Appendix 1 shows the acceleration factors, the number of iterations and the execution time in milliseconds each method takes, and also shows that in three of the four cases, any method is applicable.

The tolerances considered in the 9-node system were 1.-05, in the 30-node system 1.03, the 39-node system 1.0e-05, and the 118-node system 1.0e-04. Table 4 shows the importance of increasing the number of steps when the size of the system to be solved increases. This opens the possibility of studying Newton-type methods with a higher number of steps to analyze systems larger than 118 nodes. The objective of the acceleration function in some steps of a numerical method is to reduce the increment $f(xn)/f'(x)$ or $f(x)/(f'(x)+f'(y))$ so that the number of iterations is smaller, and therefore the

execution time in a digital program. The acceleration factors have been defined through the simulation of each of the cases, as can be seen in Table 4; they vary depending on the system under study and the method. One possible way to select their values would be with the calculation of the maximum and minimum eigenvalues; however, the corresponding simulations would also have to be carried out with each of their values and it can be very costly.

From the simulations performed in Table 4, it is observed that the value of α1 is between 3 and 7 for the three methods, although the latter value in the M2A method may not converge for the 118-node system. The value of α_2 was selected after carrying out simulations with different values, observing that the calculated voltages and the balance of power flows at each node were met. The same procedure was carried out with the values of α_3, α_5 and α_7 in the M1 and M2 methods. An inference is presented here, and there are optimal values of the acceleration factors related to the voltage and power flow in the network. From here, future work arises to calculate these optimal values. Figure 10 shows the behavior of node 88 at step 1, step 3 and step 5 when the 118-node system is solved with the M1 method.

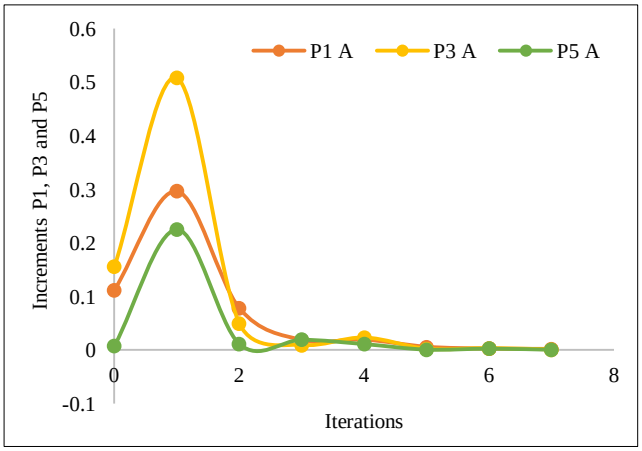

Fig. 10 Behavior of node 88 in steps 1, 3 and 5

If no acceleration factors are applied, convergence is slow. Figure 11 shows the behavior of node 88 of the system of 118 when there is no acceleration in steps 1, 3 and 5.

The factors αi of the acceleration function ϕ(f'(xn), f''(xn)) considerably reduce the number of iterations. Figures 10 and 11 show remarkable differences. Without acceleration, the convergence is slow; with acceleration, the convergence is dynamic, reducing the execution time considerably. Table 4 in Appendix 1 shows the execution times and the number of iterations of the three methods M1, M2 and M2A. When there was no acceleration, the tolerance in all cases was 1.0e-04.

Comparing Tables 3 and 4, the advantage of the acceleration of the methods can be observed. It is also important to note that the execution time depends on the characteristics and background activities of the equipment used in the simulations. For the 118-node system, multi-step methods become important, as shown in Tables 3 and 4.

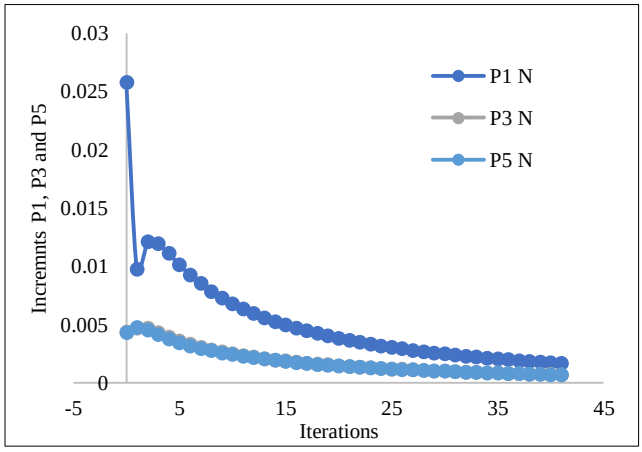

Fig. 11 Convergence in steps 1, 3 and 5 when there is no convergence

6. Conclusion

Three Newton-type methods were applied in this paper to solve Equation (6); with its solution, the voltage was obtained in each of the nodes of the test systems iteratively. The methods used are of simple code; this represents an advantage due to the fact that only elementary operations were carried out, except for the derivatives involved in the study. The following conclusions can be drawn from the results obtained;

