DEVELOPMENT OF WEB BASED SYSTEM LOAD FLOW (NEWTON RAPHSON) TOOLS USING WEB BASED APPLICATION

Nurul Syazana Binti Norman

Faculty of Electrical Engineering
Universiti Teknologi MARA
40450 Shah Alam, Selangor

ABSTRACT - *This paper presents the development of web based power system load flow (Newton-Raphson) component using web based application. Component web based is a visual and reusable program. It is easy to maintain and also low cost. User can access the web everywhere or anywhere using internet because the program does not need to be installed. This paper present the used of PHP language in applying the Newton Raphson Method.*

PHP (PHP: Hypertext Preprocessor) is a computer scripting language, originally designed for producing dynamic web pages. It is mainly used in server-side scripting, but can be used from a command line interface or in standalone graphical applications.

Keyword
Newton Raphson method, web based application, PHP language, and load flow

1.0 INTRODUCTION

The load flow study reveals the electrical performance and power flow (real and reactive) for specified conditions when the system is operating under steady-state [1]. Load flow study commonly used iterative techniques for the solution of non linear algebraic equation. These technique are applied to the practical systems using the develop software modules [2].

In power engineering, the power flow study (also known as load-flow study) is an important tool involving numerical analysis applied to a power system. Unlike traditional circuit analysis, a power flow study usually uses simplified notation such as a one-line diagram and per-unit system, and focuses on various forms of AC power (ie: reactive, real, and apparent) rather than voltage and current. It analyses the power systems in normal steady-state operation. There exist a number of software implementations of power flow studies.

In addition to a power flow study itself, many software implementations perform other types of analysis [3,4].

Since computer simulation was applied to the load flow analysis, many simulation methods are proposed. However, Newton-type load flow method is the most widely used because of its quadratic convergence characteristic and sparsity technique [5].

2.0 SCOPE OF WORK

2.1 3-BUS DIAGRAM

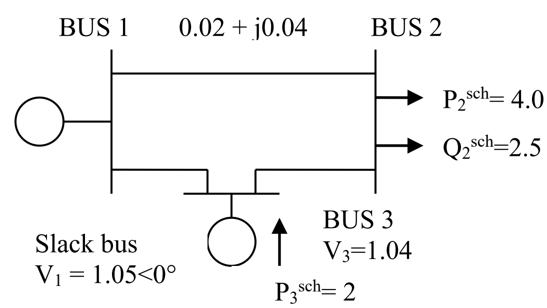


Figure 1: 3 bus diagram

2.2 LINE DATA

The data is written as a text file.

3 ,	number of total bus			
3 ,	number of total line			
1	3	0.01	0.03	0 ,
1	2	0.02	0.04	0 ,
2	3	0.0125	0.025	0 ,
Send	Receive	R	X	S

Where

R – Resistance
X – Reactance
S – Susceptance

2.3 BUS DATA

3, ←	number of total bus				
1	1	1.05	0	0	0 ,
2	2	1.0	0	-4	- 2.5 ,
3	0	1.04	0	2	0 ,
↑	↑	↑	↑	↑	↑
Bus num	Bus code	Volt mag	Volt angle	Psch	Qsch

From the data above, it shows that bus number 1 is a slack bus, bus number 2 is a load bus and bus number 3 is a generator bus.

3.0 NEWTON RAPHSON LOAD FLOW (NRLF) METHOD.

Step 1: form admittance matrix based on the data.

$$Y = \begin{bmatrix} Y_{11} & Y_{12} & \ldots\ldots & Y_{1n} \\ Y_{21} & . & & : \\ : & & . & : \\ : & & . & : \\ Y_{n1} & Y_{n2} & \ldots\ldots\ldots & Y_{nn} \end{bmatrix}$$

Step 2: P_i and Q_i

For an N-bus power system there will be n equations for real power injection Pi and n-equations for reactive power injection Qi .

