

基于牛顿-拉夫逊法的潮流计算

陈昊

2012104509

(20121096 班)

摘要：本文模拟实际电力系统，运用 MATLAB 软件，通过已知量有名值的输入，考虑了变压器、接地电容及电抗器的对结果的影响，求解得到节点导纳矩阵，并采用牛顿-拉夫逊法对其进行潮流计算,最终获得潮流计算结果。

关键词：MATLAB；潮流计算；牛顿-拉夫逊法；

1.原理及算法介绍

1.1 电力系统潮流计算简介

所谓电力系统的潮流，是指系统中所有运行参数的总体，包括各个母线电压的大小和相位、各个发电机和负荷的功率及电流，以及各个变压器和线路等元件所通过的功率、电流和其中的损耗。电力系统潮流分析计算是电力系统运行和规划中最基本和最经常的计算，其任务是要在已知(或给定)某些运行参数的情况下，计算出系统中全部的运行参数。一般地说，在潮流计算中，各个母线所供负荷的功率时已知的。由于电力系统潮流计算需要求解一组非线性方程，一般采用迭代法，显然通过手工计算难度较大，这就需要编程运用计算机求解电力系统潮流计算。通过计算机进行电力系统潮流计算的方法有很多，例如牛顿-拉夫逊算法、P-Q 法、直流潮流算法及极小化潮流算法等。

1.2 牛顿—拉夫逊潮流算法的原理介绍

1.2.1 牛顿-拉夫逊算法求解非线性方程

牛顿-拉夫逊(Newton-Raphson)法，简称牛顿法，是求解非线性代数方程的一种有效且收敛速度快的迭代计算方法。其求解的基本理念在已知欲求非线性方程 $f(x^*)=0$ 的精确解 X^* 的近似解 $X^{(k)}$ ，两者之间存在一个误差 $\Delta X^{(k)}$ 。将 $f(X^{(k)} + \Delta X^{(k)})$ 使用泰勒级数展开并保留一阶导数部分 $f(X^{(k)}) + f'(X^{(k)})\Delta X^{(k)} = 0$ (一阶以上部分由于很小忽略不计)，从中解出 $\Delta X^{(k)}$ 。为了达到足够高的要求，在估计值 $X^{(k)}$ 基础上加上误差得到 $X^{(k+1)} = X^{(k)} + \Delta X^{(k)}$ ，继续重复上述运算。直到迭代到 $X^{(*)} = X^{(k+1)} + \Delta X^{(k+1)}$ 解已经足够接近精确解了。

上述迭代过程可以整理成以下迭代格式：

$$\begin{cases} \Delta X^{(k)} = -[f'(X^{(k)})]^{-1} f(X^{(k)}) \\ X^{(k+1)} = X^{(k)} + \Delta X^{(k)} \end{cases} \quad (1)$$

对于 n 维非线性方程迭代过程与一维大同小异，迭代格式如下：

$$\begin{cases} \Delta X^{(k)} = -[J(X^{(k)})]^{-1} f(X^{(k)}) \\ X^{(k+1)} = X^{(k)} + \Delta X^{(k)} \end{cases} \quad (2)$$

只不过迭代格式中的 X 、 ΔX 、 J 都是数组，运算时进行的是数组运算。

1.2.2 潮流计算节点分类

1)PQ 节点。给定节点的注入有功功率 P 和注入无功功率 Q 。这类节点对应于实际系统中的纯负荷节点(如变电所母线)、有功和无功功率都是给定的发电机节点(包括节点上带有负荷)，以及联络节点(注入有功和无功功率都等于零)。这类节点占系统中的绝大多数，它们的节点电压有效值和相位未知。

2)PV 节点。给定节点的注入有功功率 P 和节点电压有效值 U ，待求量是节点的注入无功功率 Q 和电压的相位 θ 。这类节点通常为发电机节点，其有功功率给定而且具有比较大的无功容量，它们能依靠自动电压调节器的作用使母线电压保持为给定值。有时将一些装有无功补偿设备的变电站母线也处理为 PV 节点。

3)平衡节点。在潮流计算中，必须设置一个平衡节点，其电压有效值为给定值，电压相位为 $\theta=0$ ，即系统中其他各节点的电压相位都以它为参考；而其注入的有功功率和无功功率都是待求量。实际上，由于所有的 PQ 节点和 PV 节点的注入有功功率都已经给定，而网络中的总有功功率损耗是未知的，因此平衡节点的注入有功功率必须平衡全系统的有功功率和有功损耗而不能加以给定，这也是为什么称它为平衡节点的原因。在潮流计算中，原则上可以取任一个发电机节点作为平衡节点，但通常取容量较大出线较多的发电机节点，以便当有功功率损耗估计出入较大时，对它的注入有功功率产生的影响较小。

1.2.3 极坐标形式牛顿-拉夫逊算法运用于潮流计算

1.节点功率方程

电力系统潮流计算中，将全部节点分成 PQ 节点、PV 节点和平衡节点三类。设系统中有 n 个节点，其中有 m 个 PQ 节点，而除了 PQ 节点和一个平衡节点以外，其余的都是 PV 节点，显然 PV 节点的数目为 $n-m-1$ 。为了叙述方便起见，假定节点按照先 PQ 节点，再 PV 节点，最后为平衡节点的次序编号，即

节点编号	种类	个数
1,2,...,m	PQ 节点	m
m+1,m+2,...,n-1	PV 节点	n-m-1
n	平衡节点	1

并且为了简化方程式起见，假定所有节点都含有发电机和负荷。

(1).PQ 节点的功率方程式

$$\begin{cases} P_{Gi} - P_{Li} = U_i \sum_{j \in i} U_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \\ Q_{Gi} - Q_{Li} = U_i \sum_{j \in i} U_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \end{cases}; (i=1, 2, 3, \dots, n-1) \quad (3)$$

