

文章编号: 1000-7598 (2014) 09-2717-08

基于 OpenMP 的二维有限元-离散元并行分析方法

严成增, 郑 宏, 孙冠华, 葛修润

(中国科学院武汉岩土力学研究所 岩土力学与工程国家重点实验室, 武汉 430071)

摘 要: Munjiza 提出的有限元-离散元耦合分析方法 (FEDM) 是分析岩石破裂过程的一种十分有效的方法。然而, 为了克服网格依赖性, 需要将岩体剖分成非常细小的三角形单元, 且三角形单元之间不共用节点, 导致问题的变量数目巨大, 计算非常耗时。为了提高计算效率, 基于 OpenMP(open multi-processing)多核并行技术实现了有限元-离散元法的并行化, 克服了并行化过程中存在的数据竞争, 实现了并程序的负载均衡。提出了一套将串行程序并行化的策略, 即首先确定串行程序的热点区域, 然后尽可能地将热点区域并行化, 尽量使用私有变量来规避数据竞争; 若各线程间仍存在数据竞争, 可采用动态链表数据结构, 先将数据存于动态链表中, 最后在并行区域外, 将存于各个动态链表中的数据进行合并, 这样可以规避数据竞争, 同时避免了使用临界区或锁, 从而提高了程序的并行化效果。开发了并行版本 FDEM 程序, 将所提出的方法用于大规模工程问题的求解, 最后通过陡崖塌落的算例说明该方法的有效性。

关 键 词: 有限元-离散元法; 并行计算; 负载均衡

中图分类号: O 241

文献标识码: A

Parallel analysis of two-dimensional finite-discrete element method based on OpenMP

YAN Cheng-zeng, ZHENG Hong, SUN Guan-hua, GE Xiu-run

(State Key Laboratory of Geomechanics and Geotechnical Engineering, Institute of Rock and Soil Mechanics,
Chinese Academy of Sciences, Wuhan 430071, China)

Abstract: The coupling method of finite- discrete element method(FDEM) by Munjiza is an effective procedure for simulating rock failure. In order to overcome the mesh dependency, it is necessary to partition the rock mass into very small triangular elements; and the elements do not share the nodes, creating a vast number of variables, therefore, the calculation is very time consuming. The method overcomes competition among the existing data in the parallelization and balances loads in a parallel program. A set of strategies for parallelizing serial programs are proposed, determining the hot zones of the serial program, then parallelizing the hot zones as far as possible and using as many private variable as possible to circumvent competition. If there are still data competition among the treads, dynamic lists can be used to store these competitive data. Finally, the data stored in the various dynamic lists are merged outside the parallel zones. Data competition is accordingly avoided with no use of critical section or lock; and the effect of the program parallelization is hence increased. By the proposed method, the parallel version of the FDEM program is developed, which can be used to solve large-scale problem. A numerical example about cliffs collapse shows the effectiveness of the proposed method.

Key words: finite-discrete element method(FDEM); parallel computing; load balancing

1 引 言

岩土工程领域存在许多大规模的工程问题, 对于这些问题, 采用一般的串行程序求解, 其计算耗时让人难以接受。为此, 不少学者对岩土领域的一些数值方法的并行化实施进行了研究。以有限元方

法为例, 刘耀儒等^[1]基于 J-PCG 方法 (Jacobi 预处理共轭梯度法), 建立了有限元 EBE (element-by-element) 方法在分布存储并行机上的计算算法; 张友良等^[2]采用区域分解的分而治之的并行策略实现了岩土工程百万以上自由度的有限元并行计算; 张永斌等^[3]基于 OpenMP (open multi-processing) 在

收稿日期: 2013-05-25

基金项目: 国家自然科学基金项目 (No. 11202223, No. 11172313); 国家重点基础研究发展计划(973)项目 (No. 2011CB013505)。

第一作者简介: 严成增, 男, 1986 年生, 博士研究生, 主要从事离散元有限元耦合、离散元、颗粒物质物理与力学、多物理场耦合、DDA、大规模高性能并行计算以及软件工程等方面的研究工作。E-mail: yanchengzheng86@gmail.com

共享存储的多核计算机上实现了岩石动力学的并行计算。在离散元领域, Zhang 等^[4]基于 GPU(graphic processing unit)实现了二维三角形离散元的并行计算, 该并行算法具有速度快、扩展性好的优点。在其他数值方法领域, 黄鹏等^[5]基于 OpenMP 实现了三维显式物质点法的并行化, 该算法的并行加速比与模型规模无关, 具有较好的扩展性。在分子动力学领域, 刘青昆等^[6]研究了多核并行技术在分子动力学中的应用; 段振华等^[7]基于 OpenMP 的 Multi-Critical 对原有分子动力学并行算法进行了优化, 取得了较好的效果。

