Reverse Load Flow Analysis

A Thesis submitted to the

Dept. of Electrical & Electronic Engineering, BRAC University in partial fulfillment of the requirements for the Bachelor of Science degree in Electrical & Electronic Engineering

By

MD. IQBAL HOSSAN ASIF

MD. SOBHANUL AZIM

SALWA MAHBUB

BRAC UNIVERSITY

December 2013

Declaration

We do hereby declare that the thesis titled “Reverse Load Flow Analysis” is submitted to the Department of Electrical and Electronics Engineering of BRAC University in partial fulfillment of the Bachelor of Science in Electrical and Electronics Engineering. This is our original work and has not been submitted elsewhere for the award of any other degree or diploma.

MD. IQBAL HOSSAN ASIF
Student ID: 10 12 10 61

MD. SOBHANUL AZIM
Student ID: 09 22 10 13

SALWA MAHBUB
Student ID: 09 22 12 24

Countersigned:

Prof. Dr. S. Shahnawaz Ahmed
Department of Electrical and Electronic Engineering
Bangladesh University of Engineering and Technology
(Thesis Supervisor)

Acknowledgments

The authors are extremely grateful to their Supervisor, **Prof. Dr. S. Shahnawaz Ahmed** for his ideas, advice, guidance and assistance. Without his continuous support the thesis work would not have been possible.

Abstract

Uninterrupted supply of electricity is a key to the growth of a modern civilization. Over the years an interconnected high voltage grid system has proven to be the most viable and reliable source of electrical energy. One of the requirements for secure operation of such a system is that its lines should not be overloaded and buses should have a standard voltage level maintained. This thesis deals with a new approach to its solution.

Instead of the conventional load flow analysis i.e. solving for unknown bus voltages and line flows using a specified set of generators' real power (MW) output and loads, a reverse solution is proposed. Here from a specified set of loads, line flows within their thermal loading limits (MW) and the bus voltage magnitudes (close to 1.0 per unit) the unknown bus phase angles and then using these generators' outputs (MW) are determined. This results in compliance with line flow limits and avoidance of under voltage at buses. In the initial stage of such a work and within the available timeframe the proposed method has been tested on a four bus power system and compared against the results from a conventional load flow analysis using Gauss-Seidel algorithm.

Table of Contents

Chapter 1

Introduction

1.1 Background

Power flow or load flow analysis [1-4] is a vital part for planning, designing and real time operation of power systems. In operational stage the demand (loads) changes frequently and the operators have to continuously or at regular intervals adjust generation that should be equal to demand plus transmission loss. Sometimes considering a number of contingencies one at a time (such as anticipated loss of a generator or a line) the generation needs to be rescheduled in advance what is known as security analysis [1].

1.2 Review of Previous Work

A review of literature [1-7] shows that the conventional load flow analysis has been used for the operational performance evaluation in a transmission or a distribution system and the security analysis. To save computational time in security analysis a full AC load flow analysis is done for a short list of contingencies or a DC load flow analysis is used. All these methods perform analysis in a forward fashion i.e. unknown bus voltages and line flows are determined using a specified set of generators' real power (MW) output and loads. However, in doing so the line flow limits are more likely to exceed their thermal limits and bus voltages may fall below the standard limit. So to reschedule generations quickly without exceeding line flow limits and causing under voltage when the number of contingencies to be evaluated becomes high, a fast approach is necessary.

1.3 Objective and Scope

The objective of this thesis is to propose a new and faster approach for scheduling generators in security analysis so as to ensure that the lines of a power system are not overloaded and a standard voltage is maintained at the buses i.e. the power system can be operated safely. For this the unknown bus phase angles and then using those generator power outputs are determined in a reverse fashion from a specified set of bus voltage magnitudes (close to 1.0 per unit), loads and line real power flows. Results from the proposed method are then compared against those from the Gauss-Seidel load flow analysis applied on a four bus system in the initial stage of such a work the first of its kind and within the timeframe available for this thesis.

Chapter 2

Conventional Load Flow Analysis

2.1 Power Injection at a Bus

At a bus there can be two types of power injection, specified (scheduled) power injection and calculated power injection. Scheduled real power injected is found using the equation:

$$P_{i,sch} = P_{gi} - P_{di} \tag{2.1}$$

Where $P_{i,sch}$ is the power being injected into the network at bus i and P_{gi} is the power being generated at bus i in Fig. 1.2(a). P_{di} is the scheduled power demanded by the load at bus i.

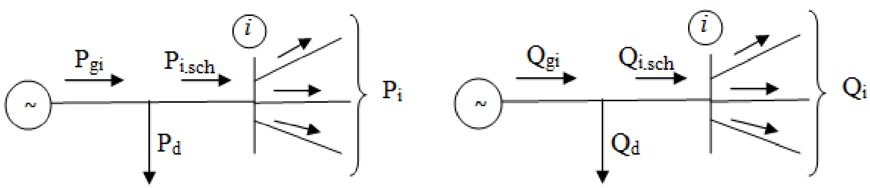

Fig 2.1 (a) Real and (b) reactive power at bus i.

The calculated value of power, P_i leads to the definition of mismatch ΔP_i,

$$\Delta P_i = P_{i,sch} - P_i = (P_{gi} - P_{di}) - P_i \tag{2.2}$$

Similarly, for reactive power at bus i,

$$\Delta Q_i = Q_{i,sch} - Q_i = (Q_{gi} - Q_{di}) - Q_i \tag{2.3}$$

To find the value of both calculated real and reactive power injection the following equations are used.

$$P_i = \sum_{n=1}^{N} |Y_{in} V_i V_n| \cos(\theta_{in} + \delta_n - \delta_i) \tag{2.4}$$

$$Q_i = -\sum_{n=1}^{N} |Y_{in} V_i V_n| \sin(\theta_{in} + \delta_n - \delta_i) \tag{2.5}$$

2.2 Bus Classification

Buses are the nodes of a system other than the reference (slack) node. One or many lines, generators and loads are connected to each bus. Each bus is associated with four unknown quantities:

- Real power, P_i
- Reactive power, Q_i
- Voltage magnitude, $|V_i|$
- Voltage angle, δ_i

From these four quantities, two quantities are specified for each bus and the other two are to be determined by solving equations. Based on the quantities known, the buses are classified [1] into three types.

1. **Load buses**: Usually these buses do not have a generator connected to them hence, P_{gi} and Q_{gi} are zero. Real power, P_{di} and reactive power, Q_{di} drawn from the system by the load are pre-specified. The two unknown quantities that have to be calculated are voltage magnitude, $|V_i|$ and voltage angle, δ_i. Load buses are often called P-Q bus as scheduled real and reactive powers $P_{i,sch} = - P_{di}$ and $Q_{i,sch} = - Q_{di}$ are known and mismatches ΔP_i and ΔQ_i can be defined.