- Without the formation of the Jacobian matrix, the voltages at each node of the electrical test power systems were calculated.
- An adaptation of a previously published method was made, and two more methods were adapted with the objective of knowing the characteristics of each one when applying them to the solution of the test systems.
- Acceleration functions were applied, referencing a previous publication by the authors.
- In the graphs and tables obtained, the three methods present similar characteristics and can be applied to the solution of test systems up to 39 nodes, with or without the application of acceleration factors.
- The seven-step method, with the acceleration factors in Table 3, is the most suitable, depending on the execution time, for the solution of the 118-node system.
- The acceleration factors can be selected by the user, and their value depends on the power system; their typical values for each method and the solved systems can be seen in Table 3.
- To date, there is no rule for selecting the acceleration factors α_i; they are selected randomly.
- The objective set by applying the methods proposed in this work was met due to the fact that the execution times

- and the number of iterations were significantly reduced when solving the larger system. The formulation used in this paper aims to avoid the formation of the Jacobian matrix and the use of linear equation solution methods. The methods are neither better nor worse, and in the authors' opinion, they are considered alternatives for the analysis of electrical systems.

There is future work arising from this work;

- Analyze power systems with more than 118 nodes.
- The equations that model a solar panel can possibly be adapted with the methods applied in this paper.
- Calculate the optimal value of the acceleration factors α_i.

References

[1] 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]

[2] W.F. Tinney, and J.W. Walker, "Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization," *Proceedings of the IEEE*, vol. 55, no. 11, pp. 1801-1809, 1967. [CrossRef] [Google Scholar] [Publisher Link]

[3] 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]

[4] F. De Leon, and A. Semlyen, "Iterative Solvers in the Newton Power Flow Problem: Preconditioners, Inexact Solutions and Partial Jacobian Updates," *IEE Proceedings-Generation, Transmission and Distribution*, vol. 149, no. 4, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[5] Reijer Idema et al., "Scalable Newton-Krylov Solver for Very Large Power Flow Problems," *IEEE Transactions on Power Systems*, vol. 27, no. 1, pp. 390-396, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[6] Reijer Idema, and Domenico J.P. Lahaye, "Nerwton-Krylov Power Flow Solver," *Computational Methods in Power System Analysis*, pp. 73-81, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[7] Stanley C. Eisenstat, and Homer F. Waker, "Chossing the Forcing Terms in an Inexact Newton Method," *SIAM Journal on Scientific Computing*, vol. 17, no. 1, 1996. [CrossRef] [Google Scholar] [Publisher Link]

[8] Reijer Idema et al., "Towards Faster Solution of Large Power Flow Problems," *IEEE Transactions on Power Systems*, vol. 28, no. 4, pp. 4918-4925, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[9] Napoleon Costilla-Enriquez, Yang Weng, and Baosen Zhang, "Combining Newton-Raphson and Stochastic Gradient Descent for Power Flow Analysis," *IEEE Transactions on Power Systems*, vol. 36, no. 1, pp. 514-517, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[10] Nobou Sato, and W.F. Tinney, "Techniques for Exploiting the Sparsity or the Network Admittance Matrix," *IEEE Transactions on Power Apparatus and Systems*, vol. 82, no. 69, pp. 944-950, 1963. [CrossRef] [Google Scholar] [Publisher Link]

[11] Glenn W. Stagg, and Ahmed H. El-Abiad, *Computer Methods in Power System Analysis*, McGraw Hill Series in Electronic Systems, 1968. [Publisher Link]

[12] Hadi Saadat, *Power System Analysis*, 3rd ed., McGraw Hill Series Electrical and Conputer Engineering, 2017. [Google Scholar] [Publisher Link]

[13] Ruben Villafuerte Diaz et al., "Analysis of Electrical Power Systems with Newton-Type Accelerated Numerical Methods," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 10, no. 11, pp. 148-157, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[14] Jaan Kiusalaas, *Numerical Methods in Engineering with Python 3*, Cambridge University Press, 2013. [Google Scholar] [Publisher Link]

[15] Shoichiro Nakamura, *Métodos Numericos Aplicados con Software*, Prentice Hall Mexico, 1992. [Google Scholar] [Publisher Link]

