

POWER FLOW ANALYSIS: SIMULATION FOR DIFFERENT BUSES OF A SYSTEM ANALIZA E RRJEDHJES SË FUQISË: SIMULIMI PËR ZBARRA TË NDRYSHME TË SISTEMIT

ISUF KRASNIQI*, AGON KOKA**

* Faculty of Electrical and Computer Engineering, University of Prishtina, Kosova,

** FH Joanneum University of Applied sciences and Technology, Graz, Austria

Email: isufkrasniqi@yahoo.com

AKTET IV, 2: 178-187, 2011

PERMBLEDHJE

Kemi krijuar një program të ri kompjuterik për të simuluar dhe hulumtuar nëse fuqia që gjeneron sistemi është në përputhje me kërkesat tona. Programi mundëson llogaritjen e madhësive dhe të këndeve të fazës të fuqisë aktive dhe reaktive në zbarra të një sistemi të caktuar. Duke përdorur metodën Newton-Raphson si pjesë kryesore të programit, fituam rezultatet që i përmbushën kërkesat e parapërcaktuara.

Fjalët çelës: rrjedha e fuqisë, metoda Newton-Raphson, simulimi.

SUMMARY

We have created a new computer program to simulate and investigate the generated power of a system according to our demands. The program calculates the real and reactive power magnitudes and their angles respectively at the buses in power system. We used Newton-Raphson method as the main body of our Program. The output values we obtained fulfilled our required objectives.

Keyword: power flow, Newton-Raphson method, simulation.

1. INTRODUCTION

What is Power Flow?

Power Flow studies are of great importance in planning and designing the future expansion of power systems as well as in determining the best operation of existing systems [1, 2]. The principal information obtained from a Power Flow study is the magnitude and phase angle of the voltage at each bus and the real and reactive power flowing in each line.

Power Flow includes:

- Generating supplies according to demand,
- Keeping Bus voltage magnitude close to the related ideal values,
- Allowing the function of generators to operate within specific real and reactive power limits,

- Obtaining statistical data concerning the transmission lines and transformers, in order to prevent overload from occurring on such peripherals.

Conditional nodal or loop analysis is not suitable for Power Flow studies because the input data for loads are normally given in terms of power, and not impedance. Generators are considered as Power sources, not voltage or current sources. Power Flow programs are most suitable to analyze large transmission grids and the complex interaction between transmission grids and power market [3, 4].

Definitions and Acronyms

n_g	Number of generators in the power system.
P_D	Real Power.
Q_D	Reactive Power.
P_Q	Load Buses.
P_G	Real Power Generated.
ϑ	Voltage Phase.
Y_{bus}	The admittance matrix of a power system.
$p.u$	Per unit
MATLAB	Is a numerical computing environment and programming language.
MATPOWER	Is a package of MATLAB M-files for solving power flow and optimal power flow problems.

PROBLEM STATEMENT

The Power Flow Problem

The goal of a Power Flow study is to obtain complete voltage angle and magnitude information for each bus in a power system for specified load and generator real power and voltage condition. Once this information is known, real and reactive power flow on each branch as well as generator reactive power output can be analytically determined. Due to the nonlinear nature of this problem, numerical methods are employed to obtain a solution that is within an acceptable tolerance.

The solution to the Power Flow problem begins with identifying the known and unknown variables in the system. The known and unknown variables are dependent on the type of bus. A bus without any generators connected to it is called a Load Bus. With one exception, a bus with at least one generator connected to it is called a Generator Bus. The exception is one arbitrarily-selected bus that has a generator. This bus is referred to as the Slack Bus.

Equivalent Circuit Model

The equivalent circuit model that we are using for our simulation is shown in Figure 1.

Schematic, indicating the input and output to the buses along with their transformer ratios in between.