以上的各项并行化研究, 多是针对一些成熟的数值方法, 本文则对一种新颖的数值方法的并行化进行研究。为了对岩体的变形、裂纹的萌生及扩展、最终破坏的全过程进行模拟, Munjiza^[8]提出了有限元-离散元分析方法(finite-discrete element method, 简称 FEDM), 该方法基于离散元的思想, 将连续体划分为一系列三角形单元, 相邻三角形单元的公共边上插入了起黏结作用的无厚度的节理单元, 这样三角单元之间不共用节点, 不用保证位移连续, 这就使得该方法可以很方便地模拟裂纹的萌生和扩展。由于裂纹只能沿着单元边界进行扩展, 亦即裂纹的扩展具有网格依赖性, 为了使裂纹的扩展路径与实际相符, 必须划分很密的网格, 而这将导致极大的计算量。这就使得该方法在求解大规模工程问题时受到限制, 为了解决这一问题, 本文基于 OpenMP 多核并行技术, 实现了 FDEM 的并行化, 将该方法应用于求解大规模工程问题。由于现行的计算机多是基于共享存储的多核架构, 因此, 本问题提出的并行化方法具有很高的实用价值, 在普通计算机及工作站上就可应用, 可充分利用现有的多核计算资源, 计算投入成本低。

2 OpenMP 多核并行设计

OpenMP 是作为共享存储标准出现的, 具有易于使用、支持增量并行的特点, 是为共享存储环境编写并行程序而设计的一个应用程序接口, 目前支持 OpenMP 的语言有 Fortran, C/C++。OpenMP 标准中包括一套编译指导语句及一个支持函数库^[9]。OpenMP 在并行执行程序时, 采用的是“Fork/Join”方式, 其并行执行方式如图 1 所示。

该并行模式执行代码的基本思想是: 程序开始时只有一个主线程, 程序中的串行部分都由主线程执行, 并行部分由主线程和派生线程共同执行。如图 1 所示, 只有等并行部分的区域全部结束才能执

行后续的串行区域。这就是 Fork/Join 并行执行模式。

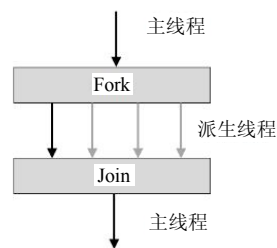


图 1 Fork/Join 并行执行模式示意图

Fig.1 Diagram for Fork/Join parallel execution mode

OpenMP 通过在源程序中加入专有的 pragma 指令来表明并行的意图, 编译器可以自动使程序并行化, 并在必要的地方加入同步互斥以及通信。同时, OpenMP 提供了很好的灵活性, 只需注释掉 pragma 指令或者将编译器选项设置为不支持 OpenMP, 并行程序即可退化为串行程序, 而不需要进行其他改动即可正常运行。

OpenMP 基于共享内存, 所有处理器都被连接到一个共享的内存单元上, 处理器在访问内存时使用的都是同一个内存地址空间。由于内存是共享的, 因此, 某一个处理器写入内存的数据可立刻被其他处理器访问到。这样就减少了通信的开销, 但也带来问题, 即数据竞争。因为 OpenMP 并行基本方法是循环分解法, 该方法要求各个循环之间必须无相关性, 数据的相关性常常会导致并行程序错误。所以, 需要采用各种措施避免读、写内存时的数据竞争, 这正是编写 OpenMP 并行程序的难点所在。

2.1 串行程序的并行化

首先需要进行热点分析, 即找出代码中 CPU 耗时最多的部分, 热点区域的串行代码是最应当将其并行化的候选代码。若这些区域恰好位于循环中, 且各循环之间无相关性, 则只需要在该循环的前部插入 #pragma omp parallel for 语句, 即可实现该循环的并行化。采用这种方法将串行部分并行化需要满足 3 个条件: ①各个循环之间必须无相关性, 否则, 将导致数据竞争使得程序运行结果错误。②每个线程需要分配足够的工作量, 否则, 若线程的工作量过小, 那么创建线程的开销可能占主导地位, 使得并行化后的程序运行时间反而更长了。③每个循环的工作量大致相同, 使所有线程全部可获得的 CPU 核心以相同的繁忙程度来工作, 亦即实现负载的均衡。

2.2 数据竞争和死锁的处理

数据竞争是由两个或更多个并行执行的线程访

问同一个内存位置引起的, 且至少其中一个线程试图写入数据到该内存位置中, 且没有使用任何同步机制。避免数据竞争的方法有: ①将全局数据变为任务或线程的局部变量, 在 OpenMP 中可以使用 `private`、`firstprivate` 以及 `lastprivate` 子句来实现; ②使用归约子句或者原子操作, 其中归约子句为 `reduction (operator:list)`, 原子操作为 `#pragma omp atomic`; ③使用锁或临界区来保护共享资源, 在 OpenMP 中设置临界区的语句为 `#pragma omp critical`, 加锁、解锁的语句分别为 `omp_set_lock()`、`omp_unset_lock()`。为了避免处理程序并行化过程中的数据竞争, 优先采用方法①、②, 但方法①、②并不能解决所有的数据竞争问题, 这时需要使用方法③, 亦即将共享资源区设置在临界区内或者对共享资源区加锁, 这样可以将共享资源的访问串行化。但方法③的同步操作的开销很大, 锁的竞争会让并行程序串行化运行, 且这种趋势随 CPU 核数增多而加剧。为此, 在方法①、②无法解决数据竞争, 同时因同步操作开销太大而不便使用方法③的情形下, 可采用方法④, 即采用调整原有代码或者改变算法, 去除原有程序中各循环的数据依赖性, 这也就是无锁编程的概念。

