

文章编号: 1000-7598-(2006) 05-0807-04

基于粗细网格的有限元并行分析方法

付朝江^{1,2}, 张 武²

(1.南昌大学 建筑工程学院, 南昌 330029; 2.上海大学 计算机工程与科学学院, 上海 200072)

摘 要: 并行计算已成为求解大规模岩土工程问题的一种强大趋势。探讨了粗细网格与预处理共轭梯度法结合的并行有限元算法。从多重网格刚度矩阵推得有效的预处理器。该算法在工作站机群上实现。用地基处理时土体强夯的数值模拟分析进行了数值测试, 对其并行性能进行了详细分析。计算结果表明: 该算法具有良好的并行加速比和效率, 是一种有效的并行算法。

关 键 词: 有限元; 并行计算; 共轭梯度; 多重网格; 岩土工程

中图分类号: TB 115

文献标识码: A

Parallel finite element analytical method based on coarse-fine mesh formulation

FU Chao-jiang^{1,2}, ZHANG Wu²

(1.School of Architectural Engineering, Nanchang University, Nanchang 330029, China;

2.School of Computer Engineering and Science, Shanghai University, Shanghai 200072, China)

Abstract: Parallel computing is becoming the strong trend of solving large scale geotechnical problems. A parallel preconditioned conjugate gradient iterative algorithm for finite element problems with coarse-mesh/fine-mesh formulation is discussed. An efficient preconditioner is derived from the multigrid stiffness matrix. The algorithm is implemented on a workstation cluster. Numerical example of solution of soils after the dynamic compaction is presented and the parallel performance is analyzed. The results show high speedup and efficiency. The algorithm is efficient for parallel computing.

Key words: finite element; parallel computing; conjugate gradient; multigrid; geotechnical engineering

1 引 言

尽管目前 CPU 的计算能力持续增加, 对大规模岩土工程问题的计算, 依然存在困难。在岩土力学计算领域如三峡、高层建筑地基分析等重大工程, 用有限元法对其进行数值计算时, 为确保数值结果的充分可靠性, 需数十万乃至上百万个单元进行模拟。但由于受计算速度及存储空间的限制, 目前, 单个 PC 机尚无法完成上述大规模的计算任务, 从而, 在某种程度上约束了岩土力学计算领域的发展。克服这个问题的一个方法就是采取多处理器计算机体系, 用并行计算来处理大规模问题。因此, 在求解岩土力学与工程问题时, 发展并行计算已成为一个重要的方向。近年来, 由于计算机网络的迅速发展, 由多台计算机通过网络能构成工作站机群。这样可组成一个低成本的并行平台。基于这个平台, 能够实现并行计算。

对大规模岩土工程问题的计算, 在这种平台上迭代求解算法如预处理共轭梯度法(PCG)^[1, 2]和多重网格法(MG), 比直接求解法能更有效地实现^[3, 4]。其中, 多级 PCG 法(HPCG)具有更好的性能^[5]。一种不同的方法是粗网格-细网格法^[6], 它被看作广义 HPCG 法。它不需要直接使用多级位移函数。本文是将粗-细网格方法应用到预处理共轭梯度算法, 研制并行求解器。该算法具有 CG 算法并行效率高和 MG 法预处理能力强的优点。利用算法在工作站机群上对岩土工程问题进行了有限元并行计算。

2 粗-细网格的有限元法

用粗、细两种不同的网格对物体进行区域分裂, 如图 1 所示。假定两种网格相互嵌套, 这样粗单元为细网格的子集, 并且在两种网格中形函数相同。按有限元方法, 粗单元内的位移场可表示为 $U^c(P) = N^c X^c$, 式中 N^c 为形函数矩阵; X^c 为节

点自由度向量。因而,在已知粗单元内的细网格节点上的位移为 $U_f^c(P) = N_f^c X^c$ 。细单元的自由度 X_f^c 可表示为 $X_f^c = H_f^c X^c$, 式中矩阵 H_f^c 由细网格节点上每个细单元的粗网格形函数的值 N_f^c 确定。

