

摘 要

基于 MATLAB 和 App Designer 的 UI 交互框架开发了一款电力系统潮流计算工具 PowerFlower, 实现了 PQ 分解法、牛顿-拉夫逊法的通用潮流计算、潮流结果报表、系统单线图可视化、连续潮流计算等功能, 数据输入格式和潮流结果报表方式与 PSASP 软件相似。在算法层面, 对 PQ 分解法中的连续求解线性方程组算法改进为 LU 分解法, 使得计算速度有一定程度提高。软件在 WSCC(Western Systems Coordinating Council)9 节点系统中验证了求解精度, 以 PSASP 的计算结果为参考, PowerFlower 的潮流计算结果平均误差在 $1e-4$ 数量级以内。

关键词: 潮流计算、PQ 分解法、牛顿拉夫逊法、MATLAB

目录

一、 软件功能	3
1.1 数据输入	3
1.2 潮流作业设置	5
1.3 潮流结果报表及可视化	6
1.4 连续潮流计算分析	6
二、 潮流计算原理	7
2.1 电力系统潮流的数学模型	7
2.2 潮流计算方法	8
2.2.1 牛顿-拉夫逊法	8
2.2.2 PQ 分解法	9
三、 软件设计思路	10
3.1 牛顿拉夫逊法潮流计算	10
3.2 PQ 分解法潮流计算、	10
3.3 节点导纳矩阵生成	11
3.3.1 自导纳计算	11
3.3.2 互导纳计算	11
3.3.3 考虑变压器对节点导纳矩阵的修正	11
3.4 数据预处理	12
3.5 结果可视化	12
3.6 连续潮流计算分析	12
3.7 UI 交互设计	12
四、 算例及测试结果	14
4.1 测试算例	14
4.2 测试结果	14
4.2.1 潮流求解精度对比	14
4.3 连续潮流分析	15
五、 评价与总结	17
六、 参考文献	17
附录 1 WSCC9 系统基础数据	18
附录 2 潮流计算所用函数	20
附录 3 PowerFlower 前端设计程序	33

一、软件功能

PowerFlow 是一款面向初学者的电力系统潮流仿真工具，数据输入格式和操作方法与 PSASP 电力系统分析软件类似，但功能更加精简，专注于电力系统潮流计算。目前软件可支持常规潮流计算（PQ 分解法、牛顿拉夫逊法）功能、潮流计算结果输出报表功能和连续潮流计算分析功能。目前软件已在 WSCC9 节点系统进行测试，潮流计算结果与 PSASP 相比，误差在 $1e-4 \sim 1e-5$ 数量级左右，此计算精度证明了软件后端代码的正确性。

软件后端基于 MATLAB 开发，自定义函数包括 PQ 分解法、牛顿拉夫逊法潮流计算函数、节点导纳矩阵生成函数、数据预处理函数等。软件前端采用 MATLAB App Designer 工具设计，利用系统控件回调函数实现 UI 交互。以下是通过 PowerFlow 实现潮流计算的详细操作流程。

1.1 数据输入

在软件主界面左侧栏可以输入系统数据信息，包括母线信息（母线名称、基准电压、节点类型）、交流线信息（交流线名称、I、J 两侧连接母线名称、交流线阻抗参数和对地导纳参数）、变压器信息（变压器名称、I、J 两侧连接母线名称、变压器变比、阻抗参数）、发电机信息（发电机名称、所连接母线名称、有功出力和机端电压）、负荷信息（负荷名称、所连接母线名称、有功和无功消耗）。此外，与 PSASP 软件不同的是，PowerFlow 在系统数据信息输入方面还包括转移节点信息，转移节点是一类没有注入功率的节点，一般是发电机升压变压器的高压侧母线，在 PowerFlow 的计算中视为 $P=0$ ， $Q=0$ 的 PQ 节点。

节点（母线）信息输入：在左下角输入母线名称、基准电压、节点类型，点击确定按钮，输入信息将在系统数据信息中的母线表格中呈现，可以检查输入是否正确。



图 1-1 母线信息输入

交流线信息输入：在左下角输入交流线名称、I、J 两侧连接母线名称、交流线阻抗参数和对地导纳参数，点击确定按钮，输入信息将在系统数据信息中的交流线表格中呈现，可以检查输入是否正确。当交流线信息输入后，在主界面中间的系统单线图区域将会呈现目前依据交流线信息（I、J 两侧节点）得到的节点对自动生成的系统单线图。



图 1-2 交流线信息输入

变压器信息输入：在左下角输入变压器名称、I、J 两侧连接母线名称、变压器变比和正序阻抗，点击确定按钮，输入信息将在系统数据信息中的变压器表格中呈现，可以检查输入是否正确。



图 1-3 变压器信息输入

发电机信息输入：在左下角输入发电机名称、连接母线名称、额定容量、有功发电、母线电压幅值和相角，点击确定按钮，输入信息将在系统数据信息中的发电机表格中呈现，可以检查输入是否正确。



图 1-4 发电机信息输入

负荷信息输入：在左下角输入负荷名称、连接母线名称、有功发电、无功发电，点击确定按钮，输入信息将在系统数据信息中的负荷表格中呈现，可以检查输入是否正确。



图 1-5 负荷信息输入

1.2 潮流作业设置

在软件主界面中间侧单线图下方可定义潮流作业，设置信息有计算电压上限、计算电压下限、计算方法、最大允许误差、最大迭代上限。当计算过程中电压超过上下限时，将停止迭代。计算方法可选择 PQ_DeCoupled（PQ 分解法）或 Newton（牛顿法），PQ 分解法的计算速度较快，但所用迭代次数相比于 Newton 法更多。其中最大允许误差可控制潮流计算结果的精度，最大允许误差越小，精度越高，但计算代价更大。当程序达到最大迭代次数上限时，如果计算结果仍不满足精度要求，则跳出迭代，认为此次潮流作业不收敛，数据存在较大的错误。

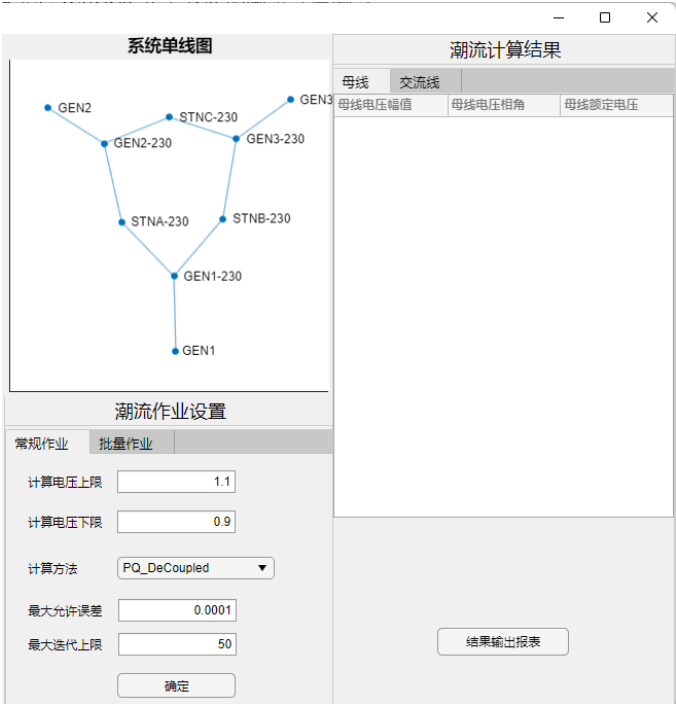


图 1-6 潮流作业设置

1.3 潮流结果报表及可视化

完成潮流作业设置后，选择确定按钮，即可开始潮流计算。若潮流计算结果正确收敛，则跳出弹框显示潮流计算用时，如图 1-7 所示，如果潮流计算未能正常收敛，迭代次数已达到最大迭代上限，则跳出警告弹窗，如图 1-8 所示。



图 1-7 潮流计算结果正常收敛

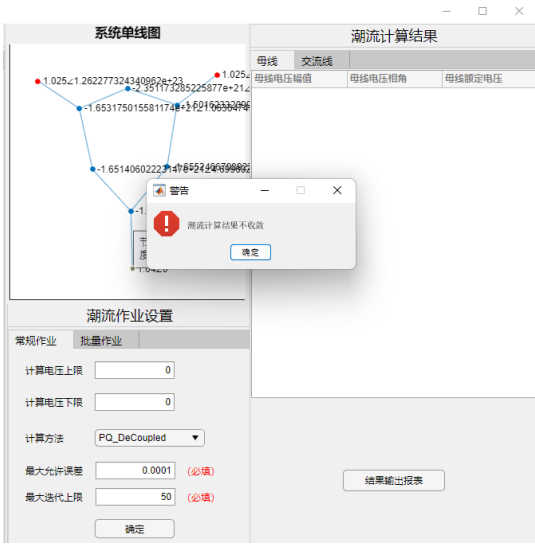


图 1-8 潮流计算结果不收敛

当潮流计算作业完成后，可选择主界面右侧结果输出报表按钮，潮流计算结果将在右侧表格中显示，目前仅设置了母线和交流线潮流报表。同时，节点（母线）电压幅值和相角计算结果将在主界面中间的系统单线图中呈现。

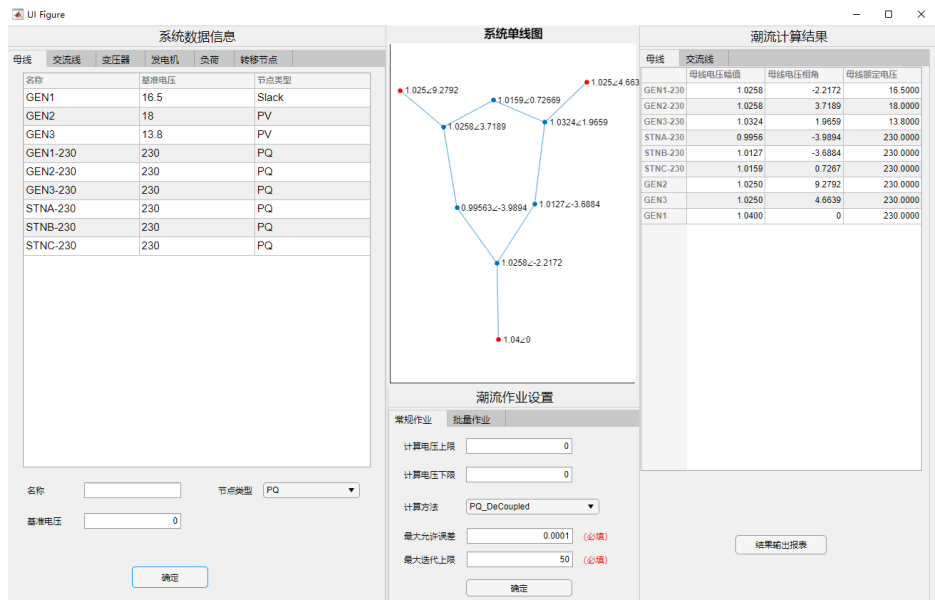


图 1-9 潮流计算结果

1.4 连续潮流计算分析

目前连续潮流计算暂未开发出交互功能，需要在软件后端代码中编程实现，具体过程为选择需要调节的元件（如某发电机或某负荷），设置仿真次数和单次调节步长，在 for 循环中反复调用潮流计算函数，储存每次计算结果，最后再调用 MATLAB 画图工具包绘制连续潮流结果。代码参照附录 2-Batch_Test.m

二、潮流计算原理

2.1 电力系统潮流的数学模型

电力系统潮流计算的数学模型如公式(1)~(2)所示

$$P_i = V_i \sum_{j=1}^n V_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \quad (1)$$

$$Q_i = V_i \sum_{j=1}^n V_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \quad (2)$$

已知系统中节点的信息有 $(P_1, P_2, \dots, P_{n-1}, Q_1, Q_2, \dots, Q_m, V_{m+1}, V_{m+2}, \dots, V_{n-1}, V_n, \theta_n)$ ，待求解的信息有 $(\theta_1, \theta_2, \dots, \theta_{n-1}, V_1, V_2, \dots, V_m)$ 其中前 m 号节点为 PQ 节点，第 $m \sim$ 第 $n-1$ 号节点为 PV 节点， n 号节点为平衡节点。对于求解(1)、(2)的非线性方程组，可采用牛顿拉夫逊法或者 PQ 分解法进行求解。

但同时，电网的运行数据是实时变化的，运行参数数据也是实时采样获得，因此通过调度自动化系统遥测得到的数据是不相容的，需要经过计算处理后得到电网的状态估计值。

状态估计的分析如下：

当系统的网络结构已知时，状态变量估计值和量测值之间的关系为：

$$Z = h(\hat{X}) + \omega \quad (3)$$

其中

$$Z = (P_1, P_2, \dots, P_{n-1}, Q_1, Q_2, \dots, Q_m, V_{m+1}, V_{m+2}, \dots, V_{n-1}, V_n, \theta_n) \quad (4)$$

$$\hat{X} = (\theta_1, \theta_2, \dots, \theta_{n-1}, V_1, V_2, \dots, V_m) \quad (5)$$

ω 为随机向量，为具有已知统计特性的测量误差

用加权最小二乘法进行状态估计的原理是求出状态变量 \hat{X} ，使得目标函数最小

$$\min_x J = \sum_{i=1}^m \frac{[Z_i - h_i(\hat{X})]^2}{\sigma_i^2} \quad (6)$$

极值点为

$$\frac{\partial J(\hat{X})}{\partial \hat{X}} = -2H^T R^{-1}[Z - h(\hat{X})] = 0 \quad (7)$$

对于方程(7)的求解可采用计算机算法中的雅可比迭代或高斯赛德尔迭代完成

由于潮流计算工具 PowerFlower 的设计目的是用于电力系统规划，不涉及实时潮流运算（因为 MATLAB 的解释型语言导致运算速度与 C/C++ 差距较大），因此在设计中没有采用状态估计的方法，而是进行方程组数量与未知数数量相等的潮流计算。

