Two Iterative Methods to Solve Nonlinear Equations of Load Flows

Rubén Villafuerte D., Jesús Medina C., Rubén A. Villafuerte S., Victorino Juárez R.

Abstract: This paper presents the results obtained when two iterative methods are applied to the solution of non-linear equations that model the load flow in electric power systems. Two iterative methods are applied; the first consists of a simplification of the rectangular form the traditional Newton-Raphson method, the second is a hybrid method and relates the simplified form proposed here and a four-step Newton-type iterative method. The convergence characteristic and the mathematical preliminaries of the iterative four-step method are included in the paper. The methods were used to calculate the voltages at each node of the IEEE test system of 118 nodes and a distribution system of 40 nodes. In each method, the formation of the Jacobian matrix, widely used in traditional forms of load flows, is avoided and only elementary operations are carried out, impacting the execution times for the test systems used, being of the order of 15.6 to 279 milliseconds. The maximum error found is for the 118 node system and is of the order of 3.7%.

Keywords: Iterative methods; load flows; nonlinear equations; power systems; Newton-Raphson.

1. INTRODUCTION

Power flow calculations evolved from the planning departments of electric utilities in the early 20th century, where they were (and still are) used to simulate the effects of network augmentations, different load scenarios, network configurations, etc. Early power flows were solved by AC network analyzers, which were analog computers containing resistors, inductors, and capacitors calibrated to be a miniature scale equivalent of the actual electrical network. For large systems, these network analyzers took up the space of entire rooms. As they were physically in nature, the network analyzers needed to be re-wired for each different configuration, augmentation or scenario to be studied. Starting with the paper by Ward and Hale in 1956, the use of digital computers would enable power flow calculations to be performed faster and in a more flexible manner [1]. The iterative method proposed by Ward and Hale (which became known as the Gauss-Seidel method) was straightforward to implement but exhibited poor convergence characteristics.

In 1961, Van Ness and Griffin introduced Newton's method for solving power flows (which later became known as the Newton-Raphson method), which had more favorable characteristics but was limited to smaller networks due to computer memory requirements [2]. In their 1967 paper, Tinney and Hart exploited the sparsity properties of the Ybus matrix (and by extension, the Jacobian matrix) and in conjunction with numerical techniques such as triangular factorization and optimal ordering, were able to make the Newton-Raphson algorithm practical for solving large networks using the computing resources available at the time [3]. In 1974, Stott and Alsac, recognized that in many practical networks there was decoupling between 1) active power and voltage and 2) reactive power and phase angle [4]. The proposed an algorithm (the fast decoupled load flow) that significantly reduced the computational requirements to construct the Jacobian matrix and therefore led to large gains in computational speed (although at the expense of a little accuracy). In paper [5] gives an overview of the various load flow techniques of the weakly meshed distribution system which are very efficient, because various classical methods are not having sufficient convergence criterion for solving the large distribution system. In paper [6] a simple and powerful algorithm has been proposed for balanced radial distribution network to obtain power flow solution. It has been found from the cases presented that the proposed method has fast convergence characteristics when compared to existing methods. The algorithm is found to be robust in nature. The method can be easily extended to solve three phase networks also. In paper [7] a procedure is established for solving the Probabilistic Load Flow in an electrical power network, considering correlation between power generated by power plants, loads demanded on each bus and power injected by wind farms. The method proposed is based on the generation of correlated series of power values, which can be used in a Monte Carlo simulation, to obtain the probability density function of the power through branches of an electrical network. In paper [8] a method is proposed to improve solutions for intervals of power flow problems due to nondeterministic characteristics of demand loads and output of generators. In paper [9] a detailed study for load flow analysis in distributed power system. A case study of modeling and simulation of the distribution network is implemented with the electrical transient analyzer program [10].