死锁就是当两个线程无休止地等待对方时会引起死锁, 这时两个线程都无法继续往下运行。处理死锁的一般方法是按照相同的顺序加锁的办法来处理。

2.3 负载均衡

负载均衡是指设法将所有线程在可获得的 CPU 核上以相同的繁忙程度来工作, 亦即所有线程都完成几乎等量的工作, 这样就不至于一些线程很快运行完而等待另一些线程, 造成 CPU 资源的浪费。最常见的负载不均衡的原因是划分的到各个线程上的工作粒度过大, 解决的办法是减少各个线程的任务量, 同时使得各个线程的工作量大小相近, 这主要通过调整程序的结构或者改变算法来实现。在上面的方法不奏效时, 由于 OpenMP 可以进行细粒度的并行, 这时可以采用动态调度语句 `schedule(dynamic)` 来实现并行化循环的负载均衡, 这是 OpenMP 并行较消息传递 (MPI) 并行在处理负载均衡问题上的独特优势所在。

3 FDEM 及并行的实现

3.1 基本思想

FDEM 的基本思想非常朴素, 其在连续体内部划分三角形网格, 该三角形网格与一般有限元网格

的区域别在于, 相邻三角形单元的公共边上插入了起黏结作用的无厚度节理单元, 这样各个三角形单元并不共用节点, 如图 2 所示。对每个三角形单元的节点应用牛顿第二定律, 更新速度、位移。在模拟连续体变形时, 因为起黏结作用的节理单元的存在, 可保证各个三角形单元变形的协调。同时, 可通过节理单元的断裂来模拟连裂纹的萌生、扩展, 该方法既可模拟连续体的变形, 又可对连续体到非连续体的转换以及破坏的过程进行模拟。

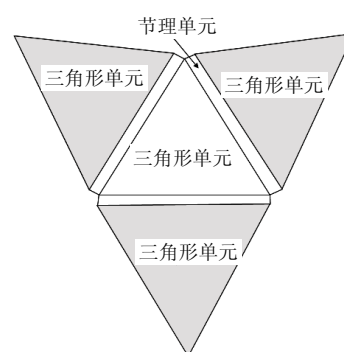


图 2 三角形单元和节理单元的连接
Fig.2 Connection of triangular elements and joint elements

FDEM 的计算流程如图 3 所示, 即先进行接触判断, 计算接触力, 再计算变形引起的节点力以及节理单元黏结作用引起的节点力, 这样就得到各个节点的不平衡力, 最后根据牛顿第二定律更新单元节点的速度和位移, 这就完成了一个时间步计算。如此循环直到达到指定的步数。同时经过串行程序的初步试算得出, 程序中各个部分的计算耗时, 如图 3 中的灰色长条, 长度越长, 表示该部分的耗时越长。从图可以看出, 最为耗时的部分也即热点区域, 为接触判断和接触力的计算, 因此这两部分应成为并行化的首选区域, 同时可以看到, 节理单元黏结作用引起的节点力计算和三角形单元变形引起的节点力计算这两部分是独立的, 可将这两部分放在不同的线程中计算, 但这种处理策略只能使用两个线程, 能获得的并行加速很有限, 且不易于实现负载均衡, 本文对这两部分采用的是将其中一个并行执行完后, 再并行执行下一个。

从图 3 中可知, 接触力判断和计算是最为耗时的, 应成为并行化需要重点关注的区域, 为此需要对原 FDEM 中的接触判断算法及接触力的计算进行仔细分析, 才有可能将这两部分较好的并行化。

3.2 接触判断

FDEM 采用了高效的 NBS 接触判断算法, 接

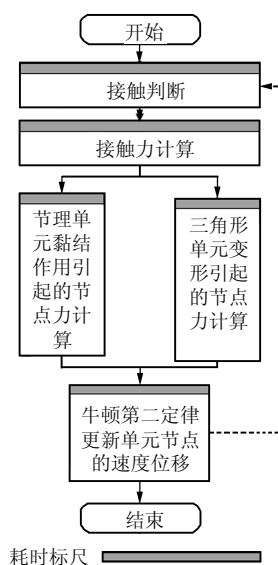


图3 FDEM求解流程及各步骤耗时
Fig.3 FDEM solving process and each time-consuming step

触判断的用时与单元数成线性关系。对于单元尺寸相差不大的系统，接触判断的效率很高。其基本思想是：先根据最大三角形单元的外接圆直径 d ，将空间求解域分解为一系列的正方形格子，再根据各个三角形单元的形心坐标将各个单元映射相应的格子中去，该过程如图4所示，图中的黑点表示三角形单元的形心，某一个单元具体位于空间中的哪一个格子，即形心坐标为 (x_c, y_c) 的映射到了位于第 n 行 m 列的格子，这可用一个整数坐标来 (m, n) 表示，公式如下：

$$\begin{aligned} m &= 1 + \text{INT}\left(\frac{x_c - x_{\min}}{d}\right) \\ n &= 1 + \text{INT}\left(\frac{y_c - y_{\min}}{d}\right) \end{aligned} \quad (1)$$

