

## 基于 Matlab 的 电力系统潮流计算

周卫星<sup>1</sup> 张颖<sup>2</sup>

(1. 长沙电力职业技术学院 长沙 410131; 2. 长沙理工大学 长沙 410076)

**摘要:** 提出了利用 MATLAB 语言来进行电力系统潮流计算。通过算例, 说明了该方法编程简便、运算效率高并符合人们的思维习惯, 计算结果能满足工程计算需要, 同时验证了方法的有效性, 为解决大电网的潮流计算问题开辟了新思路

**关键词:** 电力系统 潮流计算 Matlab

**中图分类号:** TP301.6

**文献标识码:** A

**文章编号:** 1673-0534(2007)04(a)-0070-02

潮流计算是在给定电力系统网络结构、参数和决定系统运行状态的边界条件的情况下确定系统稳态运行状态的一种基本方法, 是电力系统规划和运营中不可缺少的一个重要组成部分。可以说, 它是电力系统分析中最基本、最重要的计算, 是系统安全、经济分析和实时控制与调度的基础。是电力系统研究人员长期研究的一个课题。

MATLAB 自 1980 年问世以来, 以其学习简单、使用方便以及其它高级语言所无可比拟的强大的矩阵处理功能越来越受到世人的关注。目前, 它已成为国际控制界最流行、使用最广泛的语言了。它的强大的矩阵处理功能给电力系统的分析、计算带来许多方便。在处理潮流计算时, 其计算机软件的速度已无法满足大电网模拟和实时控制的仿真要求, 而高效的潮流问题相关软件的研究已成为大规模电力系统仿真计算的关键。随着计算机技术的不断发展和成熟, 对 MATLAB 潮流计算的研究为快速、详细地解决大电网的计算问题开辟了新思路。

## 1 潮流计算的数学模型

电力系统潮流计算的基本方程

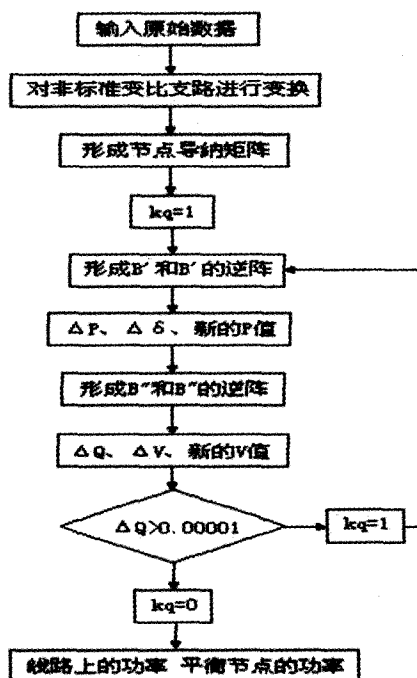


图1. 计算程序流程图

$$\sum_{j=1}^{j=n} Y_{ij} \dot{U}_j = \frac{P_i - jQ_i}{\dot{U}_i} \quad (i = 1, 2, 3 \dots n) \quad (1)$$

$$\sum_{j=1}^{j=n} Z_{ij} \dot{I}_j = \frac{P_i - jQ_i}{\dot{U}_i} \quad (i = 1, 2, 3 \dots n) \quad (2)$$