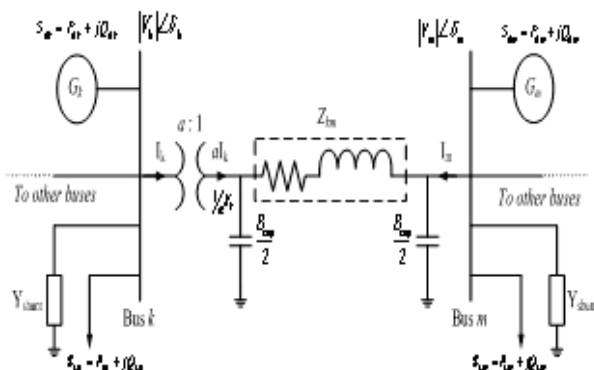


Figure 1: Equivalent Circuit Model

- $|V_k|$ = voltage magnitude at bus k ,
- δ_k = voltage phase angle at bus k ,
- I_k = current flowing from bus k ,
- I_m = current flowing from bus m ,
- G_k = generator at bus k ,
- S_{GK} = generator complex power at bus k ,
- P_{GK} = generator real power,
- Q_{GK} = generator reactive power,
- S_{LK} = load complex power at bus k ,
- P_{LM} = load real power,
- Q_{LK} = load reactive power,
- Z_{km} = line impedance,
- B_{cap} = line shunt susceptance,
- α = normalized transformer turns ratio.

Newton Raphson Solution Method

There are several different methods of solving the resulting nonlinear system of equations. The most popular is known as the Newton-Raphson Method. This method begins with initial guesses of all unknown variables (voltage magnitude and angles at Load Buses and voltage angles at Generator Buses). Next, a Taylor Series is written, with the higher order terms ignored, for each of the power balance equations included in the system of equations. The result is a linear system of equations that can be expressed as:

$$\begin{bmatrix} \Delta\theta \\ \Delta|V| \end{bmatrix} = -J^{-1} \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \quad 1$$

where ΔP and ΔQ are called the mismatch equations:

$$\Delta P_i = -P_i + \sum_{k=1}^N |V_i| |V_k| (G_{ik} \cos \theta_{ik} + B_{ik} \sin \theta_{ik}) \quad 2$$

$$\Delta Q_i = -Q_i + \sum_{k=1}^N |V_i| |V_k| (G_{ik} \sin \theta_{ik} - B_{ik} \cos \theta_{ik}) \quad 3$$

and J is a matrix of partial derivatives known as a Jacobian:

$$J = \begin{bmatrix} \frac{\delta \Delta P}{\delta \theta} & \frac{\delta \Delta P}{\delta |V|} \\ \frac{\delta \Delta Q}{\delta \theta} & \frac{\delta \Delta Q}{\delta |V|} \end{bmatrix} \quad 4$$

The linearized system of equations is solved to determine the next guess ($m + 1$) of voltage magnitude and angles based on:

$$\theta^{m+1} = \theta^m + \Delta\theta \quad 5$$

$$|V|^{m+1} = |V|^m + \Delta|V| \quad 6$$

The process continues until a stopping condition is met. A common stopping condition is to terminate if the norm of the mismatch equations are below a specified tolerance.

A rough outline of the solution to the Power Flow problem is:

1. Make an initial guess of all unknown voltage magnitudes and angles. It is common to use a "flat start" in which all voltage angles are set to zero and all voltage magnitudes are set to 1.0 p.u.,
2. Solve the power balance equations using the most recent voltage angle and magnitude values,
3. Linearize the system around the most recent voltage angle and magnitude values,
4. Solve for the change in voltage angle and magnitude,
5. Update the voltage magnitude and angles,

A flow chart of the solution to the Power Flow problem is shown in Figure 2.