[16] Mohamed Bahgat, "New Two-step Iterative Methods for Solving Nonlinear Equations," *Journal of Mathematics Research*, vol. 4, no. 3, pp. 128-131, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[17] Farooq Ahmed Shah, Muhammad Aslam Noor, and Khalida Inayat Noor, "Some Iterative Schemes for Obtaining Approximate Solution of Nonlinear Equations," *The Scientific Bulletin Series A*, vol. 78, no. 1, pp. 59-70, 2016. [Google Scholar] [Publisher Link]

[18] Mohammed Rasheed et al., "Various Numerical Methods for Solving Nonlinear Equation," *Journal of Al-Qadisiyah for Computer Science and Mathematics*, vol. 13, no. 3, pp. math 88-97, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] Alicia Cordero et al., "Multistep High-Order Methods for Nonlinear Equations Using Padé-Like Approximants," *Discrete Dynamics in Nature and Society*, vol. 2017, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[20] William D. Stevenson, *Analisis De Sistemas Eléctricos de Potencia*, McGraw-Hill, 1975. [Google Scholar] [Publisher Link]

[21] John J. Grainger, and William D. Stevenson Jr, *Análisis de Sistemas de Potencia*, McGraw-Hill, 1996. [Google Scholar] [Publisher Link]

[22] Pieter Schavemaker, and Lou van der Sluis, *Electrical Power System Essentials*, Wiley, 2008. [Google Scholar] [Publisher Link]

[23] Jan Vysocký et al., "Steady-State Analysis of Electrical Networks in Pandapower Software: Computational Performances of Newton–Raphson, Newton–Raphson with Iwamoto Multiplier, and Gauss–Seidel Methods," *Sustainability*, vol. 14, no. 4, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[24] Manolo D'orto et al., "Comparing Different Approaches for Solving Large Scale Power Flow Problems on the CPU and GPU with the Newton-Raphson Method," *IEEE Access*, vol. 9, pp. 56604-56615, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[25] Ogbereyivwe Oghovese, and Atoma O. Johnson, "New Two-Step Method with Fifth-Order Convergence for Solving Nonlinear Equations," *International Journal of Mathematics and Statistics Invention (IJMSI)*, vol. 2, no. 10, pp. 24-27, 2014. [Google Scholar] [Publisher Link]

[26] Mudassir Shams et al., "Efficient Iterative Scheme for Solving Non-Linear Equations with Engineering Applications," *Applied Mathematics in Science and Engineering*, vol. 30, no. 1, pp. 708-735, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[27] Ogbereyivwe Oghovese, and Emunefe John, "Two Steps Iterative Methods for Solving Nonlinear Equations," *International Journal of Mathematics and Computer Research*, vol. 2, no. 8, pp. 600-605, 2014. [Google Scholar] [Publisher Link]

[28] Shuping Chen, and Youhua Qian, "A Family of Combined Iterative Methods for Solving Nonlinear Equations," *American Journal of Applied Mathematics and Statistics*, vol. 5, no. 1, pp. 22-32, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[29] P.M. Anderson, and A.A. Fouad, *Power System Control and Stability*, John Wiley & Sons, 2008. [Google Scholar]

[30] Power World Corporation, 2024. [online]. Available: https://www.powerworld.com/

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

[32] Caicedo Rivadeneira, and Javier Andrés, "*Flujo Óptimo de Potencia en Sistemas Eléctricos Basado en Criterios de Mínimas Pérdidas de Potencia Activa Usando el Método de la Gradiente*," Thesis, Universidad Politécnica Salesiana Sede Quito, Marzo, 2022. [Google Scholar] [Publisher Link]

Appendix 1

Table 1 shows the magnitude and angle of the voltages of the nine-node system.

Table 1. Voltage magnitude and angle at each node

Node	M2A		Pw		M1		M2	
	V pu	δ deg	V pu	δ deg	V pu	δ deg	V pu	δ deg
1	1.04	0.0	1.04	0.0	1.04	0.0	1.04	
2	1.025	9.2802	1.025	9.49	1.025	9.2799	1.025	9.2799
3	1.025	4.6648	1.025	4.77	1.025	4.6646	1.025	4.6646
4	1.0257	-2.2167	1.01	-2.26	1.0258	-2.2168	1.0257	-2.2168
5	0.9956	-3.9887	0.97	-4.06	0.9956	-3.9889	0.9956	-3.9888
6	1.0126	-3.6873	0.99	-3.71	1.0127	-3.6875	1.0126	-3.6874
7	1.0257	3.7197	1.01	3.85	1.0258	3.7196	1.0257	3.7196
8	1.0158	0.7275	1.0	0.78	1.0159	0.7275	1.0158	0.7274
9	1.0323	1.9667	1.02	2.04	1.0324	1.9667	1.0323	1.9666

Table 2 shows the power flow of S_{ij} and S_{ji} flowing through each transmission line of the nine-node system.