2. **Voltage-controlled buses**: These are the buses where there are generators connected and voltage magnitude is kept constant. Voltage magnitude and generation real power, P_{gi} are pre-specified. Real power drawn from the system by the load, P_{di} is known so real power mismatch ΔP_i can be defined. Generation reactive power Q_{gi} required to support $|V_i|$ cannot be known in advance, so reactive power mismatch ΔQ_i cannot be defined. The two unknown quantities to be determined are voltage angle and reactive power. Voltage-controlled buses are also known as P-V bus.
3. **Slack bus**: There is only one slack bus in a system and it has a generator connected. The voltage angle of the slack bus serves as the reference angle for the angles of all the other buses. Voltage magnitude $|V_i|$ is pre-specified along with the voltage angle of the bus. The voltage angle of the bus is usually set as zero. Mismatches for the slack bus are not

defined. The two quantities which have to be calculated are the real and reactive powers. The slack bus generation caters to the balance of generation needed plus the transmission loss.

Table 2.1: Summary of Bus Types.

Bus Type	**No. of Buses**	**Quantities Specified**	**No. of available mismatch equations**	**No. of δ_i, $\lvert V_i \rvert$ state variables**
Slack (i=1)	1	δ_i, $\lvert V_i \rvert$	0	0
Voltage Controlled ($i=2,\ldots,N_g+1$)	N_g	P_i, $\lvert V_i \rvert$	N_g	N_g
Load ($i=N_g+2,\ldots,N$)	$N-N_g-1$	P_i, Q_i	$2(N-N_g-1)$	$2(N-N_g-1)$
Total	N	2N	$2N-N_g-2$	$2N-N_g-2$

Equations (2.4) and (2.5) shows that the powers are nonlinear functions of the state variables δ_i and $|V_i|$. Hence, load flow analysis usually use iterative techniques such as Gauss-Seidel and Newton-Raphson methods. The Newton-Raphson method solves the load flow equations all at a time in polar coordinates until ΔP and ΔQ mismatches at each bus in any iteration does not exceed a specified tolerance. The Gauss-Seidel method solves the load flow equations one after another in Cartesian coordinates until the difference for each bus voltage between two successive iterations is not more than a tolerance margin. Both methods use bus admittance matrix elements and eventually lead to the same solution on convergence. However, Gauss-Seidel method though needs more iterations is simple, free from matrix inversion requirements and suitable for small power systems.

2.3 The Gauss-Seidel Method

The Gauss-Seidel method of load flow analysis is an iterative method that solves the set of non-linear mismatch power equations for the buses with unknown voltage magnitude and/or phase angle in Cartesian coordinates sequentially, until the changes in bus voltages from one iteration to another are less than a specified value.

Initial estimates of (in o-th iteration) the phase angles of all load buses and voltage controlled buses are made as 0^0. The voltage magnitudes at slack and voltage controlled buses are set at the specified values. The voltage magnitude of the load buses are set in the initial iteration as 1.0 p.u. Such an initial estimate is known as the flat start.

The basic mismatch equation to be solved in Cartesian coordinates is as follows.

$\Delta P_i + j\Delta Q_i = 0$

Or, $P_{i,sch} - P_i + j(Q_{i,sch} - Q_i) = 0$

Or, $P_{i,sch} + jQ_{i,sch} = P_i + jQ_i$

Or, $P_{i,sch} - jQ_{i,sch} = P_i - jQ_i$

Or, $P_{i,sch} - jQ_{i,sch} = V_i^* \sum_{n=1}^{N} Y_{in} V_n$ (2.6)

Using the available line data such as series impedances and the line charging susceptances the bus admittance matrix [Y] is formed.

For a P-Q bus with i as its serial number in a N-bus system where the slack bus is designated as number 1 for convenience, the load flow equation is derived as follows.

Rearranging (2.6) gives;

$$\frac{P_{i,sch} - Q_{i,sch}}{V_i^*} = V_i^* \sum_{n=1}^{N} Y_{in} V_n \qquad (2.7)$$

Solution for V_i in complex (Cartesian) form in iteration k is

$$V_i^{(k)} = \frac{1}{Y_{ii}}\left[\frac{P_{i,sch} - Q_{i,sch}}{V_i^{(k-1)*}} - \sum_{j=1}^{i-1} Y_{ij}V_j^{(k)} - \sum_{j=i+1}^{N} Y_{ij}V_j^{(k-1)}\right] \quad (2.8)$$

This equation only applies for load buses where real and reactive powers are specified. To reduce the number of iterations required, the obtained voltage is multiplied by a constant known as the acceleration factor as follows. This value is then used as the i-th bus voltage.

$$V_{i,acc}^{(k)} = (1-\alpha)V_{i,acc}^{(k-1)} + \alpha V_i^{(k)} = V_{i,acc}^{(k-1)} + \alpha(V_i^{(k)} - V_{i,acc}^{(k-1)}) \quad (2.9)$$

If i-th bus is a P-V bus (i.e. voltage-controlled bus)where voltage magnitude instead of reactive power is specified, the reactive power must be computed using;

$$Q_i = -\mathrm{Im}\left\{V_i^* \sum_{j=1}^{N} Y_{ij}V_j\right\}$$

$$Q_i^{(k)} = -\mathrm{Im}\left\{V_i^{(k-1)^*}\left[\sum_{j=1}^{i-1} Y_{ij}V_j^{(k)} + \sum_{j=i}^{N} Y_{ij}V_j^{(k-1)}\right]\right\} \quad (2.10)$$

By substituting the value of $Q_i^{(k)}$ for $Q_{i,sch}$ in (2.8) the voltage of the P-V bus can be determined in k-th iteration. Since $|V_i|$ is specified for the voltage-controlled bus, the magnitude of $V_i^{(k)}$ obtained from (2.8) is corrected as follows;

$$V_{i,corr}^{(k)} = |V_i|\frac{V_i^{(k)}}{V_i^{(k)}} \quad (2.11)$$

This value is then used in the next steps of the iteration.

The entire process is repeated until the difference in the complex values of voltage at every bus between two successive iterations is less than the pre-specified tolerance margin i.e. (abs $[V_i^{(k+1)} - V_i^{(k)}]) \leq \varepsilon$.

At the end of convergence the obtained bus voltage magnitudes are used to compute line flows using equations (2.12) and (2.13) respectively for real and reactive power flows.

$$P_{in} = -|V_i|^2 G_{in} + |Y_{in}V_iV_n|\cos(\theta_{in} + \delta_n - \delta_i) \quad (2.12)$$

$$Q_{in} = -\left\{|V_i|^2\left(\frac{B'_{in}}{2} - B_{in}\right) + |V_iV_nY_{in}|\sin(\theta_{in} + \delta_n - \delta_i)\right\} \quad (2.13)$$

Chapter 3

Proposed Reverse Load Flow Analysis

3.1 Methodology

In the proposed reverse load flow analysis, the line flows and bus voltage magnitudes are specified at desired values for a given set of loads. Then the unknown phase angles and generators' real power outputs are determined so that the pre-specified conditions are fulfilled.

The following justified features of a power system are used in developing the mathematical model of the proposed method.

1. All bus voltage magnitudes are set at 1.0 per unit. However, the slack and voltage controlled bus voltage magnitudes can be set at their desired values.
2. Line resistances are neglected in forming bus admittance matrix [Y], as the value of resistance is very small compared to the value of series reactance.
3. The difference in phase angle across a line i.e. between two buses connected by a line is assumed to be very small so that the sine of the difference becomes equal to the difference in the angles being expressed in radians.

