

DOI: 10.19783/j.cnki.pspc.190171

基于矩阵分块递归求逆的电力系统机电暂态并行算法

黄远洋

(国网冀北电力有限公司工程管理分公司, 北京 100038)

摘要: 在机电暂态仿真计算中, 常用的稀疏三角分解存在分解效率不高、并行度低等问题。基于隐式梯形积分法, 首先根据雅可比矩阵的特点, 将雅可比矩阵分裂, 然后利用扩展的 Sherman-Morrison 求逆公式递推求解, 从而避免了稀疏三角分解, 得到了一类新的机电暂态仿真算法。利用高性能的图形处理器(Graphics Processing Unit, GPU) 对大规模合成系统进行仿真测试。结果分析表明, 所提出的并行算法可以获得较好的加速比和良好的扩展性。

关键词: 机电暂态; 隐式联立求解方法; Sherman-Morrison 公式; GPU

A parallel algorithm for electromechanical transient simulation by recursive block matrix inversion

HUANG Yuanyang

(State Grid Jibei Electric Power Company Limited Engineering Management Company, Beijing 100038, China)

Abstract: In electromechanical transient simulation, there are some problems in sparse triangular decomposition, such as low decomposition efficiency and low parallelism. Based on the implicit trapezoidal integration method, this paper first splits Jacobian matrix according to the characteristics of the Jacobian matrix, and then uses the extended Sherman-Morrison inverse formula to solve it recursively, so as to avoid sparse triangular decomposition, finally a novel kind of electromechanical transient simulation algorithm is presented. The large-scale synthesis system is tested and analyzed by using high-performance Graphics Processing Unit (GPU). The results show that the parallel algorithm proposed in this paper can achieve better acceleration ratio and good scalability.

This work is supported by National Natural Science Foundation of China (No. 51309258).

Key words: electromechanical transient; simultaneous implicit approach; Sherman-Morrison formula; Graphic Processing Unit

0 引言

随着互联电网规模的扩大, 对电力系统机电暂态仿真计算的时效性提出了更快、更准的要求, 并行处理技术^[1]是实现互联大电网机电暂态实时分析的有效途径^[2-5]。迄今为止, 有关电力系统机电暂态并行算法的研究很多, 从实现原理上来看, 主要划分为三类: 空间并行算法^[5-8]、时间并行算法^[9-10]、两者的结合——时空并行算法^[11-12]。

空间并行算法主要采用网络分割、矩阵分块或矩阵分裂的思路^[13]。基于网络分割或矩阵分裂方法的空间并行算法很难达到高度的空间并行度, 对中、小规模电力系统更是如此。此外, 对一个实际的

电力系统, 传统的网络分割方法很难将其划分为多个完全“均衡”的子系统。因此, 传统的、基于网络分割的空间并行算法通常还存在“负载均衡”方面的问题。若充分结合大规模电力系统的分区域特性, 传统的空间并行算法可以获得较好的并行计算效率, 但此类算法本身不具备通用性。时间并行算法是将机电暂态仿真中的微分方程组在多个时间点上进行统一离散, 将得到的方程组与网络代数方程组联立求解。时间并行算法按时间点解耦将整个计算任务在各进程或线程间均匀分配, 其缺点是积分步长过大容易导致牛顿迭代不收敛, 因此, 积分步长与牛顿迭代收敛性之间的矛盾直接影响了并行算法的加速效果^[14]。

随着近几年来, 图形处理器(Graphics Processing Unit, GPU)^[15]由于擅长矢量运算的并行处理, 已经

成为电力系统潮流^[16]、机电暂态仿真计算^[17-19]的主流硬件平台。GPU 是采用多线程实现并行计算, 尽管多核 CPU 也可以按多线程组织并行计算, 但 GPU 中可并发的多线程数量远远超出多核 CPU。GPU 线程(thread)以网格(grid)的方式组织, 而每个网格中又包含多个线程块(thread block), 每一个线程块可包含多达 512 个线程。因此, 在 GPU 中, 线程可以按照粗粒度的线程块和细粒度的线程两个层次进行组织, 这就是 CUDA^[20-21]的双层线程模型。

本文在经典隐式梯形求解方法的基础上, 结合雅克比块中的分块对角特性, 将雅克比矩阵分裂成两个矩阵的和, 利用扩展的 Sherman-Morrison 求逆公式^[22], 推导出了基于矩阵分块求逆的机电暂态稳定空间并行计算方法。在 GPU 平台测试结果表明了该方法可以获得良好的加速效果和可扩展性能。

1 隐式联立求解方法

机电暂态仿真的数学模型可以用以下的微分代数方程组(Differential Algebraic Equations, DAEs)来描述。

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{V}) \\ \mathbf{g}(\mathbf{x}, \mathbf{V}) = \mathbf{0} \end{cases} \quad (1)$$

式中: \mathbf{x} 为描述系统动态特性的状态变量; \mathbf{V} 为网络节点电压。

目前, 主要利用隐式梯形积分法将微分方程组离散化为非线性代数方程组, 然后利用牛顿法对差分代数方程组和代数方程组进行联立求解。

1) 利用隐式梯形积分方法, 将微分方程组差分化为

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{1}{2}h[\mathbf{f}(\mathbf{x}_n, \mathbf{V}_n) + \mathbf{f}(\mathbf{x}_{n+1}, \mathbf{V}_{n+1})] \quad (2)$$