式中,  $P_i, Q_i$  分别为节点  $i$  向网络注入的有功功率和无功功率;  $\dot{U}_j$  为节点  $j$  的电压相量;  $\dot{U}_i$  为节点  $i$  的电压共轭相量;  $\dot{I}_j$  为节点  $j$  的电流相量;  $Y_{ij}$  为节点导纳矩阵;  $Z_{ij}$  为节点阻抗矩阵。式(1)和式(2)各有  $n$  个非线性复数方程, 对其作不同的应用和处理, 就形成了不同的潮流计算方法。其中, Newton-Laphson 法收敛性好, 是非线性方程数值求解的有效方法。该方法把非线性方程线性化, 由于线性方程的系数矩阵结构上是稀疏的非对称矩阵, 结合稀疏矩阵技术可使计算机内存占用量大大减少, 计算速度大大加快; P-Q 分解法是在 Newton-Laphson 法基础上, 将有功功率  $P$  和无功功率  $Q$  分开交替迭代的潮流计算方法, 该方法计算过程简单, 计算速度显著加快, 是目前常用的潮流计算方法。由于近代电力系统网络节点数量极大, 节点导纳阵  $Y_B$  的稀疏度也极高, 稀疏技术的形成是必不可少的。MATLAB 是集数值计算、符号运算及图形处理等强大功能于一体的科学计算语言。现已成工程计算中普遍采用的工具。MATLAB 拥有 600 多个工程数学运算函数, 可实现潮流计算中的矩阵求积、求逆、稀疏矩阵形成、复数运算以及初等数学运算等。MATLAB 语言允许用户以数学形式的语言编写程序, 使用 MATLAB 编程, 工作量会大为减少。要达到较高的计算精度, 且兼顾矩阵程序设计的难易程度, MATLAB 则成为首选潮流计算的计算机语言。

## 2 计算机程序的实现

## 2.1 导纳矩阵的形成

(1) 节点导纳矩阵是方阵, 其阶数等于网络中出参考节点外的节点数  $n$ 。

(2) 节点导纳矩阵是稀疏矩阵, 其各行非零非对角元数就等于该行相对应节点所连接的不接地支路数。

(3) 节点导纳矩阵的对角元就等于各该节点所连接导纳的总和。  $Y_{22} = Y_{20} + Y_{21} + Y_{23}$

(4) 节点导纳矩阵的非对角元  $Y_{ij}$  等于连接节点  $i, j$  支路导纳的负值。

(5) 节点导纳矩阵一般是对称矩阵。

(6) 对于支路中有非标准变比变压器的支路来说, 利用下面的公式来计算它的导

$$N = \frac{V_h/V_m}{V_l/V_n} = \frac{\omega_h/\omega_m}{\omega_l/\omega_n}$$

假如已知非标准变比支路  $i, j$  上的阻抗 (以下没有特殊说明所有的参数都用标么值) 为  $Z_{ij}$ 。则线路导纳为  $Y_{ij} = Z_{ij}/N$ , 线路上的对地半导纳为  $YQ_{ij} = 1 - N/(Z_{ij} * N * N)$ ,  $YQ_{ji} = -TQ_{ij}$ 。

## 2.2 B' 和 B'' 的形成

$B'$  并不是最终形成的导纳矩阵的虚部, 而是不考虑接地支路情况下形成的节点导纳矩阵的虚部, 平衡节点不参加其形成;  $B''$  是由最终形成的导纳矩阵的虚部组成的, 但是 pv 节点以及平衡节点不参加 Q-V 迭代, 因此  $B''$  中不包含与这些节点有关的元素。

注意: (1)  $B'$  和  $B''$  的阶数不同,  $B'$  为  $n-1$  阶, 低于  $n-1$  阶, 为  $n-1-pv$  节点数。

(2) 当潮流程序要考虑负荷静特性时,  $B''$  中的对角元素除导纳矩阵的对角元素的虚部以外, 还要附加反映负荷静特性的部分,

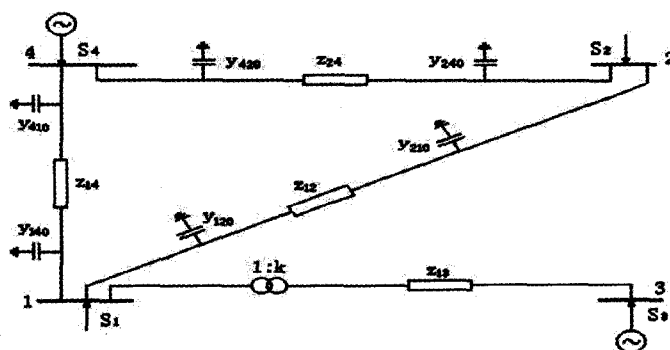


图2. 网络结构图

## 工业技术

而  $B'$  中的各个元素不需要考虑负荷静特性。

(3) 为了改善 P-Q 分解法的收敛特性,  $B'$  一般并不严格是电力系统导纳矩阵的虚部, 而  $B''$  则是由导纳矩阵的虚部组成。在本方案中采用的是前一种方法。

