

文章编号:1671-6833(2021)01-0015-06

建立电力系统状态空间方程的并行方法

王克文, 冶梦雨, 刘艳红

(郑州大学 电气工程学院, 河南 郑州 450001)

摘要:为了缩减大规模电力系统小干扰稳定性分析的计算时间,对电力系统状态空间矩阵的快速形成方法进行优化研究。针对插入式建模技术,分析状态矩阵的形成过程;采用含双重阈值的不完全 LU 分解法(ILUTP),调整相关矩阵中非零元素的位置,将矩阵转换为对角占优形式。采用双共轭梯度稳定法(BICGSTAB)对处理后的大型稀疏矩阵迭代求解;矩阵的存储方式为行压缩稀疏存储;利用 ILUTP 与 BICGSTAB 的算法特性,实现了基于 Open MP 技术的并行计算;利用两个分别包含 23 台发电机和 98 台发电机的算例,发电机均采用六阶发电机模型,励磁调节模块与原动机调速块均为系统的实际参数,对比传统方法与优化方法求解状态矩阵方法的所用时间。结果表明,该方法能够加快大型电力系统状态空间的形成过程,其并行加速比接近于 3,验证了所提方法的可行性及有效性。

关键词:状态矩阵;双共轭梯度稳定法;不完全 LU 分解;行压缩存储;Open MP

中图分类号: TM712

文献标志码: A

doi:10.13705/j.issn.1671-6833.2021.01.003

0 引言

特征值分析法以状态空间模型为基础,是电力系统小干扰稳定性分析的常用方法之一^[1]。随着现代电力系统规模不断扩大^[2],求解状态空间矩阵所需计算量急剧增大,国内外学者开始寻找快速形成状态矩阵的方法。

目前,国外通常采用拟合的方法求解状态矩阵,比如 Saitoh 等^[3]用广域监测系统(WAMS)搜集各个离散时间点的值,并利用最小二乘法拟合出状态矩阵。国内有两种主流方法:一种是用解析的方法直接得出状态矩阵,该方法计算速度很快,但是预处理较困难;另一种则是利用插入式模拟技术(PMT)^[4],先将所有系统元件转化为两种基本传输模块,再利用传输模块两端的节点编号组成关联矩阵,最终得到状态空间方程。第 2 种方法较为灵活,适用于对元件内部模块间变量的监控和灵敏度计算,但是矩阵运算较多,渐渐陷入了矩阵“维数灾难”。因此,一些学者在该方法的基础上进行了改进,比如罗丹等^[5]将状态矩阵形成过程中的相关算式进行 QR 优化和重组,通过矩阵降阶求逆的方法提高计算效率;胡威^[6]则是利用 Hager 算法缩减矩

阵的外形,再引入并行计算提高计算效率,但是 Hager 缩减过程中没有与并行计算相结合,此过程本身也会耗费一定时间。除此之外,为了缩短特征值的计算时间,有些学者利用改进的 Rayleigh 商逆迭代法来获得机电模式的特征值:首先用相关因子将机电模式分组,再参照模型降阶的方法,计算出大规模电力系统机电模式特征值^[7]。

在 PMT 的基础上充分利用相关矩阵的稀疏特性及 Open MP 技术的并行计算功能:首先将求解过程中的关键矩阵进行 ILUTP 预处理,合理舍弃掉一些非对角非零元素,使系数矩阵谱分布更集中,再采用 Krylov 子空间迭代法中的 BICGSTAB 算法,对得到的预处理模块进行快速迭代计算。ILUTP 技术与 BICGSTAB 算法均能很好地与 Open MP 结合使用,进一步加快了迭代计算速度。稀疏矩阵存储采用行压缩矩阵存储的方法降低存储量,减小计算量,从而提高整体计算效率。

1 BICGSTAB 算法及 ILUTP 技术

1.1 BICGSTAB 算法

Krylov 子空间迭代法^[8]适用于求解大型线性方程组。给定线性方程组:

