C++ project – power flow solver development based on Newton-Raphson optimization

- **Requirement**: develop a power flow solver for a three AC buses power system based on Newton-Raphson optimization theory. And print the bus voltage with phasor and PQ power value for testing purpose.
- **Conceptual design**:
  - Confirm mathematic theory -> design solver configuration and structure -> implement solver on C++ platform -> testing solver
- **Detailed design:**
  - Main mathematic theory used for the power flow solver development based on Newton-Raphson theory
    - Fundamental theory for NR to nonlinear algebra, assume:

$$f(x) = b$$

    - Then, the equation can be reorganized as following:

$$0 = b - f(x)$$

$$Ax = Ax + b - f(x)$$

$$x = x + A^{-1}\big(b - f(x)\big)$$

$$x^{n+1} = x^n + A^{-1}\big(b - f(x^n)\big) \qquad (1)$$

    - The equation $f(x) = b$ can be expanded through Taylor series expansion:

$$f(x) = b = f(x_0) + \frac{df(x)}{dx}|_{x_0}(x - x_0) + \cdots$$

    - By ignoring the high order series, the equation can be expressed as:

$$b = f(x_0) + \frac{df(x)}{dx}|_{x_0}(x - x_0)$$

$$(\frac{df(x)}{dx}|_{x_0})^{-1} b = (\frac{df(x)}{dx}|_{x_0})^{-1} f(x_0) + x - x_0$$

$$x = x_0 + \left(\frac{df(x)}{dx}|_{x_0}\right)^{-1}\big(b - f(x_0)\big) \qquad (2)$$

    - The above formula (2) represents the same format as formula (1), where x is the current state, the x_0 is the previous state, and $\left(\frac{df(x)}{dx}|_{x_0}\right)^{-1}$ is the Jacobian matrix of the previous variable value. This fundamental formula is mainly used to iterate and update the values of bus phase angle and the voltage during the power flow analysis that will be discussed below.
    - Since here, the power flow analysis will be discussed. The apparent power with phasor can be expressed as below:

$$\tilde{S} = \tilde{V}(\sum_{j=1}^{n} Y_j \tilde{V}_j)^* = P + jQ$$

- $\tilde{V}$ is the voltage with phasor, $Y_j = \frac{1}{jX_j}$ is the admittance matrix, and $\sum_{j=1}^{n} Y_j \tilde{V}_j$ is the related current. They can be expressed as below:

$$\tilde{V} = V(cos\delta + jsin\delta)$$

$$Y_j = G_j + jB_j$$

- Where G_j and B_j are associated with resistance and reactance separately. And the apparent power can be expressed as:

$$P + jQ = V(cos\delta + jsin\delta)[\sum_{j=1}^{n} (G_j + jB_j) V_j (cos\delta_j + jsin\delta_j)]^*$$

$$P = V \sum_{j=1}^{n} V_j (G_j \cos(\delta - \delta_j) + B_j \sin(\delta - \delta_j))$$

$$Q = V \sum_{j=1}^{n} V_j (G_j \sin(\delta - \delta_j) - B_j \cos(\delta - \delta_j))$$

- To obtain the accurate power values, the active power mismatches for every PV and PQ bus and reactive power mismatches for PQ bus need to be minimized under threshold. And the minimizing procedure is processed under NR optimization iteration.

$$\Delta P = P_{given} - P = P_{given} - V \sum_{j=1}^{n} V_j (G_j \cos(\delta - \delta_j) + B_j \sin(\delta - \delta_j))$$

$$\Delta Q = Q_{given} - Q = Q_{given} - V \sum_{j=1}^{n} V_j (G_j \sin(\delta - \delta_j) - B_j \cos(\delta - \delta_j))$$

- From Newton Raphson theory, the current state P and Q are updated through the update of phasor δ and voltage V. The iteration formula is described below:

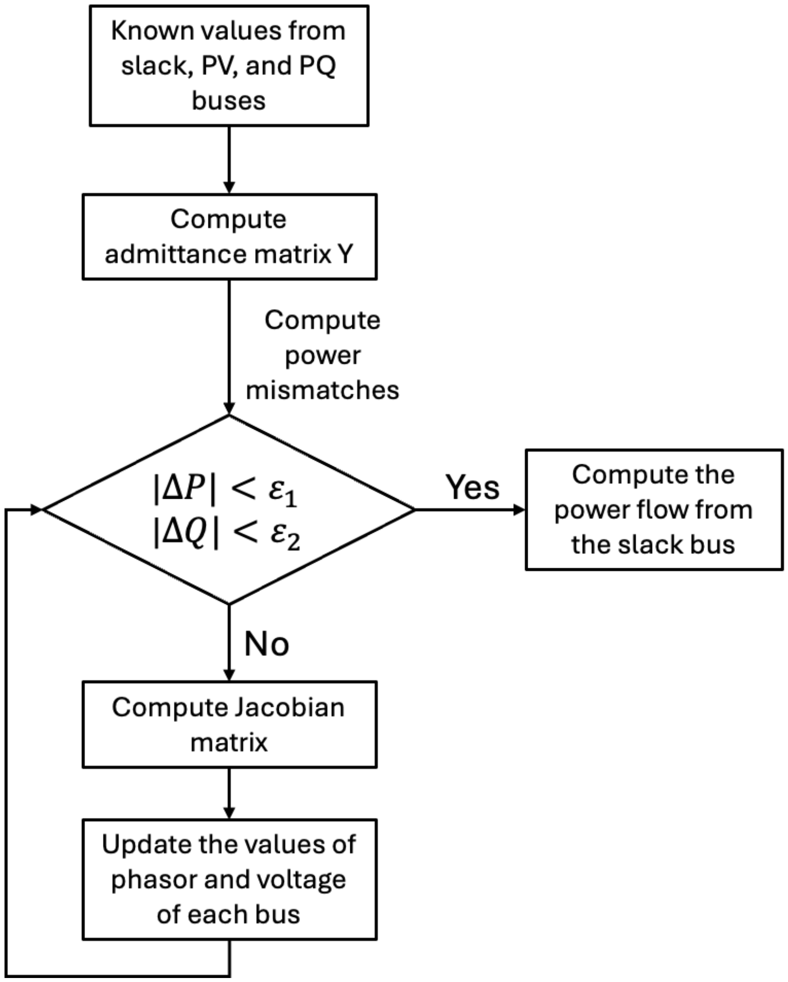
$$\begin{pmatrix} \delta^{n+1} \\ V^{n+1} \end{pmatrix} = \begin{pmatrix} \delta^{n} \\ V^{n} \end{pmatrix} + \begin{pmatrix} \frac{\partial f_p}{\partial \delta} & \frac{\partial f_p}{\partial V} \\ \frac{\partial f_q}{\partial \delta} & \frac{\partial f_q}{\partial V} \end{pmatrix}_{\delta^n, V^n}^{-1} \begin{pmatrix} P_{given} - f_p(\delta^n, V^n) \\ Q_{given} - f_q(\delta^n, V^n) \end{pmatrix}$$

- Where:

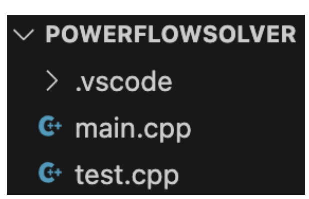
$$f_p = \sum_{j=1}^{n} V_j (G_j \cos(\delta - \delta_j) + B_j \sin(\delta - \delta_j))$$

$$f_q = \sum_{j=1}^{n} V_j(\, G_j \sin(\delta - \delta_j) - B_j \cos(\delta - \delta_j))$$

- Through above mathematic theory, the programmable flow of the NR AC power flow algorithm can be seen in the below graph:



- The algorithm implementation is performed with C++, and the folder structure is:
  - main.cpp: the power flow solver engine based on Newton-Raphson optimization algorithm.
  - test.cpp: provides a three buses power system example, and the related power flow test.



- The test result with per unit is shown as below:

```
Bus 1: |V|=1.00000  angle=0.00000 deg
  P=0.80000  Q=0.38651
Bus 2: |V|=0.95819  angle=-5.58564 deg
  P=-0.50000  Q=-0.19893
Bus 3: |V|=0.96494  angle=-4.95949 deg
  P=-0.30000  Q=-0.09900
```

- To validate the simulation result, the result can be compared with the manual calculation.

▪ The code package can be found under the Github repositories:

▪ https://github.com/Diwang0705/PowerSystemAnalysis/tree/main/PowerFlowSolver