
几种定雅可比法的潮流计算方法的比较

朱贝尔 电硕 161 2016210730

1 引言

潮流计算是电力系统分析中最基本、最重要的运算。上世纪 50 年代，随着电子计算机的发展，人们开始在计算机上开始进行潮流计算，最初以 Gauss-Seidel 和 Ward-Hale 迭代法为工具。这些方法内存需求小，运算也相对简单，但收敛性差。Newton 法在潮流计算中有着很好的收敛性，并随着稀疏矩阵和节点优化编号技术的提出在解题规模和计算效率上有了进一步的提升，但是 Newton 法涉及大量计算，计算速度并不快。20 世纪 70 年代中期，Stott 在大量数值计算实践的基础上提出了 FDL 法，固定了雅可比矩阵，加快了计算速度。1989 年，荷兰学者 Van Amerongen 在大量数值实验中发现了另一种快速潮流分解法并称之为 BX 法，该算法在支路的 R/X 较大的情况下比 Stott 的快速分解法有更好的收敛性能。

本次实验实现了 5 种定雅可比法的潮流计算方法，分别为考虑 PQ 耦合的定雅可比法，XB, BX, BB 和 XX 法。程序的主要算例来自 Matpower 的 case{14, 39, 57, 118, 2383}。本人通过 matlab 导出数据，考虑到编程习惯，用 python2.7 和 shell 进行了主要程序的编写，并将计算结果和 Matpower 的结果进行对比验证了正确性，并比较了这 5 种潮流计算方法。

2 潮流算法

2.1 定雅可比矩阵法

极坐标的下的定雅可比法的修正写法如下：

$$-\begin{bmatrix} B_H & -G_N \\ G_M & B_L \end{bmatrix} \begin{bmatrix} V\Delta\theta \\ \Delta V \end{bmatrix} = \begin{bmatrix} \Delta P/V \\ \Delta Q/V \end{bmatrix} \quad (1)$$

定雅可比矩阵是常数，只要在迭代开始时形成因子表，在以后的迭代过程中就可以连续使用，因此这是一种固定斜率的 Newton 法，具有一阶收敛性。由于每次迭代不用形成新的雅可比矩阵，也不用形成新的因子表，所以总的运算速度比 Newton 法大大提高。

2.2 XB(Stott)算法

经验表面，实际电力系统如果满足：

1. 电压在额定值附近
2. 线路两端的相角相差不大

3. 线路的电阻比电抗小

我们可以忽略雅可比矩阵的非对角块，即把 G_M 和 G_N 设为 0，从而实现有功和无功潮流的解耦。此外，Stott 忽略 B_H 中的支路电阻和接地之路的影响，即用 $-1/x$ 作为支路电纳建立的节点电纳矩阵 B' 代替 B_H ；对于 B_L ，用节点导纳矩阵中不含 PV 节点的虚部 B'' 代替。 $V\Delta\theta$ 中的电压用 1 代替。于是得到以下的修正方程。

$$-B'\Delta\theta = \Delta P/V \quad (2)$$

$$-B''\Delta V = \Delta Q/V \quad (3)$$

假定当前点为 $(\theta^{(k)}, V^{(k)})$ ，则求解 $(\theta^{(k+1)}, V^{(k+1)})$ 的连续迭代格式如下：

$$\begin{cases} \Delta V^{(k)} = -B''^{-1} \Delta Q(\theta^{(k)}, V^{(k)})/V^{(k)} \\ V^{(k+1)} = V^{(k)} + \Delta V^{(k)} \end{cases} \quad (4)$$

$$\begin{cases} \Delta\theta^{(k)} = -B'^{-1} \Delta P(\theta^{(k)}, V^{(k+1)})/V^{(k+1)} \\ \theta^{(k+1)} = \theta^{(k)} + \Delta\theta^{(k)} \end{cases} \quad (5)$$