Revised Manuscript Received on December 15, 2019.
Correspondence author*

Rubén Villafuerte D*, Universidad Veracruzana, Facultad de Ingeniería, Ixtaczoquitlán Ver., México, rubenv46@yahoo.com.mx, rvillafuerte@uv.mx

Jesús Medina C, Universidad Veracruzana, Facultad de Ingeniería, Ixtaczoquitlán Ver., México, jemedina@uv.mx

Rubén A. Villafuerte S, Instituto Tecnológico Nacional, Campus Orizaba, Orizaba, Ver., México, dreyko999@gmail.com

Victorino Juárez R., Universidad Veracruzana, Facultad de Ingeniería, Ixtaczoquitlán Ver., México, vijuarez@uv.mx
II. PROPOSED METHODS AND CONVERGENCE ANALYSIS

The objective of this work is to calculate the Voltage in each node of an electrical power system. To achieve this, the nonlinear equations that model the load flow are established and solved by applying a simplification of the Jacobian matrix used in the rectangular shape of the Newton-Raphson method [11]. Equation (1) models the load nodes of a system.

\[
\begin{bmatrix}
\frac{\partial P_1}{\partial V_1} & \frac{\partial P_1}{\partial V_2} & \cdots & \frac{\partial P_1}{\partial V_N} \\
\frac{\partial P_2}{\partial V_1} & \frac{\partial P_2}{\partial V_2} & \cdots & \frac{\partial P_2}{\partial V_N} \\
\vdots & \vdots & \ddots & \vdots \\
\frac{\partial P_N}{\partial V_1} & \frac{\partial P_N}{\partial V_2} & \cdots & \frac{\partial P_N}{\partial V_N}
\end{bmatrix}
\begin{bmatrix}
\Delta V_1 \\
\Delta V_2 \\
\vdots \\
\Delta V_N
\end{bmatrix} =
\begin{bmatrix}
\Delta P_1 \\
\Delta P_2 \\
\vdots \\
\Delta P_N
\end{bmatrix}
\]

(1)

The simplified form is generated for the diagonal elements of the submatrices of eq. (1). To calculate the rectangular components of the Voltages a system of two equations (2) is generated.

\[
\begin{bmatrix}
\frac{\partial P_i}{\partial V_i} & \frac{\partial P_i}{\partial V_j} \\
\frac{\partial Q_i}{\partial V_i} & \frac{\partial Q_i}{\partial V_j}
\end{bmatrix}
\begin{bmatrix}
\Delta e_i \\
\Delta e_j
\end{bmatrix} =
\begin{bmatrix}
\Delta P_i \\
\Delta Q_i
\end{bmatrix}
\]

(2)

The elements of the submatrix J of (1) are generated in each iteration and depend on the partial derivatives of the real power or reactive power with respect to the nodal Voltages. The real and reactive power in each node is calculated with (3) and (4), respectively [11, 12, 17].

\[
P_i = e_i \sum_{j=1}^{N} (G_{ij} e_j - B_{ij} e_j), \quad i = 1, N, i \neq \text{Slack}
\]

(3)

\[
Q_i = f_i \sum_{j=1}^{N} (G_{ij} e_j - B_{ij} e_j) - e_i \sum_{j=1}^{N} (G_{ij} e_j + B_{ij} e_j), \quad i = 1, N, i \neq \text{Slack}
\]

(4)

The partial derivatives of the real and reactive power with respect to the rectangular components are expressed by (5)-(8).

\[
\frac{\partial P_i}{\partial e_i} = 2G_{ii} e_i + \sum_{i=1,i\neq j}^{N} (G_{ij} e_j - B_{ij} e_j)
\]

(5)

\[
\frac{\partial P_i}{\partial e_j} = 2G_{ij} e_j + \sum_{i=1,j\neq i}^{N} (G_{ij} e_j + B_{ij} e_i)
\]

(6)

\[
\frac{\partial Q_i}{\partial e_i} = -2B_{ii} e_i - \sum_{i=1,i\neq j}^{N} (G_{ij} e_j + B_{ij} e_j)
\]

(7)

\[
\frac{\partial Q_i}{\partial e_j} = -2B_{ij} e_j - \sum_{i=1,j\neq i}^{N} (G_{ij} e_j - B_{ij} e_j)
\]

(8)

The increases in real and reactive power are calculated in each iteration with (9)-(10).

\[
\Delta P_i = P_{i, \text{spec}} - P_{i, \text{calc}} \quad \text{if not \, \text{Slack}}
\]