Equation for real power injection Pi

$$P_i = \sum_{j=1}^{n} |V_i||V_j|Y_{ij} \cos(\delta_i - \delta_j - \theta_{ij}) \qquad \text{“Equation(1)”}$$

```
$p_jum = $volt[$s]*$volt[$r]*$Ymatrik[$s][$r]
         *cos($del_angle[$s] - $del_angle[$r]
         - $theta[$s][$r]);

$Pcal[$i] = $p_jum + $volt[$i]*$volt[$i]*
            $Ymatrik[$i][$i]*cos($theta[$i][$i]);
```

Figure 2: PHP fragment code for real power injection *P i*

Equation for reactive power injection *Q i* .

$$Q_i = -\sum_{j=1}^{n} |V_i||V_j|Y_{ij} \sin(\delta_i - \delta_j - \theta_{ij}) \qquad \text{“Equation(2)”}$$

```
$q_jum = $q_jum + $volt[$s]*$volt[$r]*
         $Ymatrik[$s][$r]*sin($del_angle[$s]-
         $del_angle[$r]-$theta[$s][$r]);

$Qcal[$i] = $q_jum + $volt[$i]*$volt[$i]*
            $Ymatrik[$i][$i]*sin($theta[$i][$i]);
```

Figure 3: PHP fragment code for reactive power injection *Q i* .

Step 3 : To find ΔP and ΔQ

Equation for ΔP.

$$\Delta P = P^{sch} - P_i^{(0)} \qquad \text{“Equation (3)”}$$

```
$dPcal[$i] = $Psch[$i]- $Pcal[$i];
```

Figure 4: PHP fragment code for ΔP.

Equation for ΔQ.

$$\Delta Q = Q^{sch} - Q_i^{(0)} \qquad \text{“Equation (4)”}$$

```
$dQcal[$i] = $Qsch[$i]- $Qcal[$i];
```

Figure 5: PHP fragment code for ΔQ

Step 4: Calculation of Jacobian matrix

$$\begin{bmatrix} H & J1 \\ L1 & N \end{bmatrix}$$

The elements of the Jacobian matrix can be calculated using the following equations

The off diagonal for element in H, J1, L1 and N as follows:

$$H \longrightarrow \frac{dP_i}{d\delta_j} = -|V_i||V_j|Y_{ij} \sin(\delta_i - \delta_j - \theta_{ij}) \quad i \,!= j$$

“Equation (5)”

$$J1 \longrightarrow \frac{dP_i}{dV_j} = |V_i|Y_{ij} \cos(\delta_i - \delta_j - \theta_{ij}) \quad i \,!= j$$

"Equation (6)"

L1 → $\frac{dQ_i}{d\delta_i} = -|V_i||V_j|Y_{ij} \cos(\delta_i - \delta_j - \theta_{ij})$

$i \mathrel{!}= j$

"Equation (7)"

N → $\frac{dQ_i}{dV_i} = -|V_i|Y_{ij} \sin(\delta_i - \delta_j - \theta_{ij})$

$i \mathrel{!}= j$

"Equation (8)"

```
$H[$i][$j] = - $volt[$i]*$volt[$j]*$Ymatrik
             [$i][$j] *sin($del_angle[$i] –
             $del_angle[$j] –$theta[$i][$j]);

$J1[$i][$j] = $volt[$i]*$Ymatrik[$i][$j]*cos
             ($del_angle[$i] - $del_angle[$j] -
             $theta[$i][$j]);

$L1[$i][$j]= - $volt[$i]*$volt[$j]*$Ymatrik
             [$i][$j]* cos($del_angle[$i] –
             $del_angle[$j] –$theta[$i][$j]);

$N[$i][$j] = - $volt[$i]*$Ymatrik[$i][$j]*sin
             ($del_angle[$i] - $del_angle[$j] -
             $theta[$i][$j]);
```

Figure 6: PHP fragment code for the off diagonal component for element in H, J1, L1 and N

The diagonal for element in H, J1, L1 and N equation are as follows:

