Three Phase Asymmetrical Load Flow for Four-Wire Distribution Networks

M. Monfared*, A. M. Daryani. and M. Abedi
Department of Electrical Engineering, Amirkabir University of Technology
Hafez Ave., Tehran 15914, Iran
*Email: monfared81@yahoo.com

***Abstract*-- Majority of distribution networks is unbalanced due to unbalanced loads, asymmetry in transmission lines and two and single phase shunts. These characteristics of distribution systems results in deficiency of convenient methods in analysis of such cases. In this paper, we propose a successful method for solution of the four wire power flow problem. The method is based on the current injection technique and the Newton-Raphson formulation which has better numerical stability and convergence speed comparing with conventional methods. Furthermore, it is capable deal with the distributed generations as PV buses. It can also solve the sub-transmission loops as a part of solution of distribution system. Then, we suggest a modification to the proposed algorithm which drastically improves the convergence speed by reducing the number of the iterations and computational cost of each iteration. The method is applied to several cases and the results are presented.**

***Index Terms*--Distribution networks, Asymmetrical three phase four wire load flow, Current injection method**

I. Introduction

Load flow calculations provides amplitudes and the angles of all voltages and currents of the power system. Methods such as Gauss-Seidel, Newton and fast decoupled Newton-Raphson (in the polar coordinates for positive sequence networks) have been the basic tools for the load flow analysis for about 40 years. Most of the analyses have been focused on the balanced operation. However, the distribution networks are unbalanced and have low R/X ratios. Indeed, majority of distribution networks are unbalanced due to unbalanced loads, asymmetry in transmission lines and two and single phase shunts. These characteristics of distribution systems calls for effective power flow methods. A number of methods have been already developed for these problems. Recently, current injection equations have been used along with or instead of power mismatch equations. Garcia *et al* developed a new sparse formulation for unbalanced three phase power systems utilizing the Newton-Raphson method in rectangular coordinates [1]. This work claims that their method is very stable and the speed is approximately 30% better than the other existing methods. Additionally, it has the ability to represent the distributed generations as PV buses and solve the sub-transmission loops as a part of solution of the distribution system. In 2001, this method was improved to include FACTS devices and other kind of controllers [2]. Eventually, the technique presented in [1] was modified to include fourth wire in the formulation [3]. The recent report allows including the neutral wires and ground impedances in the formulations. This technique has better numerical stability and convergence speed comparing with other conventional methods [1]-[3]. Thus, there is enough motivation to focus on it for even more improvements. In this paper, we first demonstrate the technique and its implementation algorithm. Then, we suggest a modification to this algorithm which drastically improves the convergence speed by reducing number of iterations and computational cost of each iteration. Results obtained from several case studies of applying our modified load flow algorithm to unbalanced systems confirm this considerable improvement.

II. Four wire power flow technique based on the current injection method and the Newton-Raphson formulation

As previously stated, this paper is focused on improvement of a method originally developed by other researchers. For a full description of the technique as well as tracking progress of the method see references [1]-[3]. Here, we just give a straight algorithm to implement it and finally we will correct it to improve the convergence speed intensively by reducing the number of the iterations and the amount plus the time of the computations. In this method the current injection equations are written in rectangular coordinates resulting in an order 8n system of equations. The Jacobian matrix is composed of 8×8 blocks and retains the structure of the nodal admittance matrix which except for PV buses has the elements of the off diagonal blocks the same as those of the corresponding nodal admittance matrix. The elements of the diagonal blocks are updated at every iteration according to the load ZIP model parameters and the last calculated voltages for those nodes. The polynomial load model can represent constant impedance, constant current and constant power loads easily. In rectangular coordinates, complex values of voltages and currents are sum of the real and imaginary components. Nodal admittance matrix is definite and can be decomposed to G and B components:

$$V_k^d = V_{r_k}^d + jV_{m_k}^d$$
$$I_k^s = I_{r_k}^s + jI_{m_k}^s$$
$$Y = G + jB$$