## 2.3 迭代条件和约束方程

迭代条件就是如果  $\Delta Q \leq 0.001$  时就停止迭代。

对节点的约束条件分为三类: 即对节点注入功率的约束、对节点电压大小的约束和对相角的约束。其中对节点注入功率的约束, 主要是对电源注入功率的约束条件不能满足时, 将威胁到发电机的安全运行。对电压大小的约束不能满足时, 将影响电能的质量, 严重时将影响系统运行的稳定性。对相对相角的约束条件不能满足时, 也将危及系统运行的稳定性。

## 2.4 计算机潮流计算的步骤

(1) 对电力网络的所有参数设初值, 包括电压、相角、有功、无功等。

(2) 处理非标准变比支路, 使其变成标准变比为 1 的变压器支路。

(3) 形成节点导纳矩阵  $Y$ 。

(4) 计算有功功率的不平衡量  $\Delta P_i^{(0)}$ , 从而求出  $\Delta P_i^{(0)}/U_i^{(0)} (i=1, 2, 3, \dots, n, i \neq S)$ 。

(5) 根据节点的类型形成  $B'$ 。

(6) 解修正方程式, 求各节点的电压相位角的变化量  $\Delta \delta_i^{(0)} (i=1, 2, 3, \dots, n, i \neq S)$ 。

(7) 求各节点相角的新值  $\delta_i^{(1)} = \delta_i^{(0)} + \Delta \delta_i^{(0)} (i=1, 2, 3, \dots, n, i \neq S)$ 。

(8) 计算无功功率的不平衡量, 从而求出。

(9) 根据节点的类型形成。

(10) 解修正方程式, 求各节点的电压大小的变化量  $\Delta U_i^{(0)} (i=1, 2, 3, \dots, n, i \neq S)$ 。

(11) 求各节点的电压大小的新值;

$U_i^{(1)} = U_i^{(0)} + \Delta U_i^{(0)} (i=1, 2, 3, \dots, n, i \neq S)$ 。

(12) 运用个节点的电压的新值自第四步开始下一次迭代。

(13) 计算平衡节点的功率和线路功率。其中平衡节点的功率的计算公式为

$$\tilde{S}_s = \sum_{i=1}^{n-1} \tilde{S}_{si} = P_s + jQ_s$$

线路上的功率为:

$$\tilde{S}_{ij} = \tilde{S}_{ji} = \tilde{S}_{ij} + (\tilde{S}_{ij} - \tilde{S}_{ji}) = P_{ij} + jQ_{ij}$$

$$\tilde{S}_{ji} = \tilde{S}_{ij} + (\tilde{S}_{ji} - \tilde{S}_{ij}) = P_{ji} + jQ_{ji}$$

从而线路上的损耗的功率为:

$$\Delta \tilde{S}_{ij} = \tilde{S}_{ij} - \tilde{S}_{ji} = \Delta P_{ij} + j\Delta Q_{ij}$$