收稿日期:2020-10-25;修订日期:2020-12-16

基金项目:国家自然科学基金资助项目(61473265)

作者简介:王克文(1964—),男,山西平路人,郑州大学教授,博士,博士生导师,主要从事电力系统稳定分析与控制研究,E-mail:kwwang@zzu.edu.cn。

$$Ax = b. \tag{1}$$

式中: $A \in R^{n \times n}$; $x, b \in R^n$ 。

设 K_m 为 m 维子空间,一般投影方法是从 m 维仿射子空间 $x_0 + K_m$ 中寻找上式的近似解 x_m, x_0 为迭代初值,使相应的残差满足 Petrov-Galerkin 条件:

$$r_m = b - Ax_m. \tag{2}$$

其中,要求 r_m 与预设的约束空间 L_m 正交^[9]。

记 $K_m = K_m(A, r_0)$ 为 Krylov 子空间,定义为: $K_m(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, \dots, A^{m-1}r_0\}$ 。(3)

约束空间 L_m 的选择对迭代过程具有重要影响,本文中矩阵具有非对称性,故选用双正交化方法,取 $L_m = K_m(A^T, r_0)$,其中 BICG 法是双正交化方法中最典型的算法。

BICGSTAB 算法^[10]是在 BICG 的基础上发展起来的,其避免了 BICG 中对 A^T 的计算,收敛速度比 BICG 更快。

对于 n 阶线性方程组 $Ax = b$,假设初始的近似解为 x_0 ,第 k 次近似解为 x_k ,相应的第 k 次残差为 $r_k = b - Ax_k, r_0$ 是与 r_0 不正交的非零向量, K_k, \bar{K}_k 分别是与 r_0, \bar{r}_0 有关的 k 阶 Krylov 子空间:

$$\bar{K}_k = \text{span}\{r_0, A^T r_0, \dots, (A_{k-1})^T r_0\}。 \tag{4}$$

BICGSTAB 算法在 Krylov 子空间 K_k 中选取序列,通过选择参数 α_k 和 β_k ,使得 x_k 和 r_k 满足 $r_k \in x_0 + K_k$ 且 r_k 与 \bar{K}_k 正交,同时,计算的残差向量 r_k 和搜索方向 p_k 为:

$$\begin{cases} r_{k+1} = r_k - \alpha_k A p_k; \\ p_{k+1} = r_{k+1} + \beta_k p_k。 \end{cases} \tag{5}$$

BICGSTAB 算法能够在保证精度和稳定性的同时,达到快速收敛的效果。但是其收敛速度非常依赖于系数矩阵特征值的分布,所以在一些情况中,直接使用 BICGSTAB 算法会导致收敛速度很慢,甚至根本不能收敛^[11]。可使用预条件技术将原线性方程组转化为更利于迭代收敛的线性方程组^[11],保证迭代过程不中断,提高收敛速度。

1.2 ILUTP 预处理

相比于 LU 分解,不完全 LU 分解(ILU)可以减少内存占用,更适用于大型稀疏矩阵的计算^[12]。该方法是在高斯消元的过程中强制舍弃掉一些非对角非零元素,从而得到有效的预处理子矩阵 M ,其中 $M = \bar{L}\bar{U}, \bar{L}$ 和 \bar{U} 分别为 L 和 U 的近似。

ILU 预处理有很多种算法,不同算法的主要

区别是在对 \bar{L} 和 \bar{U} 填充的舍弃策略上。当填充满足某些条件,例如位置、大小或者是位置和大小

的组合,即被舍弃^[13]。
ILUT(p, τ)^[14]是 ILU 预处理中一种具有双重阈值的算法(以下简称 ILUTP),其中 τ 表示舍弃阈值,若 $|a_{ij}|$ 小于 τ 乘以第 i 行的 2-范数,那么在 (i, j) 位置不引入填充,即舍弃 a_{ij} ; p 表示最大填充个数,即分别在 \bar{L} 和 \bar{U} 的第 i 行保留 p 个绝对值最大的元素,并将其他元素舍弃。