The algorithm of the conventional load flow method which is based on the current injection technique and Newton-Raphson formulation can be summarized as follows:

Step 1: Initialization; for a flat start the initial voltages for all nodes should be equal to the root node voltage.
Step 2: Calculation of the current mismatches for all nodes from (2)-(5).
Step 3: Test for convergence; if $\max\{|\Delta I|\} < \varepsilon$ then go to step 7 else go to the next step.
Step 4: Compute the Jacobian matrix for (1) from (6) and (7). The off diagonal blocks (elements of the second part of (7)) are updated according to the load ZIP model parameters and the last calculated voltages for those nodes by using the appendix expressions.
Step 5: Calculate the complex voltage increments from (1).
Step 6: Update the complex voltages according to the calculated increments. Increment the iteration count and go to step 2.
Step 7: Print the results.

To represent the PV buses, their reactive power generation is considered as a new state variable. Refer to [3] and [4] for more details.

V_k^t : Complex voltage at node k, phase $t = \{a, b, c, n\}$
Ω_k : Set of nodes directly connected to node k
V_k^d : Complex voltage at node k, phase $d = \{a, b, c\}$
V_k^n : Neutral complex voltage at node k

$$\begin{bmatrix} \Delta I_{m_1}^{abcn} \\ \Delta I_{r_1}^{abcn} \\ \hline \Delta I_{m_2}^{abcn} \\ \Delta I_{r_2}^{abcn} \\ \hline \vdots \\ \hline \Delta I_{m_{nb}}^{abcn} \\ \Delta I_{r_{nb}}^{abcn} \end{bmatrix} = \begin{bmatrix} (J_{11})^{abcn} & (J_{12})^{abcn} & \cdots & (J_{1nb})^{abcn} \\ (J_{21})^{abcn} & (J_{22})^{abcn} & \cdots & (J_{2nb})^{abcn} \\ \vdots & \vdots & \vdots & \vdots \\ (J_{nb1})^{abcn} & (J_{nb2})^{abcn} & \cdots & (J_{nbnb})^{abcn} \end{bmatrix} \begin{bmatrix} \Delta V_{r_1}^{abcn} \\ \Delta V_{m_1}^{abcn} \\ \hline \Delta V_{r_2}^{abcn} \\ \Delta V_{m_2}^{abcn} \\ \hline \vdots \\ \hline \Delta V_{r_{nb}}^{abcn} \\ \Delta V_{m_{nb}}^{abcn} \end{bmatrix} \tag{1}$$

$$\begin{aligned} \Delta I_{r_k}^d = {} & \frac{(P_k^d)(V_{r_k}^d - V_{r_k}^n) + (Q_k^d)(V_{m_k}^d - V_{m_k}^n)}{\left((V_{r_k}^d - V_{r_k}^n)^2 + (V_{m_k}^d - V_{m_k}^n)^2\right)} + \\ & - \sum_{t \in \alpha_p} \left(G_{kk}^{dt} V_{r_k}^t - B_{kk}^{dt} V_{m_k}^t\right) - \sum_{\substack{i \in \Omega_k \\ i \neq k}} \sum_{t \in \alpha_p} \left(G_{ki}^{dt} V_{r_i}^t - B_{ki}^{dt} V_{m_i}^t\right) \end{aligned} \tag{2}$$

$$\begin{aligned} \Delta I_{m_k}^d = {} & \frac{(P_k^d)(V_{m_k}^d - V_{m_k}^n) + (Q_k^d)(V_{r_k}^d - V_{r_k}^n)}{\left((V_{r_k}^d - V_{r_k}^n)^2 + (V_{m_k}^d - V_{m_k}^n)^2\right)} + \\ & - \sum_{t \in \alpha_p} \left(B_{kk}^{dt} V_{r_k}^t + G_{kk}^{dt} V_{m_k}^t\right) - \sum_{\substack{i \in \Omega_k \\ i \neq k}} \sum_{t \in \alpha_p} \left(B_{ki}^{dt} V_{r_i}^t + G_{ki}^{dt} V_{m_i}^t\right) \end{aligned} \tag{3}$$

