

用于迭代法潮流计算的改进 Jacobi 预处理方法

唐 灿¹, 董树锋¹, 任雪桂², 尹 璐², 鞠 力²

(1. 浙江大学电气工程学院, 浙江省杭州市 310027; 2. 北京电力经济技术研究院, 北京市 100055)

摘要: 为提高潮流计算速度, 满足实时计算的要求, 线性方程组迭代法被用于电力系统潮流计算。但是当系数矩阵谱分布较为分散时, 迭代法求解线性方程组存在收敛速度慢甚至不收敛等问题, 为了解决这个问题, 需对系数矩阵进行预处理。首先, 分析电力系统潮流计算时 Jacobi 矩阵的特点, 对其按 PV, PQ 节点进行分块处理, 找出其中数值上较大的元素作为预处理子。然后, 将预处理子的逆矩阵分别与系数矩阵 A 和常量项 b 相乘, 将原线性方程组转换为新的更容易求解的等价线性方程组, 大幅提高了潮流计算中线性方程组求解的速度。实验表明, 该方法能有效解决大规模电网潮流求解问题。

关键词: 潮流计算; 稀疏线性方程组; 预处理; 稀疏矩阵; 图形处理器通用计算

0 引言

牛顿-拉夫逊法是常见的电力系统交流潮流计算方法之一。这一方法需要多次迭代运算, 每一次迭代过程中都要求解线性方程组。当方程组的规模较大时, 求解方程组非常耗时, 潮流计算的效率受到较大影响。

直接法^[1]和迭代法^[2-5]是求解线性方程组的两类方法。其中直接法主要利用矩阵分解技术求解, 例如 LU 分解法等, 虽然直接法能通过有限步骤算出精确解, 但计算复杂度较高, 且计算过程不利于并行处理, 不适合求解大规模的线性方程组。相较于直接法, 迭代法易于并行计算, 在求解大规模线性方程组时具有明显的优势^[6]。随着图形处理器 (GPU) 的飞速发展, 中央处理器 (CPU) 与 GPU 异构协同的计算体系使得串行计算与并行计算协调运作, 显著提高了计算能力^[7-10], 因此可以考虑利用这一计算体系实现迭代法的潮流计算方法。

但是, 迭代法也有不足, 相比于直接法具有不稳定性。迭代法的收敛速度与系数矩阵的条件数和谱分布紧密相关, 当谱分布较为分散时, 迭代法的收敛速度明显降低甚至会发生不收敛的情况。为保证迭代法线性方程组求解时的稳定性, 提高求解速度, 需要将原方程组转换为等价的、易于求解的线性方程组。针对这一问题, 通常采用预处理技术, 即通过对

系数矩阵进行预处理, 改善系数矩阵的谱分布, 使其分布地更为集中, 从而提高迭代法的稳定性及收敛速度。

在进行预处理时, 往往存在预处理效果与预处理时间的矛盾, 要想取得较好的预处理效果往往需要耗费更多时间。本文提出了一种适合于超大规模电力系统潮流计算中 Jacobi 矩阵的预处理方法, 这一方法基于适于并行的 Jacobi 预处理法, 并充分利用 Jacobi 矩阵自身特点对传统预处理方法进行改进。并行求解其预处理子, 不仅拥有高效的求解速度, 同时拥有超过其他轻量级预处理子的效果。经算例测试, 所提的预处理方法能有效集中谱分布, 改善条件数, 提高方程组求解速度。

1 Krylov 子空间迭代法的预处理

Krylov 子空间迭代法是一种高效求解大型稀疏线性方程组的有效方法^[11-13]。利用 Krylov 子空间迭代法求解线性方程组时, 其收敛速度取决于其谱分布^[14-15]。谱分布越集中, 其收敛速度越快。

为了提高收敛速度, 可以对系数矩阵进行预处理。一般来说, 预处理包括以下三种形式。

$$M^{-1}Ax = M^{-1}b \quad (1)$$

$$\begin{cases} AM^{-1}y = b \\ x = M^{-1}y \end{cases} \quad (2)$$

$$\begin{cases} M_1^{-1}AM_2^{-1}y = M_1^{-1}b \\ x = M_2^{-1}y \\ M = M_1M_2 \end{cases} \quad (3)$$

式中: A 为线性方程组的系数矩阵; b 为常数项; x

为待求向量; \mathbf{y} 为中间向量; \mathbf{M} 为预处理子, \mathbf{M}_1 和 \mathbf{M}_2 由预处理子分解得到。