H → $\frac{dP_i}{d\delta_i} = \sum_{j/=1} |V_i||V_j|Y_{ij} \sin(\delta_i - \delta_j - \theta_{ij})$

"Equation (9)"

J1 → $\frac{dP_i}{dV_i} = \sum_{j/=1} |V_j|Y_{ij} \cos(\delta_i - \delta_j - \theta_{ij}) + 2|V_j|Y_{ij} \cos(-\theta_{ij})$

"Equation (10)"

L1 → $\frac{dQ_i}{d\delta_i} = \sum_{j/=1} |V_i||V_j|Y_{ij} \cos(\delta_i - \delta_j - \theta_{ij})$

"Equation (11)"

N → $\frac{dQ_i}{dV_i} = -\sum_{j/=1} |V_j|Y_{ij} \sin(\delta_i - \delta_j - \theta_{ij}) - 2|V_j|Y_{ij} \sin(-\theta_{ij})$

"Equation (12)"

H component

```
$H[$i][$i]= $dp_jum_ang -$volt[$s]*$volt[$r]
            *$Ymatrik[$s][$r]*sin($del_angle
            [$s]-$del_angle[$r]- $theta[$s][$r]);
```

J1 component

```
$dp_jum_volt = $dp_jum_volt+$volt[$r]*
               $Ymatrik [$s][$r]*cos($del_angle
               [$s]-$del_angle[$r]-theta[$s][$r]);
$J1[$i][$i]= $dp_jum_volt + 2*$volt[$i]*$Ymatrik
             [$i][$i]*cos(-$theta[$i][$i]);
```

L1 component

```
$L1[$i][$i] = $volt[$s]*$volt[$r]*$Ymatrik
             [$s][$r] *cos($del_angle[$s] –
             $del_angle[$r]-$theta[$s][$r]);
```

N component

```
$dq_jum_volt = $dq_jum_volt + $volt[$r] *$Ymatrik
               [$s][$r]*sin($del_angle[$s]-
               $del_angle [$r]-$theta[$s][$r]);
$N[$i][$i] = $dq_jum_volt +2*$volt[$i]*$Ymatrik
             [$i][$i] *sin(-$theta[$i][$i]);
```

Figure 7: PHP fragment code for the diagonal component for element in H, J1, L1 and N

Step 5 : solving the Jacobian Matrix

To find the value of $d\delta_i$ and dV_i

$$\begin{bmatrix} dPi \\ dQi \end{bmatrix} = \begin{bmatrix} H & J1 \\ L1 & N \end{bmatrix} \begin{bmatrix} d\delta_i \\ dV_i \end{bmatrix}$$

Step 6 : New estimation for bus voltage

Equation for δ.

$\delta^{(k+1)} = \delta^{(k)} - d\delta^{(0)}$ "Equation (13)"

```
$del[$i] = $del[$i]+$delangle[$i];
```

Figure 8: PHP fragment code for delta.

Equation for voltage.

$V^{(k+1)} = V^{(k)} - dV^{(0)}$ "Equation (14)"

```
$volt[$i] = $volt[$i]+ $delvolt[$j];
```

Figure 9: PHP fragment code for voltage.

Step 7: Iteration Process

When the different of the value is 0.001, the iteration process is stopped and the final result of **δ** and **V** will be use for the next step.

Step 8: find the value of real and reactive power for slack and generator bus.

Equation of real power for slack bus Pi

$P_i = \sum_{j=1}^{n} |V_i||V_j|Y_{ij} \cos(\delta_i - \delta_j - \theta_{ij})$ "Equation (15)"

```
$p_jum =  $volt[$s]*$volt[$r]*$Ymatrik
              [$s][$r] *cos($del_angle[$s] -
              $del_angle[$r]  -$theta[$s][$r]);

$Pcal[$i] = $p_jum + $volt[$i]*$volt[$i]*
               $Ymatrik[$i][$i]*cos($theta[$i][$i]);
```

Figure 10: PHP fragment code for real power for slack bus

Equation of reactive power for slack and generator bus Qi.

