DOI：10.13334/j.0258-8013.pcsee.170921　文章编号：0258-8013 (2017) S-0055-08　中图分类号：TM 744

基于不完全 LU 分解预处理迭代法的电力系统潮流算法

唐坤杰，董树锋，宋永华

(浙江大学电气工程学院，浙江省 杭州市 310027)

Power Flow Algorithm Based on an Iterative Method With Incomplete LU Decomposition Preconditioning

TANG Kunjie, DONG Shufeng, SONG Yonghua

(College of Electrical Engineering, Zhejiang University, Hangzhou 310027, Zhejiang Province, China)

ABSTRACT: With the increment of scale of power system, the requirement of speed and real-time of power flow calculation becomes stricter. In order to meet the demand of large scale power flow calculation, based on the theory of Krylov subspace, an iterated algorithm with incomplete LU decomposition preconditioning was proposed to solve large scale sparse linear equations in the process of power flow calculation. This algorithm used CPU-GPU heterogeneous computing architecture to divide itself into two parts processed by CPU and GPU respectively based on their different features, in order to fulfill fast solution. GPU was used for parallel processing of the most intensive calculation solving of linear equations, and CPU was used for other steps of this algorithm. The numerical results show that, the algorithm proposed is of high efficiency, stable convergence performance, fast convergence speed. Compared with traditional power flow algorithm based on LU decomposition, it has obvious advantages in large scale system, which can meet the needs of online flow calculation of large scale power grid and has the value of engineering application.

KEY WORDS: Krylov subspace; incomplete LU decomposition preconditioning; large scale sparse linear equations; power flow calculation; CPU-GPU heterogeneous computing architecture

摘要：随着电力系统规模日益增大，对潮流计算速度与实时性的要求相应提高。为了适应大规模电力系统潮流计算需求，根据 Krylov 子空间思想，提出了一种基于迭代法求解线性方程组的潮流算法，该算法利用不完全 LU 分解作为预处理，并采用 CPU-GPU 异构运算架构，根据 CPU 和 GPU 的不同特点，将潮流算法分为 CPU 处理部分和 GPU 处理部分，其中 GPU 用于并行处理计算量最为密集的线性方程组求解步骤，CPU 用于处理潮流算法的其他步骤，实现快速求解。算例表明，所提算法收敛性能稳定、收敛速度快、算法效率高，在系统规模较大时，与传统基于 LU 分解的潮流算法相比具有明显优势，能够满足大规模电网在线潮流计算的需求，具有工程应用价值。

关键词：Krylov 子空间；不完全 LU 分解；大规模稀疏线性方程组；潮流计算；CPU-GPU 异构运算架构

0 引言

传统的电力系统潮流计算以牛顿–拉夫逊法为主，在利用这一算法进行潮流计算时，通过逐次线性化，反复求解非线性方程组对应的线性方程组的过程，占据了潮流计算的绝大多数时间。随着省地一体化的进程不断加深，省调管理的输电网模型与地调管辖范围内的配电网模型实现拼接处理计算，系统矩阵阶数增大，雅可比矩阵条件数变差，对潮流计算速度和精度的需求不断提升。因此，研究针对大规模电力系统潮流方程的高效求解方法十分重要。

求解大规模稀疏线性方程组的方法主要分为直接法[1-3]和迭代法[4-6]两种。其中，直接法主要基于矩阵分解技术，例如常用的 LU 分解法，在小规模的稀疏线性方程组求解中占有一定优势，但是该方法难以通过并行处理提高计算效率。近年来，虽然有一些研究成果针对高斯消元法进行了并行化加速处理，但这些并行处理的方法主要针对带状线性方程组[7-8]，而不具有普遍性，因此使用时有较大的局限性。

相较于直接法，迭代法适合并行计算，在求解超大规模线性方程组时具有明显的优势。但是，迭代法也有不足。一方面，相对于直接法具有不稳定

性，在系数矩阵条件数较大的时候，迭代法容易出现收敛速度慢甚至不收敛等问题。另一方面，迭代法的收敛速度对精度要求比较敏感，计算精度要求较高时，迭代次数会明显增加。在应用迭代法求解潮流时需克服上述困难，其手段是采用合适的预处理技术，即通过对系数矩阵进行预处理，降低条件数，从而有效改善迭代法的收敛速度。近年来，随着电力系统规模的不断扩大，迭代法被初步应用到电力系统计算的一些领域，主要包括潮流计算[9-10]、最优潮流[11]、暂态仿真及稳定分析[12]等，但研究内容主要集中在传统的迭代方法和预处理方法，对于新的迭代法理论和预处理技术成果还没有充分应用。

另一方面，随着 GPU 的飞速发展，GPU 通用计算被应用到诸多学科的科学研究和工程应用之中。由于 CPU 与 GPU 在工作原理及特点上有很大的不同，在 CPU-GPU 异构协同的计算体系中，可以各取所长，使得整个架构具有强大的并行计算能力，性价比和性能能耗比高[13]。GPU 通用计算特别是 CPU-GPU 异构运算框架为迭代法潮流提供了实现平台，一些学者利用 GPU 加速在电力系统潮流计算方面取得了初步成果[14-16]。上述研究利用 GPU 实现了传统迭代方法的并行化处理如高斯–塞德尔法的并行等，对于新的迭代法理论和预处理技术在电力系统潮流计算中的并行化还未深入研究。