(9)

\[
\Delta Q_i = Q_{i, \text{spec}} - Q_{i, \text{calc}} \quad \text{if not \, \text{Slack}}
\]

(10)

Where:

- P_{i, \text{spec}}: Real power load in node i
- Q_{i, \text{spec}}: Reactive power load in node i
- P_{i, \text{calc}}: Real power calculated at node i
- Q_{i, \text{calc}}: Reactive power calculated at node i

For the solution of real nonlinear equations, methods of several steps and of high order of convergence have been proposed [13-17]. The method used in this work is four steps and only the function f(x_n) and its derivative f'(x_n) are necessary.

A. Convergence analysis for the four-step method

In this section, the iterative method of four steps and fifth order of convergence is constructed, for which the following basic definitions are necessary:

\[\lim_{n \to \infty} x_{n+1} - \alpha = A \neq 0\]

or equivalently

\[x_n - \alpha = A(x_n - \alpha)^m + O((x_n - \alpha)^{m+1})\]

\[\text{Definition A.1.}\]

The efficiency of a method is measured by the index $E_1=\rho^1$, where $\rho$ is the order of the iterative method and $\beta$ is the total number of function evaluations per iteration.

Now, we consider the iteration scheme:

\[y_n = x_n - \left(\frac{f(x_n)}{f'(x_n)}\right)\alpha_1\]

(11)

\[z_n = y_n - \left(\frac{f(y_n)}{f'(y_n)}\right)\alpha_2\]

(12)

\[u_n = z_n - \left(\frac{f(z_n)}{f'(z_n)}\right)\alpha_3\]

(13)

\[x_{n+1} = u_n - \left(\frac{f(u_n)}{f'(u_n)}\right)\alpha_4\]

(14)

Where: $\alpha_1$, $\alpha_2$, $\alpha_3$ and $\alpha_4$ are real constants

\[\text{Theorem A.1.}\]

Let $\alpha$ be a simple zero of sufficiently differentiable function $f: I \subseteq C \rightarrow C$ for an open interval $I$. If $x_0$ is sufficiently close to $\alpha$, the method defined by eqs. (11)-(14) has local order of convergence at least 5, with the following error equation

\[e_{n+1} = -e_n^4 \left(38C_k^2 + 4C_k^2 C_3 - 7C_k C_4 - 2C_k^2 + 4C_5\right) + C_k e_n^2 \left(118C_k^2 - 125C_k^2 C_3 + 4C_k C_4 + 32C_k^2 - 2C_3\right) + O(e_n^5)\]

Where:

- $C_k = \frac{k^\alpha}{\alpha k^\alpha}$, $k = 1,2,3$...
- $\alpha$ and the error function is expressed as:

\[e_n = x_n - \alpha\]

\[\text{Proof.}\]

Let $\alpha$ be a simple zero of $f(x)$. Then by using Taylor series with Derive, we have.

In the convergence test of an iterative method, each of the equations is developed in Taylor series. Thus, by developing a Taylor’s series the function $f(x_n)$, we have eq. (15).

\[f(x_n) = f(\alpha) e_n + C2 e_n^2 + C3 e_n^3 + C4 e_n^4 + C5 e_n^5 + C6 e_n^6 + O(e_n^7)\]
Deriving successively $f(x_k)$, we obtain (16)-(20).

\[
f(x_k) = f'(a)(e_k + C_2 e_k^2 + C_3 e_k^3 + C_4 e_k^4 + C_5 e_k^5 + O(e_k^6))
\]  
(16)

\[
f'(x_k) = f'(a)(1 + 2C_2 e_k + 3C_3 e_k^2 + 4C_4 e_k^3 + 5C_5 e_k^4 + O(e_k^5))
\]  
(17)

\[
f''(x_k) = f'(a)(2C_2 + 6C_3 e_k + 12C_4 e_k^2 + 20C_5 e_k^3 + O(e_k^4))
\]  
(18)

\[
f'''(x_k) = f'(a)(6C_3 + 24C_4 e_k + 60C_5 e_k^2 + O(e_k^3))
\]  
(19)

\[
f^{(4)}(x_k) = f'(a)(24C_4 + 120C_5 e_k + O(e_k^2))
\]  
(20)