其中 $P-\theta$ 和 $Q-V$ 迭代分别交替进行。有功无功的偏差都用电压幅值去除。此方法成为快速分解法中的 XB 算法。

2.3 BX(Van Amerongen)算法

1989 年，荷兰学者 Van Amerongen 在 a General-Purpose Version of the Fast Decoupled Load Flow 中提出了 BX 型算法。BX 算法在线路电阻电抗(R/X)比值较大时，比 XB 算法有着更好的收敛性能。BX 型的处理方式是：在对系数矩阵 B_H 进行简化时，保留了之路电阻的影响，但忽略了接地支路项。在对系数矩阵 B_L 进行简化时，完全忽略之路电阻的影响，但保留接地之路项。

2.4 BB 算法

BB 型的处理方式是：在对系数矩阵 B_H 进行简化时，保留了之路电阻的影响，但忽略了接地支路项。在对系数矩阵 B_L 进行简化时，保留了之路电阻的影响和接地之路项。这种形式的快速分解法收敛性能很差。

2.5 XX 算法

XX 型的处理方式是：在对系数矩阵 B_H 进行简化时，忽略了之路电阻和接地支路项。在对系数矩阵 B_L 进行简化时，忽略了之路电阻的影响，保留了接地之路项。这种形式的快速分解法收敛性不会比 BX 或 XB 法更好。

3 实验结果

实验若无特殊说明, 收敛判据为 $\max(|\Delta P|, |\Delta Q|) < 10^{-6}$ 。最大迭代次数为 1000。如果某次迭代中 $\Delta P, \Delta Q$ 溢出则直接终止程序, 认为不收敛。

所有实验的复现在 README.md/README.pdf 文件中有描述。

3.1 准确性分析

潮流计算的最基本要求是潮流解的正确性, 这与收敛判据有关, 与算法是否收敛有关, 与算法好坏无关。我们采用 Matpower 中 case{39, 57} 算例作为验证, 比较几种定雅可比算法得出的值和 Matpower 给出的参考值之差的无穷范数, 精度为 $1e-6$ 。若 $\max|V - V_0|$ 和 $\max|\theta - \theta_0|$ 小于 $1e-6$, 认为其值为 0。

case 39	coupled CJ	BB	XB	BX	XX
$\max V - V_0 $	0	0	0	0	0
$\max \theta - \theta_0 $	0	0	0	0	0

表格 1 case 39 的精度比较

case 57	coupled CJ	BB	XB	BX	XX
$\max V - V_0 $	0	nc	0	0	0
$\max \theta - \theta_0 $	0	nc	0	0	0

表格 2 case 57 的精度比较 nc 表示不收敛

不考虑 BB 法在 case57 下不收敛的例子, 其他方法的计算结果和 Matpower 计算值吻合, 可验证程序的正确性。

3.2 迭代次数

在快速分解法中, 由于运算都遵循(2)(3)式的迭代格式, 所以只要比较迭代次数理论上便可比较几种方法的运行时间。我们采用 Matpower 中 case{14, 39, 57, 118, 2383} 作为算例。

case	Coupled CJ	BB	XB	BX	XX
14	8	36	7	9	13
39	131	19	8	9	14
57	10	nc	8	8	20
118	nc	10	9	7	9
2383	nc	138	12	15	16

表格 3 不同定雅可比法迭代次数

case	Coupled CJ	BB	XB	BX	XX
14	0.009080	0.036675	0.007089	0.010882	0.013286

39	1.017603	0.141874	0.141874	0.061109	0.067341
57	0.172313	nc	0.135703	0.138849	0.336292
118	nc	0.580857	0.522363	0.415349	0.518643
2383	nc	2088.415025	185.363306	233.337175	242.970785