本文根据 Krylov 子空间理论，提出基于不完全 LU 分解预处理迭代法的电力系统潮流算法，这一算法和传统迭代法、预处理方法相比在计算效率、收敛性能、内存占用方面均具有显著优势。通过算例分析算法效率、收敛性能、内存占用，验证了本文所提出的潮流算法及其预处理方法在求解大规模电力系统潮流中的有效性。

1 牛顿–拉夫逊法潮流计算

1.1 数学模型

在潮流计算中，以极坐标为例，节点功率方程[17]可表示为：

$$\begin{cases} P_i - U_i \sum_{j=1}^{n} U_j (G_{ij}\cos\delta_{ij} + B_{ij}\sin\delta_{ij}) = 0 \\ Q_i - U_i \sum_{j=1}^{n} U_j (G_{ij}\sin\delta_{ij} - B_{ij}\cos\delta_{ij}) = 0 \end{cases} \tag{1}$$

其中：P_i 表示各节点的有功注入量；Q_i 表示各节点的无功注入量；U_i 和 U_j 表示节点 i 和 j 的电压幅值；δ_{ij} 表示节点 i 和 j 的相角差。

牛顿—拉夫逊法是常用的潮流计算方法。该方法通过逐次线性化，把非线性方程组的求解过程变成了反复求解其相对应的线性方程组的过程。

极坐标下，对于节点功率方程，在第 k 次迭代时，令

$$\begin{cases} \Delta P_i^k = P_i - U_i^k \sum_{j=0}^{n} U_j^k (G_{ij}\cos\delta_{ij}^k + B_{ij}\sin\delta_{ij}^k) \\ \Delta Q_i^k = Q_i - U_i^k \sum_{j=0}^{n} U_j^k (G_{ij}\sin\delta_{ij}^k - B_{ij}\cos\delta_{ij}^k) \end{cases} \tag{2}$$

则其修正方程组为：

$$\begin{cases} \Delta P_i = \sum_{j=1}^{n} \dfrac{\partial \Delta P_i}{\partial \delta_j}\Delta\delta_j + \sum_{j=1}^{n} \dfrac{\partial \Delta P_i}{\partial U_j}\Delta U_j \\ \Delta Q_i = \sum_{j=1}^{n} \dfrac{\partial \Delta Q_i}{\partial \delta_j}\Delta\delta_j + \sum_{j=1}^{n} \dfrac{\partial \Delta Q_i}{\partial U_j}\Delta U_j \end{cases} \tag{3}$$

对于所有节点：

$$\begin{bmatrix} \Delta \boldsymbol{P} \\ \Delta \boldsymbol{Q} \end{bmatrix} = \begin{bmatrix} \boldsymbol{H} & \boldsymbol{N} \\ \boldsymbol{F} & \boldsymbol{L} \end{bmatrix} \begin{bmatrix} \Delta \boldsymbol{\delta} \\ \Delta \boldsymbol{U}/\boldsymbol{U} \end{bmatrix} \tag{4}$$

其中雅可比矩阵的非对角元素满足：

$$\begin{cases} H_{ij} = -U_i U_j (G_{ij}\sin\delta_{ij} - B_{ij}\cos\delta_{ij}) \\ N_{ij} = -U_i U_j (G_{ij}\cos\delta_{ij} + B_{ij}\cos\delta_{ij}) \\ F_{ij} = U_i U_j (G_{ij}\cos\delta_{ij} + B_{ij}\cos\delta_{ij}) \\ L_{ij} = -U_i U_j (G_{ij}\sin\delta_{ij} - B_{ij}\cos\delta_{ij}) \end{cases} \tag{5}$$

对角元素满足

$$\begin{cases} H_{ii} = Q_i + B_{ii}U_i^2 \\ N_{ii} = -P_i - G_{ii}U_i^2 \\ F_{ii} = -P_i + G_{ii}U_i^2 \\ L_{ii} = -Q_i + B_{ii}U_i^2 \end{cases} \tag{6}$$

通过修正方程组求解得到修正量$\Delta\boldsymbol{\delta}$和$\Delta\boldsymbol{U}$ 后，即可得到新的解，反复迭代直至收敛即可。

1.2 修正方程组特征分析

系统规模较小时，雅可比矩阵规模也较小，潮流计算一般采用传统方法，即采用 LU 分解的牛顿–拉夫逊算法，例如可以使用 SuperLU 库实现 LU 分解步骤。SuperLU 库主要用于矩阵运算，其中求解线性方程组的方法属于直接法，经高度优化，效率较高[18]。