$$\begin{aligned} \Delta I_{r_k}^n = {} & -\left(I_{r_{k\,cg}}^a + I_{r_{k\,cg}}^b + I_{r_{k\,cg}}^c\right) + \\ & - \sum_{t \in \alpha_p} \left(G_{kk}^{nt} V_{r_k}^t - B_{kk}^{nt} V_{m_k}^t\right) - \sum_{\substack{i \in \Omega_k \\ i \neq k}} \sum_{t \in \alpha_p} \left(G_{ki}^{nt} V_{r_i}^t - B_{ki}^{nt} V_{m_i}^t\right) \end{aligned} \tag{4}$$

$$\begin{aligned} \Delta I_{m_k}^n = {} & -\left(I_{m_{k\,cg}}^a + I_{m_{k\,cg}}^b + I_{m_{k\,cg}}^c\right) + \\ & - \sum_{t \in \alpha_p} \left(B_{kk}^{nt} V_{r_k}^t + G_{kk}^{nt} V_{m_k}^t\right) - \sum_{\substack{i \in \Omega_k \\ i \neq k}} \sum_{t \in \alpha_p} \left(B_{ki}^{nt} V_{r_i}^t + G_{ki}^{nt} V_{m_i}^t\right) \end{aligned} \tag{5}$$

$$(J_{ki})^{abcn} = \begin{bmatrix} B_{ki}^{aa} & B_{ki}^{ab} & B_{ki}^{ac} & B_{ki}^{an} & G_{ki}^{aa} & G_{ki}^{ab} & G_{ki}^{ac} & G_{ki}^{an} \\ B_{ki}^{ba} & B_{ki}^{bb} & B_{ki}^{bc} & B_{ki}^{bn} & G_{ki}^{ba} & G_{ki}^{bb} & G_{ki}^{bc} & G_{ki}^{bn} \\ B_{ki}^{ca} & B_{ki}^{cb} & B_{ki}^{cc} & B_{ki}^{cn} & G_{ki}^{ca} & G_{ki}^{cb} & G_{ki}^{cc} & G_{ki}^{cn} \\ B_{ki}^{na} & B_{ki}^{nb} & B_{ki}^{nc} & B_{ki}^{nn} & G_{ki}^{na} & G_{ki}^{nb} & G_{ki}^{nc} & G_{ki}^{nn} \\ G_{ki}^{aa} & G_{ki}^{ab} & G_{ki}^{ac} & G_{ki}^{an} & -B_{ki}^{aa} & -B_{ki}^{ab} & -B_{ki}^{ac} & -B_{ki}^{an} \\ G_{ki}^{ba} & G_{ki}^{bb} & G_{ki}^{bc} & G_{ki}^{bn} & -B_{ki}^{ba} & -B_{ki}^{bb} & -B_{ki}^{bc} & -B_{ki}^{bn} \\ G_{ki}^{ca} & G_{ki}^{cb} & G_{ki}^{cc} & G_{ki}^{cn} & -B_{ki}^{ca} & -B_{ki}^{cb} & -B_{ki}^{cc} & -B_{ki}^{cn} \\ G_{ki}^{na} & G_{ki}^{nb} & G_{ki}^{nc} & G_{ki}^{nn} & -B_{ki}^{na} & -B_{ki}^{nb} & -B_{ki}^{nc} & -B_{ki}^{nn} \end{bmatrix} \tag{6}$$