## 2.5 计算机程序流程图

## 4 算例与分析 (运用 MATLAB 编写的潮流程序求解网络潮流)

## 4.1 网络图

## 4.2 网络原始数据

$$z_{12} = 0.10 + j0.40$$

$$y_{120} = y_{210} = j0.01528$$

$$z_{13} = j0.3, k = 1.1$$

$$z_{14} = 0.12 + j0.50$$

$$y_{140} = y_{410} = j0.01920$$

$$z_{24} = 0.08 + j0.40$$

$$y_{240} = y_{420} = j0.01413$$

系统中节点 1, 2 为 pq 节点, 节点 3 为 pv

节点, 节点 4 为平衡节点。已给定

$$P_s + jQ_s = -0.55 - j0.13$$

$$P_s = 0.5, V_s = 1.10, V_{\text{ref}} = 1.05 \angle 0^\circ$$

容许误差  $\varepsilon = 0.001$ 。

## 4.3 潮流计算数据和结果

## 4.3.1 给定的潮流计算的初始数据

$$n=4; nm=4; npv=1;$$

$$a = \begin{bmatrix} 1 & 2 & 0.10 & 0.40 & 0.01528 & 0 \\ 1 & 3 & 0 & 0.30 & 0 & 1.1 \\ 1 & 4 & 0.12 & 0.50 & 0.01920 & 0 \\ 2 & 4 & 0.08 & 0.40 & 0.01413 & 0 \end{bmatrix}$$

$$p = [-0.30, -0.55, 0.5, 0]$$

$$q = [-0.18, -0.13, 0, 0]$$

$$v = [1.0, 1.0, 1.10, 1.05]$$

$$\theta = [0, 0, 0, 0]$$

$$\text{type} = [1, 1, 2, 0]$$

## 4.3.2 最终的节点电压

$$vv = [0.9978 - 0.0122i \quad 0.9658 - 0.01093i \quad 1.0931 + 0.1230i \quad 1.0500]$$

## 4.3.3 最终线路上的功率

$$s = \begin{bmatrix} 0 & 0.2470 + 0.0429i & -0.4990 - 0.0093i \\ -0.2408 - 0.2305i & 0 & 0.3333i \\ 0.4990 + 0.2030i & 0.0278i & 0 \\ 0.0491 - 0.1421i & 0.3184 + 0.1852i & 0.3333i \end{bmatrix}$$

## 4.3.4 最终平衡节点功率

$$ss = 0.3675 + 0.2185i$$

## 5 结语

在电力系统的潮流计算算法中牛顿-拉夫逊法是得到电力系统研究人员认可的算法之一, 在本文中我们采用牛顿-拉夫逊法的改进型 P-Q 分解法, 主要是同时考虑到计算的准确和程序的运行速度。没有采用经常用的的高斯迭代法, 而是采用了传统的逆阵方法, 是考虑到用 MATLAB 实现高斯迭代将会通过很多的循环迭代才能实现, 而逆阵可以直接通过命令来求解, 这必然可以大大节省时间。对于不能求逆的矩阵我们通过在电力系统中至少有一条支路上有接地支路来实现其求逆。对于 P-Q 分解法不能使用于有些  $r/x$  比较大的电力系统的缺点, 可以通过在电力系统中并联补偿法或虚构节点来得以解决。通过实

例计算分析, 取得了比较满意的效果。基于 MATLAB 的电力系统潮流计算使计算机在计算、分析、研究复杂的电力系统潮流分布问题上又前进了一步。

不管采用什么算法, 所有的潮流计算都是基于矩阵的迭代运算。而 MATLAB 语言正是以处理矩阵见长, 实践证明, MATLAB 语言在电力系统潮流计算仿真研究中的应用是可行的, 而且由于其强大的矩阵处理功能, 完全可以应用于电力系统的其它分析计算中, 用 MATLAB 语言编程效率高, 程序调试十分方便, 可大大缩减软件开发周期, 如果像控制界一样开发出电力系统自己的专用工具箱, 将系统分析用的一些基本计算以函数的形式直接调用, 那么更高层次的系统软件也可以很容易地实现。

## 参考文献

- [1] 张伯明, 陈寿孙. 高等电力网络分析. 北京: 清华大学出版社, 1996.
- [2] 薛定宇. 控制系统计算机辅助设计. 北京: 清华大学出版社, 1996.
- [3] 刘军. MATLAB 在电力系统分析中的应用. 电力系统及自动化学报, 2000. 4.
- [4] 刘卫国, 陈昭平, 张颖. MATLAB 程序设计与应用. 北京: 高等教育出版社, 2002. 6.