Therefore, a line's admittance becomes the same as its susceptance.

$$Y_{in} = \frac{1}{R_{in}+jX_{in}} \;;\quad (R_{in} \ll X_{in})$$

$$Y_{in} \cong \frac{1}{jX_{in}}$$

$$|Y_{in}| \cong B_{in};\left(B_{in} = \frac{1}{X_{in}}, \theta_{in} = 90°\right) \tag{3.1}$$

$$\sin(\delta_n - \delta_i) \approx (\delta_n - \delta_i) \tag{3.2}$$

Due to neglecting series resistance R_{in} , the conductance G_{in} is zero and the generalized equation for the calculation of real power flow in a line i-n is then

$$P_{in} = |Y_{in}V_iV_n| \cos(\theta_{in} + \delta_n - \delta_i)$$

$=|Y_{in}||V_i||V_n|\cos(90° + \delta_n - \delta_i)$;($Y_{in} \cong \frac{1}{jX_{in}}$; θ_{in}=90°)

$=|B_{in}||V_i||V_n|\{-\sin(\delta_n - \delta_i)\}$

$=|B_{in}||V_i||V_n|(\delta_i - \delta_n)$; $(\delta_n - \delta_i)$ is very small

Or, $P_{in} = (B_{in}|V_i||V_n|\delta_i - B_{in}|V_i||V_n|\delta_n)$ (3.3)

Equation (3.3) when applied for all the lines L (L=L1,L2….N_L) then the line flows can be arranged in a matrix equation as follows.

$$\begin{bmatrix} P_{l1} \\ : \\ P_{Nl} \end{bmatrix} = [B] \begin{bmatrix} \delta_2 \\ : \\ \delta_N \end{bmatrix} \quad (3.4)$$

Or, [P] = [B][δ] (3.5)

Where, δ_1 = 0 for the slack bus (numbered as 1),

Line flow vector [P] comprising P_{in} values is a N_Lx1 matrix with N_L = number of lines;

Phase angle vector [δ] comprising δ_i, (i=2,…N) is a (N-1)x1 matrix;

[B] is a N_Lx(N-1) matrix comprising products of line series susceptance elements B_{in} and specified voltage magnitudes $|V_i||V_n|$.

Pre-multiplying both sides of (3.5) by the transpose of [B] yields not only a square matrix but also caters to an overdetermined set of lines in case the number of lines is more than the number of buses in a power system (i.e. N_L> N) as follows.

$[B]^T[P] = [B]^T[B][\delta]$ (3.6)

Equation (3.6) can be solved for [δ] vector as follows.

$[\delta] = ([B]^T [B])^{-1} [B]^T [P]$ (3.7)

It is evident that the proposed reverse load flow analysis is non-iterative.

3.2 Generator Output Power Determination

The real power output of the generator connected to the bus i (i implies any bus with a generator) is calculated using the following equation.

$P_{gi} = P_i + P_{loadi}$

$$= \sum_{n=1}^{N} |Y_{in} V_i V_n| \cos(\theta_{in} + \delta_n - \delta_i) + P_{loadi} \quad (3.8)$$

Where local load (at bus i) P_{loadi}, V_i , V_n are pre-specified, $Y_{in}= B_{in}$, $\theta_{in} = 90°$ and δ_i, δ_n are obtained from (3.7) .

3.3 Flowchart

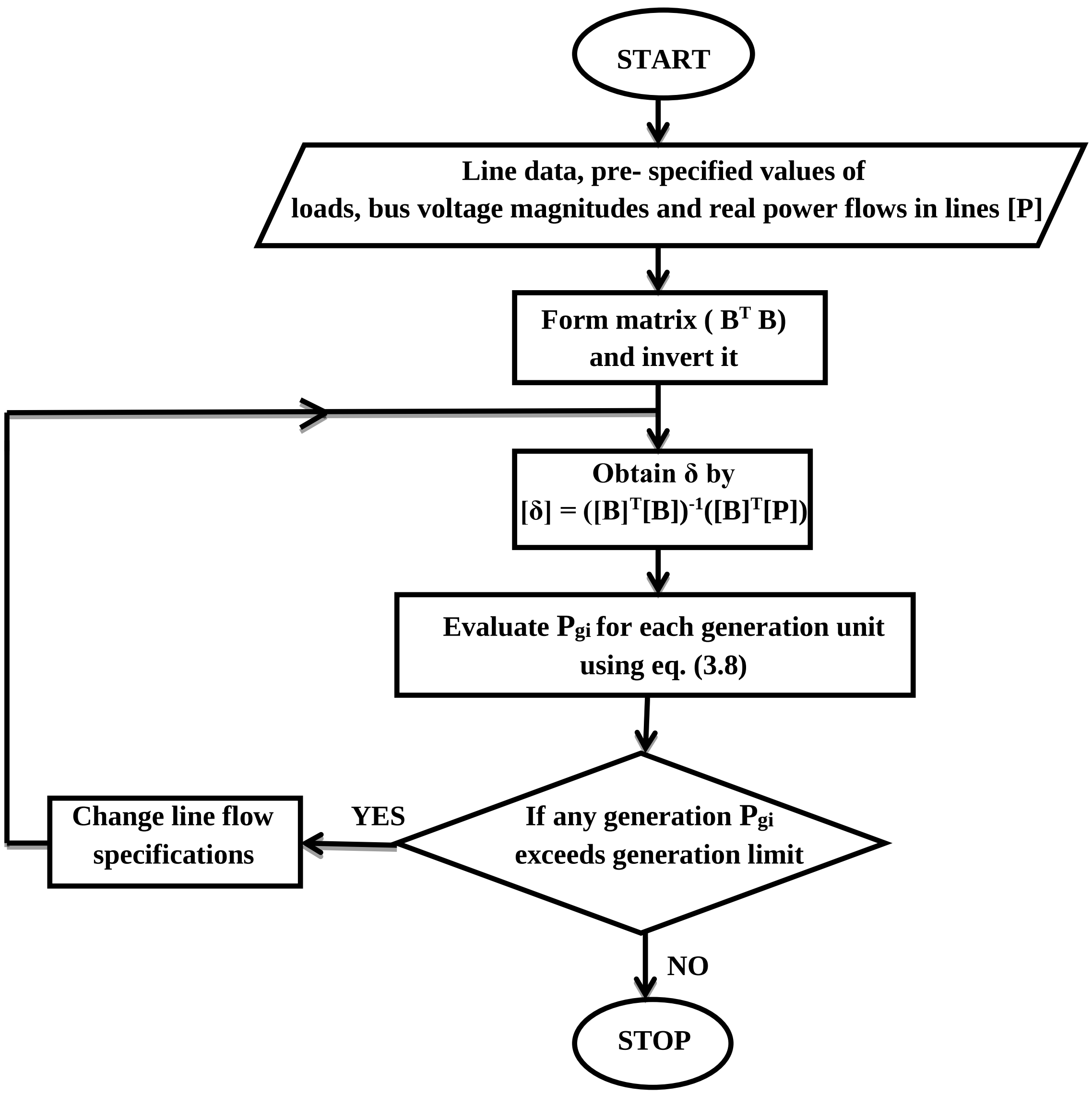

Fig 3.1 Flow chart of the proposed method.

Chapter 4

Results and Discussions

4.1 Test System model

A four bus system [1] shown in Fig. 4.1 is used to compare the performance of the proposed reverse load flow analysis method with that of a conventional load flow analysis (Gauss-Seidel) method. The line and load data are given in Tables 4.1 and 4.2. MATLAB codes were developed for the Gauss-Seidel conventional load flow analysis and the proposed method.

