

# 潮流方程的计算机解法

#电力系统分析

#郭庆来

## 主要内容

1. Gauss法
2. Newton-Raphson方法
3. PQ分解法
4. 重点理解不同潮流求解方法的优缺点

## 潮流计算机解法-基本历史

- 1956年，基于Y矩阵的Gauss迭代法
- 1960年，基于Z矩阵的Gauss迭代法
- 20世纪60年代，具有二阶收敛性的Newton-Raphson方法
- 20世纪60年代中后期，Tinney在N-R法中引入稀疏技术，大大降低了计算量，N-R法成为主流方法
- 1974年，Stott基于PQ解耦特性，提出了PQ分解法，计算速度大大提升，可以应用于在线系统

## Gauss法

### 基于Y矩阵的Gauss法

假设平衡节点复电压 $\dot{U}_s$ 已知，将其写在最后，列写如下等式：

$$\begin{bmatrix} Y_n & Y_s \\ Y_s^T & Y_{ss} \end{bmatrix} \begin{bmatrix} \dot{U}_n \\ \dot{U}_s \end{bmatrix} = \begin{bmatrix} \dot{I}_n \\ \dot{I}_s \end{bmatrix} \quad (1)$$

则有

$$Y_n \dot{U}_n = \dot{I}_n - Y_s \dot{U}_s \quad (2)$$

将  $Y_n$  分解为三个矩阵  $L$ （严格下三角阵）、 $D$ （对角阵）和  $U$ （严格上三角阵）之和，则（2）可以变为：

$$\dot{U}_n = D^{-1}(\dot{I}_n - Y_s \dot{U}_s - L \dot{U}_n - U \dot{U}_n) \quad (3)$$

这是一个 $x = g(x)$ 的不动点迭代格式，将复电流用复功率和复电压替换，则对于每一个节点，在迭代的第 $r$ 步：

$$\dot{U}_i^{(r+1)} = \frac{1}{Y_{ii}} \left( \frac{\hat{S}_i}{\hat{U}_i^{(r)}} - Y_{is} \dot{U}_s - \sum_{j=1}^{i-1} Y_{ij} \dot{U}_j^{(r)} - \sum_{j=i+1}^{n-1} Y_{ij} \dot{U}_j^{(r)} \right) \quad (4)$$

注意到一个事实，每一步迭代计算是从 $\dot{U}_1$ 开始，向着 $\dot{U}_n$ 一行一行计算的，这就意味着，当迭代的第 $r+1$ 步，计算到 $\dot{U}_i$ 时，下标小于 $i$ 的节点，已经完成了第 $r+1$ 次的迭代，因此对于 $j < i$ 的节点，此时在（4）中如果直接用最新迭代出来的值 $\dot{U}_j^{(r+1)}$ 计算，将比 $\dot{U}_j^{(r)}$ 更接近于真实解，此时称这种迭代格式为Gauss-Seidel迭代，式（4）变成：

$$\dot{U}_i^{(r+1)} = \frac{1}{Y_{ii}} \left( \frac{\hat{S}_i}{\hat{U}_i^{(r)}} - Y_{is} \dot{U}_s - \sum_{j=1}^{i-1} Y_{ij} \dot{U}_j^{\overline{(r+1)}} - \sum_{j=i+1}^{n-1} Y_{ij} \dot{U}_j^{(r)} \right) \quad (5)$$

局限性：因为节点导纳矩阵 $Y$ 是一个高度稀疏的矩阵，所以（5）中迭代的每一行，每次修正的有效范围都只是直接有边相连的节点，这决定了收敛性很差。

## 基于Z矩阵的Gauss法

为了改善收敛性，利用Z矩阵的满阵特性，可以构建基于Z矩阵的Gauss法  
由（2）式，可得：

$$\dot{U}_n = \tilde{Z}_n (\dot{I}_n - Y_s \dot{U}_s) \quad (6)$$

其中  $\tilde{Z}_n = Y_n^{-1}$   
则有：

$$\begin{aligned} \dot{U}_i^{(r+1)} &= \sum_{j=1}^{n-1} \tilde{Z}_{ij} \left( \frac{\hat{S}_j}{\hat{U}_j^{(r)}} - Y_{js} \dot{U}_s \right) && Gauss \\ &= \sum_{j=1}^{i-1} \tilde{Z}_{ij} \frac{\hat{S}_j}{\hat{U}_j^{\overline{(r+1)}}} + \sum_{j=i}^{n-1} \tilde{Z}_{ij} \frac{\hat{S}_j}{\hat{U}_j^{(r)}} - \sum_{j=1}^{n-1} \tilde{Z}_{ij} Y_{js} \dot{U}_s && Gauss - Seidel \end{aligned}$$