式中 P_{Gi} 、 P_{Li} 、 Q_{Gi} 和 Q_{Li} ($i=1, 2, \dots, m$) 都是给定值，而 U_i 和 θ_i ($i=1, 2, \dots, m$) 都待求。

(2).PV 节点的功率方程

$$\begin{cases} P_{Gi} - P_{Li} = U_i \sum_{j \in i} U_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \\ Q_{Gi} - Q_{Li} = U_i \sum_{j \in i} U_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \end{cases}; (i = m+1, \dots, n-1) \quad (4)$$

式中: P_{Gi} 、 P_{Li} 和 U_i ($i=m+1, \dots, n-1$) 都是给定值, 而 Q_{Gi} 、 Q_{Li} 和 θ_i ($i=m+1, \dots, n-1$) 都待求。

(3).平衡节点的功率方程式

$$\begin{cases} P_{Gi} - P_{Li} = U_i \sum_{j \in i} U_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \\ Q_{Gi} - Q_{Li} = U_i \sum_{j \in i} U_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \end{cases}; (i=n) \quad (5)$$

式中: U_n 和 θ_n 为给定值, 且一般取为 $\theta_n=0$, 即以平衡节点的电压相位作为参考, 而 P_{Gi} 、 P_{Li} 和 Q_{Gi} 、 Q_{Li} 为待求量。

这样, 综合以上三类节点的功率方程式, 可以看出, 在潮流计算中实际上需要求解的非线性方程组为

$$\begin{cases} P_{Gi} - P_{Li} = U_i \sum_{j \in i} U_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}); i = 1, 2, \dots, n-1 \\ Q_{Gi} - Q_{Li} = U_i \sum_{j \in i} U_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}); i = 1, 2, \dots, m \end{cases} \quad (6)$$

它包含 $n-1$ 个有功功率方程和 m 个无功功率方程, 总共 $n+m-1$ 个。其中 P_{Gi} 、 P_{Li} ($i=1, 2, \dots, n-1$) 以及 Q_{Gi} 、 Q_{Li} ($i=1, 2, \dots, m$) 都是给定值; 而 θ_i ($i=1, 2, \dots, n-1$) 和 U_i ($i=1, 2, \dots, m$) 为未知量, 总共也是 $n+m-1$ 个。

于是, 潮流计算问题便转化为求解非线性方程式(6), 从而求出其中的 $n+m-1$ 个未知量, 即 PQ 节点和 PV 节点的电压相位和 PQ 节点的电压有效值。它们和给定的平衡节点电压相位、PV 节点和平衡节点给定的电压有效值一起, 便是系统中全部 n 个节点的电压相位和电压有效值。最后, 用这些节点电压相位和有效值代入式(4)中的无功功率方程式便可以得出各个 PV 节点的注入无功功率, 而代入式(5)便可以得出平衡节点的注入有功功率和注入无功功率。

2.雅克比(Jacobi)矩阵矩阵

将式(6)变形为如下形式

$$\begin{cases} \Delta P_i(x) = P_{Gi} - P_{Li} - U_i \sum_{j \in i} U_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) = 0; i = 1, 2, \dots, n-1 \\ \Delta Q_i(x) = Q_{Gi} - Q_{Li} - U_i \sum_{j \in i} U_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) = 0; i = 1, 2, \dots, m \end{cases} \quad (7)$$

Jacobi 矩阵可通过如下式通过矩阵运算求得

$$\begin{bmatrix} \Delta P_1(x) \\ \Delta P_2(x) \\ \vdots \\ \Delta P_{n-1}(x) \\ \Delta Q_1(x) \\ \Delta Q_2(x) \\ \vdots \\ \Delta Q_m(x) \end{bmatrix} = - \begin{bmatrix} H_{11} & H_{12} & \cdots & H_{1,n-1} & N_{11} & N_{12} & \cdots & N_{1,m} \\ H_{21} & H_{22} & \cdots & H_{2,n-1} & N_{21} & N_{22} & \cdots & N_{2,m} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ H_{n-1,1} & H_{n-1,2} & \cdots & H_{n-1,n-1} & N_{n-1,1} & N_{n-1,2} & \cdots & N_{n-1,m} \\ M_{11} & M_{12} & \cdots & M_{1,n-1} & L_{11} & L_{12} & \cdots & L_{1,m} \\ M_{21} & M_{22} & \cdots & M_{2,n-1} & L_{21} & L_{22} & \cdots & L_{2,m} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ M_{m,1} & M_{m,2} & \cdots & M_{m,n-1} & L_{m,1} & L_{m,2} & \cdots & L_{m,m} \end{bmatrix} \begin{bmatrix} \Delta \theta_1 \\ \Delta \theta_2 \\ \vdots \\ \Delta \theta_{n-1} \\ \Delta U_1 / U_1 \\ \Delta U_2 / U_2 \\ \vdots \\ \Delta U_m / U_m \end{bmatrix} \quad (8)$$

其中由四块矩阵 H、N、M、L 拼接而成的矩阵即为 Jacobi 矩阵，可通过分别求出 H、N、M、L 得到 Jacobi 矩阵。

(1). 对于非对角元素(j≠i)有

$$\begin{cases} H_{ij} = \frac{\partial \Delta P_i}{\partial \theta_j} = -U_i U_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \\ N_{ij} = \frac{\partial \Delta P_i}{\partial U_j} U_j = -U_i U_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \\ M_{ij} = \frac{\partial \Delta Q_i}{\partial \theta_j} = U_i U_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \\ L_{ij} = \frac{\partial \Delta Q_i}{\partial U_j} U_j = -U_i U_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \end{cases} \quad (9)$$