$$\begin{aligned} (J_{kk})^{abcn} = {} & \begin{bmatrix} B_{kk}^{aa} & B_{kk}^{ab} & B_{kk}^{ac} & B_{kk}^{an} & G_{kk}^{aa} & G_{kk}^{ab} & G_{kk}^{ac} & G_{kk}^{an} \\ B_{kk}^{ba} & B_{kk}^{bb} & B_{kk}^{bc} & B_{kk}^{bn} & G_{kk}^{ba} & G_{kk}^{bb} & G_{kk}^{bc} & G_{kk}^{bn} \\ B_{kk}^{ca} & B_{ki}^{cb} & B_{kk}^{cc} & B_{kk}^{cn} & G_{kk}^{ca} & G_{kk}^{cb} & G_{kk}^{cc} & G_{kk}^{cn} \\ B_{kk}^{na} & B_{kk}^{nb} & B_{kk}^{nc} & B_{kk}^{nn} & G_{kk}^{na} & G_{kk}^{nb} & G_{kk}^{nc} & G_{kk}^{nn} \\ G_{kk}^{aa} & G_{kk}^{ab} & G_{ki}^{ac} & G_{ki}^{an} & -B_{kk}^{aa} & -B_{kk}^{ab} & -B_{kk}^{ac} & -B_{kk}^{an} \\ G_{kk}^{ba} & G_{kk}^{bb} & G_{kk}^{bc} & G_{kk}^{bn} & -B_{kk}^{ba} & -B_{kk}^{bb} & -B_{kk}^{bc} & -B_{kk}^{bn} \\ G_{kk}^{ca} & G_{kk}^{cb} & G_{kk}^{cc} & G_{kk}^{cn} & -B_{kk}^{ca} & -B_{kk}^{cb} & -B_{kk}^{cc} & -B_{kk}^{cn} \\ G_{kk}^{na} & G_{kk}^{nb} & G_{kk}^{nc} & G_{kk}^{nn} & -B_{kk}^{na} & -B_{kk}^{nb} & -B_{kk}^{nc} & -B_{kk}^{nn} \end{bmatrix} + \\ & + \begin{bmatrix} -e_k^a & & & e_k^a & -f_k^a & & & f_k^a \\ & -e_k^b & & e_k^b & & -f_k^b & & f_k^b \\ & & -e_k^c & e_k^c & & & -f_k^c & f_k^c \\ e_k^b & e_k^b & e_k^b & -e_k^a - e_k^b - e_k^c & f_k^a & f_k^b & f_k^c & -f_k^a - f_k^b - f_k^c \quad h_k^a \\ -g_k^a & & & g_k^a & -h_k^a & & & h_k^a \\ & -g_k^b & & g_k^b & & -h_k^b & & h_k^b \\ & & -g_k^c & g_k^c & & & -h_k^c & h_k^c \\ g_k^a & g_k^b & g_k^c & -g_k^a - g_k^b - g_k^c & h_k^a & h_k^b & h_k^c & -h_k^a - h_k^b - h_k^c \end{bmatrix} \end{aligned} \tag{7}$$

III. Modifying the Algorithm to Improve the Speed by Reducing the Number of Iterations and the Amount Plus the Time of the Computations

The Jacobian matrix, J, is sparse and retains the same structure as of the nodal admittance matrix. These advantages besides the utilizing the improved Newton-Raphson algorithm conduce to 30% speedup comparing with a state of the art production grade Newton-Raphson power flow [1]. On the other hand the elements of the diagonal blocks for all of the system's nodes must be updated at every iteration according to the load ZIP model parameters and the last calculated voltages for those nodes which calls for a large amount of computations and time. Our solution overcomes this disadvantage. Fortunately, reducing the amount of the calculations and the needed time for the load flow computations is not the only improvement. Furthermore, the proposed modification reduces the number of total iterations for each load flow problem which itself again reduces the computational costs even more. Our modification is that in every load flow problem after some initial iterations once the variation of current mismatches for all of the nodes settled in a specific band ($\max\{||\Delta I^{current}|-|\Delta I^{previous}||\}<\varepsilon'$), the Jacobian matrix won't be updated and the last updated Jacobian will be used till the end of the load flow i.e. until $\max\{|\Delta I|\}<\varepsilon$ occurs. In fact we have used rate of the changes of the current mismatches as a second error criterion and when it becomes less than a specific limit (ε'), the Jacobian matrix won't be updated. Figure (1) depicts algorithm of our modified load flow method.