ILUTP 方法^[15]的优势在于其可以降低分解因子中非零元素的个数,从而缩减不完全分解时间和预条件处理的迭代时间;同时,当矩阵性质较差时,可以通过调整参数 τ 与 p 的值来提高分解因子的质量,增强迭代有效性。但是在特定情况下,需要不断尝试才能预先选择出合适的参数 τ 和 p ,否则精度将无法得到保证。因此,最优值应当结合实际问题

和矩阵特点来合理选定。
引入一个名为 *droptol* (阈值)的参数以确定是否改变变量值,使得矩阵的系数谱变得更加集中,从而将矩阵转换为对角占优形矩阵,可提高计算效率。但是由于舍弃掉了一些非零元素,计算准确度无法保证,因此又引入一个名为 *tol* (绝对误差限)的参数以保证计算精度。通过不断尝试,在速度和精度之间找出最佳平衡点。

2 状态空间方程的建立及特点分析

2.1 插入式建模技术

在插入式建模技术中,将电力系统描述为传输模块集合:非状态变量传输块(零阶)、状态变量传输块(一阶)和 5 类基本参数 k, k_a, k_b, T_a, T_b ,再根据各个系统元件和电力网络中传输模块两端的节点编号列出关联矩阵 L :

$$\begin{bmatrix} X_i \\ Y \\ M_i \end{bmatrix} = \begin{bmatrix} L_1 & L_2 & L_3 \\ L_4 & L_5 & L_6 \\ \dot{L}_7 & \dot{L}_8 & \dot{L}_9 \end{bmatrix} \begin{bmatrix} X \\ R \\ M \end{bmatrix}。 \tag{6}$$

式中: X_i 和 M_i 分别为状态变量和非状态变量模块的输入; X 和 M 则为状态变量和非状态变量模块的输出; R, Y 对应系统输入、输出的列向量。

第 n 个零阶和一阶的传输块方程可表示为:

$$\begin{cases} m_n = k_n m_{in}; \\ x_n = x_{in}(k_{an} + pT_{an}) / (k_{bn} + pT_{bn})。 \end{cases} \tag{7}$$

那么,所有零阶和一阶的传输块方程可表示为:

$$\begin{cases} M = KM_i; \\ \dot{X} = -K_b X + K_a X_i + K_t \dot{X}_i。 \end{cases} \tag{8}$$

素);Colind 数组和 Val 数组一样,大小为矩阵 B 的非零元素的个数,保存 Val 数组中元素的列索引;Rowp 数组大小为矩阵 B 的行数,保存矩阵 B 的每行第一个非零元素在 Val 中的索引。该方法可以节省很多空间,只需要 $2nnz+n+1$ 个存储单元,而不是 n^2 个单元,其中 nnz 为稀疏矩阵中非零元素的个数。

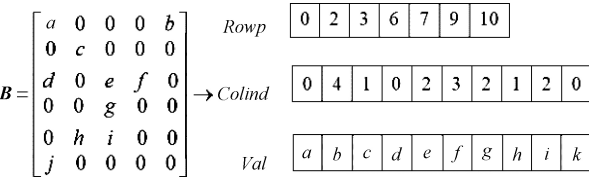


图2 稀疏矩阵 B 的行压缩示意图

Figure 2 Schematic of sparse matrix B using CRS

3.2 预处理参数设定

本文采用 ILUTP 预处理方法,将 L_0 矩阵进行预处理,处理后得到的 \bar{L}_0 矩阵增强 BICGSTAB 迭代的稳定性,加快收敛速度。ILUTP 技术应用的 关键在于对其中部分参数的设定,该研究中这些 关键参数的具体设定如下:

```
integer::ILU_mode=1
real*8::droptol=0.001
integer::lfil=10
real*8::permtol=0.5
integer::mbloc=0
```