(2). 对于对角元素(j=i)有

$$\begin{cases} H_{ii} = \frac{\partial \Delta P_i}{\partial \theta_i} = U_i \sum_{\substack{j \in i \\ j \neq i}} U_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) \\ N_{ii} = \frac{\partial \Delta P_i}{\partial U_i} U_i = -U_i \sum_{\substack{j \in i \\ j \neq i}} U_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) - 2U_i^2 G_{ii} \\ M_{ii} = \frac{\partial \Delta Q_i}{\partial \theta_i} = -U_i \sum_{\substack{j \in i \\ j \neq i}} U_j (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) \\ L_{ii} = \frac{\partial \Delta Q_i}{\partial U_i} U_i = -U_i \sum_{\substack{j \in i \\ j \neq i}} U_j (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) + 2U_i^2 B_{ii} \end{cases} \quad (10)$$

式(8)可以简写成

$$\begin{bmatrix} \Delta P^{(k)} \\ \Delta Q^{(k)} \end{bmatrix} = - \begin{bmatrix} H^{(k)} & N^{(k)} \\ M^{(k)} & L^{(k)} \end{bmatrix} \begin{bmatrix} \Delta \theta^{(k)} \\ \Delta U^{(k)} \end{bmatrix} = -J^{(k)} \begin{bmatrix} \Delta \theta^{(k)} \\ \Delta U^{(k)} \end{bmatrix} \quad (11)$$

其中

$$\begin{cases} \Delta U^{(k)} = [\Delta U_1^{(k)} / U_1^{(k)} & \Delta U_2^{(k)} / U_2^{(k)} & \cdots & \Delta U_m^{(k)} / U_m^{(k)}]^T \\ \Delta \theta^{(k)} = [\Delta \theta_1^{(k)} & \Delta \theta_2^{(k)} & \cdots & \Delta \theta_{n-1}^{(k)}]^T \end{cases} \quad (12)$$

在式(11)中，雅可比矩阵 $J^{(k)}$ 中的分块矩阵 $H^{(k)}$ 、 $N^{(k)}$ 、 $M^{(k)}$ 和 $L^{(k)}$ 的元素是在未知量的取值为 $U^{(k)}$ 和 $\theta^{(k)}$ 时，由式(9)和(10)计算而得出的。对修正方程式(11)进行求解后，可以得出修正量 $\Delta \theta^{(k)}$ 和 $\Delta U^{(k)}$ ，从而可以得出

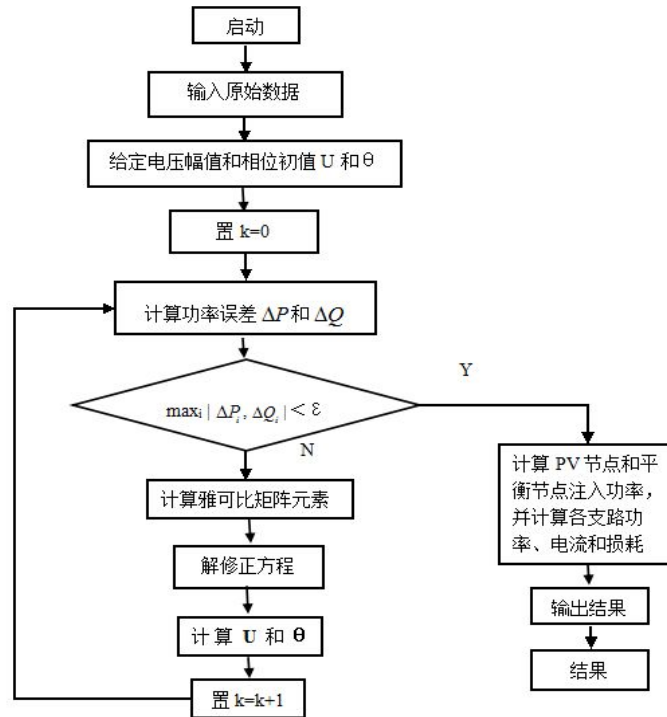
$$\begin{cases} \theta_i^{(k+1)} = \theta_i^{(k)} + \Delta \theta_i^{(k)}; i = 1, 2, \cdots, n-1 \\ U_i^{(k+1)} = U_i^{(k)} + U_i^{(k)} \times (\frac{\Delta U_i^{(k)}}{U_i^{(k)}}); i = 1, 2, \cdots, m \end{cases} \quad (13)$$

牛顿潮流计算的收敛盘踞一般取

$$\max_i |\Delta P_i^{(k)}| < \varepsilon \text{ 且 } \max_i |\Delta Q_i^{(k)}| < \varepsilon \quad (14)$$