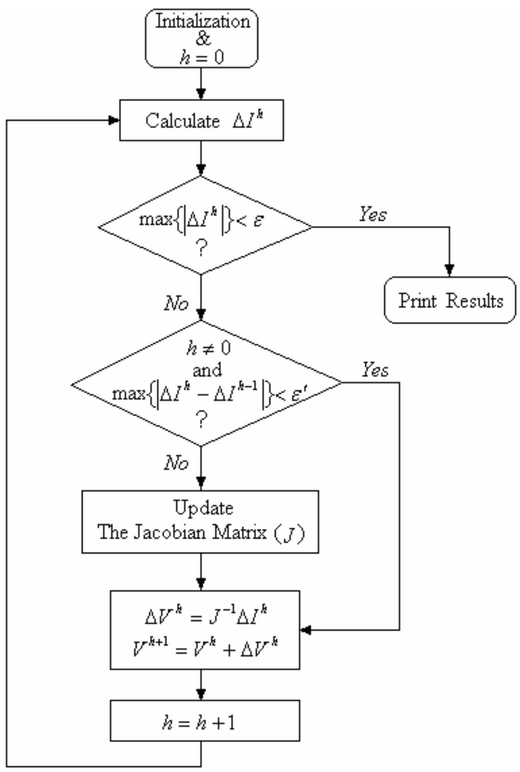

Fig. 1. Flowchart of the modified algorithm

This solution is applicable due to several reasons: First we should notice that by approaching to the results of the load flow problem the current mismatches (ΔI) decrease. In initial steps, this reduction of ΔI is very bigger than the final steps of the load flow. So the slope of these variations which is a measure of the speed of ΔI variations lessens rapidly and noticeably through the load flow steps forward so that this slope can be a proper check for approaching to the final results of the load flow problem. Then, when this slope becomes small, it implies that ΔI have become very small. Therefore, whit this small ΔI, the effect of the variations in the Jacobian on the results of $\Delta V=J^{-1}\Delta I$, (1), becomes small and can be neglected with no error for these very small current mismatches. In addition and as a result of moving toward the solution, the voltage variations decrease extremely. Thus, variations of the elements of the Jacobian matrix, which are dependent on these voltages, become negligible. So, not updating the Jacobian matrix after once entering a region near the final answers not only doesn't affect the results of the load flow but also reduces the amount of the calculations and the needed time for the load flow computations required for updating the Jacobian. Especially it's significant in dealing with large and complicated systems in which the process of updating the Jacobian is very time consuming. But it's not all; the proposed modification also reduces the number of the iterations needed to achieve the results of every load flow problem; consequent on it the computational costs reduce even more. This is because of the fact that by achieving to the final results, nodal voltages generally decrease till they take their real values which are often less than their initial values. Referring to the appendix we can find out that as a result of this reduction of voltages, the diagonal elements of the Jacobian, which are updated in every iteration according to these voltages, increase. This means that the J^{-1} generally decreases through the load flow steps forward. Now after some initial iterations and when slope of the error decreased to a particular bound (ε'), which occurs near the final answer of the load flow, the Jacobian approximates its final value accurately although it is generally a little smaller than its real value. Now if we take the Jacobian matrix constant, then the values calculated for voltages from $\Delta V=J^{-1}\Delta I$ are a small amount bigger than predictable and this lets us to achieve the final results in fewer iterations. This is a common idea in some conventional methods such as Gauss-Seidel as in this case the voltage increments are multiplied by a coefficient called accelerating factor which always exceeds unity [5]. To fulfill our idea, we define a second error limit called ε' and when the slope of the main error, current mismatches, settled in this limit, we won't update the Jacobian matrix henceforward. In this work once we have taken the second root of the main error limit, $\sqrt{\varepsilon}$, for ε' and once more the forth root of the main error limit, $\sqrt[4]{\varepsilon}$, for it.