2.2 潮流计算方法

目前的潮流计算方法主要有牛顿-拉夫逊法和 PQ 分解法,也可以结合二者的特点,先进行 PQ 分解法迭代两次,再将结果送入牛顿-拉夫逊法迭代,可以得到较优计算速度和较精确的结果,也一定程度上避免了牛顿-拉夫逊法未能正常收敛的情况。

2.2.1 牛顿-拉夫逊法

对于上述非线性方程组(1)~(2),可以转换为式(8)所示的非线性方程组迭代求解

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = - \begin{bmatrix} H & N \\ K & L \end{bmatrix} \begin{bmatrix} \Delta \theta \\ V_{D2}^{-1} \Delta V \end{bmatrix} \quad (8)$$

其中 H、N、K、L 为雅可比矩阵的四个分块矩阵,可由迭代过程中的电压幅值和相角计算得出,为常量矩阵但实时变化。 (9)

方程(8)完成求解后,更新节点电压幅值和相角:

$$V_{k+1} = V_k + \Delta V$$

$$\theta_{k+1} = \theta_k + \Delta \theta$$

其中

$$\Delta P = \begin{bmatrix} \Delta P_1 \\ \Delta P_2 \\ \dots \\ \Delta P_{n-1} \end{bmatrix} \quad \Delta \theta = \begin{bmatrix} \Delta \theta_1 \\ \Delta \theta_2 \\ \dots \\ \Delta \theta_{n-1} \end{bmatrix} \quad (10)$$

$$\Delta Q = \begin{bmatrix} \Delta Q_1 \\ \Delta Q_2 \\ \dots \\ \Delta Q_m \end{bmatrix} \quad \Delta V = \begin{bmatrix} \Delta V_1 \\ \Delta V_2 \\ \dots \\ \Delta V_m \end{bmatrix} \quad (11)$$

$$V_{D2} = \begin{bmatrix} V_1 & & \\ & \dots & \\ & & V_m \end{bmatrix} \quad (12)$$

节点注入的有功功率和无功功率偏移量为:

$$\begin{aligned} \Delta P_i &= P_{is} - P_i = P_{is} - V_i \sum_{j=1}^n V_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \\ \Delta Q_i &= Q_{is} - Q_i = Q_{is} - V_i \sum_{j=1}^n V_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \end{aligned} \quad (13)$$

其中 P_{is} 和 Q_{is} 为初始给定的节点注入功率,为已知量, P_i 和 Q_i 为方程组(1)~(2)由当前节点电压幅值和相角计算出来的量,部分为未知量。

雅可比矩阵的计算方法如公式(14)~(15)所示:

当 $i \neq j$ 时

$$H_{ii} = V_i^2 B_{ii} + Q_i \quad H_{ij} = -V_i V_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij})$$

$$\begin{aligned}
N_{ij} &= -V_i V_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \\
K_{ij} &= V_i V_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \\
L_{ij} &= -V_i V_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij})
\end{aligned} \tag{14}$$

当 $i=j$ 时

$$\begin{aligned}
H_{ii} &= V_i^2 B_{ii} + Q_i \\
N_{ii} &= -V_i G_{ii}^2 - P_i \\
K_{ii} &= V_i^2 G_{ii} - P_i \\
L_{ii} &= V_i^2 B_{ii} - Q_i
\end{aligned} \tag{15}$$

2.2.2 PQ 分解法

当前通用的 PQ 分解法的修正方程式为,与牛顿拉夫逊法不同的是,在公式(8)中的高阶修正方程组拆分为两个低阶的线性方程组求解,且 PQ 分解法中的线性方程组系数不变,不用重复生成系数矩阵,因此 PQ 分解法的求解速度往往快于牛顿拉夫逊法。

$$\begin{aligned}
\frac{\Delta P}{V} &= B' \Delta \theta \\
\frac{\Delta Q}{V} &= B'' \Delta V
\end{aligned} \tag{16}$$

$$\begin{aligned}
V_{k+1} &= V_k + \Delta V \\
\theta_{k+1} &= \theta_k + \Delta \theta
\end{aligned} \tag{17}$$

其中

$$\left\{ \begin{aligned} B'_{ij} &= -\frac{X_{ij}}{R_{ij}^2 + X_{ij}^2} \\ B'_{ii} &= \sum_{\substack{j \in i \\ j \neq i}} \frac{X_{ij}}{R_{ij}^2 + X_{ij}^2} \end{aligned} \right\} \tag{18}$$

$$\left\{ \begin{aligned} B''_{ij} &= -\frac{1}{X_{ij}} \\ B''_{ii} &= \sum_{\substack{j \in i \\ j \neq i}} \frac{1}{X_{ij}} - b_{i0} \end{aligned} \right\} \tag{19}$$

三、 软件设计思路

3.1 牛顿拉夫逊法潮流计算

牛顿拉夫逊法的潮流计算主要由函数 `powerFlowNewtonCalcu.m` 实现，其中函数的输入包括总节点数量、PQ 节点数量、PV 节点数量、Slack 节点数量、节点电压初值、节点相角初值、节点导纳矩阵的实部虚部、初始给定有功和无功功率以及容许的最大误差，输出为节点电压赋值和相角的终值以及每次迭代过程中的节点电压赋值及相角的值，反映迭代过程。

程序思路为，首先进行迭代初值的初始化，给变量预分配内存，避免迭代过程中的动态内存分配，从而加快计算速度。同时给潮流偏移量分配一个较大的值，保证第一次能够进入循环语句。在循环中，首先根据公式(1)~(2)计算潮流偏移量 ΔP 和 ΔQ ，然后计算雅可比矩阵的四个分块，采用四个函数完成，分别为有功功率 P 对相角 θ 的求导 `diffP_sita.m`，有功功率 P 对电压幅值的求导 `diffP_V.m`，无功功率 Q 对相角 θ 的求导 `diffQ_sita.m`，无功功率 Q 对电压幅值 V 的求导 `diffQ_V.m`，计算过程参考公式(14)、(15)，详细程序见附录 2-3。通过雅可比矩阵将非线性方程组(1)~(2)转化为线性方程组(8)，可采用高斯消元法(`Gauss_solve.m`)求解节点电压幅值和相角偏移量 ΔV 和 $\Delta \theta$ ，再叠加到变量 V 和 θ 中，最后用其他变量储存本次迭代结果，进入下一次循环。在整个计算过程中，需要注意的是，明确角度的单位是弧度制还是角度制，因为 MATLAB 的内置数学计算函数 `sin/cos` 是采用弧度制为单位，若全程规定为角度制运算，则需要在进行三角函数计算前通过 `deg2rad` 函数进行变换。牛顿拉夫逊法详细计算程序参考附录 2-2。

3.2 PQ 分解法潮流计算、

PQ 分解法的潮流计算主要由函数 `powerFlowPQCalcu.m` 实现，其中函数的输入包括总节点数量、PQ 节点数量、PV 节点数量、Slack 节点数量、节点电压初值、节点相角初值、节点导纳矩阵的实部虚部、初始给定有功和无功功率以及容许的最大误差，输出为节点电压赋值和相角的终值以及每次迭代过程中的节点电压赋值及相角的值，反映迭代过程。

程序设计总体思路与牛顿拉夫逊法类似，不同的是，牛顿拉夫逊法需要求解的线性方程组的系数矩阵（雅可比矩阵）是随着迭代过程而变化的，PQ 分解法需要分别求解两组系数矩阵（ B' 、 B'' ）固定不变的线性方程组。由于需要反复求解系数矩阵固定的线性方程组，考虑算法速度的情况下，可以采用 LU 分解的方法提高算法速度。

对于求解线性方程组 $Ax=b$ ，LU 分解法的具体流程如下：

$$\begin{aligned} A &= LU \\ LUx &= b \\ Ux &= c, Lc=b \\ c &= L^{-1}b \\ x &= U^{-1}c = U^{-1}L^{-1}b \end{aligned}$$

参照文献[1]关于高斯消元法与 LU 分解法的时间复杂度探讨：对于具有 n 个方程 n 个未知数的方程组 $Ax=b$ ，高斯消元法完成一次求解运算大概需要进行 $2n^3/3$ 次（忽

略低阶项)操作,时间复杂度是 $O(n^3)$,而对于如下 k 个线性方程组,高斯消元法大约需要进行 $2kn^3/3$ 次操作,而LU分解法的近似操作次数为 $2n^3/3+2kn^2$,在求解多个 n 阶线性方程组时,LU分解的运算速度明显优于高斯消元法。因此在PQ分解法的计算中,采用LU分解能有效提高运算速度。

3.3 节点导纳矩阵生成

以上的两类潮流计算均基于节点导纳矩阵已知的情况,而节点导纳矩阵的生成需要经过繁琐的计算,计算的精度也会直接影响到最终潮流计算结果的准确性。作为一款潮流计算入门软件,让零基础的用户手动输入节点导纳矩阵显然对于大型电力系统仿真的情况是不现实的。参照PSASP软件的设计方式,我们希望通过手动输入母线、交流线、发电机、负荷、变压器等数据,自动地生成对应系统的节点导纳矩阵。

节点导纳矩阵生成主要由函数GenerateY.m完成,其中函数输入为节点(母线)信息、交流线信息、变压器信息,输出为节点导纳矩阵的实部和虚部。详细程序见附录2-3GenerateY.m

3.3.1 自导纳计算

在生成自导纳时,程序遍历每个节点,寻找每个节点所连接的所有交流线及交流线的电气参数信息,自导纳的计算公式为:

$$Y_{ii} = \sum_{j \neq i}^n A_{ij} \left(\frac{1}{Z_{ij}} + y_{ij} \right) \quad (20)$$

其中 A_{ij} 为邻接矩阵元素,若节点 i 与节点 j 相连,则 $A_{ij}=1$,否则 $A_{ij}=0$

3.3.2 互导纳计算

在计算互导纳时,由于节点之间相连接才存在互导纳,因此遍历每条交流线,寻找交流线 ij 两侧所对应的节点及交流线参数信息,生成互导纳。计算公式如(21)所示

$$Y_{ij} = A_{ij} \frac{1}{Z_{ij}} \quad (21)$$

其中 A_{ij} 为邻接矩阵元素,若节点 i 与节点 j 相连,则 $A_{ij}=1$,否则 $A_{ij}=0$

3.3.3 考虑变压器对节点导纳矩阵的修正

变压器支路的 π 型等值电路图如图3-1所示,考虑变压器支路的影响,需要在原有的节点导纳矩阵中做修改。

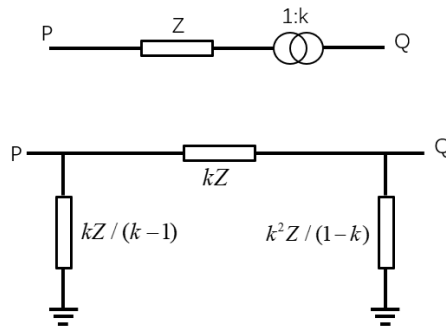


图3-1 变压器支路等值电路图

假设P节点或Q节点新增一条变压器支路P-Q(Q-P),则根据图可得出节点P

和节点 Q 的自导纳和互导纳增量为:

$$\begin{aligned}\Delta Y_{PP} &= \frac{1}{kz} + \frac{k-1}{kz} = \frac{1}{z} \\ \Delta Y_{QQ} &= \frac{1}{kz} + \frac{1-k}{k^2z} = \frac{1}{k^2z} \\ \Delta Y_{PQ} &= \Delta Y_{QP} = \frac{-1}{kz}\end{aligned}\tag{22}$$

则更新后的 P、Q 自导纳和互导纳为

$$\begin{aligned}Y_{PP}^{\text{new}} &= Y_{PP}^{\text{old}} + \Delta Y_{PP} \\ Y_{QQ}^{\text{new}} &= Y_{QQ}^{\text{old}} + \Delta Y_{QQ} \\ Y_{PQ}^{\text{new}} &= Y_{PQ}^{\text{old}} + \Delta Y_{PQ}\end{aligned}\tag{23}$$

3.4 数据预处理

数据预处理的作用是通过软件输入信息获得总节点数量、PQ 节点数量、PV 节点数量、Slack 节点数量、节点电压初值、节点相角初值信息,同时由于用户输入不能确保按照 PQ 节点、PV 节点、平衡节点的顺序输入,而在之后的潮流计算中又按照 PQ 节点、PV 节点、平衡节点的顺序形成修正方程式(8)和(16),因此还需要对用户输入的节点(母线)信息进行排序。详细程序参照附录 2-4, DataProcess.m

3.5 结果可视化

潮流计算结果包括节点电压幅值和相角信息、交流线上的有功潮流和无功潮流信息。可以采用报表输出和显示在单线图中两种方式进行展示。

系统单线图绘制采用 graph 函数,可以通过输入节点对构造图,也就是交流线的 I 侧连接母线和 J 侧连接母线的信息。在 graph 构造的图中可以对节点进行标注,从而可以用来显示节点电压幅值和相角信息。

同时,可以通过 MATLAB 内置的 writematrix 函数将节点电压幅值和相角信息写入 txt 文本文件。

3.6 连续潮流计算分析

连续潮流计算分析包括观察调整发电机有功出力、调整负荷有功或无功消耗等措施对潮流的影响。

由于单次潮流计算由函数 powerFlowPQCalcu 或 powerFlowNewtonCalcu 控制,因此只需要借助 for 循环,每次循环开始时按设置的步长更新系统参数,即可得到连续潮流结果。

连续潮流结果的绘制借助 matlab 的绘图工具包完成。

3.7 UI 交互设计

MATLAB AppDesigner 工具可以实现由源码到 App 的设计,从而使得程序具有可交互性。AppDesigner 是一种面向对象的开发方式,其中 UI 交互界面的设计可以通过 AppDesigner 软件左侧的组件以拖拽的方式形成软件主界面。当拖拽组件到主界面时, MATLAB 自动为该组件类创建了一个对象,一般会包含位置、字体和颜色、交互性等属性,可以通过编写回调函数(BackCall Function)实现交互功能。