式中： $\text{INT}()$ 为向下取整函数。

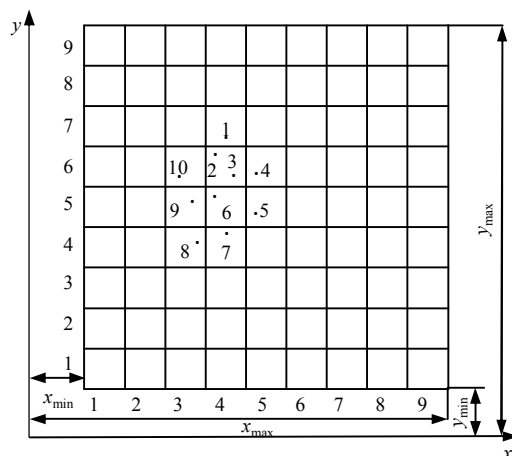


图4 单元到空间格子的映射^[8]
Fig.4 Element mapping to space lattice

经过上面的映射后，对于一个单元只需要判断与该单元所在的格子以及相邻的8个格子内的单元是否接触即可，考虑到对称性，且为消除重复判断，最终只需要对如图5所示的5各个格子内的单元进行接触判断，这样就极大地提高了接触判断的效率。

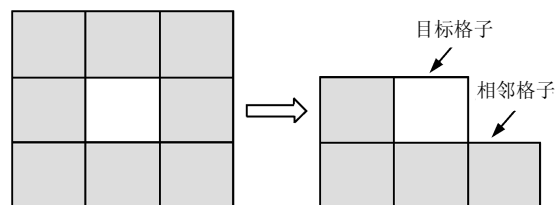


图5 三角形单元和节理单元的连接
Fig.5 Connection of triangular elements and joint elements

原有NBS算法采用的是3个整数数组来存储系统的接触，其中数组 $i1elc[\text{nelem}]$ 中元素的含义为与该单元相接触的第1个接触编号， $i1ecn[i]$ 中元素表示接触编号为 i 的下一个接触接触编号， $i1iect[i]$ 表示接触编号为 i 的接触所对应的单元号。这种处理方法对于串行程序而言，除了需要预先开辟足够大的内存空间来存储接触外，已经相当完美。但采用该方法各个单元间的接触判断存在数据共享，直接将其并行化会存在数据竞争，若采用临界区或者加锁来保护数据共享区，则因为其位于多层循环之内，会导致并行程序严重串行化，根本无法达到并行加速的目的。为此本文采用新的数据结构，将与目标单元接触的单元存在一个链表内，每个单元都有一个接触链表，其数据结构如图6所示，该链表存储与该单元相接触的单元号，由于采用的是链表数据结构，当新增接触时，可动态开辟内存来存储接触的单元号，而无需像原有方法那样需要预先开辟内存空间，且因为每个单元都拥有一个单独的接触链表，使得各个单元再进行接触判断时，可根据单元到空间格子的映射关系完全独立进行，这样可消除循环间的数据相关性，从而对接触判断部分实现无锁并行化。可以说，采用这种处理方法，有效地解决了离散元数值方法中接触判断部分的并行化，其适用于一般离散元法、MD（分子动力学）的并行化。

```
struct COUTA_struct
{
    INT eId; //与目标单元接触的单元号
    DBL d1elst[6]; //每个接触包含6个信息
    struct COUTA_struct *next;
};
```

图6 接触链表中结点的数据结构
Fig.6 Data structure of node in contact linked list

同时, 采用这种存储策略, 在单元处于紧密状态时 (在模拟连续体变形及连续体中裂纹的萌生扩展时, 单元是处于紧密状态的), 每个单元的接触链表长度大致相同, 亦即与每个目标单元相接触的单元数大致相同, 这就使得每个目标单元进行接触判断的工作量大致相同, 亦即对应于各个循环的工作量大体相同, 这很好地实现了单元接触判断并行时的负载均衡。即便单元处于松散状态, 每个链表的长度可能不同, 但因为是细粒度的并行 (此处表现为单元数比 CPU 数要多得多), 可采用 OpenMP 的动态调度语句 `schedule(dynamic)` 来实现并行化循环的负载均衡, 这也是基于 OpenMP 并行较基于消息传递 (MPI) 并行在处理负载均衡问题上的独特优势所在。

3.3 接触力的计算

FDEM 法采用了基于势的接触力计算方法, 这是该方法较一般离散元方法新颖之处。该接触力计算方法可方便地处理点点接触的问题, 不需要像常规离散元方法那样对角点进行圆角化处理。该方法接触力的计算公式为

$$\mathbf{f}_c = \oint_{\Gamma_{\beta_i \cap \beta_c}} \mathbf{n}_\Gamma (\varphi_c - \varphi_i) d\Gamma \quad (2)$$