系统规模较大时，结合算例可以发现，在线性方程组中，雅可比矩阵一般是大型稀疏非对称矩阵，其条件数远大于 1。系统规模庞大使得直接法难以满足计算需求，采用迭代法是解决大规模线性方程组求解问题的重要途径之一。另一方面，由于

矩阵的条件数差，选择稳定的迭代方法以及采取相应的预处理技术也是必需的。

2 Krylov 子空间与预处理迭代

Krylov 子空间迭代法是求解大型线性方程组的一类重要迭代方法[19-20]。求解线性方程组：

$$\boldsymbol{Ax}=\boldsymbol{b}, \boldsymbol{A}\in\mathbb{R}^{n\times n}, \boldsymbol{x},\boldsymbol{b}\in\mathbb{R}^{n} \tag{7}$$

的一般投影方法是从 m 维仿射子空间 $\boldsymbol{x}_0+\boldsymbol{K}_m$(称为搜索空间)中寻求一个近似解 $\boldsymbol{x}_m$，它使用了 Petrov-Galerkin 条件：

$$\boldsymbol{b}-\boldsymbol{Ax}_m\perp \boldsymbol{L}_m \tag{8}$$

其中 $\boldsymbol{L}_m$ 是另一个 m 维子空间(称为约束空间)。这里，$\boldsymbol{x}_0$ 表示解的迭代初值(任意一个初始猜测)。在 Krylov 子空间方法中，$\boldsymbol{K}_m$ 就是 Krylov 子空间：

$$\boldsymbol{K}_m(\boldsymbol{A},\boldsymbol{v})=span\{\boldsymbol{v},\boldsymbol{Av},\boldsymbol{A}^2\boldsymbol{v},\cdots,\boldsymbol{A}^{(m-1)}\boldsymbol{v}\} \tag{9}$$

其中 $\boldsymbol{v}$ 可选为初始残差 $\boldsymbol{r}_0=\boldsymbol{b}-\boldsymbol{Ax}_0$。同时，约束空间 $\boldsymbol{L}_m$ 的选择将对迭代技术具有重要影响，主要分为如下几类[20]：

1）取 $\boldsymbol{L}_m=\boldsymbol{K}_m(\boldsymbol{A}, \boldsymbol{r}_0)$，称为正交投影方法，可极小化误差的能量范数，如共轭梯度(CG)法，适用于处理对称正定矩阵。

2）取 $\boldsymbol{L}_m=\boldsymbol{AK}_m(\boldsymbol{A}, \boldsymbol{r}_0)$，称为正交化方法，可极小化残差的 2 范数，故也被称为极小残差方法，如广义极小残差(GMRES)方法。

3）取 $\boldsymbol{L}_m=\boldsymbol{K}_m(\boldsymbol{A}^{\mathrm{T}}, \boldsymbol{r}_0)$，称为双正交化方法，如双正交共轭梯度(BICG)方法和稳定的双正交共轭梯度(BICGSTAB)方法等，适用于处理非对称矩阵[21-22]。

4）混合型方法，基于某两种基本方法的混合。

迭代方法的有效性和健壮性都可以通过使用预处理来得到改善，通过预处理变换，将原线性方程组转换成与其同解而更易于迭代求解的线性方程组。设 $\boldsymbol{M}$ 是一个预处理子，它是一个矩阵，通常分为三类：

1）左预处理：对 $\boldsymbol{M}^{-1}\boldsymbol{Ax} = \boldsymbol{M}^{-1}\boldsymbol{b}$ 应用迭代法；

2）右预处理：对 $\boldsymbol{AM}^{-1}\boldsymbol{u} = \boldsymbol{b}$ 应用迭代法，$\boldsymbol{x} = \boldsymbol{M}^{-1}\boldsymbol{u}$；

3）分裂预处理：$\boldsymbol{M}$ 可表示为 $\boldsymbol{M} = \boldsymbol{M}_1\boldsymbol{M}_2$，对 $\boldsymbol{M}_1^{-1}\boldsymbol{AM}_2^{-1}\boldsymbol{u}=\boldsymbol{M}_1^{-1}\boldsymbol{b}$ 应用迭代法。

3 基于不完全 LU 分解预处理迭代法的电力系统潮流算法

3.1 CPU-GPU 异构运算架构及 CUDA 框架

在 CPU-GPU 异构运算架构中，CPU 与 GPU 分别有各自的内存体系，CPU 无法与显存直接通讯，GPU 也无法与主机内存直接通讯，CPU 和 GPU 之间的数据传输通过 PCI-Express 总线进行通信。一个 GPU 通用计算程序需要包括三部分[23]：CPU 计算程序、GPU 计算程序和 CPU-GPU 通信程序。

CUDA(Compute Unified Devices Architecture) 框架是 NVIDIA 公司推出的一种新的编程模型和指令集架构，用于实现 GPU 通用计算[24]。CUDA 框架对原先的 C 语言进行扩展，基于这一框架既可以定义普通的 C 语言函数，也可以定义“特别”的 C 语言函数称为“内核”，即在调用时，并行执行 n 次 n 个不同的并行线程，而不是像普通的 C 语言函数只执行一次[25]。因此，CUDA 框架为 CPU-GPU 异构运算架构的实现提供了条件。