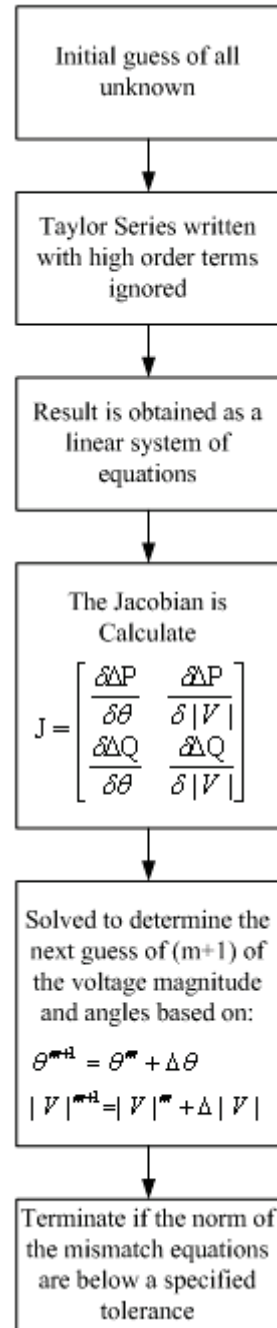


Figure 2: Flow Chart of the Solution

How to Implement Newton Raphson Method on a 4 bus case

Figure 3 shows the one-line diagram of a simple power system. Generators are connected at

buses 1 and 4 while the loads are indicated at all four buses. Base values for the transmission system are 100 MVA, 230 KV. The line data of the Table 1 give the per unit series impedance and the line charging susceptances for the equivalents of the four lines identified by the buses at which they terminate.

The bus data in Table 2 list values of P , Q and V at each bus. The Q values of load are calculated from the corresponding P values assuming a power factor of 0.85. The net scheduled values P_i , and Q_i are negative at the load buses 2 and 3. Generated Q_g is not specified where the voltage magnitude is constant. In the voltage column the values for the load buses are flat-start estimates.

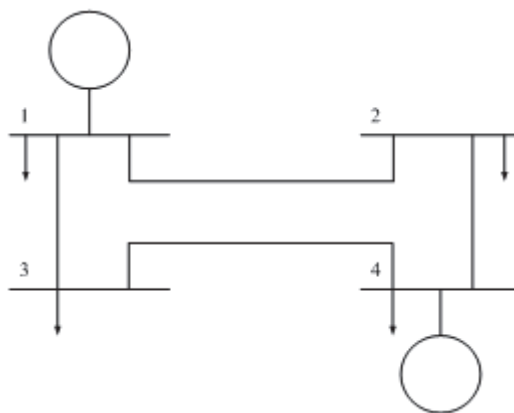


Figure 3: Bus Branch Diagram

Line bus to bus	Series Z		Series $Y=Z^{-1}$		Shunt Y	
	R per unit	X per unit	G per unit	B per unit	Total Charging Mvar	Y/2 per unit
1-2	0.01008	0.05040	3.815629	-19.078	10.25	0.05125
1-3	0.00744	0.03720	5.169561	-25.848	7.75	0.03875
2-4	0.00744	0.03720	5.169561	-25.848	7.75	0.03875
3-4	0.01272	0.06360	3.023705	-15.119	12.75	0.06375

Table 1: Bus Branch Data

Bus	Generation		Load		V (per unit)	Remarks
	P (MW)	Q (MW)	P (MW)	Q (MW)		
1	-----	-----	50	30.99	1 < 0	Slack bus
2	0	0	170	105.35	1 < 0	Load bus (inductive)
3	0	0	200	123.94	1 < 0	Load bus (inductive)
4	318	-----	80	49.58	1 < 0	Voltage controlled

Table 2: Bus Branch Individual Data

Bus Number	(1)	(2)	(3)	(4)
(1)	8.985190 -j.44.83	-3.8156 +j19.07	-5.169 +j25.84	0
(2)	-3.81562 +j19.078144	8.98 -j44.8359	0	-5.169+j25.84
(3)	-5.1695 +j25.847	0	8.193 -j40.8693	-3.023+j5.11
(4)	0	-5.169 +j25.847	-3.023 +j15.118	8.193-j40.863

Table 3: Bus Branch Individual Polar Data

Determining the number of rows and columns in the Jacobian, and then calculating the initial mismatch e.g. $\Delta p_1^{(0)}$, $\Delta p_2^{(0)}$, $\Delta p_3^{(0)}$ and $\Delta p_4^{(0)}$ along with the initial values of the Jacobian elements of the second row, third column, of the second row,

second column, and the fifth row fifth column, using all the values in Table 2.