PowerFlower 主界面设计如图 3-2 所示，包含四个部分，左侧为系统数据信息部分，实现系统初始潮流信息输入，中间一栏上半部分为系统单线图，待母线和交流线信息输入完成之后，系统单线图将会在其中呈现，下半部分为潮流作业设置，包括设置计算电压上下限、计算方法、最大允许误差和最大迭代上限。右侧一栏为潮流计算结果，通过“结果输出报表”按钮查看潮流计算结果。

系统数据信息

母线

交流线

变压器

发电机

负荷

转移节点

+

名称

基准电压

节点类型

名称

节点类型

基准电压

确定

系统单线图

潮流作业设置

常规作业

批量作业

+

计算电压上限

0

计算电压下限

0

计算方法

PQ_DeCoupled

最大允许误差

0

最大迭代上限

0

确定

潮流计算结果

母线

交流线

+

母线电压幅值

母线电压相角

母线潮流电压

结果输出报表

图 3-2 PowerFlower 主界面设计图

四、算例及测试结果

4.1 测试算例

本报告中采用 WSCC9 节点系统作为测试算例，将从计算精度、计算速度方面与 PSASP 软件计算结果对比，并借助 PowerFlower 进行 WSCC9 节点系统潮流分析。系统单线图如图 4-1 所示，详细数据见附录 1。

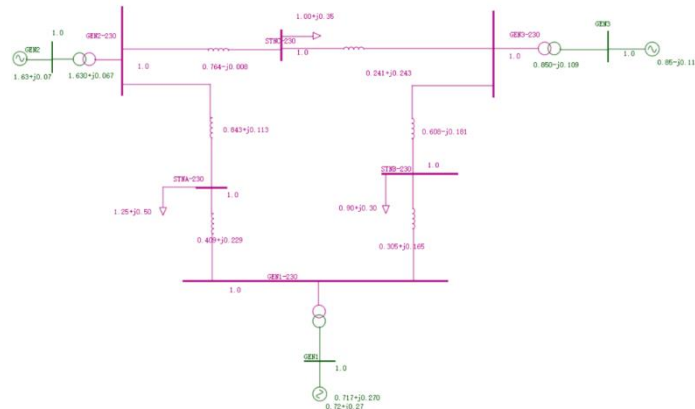


图 4-1 WSCC9 节点系统单线图

WSCC (Western Systems Coordinating Council) 9 节点系统是一常用的算例系统，为很多电力系统著作和文献所引用。该系统数据简单，用它作为算例，可以很快地熟悉和掌握 PSASP 的数据组织和录入，单线图和地理位置图的绘制以及潮流、短路、网损、最优潮流等各种计算的操作。

4.2 测试结果

4.2.1 潮流求解精度对比

分别在 PSASP 软件和 PowerFlower 软件进行 WSCC9 节点系统的潮流计算测试，计算方法均为 PQ 分解法，允许误差上限为 $1e-4$ ，电压上下限设置为 1.1 和 0.9，迭代次数上限为 50。测试结果如表 1 所示，误差以 PSASP 为基准，可以看出电压幅值和相角的最大误差数量级为 $1e-5$ ，基本与 PSASP 计算结果一致，从而证明了 PowerFlower 在潮流计算上满足精度要求。

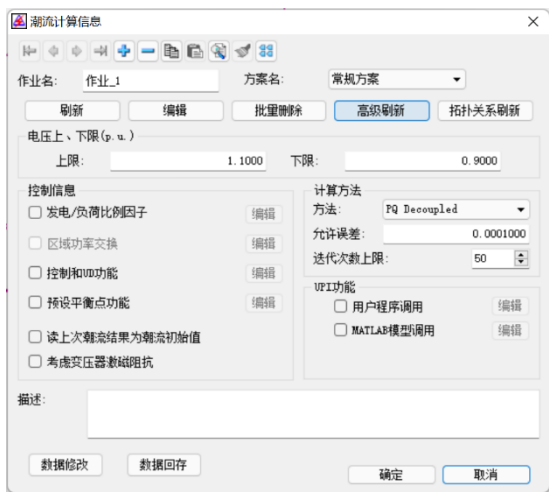


图 4-2 PSASP 潮流作业设置

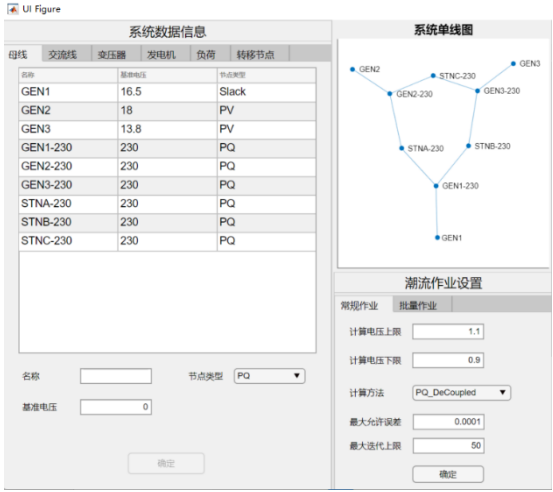


图 4-3 PowerFlower 潮流作业设置

表 1 PowerFlower 与 PSASP 软件潮流计算结果对比

母线名称	PSASP 结果		PowerFlower 结果		误差	
	电压幅值	电压相角	电压幅值	电压相角	电压幅值	电压相角
GEN1	1.04	0	1.04	0	0	0
GEN1-230	1.02579	-2.21714	1.0258	-2.2172	-9.7E-06	-2.7E-05
GEN2	1.025	9.27978	1.025	9.2792	0	6.25E-05
GEN2-230	1.02577	3.7194	1.0258	3.7189	-2.9E-05	0.000134
GEN3	1.025	4.66458	1.025	4.6639	0	0.000146
GEN3-230	1.03235	1.96648	1.0324	1.9659	-4.8E-05	0.000295
STNA-230	0.99563	-3.98924	0.9956	-3.9894	3.01E-05	-4E-05
STNB-230	1.01265	-3.68779	1.0127	-3.6884	-4.9E-05	-0.00017
STNC-230	1.01588	0.72715	1.0159	0.72669	-2E-05	0.000633

4.3 连续潮流分析

PowerFlower 相比于 PSASP 软件最大的优势在于能够进行连续潮流计算，分析发电机出力变化或者负荷功率变化曲线对系统潮流的影响。

考虑发电机有功输出对潮流的影响，例如改变发电机节点 GEN2 有功出力，以交流线 I 侧潮流参数为例，交流线有功潮流和无功潮流变化如图 4-4 所示。可以看出，发电机的有功出力在一定幅度内变化时，交流线潮流受影响程度不大，但各个线路情况不一致。本次分析中未考虑大范围变化时，线路容量过载的问题。

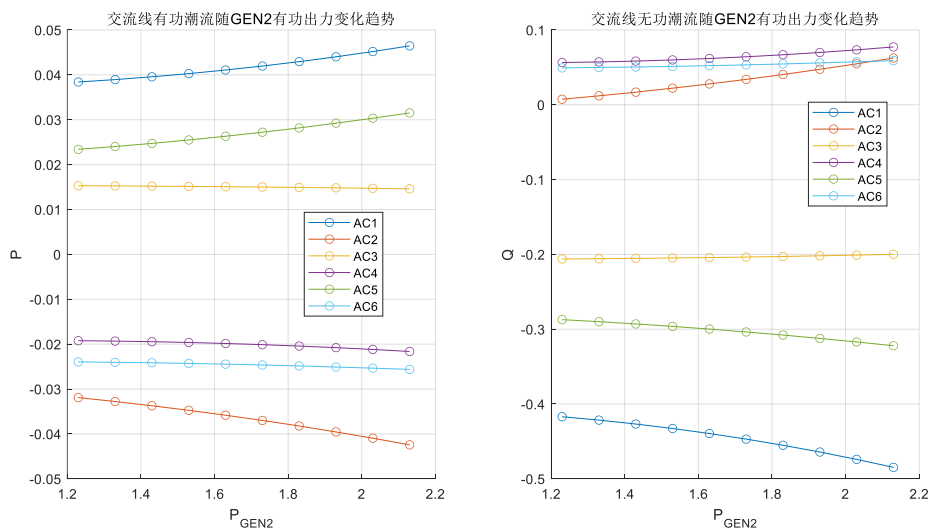


图 4-4 交流线潮流受给定节点有功输出影响情况

改变 GEN2 处的有功发电，固定用 PQ Decoupled-Newton 方法进行潮流计算，误差上限为 $1e-4$ ，节点电压相角变化结果如图 4-5 所示。可以明显看出，GEN2-230 母线的节点电压相角受 GEN2 有功出力影响最大，呈较高的曲率上升，可以明显地对应有功调频的原理。

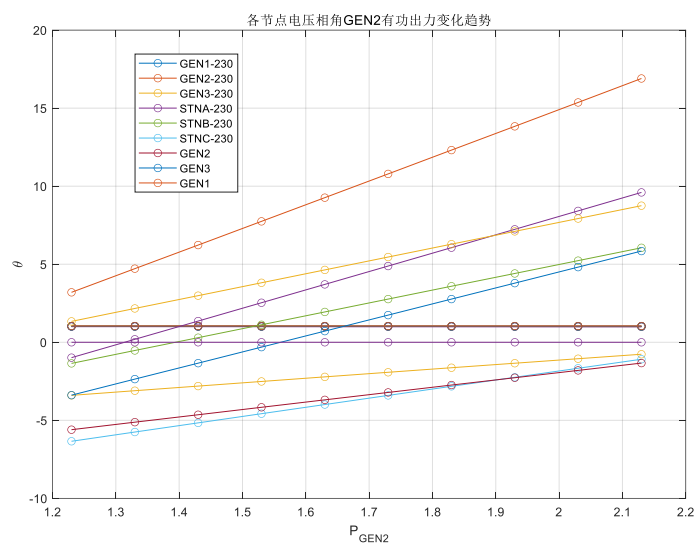


图 4-5 节点电压相角受给定节点有功输出影响情况

改变 GEN2 处的有功发电，固定用 PQ Decoupled-Newton 方法进行潮流计算，误差上限为 $1e-4$ ，节点电压幅值变化情况结果如图 4-6 所示。可以看出，除 STNA-230 节点外，其余节点受 GEN2 有功发电影响不大，但均有所波动，能够明显反映节点电压幅值主要由无功潮流决定，有功潮流对其影响不大，但有功调相与无功调压并不是完全的解耦合关系。

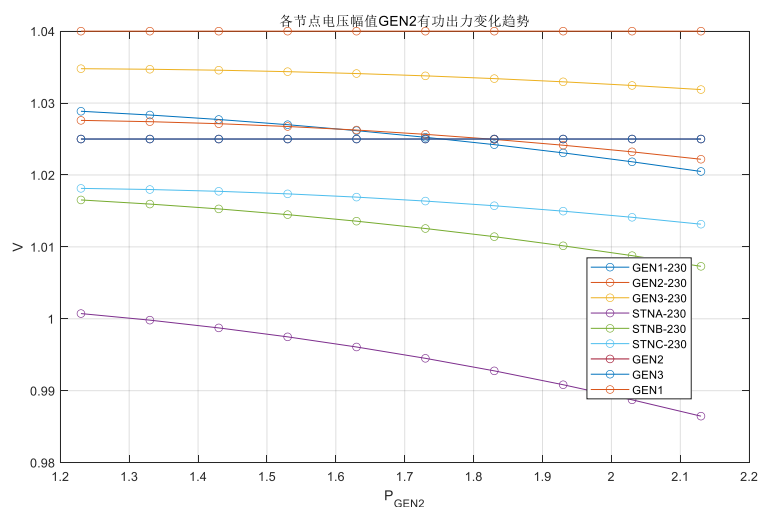


图 4-6 节点电压幅值受给定节点有功输出影响情况

五、评价与总结

软件基于 MATLAB 和 App Designer 开发，由于 MATLAB 采用解释型语言，程序运行时为逐行解释实时运算，在计算速度上逊色于 C/C++ 等编译型语言设计的潮流计算程序，但由于 App Designer 设计的软件具有出色的交互功能，相比于功能繁多、操作复杂的电力系统分析仿真软件 PSASP，此潮流计算工具 PowerFlower 能够作为初学者的入门软件，帮助初学者快速上手进行简单的潮流计算与分析。但比较遗憾的是，PowerFlower 目前并不能作为一款成熟的可供初学者使用的电力系统潮流计算软件，由于课程设计的时间原因，软件还有较多功能并未实现。例如从指定路径下的文件中读取系统数据，而不用手动输入，防止软件重启后的数据丢失。此外，PowerFlower 缺乏数据保存功能，未能在软件重启时读取上一次保存的工程文件，而是需要重新手动输入数据，这使得使用流程变得繁琐。此外，连续潮流作业功能由于待调节元件数目较多，未能想到合适的交互方法，开发此功能的工作量也较大。除了一些必要功能的缺失外，软件也有待开发暂态稳定分析、最优潮流计算等功能，成为一款多功能的电力系统分析软件。

六、参考文献

- [1]. Timothy Sauer. Numerical Analysis[M].2th Edition. Pearson Education,Inc,2014.