Where:

\[
C_j = \frac{f^{(j)}(x^\ast)}{j!f'(x^\ast)} \quad j=2,3,..., \quad y_e = x_e - x^\ast
\]

From (16)-(17), we have:

\[
\delta_k = \frac{f(x_k)}{f'(x_k)} = -e_k + C_2 e_k^2 + 2e_k^3(C_3 - C_2^2) + e_k^4(-7C_2C_3 + 4C_3^2 + 3C_4) - 2e_k^5(4C_2^2 - 10C_2C_3 + 5C_2C_4 + 3C_3 - 2C_2^2) + O(e_k^6)
\]  
(21)

From (16), (17) and (11), we have:

\[
y_k = C_2 e_k^2 + 2e_k^3(C_3 - C_2^2) + e_k^4(-7C_2C_3 + 4C_3^2 + 3C_4) - 2e_k^5(4C_2^2 - 10C_2C_3 + 5C_2C_4 + 3C_3 - 2C_2^2) + O(e_k^6)
\]  
(22)

Expanding \( f(y_k) \) in the neighborhood of \( x_k \), we have:

\[
f(y_k) = f(x_k) + f'(x_k)\delta_k + \frac{1}{2}f''(x_k)\delta_k^2 + \frac{1}{6}f^{(3)}(x_k)\delta_k^3 + \frac{1}{4!}f^{(4)}(x_k)\delta_k^4 + O(e_k^6)
\]  
(23)

From the equation:

\[
z_k = y_k - \alpha_2 f(y_k)
\]

Considering that \( \alpha_2 \) is equal to one and relating (16), (23) and (12), we have that \( z_k \) is equal to:

\[
z_k = 2C_2 e_k^2 + e_k^3(7C_3 - 9C_2^2) + e_k^4(30C_2^2 - 44C_2C_3 + 10C_2C_4 + 6C_3^2 - 2C_3 - 2C_2^2) + O(e_k^6)
\]  
(25)

Applying the same procedure as (24), we have:

\[
f(z_k) = 2C_2 e_k^2 + C_2 e_k^3(7C_3 - 9C_2^2) + e_k^4(19C_2^2 - 30C_2C_3 + 8C_2C_4 - 6C_3 + 2C_2C_4) + O(e_k^6)
\]  
(26)

With (16), (24) and substituted in (13), we have:

\[
u_k = 14e_k^3(C_3 - C_2^2) + e_k^4(20C_2^2 - 22C_2C_3 + 2C_2C_4 + 6C_3 - 12C_2 + 6C_3 + 3C_2 + 5C_2C_3 + 5C_2C_4 + 3C_3 - 2C_2^2) + O(e_k^6)
\]  
(27)

Evaluating the function \( f(u_k) \) as in the equation (23), we have:

\[
f(u_k) = 14e_k^3(C_3 - C_2^2) + e_k^4(30C_2^4 + 10C_2^2C_3 - 5C_2C_4 + 3C_3 + 5C_5 + 2C_2e_k^3 + 18C_2C_4 + 12C_2C_3 + 10C_2C_4 - 3C_3 + 2C_5 + O(e_k^6)
\]  
(28)

With (16), (28) and (14), considering that \( a_k \) is equal to one, we have:

\[
x_{k+1} = u_k - \frac{f(u_k)}{f'(u_k)} = -e_k^2(38C_2^3 + 4C_2C_3 - 7C_2C_4 - 2C_3^2 + 4C_2C_4 + 3C_3 + 3C_2^2 - 2C_2^3) + O(e_k^6)
\]  
(29)

Equation (29) shows that the four-step method has a fifth order of convergence.

The functions \( f(x_i) \) with their derivatives of the four-step method modeled by (11)-(14), applied to the complex power equations in each node of an electric power system.