式(1)中分别对 \mathbf{Ax} 和 \mathbf{b} 左乘 \mathbf{M}^{-1} , 然后求解新的线性方程组, 其结果与原方程的解相等; 式(2)中对 \mathbf{A} 右乘 $\mathbf{M}^{-1}\mathbf{y}$, 可看作是 $\mathbf{AM}^{-1}(\mathbf{Mx}) = \mathbf{b}$, 求解得到 \mathbf{Mx} , 对结果左乘 \mathbf{M}^{-1} 即可得到与原方程相同的解; 式(3)是式(1)和式(2)的结合, 同样可以得到与原方程同样的解。对于本文所选取的预处理子 \mathbf{M} , 既可以使用式(1)与式(2)进行预处理, 也可以将 \mathbf{M} 分解为 $\mathbf{M}_1\mathbf{M}_2$, 并采用式(3)进行预处理。

预处理子的选择除了应当使得处理后的方程组更易求解外, 也要注意保证原系数矩阵的稀疏性, 尽可能控制非零元注入的数量。此外, 所选的预处理子的构造过程的计算开销应尽可能低。

常用的预处理方法包括不完全分解预处理^[16-21]、近似逆预处理^[22-23]和多项式预处理^[24-26]等。不完全分解方法的实现过程难以并行, 不适用于系数矩阵规模较大的情形; 近似逆预处理容易破坏稀疏矩阵逆矩阵的稀疏性; 多项式预处理方法虽具有自然可并行性, 但是这一方法得到的预处理子的预处理效果较差。

在实际计算中, Jacobi 预处理法是较为常用的预处理方法。Jacobi 预处理法直接取矩阵的对角元作为预处理子, 其主要优势在于其预处理的速度非常快, 缺点是仅适用于对角元占主导的矩阵。电力系统的潮流计算的 Jacobi 矩阵是一种类对角元占主导的矩阵, 采用 Jacobi 预处理法能够在一定程度上提高潮流计算的速度。但是, 潮流计算的 Jacobi 矩阵并非真正意义上的对角元占主导的矩阵, 为了进一步提高预处理的效果, 本文基于适于并行的 Jacobi 预处理, 充分利用 Jacobi 矩阵自身特点, 设计了新的预处理子。

2 改进 Jacobi 预处理方法

在潮流计算中, 极坐标表示的节点功率方程为^[27]:

$$\begin{cases} \mathbf{P}_i - \mathbf{U}_i \sum_{j=1}^n \mathbf{U}_j (\mathbf{G}_{ij} \cos \delta_{ij} + \mathbf{B}_{ij} \sin \delta_{ij}) = 0 \\ \mathbf{Q}_i - \mathbf{U}_i \sum_{j=1}^n \mathbf{U}_j (\mathbf{G}_{ij} \sin \delta_{ij} - \mathbf{B}_{ij} \cos \delta_{ij}) = 0 \end{cases} \quad (4)$$

式中: \mathbf{P}_i 和 \mathbf{Q}_i 分别为节点 i 注入的有功功率和无功功率; \mathbf{U}_i 和 \mathbf{U}_j 分别为节点 i 和节点 j 的电压幅值; δ_{ij} 为节点 i 和 j 的相角差; \mathbf{G}_{ij} 和 \mathbf{B}_{ij} 分别为节点 i 和节点 j 间的互电导与互电纳。

由此可得到描述电力系统的非线性方程。根据牛顿-拉夫逊算法, 线性化潮流方程可以得到修正

方程组^[27]为:

$$\begin{bmatrix} \Delta \mathbf{P} \\ \Delta \mathbf{Q} \end{bmatrix} = \begin{bmatrix} \mathbf{H} & \mathbf{N} \\ \mathbf{J} & \mathbf{L} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{\theta} \\ \frac{\Delta \mathbf{V}}{\mathbf{V}} \end{bmatrix} \quad (5)$$

式中: $\Delta \mathbf{P}$ 和 $\Delta \mathbf{Q}$ 分别为节点实际注入功率与迭代过程中值的差额; \mathbf{H} , \mathbf{N} , \mathbf{J} , \mathbf{L} 分别为迭代过程中 Jacobi 矩阵的 4 个子矩阵; $\Delta \mathbf{V}$ 和 $\Delta \boldsymbol{\theta}$ 为相邻两次迭代中电压幅值与相角之间的差值。其中, \mathbf{H} 为 $n-1$ 阶方阵; \mathbf{N} 为 $(n-1) \times (n-1-r)$ 阶矩阵; \mathbf{J} 为 $(n-1-r) \times (n-1)$ 阶矩阵; \mathbf{L} 为 $n-1-r$ 阶方阵。 \mathbf{H} , \mathbf{N} , \mathbf{J} , \mathbf{L} 这 4 个子矩阵都可以通过补充若干全零元素组成的行和列得到 n 阶方阵。