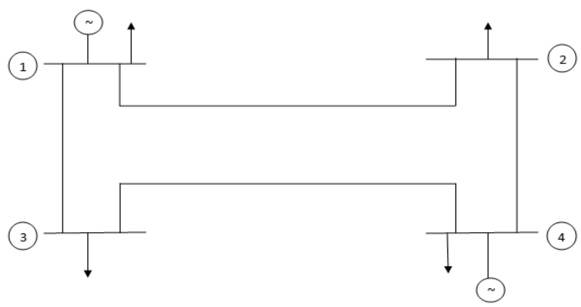

Fig 4.1: One-line diagram of a four bus power system.

Table 4.1: Line data for the four bus system

<table>
<tr><th rowspan="2">Line, (from bus to bus)</th><th colspan="2">Series Z</th><th colspan="2">Series Y = Z^{-1}</th><th colspan="2">Shunt charging susceptance Y’</th><th rowspan="6">Base : 100MVA,230 kV

Total charging MVAR is given at 230 kV</th></tr>
<tr><th>R (per unit)</th><th>X (per unit)</th><th>G (per unit)</th><th>B (per unit)</th><th>Total Charging (MVAR)</th><th>Y/2 (per unit)</th></tr>
<tr><td>1 – 2</td><td>0.01008</td><td>0.05040</td><td>3.815629</td><td>-19.078144</td><td>10.25</td><td>0.05125</td></tr>
<tr><td>1 – 3</td><td>0.00744</td><td>0.03720</td><td>5.169561</td><td>-25.847809</td><td>7.75</td><td>0.03875</td></tr>
<tr><td>2 – 4</td><td>0.00744</td><td>0.03720</td><td>5.169561</td><td>-25.847809</td><td>7.75</td><td>0.03875</td></tr>
<tr><td>3 – 4</td><td>0.01272</td><td>0.06360</td><td>3.023705</td><td>-15.118528</td><td>12.75</td><td>0.06375</td></tr>
</table>

Table 4.2: Pre-specified load data

<table>
<tr><th rowspan="2">Bus</th><th colspan="2">Load</th><th colspan="2"></th><td rowspan="6">Q values of load calculated from corresponding P values assuming a power factor of 0.85</td></tr>
<tr><th>P
(MW)</th><th>Q
(MVAR)</th><th>V
(per unit)</th><th>Remarks</th></tr>
<tr><td>1</td><td>50</td><td>30.99</td><td>1.00‹0^{θ}</td><td>Slack bus</td></tr>
<tr><td>2</td><td>170</td><td>105.35</td><td>1.00‹0^{θ}</td><td>Load bus (inductive)</td></tr>
<tr><td>3</td><td>200</td><td>123.94</td><td>1.00‹0^{θ}</td><td>Load bus (inductive)</td></tr>
<tr><td>4</td><td>80</td><td>49.58</td><td>1.02‹0^{θ}</td><td>Voltage controlled</td></tr>
</table>

4.2 Results from Proposed Method for Various Cases

4.2.1 Illustration of a Validation Case

The test system with the load data in Table 4.2 was solved [1] using the conventional load flow analysis by Gauss-Seidel algorithm and specifying the real power of generator at bus 4 as 318 MW. Table 4.3 shows the results for that.

Table 4.3 Results from Gauss-Seidel Conventional Load Flow Method

Bus Information						Line flow (only real part shown) calculated	
			Generation (only real part shown)	Load (only real part shown)	Bus Type		
Bus no.	Volts (p.u)	Angle (degrees)	P_g (MW)	P_{Load} (MW)		To Bus	(MW)
1	1.000	0	186.81	50	SL	2 3	38.69 98.12
2	0.982	-0.976	0	170	PQ	1 4	-38.46 -131.54
3	0.969	-1.872	0	200	PQ	1 4	-97.09 -102.91
4	1.020	1.523	318	80	PV	2 3	133.25 104.75
Total			504.81	500.00			

To validate the proposed method the line flows (real power in MW) were pre-specified close to those obtained from the Gauss-Seidel method as shown in Table 4.3. The directions of the line flows were maintained the same.

Fig. 4.2 shows the pre-specified values and directions of the line flows for the proposed method. The voltage magnitudes were specified 1.0 per unit (pu) for each of buses 1,2,3 and 1.02 pu for bus 4. The same loads as in Table 4.2 were specified.

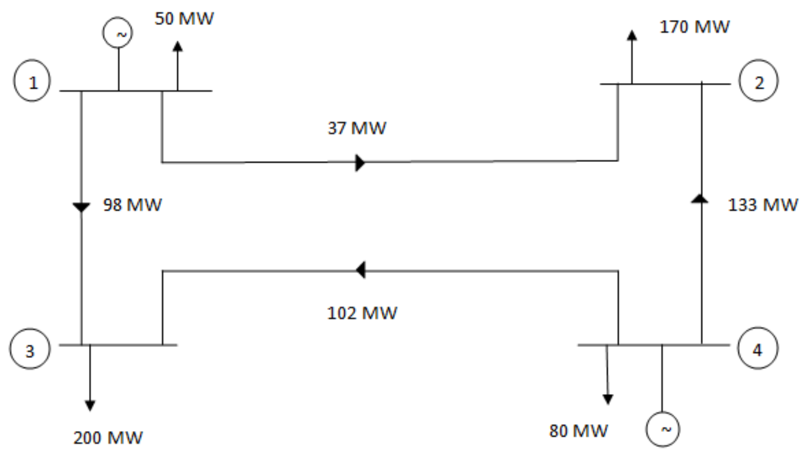

Fig 4.2: One-line diagram showing the specified loads and line flows for the proposed method

Then the real power flows in the lines are expressed using equation (3.3) of the proposed method as follows.

$P_{12}= B_{12}\delta_1 + (-B_{12}\delta_2)$

$P_{13} = B_{13}\delta_1 + (-B_{13}\delta_3)$

$P_{34} = 1.02B_{34}\delta_3 + 1.02(-B_{34})\delta_4$

$P_{24} = 1.02B_{24}\delta_2 + 1.02(-B_{24})\,\delta_4$

The above equations can be arranged in matrix form;

$$\begin{bmatrix} P_{12} \\ P_{13} \\ P_{34} \\ P_{24} \end{bmatrix} = \begin{bmatrix} B_{12} & -B_{12} & 0 & 0 \\ B_{13} & 0 & -B_{13} & 0 \\ 0 & 0 & 1.02B_{34} & -1.02B_{34} \\ 0 & 1.02B_{24} & 0 & -1.02B_{24} \end{bmatrix} \begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix}$$

Since bus 1 is slack δ_1=0 and the corresponding row and columns are omitted from above matrix equation so that

$$\begin{bmatrix} P_{12} \\ P_{13} \\ P_{34} \\ P_{24} \end{bmatrix} = \begin{bmatrix} -B_{12} & 0 & 0 \\ 0 & -B_{13} & 0 \\ 0 & 1.02B_{34} & -1.02B_{34} \\ 1.02B_{24} & 0 & -1.02B_{24} \end{bmatrix} \begin{bmatrix} \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix}$$