For node \( i \), of a power system of \( N \) nodes, we have (30).

\[
f_i(V_i, V_{i+1}, ..., V_N) = S_i - V_i ' \sum_{j=1}^{N} Y_{ij} V_j \]

\( i \neq \text{Slack} \)

(30)

Where:

\( N \) Number nodes,
\( S_{ij} \) Net complex power demanded in each node
\( Y_{ij} \) Admittance between node \( i \) and node \( j \),
\( V_i' \) Voltage node \( i \),
Also: \( \text{Y}_{i*}, \text{V}_{i*}, \text{S}_{i*} \) and \( f_i(V_i, V_{i+1}, ..., V_N) \) are complex quantities and * it means conjugate of

Equation (30) is applied to the four-step-method and (31)-(34) are established for each node:

Step 1:

\[
V_{ip1}^{(k+1)} = \frac{V_{ip1}^{(k)} - \frac{S_i - \sum_{j=1}^{N} Y_{ij} V_{jp1}^{(k)}}{\sum_{j=1}^{N} Y_{ij} V_{jp1}^{(k)}} a_i}{\alpha_1} \quad i=1,N
\]

Slack node

(31)

Step 2:

\[
V_{ip2}^{(k+1)} = \frac{V_{ip2}^{(k)} - \frac{S_i - \sum_{j=1}^{N} Y_{ij} V_{jp2}^{(k)}}{\sum_{j=1}^{N} Y_{ij} V_{jp2}^{(k)}} a_i}{\alpha_2} \quad i=1,N, \quad i \neq \text{Slack node}
\]

(32)

Step 3:

\[
V_{ip3}^{(k+1)} = \frac{V_{ip3}^{(k)} - \frac{S_i - \sum_{j=1}^{N} Y_{ij} V_{jp3}^{(k)}}{\sum_{j=1}^{N} Y_{ij} V_{jp3}^{(k)}} a_i}{\alpha_3} \quad i=1,N
\]

(33)
Two Iterative Methods to Solve Nonlinear Equations of Load Flows

FORTRAN programs were developed to solve the nonlinear equations of the Voltage in each node of an electrical power system and apply to two test systems. That is, for the equation (2) the M1 program is developed and its results are compared with the test systems of 118 nodes of the IEEE and a distribution system of 40 nodes [18, 19]. In all the simulations it is considered that the system operates in balanced conditions. In fig. 1, a flowchart shows the solution process of the proposed simplification and in fig. 2, the flowchart of the M2 program is shown, which considers the solution of (2) and the solution of (31)-(34).

III. RESULTS AND DISCUSSION

In block one the general data are read, such as; number of nodes, number of lines, base power, number of transformers and the slack node.

In each block of Figure 1, the following activities are carried out:

1. - Reading general data, such as; number of nodes, number of lines, base power, number of transformers and the slack node.

2. - Reading data of transmission lines, their impedance and shunt admittance.

3. - Reading data of generated and demanded power in each node, as well as its initial voltage and the type of node.

4. - Reading of transformer data.

5. - In block four the admittance matrix is formed.

6. - In block five the increases in real and imaginary part of the voltage are calculated and its value is corrected (eq. 2).

7. - Block six corresponds to the verification of the error allowed in the iterative process.

8. - If the tolerance is already met, the process is finished by printing voltages and if desired, the load flow is calculated.

9. - If the tolerance is not yet met, values are corrected and the iterative process is continued.

In the flowchart of fig. 2, the blocks where equations (31)-(34) are solved are added to fig. 1. The iterative process ends when the relative error is less than the selected tolerance. Finally, real and reactive power flow in transmission lines and transformers is calculated. In the rectangular form, the Voltage in the node is, corrected with the increments calculated in each iteration. The acceleration factors are applied to the increments of the rectangular components according to (35).

\[ V_i^k = e_i^k + \alpha \Delta e_i^k \alpha + i(f^k + \Delta f^k \alpha) \]  

Where:

\( \alpha \) It is the acceleration factor to the real component of the Voltage,

\( \alpha \) It is the acceleration factor to the imaginary component of the Voltage

A. Method M1

A distribution system of 40 nodes is simulated and magnitude of Voltage in pu values is shown in Fig. 3 [18]. The acceleration factors used in the simulations were: 1.0, 1.0, 1.5, 1.5, 1.8, 1.8. For the real and imaginary components, respectively. Table I shows the values calculated for the acceleration factors mentioned and are compared with those obtained in (19).