若系统的节点数为 n , 其中 PV 节点数为 r , 则修正方程组的系数矩阵为 $n-r-1$ 阶的结构对称矩阵。

Jacobi 矩阵中各元素的计算表达式为^[27]:

$$\mathbf{H}_{ii} = 0 - \sum_{j \in i, j \neq i} \mathbf{H}_{ij} \quad (6)$$

$$\mathbf{N}_{ii} = \sum_{j \in i, j \neq i} \mathbf{N}_{ij} - 2\mathbf{V}_i^2 \mathbf{G}_{ii} \quad (7)$$

$$\mathbf{J}_{ii} = 0 - \sum_{j \in i, j \neq i} \mathbf{J}_{ij} \quad (8)$$

$$\mathbf{L}_{ii} = \sum_{j \in i, j \neq i} \mathbf{L}_{ij} + 2\mathbf{V}_i^2 \mathbf{B}_{ii} \quad (9)$$

式中: $\mathbf{H}_{ii} = \partial \Delta \mathbf{P}_i / \partial \boldsymbol{\theta}_i$, $\mathbf{H}_{ij} = \partial \Delta \mathbf{P}_i / \partial \boldsymbol{\theta}_j$, $\mathbf{N}_{ii} = \partial \Delta \mathbf{P}_i / \partial \mathbf{V}_i$, $\mathbf{N}_{ij} = \partial \Delta \mathbf{P}_i / \partial \mathbf{V}_j$, $\mathbf{J}_{ii} = \partial \Delta \mathbf{Q}_i / \partial \boldsymbol{\theta}_i$, $\mathbf{J}_{ij} = \partial \Delta \mathbf{Q}_i / \partial \boldsymbol{\theta}_j$, $\mathbf{L}_{ii} = \partial \Delta \mathbf{Q}_i / \partial \mathbf{V}_i$, $\mathbf{L}_{ij} = \partial \Delta \mathbf{Q}_i / \partial \mathbf{V}_j$; i 和 j 分别为系统的节点编号, 而不表示元素在矩阵中的行列号。

由式(6)至式(9)可知, \mathbf{H}_{ii} , \mathbf{N}_{ii} , \mathbf{J}_{ii} , \mathbf{L}_{ii} 分别在 \mathbf{H} , \mathbf{N} , \mathbf{J} , \mathbf{L} 这 4 个子矩阵中占优。因此, 利用这些元素, 可以构造预处理子。即

$$\mathbf{M}^{-1} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{bmatrix}^{-1} \quad (10)$$

取 \mathbf{H} , \mathbf{N} , \mathbf{J} , \mathbf{L} 中占主导的元素, 分别得到 \mathbf{A}_{ij} , \mathbf{B}_{ij} , \mathbf{C}_{ij} , \mathbf{D}_{ij} 并满足式(11)至式(14)。即

$$\mathbf{A}_{ij} = \begin{cases} \mathbf{H}_{ij} & i=j \\ \mathbf{0} & i \neq j \end{cases} \quad (11)$$

$$\mathbf{B}_{ij} = \begin{cases} \mathbf{N}_{ij} & i=j \\ \mathbf{0} & i \neq j \end{cases} \quad (12)$$

$$\mathbf{C}_{ij} = \begin{cases} \mathbf{J}_{ij} & i=j \\ \mathbf{0} & i \neq j \end{cases} \quad (13)$$

$$\mathbf{D}_{ij} = \begin{cases} \mathbf{L}_{ij} & i=j \\ \mathbf{0} & i \neq j \end{cases} \quad (14)$$

预处理子 \mathbf{M} 具有以下性质。

1) \mathbf{M} 为高度稀疏矩阵, 其子矩阵 \mathbf{A} , \mathbf{B} , \mathbf{C} , \mathbf{D} 每行的非零元和每列的非零元数量均不超过 1 个。同时, \mathbf{A} 和 \mathbf{D} 为 $(n-1) \times (n-1)$ 阶对角矩阵, \mathbf{B}^T 和 \mathbf{C}

为 $(n-1-r) \times (n-1)$ 阶矩阵,且非零元的位置相同。

2) 对 M 进行求逆无非零元注入,证明过程如下。

令

$$M^{-1} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} X_1 & X_2 \\ X_3 & X_4 \end{bmatrix} \quad (15)$$

则

$$MM^{-1} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_1 & X_2 \\ X_3 & X_4 \end{bmatrix} = \begin{bmatrix} AX_1 + BX_3 & AX_2 + BX_4 \\ CX_1 + DX_3 & CX_2 + DX_4 \end{bmatrix} = \begin{bmatrix} E_{n-1} & \mathbf{0} \\ \mathbf{0} & E_{n-1-r} \end{bmatrix} \quad (16)$$