传统的基于 LU 分解的潮流算法建立在矩阵分解、求逆等计算上，难以达到高度并行化。本文算法基于迭代法，矩阵运算和向量运算是主要计算形式，适合并行化处理以提高计算效率。因此，本算法基于 CUDA 框架和 CPU-GPU 异构运算架构进行实现。

3.2 算法步骤

由于雅可比矩阵是非对称矩阵，且条件数远大于 1，故基于不完全 LU 分解预处理技术和 Krylov 子空间迭代法中的双正交化方法，设计一种新的电力系统潮流算法。这一算法基于 CUDA 框架，采用 CPU-GPU 异构运算架构，分为 CPU 处理部分和 GPU 处理部分，CPU 处理迭代初值的设定、节点导纳矩阵的形成、雅可比矩阵的形成、修正方程组的形成、迭代值的修正、收敛性判断等步骤，GPU 处理修正方程组的求解，以达到快速求解的目的。本文提出的潮流算法步骤具体如下：

1）根据原始数据形成节点导纳矩阵，从而得到式(2)中的 G_{ij}、B_{ij} 的值。

2）设置牛顿–拉夫逊法的迭代变量$\boldsymbol{\delta}$和 $\boldsymbol{U}$ 的初值，设置迭代次数 k=0。

3）根据式(2)、(3)、(5)、(6)，由当前迭代变量的值可得到修正方程组(4)。令：

$$\boldsymbol{b}=\begin{bmatrix}\Delta\boldsymbol{P}\\ \Delta\boldsymbol{Q}\end{bmatrix} \tag{10}$$

$$\boldsymbol{J}=\begin{bmatrix}\boldsymbol{H} & \boldsymbol{N}\\ \boldsymbol{F} & \boldsymbol{L}\end{bmatrix} \tag{11}$$

$$\boldsymbol{X}=\begin{bmatrix}\Delta\boldsymbol{\delta}\\ \Delta\boldsymbol{U}/\boldsymbol{U}\end{bmatrix} \tag{12}$$

4）在 GPU 上解线性方程组：

$$\boldsymbol{JX} = \boldsymbol{b} \tag{13}$$

考虑到 $\boldsymbol{J}$ 矩阵的非对称和高度稀疏性，可采用双正交化方法选择迭代法的约束空间。基于稳定的双正交共轭梯度(BICGSTAB)方法的思想，线性方程组求解的具体步骤如下：

①首先利用不完全 LU 分解中的 ILU(0)分解求取预处理子 $\boldsymbol{M}$。具体地，应用这一方法，将稀疏矩阵 $\boldsymbol{J}$ 分解成一个稀疏下三角矩阵 $\boldsymbol{L}$ 和一个稀疏上三角矩阵 $\boldsymbol{U}$ 的乘积，使得残差矩阵 $\boldsymbol{R} = \boldsymbol{LU} - \boldsymbol{J}$ 满足 ILU(0)分解条件。

②取 $\boldsymbol{X}$ 的初始猜测 $\boldsymbol{x}_0$ 和允许误差 ε，计算 $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{Jx}_0$，令 $\boldsymbol{r}_0^* = \boldsymbol{r}_0$、$j = 1$。

③计算 $\rho_{j-1} = (\boldsymbol{r}_{j-1}, \boldsymbol{r}_0^*)$，如果 $\rho_{j-1} = 0$，方法失败，否则进入步骤 d。

④如果，令 $\boldsymbol{p}_j = \boldsymbol{r}_{j-1}$，否则令 $\beta_{j-1} = (\rho_{j-1}/\rho_{j-2})(\alpha_{j-1}/\omega_{j-1})$，$\boldsymbol{p}_j = \boldsymbol{r}_{j-1} + \beta_{j-1}(\boldsymbol{p}_{j-1} - \omega_{j-1}\boldsymbol{v}_{j-1})$。

⑤由 $\boldsymbol{M}\hat{\boldsymbol{p}} = \boldsymbol{p}_j$ 求解 $\hat{\boldsymbol{p}}$，计算 $\boldsymbol{v}_j = \boldsymbol{J}\hat{\boldsymbol{p}}$，$\alpha_j = \rho_{j-1}/(\boldsymbol{v}_j, \boldsymbol{r}_0^*)$，$\boldsymbol{s} = \boldsymbol{r}_{j-1} - \alpha_j \boldsymbol{v}_j$。

⑥如果 $\|\boldsymbol{s}\| \le \varepsilon$，令 $\boldsymbol{x}_j = \boldsymbol{x}_{j-1} + \alpha_j \hat{\boldsymbol{p}}$，退出迭代过程。