| Table I: Calculated voltages (pu) |
|----------------------------------|
| Node | \( V(1,1) \) | \( V(1,5,1,5) \) | \( V(1,8,1,8) \) | \( V(18) \) |
| 2    | 0.98892   | 0.98892   | 0.9889     | 0.988547  |
| 3    | 0.98058   | 0.98058   | 0.98055    | 0.979944  |
| 4    | 0.97416   | 0.97415   | 0.97412    | 0.973321  |
| 5    | 0.97199   | 0.97199   | 0.97196    | 0.971156  |
| 6    | 0.96992   | 0.96992   | 0.96989    | 0.969083  |
| 7    | 0.98192   | 0.98192   | 0.98191    | 0.981442  |
| 8    | 0.97603   | 0.97604   | 0.97603    | 0.975466  |
| 9    | 0.97226   | 0.97227   | 0.97225    | 0.971642  |
| 10   | 0.96944   | 0.96945   | 0.96943    | 0.968798  |
| 11   | 0.96744   | 0.96744   | 0.96743    | 0.966787  |
| 12   | 0.97797   | 0.97797   | 0.97795    | 0.977315  |
| 13   | 0.97456   | 0.97456   | 0.97453    | 0.973894  |

DOI: 10.35940/ijeat.B2529.129219

Published By:

Blue Eyes Intelligence Engineering & Sciences Publication
In Figure 3, the values in Table I are shown graphically.

Fig. 3. Voltages for the system of 40 nodes

Fig. 4 shows the angle of the Voltages of fig. 3.

Fig. 4. Angles for the 40-node system

Fig. 5 shows iterations number and execution time in milliseconds

Fig. 5. Iterations and execution time for the 40-node system

The 118-node test system of the IEEE was simulated and the magnitude of the Voltages of each node is shown in Fig. 6 when acceleration factors are; 1.0, 1.0, 1.5, 1.5 and 1.5, 1.8, in the real and imaginary components, respectively.

Fig. 6. Voltage magnitude for the 118 nodes system [20]

The number of iterations and execution time in milliseconds are shown in fig. 7.

Fig. 7. Execution time and iterations number

Fig. 8 shows the relative error of the magnitude of the Voltage in the system of 40 nodes between the M1 method and [19].

Fig. 8. Relative error in the 40 nodes system
Two Iterative Methods to Solve Nonlinear Equations of Load Flows

Fig. 9 shows the relative error of the Voltage angle in the 40 node system between the M1 method and the one reported in [19].

Fig. 10. Voltages obtained with M1, M2 methods and [19]

The angle of Voltages is shown in fig. 11.

Fig. 11. Voltages Angle for the network of 40 nodes

The IEEE test system of 118 nodes was simulated with the M1, M2 methods and the magnitude of the voltages is shown in Fig. 12 and compared with those reported in [20].

Fig. 12. Voltages for the system of 118 nodes

The execution time and the number of iterations are calculated in the simulation with the methods M1 and M2, they are shown in the bar diagram of fig. 13.

Fig. 13. Iterations and execution time for the 118 nodes system

B. Methods M1 and M2

Table II shows the values of Voltages calculated with the two methods and the one reported in [18], Table II shows the values of Voltages calculated with the two methods and the one reported in [19].

Table II: Voltages calculated with methods M1, M2 and [19]