当 $H, L, H - NL^{-1}J, L - JH^{-1}N$ 都可逆时,解得:

$$\begin{cases} X_1 = (A - BD^{-1}C)^{-1} \\ X_2 = -A^{-1}B(D - CA^{-1}B)^{-1} \\ X_3 = -D^{-1}C(A - BD^{-1}C)^{-1} \\ X_4 = (D - CA^{-1}B)^{-1} \end{cases} \quad (17)$$

以 M^{-1} 的子矩阵 X_1 为例。由于 D 为 $n-1$ 阶对角矩阵,而 B^T 和 C 为 $(n-1-r) \times (n-1)$ 阶矩阵,且具有相同的非零元素分布,故 $BD^{-1}C$ 为对角矩阵。则 $X_1 = (A - BD^{-1}C)^{-1}$ 与 A 同为 $(n-1) \times (n-1)$ 阶对角矩阵。同理, X_2, X_3, X_4 分别与子矩阵 B, C, D 具有相同的非零元位置。故 M^{-1} 不引入任何非零元注入。

在计算方面,由于 $A, D, A - BD^{-1}C, D - CA^{-1}B$ 均为对角矩阵, B 与 C 的非零元位置互为转置且每行至多只有一个非零元。可以采用特殊的方法快速求解:①稀疏矩阵求逆时,只需对每个对角元分别求逆;②稀疏矩阵相乘时,将其中一个矩阵转置,转置后矩阵的非零元与另一个矩阵中对应的非零元分别相乘即可。这一过程具有自然可并行性,可以利用 GPU 通用计算加速实现。

3 算例分析

3.1 Jacobi 矩阵谱分布的改善效果分析

为验证本文所提出的预处理方法能够有效改善系数矩阵的谱分布,选取 IEEE 标准系统和 PSS/E 软件自带的 BENCH 算例测试,对比不同预处理方法下在预处理前后的特征值分布。由表 1 可知,利用本文方法进行预处理后,各算例谱分布集中于 $0 \sim 2$ 之间,相比较于优化前有明显改善。Jacobi 预处理也能改善矩阵谱分布,但效果不如本方法。

由图 1 可知,IEEE 39, IEEE 118, IEEE 300 算

例和 BENCH 算例经预处理后,特征值分布变得集中。相比于 Jacobi 预处理,本文所提方法效果更为显著,尤其是虚轴方向上的特征值分布,处理后的最大特征值的虚部约为 0。

表 1 预处理前后 Jacobi 矩阵最大特征值
Table 1 Maximum eigenvalue of Jacobi matrix before and after pre-treatment

算例	优化前	Jacobi 预处理	本方法
IEEE 300	-4434.50 + 600.05i	1.85 + 2.23i	1.99
IEEE 118	-574.04	1.89 + 0.40i	1.96
IEEE 57	-118.60 + 76.80i	1.94 + 2.05i	1.98
IEEE 39	-1 022.90 + 76.70i	1.94 + 0.15i	1.96
IEEE 30	-102.95 + 28.50i	1.87 + 0.93i	1.90