和基于 $Y$ 矩阵的Gauss法相比，Z矩阵法需要的存储量显著增加（不再稀疏），但收敛性变好。

## Newton-Raphson法

对于一个典型的非线性方程：

$$f(x) = y^{SP} - y(x) = 0$$

在第 $r$ 次迭代的 $x^{(r)}$ 处，做Taylor展开

$$f(x^{(r)}) + \left. \frac{\partial f}{\partial x^T} \right|_{x=x^{(r)}} \Delta x^{(r)} = 0 \quad (7)$$

由 (7) 可得：

$$\Delta x^{(r)} = - \left[ \left. \frac{\partial f}{\partial x^T} \right|_{x=x^{(r)}} \right]^{-1} f(x^{(r)}) \quad (8)$$

其中：

- $f(x^{(r)})$ 为**误差向量**，又称**失配量**，是将第 $r$ 次迭代结果 $x^{(r)}$ 代入后 $f(x)$ 后，仍然存在的误差值
- $\Delta x^{(r)}$ 为**修正向量**，是本次迭代的求解目标，下一次迭代  $x^{(r+1)} = x^{(r)} + \Delta x^{(r)}$
- $-\left[ \left. \frac{\partial f}{\partial x^T} \right|_{x=x^{(r)}} \right]$  是**Jacobi矩阵**，记为 $J(x^{(r)})$ ，表征了此次修正的方向，矩阵元素具体取值随着 $x^{(r)}$ 而变化，需要在**每步迭代时更新**

N-R法特点：

- 将非线性方程组转化为多次的线性方程组 (8) 的迭代求解
- 每次迭代，需要重新更新 $J(x^{(r)})$ ，并求逆，计算量很大
- 牛顿法具有二阶收敛性，收敛速度快（不等于计算速度快）
- 对初值敏感，初值 $x^{(0)}$ 越接近真解 $x^*$ 越好

## 直角坐标下的N-R法

直角坐标方程为

$$f(x) = \begin{bmatrix} \Delta P \\ \Delta Q \\ \Delta U^2 \end{bmatrix} = \begin{bmatrix} P^{SP} - P(e, f) \\ Q^{SP} - Q(e, f) \\ (U^{SP})^2 - e^2 - f^2 \end{bmatrix} = 0 \quad (9)$$

由 (3) 式，有

$$\begin{aligned} \begin{bmatrix} \Delta P \\ \Delta Q \\ \Delta U^2 \end{bmatrix} &= \begin{bmatrix} \frac{\partial \Delta P}{\partial e^T} & \frac{\partial \Delta P}{\partial f^T} \\ \frac{\partial \Delta Q}{\partial e^T} & \frac{\partial \Delta Q}{\partial f^T} \\ \frac{\partial \Delta U^2}{\partial e^T} & \frac{\partial \Delta U^2}{\partial f^T} \end{bmatrix} \begin{bmatrix} \Delta e \\ \Delta f \end{bmatrix} \\ &= \begin{bmatrix} H & N \\ M & L \\ R & S \end{bmatrix} \begin{bmatrix} \Delta e \\ \Delta f \end{bmatrix} \end{aligned} \quad (10)$$

矩阵中每部分的元素，需要根据 (4) 式进行推导。

$$\begin{cases} H_{ii} = \frac{\partial \Delta P_i}{\partial e_i} = -a_i - (G_{ii}e_i + B_{ii}f_i) \\ H_{ij} = \frac{\partial \Delta P_i}{\partial e_j} = -(G_{ij}e_i + B_{ij}f_i) \end{cases}$$

$$\begin{cases} N_{ii} = \frac{\partial \Delta P_i}{\partial f_i} = -b_i + (B_{ii}e_i - G_{ii}f_i) \\ N_{ij} = \frac{\partial \Delta P_i}{\partial f_j} = B_{ij}e_i - G_{ij}f_i \end{cases}$$

$$\begin{cases} M_{ii} = \frac{\partial \Delta Q_i}{\partial e_i} = b_i + (B_{ii}e_i - G_{ii}f_i) \\ M_{ij} = \frac{\partial \Delta Q_i}{\partial e_j} = B_{ij}e_i - G_{ij}f_i = N_{ij} \end{cases}$$

$$\begin{cases} L_{ii} = \frac{\partial \Delta Q_i}{\partial f_i} = -a_i + (G_{ii}e_i + B_{ii}f_i) \\ L_{ij} = \frac{\partial \Delta Q_i}{\partial f_j} = G_{ij}e_i + B_{ij}f_i = -H_{ij} \end{cases}$$

$$\begin{cases} R_{ii} = \frac{\partial \Delta U_i^2}{\partial e_i} = -2e_i \\ R_{ij} = \frac{\partial \Delta U_i^2}{\partial e_j} = 0 \end{cases}$$

$$\begin{cases} S_{ii} = \frac{\partial \Delta U_i^2}{\partial f_i} = -2f_i \\ S_{ij} = \frac{\partial \Delta U_i^2}{\partial f_j} = 0 \end{cases}$$