附录 1 WSCC9 系统基础数据

表 1 母线数据

母线名	基准电压	区域号	电压上限	电压下限	单相短路容量	三相短路容量
GEN1	16.5000	2	18.1500	14.8500	0.00000	0.00000
GEN2	18.0000	1	19.8000	16.2000	0.00000	0.00000
GEN3	13.8000	1	15.1800	12.4200	0.00000	0.00000
GEN1-230	230.0000	2	0.0000	0.0000	0.00000	0.00000
GEN2-230	230.0000	1	0.0000	0.0000	0.00000	0.00000
GEN3-230	230.0000	1	0.0000	0.0000	0.00000	0.00000
STNA-230	230.0000	2	0.0000	0.0000	0.00000	0.00000
STNB-230	230.0000	2	0.0000	0.0000	0.00000	0.00000
STNC-230	230.0000	1	0.0000	0.0000	0.00000	0.0000

表 2 交流线数据

数据组	I 侧母线	J 侧母线	编号	所 属 区 域	单 位	正序电阻	正序电抗	正序充电电 纳的 1/2
AC1	GEN1-230	STNA-230	1	I 侧	标么	0.010000	0.085000	0.088000
AC2	STNA-230	GEN2-230	2	I 侧	标么	0.032000	0.161000	0.153000
AC3	GEN2-230	STNC-230	3	I 侧	标么	0.008500	0.072000	0.074500
AC4	STNC-230	GEN3-230	4	I 侧	标么	0.011900	0.100800	0.104500
AC5	GEN3-230	STNB-230	5	I 侧	标么	0.039000	0.170000	0.179000
AC6	STNB-230	GEN1-230	6	I 侧	标么	0.017000	0.092000	0.079000

表 2（续） 交流线数据

零序电阻	零序电抗	零序充电电纳的 1/2
0.000000	0.255000	0.000000
0.000000	0.483000	0.000000
0.000000	0.216000	0.000000
0.000000	0.302400	0.000000
0.000000	0.510000	0.000000
0.000000	0.276000	0.000000

表 3 变压器数据

数据组	I 侧母线	J 侧母线	编号	连接方式	单 位	正 序 电 阻	正序电抗	零 序 电 阻	零序电抗
TRANS1	GEN1	GEN1-230	7	三角形/星 形接地	标么	0.000	0.05760	0.000	0.05760
TRANS2	GEN2	GEN2-230	8	三角形/星 形接地	标么	0.000	0.06250	0.000	0.06250
TRANS3	GEN3	GEN3-230	9	三角形/星 形接地	标么	0.000	0.05860	0.000	0.05860

表 3（续） 变压器数据

激磁电导	激磁电纳	变比	I 侧主抽 头电压	J 侧主抽 头电压	J 侧抽头 级差	J 侧抽头 位置	J 侧最大 抽头电压	J 侧最小 抽头电压
0.000	0.000	1.00	16.5	230.0	1.25	9	253.00	207.00
0.000	0.000	1.00	18.0	230.0	2.5	3	241.50	218.50
0.000	0.000	1.00	13.8	230.0	2.5	3	241.50	218.50

表 4 发电机数据

数据 组	母线 名	母线 类型	单位	额 定 容 量 (MVA)	有功发 电	无 功 发电	母 线 电 压幅值	母 线 电 压 相角	无 功 上限	无 功 下限	有 功 上限	有 功 下限
常规	发电 1	V0	标么	100.0	0.000	0.00	1.0400	0.00	0.00	0.00	0.00	0.00
常规	发电 2	PV	标么	100.0	1.630	1.00	1.0250	0.00	0.00	0.00	0.00	0.00
常规	发电 3	PV	标么	100.0	0.850	1.00	1.0250	0.00	0.00	0.00	0.00	0.00

表 4（续） 发电机数据

d 轴暂态电抗 Xd'	d 轴次暂态电抗 Xd''	负序电抗 X2	转子惯性时间常数 Tj(s)
0.0608	0.0608	0.0608	47.28
0.1198	0.1198	0.1198	12.8
0.1813	0.1813	0.1813	6.02

表 5 负荷数据

数据 组	母线名	编 号	母 线 类型	单位	有功负 荷	无功负 荷	母线电 压幅值	母 线 电 压 相角	无 功 上限	无 功 下限	有 功 上限	有 功 下限
常规	STNA-230	300	PQ	标么	1.250	0.500	0.000	0.00	0.00	0.00	0.00	0.00
常规	STNB-230	301	PQ	标么	0.900	0.300	0.000	0.00	0.00	0.00	0.00	0.00
常规	STNC-230	302	PQ	标么	1.000	0.350	0.000	0.00	0.00	0.00	0.00	0.00

附录 2 潮流计算所用函数

附录 2-1: powerFlowPQCalcu.m

作用：PQ 分解法潮流计算。主要输入：节点导纳矩阵 主要输出：节点电压幅值、相角及迭代中间值

```
function [V,sita,V_vis,sita_vis,Flag_success]=powerFlowPQCalcu(n,n_pq,n_pv,n_bal,V0,sita0,G,B,P_s,Q_s,eps,maxCount)
    %% 迭代初值初始化
    V=V0;
    sita=sita0;
    P=zeros(1,length(P_s));
    Q=zeros(1,length(Q_s));
    temp1=0;temp2=0;
    deltaP=100;
    deltaQ=100;
    eps=1e-3;
    B1=B(1:n-n_bal,1:n-n_bal);
    B2=B(1:n_pq,1:n_pq);
    k=1;%迭代计数器
    Flag_success=1;
    %% LU 分解
    [L1,U1]=lu(B1);
    [L2,U2]=lu(B2);
    %% 迭代
    while max(abs([deltaP;deltaQ]))>eps
        if k>maxCount %如果超过了迭代上限
            Flag_success=0;
            break; %终止循环
        end
        %计算潮流偏移量
        for i=1:n
            for j=1:length(V)
                temp1=temp1+V(j)*(G(i,j)*cos(sita(i)-sita(j))+B(i,j)*sin(sita(i)-sita(j)));
            end
            P(i)=V(i)*temp1;
            temp1=0;
        end
        for i=1:n
            for j=1:length(V)
                temp2=temp2+V(j)*(G(i,j)*sin(sita(i)-sita(j))-B(i,j)*cos(sita(i)-sita(j)));
            end
```

```

        Q(i)=V(i)*temp2;
        temp2=0;
    end
    deltaP=P_s'-P(1:n_pq+n_pv)';
    deltaQ=Q_s'-Q(1:n_pq)';
    %迭代求解线性方程组
    c1=Gauss_solve(L1,deltaP./V(1:n-n_bal)');
    deltaSita=-Gauss_solve(U1,c1);
    c2=Gauss_solve(L2,deltaQ./V(1:n_pq)');
    deltaV=-Gauss_solve(U2,c2);
    %deltaSita=-Gauss_solve(B1,deltaP./V(1:n-n_bal)');
    %deltaV=-Gauss_solve(B2,deltaQ./V(1:n_pq)');
    %更新节点电压相角
    for i=1:length(deltaSita)                %避开广播机制
        sita(i)=sita(i)+deltaSita(i);
    end
    for j=1:length(deltaV(:,1))
        V(j)=V(j)+deltaV(j,1);
    end
    V_vis(k,:)=V(1,:);
    sita_vis(k,:)=sita(1,:);
    k=k+1;
end
sita=rad2deg(sita);
sita_vis=rad2deg(sita_vis);
end

```

附录 2-2: powerFlowNewtonCalcu.m

作用：牛顿拉夫逊法潮流计算。主要输入：节点导纳矩阵 主要输出：节点电压幅值、相角及迭代中间值

```

%迭代初值初始化
V=V0;
sita=sita0;
P=zeros(1,length(P_s));
Q=zeros(1,length(Q_s));
temp1=0;temp2=0;
deltaP=100;
deltaQ=100;
k=1; %计数器
Flag_success=1;
%% 牛顿迭代
while max(abs([deltaP;deltaQ]))>eps
    if k>maxCount %如果超过了迭代上限

```

```

        Flag_success=0;
        break;    %终止循环
    end
    %计算潮流偏移量
    for i=1:n
        for j=1:length(V)
            temp1=temp1+V(j)*(G(i,j)*cos(sita(i)-sita(j))+B(i,j)*sin(sita(i)-
sita(j)));
        end
        P(i)=V(i)*temp1;
        temp1=0;
    end
    for i=1:n
        for j=1:length(V)
            temp2=temp2+V(j)*(G(i,j)*sin(sita(i)-sita(j))-B(i,j)*cos(sita(i)-
sita(j)));
        end
        Q(i)=V(i)*temp2;
        temp2=0;
    end
    deltaP=P_s'-P(1:n_pq+n_pv)';
    deltaQ=Q_s'-Q(1:n_pq)';
    %计算雅可比矩阵
    H=diffP_sita(P,Q,V,sita,G,B,n_pq,n_pv,n_bal);
    N=diffP_V(P,Q,V,sita,G,B,n_pq,n_pv,n_bal);
    K=diffQ_sita(P,Q,V,sita,G,B,n_pq,n_pv,n_bal);
    L=diffQ_V(P,Q,V,sita,G,B,n_pq,n_pv,n_bal);
    J=[H,N;K,L];

    %求解线性方程组 高斯消元法
    b=[deltaP;deltaQ];
    X=Gauss_solve(-J,b);
    %更新数据
    deltaSita=X(1:n_pq+n_pv);
    deltaV=X(n_pq+n_pv+1:length(X))./V(1:n_pq);
    for i=1:length(deltaSita) %避开广播机制
        sita(i)=sita(i)+deltaSita(i);
    end
    for j=1:length(deltaV(:,1))
        V(j)=V(j)+deltaV(j,1);
    end
    %保存迭代结果
    V_vis(k,:)=V(1,:);
    sita_vis(k,:)=sita(1,:);

```

```

        %J_vis(k)=J;
        k=k+1;
    end
    sita=rad2deg(sita);
    sita_vis=rad2deg(sita_vis);
end

```

附录 2-3: GenerateY.m

作用：生成节点导纳矩阵 主要输入：系统参数 输出：节点导纳矩阵实部虚部

```

function [G,B]=GenerateY(Nodes,AC_I,AC_J,AC_Z,AC_y,Trans_I,Trans_J,Trans_k)
Y=zeros(length(Nodes),length(Nodes));
% 生成节点导纳矩阵
%计算自导纳
for i=1:length(Nodes) %遍历每个节点
    index1=find(AC_I==Nodes(i));
    index2=find(AC_J==Nodes(i));
    for k=1:length(index1)
        index_trans=find(Trans_I==Nodes(i));
        if isempty(index_trans)==1 %不含变压器
            Y(i,i)=Y(i,i)+1/AC_Z(index1(k))+AC_y(index1(k));
        else %如果当前支路含变压器且当前节点是在 k 侧
            Y(i,i)=Y(i,i)+1/(Trans_k(index_trans)^2*AC_Z(index1(k))+AC_y(index1(k));
        end
    end
    for k=1:length(index2)
        index_trans=find(Trans_J==Nodes(i));
        if isempty(index_trans)==1 %不含变压器
            Y(i,i)=Y(i,i)+1/AC_Z(index2(k))+AC_y(index2(k));
        else %如果当前支路含变压器且当前节点是在 k 侧
            Y(i,i)=Y(i,i)+1/(Trans_k(index_trans)^2*AC_Z(index2(k))+AC_y(index2(k));
        end
    end
end

%计算互导纳
for i=1:length(AC_I) %遍历所有的边
    index3=find(Nodes==AC_I(i)); %找到当前索引 i 下对应哪两个节点
    index4=find(Nodes==AC_J(i));
    %一个变压器对一个发电机和母线 不会出现一个节点对两条变压器的情况
    index3_transI=find(Trans_I==Nodes(index3));
    index3_transJ=find(Trans_J==Nodes(index3));
    %一个节点两侧都不含变压器支路，一定不含变压器

```

```

        if isempty(index3_transI)==1&&isempty(index3_transJ)==1
            Y(index3,index4)=-1/AC_Z(i);
            Y(index4,index3)=-1/AC_Z(i);
        elseif isempty(index3_transI)==0&&isempty(index3_transJ)==1 %如果节点在变
        压器 I 侧
            for j=1:length(index3_transI) % 寻找在 J 侧的节点是否有 index4 号
                if Trans_J(index3_transI(j))==Nodes(index4) %如果找到了
                    K=Trans_k(index3_transI(j));%计算变比
                    Y(index3,index4)=-1/(K*AC_Z(i));
                    Y(index4,index3)=Y(index3,index4);
                    break;
                end
                %如果没找到 说明此次检索的支路不含变压器
            if j==length(index3_transI)
                Y(index3,index4)=-1/(AC_Z(i));
                Y(index4,index3)=Y(index3,index4);
            end
        end
    elseif isempty(index3_transJ)==0&&isempty(index3_transI)==1
        for j=1:length(index3_transJ) % 寻找在 J 侧的节点是否有 index4 号
            if Trans_I(index3_transJ(j))==Nodes(index4) %如果找到了
                K=Trans_k(index3_transJ(j));%计算变比
                Y(index3,index4)=-1/(K*AC_Z(i));
                Y(index4,index3)=Y(index3,index4);
                break;
            end
            %如果没找到 说明此次检索的支路不含变压器
        if j==length(index3_transJ)
            Y(index3,index4)=-1/(AC_Z(i));
            Y(index4,index3)=Y(index3,index4);
        end
    end
end
end
end
%Y(3,3)=Y(3,3)+0.04j;
G=real(Y);B=imag(Y);
end

```

附录 2-4: DataProcess.m

作用：数据预处理 输入：母线、发电机、负荷、转移节点信息 输出：重排序后的节点、初始潮流、节点电压幅值和相角初值、总节点数量、PQ、PV、Slack 节点数量

function


```

[Nodes,P_s,Q_s,V0,theta0,n,n_pq,n_pv,n_bal]=DataProcess(Nodes_old,NodeType_old,Load,LD_P,LD_Q,transNodes,Gen,GE_P,GE_V,GE_theta)

%节点数
n=length(Nodes_old);%总节点数
%考虑转移节点数量
if transNodes==" "
    n_pq=length(Load);%PQ 节点数
else
    n_pq=length(Load)+length(transNodes);%PQ 节点数
end

n_pv=length(Gen)-1;%PV 节点数
n_bal=1;%平衡节点数

%预赋值
V0=ones(1,n);
theta0=zeros(1,n);
P_s=[];
Q_s=[];

%节点顺序替换 按 PQ PV SLACK 重新整理节点表格
index_PQ=find(NodeType_old=="PQ");
index_PV=find(NodeType_old=="PV");
index_Slack=find(NodeType_old=="Slack");
Nodes=[Nodes_old(index_PQ),Nodes_old(index_PV),Nodes_old(index_Slack)];

%各个节点的注入功率
for i=1:n_pq %遍历新的节点表格 Nodes 赋值所有的 PQ 节点
    index1=find(Load==Nodes(i));
    index2=find(transNodes==Nodes(i));
    if isempty(index1)==1&&isempty(index2)==0 %如果这个节点是为转移节点
        P_s=[P_s,0];
        Q_s=[Q_s,0];
    elseif isempty(index1)==0&&isempty(index2)==1 %如果这个节点是为负荷节点
        P_s=[P_s,LD_P(index1)];
        Q_s=[Q_s,LD_Q(index1)];
    else
        %其他情况说明没有负荷节点
    end
end

for i=n_pq+1:n_pq+n_pv %赋值 PV 节点
    ;