式中： ε 为节点功率不平衡量(或称失配量)的容许误差，其取值范围一般为 $10^{-7} - 10^{-3}$ 。

2.程序框图

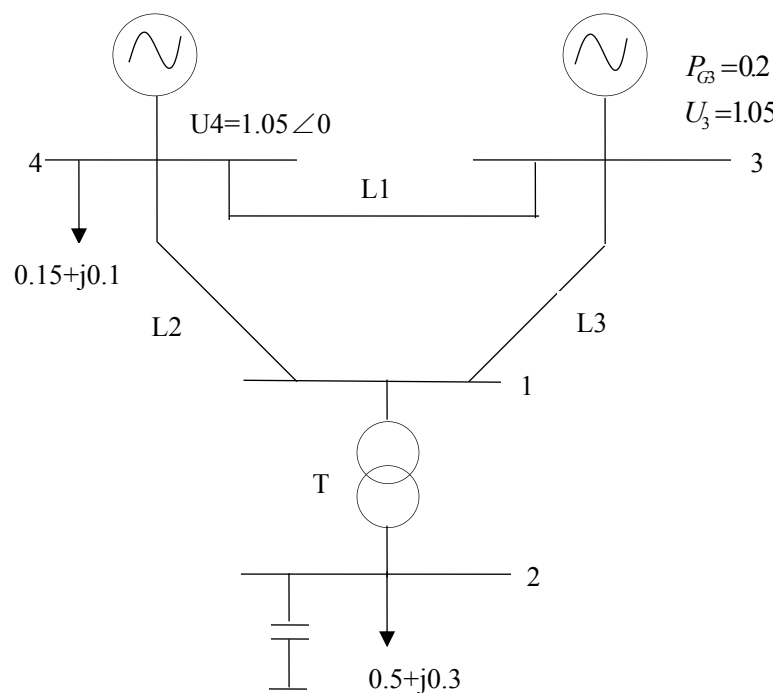


3.算例描述

夏道止——电力系统分析(第二版) 例 4-1

图示系统中，线路额定电压为 110kV，导线均采用 LGJ-120 型的，其参数为 $r_1 = 0.21\Omega/km$ ， $x_1 = 0.4\Omega/km$ ， $b_1 = 2.85 \times 10^{-6} S/km$ ，线路长度分别为 $l_1 = 150km$ ， $l_2 = 100km$ ， $l_3 = 75km$ 。变压器容量为 $63000kV \cdot A$ ，额定电压为 110/11kV，短路电压百分数 $U_k\% = 10.5$ ，在 -2.5% 分接头上运行。电容器额定容量为 $5MV \cdot A$ 。若取 $S_B = 100MV \cdot A$ ， $U_B = U_N$ ，试形成系统的节点导纳矩阵。

取节点 1 为联络节点，节点 2 为 PQ 节点，节点 3 为 PV 节点，节点 4 为平衡节点。



4.结果分析

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

潮流计算

编写者

* 陈昊 *

2014年12月

---导纳矩阵---

no. 1

no. 2

no. 3

no. 4

no. 1

no. 2

no. 3

no. 4

2.90495

0

-1.65997

-1.24498

+(-11.5031 *i)

+(5.31818 *i)

+(3.16185 *i)

+(2.37139 *i)

0

0

0

0

+(5.31818 *i)

+(-4.66384 *i)

+(0 *i)

+(0 *i)

no. 3

no. 4

-1.65997

0

2.48996

-0.829985

+(3.16185 *i)

+(0 *i)

+(-4.70398 *i)

+(1.58092 *i)

0

0

0

0

+(5.31818 *i)

+(-4.66384 *i)

+(0 *i)

+(1.58092 *i)

no. 3

no. 4

-1.65997

-1.24498

2.48996

-0.829985

+(3.16185 *i)

+(2.37139 *i)

+(-4.70398 *i)

+(1.58092 *i)

0

0

0

0

+(5.31818 *i)

+(-4.66384 *i)

+(0 *i)

+(1.58092 *i)

迭代次数: 1

最大功率误差: 0.5

迭代次数: 2

最大功率误差: 0.052346

迭代次数: 3

最大功率误差: 0.00163341

迭代次数: 4

最大功率误差: 1.62251e-06

潮流计算结果

---节点潮流---

no. i

no. 1

no. 2

no. 3

no. 4

P

0

-0.5

0.2

0.32769

Q

0

-0.3

0.19687

0.044307

U

0.9695

1.0388

1.05

1.05

phase

-0.06761

-0.1611

-0.032135

0

---支路潮流---

i

4

4

3

1

j

3

1

1

2

Pij

0.0564734

0.271213

0.254613

0.586394

Qij

-0.0570154

0.101322

0.195114

0.30125

Pji

-0.0573599

-0.259539

-0.244284

-0.706495

Qji

0.00184447

-0.109994

-0.196847

-0.21249

dP

-0.000886522

0.0116737

0.0103287

-0.120101

若想要得到其他潮流计算，只需按照实际要求修改输入数据。下面以简单的增加一个联络节点 5，并在 4,5 之间增加一条支路为例进行测试，如下所示(原始数据见附件 1)。

节点	节点	线路电阻r Ω/km	线路阻抗x Ω/km	线路电纳b S/km	线路电导g s/km	对应线路长度L km	线路电压U kv
	4	3	0.21	0.4	2.85E-06	0	150
	4	1	0.21	0.4	2.85E-06	0	100
	3	1	0.21	0.4	2.85E-06	0	75
	5	4	0.21	0.4	2.85E-06	0	75