式(2)即为

$$\mathbf{x}_{n+1} - \frac{1}{2}h\mathbf{f}(\mathbf{x}_{n+1}, \mathbf{V}_{n+1}) = \mathbf{x}_n + \frac{1}{2}h\mathbf{f}(\mathbf{x}_n, \mathbf{V}_n) \quad (3)$$

式中, $\mathbf{x}(t_n) = \mathbf{x}_n, \mathbf{V}(t_n) = \mathbf{V}_n$ 为 t_n 时刻的值。

2) 联立差分代数方程组(2)与代数方程组 $\mathbf{g}(\mathbf{x}_{n+1}, \mathbf{V}_{n+1}) = \mathbf{0}$, 利用牛顿法对其进行迭代求解。具体计算公式为

$$\mathbf{J}(\mathbf{x}, \mathbf{V}) \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{V} \end{bmatrix} = \begin{bmatrix} \nabla \mathbf{f}(\mathbf{x}, \mathbf{V}) \\ \nabla \mathbf{g}(\mathbf{x}, \mathbf{V}) \end{bmatrix} \quad (4)$$

$$\mathbf{J}(\mathbf{x}, \mathbf{V}) \equiv \begin{bmatrix} \mathbf{I} - \frac{1}{2}h \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{V})}{\partial \mathbf{x}} & -\frac{1}{2}h \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{V})}{\partial \mathbf{V}} \\ \frac{\partial \mathbf{g}(\mathbf{x}, \mathbf{V})}{\partial \mathbf{x}} & \frac{\partial \mathbf{g}(\mathbf{x}, \mathbf{V})}{\partial \mathbf{V}} \end{bmatrix} \quad (5)$$

$$\begin{cases} \nabla \mathbf{f}(\mathbf{x}, \mathbf{V}) = \mathbf{C}_n - \mathbf{x} + \frac{1}{2}h\mathbf{f}(\mathbf{x}, \mathbf{V}) \\ \nabla \mathbf{g}(\mathbf{x}, \mathbf{V}) = -\mathbf{g}(\mathbf{x}, \mathbf{V}) \end{cases} \quad (6)$$

其中, 方程(4)称为牛顿迭代方程。

从上述求解步骤中可以看出: 基于隐式梯形积分法的求解过程中没有“交接误差”, 具有较好的收敛性。但需要多次稀疏三角分解雅可比矩阵式(5), 因其分解效率不高、并行度低等问题, 一直以来难以适用于互联大电网并行计算。

雅可比矩阵 $\mathbf{J}(\mathbf{x}, \mathbf{V})$ 中 4 个分块矩阵具有一些重要的特点。下面对此进行具体的描述。

设电力系统包含 m 台发电机、 n 个网络节点, 同时设定第 j 台发电机 ($j \in (1, m)$) 连接于网络中的第 i 个节点 ($i \in (1, n)$), 并将这个连接关系记为 $j \subset i$ 。将第 j 台发电机及其相应的动态元件的状态变量记为 \mathbf{x}_j , 并设 \mathbf{x}_j 的维数为 u_j 。

由于机电暂态仿真中动态元件之间没有耦合关系, 因此, 雅可比矩阵 $\partial \mathbf{f}(\mathbf{x}, \mathbf{V}) / \partial \mathbf{x}$ 可以表述成一个分块的对角矩阵, 即

$$\begin{cases} \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{V})}{\partial \mathbf{x}} = \text{diag}(\mathbf{J}_{G_j}) \\ \mathbf{J}_{G_j} \equiv \frac{\partial \mathbf{f}_j(\mathbf{x}_j, \mathbf{V}_i)}{\partial \mathbf{x}_j} \in R^{u_j \times u_j} \end{cases}, j \in (1, m) \quad (7)$$

而代数方程组可以写为如式(8)形式。

$$\mathbf{Y}_N \mathbf{V} = \mathbf{I}(\mathbf{x}, \mathbf{V}) \quad (8)$$

式中: $\mathbf{Y}_N \in R^{2n \times 2n}$ 为节点导纳矩阵, 它是一个分块常系数矩阵; $\mathbf{I}(\mathbf{x}, \mathbf{V})$ 为网络节点注入电流列向量, 它是状态变量 \mathbf{x} 和节点电压 \mathbf{V} 的函数。依据式(8), 有

$$\frac{\partial \mathbf{g}(\mathbf{x}, \mathbf{V})}{\partial \mathbf{V}} = \mathbf{Y}_N - \frac{\partial \mathbf{I}(\mathbf{x}, \mathbf{V})}{\partial \mathbf{V}} \equiv \mathbf{Y}_N + \mathbf{Y}_G(\mathbf{x}) \quad (9)$$

$$\frac{\partial \mathbf{g}(\mathbf{x}, \mathbf{V})}{\partial \mathbf{x}} = -\frac{\partial \mathbf{I}(\mathbf{x}, \mathbf{V})}{\partial \mathbf{x}} \quad (10)$$