式中: $\Gamma_{\beta_i \cap \beta_c}$ 分别为目标单元 β_c 和与之接触的单元 β_i 重叠区域的边界; 而 φ_c 、 φ_i 分别为目标单元、与之接触单元的势; $\mathbf{n}_\Gamma$ 为重叠区域边界的外法向量。

该接触力计算的细节如图 7、8 所示, 以 AB 边为例, 作用于 AB 边的分布式接触按照图 8 中的阴影区域所示, 对 AB 边的作用力采用下式计算:

$$\mathbf{f}_{c, AB} = \frac{1}{u^2} \mathbf{u} \int_0^L p \varphi(v) dv \quad (3)$$

式中: p 为罚参数; $\mathbf{u}$ 为 AB 边的法向量; φ 为势, 也即图 8 中的阴影区域, v 为 AB 边上的点在图 8

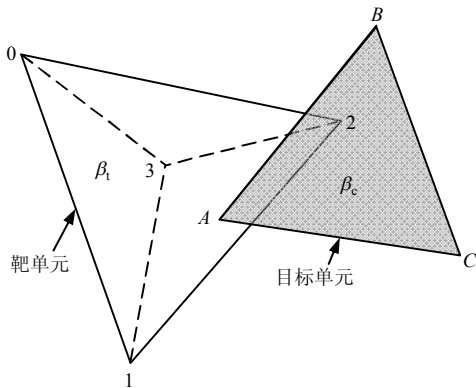


图 7 目标单元和与之接触的单元^[8]
Fig.7 Target and contact units^[8]

所示的局部坐标系 $\mathbf{u}-\mathbf{v}$ 中 $\mathbf{v}$ 方向的坐标分量; L 为 AB 边的长度。

对于重叠区域的其他各条边的接触力也采用同样的方法计算。

有关该接触力计算的详细论述, 可参考文献[10], 该文献对 Munjiza 基于势的接触力进行了详尽的论述。为了实现接触力计算的并行化, 将计算接触力最耗时的部分, 由各个线程的私有变量来处理, 从而使得这部分的计算可以完全并行化, 而在将计算处的接触力分配到各个节点上时, 采用全局数组记录节点的接触力是不可避免的, 因此会存在数据竞争, 但考虑到该共享数据的赋值耗时很短, 将该共享数据放置在单层循环内部, 这时再将共享数据区设为临界区, 并不会导致并行效率大幅下降。这样便实现了接触力计算部分的并行化。

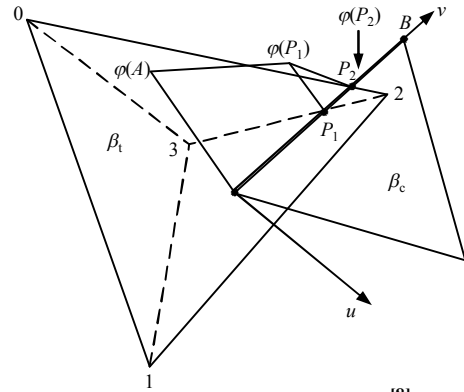


图 8 AB 边上接触力的分布^[8]
Fig.8 Contact force distribution of edge AB ^[8]

4 算例验证

为了验证本文提出的并行化方法的加速效果, 选取文献[11]的算例, 如图 9 所示为一个高为 20 m 的陡崖, 模型共划分 34 877 个单元, 陡崖在海水侵蚀下, 底面被掏空, 掏空的水平深度为 3 m。该陡崖由两组结构面控制, 两组结构面的倾角分别为 -10° 和 80° 。该陡崖岩体和节理的材料参数见表 1。

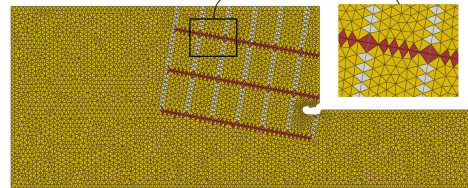


图 9 陡崖模型的几何和网格
Fig.9 Geometry and mesh of cliff models

表 1 中相应参数与图 9 的对应关系是, 图 9 中相邻的两个深色三角形单元或者浅色三角形单元之

表 1 岩体的力学参数
Table 1 Mechanical parameters of rock

岩块							结构面		
ρ	E	ν	T_s	c	ϕ	G_f	T_s	c	ϕ
$/(\text{kg/m}^3)$	$/\text{GPa}$		$/\text{kPa}$	$/\text{kPa}$	$/(^{\circ})$	$/(\text{J/m}^2)$	$/\text{kPa}$	$/\text{kPa}$	$/(^{\circ})$
2 344	1	0.25	300	460	47	10	46	10	30

注: T_s 为抗拉强度, G_f 为能量释放率。

间的节理单元位于结构面处, 这些节理单元的参数取表 1 中结构面的参数。而其余相邻的三角形单元之间的节理单元因位于完整岩块内部, 这些节理单元的强度参数取表 1 中岩块一栏中的 T_s 、 c 、 ϕ 。