测试数据潮流计算运行结果

1											
2											
3	潮流计算										
4	编写者										
5	* 陈 昊 *										
6	2014年12月										
7											
8	---导纳矩阵---										
9	no.1		no.2		no.3		no.4		no.5		
10	no.1	2.90495	+(-11.5031 *i)	0	+ (5.31818 *i)	-1.65997	+ (3.16185 *i)	-1.24498	+ (2.37139 *i)	0	+ (0 *i)
11	no.2	0	+ (5.31818 *i)	0	+ (-4.66384 *i)	0	+ (0 *i)	0	+ (0 *i)	0	+ (0 *i)
12	no.3	-1.65997	+ (3.16185 *i)	0	+ (0 *i)	2.48996	+ (-4.70398 *i)	-0.829985	+ (1.58092 *i)	0	+ (0 *i)
13	no.4	-1.24498	+ (2.37139 *i)	0	+ (0 *i)	-0.829985	+ (1.58092 *i)	3.73493	+ (-7.05812 *i)	-1.65997	+ (3.16185 *i)
14	no.5	0	+ (0 *i)	0	+ (0 *i)	0	+ (0 *i)	-1.65997	+ (3.16185 *i)	1.65997	+ (-3.14892 *i)
15											
16											
17											
18	迭代次数: 1	最大功率误差: 0.654338									
19	迭代次数: 2	最大功率误差: 0.148654									
20	迭代次数: 3	最大功率误差: 0.00848777									
21	迭代次数: 4	最大功率误差: 3.72855e-05									
22											
23	***潮流计算结果***										
24	---节点潮流---										
25											
26	no. i	P	Q	U	phase						
27	no. 1	0	0	0.97644	-0.16873						
28	no. 2	0	0	1.1134	-0.16873						
29	no. 3	-0.5	-0.3	0.90088	-0.21198						
30	no. 4	0.2	0.42425	1.05	-0.10397						
31	no. 5	0.37166	-0.18537	1.05	0						
32	---支路潮流---										
33	i	j	Pij	Qij	Pji	Qji	dP	dQ			
34	4	3	0.289843	0.143711	-0.27676	-0.138163	0.0130832	0.0055477			
35	4	1	0.252245	0.0870596	-0.249525	-0.0981482	0.00272024	-0.0110886			
36	3	1	-0.236193	-0.159049	0.24	0.1611	0.00380729	0.00205078			
37	5	4	0.371665	-0.185369	-0.354843	0.19481	0.0168221	0.00944096			
38	1	2	0.215229	-0.0979928	-0.248057	0.104242	-0.0328278	0.00624962			

径，学海无涯苦作舟。学习是永无止境的，我还有许多东西需要不断学习、不断总结，在学习中提高自己。

不登高山，不知天之高也；不临深谷，不知地之厚也。凡事亲自动手去实践，不轻言放弃，你收获到的也许远比你期待的要多！

参考文献：

[1] 夏道止.电力系统分析(第二版)[M].中国电力出版社,2010,12.
[2] 陈珩.电力系统稳态分析(第三版)[M].中国电力出版社,2007,06.
[3] 杨德平..MATLAB 基础教程[M].机械工业出版社,2013,2.

附 1：程序代码数据的输入及使用说明

- (1) 新建一个文件夹
- (2) 在新建的文件中创建一个新的 Excel 命名为”线路”(注意名字一定要叫这个如果想使用不同文件名则需要修改程序中相应的 xlsread(文件名)中的文件名。下同不再赘述)。打开 “线路.xls” 按照下列表格填入相应数据。

线路数据：

节点	节点	线路电 阻 r Ω /km	线路阻 抗 x Ω /km	线路电 纳 b S/km	线路电 导 g s/km	线路长 度 L km	线路电 压 U kV
4	3	0.21	0.4	2.85×10^{-6}	0	150	110
4	1	0.21	0.4	2.85×10^{-6}	0	100	110
3	1	0.21	0.4	2.85×10^{-6}	0	75	110

- (3) 在同一文件夹下新建 “变压器.xls” 按照下列表格填入数据

变压器数据：

高压侧节 点	低压侧节 点	变压器容 量 S MVA	高压侧额 定电压 kV	低压侧额 定电压 kV	短路电压 百分比%	分头
1	2	63	110	11	10.5	-0.025

- (4) 在同一文件夹下新建 “接地电容.xls” 按照下列表格填入数据

外接节点	外接接地电容功率 MVA
2	5

- (5) 在同一文件夹下新建 “PV 节点和平衡节点电压幅值和相位初值.xls” 按照下列表格填入数据(标么值)。（注：一定要先填入 PV 节点后填入平衡节点的顺序）。

PV 节点与平衡节点	电压幅值	相位初值
3	1.05	0
4	1.05	0

(6) 在同一文件夹下新建“注入功率.xls”按照下列表格填入数据。（注：只需填入 PQ 节点、PV 节点、平衡节点功率输入，且填入数据顺序也为 PQ 节点、PV 节点、平衡节点功率，流入节点的功率为正，流出节点的功率为负。）