式(9)中的 $\mathbf{Y}_G(\mathbf{x})$ 可以进一步表述成为

$$\mathbf{Y}_G(\mathbf{x}) = \text{diag}(\mathbf{y}_{G_i}(\mathbf{x}_j)), i \in (1, n) \quad (11)$$

$$\begin{cases} \mathbf{y}_{G_i}(\mathbf{x}_j) = -\frac{\partial \mathbf{I}_i(\mathbf{x}_j, \mathbf{V}_i)}{\partial \mathbf{V}_i}; j \subset i \\ \mathbf{y}_{G_i}(\mathbf{x}_j) = \mathbf{0}; j \not\subset i \end{cases} \quad (12)$$

定义

$$\frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{V})}{\partial \mathbf{V}} = [\mathbf{a}_{ji}] \equiv \mathbf{a}, j \in (1, m); i \in (1, n) \quad (13)$$

$$\frac{\partial \mathbf{g}(\mathbf{x}, \mathbf{V})}{\partial \mathbf{x}} = [\mathbf{\beta}_{ij}] \equiv \mathbf{\beta}, i \in (1, n); j \in (1, m) \quad (14)$$

则有

$$\begin{cases} \alpha_{ji} = \frac{\partial f_j(\mathbf{x}_j, V_i)}{\partial V_i}, j \subset i \\ \alpha_{ji} = \mathbf{0}, j \not\subset i \end{cases} \quad (15)$$

$$\begin{cases} \beta_{ij} = -\frac{\partial I_i(\mathbf{x}_j, V_i)}{\partial \mathbf{x}_j}, j \subset i \\ \beta_{ij} = \mathbf{0}, j \not\subset i \end{cases} \quad (16)$$

2 基于扩展 Sherman-Morrison 公式的并行算法

定义

$$\mathbf{Y} = \mathbf{Y}_N + \mathbf{Y}_G(\mathbf{x}) \quad (17)$$

$$\mathbf{A}_i = \mathbf{I}_{u_j} - \frac{1}{2} h \mathbf{J}_{G_j}, j \in (1, m) \quad (18)$$

式中, \mathbf{I}_{u_j} 为 u_j 维的单位矩阵。将 \mathbf{Y}_N 中的(分块)对角元素记为 $y_N^i, i \in (1, n)$, 对矩阵 \mathbf{Y} 进行分裂, 即

$$\mathbf{Y} = \bar{\mathbf{Y}} + \Delta \mathbf{Y}, \Delta \mathbf{Y} = \text{diag}(y_N^i) + \mathbf{Y}_G(\mathbf{x}) \quad (19)$$

显然, $\bar{\mathbf{Y}}$ 是一个常系数矩阵, 而 $\Delta \mathbf{Y}$ 是一个分块对角矩阵。

基于上述定义, 方程(5)即雅可比矩阵 $\mathbf{J}(\mathbf{x}, \mathbf{V})$ 可以表述为

$$\mathbf{J}(\mathbf{x}, \mathbf{V}) = \begin{bmatrix} \mathbf{A}_1 & & & \kappa \tilde{\alpha}_1 \\ & \ddots & & \vdots \\ & & \mathbf{A}_m & \kappa \tilde{\alpha}_m \\ \hline \tilde{\beta}_1 & \cdots & \tilde{\beta}_m & \bar{\mathbf{Y}} + \Delta \mathbf{Y} \end{bmatrix} \quad (20)$$

式中: $\kappa = -h/2$; $\tilde{\beta}_j, j \in (1, m)$ 为矩阵中 β 中的第 j 个(分块)列向量; $\tilde{\alpha}_j, j \in (1, m)$ 为矩阵中 α 中的第 j 个(分块)行向量。对上述雅可比矩阵进行分裂, 即

$$\mathbf{J}(\mathbf{x}, \mathbf{V}) = \mathbf{J}_d + \Delta \mathbf{J} \quad (21)$$

$$\mathbf{J}_d \equiv \begin{bmatrix} \mathbf{A} & \\ & \Delta \mathbf{Y} \end{bmatrix}, \mathbf{A} = \text{diag}(\mathbf{A}_j), j \in (1, m) \quad (22)$$

$$\Delta \mathbf{J} = \begin{bmatrix} & & & \kappa \tilde{\alpha}_1 \\ & \mathbf{0} & & \vdots \\ & & & \kappa \tilde{\alpha}_m \\ \hline \tilde{\beta}_1 & \cdots & \tilde{\beta}_m & \bar{\mathbf{Y}} \end{bmatrix} \quad (23)$$

则有

$$\mathbf{J}(\mathbf{x}, \mathbf{V}) = \mathbf{J}_d \begin{bmatrix} \mathbf{I}_{u_1} & & & \mathbf{E}_1 \\ & \ddots & & \vdots \\ & & \mathbf{I}_{u_m} & \mathbf{E}_m \\ \hline \mathbf{H}_1 & \cdots & \mathbf{H}_m & \mathbf{I} + \Delta \mathbf{Y}^{-1} \bar{\mathbf{Y}} \end{bmatrix} \quad (24)$$