```

```

        index=find(Gen==Nodes(i));
        P_s=[P_s,GE_P(index)];
        V0(i)=GE_V(index);
    end
    index_temp=find(Gen==Nodes(length(Nodes))); %经过重排序后，Nodes 的最后一个
    节点为 SLACK
    theta0(length(Nodes))=GE_theta;
    V0(length(Nodes))=GE_V(index_temp);

```

附录 2-5: diffP_sita.m/diffP_V.m/diffQ_sita.m/diffQ_V.m

作用：雅可比矩阵四个分块矩阵生成

```

function H=diffP_sita(P,Q,V,sita,G,B,n_pq,n_pv,n_bal)
    n=n_pq+n_pv+n_bal;
    H=zeros(n-n_bal,n-n_bal);
    for i=1:n-n_bal
        for j=1:n-n_bal
            if i==j
                H(i,j)=(V(i)^2)*B(i,j)+Q(i);
            else
                H(i,j)=-V(i)*V(j)*(G(i,j)*sin(sita(i)-sita(j))-B(i,j)*cos(sita(i)-sita(j)));
            end
        end
    end
end

function N=diffP_V(P,Q,V,sita,G,B,n_pq,n_pv,n_bal)
    n=n_pq+n_pv+n_bal;
    N=zeros(n-n_bal,n_pq);
    for i=1:n-n_bal
        for j=1:n_pq
            if i==j
                N(i,j)=-(V(i)^2)*G(i,j)-P(i);
            else
                N(i,j)=-V(i)*V(j)*(G(i,j)*cos(sita(i)-sita(j))+B(i,j)*sin(sita(i)-
sita(j)));
            end
        end
    end
end

function K=diffQ_sita(P,Q,V,sita,G,B,n_pq,n_pv,n_bal)
    n=n_pq+n_pv+n_bal;
    K=zeros(n_pq,n-n_bal);
    for i=1:n_pq
        for j=1:n-n_bal

```

```

        if i==j
            K(i,j)=(V(i)^2)*G(i,j)-P(i);
        else
            K(i,j)=V(i)*V(j)*(G(i,j)*cos(sita(i)-sita(j))+B(i,j)*sin(sita(i)-sita(j)));
        end
    end
end
end
function L=diffQ_V(P,Q,V,sita,G,B,n_pq,n_pv,n_bal)
    L=zeros(n_pq,n_pq);
    for i=1:n_pq
        for j=1:n_pq
            if i==j
                L(i,j)=(V(i)^2)*B(i,j)-Q(i);
            else
                L(i,j)=-V(i)*V(j)*(G(i,j)*sin(sita(i)-sita(j))-B(i,j)*cos(sita(i)-sita(j)));
            end
        end
    end
end
end
end

```

附录 2-6: WSCC9_Test.m

说明: WSCC9 节点系统潮流计算测试

```

clear

%数据输入
% 母线（节点）信息
Nodes_old=["GEN1","GEN2","GEN3","GEN1-230","GEN2-230","GEN3-230","STNA-230","STNB-230","STNC-230"]; %名称
NodeType_old=["Slack","PV","PV","PQ","PQ","PQ","PQ","PQ","PQ"]; %节点类型

%负荷信息
Load=["STNA-230","STNB-230","STNC-230"]; %负荷对应母线
LD_P=[-1.25,-0.9,-1.00]; %有功负荷
LD_Q=[-0.5,-0.3,-0.35]; %无功负荷

%转移节点信息
transNodes=["GEN1-230","GEN2-230","GEN3-230"];

%发电机信息
Gen=["GEN1","GEN2","GEN3"]; %发电机对应母线
GE_P=[0,1.63,0.85]; %有功发电
GE_V=[1.04,1.025,1.025]; %母线电压幅值

```

```

GE_theta=[0];    %母线电压相角

%交流线信息
AC_I=["GEN1-230","STNA-230","GEN2-230","STNC-230","GEN3-230","STNB-230","GEN1","GEN2","GEN3"]; %I 侧节点
AC_J=["STNA-230","GEN2-230","STNC-230","GEN3-230","STNB-230","GEN1-230","GEN1-230","GEN2-230","GEN3-230"]; %J 侧节点
AC_Z=[0.01+0.085j,0.032+0.161j,0.0085+0.072j,0.0119+0.1008j,0.039+0.17j,0.017+0.092j,0.0576j,0.0625j,0.05860j]; %I-J 阻抗
AC_y=[0.088j,0.153j,0.0745j,0.1045j,0.179j,0.079j,0,0,0]; %I-J 对地导纳参数

%变压器信息
Trans_I=["GEN1","GEN2","GEN3"]; %k 侧节点
Trans_J=["GEN1-230","GEN2-230","GEN3-230"]; %非 k 侧节点
Trans_k=[1,1,1]; %变比

%潮流计算
%节点数据重排序 生成标准格式
[Nodes,P_s,Q_s,V0,theta0,n,n_pq,n_pv,n_bal]=DataProcess(Nodes_old,NodeType_old,Load,LD_P,LD_Q,transNodes,Gen,GE_P,GE_V,GE_theta);
%生成节点导纳矩阵
[G,B]=GenerateY(Nodes,AC_I,AC_J,AC_Z,AC_y,Trans_I,Trans_J,Trans_k);
Y=G+j*B;
eps=1e-4;
%迭代计算 并计时
tStart=tic;
[V,sita,V_vis,sita_vis]=powerFlowPQCalcu(n,n_pq,n_pv,n_bal,V0,theta0,G,B,P_s,Q_s,eps);
tEnd=toc(tStart);
%计算线路潮流
for j=1:length(AC_I)
    indexI=find(Nodes==AC_I(j)); %当前在 I 侧遍历的节点对应 Nodes 的位置
    indexJ=find(Nodes==AC_J(j)); %当前在 J 侧遍历的节点对应 Nodes 的位置

    F_AC(:,j)=((V(indexI)-V(indexJ))/AC_Z(j))-V(indexI)*AC_y(j);
end

%结果输出
G=graph(AC_I,AC_J);

```

```

subplot(1,2,1)
H1=plot(G)%展示系统单线图
labeledge(H1,1:numedges(G),AC_Z);
highlight(H1,[7 8 9],'NodeColor','r');
title("系统单线图")

subplot(1,2,2)
H2=plot(G);
highlight(H2,[7 8 9],'NodeColor','r');
for i=1:length(Nodes)
    labelnode(H2,Nodes(i),strcat(num2str(V(i)),"∠",num2str(sita(i))))
end
title("潮流计算")

tEnd
writematrix([Nodes;V;sita],"result.txt");

```

附录 2-7: Batch_Test.m

说明：连续潮流计算测试，此文件是调节发电机 GEN2 的有功输出，观察系统潮流变化情况。

```

clear

数据输入
% 母线（节点）信息
Nodes_old=["GEN1","GEN2","GEN3","GEN1-230","GEN2-230","GEN3-230","STNA-230","STNB-230","STNC-230"]; %名称
NodeType_old=["Slack","PV","PV","PQ","PQ","PQ","PQ","PQ","PQ"]; %节点类型

%负荷信息
Load=["STNA-230","STNB-230","STNC-230"]; %负荷对应母线
LD_P=[-1.25,-0.9,-1.00]; %有功负荷
LD_Q=[-0.5,-0.3,-0.35]; %无功负荷

%转移节点信息
transNodes=["GEN1-230","GEN2-230","GEN3-230"];

%发电机信息
Gen=["GEN1","GEN2","GEN3"]; %发电机对应母线
GE_P=[0,1.63,0.85]; %有功发电
GE_V=[1.04,1.025,1.025]; %母线电压幅值
GE_theta=[0]; %母线电压相角

%交流线信息
AC_I=["GEN1-230","STNA-230","GEN2-230","STNC-230","GEN3-230","STNB-

```

```

230","GEN1","GEN2","GEN3"]; %I 侧节点
AC_J=["STNA-230","GEN2-230","STNC-230","GEN3-230","STNB-230","GEN1-
230","GEN1-230","GEN2-230","GEN3-230"]; %J 侧节点
AC_Z=[0.01+0.085j,0.032+0.161j,0.0085+0.072j,0.0119+0.1008j,0.039+0.17j,0.017+0.0
92j,0.0576j,0.0625j,0.05860j]; %I-J 阻抗
AC_y=[0.088j,0.153j,0.0745j,0.1045j,0.179j,0.079j,0,0,0]; %I-J 对地导纳参数

%变压器信息
Trans_I=["GEN1","GEN2","GEN3"]; %k 侧节点
Trans_J=["GEN1-230","GEN2-230","GEN3-230"]; %非 k 侧节点
Trans_k=[1,1,1]; %变比
潮流计算
m=20; %总共计算次数
steps=0.025;%步长
%选择调节的因素
Trans_k(1)=1;

for i=1:m
    %调整发电机的有功输出
    GE_V(2)= GE_V(2)-steps;
    %记录自变量数据
    temp(i)= GE_V(2);
    %节点数据重排序 生成标准格式

[Nodes,P_s,Q_s,V0,theta0,n,n_pq,n_pv,n_bal]=DataProcess(Nodes_old,NodeType_old,L
oad,LD_P,LD_Q,transNodes,Gen,GE_P,GE_V,GE_theta);
    %生成节点导纳矩阵
    [G,B]=GenerateY(Nodes,AC_I,AC_J,AC_Z,AC_y,Trans_I,Trans_J,Trans_k);
    eps=1e-5;
    %迭代计算 并计时
    tStart=tic;

[V,sita,V_vis,sita_vis,flags]=powerFlowPQCalcu(n,n_pq,n_pv,n_bal,V0,theta0,G,B,P_s,
Q_s,eps,50);

    %记录迭代中节点电压和相角变化数据
    V_exp(i,:)=V;
    theta_exp(i,:)=sita;

    %计算交流线路上的潮流 I 侧参数
    for j=1:length(AC_I)
        indexI=find(Nodes==AC_I(j)); %当前在 I 侧遍历的节点对应 Nodes 的位置
        indexJ=find(Nodes==AC_J(j)); %当前在 J 侧遍历的节点对应 Nodes 的位置
        F_AC(i,j)=((V(indexI)-V(indexJ))/AC_Z(j))-V(indexI)*AC_y(j);

```

```

        end
        tEnd=toc(tStart);
    end

    可视化
    tit="发电机 GEN2 机端电压"
    xla="V_{GEN2}"
    figure(1)
    subplot(1,2,1)
    hold on
    grid on
    %绘制交流线潮流变化趋势
    for i=1:6
        plot(temp,real(F_AC(1:m,i)),'Marker','o');
    end
    title(strcat("交流线有功潮流随",tit,"变化趋势"))
    xlabel(xla)
    ylabel("P")
    legend("AC1","AC2","AC3","AC4","AC5","AC6")
    subplot(1,2,2)
    %绘制交流线无功潮流变化趋势
    hold on
    grid on
    for i=1:6
        plot(temp,imag(F_AC(1:m,i)),'Marker','o');
    end
    title(strcat("交流线无功潮流随",tit,"变化趋势"))
    xlabel(xla)
    ylabel("Q")
    legend("AC1","AC2","AC3","AC4","AC5","AC6")
    hold off

    %绘制节点电压幅值变化趋势
    figure(3)
    hold on
    for i=1:length(V_exp(1,:))
        if i==4
            plot(temp,V_exp(:,i),'LineWidth',1.5);
        else
            plot(temp,V_exp(:,i));
        end
    end
    end
    legend("GEN1-230","GEN2-230","GEN3-230","STNA-230","STNB-230","STNC-

```

```

230","GEN2","GEN3","GEN1");
grid on
title(strcat("各节点电压幅值随",tit,"变化趋势"))
xlabel(xla)
ylabel("V")

%绘制节点电压相角变化趋势
figure(4)
hold on
for i=1:length(theta_exp(1,:))
    plot(temp,theta_exp(:,i));
end
grid on
legend("GEN1-230","GEN2-230","GEN3-230","STNA-230","STNB-230","STNC-230","GEN2","GEN3","GEN1");
title(strcat("各节点电压相角随",tit,"变化趋势"))
xlabel(xla)
ylabel("\theta")

```


附录 3 PowerFlower 前端设计程序

附录 3: PowerFlower.mlapp

说明：用 MATLAB App Designer 开发的前端程序

```
classdef PowerFlower < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                matlab.ui.Figure
        Panel                    matlab.ui.container.Panel
        TabGroup                  matlab.ui.container.TabGroup
        Tab_Bus                   matlab.ui.container.Tab
        BusTable                   matlab.ui.control.Table
        AddButton_Bus              matlab.ui.control.Button
        EditFieldLabel              matlab.ui.control.Label
        nameEditField_Bus           matlab.ui.control.EditField
        DropDownLabel               matlab.ui.control.Label
        typeDropDown_Bus            matlab.ui.control.DropDown
        EditField_2Label            matlab.ui.control.Label
        RVEditField_Bus             matlab.ui.control.NumericEditField
        Tab_AC                      matlab.ui.container.Tab
        ACTable                      matlab.ui.control.Table
        AddButton_AC                 matlab.ui.control.Button
        Label                       matlab.ui.control.Label
        NameEditField_AC             matlab.ui.control.EditField
        JLabel                      matlab.ui.control.Label
        BusofJDropDown_AC            matlab.ui.control.DropDown
        ILabel                      matlab.ui.control.Label
        BusofIDropDown_AC            matlab.ui.control.DropDown
        R1EditFieldLabel              matlab.ui.control.Label
        R1EditField_AC               matlab.ui.control.NumericEditField
        X1EditFieldLabel              matlab.ui.control.Label
        X1EditField_AC               matlab.ui.control.NumericEditField
        yLabel                      matlab.ui.control.Label
        yEditField_AC                matlab.ui.control.NumericEditField
        Tab_TR                       matlab.ui.container.Tab
        TransTable                   matlab.ui.control.Table
        ILabel_2                    matlab.ui.control.Label
        BusofIDropDown_TR            matlab.ui.control.DropDown
        JLabel_2                    matlab.ui.control.Label
        BusofJDropDown_TR            matlab.ui.control.DropDown
        kEditFieldLabel              matlab.ui.control.Label
        kEditField                   matlab.ui.control.NumericEditField
        R1EditFieldLabel_2            matlab.ui.control.Label
    end
end
```