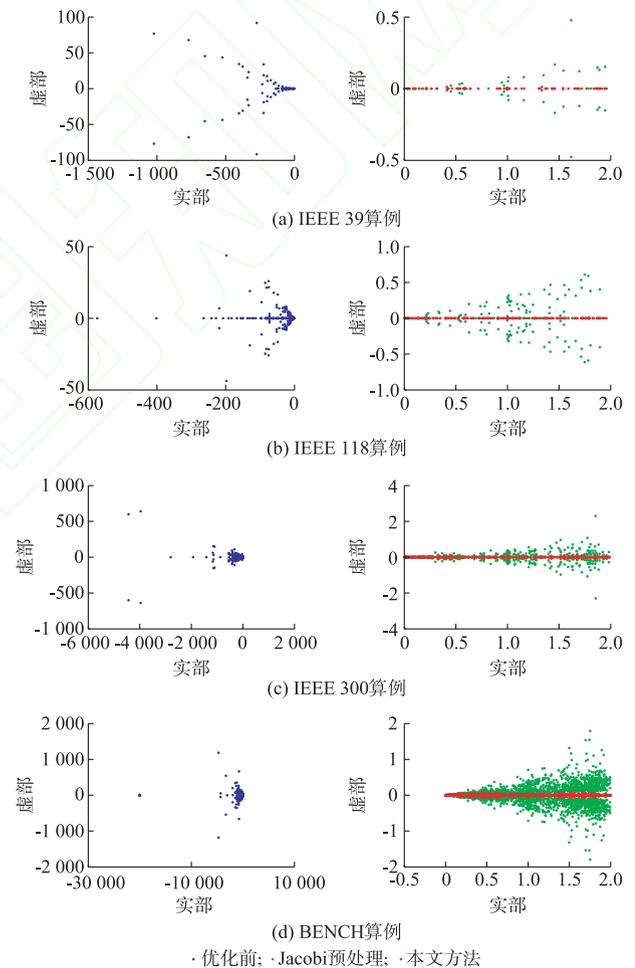


图 1 预处理前后 Jacobi 矩阵特征值分布
Fig.1 Distribution of eigenvalue of Jacobi matrix before and after preconditioned

3.2 M^{-1} 与 A^{-1} 的近似程度

为了比较采用上述方法求取预处理子时 A^{-1} 和 M^{-1} 的近似程度,选取 $\|E_{rr}\|_F/n^2$, $\|E_{rr}\|_1/n$, $\max(E_{rr})$ 作为参考标准。其中 E_{rr} 为误差矩阵,等于 A^{-1} 和 M^{-1} 差值的绝对值; $\|E_{rr}\|_F$ 为 E_{rr} 的

Frobenius 范数, $\|E_{rr}\|_1$ 为 E_{rr} 的 1-范数。 $\|E_{rr}\|_F/n^2$ 为 A^{-1} 与 M^{-1} 每个元素的平均偏差, $\|E_{rr}\|_1/n$ 为 A^{-1} 与 M^{-1} 每行平均值的最大偏差, $\max(E_{rr})$ 为 A^{-1} 与 M^{-1} 每个元素之间的最大差值。表 2 统计了用不同算例时, A^{-1} 和 M^{-1} 的近似程度。

表 2 预处理后 A^{-1} 和 M^{-1} 的近似程度
Table 2 Degree of approximation of A^{-1} and M^{-1} after pre-treatment

算例	$\frac{\ E_{rr}\ _F}{n^2}$	$\frac{\ E_{rr}\ _1}{n}$	$\max(E_{rr})$
BENCH	7.345×10^{-6}	0.040 91	0.879 9
IEEE 300	9.525×10^{-5}	0.056 50	1.906 0
IEEE 118	2.074×10^{-4}	0.046 53	0.270 9
IEEE 39	1.242×10^{-3}	0.108 00	0.250 9

从表 2 的结果可以看出,采用该方法进行预处理后, M^{-1} 和 A^{-1} 在数值上十分接近;从 $\|E_{rr}\|_F/n^2$ 的值可以看出,预处理后每个元素的平均偏差小于 0.001;从 $\|E_{rr}\|_1/n$ 可以看出,预处理后每行平均值最大偏差小于 0.1;从 $\max(E_{rr})$ 可以看出,预处理后对应元素之间差额小于 1.96。由此可见,该预处理方法拥有较好的效果。

3.3 预处理方法的有效性验证

为了进一步验证本文提出的预处理方法的有效性,选取多个算例测试预处理方法对迭代法迭代次数的影响。其中,IEEE 8971 算例由 30 个完全相同的 IEEE 300 系统拼接平衡节点得到。迭代法采用了稳定化的双共轭梯度(BiCGSTAB)法、双共轭梯度(BiCG)法、共轭梯度平方(CGS)法、最小残差法共准(QMR)这 4 种常用的 Krylov 子空间迭代法。表 3 对比了无预处理、Jacobi 预处理和本文预处理 3 种情形下求解线性方程组所需的平均迭代次数。其中,迭代法的计算精度取 0.001。

从表 3 可以看出,将本文的预处理方法应用于不同的迭代法进行线性方程组求解均可以大幅减少迭代次数,从而减少计算时间。当系统规模较大时,如 BENCH 迭代法,预处理效果更加明显,能够减少 90% 以上的迭代次数。相比较于常见的 Jacobi 预处理方法,具有显著的优势。

采用预处理时能够减少潮流计算中求解线性方程组时的迭代次数与迭代时间,但预处理过程本身需要占据时间。为了权衡预处理方法本身计算的耗时和提升迭代收敛性带来的计算效率提升之间的关系,同时考虑到 ILU0 预处理子也是商业软件常用的预处理方法,表 4 统计了本文方法、Jacobi 预处理方法和 ILU0 预处理方法应用于计算潮流时所需要的预处理时间及总时间。计算采用的 CPU 型号为