式中:

$$\mathbf{J}(\mathbf{x}, \mathbf{V}) = \mathbf{J}_d \begin{bmatrix} \mathbf{I}_{u_1} & & & \kappa \mathbf{A}_1^{-1} \tilde{\alpha}_1 \\ & \ddots & & \vdots \\ & & \mathbf{I}_{u_m} & \kappa \mathbf{A}_m^{-1} \tilde{\alpha}_m \\ \hline \tilde{\beta}_1 & \cdots & \tilde{\beta}_m & \Delta \mathbf{Y}^{-1} \bar{\mathbf{Y}} \end{bmatrix} \quad (25)$$

$$\mathbf{J}_d = \begin{bmatrix} \bar{\mathbf{J}} & \mathbf{0} \\ \mathbf{0} & \mathbf{Y}_N \end{bmatrix}, \Delta \mathbf{J} = \begin{bmatrix} \mathbf{0} & -\frac{1}{2} h \alpha \\ \beta & \mathbf{Y}_G(\mathbf{x}) \end{bmatrix} \quad (26)$$

定义

$$d_x = \sum_{j=1}^m u_j, d = d_x + 2n \quad (27)$$

设矩阵 α 中含有非零元素的总行数为 τ , 矩阵 $\Delta \mathbf{J}$ 中含有非零元素的总行数为 ϕ ; 设 $\mathbf{p}_k, \mathbf{q}_k \in R^{d \times 1}$ 为 d 维的列向量; 设 $\mathbf{e}_k \in R^{d \times 1}$ 为 d 维的单位列向量, 其中的第 k 个元素为 1, 其他全部为 0。令

$$\mathbf{P} = [\mathbf{p}_k] = [\mathbf{e}_k], k \in (1, \phi) \quad (28)$$

$$\mathbf{Q}^T = [\mathbf{q}_k], k \in (1, \phi) \quad (29)$$

$$\sum_{k=1}^{\phi} \mathbf{p}_k \mathbf{q}_k^T = \mathbf{P} \mathbf{Q} = \begin{bmatrix} \mathbf{0} & -\frac{1}{2} h \alpha \\ \beta & \mathbf{Y}_G(\mathbf{x}) \end{bmatrix} \quad (30)$$

则有

$$\Delta \mathbf{J} = \sum_{k=1}^{\phi} \mathbf{p}_k \mathbf{q}_k^T \quad (31)$$

经典的 Sherman-Morrison 公式可概述为

$$(\mathbf{A}_0 + \mu \mathbf{v}^T)^{-1} = \mathbf{A}_0^{-1} - \frac{\mathbf{A}_0^{-1} \mu \mathbf{v}^T \mathbf{A}_0^{-1}}{1 + \mathbf{v}^T \mathbf{A}_0^{-1} \mu} \quad (32)$$

式中, $\mathbf{A}_0 \in R^{n \times n}$; $\mu, \mathbf{v} \in R^{n \times 1}$ 。文献[22]对传统的 Sherman-Morrison 公式进行扩展, 导出了更为普适的 Sherman-Morrison 公式, 即

$$(\mathbf{A}_0 + \sum_{k=1}^s \mu_k \mathbf{v}_k^T)^{-1} = \mathbf{A}_0^{-1} - \mathbf{A}_0^{-1} \mathbf{U} \mathbf{R}^{-1} \mathbf{W}^T \mathbf{A}_0^{-1} \quad (33)$$

式中, $\mu_k, \mathbf{v}_k \in R^{n \times 1}, k \in (1, s)$ 。

$$\mathbf{U} = [\mathbf{u}_k], \mathbf{u}_k \in R^{n \times 1}, k \in (1, s) \quad (34)$$

$$\mathbf{W} = [\mathbf{w}_k], \mathbf{w}_k \in R^{n \times 1}, k \in (1, s) \quad (35)$$

$$\mathbf{R} = \text{diag}(r_k), k \in (1, s) \quad (36)$$

$$\begin{cases} \mathbf{u}_k = \mu_k - \sum_{i=1}^{k-1} \frac{\mathbf{w}_i^T \mathbf{A}_0^{-1} \mu_k}{r_i} \mathbf{u}_i \\ \mathbf{w}_k = \mathbf{v}_k - \sum_{i=1}^{k-1} \frac{\mathbf{v}_k^T \mathbf{A}_0^{-1} \mathbf{u}_i}{r_i} \mathbf{w}_i \end{cases}, k \in (1, s) \quad (37)$$

$$r_k = 1 + \mathbf{v}_k^T \mathbf{A}_0^{-1} \mathbf{u}_k = 1 + \mathbf{w}_k^T \mathbf{A}_0^{-1} \mu_k, k \in (1, s) \quad (38)$$

定义

$$\Delta \mathbf{z} = \begin{bmatrix} \Delta \mathbf{x} \\ \Delta \mathbf{V} \end{bmatrix}, \quad \nabla F(\mathbf{x}, \mathbf{V}) = \begin{bmatrix} \nabla f(\mathbf{x}, \mathbf{V}) \\ \nabla g(\mathbf{x}, \mathbf{V}) \end{bmatrix} \quad (39)$$

结合上述推导, 将扩展的 Sherman-Morrison 矩阵求逆公式应用于牛顿迭代方程(4)可得

$$\Delta \mathbf{z} = \Delta \tilde{\mathbf{z}} - \mathbf{J}_d^{-1} \tilde{\mathbf{P}} \tilde{\mathbf{R}}^{-1} \tilde{\mathbf{Q}}^T \Delta \tilde{\mathbf{z}} \quad (40)$$