$Q_i = -\sum_{j=1}^{n} |V_i||V_j|Y_{ij} \sin(\delta_i - \delta_j - \theta_{ij})$ "Equation(13)"

```
$q_jum = $q_jum + $volt[$s]*$volt[$r]*
             $Ymatrik[$s][$r]*sin($del_angle[$s]-
             $del_angle[$r]-$theta[$s][$r]);
$Qcal[$i] = $q_jum + $volt[$i]*$volt[$i]*
               $Ymatrik[$i][$i]*sin(-$theta[$i][$i]);
```

Figure 11: PHP fragment code for reactive power for slack bus and generator bus.

4.0 WEB BASED APPLICATION

The reason why PHP language is used to develop the web based system load flow (Newton-Raphson) is because the Web based application is reusable and can increase the efficiency of commercial offering of power system software.

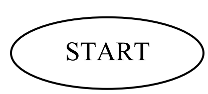
Web-based applications exist within the framework of the web browser itself. For desktop application, this can seem incredibly limiting. There are fewer controls available, it's not nearly as dynamic, that annoying Back button keeps cropping up, and you have to wait and wait for the server to display something new.

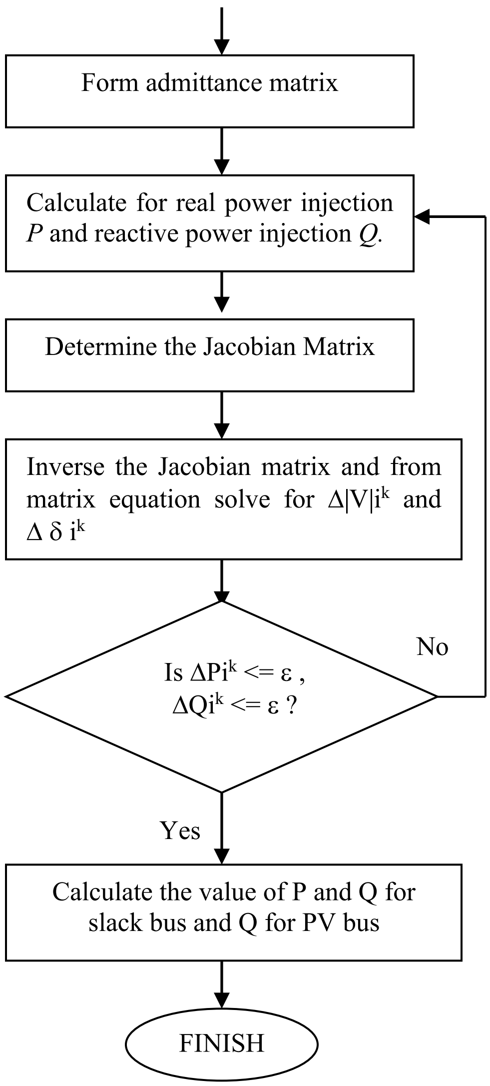
Of course, the platform is constantly evolving. The rapid proliferation of the Internet and the cost-effective growth of its key enabling technologies are revolutionizing information technology and creating unprecedented opportunities for developing large scale distributed applications. At the same time, there is a growing concern over the security of Web-based applications, which are rapidly being deployed over the Internet [6]. Unlike the desktop platform, which didn't change very much in the 15-20 years that it's been around.

Web application design is breaking some new ground in how we design applications. As designers, we have a lot more control over the presentation of information in our application and the presentation of the application as a whole. On the desktop, it was always battleship gray and tables always looked identical [7].

Web-based applications are programs that run on Web servers and use Web pages as the user interface. For the average user this new kind of software will be easier, cheaper, more mobile, more reliable, and often more powerful than desktop software.

5.0 METHODOLOGY





This project will start with the forming an admittance matrix from the data. It is very important thing that have to be done since the value of the admittance matrix will be used in the equation of P and Q.