节点	P	Q
2	-0.5	-0.3
3	0.2	0
4	-0.15	-0.1

(7) 将数据调入 MATLAB 中：在主界面中选择 Current Folder 单元右侧的按键选择数据所放



(8) 运行潮流程序。

在 Command Window 中按照提示输入数据



(9) 运行潮流程序

(10) 在 MATLAB 的 Current Folder 单元栏中可以找到“结果.txt”(或在开始新建的文件夹内)即为所需要的潮流计算结果

附 2：程序代码

```
%-----
%潮流计算
%编写者 陈昊
%三峡大学
```

%2014 年 12 月

```
%-----  
-----  
%%  
clc,clear  
A=xlsread('线路.xls');  
B=xlsread('变压器.xls');  
C=xlsread('接地电容.xls');  
%求节点个数  
n=max(max(A(:,1:2)));  
d=n;  
if isempty(B)~=1  
    m=max(max(B(:,1:2)));d=max(max(n,m));    %节点的个数为 d 且为矩阵维数  
end  
%求线路参数的标么值  
Ub=input('请输入电压基准值 kV=');Sb=input('请输入电流基准值 MVA=');  
[V n]=size(A);Zb=(Ub*Ub)/Sb;  
A1=A;  
for j=1:V  
    A1(j,3)=A(j,3)*A(j,7);  
    A1(j,4)=A(j,4)*A(j,7);  
    A1(j,5)=(A(j,5)*A(j,7));  
    A1(j,3)=A1(j,3)/Zb;                %线路阻抗实部标么值  
    A1(j,4)=A1(j,4)/Zb;                %线路阻抗虚部标么值  
    A1(j,5)=(i*A1(j,5)*Zb+A1(j,6))/2;    %线路电纳标么值  
    A1(j,7)=1/(A1(j,3)+i*A1(j,4));        %线路阻抗的倒数标么值  
end  
%对地电容标么值  
[V n]=size(C); C1=C;  
for j=1:V  
    C1(j,2)=i*C(j,2)/Sb;  
end  
%求变压器参数  
k=1.128205;    %有待修改  
B1=B;  
if isempty(B)~=1  
    [V n]=size(B);  
for j=1:V  
    Xt(j,3)=B(j,6)*B(j,4)*B(j,4)/100/B(j,3)/Zb;  
    B1(j,3)=k*Xt(j,3);                %变压器阻抗  
    B1(j,4)=i*(1-k)/k/Xt(j,3);        %变压器一次侧导纳  
    B1(j,5)=i*(k-1)/k/Xt(j,3);        %变压器二次侧导纳  
    B1(j,6)=1/(i*B1(j,3));            %变压器阻抗倒数
```

```

end
end
Y=zeros(d,d); %导纳矩阵初始化
for k=1:d
    for j=1:d
        Y(k,j)=0;
    end
end
%求自导纳
[V n]=size(A); %线路的维数
[T n]=size(B); %变压器的维数
[R n]=size(C); %外接接地电容维数
for m=1:d %将线路参数加入自导纳
    for n=1:V
        if A1(n,1)==m || A1(n,2)==m
            Y(m,m)=Y(m,m)+A1(n,7)+A1(n,5);
        end
    end
end
if isempty(B)~=1
    for m=1:d %将变压器参数加入自导纳
        for n=1:T
            if B1(n,1)==m %变压器高压侧
                Y(m,m)=Y(m,m)+B1(n,4)+B1(n,6);
            elseif B1(n,2)==m %变压器低压侧
                Y(m,m)=Y(m,m)+B1(n,5)+B1(n,6);
            end
        end
    end
end
for m=1:d %将接地电容参数加入自导纳
    for n=1:R
        if C1(n,1)==m
            Y(m,m)=Y(m,m)+C1(n,2);
        end
    end
end
%求互导纳
for w=1:V %将线路参数加入互导纳
    m=A1(w,1);n=A1(w,2);Y(m,n)=Y(m,n)-A1(w,7);Y(n,m)=Y(m,n);
end
if isempty(B)~=1
    for w=1:T %将变压器参数加入互导纳
        m=B1(w,1);n=B1(w,2);Y(m,n)=Y(m,n)-B1(w,6);Y(n,m)=Y(m,n);
    end
end

```

```

        end
    end

%%
%输入 PQ 节点 PV 节点平衡节点条件
D=xlsread('PV 节点和平衡节点电压幅值和相位初值.xls');
ZR=xlsread('注入功率.xls');
PQ_n=input('请输入 PQ 节点个数=');
PV_n=input('请输入 PV 节点个数='); %电压相位的 els 只能按照 PV 节点、平衡节点输入
WC=input('请输入潮流计算所需要的精度=');
[M N]=size(ZR);
PQ_n=0;PV_n=0;
for j=1:M
    if ZR(j,4)==1
        PQ_n=PQ_n+1;
    elseif ZR(j,4)==-1
        PV_n=PV_n+1;
    end
end
end
LL_n=d-PQ_n-PV_n-1; %联络节点个数%注入功率的 els 只能按照联络节点、PQ、PV、平衡
节点输入
G=real(Y);B=imag(Y);
U(1:LL_n)=1; %在矩阵 U 中存储联络节点的电压初值
U((LL_n+1):(PQ_n+LL_n))=1; %在矩阵 U 中存储 PQ 节点的电压初值
[V n]=size(D);
U(PQ_n+LL_n+1:PV_n+PQ_n+LL_n)=D(1:V-1,2); %在矩阵 U 中存储 PV 节点的电压初值
U(PV_n+PQ_n+LL_n+1)=D(V,2); %在矩阵 U 中存储平衡节点的电压初值
sita=zeros(1,d); %相位初始化
DDCS=0; %迭代次数初始化
pd=1; %最大功率误差初始化
%%
while pd>=WC;
    if k>=10||pd>=1000;
        fprintf('潮流不收敛');
        break
    end
    for m=1:d-1 %有功 P 初始化
        p(m)=0;
    end
    % p(1,d-1)=0;
    for m=1:d-1-PQ_n %无功 Q 初始化
        q(m)=0;
    end
end
%求有功 P

```

```

    for m=1:(d-1)
        for n=1:d
            if m>LL_n && m<=(LL_n+PQ_n)    %PQ 节点有功
p(m)=p(m)-U(m)*(U(n)*(G(m,n)*cos(sita(m)-sita(n))+B(m,n)*sin(sita(m)-
sita(n)))));
            elseif m>=(PQ_n+LL_n+1)        %PV 节点有功

p(m)=p(m)-D(m-PQ_n-LL_n,2)*(U(n)*(G(m,n)*cos(sita(m)-sita(n))+B(m,n)*
sin(sita(m)-sita(n)))));
            else                            %联络节点有功

p(m)=p(m)+0-U(m)*(U(n)*(G(m,n)*cos(sita(m)-sita(n))+B(m,n)*sin(sita(m)
-sita(n)))));
            end
        end
    end
    for m=1:(d-1)
        if m>LL_n && m<=(LL_n+PQ_n)    %PQ 节点有功
            p(m)=p(m)+ZR(m-LL_n,2);
        elseif m>=(PQ_n+LL_n+1)        %PV 节点有功
            p(m)=p(m)+ZR(m-LL_n,2);
        end
    end
    end
    %求无功 Q
    for m=1:(d-1-PQ_n)
        for n=1:d
            if m>LL_n                %PQ 节点无功

q(m)=q(m)-U(m)*(U(n)*(G(m,n)*sin(sita(m)-sita(n))-B(m,n)*cos(sita(m)-
sita(n)))));
            else                    %联络节点无功

q(m)=q(m)+0-U(m)*(U(n)*(G(m,n)*sin(sita(m)-sita(n))-B(m,n)*cos(sita(m)
-sita(n)))));
            end
        end
    end
    end
    for m=1:(d-1-PQ_n)
        if m>LL_n                %PQ 节点无功
            q(m)=q(m)+ZR(m-LL_n,3);
        end
    end
    end
    F=[p q]';
    pd=max(max(abs(p)),max(abs(q)));    %最大功率误差