式中:

$$\Delta \tilde{\mathbf{z}}^T = \left[\left(\Delta \tilde{\mathbf{x}}_j^T \right) \mid \left(\Delta \tilde{\mathbf{V}}^T \right) \right], \quad j \in (1, m) \quad (41)$$

$$\Delta \tilde{\mathbf{x}}_j = \tilde{\mathbf{J}}_j^{-1} \nabla f_j(\mathbf{x}_j, \mathbf{V}_j), \quad j \in (1, m) \quad (42)$$

$$\Delta \tilde{\mathbf{V}} = \mathbf{Y}_N^{-1} \nabla g(\mathbf{x}, \mathbf{V}) \quad (43)$$

$$\tilde{\mathbf{P}} = [\tilde{\mathbf{p}}_k], \quad \tilde{\mathbf{Q}} = [\tilde{\mathbf{q}}_k], \quad k \in (1, \phi) \quad (44)$$

$$\tilde{\mathbf{R}} = \text{diag}(\tilde{r}_k), \quad k \in (1, \phi) \quad (45)$$

$$\left\{ \begin{aligned} \tilde{\mathbf{p}}_k &= \mathbf{p}_k - \sum_{i=1}^{k-1} \frac{\tilde{\mathbf{q}}_i^T (\mathbf{J}_d^{-1} \mathbf{p}_k)}{r_i} \tilde{\mathbf{p}}_i \\ \tilde{\mathbf{q}}_k &= \mathbf{q}_k - \sum_{i=1}^{k-1} \frac{(\mathbf{q}_k^T \mathbf{J}_d^{-1}) \tilde{\mathbf{p}}_i}{r_i} \tilde{\mathbf{q}}_i \end{aligned} \right., \quad k \in (1, \phi) \quad (46)$$

$$\tilde{r}_k = 1 + \tilde{\mathbf{q}}_k^T (\mathbf{J}_d^{-1} \mathbf{p}_k), \quad k \in (1, \phi) \quad (47)$$

从上述推导出的计算公式, 可以看出:

1) $\Delta \tilde{\mathbf{x}}_j$, $j \in (1, m)$ 的求解是完全解耦的, 具有 m 个空间并行度。

2) 当 $k \leq \tau$ 时, $\mathbf{J}_d^{-1} \mathbf{p}_k$ 的计算主要涉及 $\tilde{\mathbf{J}}^{-1} \tilde{\mathbf{e}}_k$, $k \in (1, \tau)$ 的计算, 这里, $\tilde{\mathbf{e}}_k$ 是 d_x 维的列向量; 当 $k > \tau$ 时, $\mathbf{J}_d^{-1} \mathbf{p}_k$ 的计算主要涉及 $\mathbf{Y}_N^{-1} \tilde{\mathbf{e}}_j$, $j \in (1, \phi - \tau)$ 的计算, 这里, $\tilde{\mathbf{e}}_j$ 是 $2n$ 维的列向量。显然, 前者的计算具有 m 个空间并行度, 而后者计算具有 $(\phi - \tau)$ 个空间并行度。

3) 当 $k \leq \tau$ 时, $\mathbf{q}_k^T \mathbf{J}_d^{-1}$ 的计算主要涉及 $\alpha \mathbf{Y}_N^{-1}$ 的计算。由于 α 只包含 m 个不全为零的稀疏向量, 因此, $\alpha \mathbf{Y}_N^{-1}$ 的计算具有 m 个空间并行度。

4) 当 $k > \tau$ 时, $\mathbf{q}_k^T \mathbf{J}_d^{-1}$ 的计算主要涉及 $\beta \tilde{\mathbf{J}}^{-1}$ 以及 $\mathbf{Y}_G(\mathbf{x}) \mathbf{Y}_N^{-1}$ 的计算。与 α 一样, β 和 $\mathbf{Y}_G(\mathbf{x})$ 均只包含 m 个不全为零的稀疏向量, 因此, $\beta \tilde{\mathbf{J}}^{-1}$ 与 $\mathbf{Y}_G(\mathbf{x}) \mathbf{Y}_N^{-1}$ 的计算各自均具有 m 个空间并行度。

5) 依据式(46), $\tilde{\mathbf{p}}_k$ 与 $\tilde{\mathbf{q}}_k$ 的计算可以同时进行; 在并行求解出 $\mathbf{J}_d^{-1} \mathbf{p}_k$ 、 $\mathbf{q}_k^T \mathbf{J}_d^{-1}$ 的基础上, $\tilde{\mathbf{p}}_k$ 与 $\tilde{\mathbf{q}}_k$ 的计算主要涉及简单的矩阵-向量计算;

6) 依据式(40), $\Delta \mathbf{z}$ 的求解亦主要涉及简单的矩阵-向量计算;

概括起来, 上述算法将原始雅可比矩阵 $\mathbf{J}(\mathbf{x}, \mathbf{V})$ 的稀疏三角分解转换成对 m 个子矩阵 $\tilde{\mathbf{J}}_j$ 以及一个 $2n$ 维的矩阵 \mathbf{Y}_N 的稀疏三角分解, 具有很好的空间

并行性, 而 \mathbf{Y}_N 是一个常系数矩阵, 只是在故障期间以及故障切除后分别需要对其进行一次性稀疏三角分解。

3 算例仿真与分析

3.1 算法的基本性能测试

本文采用 10 机 39 节点的新英格兰系统对本文所提出的并行算法进行分析验证, 新英格兰系统接线如图 1 所示, 系统基准功率为 100 MVA, 基准电压为 345 kV。暂态仿真中, 所有发电机采用三阶模型; 假设在某一母线处发生三相短路故障, 经 0.1 s 后切除故障; 隐式梯形积分步长为 0.01 s, 总仿真时间为 1.5 s。