R1EditField_TR	matlab.ui.control.NumericEditField
X1EditFieldLabel_2	matlab.ui.control.Label
X1EditField_TR	matlab.ui.control.NumericEditField
AddButton_TR	matlab.ui.control.Button
Label_16	matlab.ui.control.Label
NameEditField_TR	matlab.ui.control.EditField
Tab_GE	matlab.ui.container.Tab
GenerTable	matlab.ui.control.Table
Label_3	matlab.ui.control.Label
NameEditField_GE	matlab.ui.control.EditField
Label_4	matlab.ui.control.Label
BusDropDown_GE	matlab.ui.control.DropDown
Label_5	matlab.ui.control.Label
SNEditField	matlab.ui.control.NumericEditField
Label_6	matlab.ui.control.Label
VEditField_GE	matlab.ui.control.NumericEditField
Label_7	matlab.ui.control.Label
PditField_GE	matlab.ui.control.NumericEditField
AddButton_GE	matlab.ui.control.Button
EditField_4Label	matlab.ui.control.Label
thetaEditField_GE	matlab.ui.control.NumericEditField
Tab_LD	matlab.ui.container.Tab
LDTable	matlab.ui.control.Table
Label_9	matlab.ui.control.Label
NameEditField_LD	matlab.ui.control.EditField
Label_10	matlab.ui.control.Label
BusDropDown_LD	matlab.ui.control.DropDown
Label_13	matlab.ui.control.Label
PditField_LD	matlab.ui.control.NumericEditField
Label_14	matlab.ui.control.Label
QEditField_LD	matlab.ui.control.NumericEditField
AddButton_LD	matlab.ui.control.Button
TransNodes	matlab.ui.container.Tab
TransNodesTable	matlab.ui.control.Table
Label_17	matlab.ui.control.Label
BusDropDown_TRN	matlab.ui.control.DropDown
AddButton_TRN	matlab.ui.control.Button
Panel_2	matlab.ui.container.Panel
TabGroup2	matlab.ui.container.TabGroup
Tab	matlab.ui.container.Tab
EditField_5Label	matlab.ui.control.Label
EditField_5	matlab.ui.control.NumericEditField
EditField_6Label	matlab.ui.control.Label
EditField_6	matlab.ui.control.NumericEditField

```

DropDown_2Label      matlab.ui.control.Label
DropDown_Method      matlab.ui.control.DropDown
EditField_7Label     matlab.ui.control.Label
EditField_eps        matlab.ui.control.NumericEditField
EditField_8Label     matlab.ui.control.Label
EditField_maxCount   matlab.ui.control.NumericEditField
Button               matlab.ui.control.Button
Label_18              matlab.ui.control.Label
Label_19              matlab.ui.control.Label
contiWork            matlab.ui.container.Tab
Label_20              matlab.ui.control.Label
EditField_calcuTimes matlab.ui.control.NumericEditField
Label_21              matlab.ui.control.Label
EditField_steps      matlab.ui.control.NumericEditField
contiWorkButton      matlab.ui.control.Button
Label_22              matlab.ui.control.Label
ListBox_OBJ          matlab.ui.control.ListBox
UIAxes               matlab.ui.control.UIAxes
Panel_3              matlab.ui.container.Panel
TabGroup3            matlab.ui.container.TabGroup
Tab_3                matlab.ui.container.Tab
UITable_Result       matlab.ui.control.Table
Button_2              matlab.ui.control.Button
Tab_4                matlab.ui.container.Tab
end

```

```

properties (Access = public)
    %节点信息
    nodeName=["GEN1","GEN2","GEN3","GEN1-230","GEN2-230","GEN3-230",
"STNA-230","STNB-230","STNC-230"];
    nodeRV=[16.5,18,13.8,230,230,230,230,230,230];
    nodeType=["Slack","PV","PV","PQ","PQ","PQ","PQ","PQ","PQ"];
    %交流线信息
    ACName;
    AC_I=["GEN1-230","STNA-230","GEN2-230","STNC-230","GEN3-230",
"STNB-230","GEN1","GEN2","GEN3"]
    AC_J=["STNA-230","GEN2-230","STNC-230","GEN3-230","STNB-230",
"GEN1-230","GEN1-230","GEN2-230","GEN3-230"]; %J 侧节点
    AC_Z=[0.01+0.085j,0.032+0.161j,0.0085+0.072j,0.0119+0.1008j,0.039+0.17j,0.017+0.092j,0.0576j,0.0625j,0.05860j];%I-J 阻抗
    AC_y=[0.088j,0.153j,0.0745j,0.1045j,0.179j,0.079j,0,0,0];%I-J 对地导纳参数
    %变压器信息

```

```

trName;
TR_I=["GEN1","GEN2","GEN3"];
TR_J=["GEN1-230","GEN2-230","GEN3-230"];
TR_k=[1,1,1];
TR_Z;
%发电机信息
geName;
Gen=["GEN1","GEN2","GEN3"]; %对应母线
S_N; %额定容量
GE_P=[0,1.63,0.85]; %有功发电
GE_V=[1.04,1.025,1.025]; %母线电压幅值
GE_theta=[0]; %母线电压相角
%负荷信息
ldName;
Load=["STNA-230","STNB-230","STNC-230"]; %负荷对应母线
LD_P=[-125,-0.9,-1.00]; %有功负荷
LD_Q=[-0.5,-0.3,-0.35]; %无功负荷
%转移节点
transNodes=["GEN1-230","GEN2-230","GEN3-230"];
%系统状态信息
V0;
theta0;
P_s;
Q_s;
Nodes;
Y; %节点导纳矩阵
G;
B;
n;
n_pq;
n_pv;
n_slack;
%迭代信息

%潮流计算结果
V;
theta;
V_vis;
theta_vis;

maxCount;
eps;
end

```

```

% Callbacks that handle component events
methods (Access = private)

% Button pushed function: AddButton_Bus
function AddButton_BusPushed(app, event)
    app.AddButton_Bus.Enable="off";

    %输入数据
    nodeName=string(app.nameEditField_Bus.Value);
    nodeRV=app.RVEditField_Bus.Value;
    nodeType=string(app.typeDropDown_Bus.Value);

    %赋值到全局变量
    app.nodeName=[app.nodeName,nodeName];
    app.nodeRV=[app.nodeRV,nodeRV];
    app.nodeType=[app.nodeType,nodeType];

    %在表格中显示
    newdata=[nodeName,nodeRV,nodeType];
    newdata=array2table(newdata);
    app.BusTable.Data=[app.BusTable.Data;newdata];

    %清空数据
    app.nameEditField_Bus.Value=" ";
    app.RVEditField_Bus.Value=0;
    app.typeDropDown_Bus.Value="PQ";

    %更新交流线 I-J 两侧节点选项
    app.BusofJDropDown_AC.Items=app.nodeName;
    app.BusofIDDropDown_AC.Items=app.nodeName;
    %更新变压器 I-J 两侧节点选项
    app.BusofIDDropDown_TR.Items=app.nodeName;
    app.BusofJDropDown_TR.Items=app.nodeName;
    %更新发电机连接的母线选项
    app.BusDropDown_GE.Items=app.nodeName;
    %更新负荷连接的母线选项
    app.BusDropDown_LD.Items=app.nodeName;
    %更新转移节点选项
    app.BusDropDown_TRN.Items=app.nodeName;

    app.AddButton_Bus.Enable="on";
end

```

```

% Size changed function: Tab_AC
function Tab_ACSizedChanged(app, event)
    position = app.Tab_AC.Position;

end

% Button pushed function: AddButton_AC
function AddButton_ACPushed(app, event)
    app.AddButton_AC.Enable="off";

    %输入数据
    ACName=string(app.NameEditField_AC.Value);
    AC_I=string(app.BusofIDropDown_AC.Value);
    AC_J=string(app.BusofJDropDown_AC.Value);
    R1=app.R1EditField_AC.Value;
    X1=app.X1EditField_AC.Value;
    AC_Z=R1+j*X1;
    AC_y=app.yEditField_AC.Value*j;

    %数据转换并在表格中显示
    newdata=[ACName,AC_I,AC_J,R1,X1,AC_y];
    newdata=array2table(newdata);
    app.ACTable.Data=[app.ACTable.Data;newdata];

    %赋值到全局变量
    app.ACName=[app.ACName,ACName];
    app.AC_I=[app.AC_I,AC_I];
    app.AC_J=[app.AC_J,AC_J];
    app.AC_Z=[app.AC_Z,AC_Z];
    app.AC_y=[app.AC_y,AC_y];

    %删除空数组
    if ismember("",app.AC_I)
        app.AC_I=app.AC_I(2:end);
        app.AC_J=app.AC_J(2:end);
    end

    %清空数据
    app.NameEditField_AC.Value=" ";
    app.R1EditField_AC.Value=0;
    app.X1EditField_AC.Value=0;
    app.yEditField_AC.Value=0;

    %绘制单线图

```

```

        G=graph(app.AC_I,app.AC_J);
        plot(app.UIAxes,G);

%          %发到工作区看看
%          assignin('base','AC_I',app.AC_I);
%          assignin('base','AC_J',app.AC_J);
%          assignin('base','AC_Z',app.AC_Z);
%          assignin('base','AC_y',app.AC_y);
        app.AddButton_AC.Enable="on";
    end

% Button pushed function: AddButton_TR
function AddButton_TRPushed(app, event)
    app.AddButton_TR.Enable="off";

    %输入数据
    trName=string(app.NameEditField_TR.Value);
    TR_I=string(app.BusofIDropDown_TR.Value);
    TR_J=string(app.BusofJDropDown_TR.Value);
    TR_k=app.kEditField.Value;
    R1=app.R1EditField_TR.Value;
    X1=app.X1EditField_TR.Value;

    %数据转换并在表格中显示
    newdata=[trName,TR_I,TR_J,TR_k,R1,X1];
    newdata=array2table(newdata);
    app.TransTable.Data=[app.TransTable.Data;newdata];

    %赋值到全局变量
    app.trName=[app.trName,trName];
    app.TR_I=[app.TR_I,TR_I];
    app.TR_J=[app.TR_J,TR_J];
    app.TR_k=[app.TR_k,TR_k];
    app.TR_Z=[app.TR_Z,R1+j*X1];

    %清空数据
    app.NameEditField_TR.Value=" ";
    app.R1EditField_TR.Value=0;
    app.X1EditField_TR.Value=0;
    app.kEditField.Value=0;

    app.AddButton_TR.Enable="on";

```

```

        %更新连续潮流作业调节元件选项
        app.ListBox_OBJ.Items=[ app.ListBox_OBJ.Items,"trName"];
    end

    % Button pushed function: AddButton_GE
    function AddButton_GEPushed(app, event)
        app.AddButton_GE.Enable="off";

        %输入数据
        geName=string(app.NameEditField_GE.Value);
        Gen=string(app.BusDropDown_GE.Value);
        S_N=app.SNEditField.Value;
        GE_P=app.PditField_GE.Value;
        GE_V=app.VEditField_GE.Value;
        GE_theta=app.thetaEditField_GE.Value;

        %数据转换并在表格中显示
        newdata=[geName,Gen,S_N,GE_V,GE_theta,GE_P];
        newdata=array2table(newdata);
        app.GenerTable.Data=[app.GenerTable.Data;newdata];

        %赋值到全局变量
        app.geName=[app.geName,geName];
        app.Gen=[app.Gen,Gen];
        app.S_N=[app.S_N,S_N];
        app.GE_P=[app.GE_P,GE_P];
        app.GE_V=[app.GE_V,GE_V];
        app.GE_theta=GE_theta;

        %清空数据
        app.NameEditField_GE.Value=" ";
        app.SNEditField.Value=0;
        app.PditField_GE.Value=0;
        app.VEditField_GE.Value=0;
        app.thetaEditField_GE.Value=0;

        app.AddButton_GE.Enable="on";

    end

    % Button pushed function: AddButton_LD
    function AddButton_LDPushed(app, event)
        app.AddButton_LD.Enable="off";

```



```

%输入数据
ldName=string(app.NameEditField_LD.Value);
Load=string(app.BusDropDown_LD.Value);
LD_P=app.PditField_LD.Value;
LD_Q=app.QEditField_LD.Value;

%数据转换并在表格中显示
newdata=[ldName,Load,LD_P,LD_Q];
newdata=array2table(newdata);
app.LDTable.Data=[app.LDTable.Data;newdata];

%赋值到全局变量
app.ldName=[app.ldName,ldName];
app.Load=[app.Load,Load];
app.LD_P=[app.LD_P,LD_P];

%清空数据
app.NameEditField_LD.Value=" ";
app.PditField_LD.Value=0;
app.QEditField_LD.Value=0;

app.AddButton_LD.Enable="on";
end

% Button pushed function: AddButton_TRN
function AddButton_TRNPushed(app, event)
    app.AddButton_TRN.Enable="off";
    transNodes=string(app.BusDropDown_TRN.Value);

    %赋值到全局变量
    app.transNodes=[app.transNodes,transNodes];

    %数据转换并在表格中显示
    newdata=[transNodes,"0","0"];
    newdata=array2table(newdata);
    app.TransNodesTable.Data=[app.TransNodesTable.Data;newdata];

    app.AddButton_TRN.Enable="on";
end

% Button pushed function: Button
function ButtonPushed(app, event)
%数据预处理