⑦由 $\boldsymbol{M}\hat{\boldsymbol{s}} = \boldsymbol{s}_j$ 求解 $\hat{\boldsymbol{s}}$，令 $\boldsymbol{t} = \boldsymbol{J}\hat{\boldsymbol{s}}$，$\omega_j = (\boldsymbol{s}, \boldsymbol{t})/(\boldsymbol{t}, \boldsymbol{t})$，$\boldsymbol{x}_j = \boldsymbol{x}_{j-1} + \alpha_j \hat{\boldsymbol{p}} + \omega_j \hat{\boldsymbol{s}}$。

⑧如果 $\boldsymbol{x}_j$ 满足精度要求则退出迭代过程，否则令 $\boldsymbol{r}_j = \boldsymbol{s} - \omega_j \boldsymbol{t}$，将 j 值加 1，转至步骤③。

5）在 CPU 上根据步骤 4）求得的 $\boldsymbol{X}$ 的值，即修正量 $\Delta\boldsymbol{\delta}$ 和 $\Delta\boldsymbol{U}$，进行牛顿–拉夫逊法的收敛性判断，如满足精度要求，则退出潮流算法，认为算法收敛；否则，修正迭代变量 $\boldsymbol{\delta}$ 和 $\boldsymbol{U}$。

6）在 CPU 上令迭代次数 $k = k + 1$，若已达到预设的最大迭代次数，则退出潮流算法，认为算法不收敛；否则，转至步骤 3）。

3.3 算法特点

1）算法采用 ILU(0)分解预处理技术，该方法对原系数矩阵做无任何额外非零元填充的 ILU 的分解，预处理速度较快[22]。大规模的电力系统的系数矩阵一般具有稀疏性，ILU(0)分解不注入非零元，能够有效保持系数矩阵的稀疏性，因此这一预处理方法适用于电力系统系数矩阵。而不完全 LU 分解中的其他两类方法 ILU(1)和 ILU(2)，虽然相较于 ILU(0)能够一定程度上提高预处理效果、减少方程求解的迭代次数，但是这两种预处理方法速度较慢，且均需要注入非零元，破坏电力系统系数矩阵的稀疏性，在稀疏矩阵的乘法、内积等运算中显著加大计算量，降低潮流计算的效率。

2）算法基于 BICGSTAB 迭代法实现。电力系统系数矩阵具有非对称的特点，以 GMRES 法为代表的正交化方法和以 BICGSTAB 法为代表的双正交化方法均适用于此类矩阵对应的线性方程组的求解。然而，GMRES 法相比 BICGSTAB 方法及其他双正交化具有明显的缺点，一方面，在 GMRES 法中每多一次迭代，所需计算次数和存储空间都呈线性增长，另一方面 GMRES 法不具有短递归性质。而 BICGSTAB 的存储空间稳定，不随迭代次数变化，且具有短递归性质，因此更适用于大规模的电力系统潮流计算。

3）算法采用 CPU-GPU 异构运算架构，基于 CUDA 框架实现。利用 CPU-GPU 通信程序实现 CPU 和 GPU 相互间的数据移植，CPU 向 GPU 移植 $\boldsymbol{JX} = \boldsymbol{b}$ 中的 $\boldsymbol{J}$、$\boldsymbol{b}$ 矩阵数据，GPU 向 CPU 移植计算得到的 $\boldsymbol{X}$ 矩阵。GPU 计算程序并行处理计算量最密集的步骤，即修正方程组的求解，从而在迭代法的基础上进一步加速算法。

4 算例分析

为了分析算法的优越性，利用多组测试算例进行测试。其中 CASE57 至 CASE300 是 IEEE 标准算例，BENCH 是 PSS\E 软件自带的算例，CASE8971 算例是将 30 个 CASE300 算例平衡节点合并得到的。编译程序所使用的编译器为 Microsoft Visual Studio 2013 Update 3 与 NVIDIA Nsight Visual Studio Edition，运行在 64bit 的 windows10 操作系统上。测试使用的 CPU 型号为 Intel Core i7-5820K，运行主频 3.30GHz；内存为 16GB；GPU 为 NVIDIA GeForce GTX980，支持 CUDA7.0 标准。

4.1 与传统方法计算用时的对比

为了验证该方法在提高求解潮流计算效率及其在大规模稀疏线性方程组求解中的作用，利用基于求解库 SuperLU 的传统方法做对比分析。测试所用算例由多个 CASE300 拼接平衡节点得到，其中牛顿–拉夫逊法的精度要求取 0.01，线性方程组求解的迭代法精度要求取 10^{-6}。

从表 1 和图 1 可以看出，在节点规模小于 3000 节点时，采用基于 SuperLU 库的传统方法运算速度上具有优势；当节点规模较大(>3000 节点)时，本文提出的方法计算速度更快，而且系统规模越大，本文方法运算速度上优势越明显。上述算例表明本

文方法在系统规模较大时能够大幅提高潮流计算速度，较传统方法具有明显优势。

表 1　本文算法与传统算法的计算用时比较

Tab. 1　Comparison of computation time between traditional algorithm and algorithm in this paper

拼接个数	节点数	支路数	本文算法用时/ms	传统法用时/ms
1	300	411	656	234
5	1496	2055	734	486
10	2991	4110	859	823
50	14951	20550	1037	3653
500	149501	205500	2594	36247
5000	1495001	2055000	20230	367303