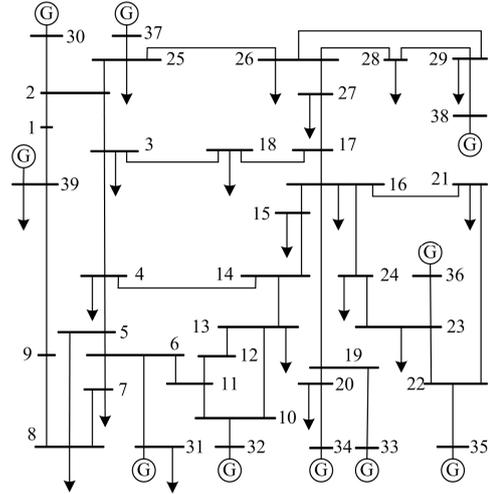


图 1 10 机 39 节点新英格兰系统接线图

Fig. 1 Wiring diagram of the 10-machine 39-node New England system

首先针对雅可比矩阵块的特性, 本文描述出了按发电机节点编号前后的矩阵元素分布图, 如图 2 所示。从图中不难发现: 排序后的雅可比矩阵元素分布呈现对角加边的形式(图2(b)中的红色标记部分即为对角部分)。在此基础上, 利用矩阵分裂技术得到雅可比矩阵的分裂形式。

首先针对雅可比矩阵块的特性, 本文描述出了按发电机节点编号前后的矩阵元素分布图, 如图 2 所示。从图中不难发现: 排序后的雅可比矩阵元素分布呈现对角加边的形式(图2(b)中的红色标记部分即为对角部分)。在此基础上, 利用矩阵分裂技术得到雅可比矩阵的分裂形式。

图 3 中为步长分别取 $h=0.01$ s、 0.02 s、 0.03 s、 0.04 s、 0.025 s 时, 利用本文方法与标准机电暂态仿真软件 PSASP 计算结果的最大相对功角误差曲线对比图。

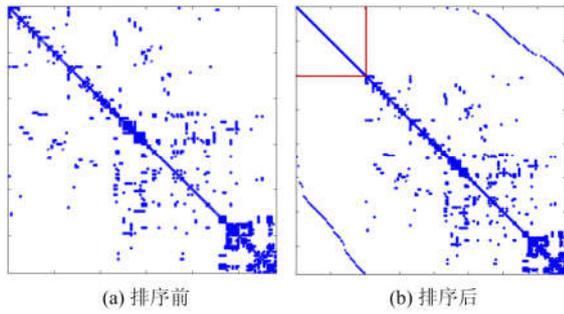


图 2 按发电机节点排序前后的雅克比矩阵元素分布图
Fig. 2 Elements distribution of Jacobian matrix before and after generator nodes sorting

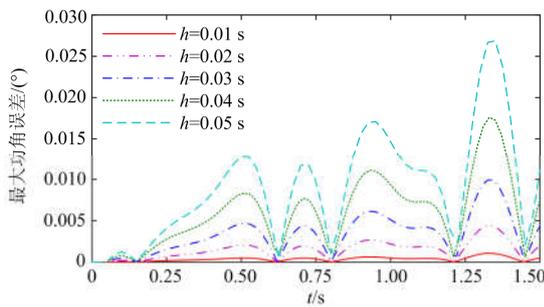


图 3 采用本文方法与 PSASP 软件结果误差曲线对比图
Fig. 3 Error curve comparison chart of the proposed method and PSASP simulation

从图 3 中可以看出：本文算法与标准机电暂态仿真软件 PSASP 的最大相对功角误差在步长为 0.05 s 时大约为 0.028°，而在步长为 0.01 s 时，最大相对功角误差不超过 0.001°；而最大相对功角误差随着步长的增大而增大，概括起来而言，本文所提的方法可以获得良好的计算精度，这为后续的并行计算奠定了基础。

3.2 并行性能测试

从本文所提出的矩阵分块递归求逆计算公式可以看出：该算法主要涉及矢量计算，这比较适合单指令流多数据流 (Single-Instruction-Multiple-Data, SIMD) 处理器^[20]—GPU。本文将新英格兰通过节点互联起来，从而得到更大规模的测试系统，具体如表 1 所示。对所提出的并行算法的性能进行测试，本文将多个 GPU 卡 (型号为 NVIDIA Tesla K10) 插入 CPU 计算机中的 PCIe 插槽，构建了一个简易的 GPUs 计算机。所用的 CPU 型号为 Intel Core i3 550，其主频为 3.20 GHz。

表 2 给出了第一次求解过程中雅克比矩阵在 CPU 和单个 GPU 上的计算时间。从表 2 中可以看出：随着系统规模的扩大，GPU 的加速效果越明显，当雅克比矩阵的维数达到 1 万多时，加速比达到 10.19，而当雅克比矩阵的维数较低时，其数据在