```

```

[app.Nodes,app.P_s,app.Q_s,app.V0,app.theta0,app.n,app.n_pq,app.n_pv,app.n_slack]=
DataProcess(app.nodeName,app.nodeType,app.Load,app.LD_P,app.LD_Q,app.transNodes,app.Gen,app.GE_P,app.GE_V,app.GE_theta);
    %发到工作区看看
    assignin('base','Nodes1',app.Nodes);
    assignin('base','P_s1',app.P_s);
    assignin('base','Q_s1',app.Q_s);
    assignin('base','V01',app.V0);
    assignin('base','theta01',app.theta0);
    assignin('base','n1',app.n);
    assignin('base','n_pq',app.n_pq);
    assignin('base','n_pv',app.n_pv);
    assignin('base','n_slack',app.n_slack);

    %潮流作业设置信息读取
    app.eps=app.EditField_eps.Value;
    app.maxCount=app.EditField_maxCount.Value;
    %生成节点导纳矩阵

[app.G,app.B]=GenerateY(app.Nodes,app.AC_I,app.AC_J,app.AC_Z,app.AC_y,app.TR_I,app.TR_J,app.TR_k);
    app.Y=app.G+j*app.B;
    Method=string(app.DropDown_Method.Value);

    %PQ 法潮流计算
    if strcmp(Method,"PQ_DeCoupled")==1
        tStart=tic;

[app.V,app.theta,app.V_vis,app.theta_vis,Flag_success]=powerFlowPQCalcu(app.n,app.n_pq,app.n_pv,app.n_slack,app.V0,app.theta0,app.G,app.B,app.P_s,app.Q_s,app.eps,app.maxCount);
        tEnd=toc(tStart);
        if Flag_success==1
            strFinish=strcat("潮流计算完成，用时",string(tEnd),"s");
            msgbox(strFinish,"提示");
        else
            errorlg("潮流计算结果不收敛","警告");
        end
    elseif strcmp(Method,"Newton")==1
        tStart=tic;

[app.V,app.theta,app.V_vis,app.theta_vis,Flag_success]=powerFlowNewtonCalcu(app.n,app.n_pq,app.n_pv,app.n_slack,app.V0,app.theta0,app.G,app.B,app.P_s,app.Q_s,app.eps,app.maxCount);

```

```

        tEnd=toc(tStart);
        if Flag_success==1
            strFinish=strcat("潮流计算完成，用时",string(tEnd),"s");
            msgbox(strFinish,"提示");
        else
            errorlg("潮流计算结果不收敛","警告");
        end
    end
    assignin('base','V_app',app.V);
    assignin('base','theta_app',app.theta);

    %绘制系统单线图
    app.G=graph(app.AC_I,app.AC_J);
    H2=plot(app.UIAxes,app.G);
    highlight(H2,[7 8 9],'NodeColor','r');
    for i=1:length(app.Nodes)
        labelnode(H2,app.Nodes(i),strcat(num2str(app.V(i)),"
",num2str(app.theta(i))))
    end
end

% Button pushed function: Button_2
function Button_2Pushed(app, event)
    %潮流结果呈现
    app.UITable_Result.RowName=app.Nodes';
    app.UITable_Result.Data(:,1)=string(app.V');
    app.UITable_Result.Data(:,2)=string(app.theta');
    app.UITable_Result.Data(:,3)=string(app.nodeRV');
end

% Button pushed function: contiWorkButton
function contiWorkButtonPushed(app, event)

end

end

% Component initialization
methods (Access = private)

    % Create UIFigure and components
    function createComponents(app)

        % Create UIFigure and hide until all components are created
        app.UIFigure = uifigure('Visible', 'off');

```

```

app.UIFigure.Position = [100 100 1128 656];
app.UIFigure.Name = 'UI Figure';

% Create Panel
app.Panel = uipanel(app.UIFigure);
app.Panel.TitlePosition = 'centertop';
app.Panel.Title = '系统数据信息';
app.Panel.FontSize = 18;
app.Panel.Position = [1 1 461 656];

% Create TabGroup
app.TabGroup = uitabgroup(app.Panel);
app.TabGroup.Position = [0 0 460 622];

% Create Tab_Bus
app.Tab_Bus = uitab(app.TabGroup);
app.Tab_Bus.Title = '母线';

% Create BusTable
app.BusTable = uitable(app.Tab_Bus);
app.BusTable.ColumnName = {'名称'; '基准电压'; '节点类型'};
app.BusTable.RowName = {};
app.BusTable.FontSize = 15;
app.BusTable.Position = [23 191 416 399];

% Create AddButton_Bus
app.AddButton_Bus = uibutton(app.Tab_Bus, 'push');
app.AddButton_Bus.ButtonPushedFcn = createCallbackFcn(app,
@AddButton_BusPushed, true);
app.AddButton_Bus.Position = [176 22 107 30];
app.AddButton_Bus.Text = '确定';

% Create EditFieldLabel
app.EditFieldLabel = uilabel(app.Tab_Bus);
app.EditFieldLabel.HorizontalAlignment = 'right';
app.EditFieldLabel.Position = [24 148 29 22];
app.EditFieldLabel.Text = '名称';

% Create nameEditField_Bus
app.nameEditField_Bus = uieditfield(app.Tab_Bus, 'text');
app.nameEditField_Bus.Position = [109 148 100 22];

% Create DropDownLabel
app.DropDownLabel = uilabel(app.Tab_Bus);

```

```

app.DropDownLabel.HorizontalAlignment = 'right';
app.DropDownLabel.Position = [256 148 53 22];
app.DropDownLabel.Text = '节点类型';

% Create typeDropDown_Bus
app.typeDropDown_Bus = uidropdown(app.Tab_Bus);
app.typeDropDown_Bus.Items = {'PQ', 'PV', 'Slack'};
app.typeDropDown_Bus.Position = [324 148 100 22];
app.typeDropDown_Bus.Value = 'PQ';

% Create EditField_2Label
app.EditField_2Label = uilabel(app.Tab_Bus);
app.EditField_2Label.HorizontalAlignment = 'right';
app.EditField_2Label.Position = [24 105 53 22];
app.EditField_2Label.Text = '基准电压';

% Create RVEditField_Bus
app.RVEditField_Bus = uieditfield(app.Tab_Bus, 'numeric');
app.RVEditField_Bus.Position = [109 105 100 22];

% Create Tab_AC
app.Tab_AC = uitab(app.TabGroup);
app.Tab_AC.SizeChangedFcn = createCallbackFcn(app,
@Tab_ACSized, true);
app.Tab_AC.Title = '交流线';

% Create ACTable
app.ACTable = uitable(app.Tab_AC);
app.ACTable.ColumnName = {'名称'; 'I 侧母线'; 'J 侧母线'; 'R1'; 'X1';
'y0'};

app.ACTable.RowName = {};
app.ACTable.Position = [17 173 425 412];

% Create AddButton_AC
app.AddButton_AC = uibutton(app.Tab_AC, 'push');
app.AddButton_AC.ButtonPushedFcn = createCallbackFcn(app,
@AddButton_ACPushed, true);
app.AddButton_AC.Position = [173 6 114 28];
app.AddButton_AC.Text = '确定';

% Create Label
app.Label = uilabel(app.Tab_AC);
app.Label.HorizontalAlignment = 'right';
app.Label.Position = [26 129 29 22];

```

```

app.Label.Text = '名称';

% Create NameEditField_AC
app.NameEditField_AC = uieditfield(app.Tab_AC, 'text');
app.NameEditField_AC.Position = [78 129 100 22];

% Create JLabel
app.JLabel = uilabel(app.Tab_AC);
app.JLabel.HorizontalAlignment = 'right';
app.JLabel.Position = [19 55 47 22];
app.JLabel.Text = 'J 侧母线';

% Create BusofJDropDown_AC
app.BusofJDropDown_AC = uidropdown(app.Tab_AC);
app.BusofJDropDown_AC.Items = {};
app.BusofJDropDown_AC.Position = [81 55 100 22];
app.BusofJDropDown_AC.Value = {};

% Create ILabel
app.ILabel = uilabel(app.Tab_AC);
app.ILabel.HorizontalAlignment = 'right';
app.ILabel.Position = [18 92 45 22];
app.ILabel.Text = 'I 侧母线';

% Create BusofIDropDown_AC
app.BusofIDropDown_AC = uidropdown(app.Tab_AC);
app.BusofIDropDown_AC.Items = {};
app.BusofIDropDown_AC.Position = [78 92 100 22];
app.BusofIDropDown_AC.Value = {};

% Create R1EditFieldLabel
app.R1EditFieldLabel = uilabel(app.Tab_AC);
app.R1EditFieldLabel.HorizontalAlignment = 'right';
app.R1EditFieldLabel.Position = [267 129 25 22];
app.R1EditFieldLabel.Text = 'R1';

% Create R1EditField_AC
app.R1EditField_AC = uieditfield(app.Tab_AC, 'numeric');
app.R1EditField_AC.Position = [307 129 100 22];

% Create X1EditFieldLabel
app.X1EditFieldLabel = uilabel(app.Tab_AC);
app.X1EditFieldLabel.HorizontalAlignment = 'right';
app.X1EditFieldLabel.Position = [267 92 25 22];

```

```

app.X1EditFieldLabel.Text = 'X1';

% Create X1EditField_AC
app.X1EditField_AC = uieditfield(app.Tab_AC, 'numeric');
app.X1EditField_AC.Position = [307 92 100 22];

% Create yLabel
app.yLabel = uilabel(app.Tab_AC);
app.yLabel.HorizontalAlignment = 'right';
app.yLabel.Position = [267 55 25 22];
app.yLabel.Text = 'y';

% Create yEditField_AC
app.yEditField_AC = uieditfield(app.Tab_AC, 'numeric');
app.yEditField_AC.Position = [307 55 100 22];

% Create Tab_TR
app.Tab_TR = uitab(app.TabGroup);
app.Tab_TR.Title = '变压器';

% Create TransTable
app.TransTable = uitable(app.Tab_TR);
app.TransTable.ColumnName = {'名称'; 'I 侧母线'; 'J 侧母线'; '变比 k';
'R1'; 'X1'};
app.TransTable.RowName = {};
app.TransTable.Position = [13 226 437 352];

% Create ILabel_2
app.ILabel_2 = uilabel(app.Tab_TR);
app.ILabel_2.HorizontalAlignment = 'right';
app.ILabel_2.Position = [14 132 45 22];
app.ILabel_2.Text = 'I 侧母线';

% Create BusofIDropDown_TR
app.BusofIDropDown_TR = uidropdown(app.Tab_TR);
app.BusofIDropDown_TR.Items = {};
app.BusofIDropDown_TR.Position = [74 132 100 22];
app.BusofIDropDown_TR.Value = {};

% Create JLabel_2
app.JLabel_2 = uilabel(app.Tab_TR);
app.JLabel_2.HorizontalAlignment = 'right';
app.JLabel_2.Position = [12 95 47 22];
app.JLabel_2.Text = 'J 侧母线';

```

```

% Create BusofJDropDown_TR
app.BusofJDropDown_TR = uidropdown(app.Tab_TR);
app.BusofJDropDown_TR.Items = {};
app.BusofJDropDown_TR.Position = [74 95 100 22];
app.BusofJDropDown_TR.Value = {};

% Create kEditFieldLabel
app.kEditFieldLabel = uilabel(app.Tab_TR);
app.kEditFieldLabel.HorizontalAlignment = 'right';
app.kEditFieldLabel.Position = [273 173 25 22];
app.kEditFieldLabel.Text = 'k';

% Create kEditField
app.kEditField = uieditfield(app.Tab_TR, 'numeric');
app.kEditField.Position = [313 173 100 22];

% Create R1EditFieldLabel_2
app.R1EditFieldLabel_2 = uilabel(app.Tab_TR);
app.R1EditFieldLabel_2.HorizontalAlignment = 'right';
app.R1EditFieldLabel_2.Position = [273 132 25 22];
app.R1EditFieldLabel_2.Text = 'R1';

% Create R1EditField_TR
app.R1EditField_TR = uieditfield(app.Tab_TR, 'numeric');
app.R1EditField_TR.Position = [313 132 100 22];

% Create X1EditFieldLabel_2
app.X1EditFieldLabel_2 = uilabel(app.Tab_TR);
app.X1EditFieldLabel_2.HorizontalAlignment = 'right';
app.X1EditFieldLabel_2.Position = [273 95 25 22];
app.X1EditFieldLabel_2.Text = 'X1';

% Create X1EditField_TR
app.X1EditField_TR = uieditfield(app.Tab_TR, 'numeric');
app.X1EditField_TR.Position = [313 95 100 22];

% Create AddButton_TR
app.AddButton_TR = uibutton(app.Tab_TR, 'push');
app.AddButton_TR.ButtonPushedFcn = createCallbackFcn(app,
@AddButton_TRPushed, true);
app.AddButton_TR.Position = [173 28 110 34];
app.AddButton_TR.Text = '确定';