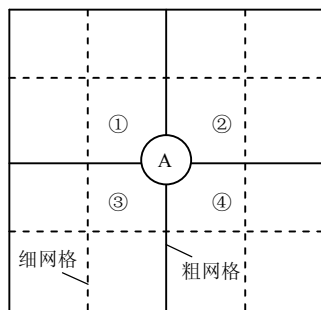


图1 细/粗网格的实例

Fig.1 An example of fine and coarse meshes

粗-细网格方法中^[6],细单元中的位移场可表示为变量的“粗”的特性和“细”的局部校正。即: $U^f(P) = N^c X^c + N^f d^f$ 。这表明问题的解可假定为:是由粗网格上粗的近似值和细网格上局部函数的修正值确定。细网格单元的节点位移向量可表示为

$$X^f = H_f^c X^c + d^f \quad (1)$$

按标准的有限元符号^[7],单元平衡方程可以通过虚位移原理对每个细单元积分得到。

$$\begin{bmatrix} K_f^c & K^{cf} \\ K^{cft} & K^f \end{bmatrix} \begin{Bmatrix} X^c \\ d^f \end{Bmatrix} = \begin{Bmatrix} H_f^{cT} F \\ F \end{Bmatrix} \quad (2)$$

令 X' 为每个细单元未知的自由度向量:

$$X' = \begin{Bmatrix} X^c \\ d^f \end{Bmatrix} \quad (3)$$

定义 T_f 为转换矩阵,且 $T_f = [H_f^c \quad I]$ 。则式(2)表示为

$$K X' = F' \quad (4)$$

$$\text{其中 } F' = T_f^T F; \quad K' = \begin{bmatrix} K_f^c & K^{cf} \\ K^{cft} & K^f \end{bmatrix} \quad (5)$$

式(4)为粗-细网格方法中,每个细单元的平衡方程。

将标准的有限元法应用到细网格,可得到通常的单元平衡方程

$$K^f X^f = F \quad (6)$$

可以看出,式(5)中的对角块 K^f 恰好为式(6)中的细单元刚度矩阵 K^f 。对角块 K_f^c 是对粗单元刚度矩阵 K^c 的相应贡献,它是由每个细单元组集而成,即

$$K^c = \sum_{f \in c} K_f^c \quad (7)$$

式(4)中的刚度矩阵 K' 是通过转换矩阵 T 直接与式(6)中的 K^f 相联系:

$$K' = T^T K^f T \quad (8)$$

下一步是在整个系统中,组集所有的单元方程式(4)。设 X'_u 为单元向量 X' 的未组集的整体向量。 X'_a 为由所有节点变量组集而成的整体向量。 X'_u 和 X'_a 的关系可用整体联系矩阵 A 表示为^[8]: $X'_u = A X'_a$ 。

$A X'_a$ 的运算表示从组集系统到局部单元分配节点变量,而 $A^T X'_u$ 的运算等价于将所有单元贡献到一个整体向量进行局部量求和。对细网格节点变量 d^f ,按标准的分裂,局部求和只涉及到直接联系节点本身的细单元(图1中对节点A的单元1~4)。对向量分量 X^c (粗网格节点变量)求和扩展到与给定粗节点相连的任一粗单元的所有细单元(图1中对节点A的所有16个细单元)。这相当于首先对每个粗单元的所有子单元的 X^c 分量进行组集,然后,在粗网格层将这些值进行局部求和。

整个离散体的平衡条件为 $K'_u X'_u = F'_u$, 这是一个单元方程式(4)的未组集系统。 K'_u 为一个单元刚度矩阵 K' 的块对角矩阵, K' 由式(5)定义,并由式(8)确定。单元方程组集为

$$K'_a X'_a = F'_a \quad (9)$$

式中 $K'_a = (T_u A)^T K'_u (T_u A)$, $F'_a = (T_u A)^T F_u$ 。在这个粗-细网格方法中,式(9)将是要求解的最终平衡方程系统,刚度矩阵用细网格单元刚度通过连接和转换矩阵表示。

3 并行预处理共轭梯度法

对对称正定问题,PCG法是一种常用的迭代算法^[9],它要求在每一次迭代时计算如下向量和变量:

$$\alpha_k = \frac{r_{a,k}^T z'_{a,k}}{P_{a,k}^T K'_a P_{a,k}} \quad (10)$$

$$r'_{a,k+1} = r'_{a,k} - \alpha_k K'_a P'_{a,k} \quad (11)$$

$$X'_{a,k+1} = X'_{a,k} + \alpha_k P'_{a,k} \quad (12)$$

$$\beta_k = \frac{r_{a,k+1}^T z'_{a,k+1}}{r_{a,k}^T z'_{a,k}} \quad (13)$$

$$P'_{a,k+1} = z'_{a,k} + \beta_k P'_{a,k} \quad (14)$$

由预处理矩阵 M_a , 从 $M_a z'_{a,k} = r'_{a,k}$ 可得到 $z'_{a,k}$ 。

初始条件为 $\mathbf{r}'_{a,0} = \mathbf{F}'_a - \mathbf{K}'_a \mathbf{X}'_{a,0}$, $\mathbf{P}'_{a,0} = \mathbf{M}_a^{-1} \mathbf{r}'_{a,0}$ 。从初始估计值 $\mathbf{X}'_{a,0}$ 开始求解, 得到一系列迭代值 $\mathbf{X}'_{a,k}$, 它将收敛于式 (9) 的解。当满足收敛准则时, 迭代过程将终止。

为提高迭代效率, 不采取任何整体刚度矩阵 $\mathbf{K}'_a$ 和整体向量, 而是通过连接矩阵将 PCG 法用单元矩阵和单元向量 (用非组集形式) 来表示。引用如下新的单元向量, 从而得到分布式计算公式^[2]:

$$\mathbf{z}'_u = \mathbf{A} \mathbf{z}'_a, \quad \mathbf{r}'_u = \mathbf{F}'_u - \mathbf{K}'_u \quad (15)$$

由此定义, 可得到

$$\mathbf{r}'_a = \mathbf{A}^T \mathbf{r}'_u \quad (16)$$

这样可得到粗-细网格方法与 PCG 迭代算法相结合的算法, 可表示为:

$$\mathbf{g}_u = \mathbf{T}_u \mathbf{P}'_{u,k} \quad (17)$$

$$\mathbf{q}_u = \mathbf{K}_u \mathbf{g}_{u,k} \quad (18)$$

$$\alpha_k = \mathbf{r}'_{u,k}^T \mathbf{z}'_{u,k} / \mathbf{g}_{u,k}^T \mathbf{q}_{u,k} \quad (19)$$

$$\mathbf{V}'_{u,k+1} = \mathbf{T}_u^T \mathbf{q}_{u,k} \quad (20)$$

$$\mathbf{r}'_{u,k+1} = \mathbf{r}'_{u,k} - \alpha_k \mathbf{V}'_{u,k} \quad (21)$$

$$\text{预处理: } \mathbf{r}'_{u,k} \Rightarrow \mathbf{z}'_{u,k} \quad (22)$$

$$\mathbf{X}'_{u,k+1} = \mathbf{X}'_{u,k} + \alpha_k \mathbf{P}'_{u,k} \quad (23)$$

$$\beta_k = \mathbf{r}'_{u,k+1}^T \mathbf{z}'_{u,k+1} / \mathbf{r}'_{u,k}^T \mathbf{z}'_{u,k} \quad (24)$$

$$\mathbf{P}'_{u,k+1} = \mathbf{z}'_{u,k} + \beta_k \mathbf{P}'_{u,k} \quad (25)$$

可以看出, 除少数几步, 如在式 (19)、式 (24) 对所有单元的最后求和, 预处理式 (22) 的单元之间通信之外, 所有计算能完全并行化, 并且仅用局部数据实现单元接单元的运算, 即用单元数据进行处理, 从而避免了单元数据的组集。

4 预处理

式 (9) 中结构刚度矩阵 $\mathbf{K}'_a$ 的形式为 $\mathbf{K}'_a = \begin{bmatrix} \mathbf{K}_a^c & \mathbf{K}_a^{cf} \\ \mathbf{K}_a^{cfT} & \mathbf{K}_a^f \end{bmatrix}$, 其中 $\mathbf{K}_a^c$ 和 $\mathbf{K}_a^f$ 分别为粗、细网格的刚度矩阵。粗-细网格法的主要思想是采用一个有效的预处理矩阵^[10], 即 $\tilde{\mathbf{K}} = \begin{bmatrix} \mathbf{K}_a^c & 0 \\ 0 & \mathbf{D}_a^f \end{bmatrix}$, 其中 $\mathbf{D}_a^f$ 为 $\mathbf{K}_a^f$ 的对角矩阵。 $\mathbf{K}_a^c$ 可直接从粗网格单元或式 (7) 和 (8) 中得到。在 CG 法中每一次迭代需要预处理, 求解 2 个非联立方程系统, 即

$$\mathbf{K}_a^c \mathbf{z}_a^c = \mathbf{r}_a^c, \quad \mathbf{D}_a^f \mathbf{z}_a^f = \mathbf{r}_a^f \quad (26)$$