本文的计算硬件环境采用本课题组的 Dell precision T5500 工作站, 内部配置为 intel 的 Xeon X5690@3.47GHZ 3.46GHZ 6 核处理器 (有 12 个逻辑内核), 使用 DDR3 SDRAM@48G 内存, 硬盘为 2TB-SCSI。分别采用串行程序和本文提出的并行化方法计算算例, 并比较两者的耗时。计算结果见图 10, 需要特别说明的是, 由于并行程序的计算结果与串行程序的计算结果完全一致, 限于篇幅仅在图 10 中给出并行程序计算出的陡崖的塌落形态随时间的演化过程, 此处仅仅是计算结果的展示, 而非并行程序和串行程序计算结果的对比。

如图 10 所示, 陡崖最先在掏空区沿着既有节理产生拉裂缝, 该裂缝沿着既有节理面 (80° 倾角) 从下往上逐步扩展, 且由于掏空区端部拉应力集中而产生拉裂纹, 如图中的 0.085、0.096、0.116 s 所示, 同时陡崖的后缘开始沿既有节理面 (80° 倾角) 产生张拉裂缝, 该裂缝由上往下逐步扩展, 如图中的 0.116、0.124 s。随着裂缝的继续发展, 被原有两组节理围限切割的块体开始发生倾倒, 由于块体倾倒, 使得在底部沿原有节理 (-10° 倾角) 产生拉张裂缝, 裂缝沿着既有节理面从左至右扩展, 直至贯通, 如图中的 0.233~0.543 s 所示, 以上的计算结果所展现的破坏模式与文献[11]中所描述的是一致的, 同时与串行程序的计算结果一致, 这验证了程序计算结果的正确。鉴于本文通过本算例的目的, 更多的是讨论并行程序的加速效果, 关于计算中陡崖塌落的过程和应力演化, 在此不再详述。

为了考证并行程序的加速效果和可扩展性, 分别研究了加速比随核心个数、并行效率随核心个数和加速比随单元数的变化关系。为了考证加速比随单元数的变化关系, 分别对上面的算例划分了 34 877、65 091、96 692 个单元。得到如图 11~13 所示的曲线。