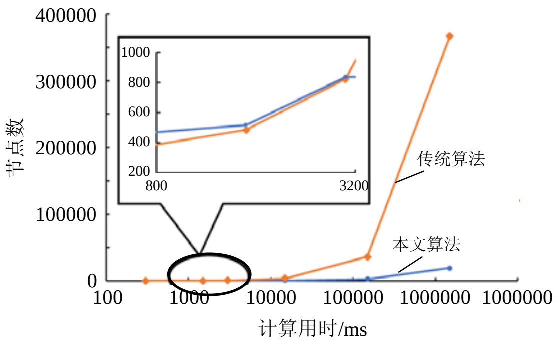

图 1　本文算法与传统算法的计算用时比较

Fig. 1　Comparison of computation time between traditional algorithm and algorithm in this paper

4.2　算法中修正方程组求解部分的收敛性分析

为了对该算法中修正方程组求解部分进行收敛性分析，利用 CASE300 作算例，测试在不同的精度要求下每一次线性方程组求解所需的平均迭代次数，以及牛顿拉夫逊法第一次迭代过程中求解线性方程组所需的迭代次数。

从图 2 的平均迭代次数曲线可以看出，本文提出的算法是一种有效的求解大型稀疏线性方程组的方法，修正方程组求解所需的迭代次数与迭代精度要求基本呈现线性关系，在 10^{-8} 的精度要求下迭代次数能够控制在 50 次以内。同时，从两条曲线

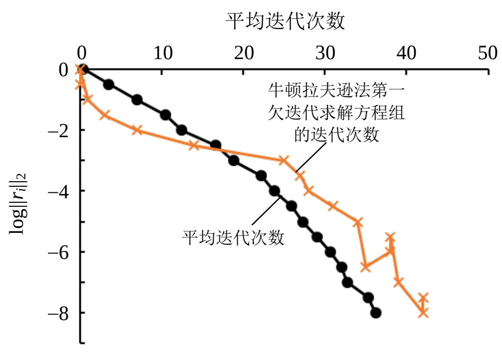

图 2　算法收敛性分析

Fig. 2　Convergence analysis of the algorithm

可以看出，收敛曲线是比较光滑的，说明修正方程组的求解方法有利于进行收敛性判断。

本文算法中修正方程组的求解方法与 BICG、CGS、QMR 等方法同样是基于 Krylov 子空间理论的双正交化迭代方法[26]，且同样适用于大型稀疏非对称矩阵所对应的线性方程组的求解。在不进行预处理的条件下，利用多个算例，可以比较不同迭代法的稳定性。下表统计了不同算例在无预处理条件下，利用不同迭代方法进行潮流计算时求解线性方程组所需的平均迭代次数，其中计算精度为 0.001，如表 2 所示。

表 2　各类迭代法平均迭代次数比较

Tab. 2　Comparison of average iteration times of various iterative methods

算例	节点数	迭代法	平均迭代次数
CASE8971	8971	本文算法	916
		BICG	964
		CGS	905
		QMR	919
BENCH	1648	本文算法	1703
		BICG	2130
		CGS	2446
		QMR	2134
CASE 300	300	本文算法	944
		BICG	916
		CGS	992
		QMR	919
CASE 118	118	本文算法	72
		BICG	110
		CGS	107
		QMR	113
CASE 57	57	本文算法	75
		BICG	125
		CGS	104
		QMR	124

综上，本文提出的方法相比于其他迭代方法，稳定性更好，收敛速度更快。因此，基于这一方法进行预处理，将获得更好的收敛性能。

4.3　预处理方法有效性验证

预处理有效性验证分成两部分进行，一部分是本文预处理方法与不完全 LU 分解预处理中其他的两种方法 ILU(1)、ILU(2)的对比验证，以验证本文预处理方法在同类预处理方法中具有优势；第二部分是将本文预处理方法与其他类别的预处理方法进行对比验证。

不完全 LU 分解预处理中常用的方法包括ILU(0)、ILU(1)、ILU(2)。为了检验本文预处理方法的效果，表 3 给出了本文预处理方法、ILU(1)、ILU(2)下不同算例的非零元注入。非零元注入是指经不完全 LU 分解后，***L*** 和 ***U*** 两个矩阵的非零元个数之和较原系数矩阵多出的非零元个数。

表 3　采用不同预处理方法时非零元注入比较
Tab. 3　Comparison of nonzero element injection among different methods of preconditioning

算例	本文方法	ILU(1)	ILU(2)
CASE39	0	309	683
CASE118	0	700	1577
CASE300	0	3842	8620
CASE8971	0	93990	205350

由表 3 可以看出，本文的预处理方法无非零元注入，维持了矩阵的稀疏性。ILU(1)、ILU(2)两类方法均有非零元注入，随着数据规模增大，非零元注入骤增，导致后续矩阵计算耗时显著增加，降低潮流整体计算效率。