表格 4 不同定雅可比迭代运行时间 单位 s

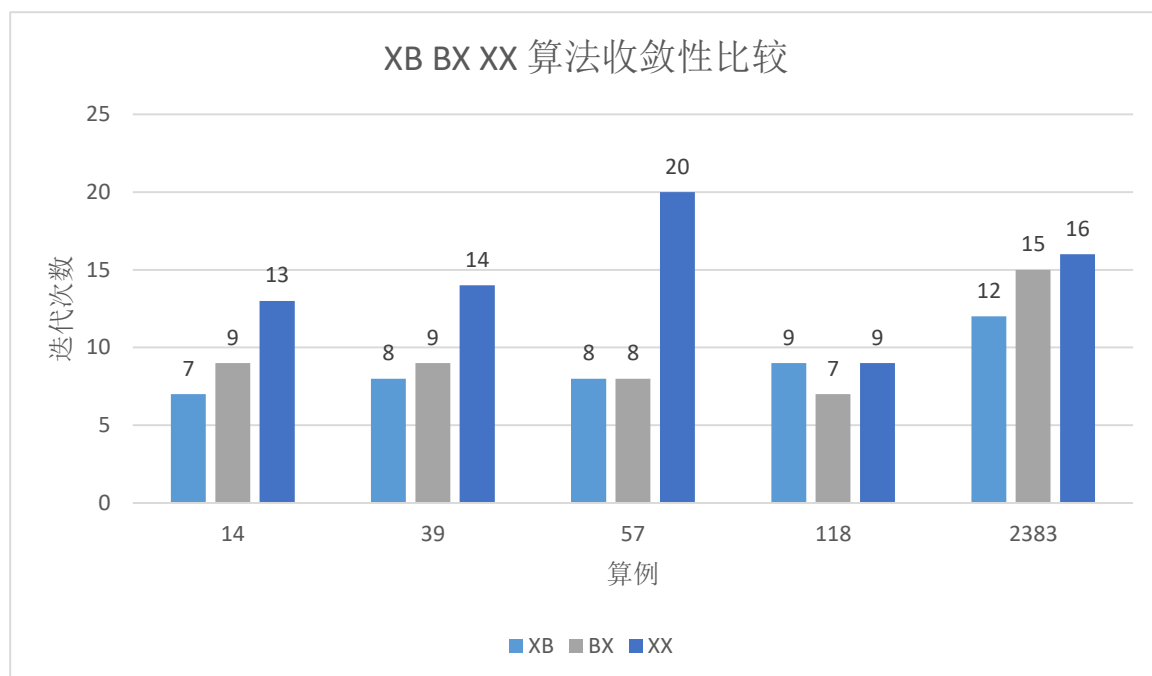


图 1 XB BX XX 收敛性比较

1. 在上面 5 个算例中，只有 XB, BX 和 XX 是都收敛的，XB 法和 BX 法所需要的迭代次数最少，收敛性能最佳，BX 法稍稍弱于 XB 法。XB 和 BX 的性能在这个案例中相差不大，其具体讨论在后面实验 3.4 中提及。
2. 可以看出耦合定雅可比法和 BB 法的收敛性能明显不如 XB 法和 BX 法，迭代次数明显高于其他三种方法。BB 在 57 算例下，耦合定雅可比在 118, 2383 算例下甚至不收敛。
3. XX 法相对比 BB 法和耦合定雅可比收敛性能好些，但不会比 XB 法和 BX 法好。

XB(Stott)法的讨论

3.3 XB 法的讨论

3.3.1 B' 中考虑支路 r 的影响

根据书中推导可知，用 $-1/x$ 形成的 B' 反而考虑了无功对有功的影响，比直接用节点导纳矩阵形成的 B' 有更好的收敛性，郭老师也在课上提及过这一点。现在对比这种方法的性能。

case	不考虑支路电阻	考虑支路电阻
14	7	36
39	8	20
57	8	nc
118	9	10

表格 5 考虑支路 r 的迭代次数对比

从表格 5 可以看出不考虑 r 的收敛性明显优于考虑 r 的收敛性。正如书上所证明的，省略 r 简化的 B' 反而考虑了 P 和 Q 的耦合，比直接用 Y 矩阵建立的 B' 有更好的收敛性。