Intel i7-4710MQ,主频为 2.5 GHz,内存为 16 GB,线性方程组求解部分采用 CULA 计算包中的 BiCGSTAB 方法求解,采用的显卡为 NVIDIA GTX860M。其中 IEEE 8971 算例和 IEEE 1496 算例由多个 IEEE 300 算例系统拼接平衡节点得到。

表 3 预处理与无预处理时潮流计算中求解线性方程组时的迭代次数
Table 3 Iterations of solving linear equations when doing power flow calculation

算例	节点数	迭代法	无预处理	Jacobi 预处理	本文预处理
IEEE 8971	8 971	BiCGSTAB	860.1	147.5	81.5
		BiCG	664.0	186.2	126.3
		CGS	605.3	176.8	115.4
		QMR	619.7	177.4	122.1
BENCH	1 648	BiCGSTAB	1 703.5	525.0	244.0
		BiCG	2 130.0	610.0	347.0
		CGS	2 446.5	527.5	337.0
		QMR	2 134.0	633.5	348.0
IEEE 300	300	BiCGSTAB	678.5	152.5	81.5
		BiCG	616.0	187.0	126.0
		CGS	592.0	174.0	115.0
		QMR	619.0	176.0	122.0
IEEE 118	118	BiCGSTAB	71.5	47.5	36.0
		BiCG	110.0	73.0	54.0
		CGS	107.0	66.0	49.0
		QMR	113.0	70.0	53.0
IEEE 57	57	BiCGSTAB	74.5	60.5	37.0
		BiCG	125.0	78.0	56.0
		CGS	104.0	72.0	43.0
		QMR	124.0	77.0	53.0

由表 4 可以看出,采用本文方法的预处理方法可以大大减少线性方程组的求解时间,并增加求解的稳定性。在以上算例中, Jacobi 预处理子耗时均小于 1 ms,可以忽略不计; ILU0 预处理子耗时相对较大,占总耗时的 6.6%~16%;本文方法耗时介于二者之间,小于总耗时的 3%。从总耗时来看本方法远好于 Jacobi 预处理子,与 ILU0 预处理子耗时接近。

3.4 算法总体效率测试

本文的预处理方法能够有效地加速潮流计算,本文选取了 CULA 软件包进行对比测试。在采用 CULA 软件包进行潮流计算线性方程组求解时,很多求解器与预处理子的组合都不能有效地收敛,其中较为稳定的组合是 BiCGSTAB 求解器与 ILU0 预处理子或 Jacobi 预处理子,但是由于 ILU0 预处理子速度更快,于是 CULA 的数据采用了 BiCGSTAB+ILU0 进行计算。具体数据见附录 A 表 A1。本文提出的预处理方法应用于潮流计算,总体性能优于 CULA 软件包,能够满足在线计算的要

求。

参 考 文 献

表 4 不同预处理方法下进行潮流计算时求解线性方程组耗时

Table 4 Time of solving linear equations when doing power flow calculation with different pre-treatment methods

算例	节点数	预处理法	平均预处理耗时/ms	平均总耗时/ms
IEEE 8971	8 971	无预处理	0	4 140.0
		Jacobi	<1	394.2
		ILU0	12	106.6
		本文方法	3	117.0
BENCH	1 648	无预处理	0	1 719.0
		Jacobi	<1	412.0
		ILU0	12	180.0
		本文方法	3	130.0
IEEE 1496	1 496	无预处理	0	不收敛
		Jacobi	<1	296.0
		ILU0	13	90.0
		本文方法	<1	77.0
IEEE 300	300	无预处理	0	3 641.2
		Jacobi	<1	179.3
		ILU0	8	71.7
		本文方法	<1	97.5
IEEE 118	118	无预处理	0	95.0
		Jacobi	<1	60.0
		ILU0	9	54.0
		本文方法	<1	47.0
IEEE 57	57	无预处理	0	129.0
		Jacobi	<1	74.5
		ILU0	8	60.7
		本文方法	<1	31.0

4 结 语

本文提出了一种用于迭代法潮流计算的改进 Jacobi 预处理方法。经算例测试表明,所提的预处理方法构建预处理子速度快,同时能够有效地集中特征值分布、改善条件数、显著减少迭代次数,满足大规模电网在线潮流计算的需求,具有工程应用的潜力和价值。

由于本方法基于牛顿-拉夫逊法潮流计算中的 Jacobi 矩阵的特性来构建预处理子,尚无法直接应用于其他的电力系统计算,在今后的研究中,应当考虑在其他领域的应用。另一方面,改进 Jacobi 预处理法预处理后的系数矩阵仍有优化空间,可考虑多个预处理子结合的多步预处理,进一步加速 Krylov 子空间法的收敛速度。