From the one-line diagram (Fig 4.2) the real power flows in the lines in per unit of 100 MVA are specified as

$$\begin{bmatrix} P_{12} \\ P_{13} \\ P_{34} \\ P_{24} \end{bmatrix} = \begin{bmatrix} 0.37 \\ 0.98 \\ -1.02 \\ -1.33 \end{bmatrix}$$

So,

[P] = [B][δ]

Or, $$\begin{bmatrix} 0.37 \\ 0.98 \\ -1.02 \\ -1.33 \end{bmatrix} = \begin{bmatrix} -B_{12} & 0 & 0 \\ 0 & -B_{13} & 0 \\ 0 & 1.02B_{34} & -1.02B_{34} \\ 1.02B_{24} & 0 & -1.02B_{24} \end{bmatrix} \begin{bmatrix} \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix}$$

Where

$$[B] = \begin{bmatrix} -B_{12} & 0 & 0 \\ 0 & -B_{13} & 0 \\ 0 & 1.02B_{34} & -1.02B_{34} \\ 1.02B_{24} & 0 & -1.02B_{24} \end{bmatrix}$$

$$= \begin{bmatrix} -19.84127 & 0 & 0 \\ 0 & -26.88172 & 0 \\ 0 & 16.03774 & -16.03774 \\ 27.41935 & 0 & -27.41935 \end{bmatrix}$$

The values of [B] matrix elements are calculated using equation 3.1 i.e. taking the inverse of respective line series reactance (X) part only as given in Table 4.1.

The transpose of B matrix is

$$[B]^T = \begin{bmatrix} -19.84127 & 0 & 0 & 27.41935 \\ 0 & -26.88172 & 16.03774 & 0 \\ 0 & 0 & -16.03774 & -27.41935 \end{bmatrix}$$

$$[B]^T [B] = \begin{bmatrix} 1145.49675 & 0 & -751.82075 \\ 0 & 979.83579 & -257.20910 \\ -751.82075 & -257.20910 & 1009.02986 \end{bmatrix}$$

$([B]^T\ [B])^{-1} = \begin{bmatrix} 0.00183 & 0.00038 & 0.00146 \\ 0.00038 & 0.00117 & 0.00059 \\ 0.00146 & 0.00059 & 0.00223 \end{bmatrix}$

$([B]^T [P] = \begin{bmatrix} -43.80901 \\ -42.70258 \\ 52.82623 \end{bmatrix}$

Now applying equation (3.7)

$[\delta] = \begin{bmatrix} \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = ([B]^T\ [B])^{-1}\ ([B]^T\ [P])$

$= \begin{bmatrix} -0.01927 \\ -0.03544 \\ 0.02865 \end{bmatrix}$ in radians

$= \begin{bmatrix} -1.10417 \\ -2.03071 \\ 1.64165 \end{bmatrix}$ in degrees

δ_1=0° for bus 1 i.e. the slack bus as already mentioned.

Then the power outputs of the generators are found applying equations (3.8) to bus 1 and 4.

$P_{g1} = \sum_{n=1}^{4} |Y_{in} V_i V_n| \cos(\theta_{in} + \delta_n - \delta_i) + P_{load1}$

$P_{g4} = \sum_{n=1}^{4} |Y_{in} V_i V_n| \cos(\theta_{in} + \delta_n - \delta_i) + P_{load4}$

Using the obtained δ values, Y= B values (neglecting resistance) and the specified load values in Table 4.2

$P_{g1} = \sum_{n=1}^{4} |Y_{in} V_i V_n| \cos(\theta_{in} + \delta_n - \delta_i) + P_{load1}$

$= Y_{11}V_1V_1 \cos(90°+\delta_1-\delta_1) + Y_{12}V_1V_2\cos(90°+\delta_2-\delta_1) + Y_{13}V_1V_3\cos(90°+\delta_3-\delta_1)+$

$Y_{14}V_1V_4\cos(90°+\delta_4-\delta_1) + 0.5$; (P_{load1} = 50MW =0.5 pu, Y_{14}=0 as no line between bus 1 and 4, $\theta_{in} = 90°$)

= 1.8349 per unit = 183.49 MW

Similarly,

$$P_{g4} = \sum_{n=1}^{4} |Y_{in}V_iV_n| \cos(\theta_{in} + \delta_n - \delta_i) + P_{load4}$$

$$= Y_{41}V_4V_1\cos(90°+\delta_1-\delta_4) + Y_{42}V_4V_2\cos(90°+\delta_2-\delta_4) + Y_{43}V_4V_3\cos(90°+\delta_3-\delta_4) +$$

$$Y_{44}V_4V_4\cos(90°+\delta_4-\delta_4) + 0.8 \; ; \qquad (P_{load4} = 80\text{MW} = 0.8 \text{ pu})$$

$$= 3.14076 \text{ per unit}$$

$$= 314.076 \text{ MW}$$

Table 4.4 summarizes the results from the proposed method for this validation case.

Table 4.4 Results obtained from the proposed method for the validation case

Bus Information					Line Flow (specified)	
			Generation (calculated)	Load		
Bus no.	Volts (p.u)	Angle (degrees)	P_g (MW)	P_{Load} (MW)	To Bus	(MW)
1	1.0	0	183.490	50	2 3	37 98
2	1.0	-1.104	-	170	-	-
3	1.0	-2.030	-	200	-	-
4	1.020	1.642	314.076	80	2 3	133 102
Total			497.566	500.00		

A comparison of Tables 4.3 and 4.4 shows that for a given set of loads, bus voltage magnitudes and pre-specified line flows (MW) the obtained generations (MW) from the proposed reverse load flow method are slightly less than the generations specified (for a voltage controlled i.e. PV bus) and calculated (for the slack bus) by the conventional load flow (Gauss-Seidel) method. The difference is due to mainly neglecting line resistance and line loss by the proposed method and due to the slight difference of the specified voltage magnitudes and line flows in the proposed method from those calculated by the Gauss-Seidel method.

It should be noted that the generations obtained by the proposed method can be interpreted as indicative for a quick scheduling so that the lines maintain a desired MW flow and are not overloaded while the buses do not face under voltage. In case the load is slightly higher than the obtained generation (as in the above case 497 MW generation vs. 500 MW load) then any of the generation unit can be adjusted around the obtained indicative values to match the load plus loss.

4.2.2 Case without Considering Transmission Loss

In this case, the same load data and bus voltages as in Table 4.4 and a different sets of MW flows in lines as shown in Fig. 4.3 are specified.

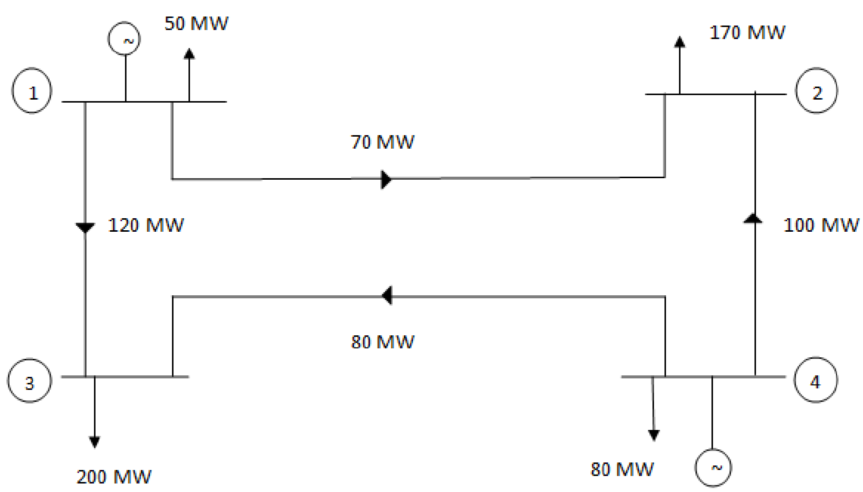

Fig 4.3: Single line diagram specifying a different set of line flows

Since, the test system remains the same the value of [B] matrix, its transpose and $([B]^T [B])^{-1}$ will remain the same.