IV. APPLING THE MODIFIED ALGORITHM TO TEST SYSTEMS AND COMPARING ITS EFFICIENCY WITH THE CONVENTIONAL METHOD

To illustrate the practicality of our proposed modification, we have applied both the conventional and modified algorithms to four different networks. The first is a simple test system presented in [3] for demonstration of the conventional methodology. Figure (2) describes this system. Ground impedances are unequal and the three phase load is unbalanced, additionally to increase the unbalances, the phase c of transmission line 2-3 was assumed open. Two other test networks of figures (3) and (4) are expansions of the first one related to real cases. These two networks are four wire systems and have been grounded at all nodes with unequal impedances. Single line diagram representation is just for simplicity. The loads are unbalanced and the phases c of line 2-3 and a of line 4-5 in 7 node system and the phases c of line 5-7, b of line 8-11 and a of line 14-15 in 15 node system are considered open. The forth test network is the IEEE 4 node test feeder [6]; in this case there aren't neutral wire and ground impedances and the unbalance is minimum and only due to the unbalanced loads. We have executed the algorithms which were implemented in a high level programming language on a 1.5 MHz PC. The number of total iterations, the number of the times of updating the Jacobian matrix and the total time of the load flow computations for these four test systems, in conventional and modified methods are summarized in table (1). We have applied the modified technique for two cases once with $\varepsilon' = \sqrt{\varepsilon}$ and once more with $\varepsilon' = \sqrt[4]{\varepsilon}$. Of course because of an equal error limit (ε) in both conventional and modified methods the load flow results are the same in this error band. Table (1) shows a great reduction in the amount of updating the Jacobian matrix in the modified method. Also it shows that the number of total iteration is decreased. Both of these have led to a significant saving of computational expenses so that the time of the computations has been reduced considerably by means of the proposed modification. The bigger value for ε' has better improvement. This value for the 7 and 15 node test systems has achieved an average 50% speedup.

TABLE I
COMPARISONS OF THE SPEED AND THE COMPUTATION TIME BETWEEN THE CONVENTIONAL [3] AND THE MODIFIED ALGORITHMS

	3 node test system			7 node test system			15 node test system			IEEE 4 node test feeder		
	$\varepsilon' = \sqrt[4]{\varepsilon}$	$\varepsilon' = \sqrt[2]{\varepsilon}$	Conv. method	$\varepsilon' = \sqrt[4]{\varepsilon}$	$\varepsilon' = \sqrt[2]{\varepsilon}$	Conv. method	$\varepsilon' = \sqrt[4]{\varepsilon}$	$\varepsilon' = \sqrt[2]{\varepsilon}$	Conv. method	$\varepsilon' = \sqrt[4]{\varepsilon}$	$\varepsilon' = \sqrt[2]{\varepsilon}$	Conv. method
Number of the total iterations	11	12	12	23	32	38	11	19	22	11	12	12
Number of the Jacobian updates	1	4	12	7	9	38	1	4	22	2	4	12
Time of the load flow (ms)	188	204	219	290	328	510	344	484	797	185	203	209