附录见本刊网络版 (<http://www.aeps-info.com/aeps/ch/index.aspx>)。

- [1] 徐晓飞,曹祥玉,姚旭,等.一种基于 Doolittle LU 分解的线性方程组并行求解方法[J].电子与信息学报,2010,32(8):2019-2022.
XU Xiaofei, CAO Xiangyu, YAO Xu, et al. Parallel solving method of linear equations based on Doolittle LU decomposition [J]. Journal of Electronics & Information Technology, 2010, 32(8): 2019-2022.
- [2] HOU G, WANG L. A generalized iterative method and comparison results using projection techniques for solving linear systems[J]. Applied Mathematics and Computation, 2009, 215(2): 806-817.
- [3] AHAMED A C, MAGOULES F. Iterative methods for sparse linear systems on graphics processing unit [C]// 2012 IEEE 14th International Conference on High Performance Computing and Communications & 2012 IEEE 9th International Conference on Embedded Software and Systems, June 25-27, Liverpool, United Kingdom, 2012: 836-842.
- [4] LEON F D, SERNLLEN A. Iterative solvers in the Newton power flow problem: preconditioners, inexact solutions and partial Jacobi updates [J]. Generation, Transmission and Distribution, IEEE Proceedings, 2002, 149(4): 479-484.
- [5] KULKARNI A Y, PAI M A, SAUER P W. Iterative solver techniques in fast dynamic calculations of power systems[J]. International Journal of Electrical Power & Energy Systems, 2001, 23(3): 237-244.
- [6] 蔡大用,陈玉荣.用不完全 LU 分解预处理的不精确潮流计算方法[J].电力系统自动化,2002,26(8):11-14.
CAI Dayong, CHEN Yurong. Solving power flow equations with inexact newton methods preconditioned by incomplete LU factorization with partially fill-in [J]. Automation of Electric Power Systems, 2002, 26(8): 11-14.
- [7] 王海峰,陈庆奎.图形处理器通用计算关键技术研究综述[J].计算机学报,2013,36(4):757-772.
WANG Haifeng, CHEN Qingkui. General purpose computing of graphics processing unit: a survey [J]. Chinese Journal of Computers, 2013, 36(4): 757-772.
- [8] 郭春辉.基于 GPU 的电力系统并行计算的研究[D].济南:山东大学,2013.
- [9] 张逸飞,严正,赵文恺,等.基于 GPU 的分块约化算法在小干扰稳定分析中的应用[J].电力系统自动化,2015,39(22):90-97. DOI:10.7500/AEPS20150126012.
ZHANG Yifei, YAN Zheng, ZHAO Wenkai, et al. Application of block reduction algorithm in power system small-signal stability analysis [J]. Automation of Electric Power Systems, 2015, 39(22): 90-97. DOI: 10.7500/AEPS20150126012.
- [10] 周挺辉,赵文恺,严正,等.基于图形处理器的电力系统稀疏线性方程组求解方法[J].电力系统自动化,2015,39(2):74-80. DOI:10.7500/AEPS20131119005.
ZHOU Tinghui, ZHAO Wenkai, YAN Zheng, et al. A method for solving sparse linear equations power systems based on GPU [J]. Automation of Electric Power Systems, 2015, 39(2): 74-80. DOI: 10.7500/AEPS20131119005.
- [11] ZHANG J. Preconditioned Krylov subspace methods for solving nonsymmetric matrices from CFD applications [J]. Computer Methods in Applied Mechanics and Engineering, 1998, 189(3): 825-840.
- [12] SAAD Y, VORST H. Iterative solution of linear systems in