Jacobi 预处理是基本迭代法如 Jacobi 和 Gauss-Seidel 迭代法中常用的预处理方法，Jacobi 预处理子将系数矩阵的主对角元素进行尺度化，是一种构造花费低、易于实施且自然并行的预处理方式[26]。为了检验本文采用的不完全 LU 分解预处理方法的效果，表 4 给出了无预处理、Jacobi 预处理及本文预处理方法下不同算例的迭代次数。其中计算精度为 0.001，表格中迭代次数均指平均迭代次数。

表 4　采用不同预处理方法时迭代次数比较
Tab. 4　Comparison of iteration times among different methods of preconditioning

算例	节点数	无预处理	Jacobi预处理	本文预处理	本文预处理迭代次数减少比例/%
CASE8971	8971	916	148	19	97.9
BENCH	1648	1703	525	44	97.4
CASE 300	300	944	153	19	98.0
CASE 118	118	72	48	12	83.3
CASE 57	57	75	60	11	85.3

由表 4 可以看出，采用本文的预处理方法，可以大幅减少迭代次数，预处理后的实际迭代次数仅为 Jacobi 预处理后的十分之一到四分之一。特别是对于节点规模大于 300 的系统效果更显著，能够节约超过 97%的迭代次数。因此，本文使用的预处理方法非常适合大规模系统的在线潮流计算。

4.4　内存占用测试

考虑到大规模系统的雅可比矩阵的高度稀疏性，本算法中使用 CSR 格式存储雅可比矩阵，且迭代计算过程中只需开辟若干一维空间，整体算法内存占用率低。从表 5 中可以看出，在 150000 节点、210000 支路规模内，算法 RAM 存储占用率可控制在 500MB 以内，而当系统规模达到百万级别即1500000 节点、2100000 条支路时，RAM 存储占用峰值仅为 2.45GB。在 GPU 缓存方面，在 150000节点、210000 支路规模内可控制在 180MB 内，而当系统规模达到百万级别即 1500000 节点、2100000条支路时，GPU 缓存占用峰值仅为 1.16GB，2G 显存的 GPU 即可支持。测试所用算例由多个CASE300 拼接平衡节点得到，其中牛顿–拉夫逊法的精度要求取 0.01，修正方程组求解的迭代法精度要求也取 0.01。

表 5　算法内存占用
Tab. 5　Memory footprint of the algorithm

拼接个数	节点数	支路数	RAM 存储占用峰值/MB	GPU 缓存占用峰值/MB
1	300	411	253.6	130.0
5	1496	2055	258.2	170.0
10	2991	4110	264.0	176.0
50	14951	20550	280.2	176.0
500	149501	205500	482.9	176.0
5000	1495001	2055000	2450.7	1164.0

5　结论

本文提出了一种新的基于不完全 LU 分解预处理迭代法的电力系统潮流算法，所提方法具有以下特点：

1）算法中的迭代方法是求解大规模线性方程组的一种稳定方法，光滑收敛，收敛速度快。系统规模较大时，与传统算法相比具有明显优势。

2)算法采用的不完全 LU 分解预处理方法相比其他预处理方法，例如 Jacobi 预处理，在减少迭代次数、提高收敛性能方面更具优势。

3）算法 RAM 存储占用和 GPU 缓存占用小，普通 PC 机配备的 GPU 即可支持。

算例测试表明，本文所提方法能够大幅减少迭代次数，提高线性方程组求解速度，在大规模系统中具有显著的优势，能够满足大规模电网在线潮流计算的需求，具有工程应用的价值。

在未来的研究过程中，本文所提的预处理和迭代方法可以尝试应用到电力系统计算的其他领域中，包括最优潮流、开断潮流、状态估计等。

参考文献

[1] 张永杰，孙秦．大型稀疏线性方程组符号 LU 分解法[C]//2006 航空宇航科学与技术全国博士生学术论坛．2006．
Zhang Yongjie，Sun Qin．Symbol LU decomposition method of large scale sparse linear equations [C]//Aeronautics & Astronautics Aircraft Design & Mechanics Branch Forum，Beijing，China，2006(in Chinese)．

[2] 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，Yaoxu，et al．Parallel solving method of linear equations based on Doolittle LU decomposition[J]．Journal of Electronics & Information Technology，2010，32(8)：2019-2022(in Chinese)．

[3] 邓兴升，孙虹虹．自适应谱修正 LU 分解法解算高病态法方程[J]．大地测量与地球动力学，2014，34(6)：135-139．
Deng Xingsheng，Sun Honghong．Self-adaptive spectrum correction LU decomposition algorithm for solving a normal equation with severity ill-conditioned matrix [J]．Journal of geodesy and geodynamics，2014，34(6)：135-139(in Chinese)．

[4] HOU，G．and L．Wang，A generalized iterative method and comparison results using projection techniques for solving linear systems[J]．Applied Mathematics and Computation，2009，215(2): 806-817．

[5] 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 (HPCC-ICESS)，2012：836-842．