From the one-line diagram (Fig 4.3) the real power flows in the lines in per unit of 100 MVA are

$$\begin{bmatrix} P_{12} \\ P_{13} \\ P_{34} \\ P_{24} \end{bmatrix} = \begin{bmatrix} 0.70 \\ 1.20 \\ -0.80 \\ -1.00 \end{bmatrix}$$

Then following the same procedure as illustrated in Section 4.2.1 the phase angles are obtained as

$$[\delta] = \begin{bmatrix} \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} = ([B]^T\ [B])^{-1}\ ([B]^T\ [P])$$

$$= \begin{bmatrix} -0.03396 \\ -0.04470 \\ 0.00284 \end{bmatrix} \text{in radians}$$

$$= \begin{bmatrix} -1.945908 \\ -2.56131 \\ 0.162732 \end{bmatrix} \text{in degrees}$$

$\delta_1 = 0°$ as bus 1 is the slack bus where voltage angle is assumed to be zero.

Then as before the generations at buses 1 and 4 are determined as

P_{g1} = 2.33862 per unit = 233.862 MW

P_{g4} =2.53669 per unit = 253.669 MW

Then this case has been analyzed using Gauss-Seidel (GS) method where the generation at bus 4 was pre-specified at the value obtained by the proposed method i.e. P_{g4} = 254 MW while same loads were used. The slack and PV bus voltage magnitudes were specified as 1.0 and 1.02 pu. Of course Gauss-Seidel method does not neglect line resistance and line loss. However, Table 4.5 shows an at a glance comparison of the proposed method and the GS method.

Table 4.5 Comparison of the proposed method (for a different case without considering transmission loss) against the Gauss-Seidel method

Line Flow (MW)	Gauss-Seidel	Proposed Method
1 - 2	73.016	70
1 - 3	127.37	120
4 - 2	99.022	100
4 - 3	74.978	80
Generation (MW)		
P_{g1}	250.39 (Slack bus)	233.862
P_{g4}	254.00	253.669
Bus voltage magnitudes (per unit)		
V_1	1.00	1.00
V_2	0.98254	1.00
V_3	0.96906	1.00
V_4	1.02	1.02
Bus voltage phase angle (degrees)		
δ_1	0.00	0.00
δ_2	-2.0215	-1.945908
δ_3	-2.5372	-2.56131
δ_4	-0.27322	0.162732

It is evident that the proposed method calculates generation units' MW outputs are slightly less than those from the GS method in order to satisfy a pre-specified set of line flows. However, the generation at bus 1 (which is a slack bus in GS method) obtained by the proposed method can be adjusted by about 3.4% more to match the load plus loss.

4.2.3 Case Considering Transmission Loss

In order to match the load plus loss by the generations to be calculated by the proposed reverse load flow method the line flows in the case described in Section 4.2.2 were pre-specified for the same set of loads by adding a margin of 0.5%, 1% and 3% respectively.

It was observed that a 3% addition (as shown in Fig. 4.4) in the required bare minimum line flows corresponding to the given set of loads without considering line loss, produces the best effect when compared with the GS load flow analysis. As before the obtained generations from the proposed method were specified for the PV bus and the GS method of load flow analysis was run. The total generation determined by the proposed method is now 504.38 MW against a specified load of 500 MW and the total generation (specified plus calculated) of 504.39 MW by the GS method.

Table 4.6 shows the results from both the methods along with those from the previous case i.e. Table 4.5.

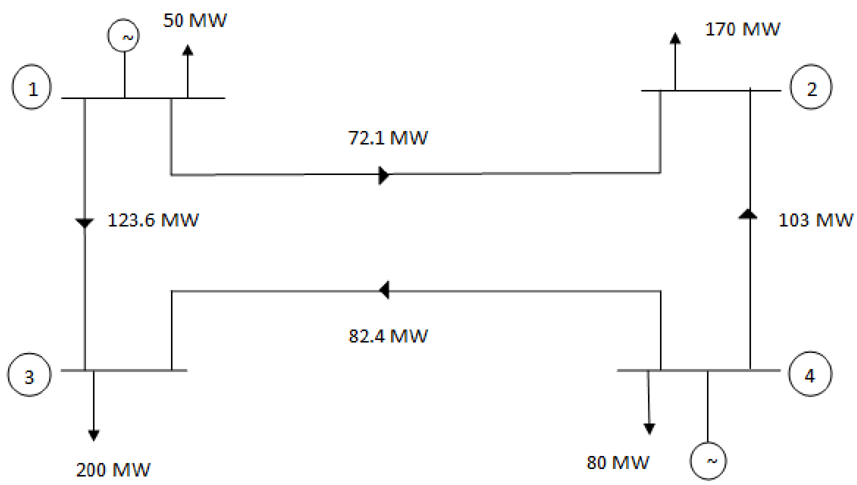

Fig 4.4: One-line diagram for a Case specifying the line flows by 3% more than those without considering transmission loss

Table 4.6 Comparison of the proposed method (for a case of pre-specified line flows considering transmission loss) against the Gauss-Seidel method

	Case (Table 4.5 repeated)		**Case (3% addition in specified line flows of the case in Table 4.5)**	
Line Flow (MW)	Gauss-Seidel	Proposed Method	Gauss-Seidel	Proposed Method
1 - 2	73.016	70	68.479	72.10
1 - 3	127.37	120	123.46	123.60
4 - 2	99.022	100	103.56	103.00
4 - 3	74.978	80	78.885	82.40
Generation (MW)				
P_{g1}	250.39	233.862	241.94	243.12
P_{g4}	254.00	253.669	262.44	262.44
Voltage (per unit)				
V_1	1.00	1.00	1.00	1.00
V_2	0.98254	1.00	0.98254	1.00
V_3	0.96906	1.00	0.96906	1.00
V_4	1.02	1.02	1.02	1.02
Angle (degree)				
δ_1	0.00	0.00	0.00	0.00
δ_2	-2.0215	-1.945908	-1.8837	-2.00435
δ_3	-2.5372	-2.56131	-2.4484	-2.63809
δ_4	-0.27322	0.162732	-0.035705	0.167889

Chapter 5

Conclusion

5.1 Conclusion

A method has been proposed for determining the indicative values of generators' real power outputs in a power system so as to satisfy a specified set of line flows (MW), bus voltage magnitudes and loads. The method is non iterative and it averts under voltage at the buses and overloading of lines.

This work is expected to be helpful in (a) security analysis of a power system and (b) transacting a specified magnitude of power through a given line in the deregulated (electricity market) operation of a power system.

The method has been applied on a 4 bus power system in the initial stage of such a work which is the first of its kind and verified by comparing with the conventional Gauss-Seidel load flow analysis that determines the line flows from a specified set of generation outputs and loads.