Since the slack bus has no rows or columns in the Jacobian, a 6x6 matrix would be necessary if P and Q were specified for the other individual buses. In fact however the voltage magnitude is

specified (held constant) at bus 4, and thus the Jacobian will be 5x5 matrix. In order to calculate the power based on the estimated and the specified voltages of Table 2, we require the polar form of the off-diagonal entries of Table 3.

$$Y_{31} = 26.359695 \angle 101.30993$$

$$Y_{34} = 15.417934 \angle 101.30993$$

and the diagonal element $Y_{33} = 8.193267 - j40.863838$.

Since Y_{32} and the initial values $\delta_3(0)$ and $\delta_4(0)$ are all zero, we obtain

$$P_3^{(0), calc} = |V_3|^2 G_{33} + |V_3 V_1 V_{31}| \cos \theta_{31} + |V_3 V_4 V_{34}| \cos \theta_{34}$$

$$= (1.0)^2 (8.193267) + (1.0 \times 1.0 \times 26.359695) \cos(101.30993)$$

$$+ (1.0 \times 1.02 \times 15.417934) \cos(101.30993)$$

$$= -0.06047 \text{ per unit}$$

The essential real power in the network at bus 3 is -2.00/unit, and so the initial mismatch which we want to calculate has the value

$$\Delta P_3^{(0), calc} = -2.00 - (-0.06047) = -1.93953 \text{ per unit}$$

$$\text{From this : } dP_i/d\delta_j = -|V_i V_j Y_{ij}| \sin(\theta_{ij} + \delta_j - \delta_i)$$

Using the initial input data, we can similarly calculate initial values of the other elements of the Jacobian and of the power mismatches at all buses of the system.

For the system of the preceding example the numerical values for the initialization of the mismatch equations are now shown below:

$$\begin{bmatrix} 45.443 & 0 & -26.365 & 8.882 & 0 \\ 0 & 41.269 & -15.421 & 0 & 8.133 \\ -26.365 & -15.421 & 41.786 & -5.273 & -3.084 \\ -9.089 & 0 & 5.273 & 44.229 & 0 \\ 0 & -8.254 & 3.084 & 0 & 40.459 \end{bmatrix} \begin{bmatrix} \Delta\delta_1 \\ \Delta\delta_2 \\ \Delta\delta_3 \\ \Delta|V_2| \\ \Delta|V_4| \end{bmatrix} = \begin{bmatrix} -1.597 \\ -1.940 \\ 2.213 \\ -0.447 \\ -0.835 \end{bmatrix}$$

This system of equations yields values for the voltage corrections of the first iteration which are needed to update the state variables according to equation $\delta_i(k+1) = \delta_i(k) + \Delta\delta_i(k)$.

At the end of the first iteration the set of updated voltages at the buses is shown in Table 4.

These updated voltages are then used to recalculate the Jacobian and mismatches of the second iteration, and so on. The iterative procedure continues until either the mismatches ΔP_i and ΔQ_i become less than their stipulated allowable values or all $\Delta\delta_i$ and $\Delta|V_i|$ are less than the chosen precision index.

Bus no i	(1)	(2)	(3)	(4)
δ_i (deg)	0	-0.93094	-1.78790	-1.54383
V _i (per unit)	1.00	0.98335	0.97095	1.02

Table 4: Iteration Data

```

File Edit Text Go Cell Tools Debug Desktop Window Help
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  % Kuwait University - Electrical Engineering Department
3  % Newton Raphson Power Flow Solver
4  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5  %% ----- Initialization -----
6
7  - clear all
8  - close all
9  - format compact
10 - format short
11 - j=sqrt(-1);
12 - casefile = input('Enter data file name = ','S');
13 - run(casefile);
14
  
```

Figure 4: Initialization Sector Program

The number of iterations required by the Newton-Raphson method using bus admittances is practically independent of the number of buses. The time for the Gauss-Seidel Method (employing bus admittances) increases almost directly with the number of buses. On the other hand, computing the elements of the Jacobian is time consuming, and the time per iteration is considerably longer for the Newton-Raphson method. When sparse matrix techniques are employed the advantages of shorter computer time for a solution of the same accuracy is in favor of the Newton-Raphson method for all but very small systems.