3.3.2 不忽略接地支路项的影响

因为 B' 是针对有功潮流的计算，而接地支路上并不通有功潮流，所以在建立 B' 时忽略接地支路的影响，这改善了收敛性能。对是否忽略接地支路项，得到如下结果：

case	不考虑接地支路	考虑接地支路
14	7	8
39	8	122
57	8	8
118	9	nc

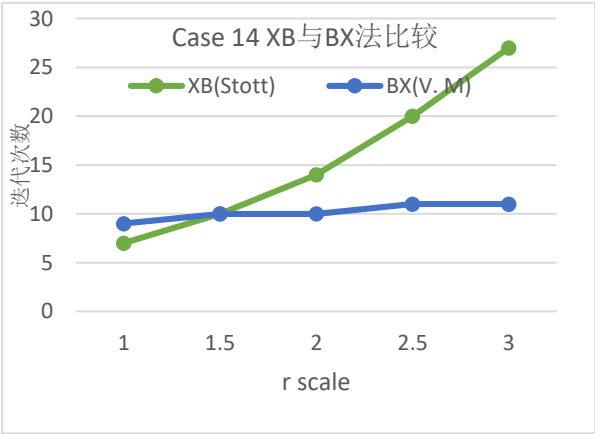
表格 6 考虑接地支路对 B' 影响的迭代次数对比

14 节点和 57 节点两种算例基本没有接地之路，所以二者收敛性能基本相同。在 39 节点和 118 节点中，不考虑接地支路的算法收敛性好于考虑接地支路算法收敛性。

3.4 支路 R/X 的大小对 XB 和 BX 收敛性的影响

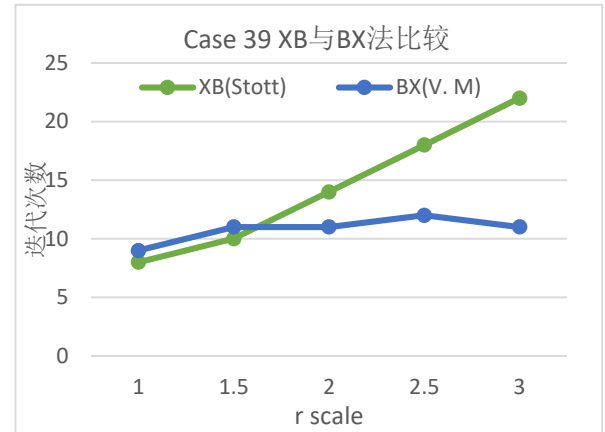
为了解决 XB 法在支路 R/X 较大情况下收敛性能不佳的问题，Van Amerongen 在 1989 年提出 BX 法。BX 在 R/X 比值增加的情况下，比 Stott 的 XB 的收敛速度更快。在本部分，所有支路上的电阻 r 和一个系数因子相乘，系数因子取 $\{1, 1.5, 2, 2.5, 3\}$ 。采用 case $\{14, 39, 57, 118\}$ 案例分析

case 14		
r scale factor	XB(Stott)	BX(V. Amerongon)
1	7	9
1.5	10	10
2	14	10
2.5	20	11
3	27	11



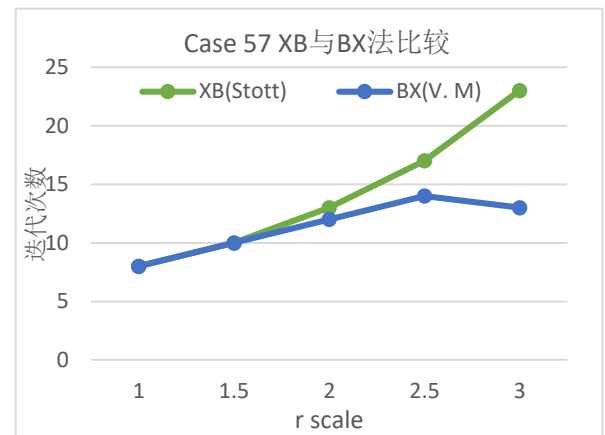
表格 7 case 14 下 不同 r scale 下 XB 和 BX 的比较

case 39		
r scale	XB(Stott)	BX(V. Amerongon)
factor		
1	8	9
1.5	10	11
2	14	11
2.5	18	12
3	22	13