几点注意事项：

- 直角坐标下，Jacobi矩阵的维度为 $2(n-1)$ 阶方阵
- 本身矩阵是非对称的，但是每一个子块的非零元分布都和节点导纳矩阵 $Y$ 对应，因此是稀疏的，可以利用稀疏技术来提高计算效率
- 每步迭代时，都需要重新计算Jacobi矩阵数值，并求逆，因此计算量大

## 极坐标下的N-R法

极坐标下的潮流方程为 $n-1$ 个有功方程，和 $n-1-r$ 个无功方程：

$$f(x) = \begin{bmatrix} \Delta P(U, \delta) \\ \Delta Q(U, \delta) \end{bmatrix} = \begin{bmatrix} P^{SP} - P(U, \delta) \\ Q^{SP} - Q(U, \delta) \end{bmatrix} = 0 \quad (11)$$

则迭代方程为

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} \frac{\partial \Delta P}{\partial \delta^T} & \frac{\partial \Delta P}{\partial U^T} \\ \frac{\partial \Delta Q}{\partial \delta^T} & \frac{\partial \Delta Q}{\partial U^T} \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta U \end{bmatrix} \quad (12)$$

注意，求导过程中，对于 $\delta$ 求导，得到的仍然是 $U$ 的二次型，而对于 $U$ 求导，二次型就变成了一次函数，为了保持与前面对 $\delta$ 求导结果的一致性，我们将 $J$ 矩阵和 $U$ 相关的部分恢复成二次函数：

$$\begin{aligned}
\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} &= \begin{bmatrix} \frac{\partial \Delta P}{\partial \delta^T} & \frac{\partial \Delta P}{\partial U^T} \\ \frac{\partial \Delta Q}{\partial \delta^T} & \frac{\partial \Delta Q}{\partial U^T} \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta U \end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial \Delta P}{\partial \delta^T} & \frac{\partial \Delta P}{\partial U^T} U \\ \frac{\partial \Delta Q}{\partial \delta^T} & \frac{\partial \Delta Q}{\partial U^T} U \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta U/U \end{bmatrix} \\
&= \begin{bmatrix} H & N \\ M & L \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta U/U \end{bmatrix}
\end{aligned} \tag{13}$$

可推导得到Jacobi矩阵每部分的元素取值。

$$\begin{cases} H_{ii} = \frac{\partial \Delta P_i}{\partial \delta_i} = -U_i \sum_{j=1, j \neq i}^n U_j (-G_{ij} \sin \delta_{ij} + B_{ij} \cos \delta_{ij}) = Q_i + U_i^2 B_{ii} \\ H_{ij} = \frac{\partial \Delta P_i}{\partial \delta_j} = -U_i U_j (G_{ij} \sin \delta_{ij} - B_{ij} \cos \delta_{ij}) \end{cases}$$

$$\begin{cases} N_{ii} = \frac{\partial \Delta P_i}{\partial U_i} U_i = -P_i - U_i^2 G_{ii} \\ N_{ij} = \frac{\partial \Delta P_i}{\partial U_j} U_j = -U_i U_j (G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij}) \end{cases}$$

$$\begin{cases} M_{ii} = \frac{\partial \Delta Q_i}{\partial \delta_i} = -U_i \sum_{j=1, j \neq i}^n U_j (G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij}) = -P_i + U_i^2 G_{ii} \\ M_{ij} = \frac{\partial \Delta Q_i}{\partial \delta_j} = U_i U_j (G_{ij} \cos \delta_{ij} + B_{ij} \sin \delta_{ij}) = -N_{ij} \end{cases}$$

$$\begin{cases} L_{ii} = \frac{\partial \Delta Q_i}{\partial U_i} U_i = -Q_i + U_i^2 B_{ii} \\ L_{ij} = \frac{\partial \Delta Q_i}{\partial U_j} U_j = -U_i U_j (G_{ij} \sin \delta_{ij} - B_{ij} \cos \delta_{ij}) = H_{ij} \end{cases}$$

注意事项：

- 极坐标下，Jacobi矩阵为  $2(n-1)$ -r阶方阵，维度比直角坐标下小
- 同样不是对称矩阵，但是每一个子块的非零元分布都和节点导纳矩阵Y对应，因此是稀疏的，可以利用稀疏技术来提高计算效率
- 每步迭代时，都需要重新计算Jacobi矩阵数值，并求逆，因此计算量大
- 选择初值的时候，一般选择  $U^{(0)} = 1.0$ ， $\delta^{(0)} = 0$ ，这是标么制的好处