- the 20th century[J]. Journal of Computational and Applied Mathematics, 2000, 123(1): 1-33.
- [13] 陈颖,沈沉,梅生伟,等.基于改进 Jacobi-Free Newton-GMRES (m)的电力系统分布式潮流计算[J].电力系统自动化,2006, 30(9):5-8.
CHEN Ying, SHEN Cheng, MEI Shengwei, et al. Distributed power flow calculation based on an improved Jacobian-free Newton-GMRES (m) method [J]. Automation of Electric Power Systems, 2006, 30(9): 5-8.
- [14] AXELSSON O, LINDSKOG G. On the eigenvalue distribution of a class of preconditioning methods [J]. Numerische Mathematik, 1986, 48(5): 479-498.
- [15] JIA Z. The convergence of Krylov subspace methods for large unsymmetric linear systems [J]. Acta Mathematica Sinica, New Series, 1998, 14(4): 507-518.
- [16] ZHANG J. A grid-based multilevel incomplete LU factorization preconditioning technique for general sparse matrices [J]. Applied Mathematics and Computation, 2001, 43(4): 483-500.
- [17] SAAD Y, ZHANG J. Enhanced multi-level block ILU preconditioning strategies for general sparse linear systems[J]. Journal of Computational and Applied Mathematics, 2001, 130(1): 99-118.
- [18] WILLE S O, STAFF O, LOULA A. Block and full matrix ILU preconditioners for parallel finite element solvers [J]. Computer Methods in Applied Mechanics and Engineering, 2002, 191(13): 1381-1394.
- [19] SHEN C, ZHANG J. Parallel two level block ILU preconditioning techniques for solving large sparse linear systems[J]. Parallel Computing, 2002, 28(10):1451-1475.
- [20] LEE J, ZHANG J, LU C. Incomplete LU preconditioning for large scale dense complex linear systems from electromagnetic wave scattering problems [J]. Journal of Computational Physics, 2003, 185(1): 158-175.
- [21] 柳建新,蒋鹏飞,童孝忠,等.不完全 LU 分解预处理的 BICGSTAB 算法在大地电磁二维正演模拟中的应用[J].中南大学学报(自然科学版),2009,40(2):484-491.
LIU Jianxin, JIANG Pengfei, TONG Xiaozhong, et al. Application of BICGSTAB algorithm with incomplete LU decomposition preconditioning to two-dimensional magnetotelluric forward modeling[J]. Journal of Central South University (Science and Technology), 2009, 40(2): 484-491.
- [22] KAEBI A, KERAYECHAN A, TOUTOUNIAN F. Approximate inverse preconditioner by computing approximate solution of Sylvester equation [J]. Applied Mathematics and Computation, 2005, 170(2): 1067-1076.
- [23] ZHANG J. A sparse approximate inverse preconditioner for parallel preconditioning of general sparse matrices[J]. Applied Mathematics and Computation, 2002, 130(1): 63-85.
- [24] NOTAY Y. Polynomial acceleration of iterative schemes associated with subproper splittings [J]. Journal of Computational and Applied Mathematics, 1988, 24(1): 153-167.
- [25] CERDAN J, MAR N, MART N A. Polynomial preconditioners based on factorized sparse approximate inverses [J]. Applied Mathematics and Computation, 2002, 133(1): 171-186.
- [26] ZHANG J, ZHANG L. Efficient CUDA polynomial preconditioned conjugate gradient solver for finite element computation of elasticity problems[J]. Mathematical Problems in Engineering, 2013(6): 1-12.
- [27] 陈德扬,李亚楼,江涵,等.基于道路树分层的大电网潮流并行算法及其 GPU 优化实现[J].电力系统自动化,2014,38(22): 63-69. DOI:10.7500/AEPS20131014009.
CHENG Deyang, LI Yalou, JIANG Han, et al. A parallel power flow algorithm for large-scale grid based on stratified path trees and its implementation on GPU[J]. Automation of Electric Power Systems, 2014, 38(22): 63-69. DOI: 10.7500/AEPS20131014009.

唐 灿(1992—),男,硕士,主要研究方向:电力系统并行计算。E-mail: tangcan@live.cn

董树锋(1982—),男,通信作者,博士,副教授,主要研究方向:状态估计和配电网建模和分析。E-mail: dongshufeng@zju.edu.cn

任雪桂(1984—),女,工程师,主要研究方向:配电网规划、电网智能化。E-mail: renxuegui@126.com

(编辑 鲁尔姣)

Improved Jacobi Pre-processing Method for Solving Iterative Power Flow Calculation

TANG Can¹, DONG Shufeng¹, REN Xuegui², YIN Lu², JU Li²

(1. College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China;

2. Beijing Electric Power Economic Research Institute, Beijing 100055, China)

Abstract: In order to improve the speed of power flow calculation and meet the requirement of real-time calculation, the iterative methods for linear equations are applied to the power flow calculation. However, when the spectral distribution of the coefficient matrix is relatively scattered, the speed of iterative method is slow or even not convergent. The coefficient matrix is needed to be pre-processed to solve the problems mentioned above. Firstly, the characteristic of Jacobi matrix in the power flow calculation is analyzed and segmentally processed based on the PV and PQ nodes. The bigger numbers are found and act as preconditioners. Then, the inverse matrix of the preconditioners is multiplied by the coefficient matrix A and the constant item b, and the original linear system is transformed into a new equivalent linear system which can be easily solved. The speed of solving linear equations in power flow calculation is highly improved. The simulation result shows that the algorithm can solve the power flow equations efficiently.

Key words: power flow calculation; sparse linear equations; pre-treatment; sparse matrix; general calculation of graphics processor