通过阈值来舍弃数值小的元素,从而影响 生成的 ILU 矩阵大小。这个值设定得太小会导 致无法收敛,太大则无法保证精度。本文经过 多次修改对比,最终选用 $droptol_def=0.001$, $tol=0.000\ 01$ 达到理想的平衡点。下降公差 $lfil$ 用来控制 L 和 U 中每行限定的最大非对角元素 个数,填充参数设定得较小是为了保证计算精 度。列交换容差 $permtol$ 合理的值通常介于 0.5 和 0.01,本文矩阵维数很大,因此将列交换容差 设为 0.5。图 3 为预处理之后得到的 L_0 矩阵的 数据分布示意图,经过 ILUTP 预处理之后,矩阵 转换为对角占优型矩阵,一方面更有利于 BICG- STAB 算法的求逆迭代,另一方面更易与 Open MP 的并行功能相结合。

3.3 BICGSTAB 算法实现

程序中 BICGSTAB 算法流程如下:

Step 1 对于线性方程组 $Ax=b$,给定初始值 x_0 ,最大迭代次数 k_{max} ,计算 $r_0=b-Ax_0$,其中给定 $\bar{r}_0=r_0$,使得内积 $(\bar{r}_0,r_0)\neq 0$,同时给定 $\rho_0=\alpha=$

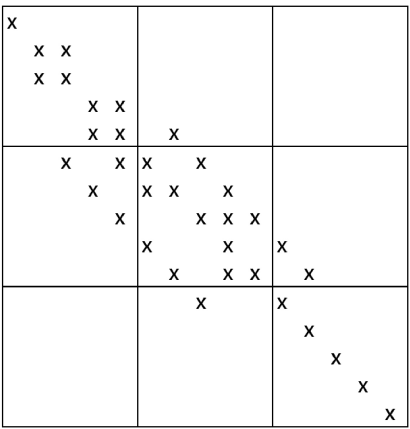


图3 转换后的 L_0 矩阵

Figure 3 The transformed L_0 matrix

```
omega_0 = 1, nu_0 = p_0 = 0, i = 1;
Step 2 rho_i = (r_0, r_0), beta = (rho_i / rho_{i-1}) * (alpha / omega_{i-1}),
p_i = r_{i-1} + beta * (p_{i-1} - omega_{i-1} * v_{i-1}), v_i = A * p_i;
Step 3 alpha = rho_i * (r_0, v_i), s = r_{i-1} - alpha * v_i, t = A * s,
omega_i = (t, s) / (t, t);
Step 4 x_i = x_{i-1} + alpha * p_i + omega_i * s;
Step 5 若 x_i 达到精度要求,则转 Step 7;
Step 6 r_i = s - omega_i * t, i = i + 1, 若 i < k_max, 转
Step 2;
Step 7 输出 x_i, 算法结束。
```

3.4 Open MP 并行实现

Open MP 作为并行提速计算的一种工具,适 合处理独立循环或者分开的子任务,程序编写简 单,具有强扩展性,可支持 Fortran 等多种编程 语言^[8]。

在状态矩阵形成的程序中有很多 DO 循环语 句,如对矩阵 L_1, L_3, L_7, L_9 赋值、对矩阵 L_0 求逆 过程中的迭代等,这些任务在程序执行过程中可 以独立执行,因此结合 Open MP 进行并行处理。 其中 BICGSTAB 算法中求解方程组时的并行语句 如下:

```
!$OMP Parallel
!$OMP do Private(i,x0,x,bi)
do i=1,N
bi=0.d0; bi(i)=1.d0; x0=0.d0
call Precond(bi,x0)
x0=x0(Iperm(1:N))
call Bicgstab(bi,x0,x,max_it,tol)
a(1:N,i)=x(Iperm(1:N))
enddo
!$OMP enddo
!$OMP end Parallel
```

4 算例分析

某省电力网络的两个实际算例分别包含 23 台发电机和 98 台发电机,发电机均采用六阶发电机模型,励磁调节模块与原动机调速模块均为系统的实际参数。

程序采用 Fortran 语言编写,编写环境为 Visual Studio 2010 平台、Intel Fortran 12,在 2.14 GHz 主频、32 GB 内存、8 核计算机上运行实现。存储方式为行压缩存储形式。