## PQ分解法

N-R法收敛性好，但是每次迭代都需要重新更新Jacobi矩阵，重新求逆，因此计算量和存储量都比较大。Stott利用电力系统P-Q解耦特点，从**极坐标N-R法出发**，经过**降阶**、**常数化**两步简化，提出了**PQ分解法**，从而大大提升了计算速度，实现**在线应用**。

基本简化思路：

### 1. 降阶

从公式（6）对应的修正方程出发，注意到当  $R \ll X$  时， $\delta$ 变化主要影响P， $U$ 变化主要影响Q，所以  $N < H$ ， $M < L$ ，因此，我们可以忽略非对角块，得到一个近似的修正方程：

- - - - -

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} H & 0 \\ 0 & L \end{bmatrix} \begin{bmatrix} \Delta \delta \\ \Delta U/U \end{bmatrix} \quad (14)$$

展开可得：

$$\begin{cases} \Delta P = -H \cdot \Delta \delta \\ \Delta Q = -L \cdot \Delta U/U \end{cases} \quad (15)$$

按照（8）式的迭代格式， $P$ 的迭代只和 $\delta$ 相关， $Q$ 的迭代只和 $U$ 相关，二者可以解耦进行，这也是该方法被称之为**快速分解法**的原因。这样原来（6）式要进行的高维的矩阵计算，变成了两个单独进行的**降阶**后的矩阵计算，计算量可以显著下降。

## 2. 常数化

（8）式中的 $H$ 和 $L$ 依然要随着迭代的进行而实时更新，重新计算，所以计算量还是太大，能否常数化？

首先看非对角元。注意到， $H$ 和 $L$ 非对角元元素的取值：

$$H_{ij} = L_{ij} = -U_i U_j (G_{ij} \sin \delta_{ij} - B_{ij} \cos \delta_{ij})$$

一般线路两端的 $\delta_{ij}$ 很小，且 $G_{ij} \ll B_{ij}$ ，所以有：

$$\cos \delta_{ij} \approx 1$$

$$G_{ij} \sin \delta_{ij} \ll B_{ij} \cos \delta_{ij}$$

简化后：

$$\begin{aligned} H_{ij} &= U_i U_j B_{ij} & i, j &= 1, 2, \dots, n-1, i \neq j \\ L_{ij} &= U_i U_j B_{ij} & i, j &= 1, 2, \dots, n-r-1, i \neq j \end{aligned} \quad (16)$$

再看一下对角元：

$$\begin{aligned} H_{ii} &= Q_i + U_i^2 B_{ii} \\ L_{ii} &= -Q_i + U_i^2 B_{ii} \end{aligned} \quad (17)$$

从物理意义上可知， $U_i^2 B_{ii} \gg Q_i$ ，所以可以忽略 $Q_i$ ，得

$$\begin{aligned} H_{ii} &= U_i^2 B_{ii} & i &= 1, 2, \dots, n-1 \\ L_{ii} &= U_i^2 B_{ii} & i &= 1, 2, \dots, n-r-1 \end{aligned} \quad (18)$$

由此  $H$ 和 $L$ 都可以写成 $UBU$ 的形式，将左边的对角阵 $U$ 除到（8）式左侧，右侧的对角阵 $U$ 和右侧变量乘在一起，可得

$$\begin{cases} \Delta P/U = -B' \cdot U \Delta \delta \\ \Delta Q/U = -B'' \cdot \Delta U \end{cases} \quad (19)$$

注意事项：

- 矩阵 $B'$ 、 $B''$ 就是**节点导纳矩阵 $Y$ 的虚部**，是**稀疏、对称的实数矩阵**。

- 矩阵 $B'$ 、 $B''$ 维度不同， $B'$ 是 $(n-1) \times (n-1)$ 阶， $B''$ 是 $(n-1-r) \times (n-1-r)$ 阶
- 矩阵 $B'$ 、 $B''$ 是常系数矩阵，根据节点导纳矩阵可以直接生成，一经生成，就保持不变，可以一次求逆，然后在每次迭代中反复使用。所以整个计算量大大减少
- 存储量小、计算量小，特别适用于在线分析

## PQ分解法与N-R法对比

- PQ分解法收敛性比N-R差（Why?），但是计算速度快于N-R法（Why?）。（注意区分收敛速度和计算速度）
- 只要收敛判据相同，二者的计算精度就相同。各种简化只影响迭代过程，但是不影响最终的结果