Table 2. Power flow in transmission lines

Power Flow in Transmission Lines					
Np	Nq	S_{ij}	Nq	Np	S_{ji}
1	4	71.6410 +j27.0461	4	1	-71.6410-j23.9233
2	7	163.0000+j6.6533	7	2	-163.0000+j9.1785
3	9	84.9999-j10.8600	9	3	-84.9999+j14.955
4	5	40.9374+j22.8931	5	4	-40.6799-j38.6873
4	6	30.7038+j1.0300	6	4	-30.5374-j16.5434
5	7	-84.3202-j11.3130	7	5	86.6201-j8.3806
6	9	-59.4628-j13.4568	9	6	60.8167-j18.0747
7	8	76.3799-j0.7969	8	7	-75.9046-j10.7046
8	9	-24.0955-j24.2960	9	8	24.1835+j3.1197

Table 3 shows the acceleration factors, iterations and execution times of methods M1, M2 and M2A.

Table 3. Acceleration factors used in programs M1, M2 and M2A

Node	M2A				M1					M2					
	α_1	α_2	It	Cpu ms.	α_1	α_3	α_5	It	Cpu ms.	α_1	α_3	α_5	α_7	It	Cpu ms.
9	3.5	10	16	15.625	5.5	10	10	6	15.625	3	10	10	10	5	15.625
30	2.9	10	19	15.625	5.84	10	10	8	15.625	4	10	10	10	9	15.625
39	4.5	10	102	46.875	6	10	10	25	31.250	6	10	10	10	18	15.625
118	5.5	10	140	187.5	7	10	12	45	140.625	7	12	12	12	17	78.125

Table 4 shows the execution time and the number of iterations when the methods are not accelerated.

Table 4. Execution times and iterations without acceleration in programs M1, M2 and M2A

	M2A		M1		M2	
Node	It	Cpu (ms)	It	Cpu (ms)	It	Cpu (ms)
9	42	15.625	15	15.625	11	15.625
30	170	15.625	62	31.25	46	15.625
39	403	62.5	145	46.875	107	78.125
118	926	1312.5	339	1078.125	240	1078.125

Appendix 2

The acceleration function is obtained by applying the Taylor series expansion to the Newton-Raphson method.

$$x_{n+1} = x_n - \left[\frac{f(x_n)}{f\prime(x_n)}\right]\Phi \tag{15}$$

The coefficient Φ is an over-relaxation factor that is used to speed up the convergence of Newton's method. Generally, its value oscillates between 1 and 2.

From (15)

$$\frac{f(x_n)\Phi}{f'(x_n)} = (e_n - C_2e_n^2 + (2C_3 - 2C_2^2)e_n^3 + e_n^4(4C_2^3 - 7C_2C_3 + 3C_4) + (O)e_n^5)\Phi \tag{16}$$

The error e_{n+1} is written by equation (17)

$$e_{n+1} = e_n - (e_n - C_2e_n^2 + (2C_3 - 2C_2^2)e_n^3 + (O)e_n^4) \tag{17}$$

Simplifying (17).

$$e_{n+1} = e_n(1-\Phi) + \Phi C_2e_n^2 + 2\Phi e_n^3(C_3 - C_2^2) + (O)e_n^4 \tag{18}$$

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

Taking two terms of equation (18) and considering that $e_{n+1} = 0$, $e_n = 1$, substituting C_2 and simplifying.

$$\Phi\big(f'(x_n), f''(x_n)\big) = \frac{2f'(x_n)}{2f'^{(x_n)} - f''(x_n)} = \frac{\alpha_1 f'(x_n)}{2f'^{(x_n)} - f''(x_n)} \tag{19}$$

In equation (19), Φ is an over-relaxation function that depends on the first and second derivatives of $f(x)$. For reasons of better convergence, Equation (19) is inverted.

$$\Phi(f'(x_n), f''(x_n)) = \frac{\alpha_1 f\prime(x_n) - f\prime\prime(x_n)}{2f\prime(x_n)} \tag{20}$$

Where α_1 is the acceleration factor and is between (19) and (13).