Generating the program for the Power Flow Problem

The program we've written is a general program that is used to calculate the element values for all different bus cases. The first section of the program consists of the initialization stage.

The second section of the program consists of the Y-bus matrix calculations. It is composed of codes and loops that are associated with the Y-bus interaction elements. The central backbone of the program used to implement the interaction between MATLAB and the different cases being studied is the analyzed using Newton-Raphson section.

```

14
15 % ----- Y Bus Calculations -----
16
17 - numline = size(branch,1);
18 - numbus = size(bus,1);
19 - Gs = bus(:,5);
20 - Bs = bus(:,6);
21 - Yshunt = (Gs + j*Bs)/baseMVA;
22 - frombus = branch(:,1);
23 - tobus = branch(:,2);
24 - R=branch(:,3);
25 - X= branch(:,4);
26 - Scap = branch(:,5);
27 - BranchStatus = branch(:,11);
28 - Y = zeros(numbus, numbus);
29 % This loop will build Ybus Matrix elements
30 - for i=1:numline,
31     % Assigning Tap Values:
32     if branch(i,9)==0.00,
33         tap(i,1) =1;
34     elseif branch(i,9) ~=0.00,
35         tap(i,1) = branch(i,9);
36     end
37     %checking Branch Status (online of Offline):
38     if BranchStatus(i) ==1,
39         %-----
40         % Ybus Matrix:
41         Yline = 1/(R(i) +j*X(i));
42         Ycap = j*Scap(i)/2;
43         % Ysh:
44         Y(frombus(i),tobus(i)) = Y(frombus(i), tobus(i)) - (Yline/conj(tap(i)));
45         %YshK:
46         Y(tobus(i),frombus(i)) = Y(tobus(i), frombus(i)) - (Yline/tap(i));

```

Figure 5: Y-bus Equations

```

File Edit Text Go Cell Tools Debug Desktop Window Help
- 10 + + 11 x Stack Base
37 - %T10,47 = T10,47 * T100000000;
38 - end
39 - end
40
41 %----- Preparing for Newton Raphson -----
42 - numbus = size(bus,1);
43 - numGen = size(gen,1);
44
45 - gen_bus=find(bus(:,2)**1); %gen buses index without ref bus = [2:3];
46 - load_bus=find(bus(:,1)**1);
47
48 - num_bus = numbus-1; %number of buses without reference
49 - num_load = numbus - numGen; %number of buses without reference or generators (number of load buses)
50
51 - Vb = bus(:,8); %extracting column 8 of the bus matrix which equals to initial voltage magnitude
52 - Va = bus(:,9)*(pi/180); %extracting column 9 of the bus matrix which equals to initial voltage phase
53
54 - vmi = Vb;
55 - vai = Va;
56
57 - vmi(1:gen_bus) = []; % deleting reference and generator bus corresponding elements
58 - vai(1) = []; % deleting the corresponding reference elements
59
60 - gm = abs(V);
61 - ga = angle(V);
62
63 - pd = bus(:,3);
64 - Qd = abs(pd);
65 - qd = bus(:,4);
66 - Cd = abs(qd);
67 - Pac = -pd/baseMVA; % extracting the demanded real power and dividing it by base MVA to get the per unit values, note the sign
68 - Qdc = -qd/baseMVA; % extracting the demanded reactive power and dividing it by base MVA to get the per unit values, note the sign
69
70 - for i=1:size(gen_bus)
71 - % num = find(genbus); % gen bus index

```

Figure 6: Programming Newton-Raphson

```

File Edit Text Go Cell Tools Debug Desktop Window Help
- 10 + + 11 x Stack Base
Run newton_raphson_final_taps.m
1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 % Kuwait University - Electrical Engineering Department
3 % Newton Raphson Power Flow Solver
4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5 %----- Initialization -----
6
7 - clear all
8 - close all
9 - format compact
10 - format short
11 - j=sqrt(-1);
12 - casefile = input('Enter data file name = ','S');
13 - run(casefile);
14
15 %----- Y Bus Calculations -----
16
17 - numline = size(branch,1);
18 - numbus = size(bus,1);
19 - Gs = bus(:,5);
20 - Bs = bus(:,6);
21 - Yshunt = (Gs + j*Bb)/baseMVA;
22 - frombus = branch(:,1);
23 - tobus = branch(:,2);