```



```

% Create Label_16
app.Label_16 = uilabel(app.Tab_TR);
app.Label_16.HorizontalAlignment = 'right';
app.Label_16.Position = [18 173 29 22];
app.Label_16.Text = '名称';

% Create NameEditField_TR
app.NameEditField_TR = uieditfield(app.Tab_TR, 'text');
app.NameEditField_TR.Position = [70 173 100 22];

% Create Tab_GE
app.Tab_GE = uitab(app.TabGroup);
app.Tab_GE.Title = '发电机';

% Create GenerTable
app.GenerTable = uitable(app.Tab_GE);
app.GenerTable.ColumnName = {'名称'; '母线名称'; '额定容量'; '母线电
压幅值'; '母线电压相角'; '有功发电'};
app.GenerTable.RowName = {};
app.GenerTable.Position = [7 231 446 355];

% Create Label_3
app.Label_3 = uilabel(app.Tab_GE);
app.Label_3.HorizontalAlignment = 'right';
app.Label_3.Position = [13 173 29 22];
app.Label_3.Text = '名称';

% Create NameEditField_GE
app.NameEditField_GE = uieditfield(app.Tab_GE, 'text');
app.NameEditField_GE.Position = [111 173 100 22];

% Create Label_4
app.Label_4 = uilabel(app.Tab_GE);
app.Label_4.HorizontalAlignment = 'right';
app.Label_4.Position = [249 173 53 22];
app.Label_4.Text = '母线名称';

% Create BusDropDown_GE
app.BusDropDown_GE = uidropdown(app.Tab_GE);
app.BusDropDown_GE.Items = {};
app.BusDropDown_GE.Position = [317 173 100 22];
app.BusDropDown_GE.Value = {};

% Create Label_5

```

```

app.Label_5 = uilabel(app.Tab_GE);
app.Label_5.HorizontalAlignment = 'right';
app.Label_5.Position = [13 138 53 22];
app.Label_5.Text = '额定容量';

% Create SNEditField
app.SNEditField = uieditfield(app.Tab_GE, 'numeric');
app.SNEditField.Position = [111 138 100 22];

% Create Label_6
app.Label_6 = uilabel(app.Tab_GE);
app.Label_6.HorizontalAlignment = 'right';
app.Label_6.Position = [13 100 53 22];
app.Label_6.Text = '母线电压';

% Create VEditField_GE
app.VEditField_GE = uieditfield(app.Tab_GE, 'numeric');
app.VEditField_GE.Position = [111 100 100 22];

% Create Label_7
app.Label_7 = uilabel(app.Tab_GE);
app.Label_7.HorizontalAlignment = 'right';
app.Label_7.Position = [249 138 53 22];
app.Label_7.Text = '有功发电';

% Create PditField_GE
app.PditField_GE = uieditfield(app.Tab_GE, 'numeric');
app.PditField_GE.Position = [317 138 100 22];

% Create AddButton_GE
app.AddButton_GE = uibutton(app.Tab_GE, 'push');
app.AddButton_GE.ButtonPushedFcn = createCallbackFcn(app,
@AddButton_GEPushed, true);
app.AddButton_GE.Position = [177 14 107 28];
app.AddButton_GE.Text = '确定';

% Create EditField_4Label
app.EditField_4Label = uilabel(app.Tab_GE);
app.EditField_4Label.HorizontalAlignment = 'right';
app.EditField_4Label.Position = [219 100 77 22];
app.EditField_4Label.Text = '母线电压相角';

% Create thetaEditField_GE
app.thetaEditField_GE = uieditfield(app.Tab_GE, 'numeric');

```

```

app.thetaEditField_GE.Position = [317 100 100 22];

% Create Tab_LD
app.Tab_LD = uitab(app.TabGroup);
app.Tab_LD.Title = '负荷';

% Create LDTable
app.LDTable = uitable(app.Tab_LD);
app.LDTable.ColumnName = {'名称'; '母线名称'; '有功负荷'; '无功负荷'};

};

app.LDTable.RowName = {};
app.LDTable.Position = [19 246 423 343];

% Create Label_9
app.Label_9 = uilabel(app.Tab_LD);
app.Label_9.HorizontalAlignment = 'right';
app.Label_9.Position = [11 195 29 22];
app.Label_9.Text = '名称';

% Create NameEditField_LD
app.NameEditField_LD = uieditfield(app.Tab_LD, 'text');
app.NameEditField_LD.Position = [109 195 100 22];

% Create Label_10
app.Label_10 = uilabel(app.Tab_LD);
app.Label_10.HorizontalAlignment = 'right';
app.Label_10.Position = [247 195 53 22];
app.Label_10.Text = '母线名称';

% Create BusDropDown_LD
app.BusDropDown_LD = uidropdown(app.Tab_LD);
app.BusDropDown_LD.Items = {};
app.BusDropDown_LD.Position = [315 195 100 22];
app.BusDropDown_LD.Value = {};

% Create Label_13
app.Label_13 = uilabel(app.Tab_LD);
app.Label_13.HorizontalAlignment = 'right';
app.Label_13.Position = [11 149 53 22];
app.Label_13.Text = '有功负荷';

% Create PditField_LD
app.PditField_LD = uieditfield(app.Tab_LD, 'numeric');
app.PditField_LD.Position = [109 149 100 22];

```

```

% Create Label_14
app.Label_14 = uilabel(app.Tab_LD);
app.Label_14.HorizontalAlignment = 'right';
app.Label_14.Position = [247 149 53 22];
app.Label_14.Text = '无功负荷';

% Create QEditField_LD
app.QEditField_LD = uieditfield(app.Tab_LD, 'numeric');
app.QEditField_LD.Position = [315 149 100 22];

% Create AddButton_LD
app.AddButton_LD = uibutton(app.Tab_LD, 'push');
app.AddButton_LD.ButtonPushedFcn = createCallbackFcn(app,
@AddButton_LDPushed, true);
app.AddButton_LD.Position = [175 25 110 35];
app.AddButton_LD.Text = '确定';

% Create TransNodes
app.TransNodes = uitab(app.TabGroup);
app.TransNodes.Title = '转移节点';

% Create TransNodesTable
app.TransNodesTable = uitable(app.TransNodes);
app.TransNodesTable.ColumnName = {'名称'; '有功负荷'; '无功负荷'};
app.TransNodesTable.RowName = {};
app.TransNodesTable.Position = [19 246 423 343];

% Create Label_17
app.Label_17 = uilabel(app.TransNodes);
app.Label_17.HorizontalAlignment = 'right';
app.Label_17.Position = [147 185 53 22];
app.Label_17.Text = '母线名称';

% Create BusDropDown_TRN
app.BusDropDown_TRN = uidropdown(app.TransNodes);
app.BusDropDown_TRN.Items = {};
app.BusDropDown_TRN.Position = [215 185 100 22];
app.BusDropDown_TRN.Value = {};

% Create AddButton_TRN
app.AddButton_TRN = uibutton(app.TransNodes, 'push');
app.AddButton_TRN.ButtonPushedFcn = createCallbackFcn(app,
@AddButton_TRNPushed, true);

```

```

app.AddButton_TRN.Position = [176 91 110 35];
app.AddButton_TRN.Text = '确定';

% Create Panel_2
app.Panel_2 = uipanel(app.UIFigure);
app.Panel_2.TitlePosition = 'centertop';
app.Panel_2.Title = '潮流作业设置';
app.Panel_2.FontSize = 18;
app.Panel_2.Position = [464 1 306 305];

% Create TabGroup2
app.TabGroup2 = uitabgroup(app.Panel_2);
app.TabGroup2.Position = [1 0 304 272];

% Create Tab
app.Tab = uitab(app.TabGroup2);
app.Tab.Title = '常规作业';

% Create EditField_5Label
app.EditField_5Label = uilabel(app.Tab);
app.EditField_5Label.HorizontalAlignment = 'right';
app.EditField_5Label.Position = [17 210 77 22];
app.EditField_5Label.Text = '计算电压上限';

% Create EditField_5
app.EditField_5 = uieditfield(app.Tab, 'numeric');
app.EditField_5.Position = [109 210 100 22];

% Create EditField_6Label
app.EditField_6Label = uilabel(app.Tab);
app.EditField_6Label.HorizontalAlignment = 'right';
app.EditField_6Label.Position = [17 170 77 22];
app.EditField_6Label.Text = '计算电压下限';

% Create EditField_6
app.EditField_6 = uieditfield(app.Tab, 'numeric');
app.EditField_6.Position = [109 170 100 22];

% Create DropDown_2Label
app.DropDown_2Label = uilabel(app.Tab);
app.DropDown_2Label.HorizontalAlignment = 'right';
app.DropDown_2Label.Position = [17 125 53 22];
app.DropDown_2Label.Text = '计算方法';

```

```

% Create DropDown_Method
app.DropDown_Method = uidropdown(app.Tab);
app.DropDown_Method.Items = {'PQ_DeCoupled', 'Newton'};
app.DropDown_Method.Position = [109 125 138 22];
app.DropDown_Method.Value = 'PQ_DeCoupled';

% Create EditField_7Label
app.EditField_7Label = uilabel(app.Tab);
app.EditField_7Label.HorizontalAlignment = 'right';
app.EditField_7Label.Position = [17 84 77 22];
app.EditField_7Label.Text = '最大允许误差';

% Create EditField_eps
app.EditField_eps = uieditfield(app.Tab, 'numeric');
app.EditField_eps.Position = [110 84 100 22];
app.EditField_eps.Value = 0.0001;

% Create EditField_8Label
app.EditField_8Label = uilabel(app.Tab);
app.EditField_8Label.HorizontalAlignment = 'right';
app.EditField_8Label.Position = [17 51 77 22];
app.EditField_8Label.Text = '最大迭代上限';

% Create EditField_maxCount
app.EditField_maxCount = uieditfield(app.Tab, 'numeric');
app.EditField_maxCount.Position = [110 51 100 22];
app.EditField_maxCount.Value = 50;

% Create Button
app.Button = uibutton(app.Tab, 'push');
app.Button.ButtonPushedFcn = createCallbackFcn(app, @ButtonPushed,
true);

app.Button.Position = [109 10 100 24];
app.Button.Text = '确定';

% Create Label_18
app.Label_18 = uilabel(app.Tab);
app.Label_18.FontName = 'Californian FB';
app.Label_18.FontColor = [1 0 0];
app.Label_18.Position = [218 84 53 22];
app.Label_18.Text = '（必填）';

% Create Label_19
app.Label_19 = uilabel(app.Tab);

```

```

app.Label_19.FontName = 'Californian FB';
app.Label_19.FontColor = [1 0 0];
app.Label_19.Position = [218 51 53 22];
app.Label_19.Text = '（必填）';

% Create contiWork
app.contiWork = uitab(app.TabGroup2);
app.contiWork.Title = '批量作业';

% Create Label_20
app.Label_20 = uilabel(app.contiWork);
app.Label_20.HorizontalAlignment = 'right';
app.Label_20.Position = [10 204 65 22];
app.Label_20.Text = '总计算次数';

% Create EditField_calcuTimes
app.EditField_calcuTimes = uieditfield(app.contiWork, 'numeric');
app.EditField_calcuTimes.Position = [90 204 100 22];
app.EditField_calcuTimes.Value = 20;

% Create Label_21
app.Label_21 = uilabel(app.contiWork);
app.Label_21.HorizontalAlignment = 'right';
app.Label_21.Position = [46 169 29 22];
app.Label_21.Text = '步长';

% Create EditField_steps
app.EditField_steps = uieditfield(app.contiWork, 'numeric');
app.EditField_steps.Position = [90 169 100 22];
app.EditField_steps.Value = 0.025;

% Create contiWorkButton
app.contiWorkButton = uibutton(app.contiWork, 'push');
app.contiWorkButton.ButtonPushedFcn = createCallbackFcn(app,
@contiWorkButtonPushed, true);
app.contiWorkButton.Position = [96 11 107 31];
app.contiWorkButton.Text = '确定';

% Create Label_22
app.Label_22 = uilabel(app.contiWork);
app.Label_22.HorizontalAlignment = 'right';
app.Label_22.Position = [13 125 53 22];
app.Label_22.Text = '调节元件';

```

```

% Create ListBox_OBJ
app.ListBox_OBJ = uilistbox(app.contiWork);
app.ListBox_OBJ.Items = {};
app.ListBox_OBJ.Position = [81 75 109 74];
app.ListBox_OBJ.Value = {};

% Create UIAxes
app.UIAxes = uiaxes(app.UIFigure);
title(app.UIAxes, '系统单线图')
xlabel(app.UIAxes, '')
ylabel(app.UIAxes, '')
app.UIAxes.FontSize = 15;
app.UIAxes.FontWeight = 'bold';
app.UIAxes.XTick = [];
app.UIAxes.YTick = [];
app.UIAxes.TitleFontWeight = 'bold';
app.UIAxes.Position = [462 305 306 352];

% Create Panel_3
app.Panel_3 = uipanel(app.UIFigure);
app.Panel_3.TitlePosition = 'centertop';
app.Panel_3.Title = '潮流计算结果';
app.Panel_3.FontSize = 18;
app.Panel_3.Position = [768 2 361 655];

% Create TabGroup3
app.TabGroup3 = uitabgroup(app.Panel_3);
app.TabGroup3.Position = [0 1 361 621];

% Create Tab_3
app.Tab_3 = uitab(app.TabGroup3);
app.Tab_3.Title = '母线';

% Create UITable_Result
app.UITable_Result = uitable(app.Tab_3);
app.UITable_Result.ColumnName = {'母线电压幅值'; '母线电压相角'; '
母线额定电压'};
app.UITable_Result.RowName = {'12'; '12sad'; 'we'};
app.UITable_Result.Position = [2 184 359 412];

% Create Button_2
app.Button_2 = uibutton(app.Tab_3, 'push');
app.Button_2.ButtonPushedFcn = createCallbackFcn(app,
@Button_2Pushed, true);

```



```

        app.Button_2.Position = [117 67 128 27];
        app.Button_2.Text = '结果输出报表';

        % Create Tab_4
        app.Tab_4 = uitab(app.TabGroup3);
        app.Tab_4.Title = '交流线';

        % Show the figure after all components are created
        app.UIFigure.Visible = 'on';
    end
end

% App creation and deletion
methods (Access = public)

    % Construct app
    function app = PowerFlower

        % Create UIFigure and components
        createComponents(app)

        % Register the app with App Designer
        registerApp(app, app.UIFigure)

        if nargin == 0
            clear app
        end
    end

    % Code that executes before app deletion
    function delete(app)

        % Delete UIFigure when app is deleted
        delete(app.UIFigure)
    end
end
end
end

```