5.2 Suggestions for Further Research

The performance of the proposed method can be evaluated by applying it on a real life or larger test system.

Investigation into incorporating real power loss in the lines and the real power limits of the adjacent generators in a more systematic way while pre- specifying the MW flow in a line can be of further interest.

References

1. John J. Grainger and William D. Stevenson Jr. “Power System Analysis”, (McGraw-Hill, 1994)
2. D.P.Kothari, I.J.Nagarath, “Modern Power System Analysis”, (McGraw- Hill, 2006)
3. LakeneshwarPrakash Singh, “Advanced Power System Analysis and Dynamics”, (New Age International, 2008)
4. S.Sivanagaraju, B.V.Rami Reddy, “Electrical Power System Analysis”, (Firewall Media, 2007)
5. Gilbert, G.M.; Bouchard, D.E.; Chikhani, A.Y., "A comparison of load flow analysis using DistFlow, Gauss-Seidel, and optimal load flow algorithms," IEEE Canadian Conferenceon Electrical and Computer Engineering, 1998., Proc. vol.2, no., pp.850,853 vol.2, 24-28 May 1998).
6. NaziaMehnaz, Asiful Islam Bhuiyan, Manashi Roy, FarukHossain, "Load Flow Analysis and Abnormality Removal of Bangladesh Power System Using Software CYME PSAF," isms, pp.384-388, 2013 4th International Conference on Intelligent Systems, Modelling and Simulation, 2013.
7. Olamaei, J. South Tehran Branch, Islamic Azad Uni.,Tehran, Iran Ghasemabadi, M.A.; Kapourchali, M.H., “An efficient method for load flow analysis of distribution networks including PV nodes”, Electric Power and Energy Conversion Systems (EPECS), 2011 2nd International Conference.

Appendixes

A.1 MATLAB Code Developed for the Proposed Method

The following functions were performed together in MATLAB 7.10.0;

DESIGNED BY:
MD. IQBAL HOSSAN ASIF
MD. SOBHANUL AZIM
SALWA MAHBUB
CREATED: Nov-2013

```
function [Pg1, Pg4] = proposedMethod(ybus)

format short g;
in = inputdlg('Enter P');
numbers = str2num(in{1});
P = zeros(4, 1);
P(1, 1) = numbers(1);
P(2, 1) = numbers(2);
P(3, 1) = numbers(3);
P(4, 1) = numbers(4);
%%%%%%%%%%%%%%%%%%%%%%%%%%SETTING DEFAULT VALUES%%%%%%%%%%%%%%%%%%%%%%%%%%%
    x12 = 0.0504;
    x13 = 0.03720;
    x42 = 0.03720;
    x43 = 0.06360;
%----------------------------------
    %P = [0.37; 0.98; -1.02; -1.33];
    V = [ 1, 1, 1, 1.02];
    Pload1 = 0.50;
    Pload4 = 0.80;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%CALCULATING VALUES FOR ANGLE%%%%%%%%%%%%%%%%%%%%%%%%

%Calculate B-----------------------
    B = zeros(4, 3);
    B(1,1) = -(1/0.0504);
    B(2,2) = -(1/0.03720);
    B(3,2) = 1.02*(1/0.06360);
    B(3,3) = -1.02*(1/0.06360);
    B(4,1) = 1.02*(1/0.03720);
    B(4,3) = -1.02*(1/0.03720);
%----------------------------------
    BTransposed = B.';
    x = BTransposed*B;
```

```
    y = x^-1;
    z = BTransposed*P;
%-----------------------------------
    angle = y * z;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%%%%%CALCULATING Y VAULES%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    [theta, rho] = cart2pol(real(ybus(1, 2)), imag(ybus(1, 2)));
    Y12 = rho;
    [theta, rho] = cart2pol(real(ybus(1, 3)), imag(ybus(1, 3)));
    Y13 = rho;
    [theta, rho] = cart2pol(real(ybus(4, 2)), imag(ybus(4, 2)));
    Y42 = rho;
    [theta, rho] = cart2pol(real(ybus(4, 3)), imag(ybus(4, 3)));
    Y43 = rho;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Pg1-------------------------------------------------
    temp1 = Y12*V(1)*V(2)*cos(pi/2 + angle(1) - 0);
    temp2 = Y13*V(1)*V(3)*cos(pi/2 + angle(2) - 0);
    Pg1 = temp1 + temp2 + Pload1
%Pg2-------------------------------------------------
    temp1 = Y42*V(4)*V(2)*cos(3.1418/2 + angle(1) - angle(3));
    temp2 = Y43*V(4)*V(3)*cos(3.1418/2 + angle(2) - angle(3));
    Pg4 = temp1 + temp2 + Pload4
%----------------------------------------------------
end
```