由后一个方程所表示的细网格预处理得到

$$\mathbf{z}_u^f = \sum_{\text{loc}} (\mathbf{D}_u^f)^{-1} \mathbf{r}_u^f \quad (27)$$

式中 $\mathbf{D}_u^f$ 为一个对角矩阵, 定义为 $\{\text{diag}(\mathbf{D}_u^f)\} = \left\{ \sum_{\text{loc}} \text{diag}(\mathbf{K}_u^f) \right\}$, 局部求和 $\sum_{\text{loc}}$ 表示在细网格层上所包含的所有相关单元的节点量求和。

粗网格问题式 (26) 的解能用单一网格分布式 PCG 法得到, 该 PCG 法采用对角预处理子。若按收敛速度, 对角预处理子不是最合适, 但它能完全并行化并应用于单元级^[2]。对这个新迭代过程, 初始条件为 $\mathbf{r}_0^c = \sum_{f \in c} \mathbf{r}_u^c$, 即每个粗网格单元的初始残余向量 $\mathbf{r}_0^c$ 可通过所有其子单元的残余向量 $\mathbf{r}_u^c$ 求和得到。这个内迭代循环 (在粗网格层上) 的结果为 $\mathbf{z}_u^c$, 即粗网格节点的近似节点位移。粗单元向量 $\mathbf{z}_u^c$ 和式 (27) 中的细单元向量 $\mathbf{z}_u^f$ 得到预处理步在外迭代循环 (在细网格层上) 所求的解 $\mathbf{z}'_u$ 。一个粗-细网格预处理子的计算优点为对细网格的解是直接得到, 由于 $\mathbf{D}_a^f$ 为对角, 而 $\mathbf{K}_a^c$ 与原始刚度矩阵 $\mathbf{K}'_a$ 相比规模更小和更密集。

5 工程算例

基于上述粗细网格与预处理共轭梯度法结合的并行求解算法, 以某工程场地土体强夯作为计算实例, 进行有限元分析。并行计算环境是采用上海大学计算机学院的 DELL 工作站机群系统, 该机群由 5 台 DELL 工作站通过 100.0Mbps 以太网连接而成, 共有 12 个 CPU (1 台 4CPU, 2.0GHz Xeon chips, 1Mb cache 和 4 台双 CPU, 2.4GHz Xeon chips, 512KB cache), 每个节点内存 1.0 GB。使用消息传递接口 (MPI) 作为并行开发环境。操作系统为 Red Hat Linux9.0。

5.1 算例计算模型

以某工程场地土体强夯作为计算实例。落锤直径为 3 m, 高为 1 m, 在地面产生的最大作用力为 3.5 MPa。土体简化为弹性问题计算, 弹性模量为 60 MPa, 泊松比为 0.3, 内摩擦角为 30°, 凝聚力为 86 kPa。由于计算为空间体且尺寸大, 计算时按轴对称条件对模型进行简化。截面计算域为: 由锤体中心向左右各建模 5 m、深 7.5 m。采用无反射边界条件。图 2 为土体强夯的数值分析建模示意图, 用 8 节点二次等参元对土体动应力进行分析。采取两组粗细网格分别进行求解, 即 25×25 的粗网格和 50×50 的加密细网格 (表示为 50×50); 及 25×25 的粗网格和 100×100 的加密细网格 (表示为 100×100)。在工作站机群上用上述算法进行并行求解。计算时间、加速比和效率随 CPU 数目的变化分别如