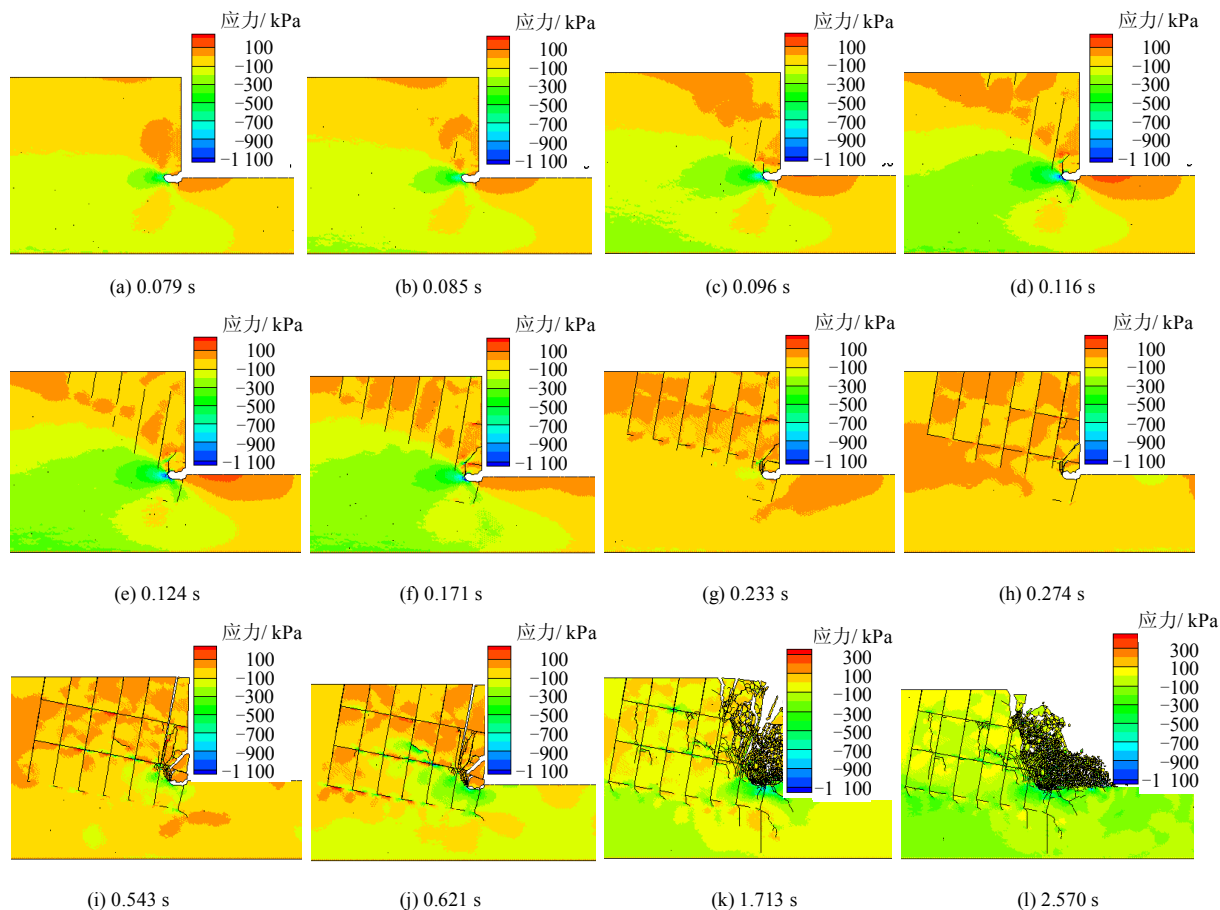


图 10 陡崖塌落及应力演化
Fig.10 Cliffs collapse and stress evolution

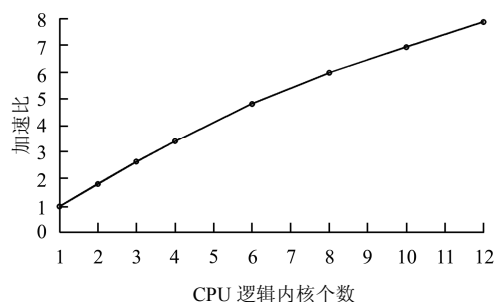


图 11 加速比随核个数的变化

Fig.11 Curve of speedup ratio and kernel number

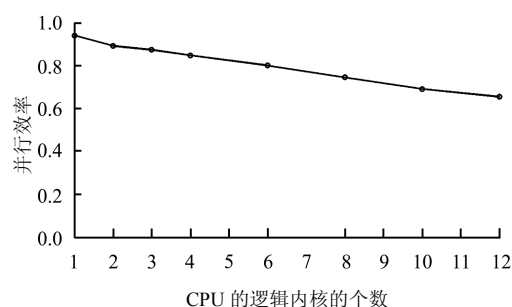


图 12 并行效率随核个数的变化

Fig.12 Curve of parallel efficiency and kernel number

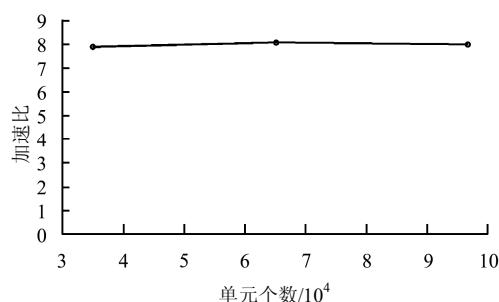


图 13 加速比随单元个数的变化

Fig.13 Curve of speedup ratio and element number

从图 11 可以看出,加速比随逻辑内核的个数呈近似线性提高,展现了该并行程序良好的扩展性。从图中可以看出,在使用 4 个逻辑内核时,较串行程序可获得 3.5 倍加速;在使用 12 个逻辑内核(只有 6 个物理内核)时,可以获得 8.0 倍加速;整个并行程序的加速效果很明显。从图 12 中可以看出,并行效率随所使用的 CPU 逻辑内核个数增加而减少,这与预期是吻合的,因为随着逻辑内核的个数的增加,相应地用于创建进程的开销也在增加,因而并行效率下降。图 13 表明,加速比几乎不随单元个数(问题的规模)变化,加速比均稳定在 8.0 左右,说明该并行程序具有很好的适用性,对于疏密程度不同的网格具有几乎相同的加速比,体现了该并行程序具有与问题规模的特点无关性。

5 结 论

基于 OpenMP 多核并行技术实现了 FDEM 的并行化,使该方法用于求解大规模工程问题变得可行。该方法克服了并行化过程中存在的数据竞争,实现了并行程序的负载平衡。同时文中给出一个陡崖塌落算例,验证了本文的并行化方法具有良好的加速比、可扩展性以及问题无关性。

(1) 提出了一套串行程序并行化的策略,即首先确定串行程序的热点区域,然后尽可能地将热点区域并行化,尽量使用私有变量来规避数据竞争;若各线程间仍存在数据竞争,可采用动态链表数据结构,即先将数据存于动态链表中,最后在并行区域外,将存于各个动态链表中的数据进行合并,这样可以规避数据竞争,同时避免了使用临界区或锁,从而提高了程序的并行化效果。

(2) 陡崖算例的计算结果与串行程序的计算结果一致,且与文献[11]描述的基本一致,验证了本文的并行程序的正确性。

(3) 加速比随着逻辑内核个数呈近似线性增加,在 4 个内核时可获得 3.5 倍的加速比,在 12 个逻辑内核(只有 6 个物理内核),可获得 8.0 倍的加速效果,并行程序具有较好的可扩展性,并行效率随着核个数增加而下降。加速比随单元个数几乎没有变化,均可得 8.0 倍的加速比,体现了该并行程序的加速比与问题的规模无关,具有较好的适用性。