| Node | $V_{pu}(M1)$ | $V_{pu}(M2)$ | $V_{pu}(18)$ |
|------|--------------|--------------|--------------|
| 1    | 0.9889       | 0.9863       | 0.988547     |
| 2    | 0.98055      | 0.9757       | 0.979949     |
| 3    | 0.97412      | 0.9674       | 0.973321     |
| 4    | 0.97196      | 0.9652       | 0.971156     |
| 5    | 0.96989      | 0.9632       | 0.969083     |
| 6    | 0.96919      | 0.9632       | 0.969142     |
| 7    | 0.97603      | 0.9733       | 0.975466     |
| 8    | 0.97225      | 0.9694       | 0.971642     |
| 9    | 0.96943      | 0.9666       | 0.968798     |
| 10   | 0.96743      | 0.9646       | 0.966787     |
| 11   | 0.97795      | 0.9731       | 0.977315     |
| 12   | 0.97453      | 0.9696       | 0.973004     |
| 13   | 0.97182      | 0.9651       | 0.971021     |
| 14   | 0.96759      | 0.9605       | 0.966719     |
| 15   | 0.96397      | 0.9556       | 0.963045     |
| 16   | 0.96195      | 0.9546       | 0.961022     |
| 17   | 0.96954      | 0.9628       | 0.968708     |
| 18   | 0.96681      | 0.9601       | 0.969593     |
| 19   | 0.96451      | 0.9577       | 0.963648     |
| 20   | 0.98027      | 0.9776       | 0.979798     |
| 21   | 0.97391      | 0.9712       | 0.973329     |
| 22   | 0.97277      | 0.972011     |
| 23   | 0.97083      | 0.968       | 0.966434     |
| 24   | 0.96729      | 0.9644       | 0.966643     |
| 25   | 0.97354      | 0.9704       | 0.974683     |
| 26   | 0.97321      | 0.9683       | 0.972542     |
| 27   | 0.97039      | 0.9636       | 0.969589     |
| 28   | 0.96587      | 0.9588       | 0.964992     |
| 29   | 0.95849      | 0.9508       | 0.957499     |
| 30   | 0.95469      | 0.9467       | 0.953644     |
| 31   | 0.95294      | 0.9454       | 0.951894     |
| 32   | 0.96681      | 0.96       | 0.965977     |
| 33   | 0.96508      | 0.9583       | 0.964226     |
| 34   | 0.97191      | 0.9691       | 0.971318     |
| 35   | 0.97364      | 0.9687       | 0.972971     |
| 36   | 0.95603      | 0.9483       | 0.955027     |
| 37   | 0.95205      | 0.9444       | 0.950886     |
| 38   | 0.95036      | 0.9423       | 0.94929      |
| 39   | 0.94897      | 0.9418       | 0.948792     |
| 40   | 0.94762      | 0.9411       | 0.948082     |

Fig. 10 shows the magnitude of the Voltages for 40 node system when carrying out the simulation with M1, M2 methods and those obtained in [19].
For the system of 118 nodes, the error percentage is calculated in the nodes where the difference is greater when carrying out the simulation with two methods, and is shown in Fig. 14.

Equation (36) represents the simplification used in this work and was applied for load nodes and voltage controlled nodes, considering that any node can be the reference. In the controlled Voltage nodes, reactive power is calculated and included in the iterative process like any other. This formulation eliminates the formation of the Jacobian matrix, which requires a significant amount of computational resources. The simplification used in this work has a high number of iterations, as seen in fig. 5, 7 and 13. It is observed in fig. 7, the effect of the acceleration factors with which you have an execution time of 249 milliseconds. The combination of the simplification of the rectangular version and the four-step method reduces the number of iterations and the execution time for the systems considered in this work. The programs were developed in FORTRAN with the free Force-2 software and with a 6 GB RAM computer. With the iterative methods used in this work, the voltage at each node is calculated. The power flow is a consequence of the voltages and their values are not reported.

IV. CONCLUSIONS

From the results obtained in this work, we have the following conclusions: 1) The proposed simplification requires only elementary operations, an analysis of the equations involved demonstrates this; 2) For the test systems used, the applied simplification provides acceptable results with the errors reported in the Voltage graphs; 3) The application of the hybrid method allows reducing the number of iterations and what is considered more important, the execution time, 4) With the two methods proposed, only \((N-J)\) non-linear equations are solved and one system of two equations; 5) Only the nodal admittance matrix is generated in the development of the programs. Has can be seen in Figure 10, the M1 method is more approximate than the M2 method. This may be because the acceleration factors are chosen at random and a sensitivity analysis could provide the appropriate values for each system. The authors continue to work with simple code numerical methods such as those proposed here, this in order to have simpler programming algorithms.