通过对两种求解状态矩阵方法的所用时间进行比较来验证可行性。两种方法分别为原方法和现方法,均利用插入式建模技术及 Open MP 的并行功能。其中,原方法为文献[4]中基础的 PMT 方法,矩阵求逆采用直接 LU 分解法,矩阵存储采用三元组技术存储;现方法为预处理的不完全 LU 分解法,矩阵求逆采用 BIGSTAB 算法迭代求逆,矩阵存储采用行压缩存储技术。分别将原方法及现方法无并行所用时间、原方法及现方法加入并行所用时间进行比对,验证现方法的加速效果。

算例 1 为 23 台发电机,系统中包含 90 个节点,210 条支路,状态矩阵的阶数为 257 阶。算例 1 中矩阵 L_1 、 L_3 、 L_9 及状态矩阵 A 的阶数、非零元素个数和对应的稀疏程度如表 1 所示。本算例为了对比算法方面的改进,设为单线程执行,各方法运行时间的对比如表 2 所示。

表 1 算例 1 中各矩阵的信息

Table 1 Information of each matrix in example 1

编号	矩阵	矩阵规模	非零元素数	稀疏度/%
1	L_1	257×257	126	99.80
2	L_3	257×570	184	99.91
3	L_9	570×570	2 120	99.39
4	A	257×257	8 281	87.49

表 2 算例 1 中各方法的运行时间

Table 2 Run time of each method in example 1

编号	运算项	原方法无并行时间/s	现方法无并行时间/s
1	L_1, L_3, L_9	2	2
2	L_9 逆矩阵	8	7
3	A	10	9

由表 2 得出,在矩阵阶数较少时,BIGSTAB 算法及预处理技术的运用对计算效率并没有非常明显的提高。

算例 2 为 98 台发电机,系统中包含 2 394 个节点,5 512 条支路,状态矩阵的阶数为 1 736 阶。

其中矩阵 L_1 、 L_3 、 L_9 及状态矩阵 A 的矩阵规模、非零元素数和对应的稀疏程度如表 3 所示。算例 2 中加入了并行计算功能,为八线程执行,各方法运行时间的对比如表 4 所示。

表 3 算例 2 中各矩阵的信息

Table 3 Information of each matrix in example 2

编号	矩阵	矩阵规模	非零元素数	稀疏度/%
1	L_1	1 736×1 736	770	99.98
2	L_3	1 736×7 169	1 016	99.98
3	L_9	7 169×7 169	35 078	99.94
4	A	1 736×1 736	242 273	91.95

表 4 算例 2 中各方法的运行时间

Table 4 Run time of each method in example 2

编号	运算项	原方法无并行时间/s	现方法无并行时间/s	原方法时间/s	现方法时间/s
1	L_1, L_3, L_9	4	4	3	3
2	L_9 逆矩阵	2 948	1 924	979	175
3	A	3 485	2 463	1 196	387

综合各表中数据可得:当矩阵规模较小时,ILU 预处理及 BICGSTAB 算法对运行时间有一定改善,但效果并不明显;当系统规模较大时, L_9 矩阵稀疏度增高,越来越接近于 1,稀疏技术的利用率变高。

通过原方法无并行和现方法无并行的运算时间对比,说明 ILUTP 及 BICGSTAB 算法可以提高迭代速度;引入并行计算后,加速效果更加明显,并行加速比接近于 3。

5 结论

在插入式模拟技术的基础上,对 L_9 矩阵的求逆过程进行优化,利用矩阵的稀疏特性引入 ILUTP 预处理技术,并与 BICGSTAB 算法相结合,提高了迭代速度,使矩阵转化为更有利于并行计算的形式;然后通过行压缩存储技术和 Open MP 中的并行功能,加速了整个求逆过程;最后分别对两个不同规模的算例进行分析计算,验证了本文方法能够缩短状态矩阵形成的时间,有利于电力系统小干扰稳定性的计算与分析。