The next step is to calculate the value of P and Q depends on the type of bus. If it is generator bus, the value of P and Q will be calculated. If it is load bus, the value of Q will be calculated.

Then, the Jacobian matrix is form depend on the size of the matrix. The inversion of the matrix will be done to get the value of $\Delta|V|i^k$ and $\Delta\,\delta\, i^k$.

To get the accuracy of the value calculated, the iteration process will be done until the residuals ΔPi^k and ΔQi^k are less than the specified accuracy.

After all the process is followed, the value of P and Q for slack bus and value Q of generator bus is determine.

6.0 RESULT AND DISCUSSION

The result from the data has been test with the program develop using php language. From the data above, the Y_{bus} matrix will perform:

Y_{bus} matrix

53.85<-1.19	22.36<2.03	31.62<1.89
22.36<2.03	58.14<-1.11	35.78<2.03
31.62<1.89	35.78<2.03	67.23<-1.17

Then the process of determining the type of bus by following the bus number and bus code of the bus which is

Bus number [1] is a slack bus
Bus number [2] is a generator bus (PV)
Bus number [3] is a load bus(PQ)

Since bus 2 is generator bus, ΔP and ΔQ will be calculated. And since bus 3 is load bus, only ΔP will take under consideration because the Q value is a known value. The result will be as follow:

ΔPcalculate[2] = - 2.95307848414
ΔQcalculate[2] = - 0.173774279543
ΔPcalculate[3] = 0.958604453442

Jacobian matrix

$$\begin{bmatrix} 54.4021 & -33.3563 & 24.8057 \\ -33.3563 & 66.1412 & -16.4933 \\ -26.8995 & 16.4933 & 49.7497 \end{bmatrix}$$

After the jacobian matrix is solved, the value of δ and V is calculated for the first iteration.

del_ang[2] = - 0.0547
del_ang[3] = - 0.0200
volt[2] = 0.9737
The process is continued until we get the different of the value is 0.001. After 4 iterations, the process is stop and the new voltage and angle that we get is:

new del_ang[2]=-0.05415
new del_ang[3]=-0.01937
new volt[2]= 0.97366

Then the value of P and Q for slack bus and Q for generator bus is calculated. The results are as follow:

Pcalculate[1]=1.0975 pu
Qcalculate[1]=1.8406 pu
Qcalculate[3]=1.6907 pu

6.0 FUTURE DEVELOPMENT

Users expect an application to be easy to use. The project can be display on its own website for end user. User can enter the data whenever or where ever they want. They also can send the data via email and get the answer immediately.

Recently, the data is store as a text file. For the future development, the data can be store using MySQL database which is more reliable for every user. It also can be link with the open source content management system, Joomla!.

CONCLUSION

The program of power flow (Newton Raphson method) has been developed and successfully running. This program is valid for whatever data been entered. From the program that has been developed using PHP language above, the objective of load flow is achieved. However, the value is different depend on the decimal point that we take.

REFERRENCES

[1] B.R. Gupta, S.Chand *"Power System Analysis and Design"*

[2] Prof. Hadi Saadat, *"Power System Analysis, 2nd Edition, Mc Graw Hill."*

[3] "*Power flow study: From Wikipedia, the free encyclopedia*" http://en.wikipedia.org/wiki/Power_flow_study#Power_Flow_Problem_Formulation

[4] Stephen J. Chapman, "*Electrical Machinery and Power System Fundamentals, Mc Graw Hill, International Edition."*

[5] J. B. Ward, H. W. Hale, *"Digital Computer Solutions of Power-Flow Problems," AIEE Trans. (Power App. Syst.), Vol.75, pp. 398-404, June 1956.*

[6] Garfinkel, S. and Spafford, E.H. "*Web Security and Commerce"*. O'Reilly and Associates, Sebastopol, CA, 1997.

[7] *"Web application"* http://en.wikipedia.org/wiki/Web_application