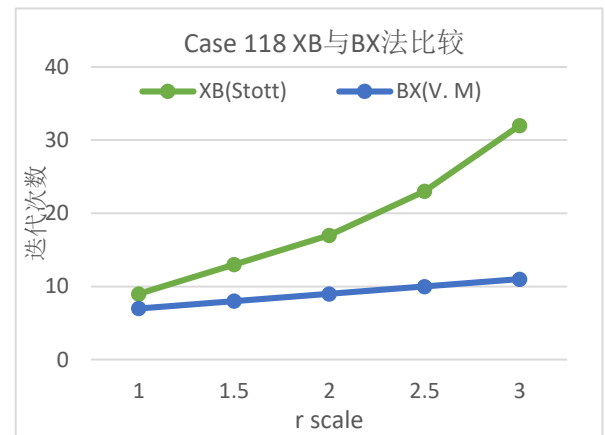
表格 8 case 39 下 不同 r scale 下 XB 和 BX 的比较

case 57		
r scale	XB(Stott)	BX(V. Amerongon)
factor		
1	8	8
1.5	10	10
2	13	12
2.5	17	14
3	23	16



表格 9 case57 下 不同 r scale 下 XB 和 BX 的比较

case 118		
r scale	XB(Stott)	BX(V. Amerongon)
factor		
1	9	7
1.5	13	8
2	17	9
2.5	23	10
3	32	11



表格 10 case118 下 不同 r scale 下 XB 和 BX 的比较

通过以上表格可以得到如下结论：

1. 在 r/x 比例较小的情况下，没有必要将 XB(Stott)法变为 BX(V. Amerongen)法，在这种情况下 BX 法迭代收敛性稍稍弱于 XB 法。
2. 在 r/x 变大的情况下，BX 法的收敛性明显好于 XB 法。当 r scale 为 3 的时候，case14 中 BX 迭代次数为 BX 的 40.74%，case39 中为 59.09%，case57 中为 69.67%，case118 为 34.38%。

此外，我通过减少 x 来提高 r/x ，也做了一系列的实验，实验结论和增大 r 的一致，故不再赘述。实验结果如下所示。

x	case 14		case 39		case 57		case 118	
scale	XB	BX	XB	BX	XB	BX	XB	BX
1	7	9	8	9	8	8	9	7
0.75	9	10	9	10	9	9	11	8
0.5	12	12	14	11	13	11	17	9
0.25	30	14	32	15	30	19	45	11

表格 11 不同 x scale 下 BX 和 XB 收敛迭代次数比较

4 总结

本次实验，本人通过 Python 实现了耦合定雅可比，BB, XB, BX 和 XX 算法，并通过 Matpower 加以验证。通过这次实验，我对电网中的潮流解法有了更深刻的认识，并得到主要有以下几个结论：

1. 5 种潮流计算方法中，通过对比计算时间和迭代次数，BX 和 XB 法的计算速度最高。
2. BB 法和耦合定雅可比方法收敛性一般，有些算例中甚至不收敛
3. XX 法性能居于 BB, CCJ 和 XB, BX 法之间。
4. XB 法中 B' 忽略支路 r 和接地支路提升了收敛性能。
5. BX 法在支路的 r/x 值较大时的收敛性比 XB 法好。

5 主要参考文献

- [1] 张伯明.高等电力网络分析(第三版)[M].北京:清华大学 出版社,2007.
- [2] Van Amerongen R A M. A general-purpose version of the fast decoupled load flow. IEEE Trans. on power systems, May 1989, PWRS-4(2):760~770