在之后的研究中,一方面可以尝试不同的算法转换矩阵形式,比如将 L_9 矩阵变换成对角加边的形式,再进行并行化处理;另一方面也可尝试使用 Open MP+MPI 混合编程,利用不同的并行处理方式加快程序执行。

参考文献：

- [1] 刘畅,刘卉,王克文.特征值分析法在地区电网稳定分析中的应用[J].电测与仪表,2012,49(9):43-46.
- [2] 张振,刘艳红.基于特征值的单机无穷大电力系统随机稳定性分析[J].郑州大学学报(工学版),2018,39(4):58-63.
- [3] SAITOH H,MIURA K,ISHIOKA O,et al. On-line modal analysis based on synchronized measurement technology [C]//Proceedings of International Conference on Power System Technology. New York: IEEE, 2002: 817-822.
- [4] 钟志勇,谢志棠,王克文.适用于电力系统动态稳定分析的元件建模新方法[J].中国电机工程学报,2000,20(3):30-33.
- [5] 罗丹,王克文,王君亮.小干扰稳定分析中状态矩阵的快速形成[J].电测与仪表,2015,52(10):56-60,91.
- [6] 胡巍.构建电力系统状态矩阵的并行计算研究[D].郑州:郑州大学,2017.
- [7] 王宁.电力系统状态空间方程的优化降阶研究[D].郑州:郑州大学,2017.
- [8] 李晓梅,吴建平.Krylov 子空间方法及其并行计算[J].计算机科学,2005,32(1):19-20,40.
- [9] 范晋军.基于 MATLAB 软件的大地电磁测深正演模拟研究[J].能源技术与管理,2019,44(2):155-156.
- [10] 李晓梅,迟利华.并行求解大型稀疏线性方程组的研究概况[J].指挥技术学院学报,1999,10(3):1-8.
- [11] 李月卉.电磁场数值求解中迭代方法与预条件技术研究[D].成都:电子科技大学,2011.
- [12] 张炎.大规模稀疏线性系统的稀疏近似逆预处理技术[D].北京:清华大学,2013.
- [13] MAYER J. Alternative weighted dropping strategies for ILUTP [J]. SIAM journal on scientific computing, 2006, 27(4):1424-1437.
- [14] SAAD Y. Iterative methods for sparse linear systems [M]. Philadelphia: SIAM, 2003.
- [15] 吴建平,刘兴平,王正华,等.二维三温能量方程组离散求解的两个新预处理技术[J].计算物理,2005(4):283-291.
- [16] 周凌焱,刘成龙,张强,等.稀疏矩阵压缩技术在 CPⅢ网平差中的应用研究[J].铁道科学与工程学报,2014,11(6):142-147.

Parallel Method for Establishing State Space Equation of Power System

WANG Kewen, YE Mengyu, LIU Yanhong

(School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract: In order to reduce the calculation time of small-disturbance stability analysis of large-scale power systems, the method of rapid formation of power system state space matrix was optimized. According to the plug-in modeling technology, the formation process of the state matrix was analyzed; the incomplete LU decomposition method (ILUTP) with double thresholds was used to adjust the position of the non-zero elements in the correlation matrix, and the matrix was converted to a diagonally dominant form; double conjugate gradient stabilization method (BICGSTAB) was used iteratively to solve the processed large sparse matrix; the storage method of the matrix was row-compressed sparse storage; the algorithm characteristics of ILUTP and BICGSTAB were used to realize the parallel computing based on Open MP technology; two calculation examples of 23 generators and 98 generators were included respectively. The generators adopted the sixth-order generator model. The excitation adjustment module and the prime mover speed control block were the actual parameters of the system, and the traditional methods and optimization methods were compared. The time taken to solve the state matrix method verified the feasibility and effectiveness of the proposed method. The results showed that this method could speed up the formation of the state space of large power systems, and its parallel speed-up ratio was close to 3.

Key words: state matrix; BICGSTAB; incomplete LU; compressed row storage; Open MP