CPU 和 GPU 之间的传递耗费了大部分时间，进而加速效果不明显。

表 1 测试系规模

Table 1 Scale of test system

系统规模	发电机数目	母线数目	微分代数方程维数	雅克比矩阵维数
1	10	39	107	87×87
2	20	78	184	174×174
4	40	156	428	348×348
8	80	312	856	696×696
16	160	624	1 712	1 392×1 392
32	320	1 248	3 424	2 784×2 784
64	640	2 496	3 848	5 568×5 568
128	1 280	4 992	13 696	11 136×11 136
256	2 560	9 884	27 392	22 272×22 272

表 2 CPU 和 GPU 的测试结果对比

Table 2 Comparison of test results on CPU and GPU

系统规模	CPU	GPU	加速比
1	0.070 s	0.118 s	0.59
2	0.131 s	0.121 s	1.08
4	0.253 s	0.126 s	2.01
8	0.501 s	0.136 s	3.68
16	1.013 s	0.201 s	5.04
32	2.103 s	0.291 s	7.23
64	4.203 s	0.507 s	8.29
128	8.156 s	0.800 s	10.19
256	16.447 s	1.575 s	10.44

进一步，本文将单个 GPU 的测试扩展到多个 GPU 上，得到如表 3 所示的测试结果。随着 GPU 的成倍数的增加，其计算时间也相应地减少，其减小比例也称为扩展因子。扩展因子反映了算法的可移植性，图 4 给出了扩展因子在不同系统规模下柱状图。不难看出：除了系统 1×IEEE-39 外，其他系统的扩展因子基本上和使用的 GPU 个数大致相同，这就印证了本文算法的良好扩展性。

表 3 多 GPU 测试结果

Table 3 Multi-GPUs test results

系统规模	1 个 GPU	2 个 GPU	4 个 GPU	8 个 GPU
1	0.439 s	0.264 s	0.223 s	0.123 s
2	1.122 s	0.488 s	0.247 s	0.135 s
4	1.125 s	0.557 s	0.280 s	0.148 s
8	1.404 s	0.764 s	0.386 s	0.188 s
16	3.143 s	1.600 s	0.870 s	0.433 s
32	4.617 s	2.379 s	1.296 s	0.627 s
64	20.252 s	10.207 s	5.843 s	2.803 s
128	132.938 s	67.005 s	34.242 s	17.061 s
256	1 237.990 s	616.534 s	310.769 s	156.889 s

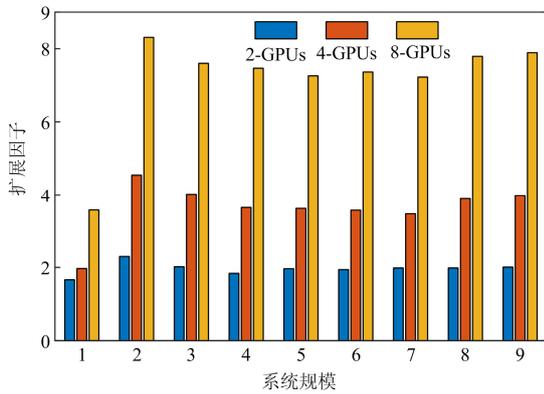


图 4 不同系统规模下的扩展因子

Fig. 4 Scaling factor for different systems

4 结论

本文在隐式梯形积分求解法基础上，基于扩展的 Sherman-Morrison 求逆公式推导出了基于矩阵分块递归求逆的机电暂态并行算法，在 GPUs 上的测试结果表明：