图3和图4、表1所示。

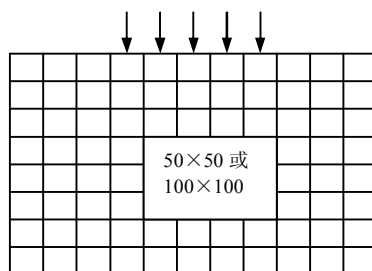


图2 数值算例示意图

Fig.2 Sketch of numerical example

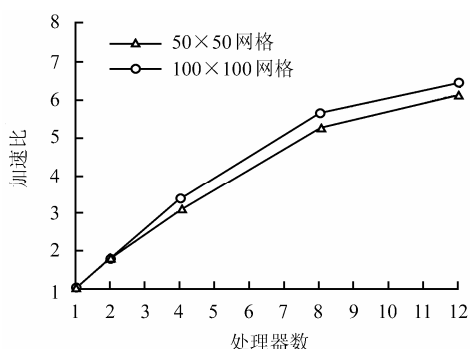


图3 并行计算加速比

Fig.3 Plot of the speedup for coarse/fine mesh

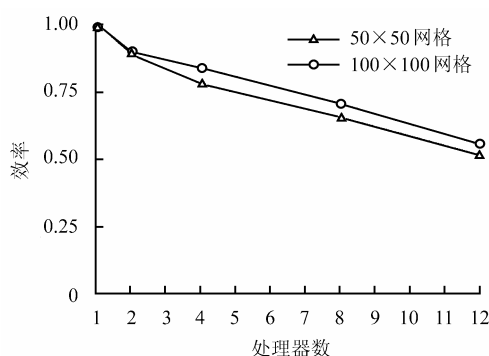


图4 并行计算效率

Fig.4 Plot of the CPU efficiency for coarse/fine mesh

表1 算例在12个处理器机群上的计算时间(单位: s)

Table 1 Timings in seconds on the 12-processor cluster (unit: s)

网格	处理器数				
	1	2	4	8	12
50×50	0.816	0.458	0.263	0.154	0.132
100×100	3.315	1.841	0.977	0.583	0.511

5.2 计算结果分析

图3和图4显示所求问题的加速比和效率。结果表明:算法具有良好的并行性能。加速比依赖于单元数和迭代次数。在计算分配给每个处理器的子区域边界节点时,各处理器之间需要进行通信,这种处理器之间的通信开销会影响算法的效率。当增加细单元数后,边界节点的相对比例减小,计算效率明显提高,如图4。因而在并行计算中,合理地

分配子区域给各处理器,实现负载平衡,减小通信开销,以提高算法的效率。

从表1可以看出,随着处理器个数增加,计算时间明显减小。当处理器个数超过4个时,由于处理器之间的通信开销增加,加速比增加明显减缓、CPU效率明显降低,如图3、图4所示。而用4个处理器时,计算是在4个处理器的节点上进行。处理器之间的通信开销很小,因而其性能很好。

6 结论

将粗-细网格方法应用到预处理共轭梯度求解算法,实现有限元并行计算。从多重网格刚度矩阵推导出预处理子。该算法在 workstation 机群上实现。用于地基处理时,土体强夯的数值模拟进行了数值测试,对并行性能进行分析,数值结果表明,该算法具有较高的并行加速比和效率,获得了满意的计算结果。对岩土工程大规模的计算问题,该算法是一种有效的有限元并行计算方法。

参考文献

- [1] 张健飞, 姜弘道. 有限元并行计算的预处理[J]. 计算力学学报, 2003, 20(4): 500—503.
ZHANG Jian-fei, JIANG Hong-dao. Preprocessing of FEM parallel computation[J]. *Chinese Journal of Computational Mechanics*, 2003, 20(4): 500—503.
- [2] Zucchini A. Finite element solution method for quadrics parallel computer[R]. Roma: Technical Report ENEA, 1996.
- [3] Wriggers P, Boersma A. A parallel algebraic multigrid solver for problems in solid mechanics discretized by finite elements[J]. *Computers & Structures*, 1998, 69: 129—137.
- [4] 罗铁祥. 多重网格区域分裂分布式计算[J]. 地球科学——中国地质大学学报, 2001, 26(3): 323—327.
LUO Tie-xiang. Distributed multigrid computation with domain decomposition[J]. *Earth Science-Journal of China University of Geosciences*, 2001, 26(3): 323—327.
- [5] Jouglard C E, Coutinho ALGA. A comparison of iterative multi-level finite element solvers[J]. *Computers & Structures*, 1998, 69: 655—670.
- [6] Criesfield M A. Finite elements and solution procedures for structural analysis[M]. Swansea: Pineridge Press, 1986.
- [7] Zienkiewicz O C. The finite element method[M]. New York: McGraw-Hill, 1991.
- [8] Gallagher R H. Finite element analysis-fundamentals[M]. New Jersey: Prentice-Hall, 1975.
- [9] Bijorck A. Least squares methods[M]. Amsterdam: North-Holland, 1990.
- [10] Zucchini A. A parallel preconditioned conjugate gradient solution method for finite element problems with coarse-fine mesh formulation[J]. *Computers & Structures*, 2000, 78: 781—787.