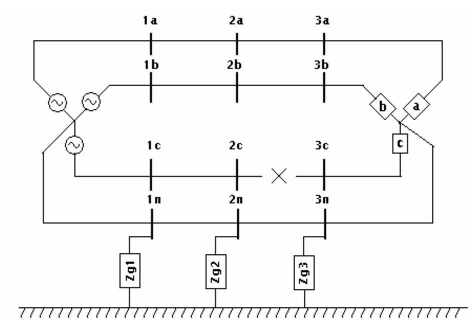

Fig. 2. 3-bus test system diagram

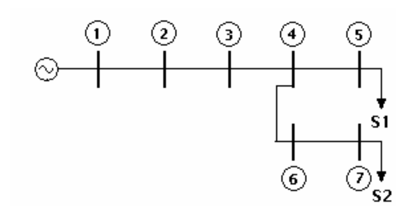

Fig. 3. 7-bus test system

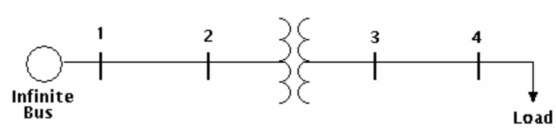

Fig. 5. IEEE 4-bus test feeder

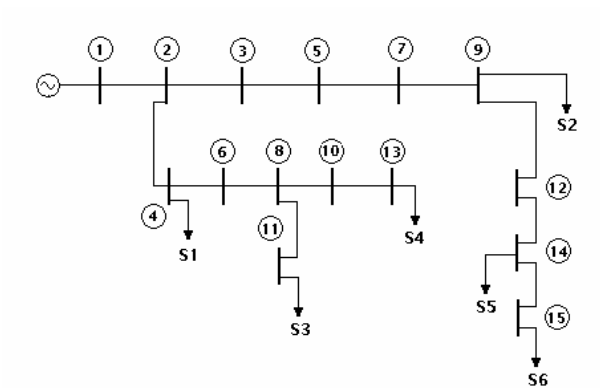

Fig. 4. 15-bus test system

V. CONCLUSIONS

In this paper, we propose a successful method for solution of the four wire power flow problem. The method is based on the current injection technique and the Newton-Raphson formulation which has better numerical stability and convergence speed comparing with conventional methods which use power mismatch equations. Furthermore, it is capable deal with the distributed generations as PV buses. It can also solve the sub-transmission loops as a part of solution of distribution system. In this method the neutral wires and ground impedances are considered separately. The other benefit is that the Jacobian matrix is sparse and retains the same structure as of the nodal admittance matrix. **A modification is proposed for the algorithm which drastically improves the convergence speed by reducing the number of the iterations and computational cost of each iteration.** Results obtained from several case studies of applying our modified load flow algorithm to unbalanced systems confirm this considerable improvement.

VI. APPENDIX

The elements of the second part of (7) are calculated according to the load ZIP model parameters and the last calculated voltage for each node:

$$e_k^d = \frac{\left(Q_{0k}^d\right)\left(V_{r_k}^d - V_{r_k}^n\right)^2 - \left(Q_{0k}^d\right)\left(V_{m_k}^d - V_{m_k}^n\right)^2 - 2\left(P_{0k}^d\right)\left(V_{m_k}^d - V_{m_k}^n\right)\left(V_{r_k}^d - V_{r_k}^n\right)}{\left[\left(V_{r_k}^d - V_{r_k}^n\right)^2 + \left(V_{m_k}^d - V_{m_k}^n\right)^2\right]^2} + \\ + \frac{\left(P_{1k}^d\right)\left(V_{r_k}^d - V_{r_k}^n\right)\left(V_{m_k}^d - V_{m_k}^n\right) + \left(Q_{1k}^d\right)\left(V_{m_k}^d - V_{m_k}^n\right)^2}{\left[\left(V_{r_k}^d - V_{r_k}^n\right)^2 + \left(V_{m_k}^d - V_{m_k}^n\right)^2\right]^{\frac{3}{2}}} + Q_{2k}^d$$

$$f_k^d = \frac{\left(P_{0k}^d\right)\left(V_{r_k}^d - V_{r_k}^n\right)^2 - \left(P_{0k}^d\right)\left(V_{m_k}^d - V_{m_k}^n\right)^2 + 2\left(Q_{0k}^d\right)\left(V_{r_k}^d - V_{r_k}^n\right)\left(V_{m_k}^d - V_{m_k}^n\right)}{\left[\left(V_{r_k}^d - V_{r_k}^n\right)^2 + \left(V_{m_k}^d - V_{m_k}^n\right)^2\right]^2} + \\ + \frac{\left(Q_{1k}^d\right)\left(V_{r_k}^d - V_{r_k}^n\right)\left(V_{m_k}^d - V_{m_k}^n\right) + \left(P_{1k}^d\right)\left(V_{r_k}^d - V_{r_k}^n\right)^2}{\left[\left(V_{r_k}^d - V_{r_k}^n\right)^2 + \left(V_{m_k}^d - V_{m_k}^n\right)^2\right]^{\frac{3}{2}}} - P_{2k}^d$$