1) 本文所提出的并行算法是在隐式梯形积分法的基础上推导出来的；在推导过程中没有采用任何近似。因此，该算法保持了良好的收敛性。

2) 本文所提算法涉及大量的向量运算，比较适合 multi-GPUs 计算，而且在 multi-GPUs 计算上具有很好的移植性。

本文所提并行算法对互联大电网的机电暂态仿真，可以获得较为满意的加速比，适应了现代互联大电网机电暂态稳定仿真实时分析计算的需求。

参考文献

- [1] YAN Cai, ZHOU Linli, YAO Wei, et al. Probabilistic small signal stability analysis of power system with wind power and photovoltaic power based on probability collocation method[J]. *Global Energy Interconnection*, 2019, 2(1): 19-28.
- [2] LIU K, LIAO X, LI Y. Parallel simulation of power systems transient stability based on implicit Runge-Kutta methods and W-transformation[J]. *Electric Power Components and Systems*, 2017, 45(20): 2246-2256.
- [3] SHI J, SULLIVAN B, MAZZOLA M, et al. A relaxation-based network decomposition algorithm for parallel transient stability simulation with improved convergence[J]. *IEEE Transactions on Parallel & Distributed Systems*, 2018, 29(3): 496-511.
- [4] 蔡晖, 杨健, 彭竹羿, 等. 基于 PSS/E 的 MMC-UPFC 机电暂态仿真方法[J]. *电力系统保护与控制*, 2018, 46(19): 123-131.
- [5] 潘明帅, 汪芳宗, 宋墩文, 等. 基于广义向后差分方法的电力系统暂态稳定性快速数值计算方法[J]. *电力系统保护与控制*, 2018, 46(1): 9-15.
- [6] PAN Mingshuai, WANG Fangzong, SONG Dunwen, et al. Fast simulation algorithm for power system transient stability based on a new predictor-corrector strategy[J]. *Power System Protection and Control*, 2018, 46(1): 9-15.
- [7] 赵文恺, 房鑫炎, 严正. 电力系统并行计算的嵌套分块对角加边形式划分解法[J]. *中国电机工程学报*, 2010, 30(25): 66-73.
- [8] ZHAO Wenkai, FANG Xinyan, YAN Zheng. Nested BBDF partitioning algorithm in power system parallel computation[J]. *Proceedings of the CSEE*, 2010, 30(25): 66-73.
- [9] TOMIM M A, MARTI J R, FILHO J A P. Parallel transient stability simulation based on multi-area Thévenin equivalents[J]. *IEEE Transactions on Smart Grid*, 2017, 8(3): 1366-1377.
- [10] 江涵. 大规模电力系统暂态稳定并行计算研究[D]. 杭州: 浙江大学, 2012.
- [11] JIANG Han. Transient stability parallel computing for large-scale power systems[D]. Hangzhou: Zhejiang University, 2012.
- [12] 汪芳宗. 大规模电力系统暂态稳定性数值计算方法[M]. 北京: 科学出版社, 2013.
- [13] 温柏坚, 胡佳怡, 郭文鑫, 等. 基于辛 Gauss 方法及预处理 GMRES 方法的暂态稳定性并行计算[J]. *电力系统保护与控制*, 2012, 40(22): 19-24.
- [14] WEN Baijian, HU Jiayi, GUO Wenxin, et al. Parallel computation of power system transient stability based on symplectic Gauss algorithm and preconditioned GMRES method[J]. *Power System Protection and Control*, 2012, 40(22): 19-24.
- [15] 张海波, 朱存浩. 考虑时空及应用维度的分布式并行计算管理机制[J]. *电力系统自动化*, 2016, 40(15): 106-112.
- [16] ZHAO Haibo, ZHU Cunhao. Distributed parallel calculation management mechanism considering time-space and application dimensions[J]. *Automation of Electric Power Systems*, 2016, 40(15): 106-112.
- [17] 廖小兵, 汪芳宗, 杨萌. 基于高斯方法及 Sherman-

- Morrison 公式的暂态稳定性并行计算方法[J]. 电力系统保护与控制, 2017, 45(4): 1-8.
- LIAO Xiaobing, WANG Fangzong, YANG Meng. Parallel algorithm for transient stability simulation using Gauss method and Sherman-Morrison formula[J]. Power System Protection and Control, 2017, 45(4): 1-8.
- [13] 徐得超, 赵敏, 江涵, 等. 用于大规模电力系统机电暂态实时仿真的并行网络拓扑算法[J]. 高电压技术, 2016, 42(1): 296-302.
- XU Dechao, ZHAO Min, JIANG Han, et al. Power system parallel network topology algorithm applied to electromechanical transient real-time simulation of large-scale power system[J]. High Voltage Engineering, 2016, 42(1): 296-302.
- [14] 汪芳宗. 基于高度并行松弛牛顿方法的暂态稳定性实时分析计算的并行算法[J]. 中国电机工程学报, 1999, 19(11): 14-18.
- WANG Fangzong. Parallel algorithm of highly parallel relaxed Newton method for real-time simulation of transient stability[J]. Proceedings of the CSEE, 1999, 19(11): 14-18.
- [15] OWENS J D, HOUSTON M, LUEBKE D, et al. GPU computing[J]. Proceedings of the IEEE, 2008, 96(5): 879-899.
- [16] ARAUJO Igor, VINCENT T, DIEGO C, et al. Simultaneous parallel power flow calculations using hybrid CPU-GPU approach[J]. International Journal of Electrical Power & Energy Systems, 2019, 105: 229-236.
- [17] ZHOU Z, DINAVAHI V. Parallel massive-thread electromagnetic transient simulation on GPU[J]. IEEE Transactions on Power Delivery, 2014, 29(3): 1045-1053.
- [18] LIAO X, WANG F. Parallel computation of transient stability using symplectic Gauss method and GPU[J]. IET Generation, Transmission & Distribution, 2016, 10(15): 3727-3735.
- [19] ZHOU G, ZHANG X, LANG Y, et al. A novel GPU-accelerated strategy for contingency screening of static security analysis[J]. International Journal of Electrical Power & Energy Systems, 2016, 83: 33-39.
- [20] 白洪涛. 基于 GPU 的高性能并行算法研究[D]. 长春: 吉林大学, 2010.
- [21] 豆朋, 马明, 刘东, 等. 基于混合 LU 分解预处理 GMRES 的暂态稳定仿真并行算法[J]. 高电压技术, 2017, 43(10): 3419-3426.
- DOU Peng, MA Ming, LIU Dong, et al. Parallel algorithm for transient stability simulation based on GMRES method preconditioned by hybrid LU decomposition[J]. High Voltage Engineering, 2017, 43(10): 3419-3426.
- [22] BRU R, CERDÁN J, MARÍN J, et al. Preconditioning sparse nonsymmetric linear systems with the Sherman-Morrison formula[J]. SIAM Journal on Scientific Computing, 2003, 25(2): 701-715.

收稿日期: 2019-02-17; 修回日期: 2019-05-26

作者简介:

黄远洋(1986—), 男, 工程师, 研究方向为电力系统稳定与自动化工程管理。E-mail: lxbctgu@163.com

(编辑 周金梅)