参 考 文 献

- [1] 刘耀儒,周维垣,杨强,等. 三维有限元并行计算及其工程应用[J]. 岩石力学与工程学报, 2005, 24(14): 2434—2438.
LIU Yao-ru, ZHOU Wei-yuan, YANG Qiang, et al. Parallel 3D finite element analysis and its application to hydraulic engineering[J]. **Chinese Journal of Rock Mechanics and Engineering**, 2005, 24(14): 2434—2438.
- [2] 张友良,冯夏庭. 岩土工程百万以上自由度有限元并行计算[J]. 岩土力学, 2007, 28(4): 684—688.
ZHANG You-liang, FENG Xia-ting. Parallel computation of finite element model with millions of DOF in geotechnical engineering[J]. **Rock and Soil Mechanics**, 2007, 28(4): 684—688.
- [3] 张永斌,唐春安,贾敬辉,等. OpenMP 在岩石动力并行计算中的应用[J]. 武汉理工大学学报, 2013, 35(2):

- 97—101.
- ZHANG Yong-bin, TANG Chun-an, JIA Jing-hui, et al. Application of OpenMP in rock dynamic parallel computation[J]. **Journal of Wuhan University of Technology**, 2013, 35(2): 97—101.
- [4] LIN ZHANG, STEVEN F QUIGLEY, ANDREW H C CHAN. A fast scalable implementation of the two-dimensional triangular discrete element method on GPU platform[J]. **Advances in Engineering Software**, 2013, (60—61): 70—80.
- [5] 黄鹏, 张雄, 马上. 基于 OpenMP 的三维显式物质点法并行化研究[J]. 计算力学学报, 2010, 27(1): 21—27.
- HUANG Peng, ZHANG Xiong, MA Shang. Parallelization of 3D explicit material point method using OpenMP[J]. **Chinese Journal of Computational Mechanics**, 2010, 27(1): 21—27.
- [6] 刘青昆, 腾人达, 刘凤. 多核并行技术在分子动力学模拟中的应用[J]. 计算机工程与设计, 2011, 32(10): 3395—3398.
- LIU Qing-kun, TENG Ren-da, LIU Feng. Application of multi-core parallel technology in molecular dynamics simulation[J]. **Computer Engineering and Design**, 2011, 32(10): 3395—3398.
- [7] 段振华, 白明泽, 豆育升. 基于 OpenMP 的 Multi-Critical 分子动力学并行算法优化[J]. 计算机应用研究, 2012, 29(7): 2432—2434.
- DUAN Zhen-hua, BAI Ming-ze, DOU Yu-sheng. Optimization of parallel algorithm for molecular dynamics based on OpenMP and Multi-Critical[J]. **Application Research of Computers**, 2012, 29(7): 2432—2434.
- [8] MUNJIZA A. The combined finite-discrete element method[M]. London: John Wiley and Sons, Ltd., 2004: 29—32.
- [9] 周伟明. 多核计算与程序设计[M]. 武汉: 华中科技大学出版社, 2008.
- [10] 徐栋栋, 孙冠华, 郑宏. 接触处理 Munjiza 方法在数值流形法中的应用[J]. 岩土力学, 2013, 34(2): 526—534.
- XU Dong-dong, SUN Guan-hua, ZHENG Hong. Application of Munjiza's algorithm to contact treatment in numerical manifold method[J]. **Rock and Soil Mechanics**, 2013, 34(2): 526—534.
- [11] MAHABADI O K, LISJAK A, MUNJIZA A, et al. Y-Geo: A new combined finite-discrete element numerical code for geomechanical applications[J]. **International Journal of Geomechanics**, 2012, 12(6): 676—688.

城市岩土工程前沿论坛第 2 号通知

(2014 年 10 月 18~19 日, 杭州)

主办单位: 中国工程院土木、水利与建筑工程学部; 中国土木工程学会土力学及岩土工程分会; 浙江大学滨海和城市岩土工程研究中心; 浙江省城市化发展研究中心

会议背景: 由龚晓南院士发起的“城市岩土工程前沿论坛”将于 2014 年 10 月 18~19 日在杭州召开。会议将就近年来城市岩土工程的热点与难点问题, 邀请著名专家、学者及相关企事业单位领导进行深入探讨, 寻找解决有关问题的思路和方法。欢迎全国相关学科的专家、学者、科技工作者、工程技术人员和研究生踊跃参加。

会议主题: (1) 城市地下空间开发利用; (2) 城市地下交通工程; (3) 城市地下水控制; (4) 高层建筑基础工程; (5) 城市地面沉降控制; (6) 城市地质灾害防治; (7) 城市抗震; (8) 城市古建筑基础保护。

出版: 根据会议的交流情况, 会后将组织有关专家学者, 编写出版《城市岩土工程前沿论坛》。

会议地点: 浙江大学建筑工程学院安中大楼报告厅(一楼西侧)

地址: 杭州市西湖区申花路(西头)浙江大学紫金港校区

会议秘书处:

联系人(紫金港国际饭店): 王笑笑(13646828087, wangxiuxiu713@163.com)

联系人(威斯希顿大酒店): 徐丽阳(15158070256, Xuliyang1991@126.com)