```

```

H=zeros(d-1,d-1);N=zeros(d-1,PQ_n+LL_n);
M=zeros(PQ_n+LL_n,d-1);L=zeros(PQ_n+LL_n,PQ_n+LL_n);J=[H N;M L]; %清空
矩阵防止迭代时加入上一次迭代的值
%-----
-----

%求雅克比矩阵
for k=1:(d-1) %H 非对角线元素
    for j=1:(d-1)
        if k~=j
H(k,j)=-U(k)*U(j)*(G(k,j)*sin(sita(k)-sita(j))-B(k,j)*cos(sita(k)-sita(j)));
        end
    end
end

for k=1:(d-1) %N 非对角线元素
    for j=1:(PQ_n+LL_n)
        if k~=j
N(k,j)=-U(k)*U(j)*(G(k,j)*cos(sita(k)-sita(j))+B(k,j)*sin(sita(k)-sita(j)));
        end
    end
end

for k=1:(PQ_n+LL_n) %M 非对角线元素
    for j=1:(d-1)
        if k~=j
M(k,j)=U(k)*U(j)*(G(k,j)*cos(sita(k)-sita(j))+B(k,j)*sin(sita(k)-sita(j)));
        end
    end
end

for k=1:(PQ_n+LL_n) %L 非对角线元素
    for j=1:(PQ_n+LL_n)
        if k~=j
L(k,j)=-U(k)*U(j)*(G(k,j)*sin(sita(k)-sita(j))-B(k,j)*cos(sita(k)-sita(j)));
        end
    end
end

for k=1:(d-1) %H 对角线元素
    for j=1:d
H(k,k)=H(k,k)+U(k)*U(j)*(G(k,j)*sin(sita(k)-sita(j))-B(k,j)*cos(sita(k)-sita(j)));
    end
end
end

```

```

for k=1:(d-1)
    for j=1:d
        if k==j
H(k,k)=H(k,k)-U(k)*U(j)*(G(k,j)*sin(sita(k)-sita(j))-B(k,j)*cos(sita(
k)-sita(j)));
        end
    end
end
if (d-1)>=(PQ_n+LL_n)
    m=PQ_n+LL_n;
else
    m=d-1;
end
for k=1:m      %N 对角线元素
    for j=1:d
N(k,k)=N(k,k)-U(k)*U(j)*(G(k,j)*cos(sita(k)-sita(j))+B(k,j)*sin(sita(
k)-sita(j)));
    end
end
for k=1:m
    for j=1:d
        if k==j
N(k,k)=N(k,k)+U(k)*U(j)*(G(k,j)*cos(sita(k)-sita(j))+B(k,j)*sin(sita(
k)-sita(j)))-2*U(k)*U(k)*G(k,k);
        end
    end
end
for k=1:m      %M 非对角线元素
    for j=1:d
M(k,k)=M(k,k)-U(k)*U(j)*(G(k,j)*cos(sita(k)-sita(j))+B(k,j)*sin(sita(
k)-sita(j)));
    end
end
for k=1:m
    for j=1:d
        if k==j
M(k,k)=M(k,k)+U(k)*U(j)*(G(k,j)*cos(sita(k)-sita(j))+B(k,j)*sin(sita(
k)-sita(j)));
        end
    end
end
for k=1:(PQ_n+LL_n)      %L 非对角线元素
    for j=1:d
L(k,k)=L(k,k)-U(k)*U(j)*(G(k,j)*sin(sita(k)-sita(j))-B(k,j)*cos(sita(

```



```

k)-sita(j)));
    end
end
for k=1:(PQ_n+LL_n)
    for j=1:d
        if k==j
L(k,k)=L(k,k)+U(k)*U(j)*(G(k,j)*sin(sita(k)-sita(j))-B(k,j)*cos(sita(
k)-sita(j)))+2*U(k)*U(k)*B(k,k);
        end
    end
end
DDCS=DDCS+1;          %迭代次数
DDCS_GL(DDCS)=pd;      %将本次迭代的最大功率误差存下作为输出
J=[H,N;M,L];D_X=-inv(J)*F;
sita(1:d-1)=sita(1:d-1)+D_X(1:d-1)';
U(1:PQ_n+LL_n)=U(1:PQ_n+LL_n)+D_X(d:d+PQ_n+LL_n-1)'.*U(1:PQ_n+LL_n);
end
PV_Q=zeros(1,d);PH_P=zeros(1);PH_Q=zeros(1);
for k=PQ_n+1:d %PV 节点无功
    for j=1:d
PV_Q(k)=PV_Q(k)+U(k)*(U(j)*(G(k,j)*sin(sita(k)-sita(j))-B(k,j)*cos(sita(k)-sita(j))));
    end
end
for j=1:d %平衡节点有功和无功
PH_P=PH_P+U(d)*U(j)*(G(d,j)*cos(sita(d)-sita(j))+B(d,j)*sin(sita(d)-sita(j)));
PH_Q=PH_Q+U(d)*U(j)*(G(d,j)*sin(sita(d)-sita(j))-B(d,j)*cos(sita(d)-sita(j)));
end
S=zeros(d,d);
[n m]=size(A1);
for m=1:d
    U_ZJ(m)=U(m)*(cos(sita(m))+i*sin(sita(m)));
end
for m=1:n %线路节点间功率
S(A1(m,1),A1(m,2))=U_ZJ(A1(m,1))*conj(U_ZJ(A1(m,1)))+U_ZJ(A1(m,1))*(conj(U_ZJ(A1(m,1)))-conj(U_ZJ(A1(m,2))))*conj(A1(m,7));

S(A1(m,2),A1(m,1))=U_ZJ(A1(m,2))*conj(U_ZJ(A1(m,2)))+U_ZJ(A1(m,2))*(conj(U_ZJ(A1(m,2)))-conj(U_ZJ(A1(m,1))))*conj(A1(m,7));
end
[n m]=size(B1);
for m=1:n %变压器节点间功率