REFERENCES

1. Ward, J. B., and Hale, H. W., “Digital Computer Solution of Power Flow problems”, AIEEE Transactions on Power Apparatus and Systems, Vol.PAS-75, pp. 398-404, June 1956
2. Van Ness, J. E., and Griffin, J. H., “Elimination methods for flow studies”, Trans. IEEE Transactions on Power Apparatus and Systems, Vol. PAS-80, p. 299, June 1961.
3. Timney, W. H., and Hart, C. E., “Power flow solution by Newton's method”, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-86, No. 11, November 1967
4. Stott, B., and Alsac, O., “Fast decoupled load flow”, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-93, No. 3, May 1974
5. D. Villanueva, A. Feijoó, Jose, L.Pazos, “Probabilistic load flow considering correlation between generation, loads and wind power smart grid and renewable energy”, 2011, 2-12 doi:10.4236/sgre.2011.21002 Published Online February 2011.
6. K. Balaraman and Dipi Srivinvasan, “Review of power flow studies on distribution network with distributed generation” IEEE PEDS 2011, Singapore, 5-8 December 2011
7. T. Ramana, V. Ganesh, and S. Sivanagara, “Simple and fast load flow solution for electrical power distribution systems”, International Journal on Electrical Engineering and Informatics - Volume 5, Number 3, September 2013.
Two Iterative Methods to Solve Nonlinear Equations of Load Flows

Rubén Abiud Villafuerte Salcedo. He is a mechatronic engineer, graduated from the Popular Autonomous University of the State of Puebla in 2012. He completed his Master's studies in the area of biomedical engineering at the Popular Autonomous University of the State of Puebla in 2015. He is currently studying for a doctorate in Biomedical Engineering in the Popular Autonomous University of the State of Puebla and is a professor for hours assigned to the Faculty of Electrical Engineering at the Technological Institute of Mexico campus Orizaba, in Orizaba, Veracruz. He is a member of the institute of electrical and electronic engineers section Mexico from 2004 to date and member of the Mexican Network of biomaterials and organ and tissue engineering.

Email: dreyko999@gmail.com

Víctorino Juárez Rivera, received Eng. in Industrial Engineering in 2004 by the Instituto Tecnológico de Orizaba, his specialty line is in manufacturing, the MS degree in Mechanical Engineering in 2007 from Instituto Tecnológico de Veracruz, his research line is in materials testing he is a Full Professor in the Industrial Engineering Department, Facultad Ingeniería, Universidad Veracruzana. He is currently in charge of the Programa de Fortalecimiento a la Excelencia Educativa (PROFEXCE). Actually he is in PHD in Engineering by the Universidad Martí. His research interest includes; Six sigma, Statistical process control, Operational research and Ergonomic methods for assessing exposure to risk factor for work. Email: vjuarez@uv.mx

AUTHORS PROFILE

Rubén Villafuerte Díaz. He is an electrical industrial engineer, graduated from the Instituto Tecnológico de Morelia in 1979. He completed his master's and doctoral studies in the area of electrical engineering at the Instituto Politécnico Nacional, receiving the corresponding degrees in 1988 and 1993, respectively. He is currently a full-time professor at the Facultad Ingeniería de la Universidad Autónoma de Nuevo León, México, 2002

Email: ruben.v46@yahoo.com.mx, rvillafuerte@uv.mx

Jesús Medina Cervantes. Was born in Orizaba, Ver., México, in 1976. He received the graduate degree in Mechanical Engineering from the Instituto Tecnológico de Orizaba (ITO), Orizaba, Ver., México, in 1998, and the Master of Science degree in Mechanical Engineering from the Centro Nacional de Investigación y Desarrollo Tecnológico (CENIDET), Cuernavaca, Mor., México, in 2002. In 2001, he joined the Department of Textile Engineering, Universidad Emiliano Zapata, Cuernavaca, Morelia, México, as a Professor. In 2003, he joined the Department of Design Engineering, Universidad del Papaloapan, Loma Bonita, Oaxaca., México, as a Professor. Since February 2008, he has been with the Department of Mechanical Engineering, Universidad Veracruzana, Ciudad Mendoza, Ver., México, where he is a Professor. His current research interests include Mechanical and Mechatronics Design, Automation, IoT, Robotics and Control. Email: jmedina@uv.mx