$$g_k^d = \frac{-\left(P_{0k}^d\right)\left(V_{r_k}^d - V_{r_k}^n\right)^2 + \left(P_{0k}^d\right)\left(V_{m_k}^d - V_{m_k}^n\right)^2 - 2\left(Q_{0k}^d\right)\left(V_{m_k}^d - V_{m_k}^n\right)\left(V_{r_k}^d - V_{r_k}^n\right)}{\left[\left(V_{r_k}^d - V_{r_k}^n\right)^2 + \left(V_{m_k}^d - V_{m_k}^n\right)^2\right]^2} + \\ + \frac{\left(Q_{1k}^d\right)\left(V_{r_k}^d - V_{r_k}^n\right)\left(V_{m_k}^d - V_{m_k}^n\right) - \left(P_{1k}^d\right)\left(V_{m_k}^d - V_{m_k}^n\right)^2}{\left[\left(V_{r_k}^d - V_{r_k}^n\right)^2 + \left(V_{m_k}^d - V_{m_k}^n\right)^2\right]^{\frac{3}{2}}} - P_{2k}^d$$

$$h_k^d = \frac{\left(Q_{0k}^d\right)\left(V_{r_k}^d - V_{r_k}^n\right)^2 - \left(Q_{0k}^d\right)\left(V_{m_k}^d - V_{m_k}^n\right)^2 - 2\left(P_{0k}^d\right)\left(V_{r_k}^d - V_{r_k}^n\right)\left(V_{m_k}^d - V_{m_k}^n\right)}{\left[\left(V_{r_k}^d - V_{r_k}^n\right)^2 + \left(V_{m_k}^d - V_{m_k}^n\right)^2\right]^2} + \\ + \frac{\left(P_{1k}^d\right)\left(V_{r_k}^d - V_{r_k}^n\right)\left(V_{m_k}^d - V_{m_k}^n\right) - \left(Q_{1k}^d\right)\left(V_{r_k}^d - V_{r_k}^n\right)^2}{\left[\left(V_{r_k}^d - V_{r_k}^n\right)^2 + \left(V_{m_k}^d - V_{m_k}^n\right)^2\right]^{\frac{3}{2}}} - Q_{2k}^d$$

VII. REFERENCES

[1] P. A. N. Garcia, J. L. R. Pereira, J. Carneiro, V. M. Costa, and N. Martins, "Three-phase power flow calculations using the current injection method", *IEEE Trans. Power systems*, vol. 15, no. 2, pp. 508-514, May 2000.

[2] V. M. Costa, J. L. R. Pereira, and N. Martins, "An augmented Newton-Raphson power flow formulation based on current injections", *Electrical power and energy systems*, vol. 23, pp. 305-312, 2001.

[3] D. R. R. Penido, L. R. Araujo, J. L. R. Pereira, P. A. N. Garcia, and J. Carneiro, "Four wire Newton-Raphson power flow based on the current injection method", in *proc. 2004 IEEE Power Engineering Society Power systems conference and exposition*, vol. 1, pp. 239-242.

[4] P. A. N. Garcia, J. L. R. Pereira, J. Carneiro, M. P. Vinagre, and F. V. Gomes, "Improvements in the Representation of PV Buses on Three-Phase Distribution Power Flow*", IEEE Trans. Power Delivery,* vol. 19, no. 2, pp. 894-896, April 2004.

[5] I. J. Nagrath, and D. P. Kothari, *Modern Power System Analysis*, McGraw-Hill, 1983.

[6] *IEEE 4 node test feeder*, IEEE Power Engineering Society, Distribution System Analysis Subcommittee. [Online]. Available: http://