```

```

S(B1(m,1),B1(m,2))=U_ZJ(B1(m,1))*U_ZJ(B1(m,1))*conj(B1(m,4))+U_ZJ(B1(m,1))*(conj(U_ZJ(B1(m,1)))-conj(U_ZJ(B1(m,2))))*conj(B1(m,6));

S(B1(m,2),B1(m,1))=U_ZJ(B1(m,2))*U_ZJ(B1(m,2))*conj(B1(m,5))+U_ZJ(B1(m,2))*(conj(U_ZJ(B1(m,2)))-conj(U_ZJ(B1(m,1))))*conj(B1(m,6));
end
Pij=real(S);Qij=imag(S);
for m=1:d
    for n=1:d
I(m,n)=conj(S(m,n))/(U(m)*(cos(sita(m))-i*sin(sita(m))));I(n,m)=conj(S(n,m))/(U(n)*(cos(sita(n))-i*sin(sita(n))));
    end
end
S_sh=zeros(d,d);
for m=1:d
    for n=1:d
S_sh(m,n)=S(m,n)+S(n,m);
    end
end
dP=real(S_sh);dQ=imag(S_sh);
for j=1:d
    if j<=LL_n %联络节点
        PZR(j)=0;QZR(j)=0;
    elseif j>LL_n && j<=LL_n+PQ_n %PQ 节点
        PZR(j)=ZR(j-LL_n,2);QZR(j)=ZR(j-LL_n,3);
    elseif j==d %平衡节点
        PZR(j)=PH_P;QZR(j)=PH_Q;
    else %PV 节点
        PZR(j)=ZR(j-LL_n,2);QZR(j)=PV_Q(j);
    end
end
end
%%
%潮流计算结果以 TXT 方式输出
if DDCS>=4
    number_print=4;
else
    number_print=DDCS;
end
fid=fopen('结果.txt','w');
fprintf(fid,'\r\n');
fprintf(fid,'\t\t\t——导纳矩阵——\r\n\t\t\t\t');
for j=1:d
    fprintf(fid,'no.%d\t\t\t\t\t\t\t\t',j);
end

```

```
fprintf(fid, '\r\n');
for j=1:d
    fprintf(fid, 'no.%d\t', j);
    for k=1:d
        fprintf(fid, '%-10g+(%-10g*i) \t\t\t', G(j,k), B(j,k));
    end
    fprintf(fid, '\r\n');
end
fprintf(fid, '\r\n\r\n');
fprintf(fid, '-----\n');
fprintf(fid, '\r\n\r\n');
for j=1:number_print
    fprintf(fid, '迭代次数: %d\t\t\t 最大功率误差: %g', j, DDCS_GL(j));
    fprintf(fid, '\r\n');
end
fprintf(fid, '\r\n\r\n');
fprintf(fid, '\t\t\t\t\t\t\t*-*-*潮流计算结果*-*--\r\n');
fprintf(fid, '\t\t\t\t\t\t\t——节点潮流——\r\n');
fprintf(fid, 'no.i\t\t P\t\t\t\t Q\t\t\t\t\t U\t\t\t\t\t phase\r\n');
for j=1:d
    fprintf(fid, 'no.%d\t\t %-8.5g\t\t %-8.5g\t\t %-8.5g\t\t\n', j, PZR(j), QZR(j), U(j), sita(j));
end
fprintf(fid, '\r\n\r\n');
fprintf(fid, '\t\t\t\t\t\t\t——支路潮流——\r\n');
fprintf(fid, 'i\t\t j\t\t Pij\t\t\t\t Qij\t\t\t\t\t Pji\t\t\t\t\t Qji\t\t\t\t\t dP\t\t\t\t\t dQ\r\n');
[m n]=size(A1); %求线路矩阵的维数
for j=1:m
    h1=A1(j,1);
    h2=A1(j,2);
    fprintf(fid, '%d\t\t %d\t\t %-13g\t\t %-13g\t\t %-13g\t\t %-13g\t\t\n', h1, h2, Pij(h1, h2), Qij(h1, h2), Pij(h2, h1), Qij(h2, h1), dP(h1, h2), dQ(h1, h2));
    fprintf(fid, '\r\n');
end
[m n]=size(B1); %求变压器的维数
for j=1:m
    h1=B1(j,1);
    h2=B1(j,2);
    fprintf(fid, '%d\t\t %d\t\t %-13g\t\t %-13g\t\t %-13g\t\t %-13g\t\t\n', h1, h2, Pij(h1, h2), Qij(h1, h2), Pij(h2, h1), Qij(h2, h1), dP(h1, h2), dQ(h1, h2));
    fprintf(fid, '\r\n');
end
```

```
%-13g\t\t',h1,h2,Pij(h1,h2),Qij(h1,h2),Pij(h2,h1),Qij(h2,h1),dP(h1,h2)
,dQ(h1,h2));
    fprintf(fid, '\r\n');
end
```