A.2 MATLAB Code Developed for Gauss-Seidel Method

The following functions were performed together in MATLAB 7.10.0

DESIGNED BY:
MD. IQBAL HOSSAN ASIF
MD. SOBHANUL AZIM
SALWA MAHBUB
CREATED: Nov-2013

```
function [z, lineFlow] = zxc()
format short g
disp (' TABLE 9.2 PAGE # 337   LINE DATA FOR')
linedata=[1     2       0.01008,    0.05040,   3.815629,     -19.078144,
10.25,  0.05125;
          1     3       0.00744,    0.03720,   5.169561,     -25.847809,
7.75,   0.03875;
          2     4       0.00744,    0.03720,   5.169561,     -25.847809,
7.75,   0.03875;
          3     4       0.01272,    0.06360,   3.023705,     -15.118528,
12.75,  0.06375]


disp (' TABLE 9.3 PAGE # 338   BUS DATA ')
busdata=[1   0,     0,  50,     30.99,  1.00,   0   1;
         2   0,     0,  170,    105.35, 1.00,   0   2;
         3   0,     0,  200,    123.94, 1.00,   0   2;
         4   262.44,   0 , 80,     49.58,  1.02,   0   3]
% Bus Type: 1.Slack Bus    2.PQ Bus    3.PV Bus

ss=j*linedata(:,8);

y=linedata(:,5)+j*linedata(:,6);

totalbuses = max(max(linedata(:,1)),max(linedata(:,2)));    % total buses
totalbranches = length(linedata(:,1));                      % no. of branches
ybus = zeros(totalbuses,totalbuses);

for b=1:totalbranches
ybus((linedata(b,1)),(linedata(b,2)))=-y(b);
ybus((linedata(b,2)),(linedata(b,1))) =ybus((linedata(b,1)),(linedata(b,2)));

end

for c=1:totalbuses
for d=1:totalbranches
iflinedata(d,1) == c || linedata(d,2) == c
ybus(c,c) = ybus(c,c) + y(d) + ss(d);
end
end
end
disp('TABLE 9.3 PAGE # 338   BUS ADMITTANCE MATRIX')
ybus
z=zeros(totalbuses,4);
busnumber=busdata(:,1);
```

```
PG=busdata(:,2);
QG=busdata(:,3);
PL=busdata(:,4);
QL=busdata(:,5);
V=busdata(:,6);
VV=V;
ANG=busdata(:,7);
type = busdata(:,8);

P = (PG-PL)./100;    % per unit active power at buses
Q = (QG-QL)./100;    % per unit reactive power at buses
tol=1;
iter=0;
kk=input('Enter the tolerance for iteration ');
%alfa=input('Enter the value of ALPHA ');
alfa=1.6
whiletol>kk

for i = 2:totalbuses

      YV = 0;

for k = 1:totalbuses
if i~=k
                YV = YV + ybus(i,k)* V(k);  % multiplying admittance &
voltage
end
        YV;
end
ifbusdata(i,8) == 3     %Calculating Qi for PV bus
            %Q(i) = -imag(conj(V(i))*(YV + ybus(i,i)*V(i)));
Q(i) =  -imag(conj(V(i))*(YV + ybus(i,i)*V(i)));
busdata(i,3)=Q(i);
end

       % end
        V(i) = (1/ybus(i,i))*((P(i)-j*Q(i))/conj(V(i)) - YV); % Compute Bus
Voltages.

         % Calculating Corrected Voltage for PV bus
ifbusdata(i,8) == 3

vc(i)=abs(VV(i))*(V(i)/abs(V(i)));
busdata(i,6)=vc(i);
V(i)=vc(i);
end


      % Calculating Accelerated Voltage for PQ bus
ifbusdata(i,8) == 2
VACC(i)= VV(i)+alfa*(V(i)-VV(i));
busdata(i,6)=VACC(i);
V(i)=VACC(i);
end
        %V(i)=V;

end
```

```
iter = iter + 1;      % Increment iteration count.
tol = max(abs(abs(V) - abs(VV)));     % Calculate tolerance.
   VV = V;
end
Q;
iter
YV;
V;

z(1:totalbuses,1)=busdata(:,1);
z(1:totalbuses,2)=busdata(:,8);
z(1:totalbuses,3)=abs(busdata(:,6));
z(1:totalbuses,4)=radtodeg(angle(V));


%We calculate the values of P1, Q1 and Q4 from the function below
[p1 q1 q4] = getPAndQ(ybus, z);

%Now we make the respective changes in the 'busdata' table
busdata(1, 2) = p1 + busdata(1, 4);
busdata(1, 3) = q1 + busdata(1, 5);
busdata(4, 3) = q4 + busdata(4, 5);

%Display MW and its sum
disp('          MW    ');
busData = busdata(:, 2)
disp('         Sum ' );
sum(busdata(:, 2))

%Display MVar and its sum
disp('          Mvar    ');
busData = busdata(:, 3)
disp('         Sum ' );
sum(busdata(:, 3))

%%%%%%%%%%%%%%%%%%%%%%%%%%%% END OF SECTION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

disp('           Bus No.   Bus Type     Voltage      Angle      ');
z
lineFlow = getLineFlow(ybus, linedata, z);
disp('           Bus No.   Bus Name       MW          Mvar      ');
lineFlow

%%%%%%%%%%%%%%%%%%%%%%%%%% CALCULATING THE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%% DIFFERENCE IN LINEFLOW FOR %%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%% MW AND MVAR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[df, dfM] = dfrnc(lineFlow); %calculate using method
df %display
dfM %display

%%%%%%%%%%%%%%%%%%%%%%%%%% END OF SECTION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
proposedMethod(ybus)
end
```

Real Power (MW)& Reactive Power (MVAR)

```
function [p1 Q1 Q4] = getPAndQ(ybus, z)

format short g

p1 = 0;
Q1 = 0;
Q4 = 0;

V = z(:, 3);
Angle = z( :, 4);

for i=1:4
temp = ybus(i, 1);
    [theta, rho] = cart2pol(real(temp), imag(temp));
    theta1i = rad2deg(theta);
    Y1i = rho;

    term1 = abs( Y1i*V(1)*V(i) );
    term2 = cosd( theta1i + Angle(i) - Angle(1) );
    p1 = p1 + term1*term2;

end

 p1 = p1*100;

for i=1:4
temp = ybus(1, i);
    [theta, rho] = cart2pol(real(temp), imag(temp));
    theta1i = rad2deg(theta);
    Y1i = rho;

    term1 = abs( Y1i*V(1)*V(i) );
    term2 = sind( theta1i + Angle(i) - Angle(1) );

    Q1 = Q1 + term1*term2;
end

      Q1 = -Q1*100;

for i=1:4
temp = ybus(4, i);
    [theta, rho] = cart2pol(real(temp), imag(temp));
    theta4i = rad2deg(theta);
    Y4i = rho;

    term1 = abs( Y4i*V(4)*V(i) );
    term2 = sind( theta4i + Angle(i) - Angle(4) );

    Q4 = Q4 + term1*term2;
end

    Q4 = -Q4*100;

End
```

Line Flow

```
functionlineFlow = getLineFlow(ybus, linedata, z)
format short g

Pij = 0;
Qij = 0;

busNumberToBusName = [2 3;1 4; 1 4; 2 3];
lineFlow = zeros(8, 4);

placeIn = 1;
for c=1:4
lineFlow(placeIn, 1) = lineFlow(placeIn, 1) + c;
lineFlow(placeIn, 2) = busNumberToBusName(c,1);
placeIn = placeIn + 1;

lineFlow(placeIn, 1) = lineFlow(placeIn, 1) + c;
lineFlow(placeIn, 2) = busNumberToBusName(c,2);
placeIn = placeIn + 1;
end

placeIn = 1;
for i=1:4
for x=1:2
            j = busNumberToBusName(i, x);

            Vi = z(i, 3);
Vj = z(j, 3);
lambdai = z(i, 4);
lambdaj = z(j, 4);

Gij = 0;
Bij = 0;
Bprimeij = 0;

for c=1:4
if (linedata(c, 1) == i &&linedata(c, 2) == j)||(linedata(c, 1) == j
&&linedata(c, 2) == i)
Gij = -linedata(c,5);
Bij = linedata(c, 6);
Bprimeij = linedata(c, 8);
break;
end
end

temp = ybus(i, j);
            [theta, rho] = cart2pol(real(temp), imag(temp));
thetaij = rad2deg(theta);
Yij = rho;

            term1 = -(abs(Vi)^2) * Gij;

            term2 = Vi*Vj*Yij;

            term3 = cosd(thetaij + lambdaj - lambdai);
```

```
Pij=( term1 + abs(term2)*term3 )*100;


            qterm1 = (abs(Vi)^2)*(Bprimeij - (-Bij));

            qterm2 = term2;

            qterm3 = sind(thetaij + lambdaj - lambdai);

Qij = -( qterm1 + abs(qterm2)*qterm3 )*100;

lineFlow(placeIn, 3) = Pij;
lineFlow(placeIn, 4) = Qij;

placeIn = placeIn + 1;
end
end
end
```

Checking Difference between Line flow and injection power

```
function [difMW, difMvar] = dfrnc(lineFlow)


difMW = 0;
difMvar = 0;

temp = size(lineFlow);
check = zeros(temp(1, 1), 1)


for i=1:temp(1,1)
    x = lineFlow(i, 1);
    y = lineFlow(i, 2);
for j=i:temp(1,1)
if(lineFlow(j, 2) == x &&lineFlow(j, 1) == y && check(j) == 0)
difMW = difMW + abs(lineFlow(j, 3) + lineFlow(i, 3));
difMvar = difMvar + abs(lineFlow(j, 4) + lineFlow(i, 4));
end
end
end


end
```