[6] 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．

[7] 段治健，杨永，马欣荣，等．求解带状线性方程组的一种并行算法[J]．计算机学报，2010，37(3)：242-244．
Duan Zhijian，Yang Yong，Ma Xinrong，et al．Parallel algorithm for solving banded linear systems[J]．Journal of Computer Science，2010，37(3)：242-244(in Chinese)．

[8] Yi Xiao，Yu Liu．GPU acceleration for the gaussian elimination in magnetotelluric Occam inversion algorithm [C]//Proceedings of the 4th International Conference On Computer Engineering and Networks．Berlin：Springer-Verlag，2015：123-131．

[9] 殷战稳，韩耀飞，王亚东，等．基于 Matlab 的 Gauss-Seidel 迭代法电力系统潮流计算[J]．河南大学学报：自然版，2012，42(3)：249-253．
Yin Zhanwen，Han Yaofei，Wang Yadong，et al．Gauss-seidel iteration method in flow calculation based on matlab[J]．Journal of Henan University：Natural Science，2012，42(3)：249-253(in Chinese)．

[10] 孙秋野，陈会敏，杨家农，等．牛顿类潮流计算方法的收敛性分析[J]．中国电机工程学报，2014，34(13)：2196-2200．
Sun Qiuye，Chen Huimin，Yang Jianong，et al．Analysis on convergence of newton-like power flow algorithm [J]．Proceedings of the CSEE，2014，34(13)：2196-2200(in Chinese)．

[11] 田君杨，韦化，白晓清．半定规划优潮流的并行计算方法[J]．电网技术，2014，38(1)：175-180．
Tian Junyang，Wei Hua，Bai Xiaoqing．A parallel computational method for optimal power flow in semidefinite programming[J]．Power System Technology，2014，38(1)：175-180(in Chinese)．

[12] 于之虹，施浩波，安宁，等．暂态稳定多故障协调预防控制策略在线计算方法[J]．电网技术，2014，38(6)：1554-1561．
Yu Zhihong，Shi Haobo，An Ning，et al．A Computational approach of preventive control strategy with multi-contingency constraints for power grid transient stability [J]．Power System Technology，2014，38(6)：1554-1561 (in Chinese)．

[13] 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．Journal of Computer Science，2013，36(4)：757-772(in Chinese)．

[14] Byun J H，Ravindran A，Mukherjee A，et al．Accelerating the gauss-seidel power flow solver on a high performance reconfigurable computer[C]．IEEE Symposium on Field Programmable Custom Computing Machines．IEEE，2009：227-230．

[15] 杨罡，喻乐，刘明光．基于多轻量线程的电力系统潮流计算加速方法[J]．电网技术，2013，37(6)：1666-1671．
Yang Gang，Yu Le，Liu Mingguang．A method for accelerating power flow calculation based on multiple light threads[J]．Power System Technology，2013，37(6)：1666-1671(in Chinese)．

[16] Singh J，Aruni I．Accelerating power flow studies on

graphics processing unit[C]//India Conference．IEEE Xplore，2011：1-5．

[17] 韩祯祥．电力系统分析[M]．第 4 版．浙江大学出版社，2009：114-124．

[18] Li X S，Demmel J，Gilbert J，et al．SuperLU[M]．Springer US，2011．

[19] Zhang J．Preconditioned Krylov subspace methods for solving nonsymmetric matrices from CFD applications [J]．Computer Methods in Applied Mechanics and Engineering，2000，189(3)：825-840．

[20] SAAD Y，Vorst H．Iterative solution of linear systems in the 20th century[J]．Journal of Computational and Applied Mathematics，2001，123(1-2)：1-33．

[21] Van d V H A．BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems[J]．Siam Journal on Scientific & Statistical Computing，1992，13(2)：631-644．

[22] 柳建新，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(02)：484-491．
Liu Jianxin，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(02)：484-491(in Chinese)．

[23] 张朝晖，刘俊起，徐勤建．GPU 并行计算技术分析与应用[J]．信息技术，2009(11)：86-89．
Zhang Chaohui，Liu Junqi，Xu Qinjian．Analysis and application of the GPU parallel computing technology [J]．Information Technology，2009(11)：86-89(in Chinese)．

[24] NVIDIA．NVIDIA CUDA Compute Unified Device Architecture-Programming Guide．Version 1.1，2007．

[25] Singh J，Aruni I．Accelerating power flow studies on graphics processing unit[C]//India Conference．IEEE，2010：1-5．

[26] 谷同祥，等．迭代方法和预处理技术[M]．北京：科学出版社，2015：159-232．

唐坤杰

收稿日期：2017-05-10。

作者简介：

唐坤杰(1994)，男，博士研究生，研究方向为电力系统高性能计算方法，tangkunjie1994@163.com；

董树锋(1982)，男，博士，副教授，研究方向为状态估计和有源配电网分析等，dongshufeng@zju.edu.cn；

宋永华(1964)，男，博士，教授，研究方向为电网运行与控制和电力市场。

(**实习编辑** 郝敬乾)