```

Figure 7: Running the Program

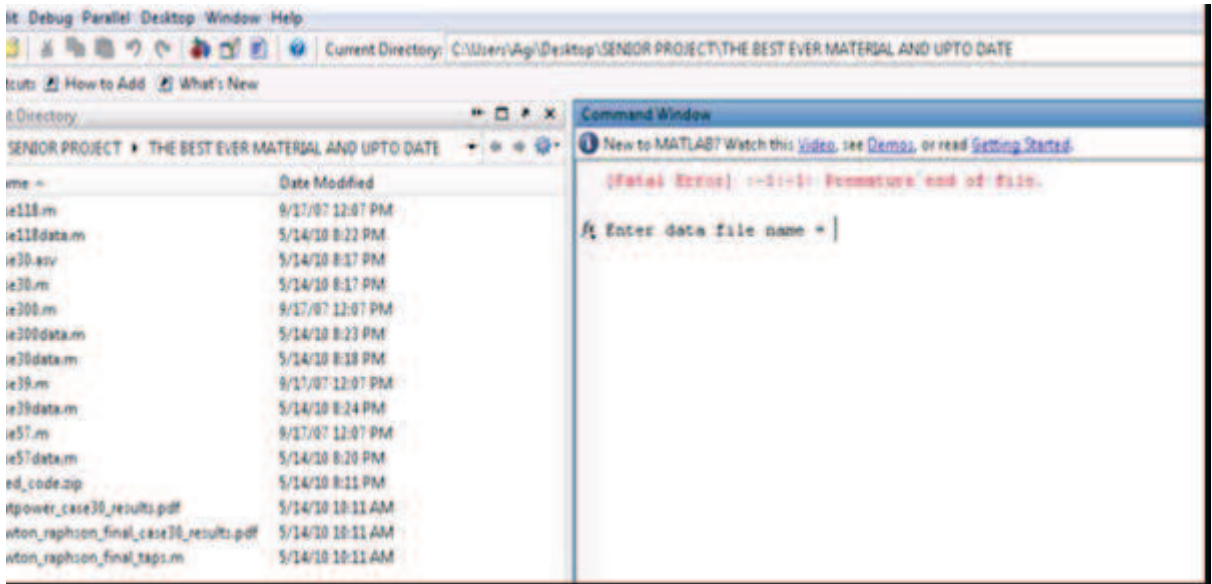


Figure 8: Starting the Program

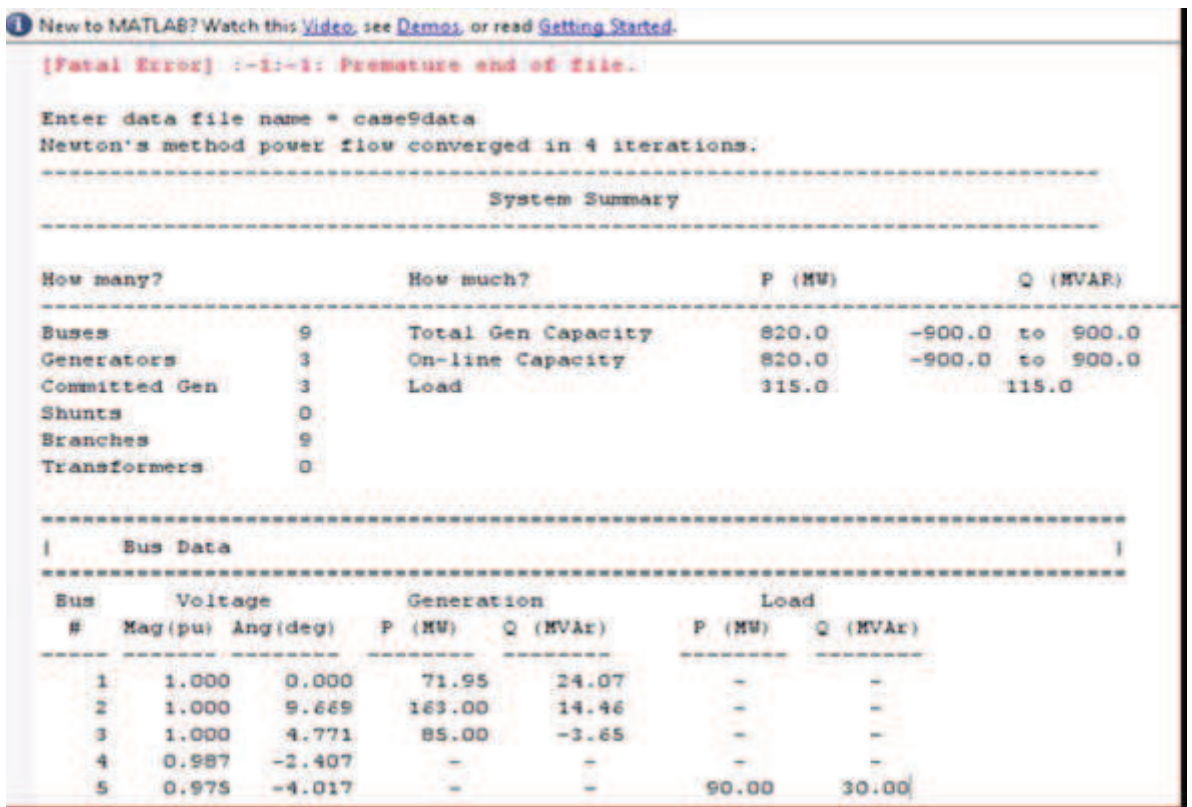


Figure 9: Magnitude and Angles of case 9 data

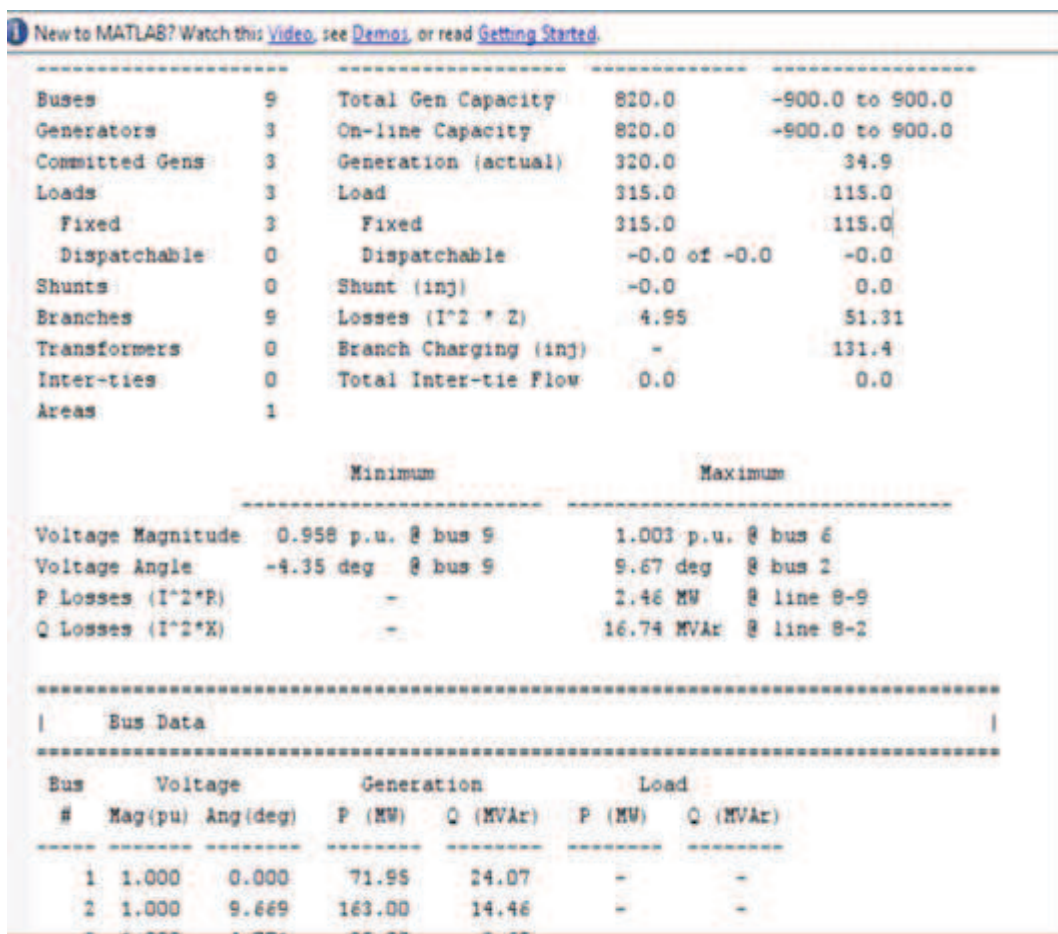


Figure 10: Case 9 data results from MATPOWER 3.2

RESULTS AND DISCUSSIONS

Implementation of the general program

If one needs to calculate the element values for a 9buscase; we click on run as shown in Figure 7. This appears up on the command window as shown in Figure 8; then write the name of the case data we want to run, i.e case 9 data. We will obtain the element values for each bus, i.e magnitude and angle values as shown in Figure 9. In order to make sure that our solution is correct for case9data, we run the same case9data on MATPOWER3.2 and we got the following result as shown in Figure 10.

This indicates that we obtained exactly the same values as the calculated values from our Newton-Raphson program that we implemented.

Application

This is applied in order to simulate our power station work capabilities. The advantage of this is that we do not need to setup a bus system before having done our complete analysis of the different run cases. Power system simulation models are a class of computer simulation programs that focus on the operation of electrical power systems. These computer programs are used in a wide range of planning and operational situations including:

1. Long-term generation and transmission expansion planning,
2. Short-term operational simulations,
3. Market analysis (e.g. price forecasting).

These programs typically make use of mathematical optimization techniques such linear programming, quadratic programming, and mixed integer programming.

Key elements of power systems that are modeled include:

1. Load flow (power flow study),
2. Short circuit (Short Circuit Analysis),
3. Transient stability (Transient Stability Simulation),
4. Optimal dispatch of generating units (unit commitment),
5. Transmission optimal power flow.

Electric power programs are written routinely by General Electric Company and clients to handle a wide range of system analysis and reporting tasks, for example:

- Line and generator-outage contingency checking and reporting of overloads and under voltages,
 - Reassigning loads to simulate annual growth, seasonal and hourly load profiles, and proposed large power consumers,
 - Simulation of time-switched and voltage/load-switched capacitors on distribution systems,
 - Re-dispatch of generation to simulate proposed sales and purchases of power, emergency supply plans, and independent power producer operations,
- Calculation of distribution factor tables and transfer limits for use by system operators.

CONCLUSION

It is important to emphasize the fact that the resulting values I obtained were identical to those of the MATPOWER 3.2. MATPOWER 3.2 allowed us to obtain the exact theoretical background values of the different cases being implemented. Although MATPOWER 3.2 had Ideal values for all the calculated elements, we were still able to get as close to the MATPOWER values within a certainty of 0.02. This for us was a big achievement because the program we designed to calculate using Newton-Raphson (numerical methods) was very successful.

REFERENCES

- [1] J. Duncan Glover, Mulukutla S. Sarma and Thomas J. Overbye. *Power System Analysis and Design*. Toronto, Canada: Thomson, 2008, pp. 354-280.
- [2] <http://journals.tubitak.gov.tr/elektrik/issues/elk-07-15-3/elk-15-3-2-0607-4.pdf>, [Feb. 15, 2010].
- [3] Hadi Saadat, *Power Analysis*, McGraw-Hill Series in Electrical and Computer Engineering, 2003, New York
- [4] <http://www.forwind.de/publications/HVDC.pdf>, Institute of Electric Power Systems, University of Hannover, Germany.