并行有限元计算中的接触算法[1)]

王福军 *,2)　王利萍 *　程建钢 †　姚振汉 †

*(中国农业大学水利与土木工程学院，北京 100083)
†(清华大学航空航天学院，北京 100084)

摘要　针对动态接触问题的有限元并行计算，提出了一种新的接触算法. 新算法引入局部拉氏乘子技术来计算接触力. 由于同时考虑了无穿透的接触约束条件和相邻接触对的相互影响，较之广泛使用的罚参数法，新算法使接触约束条件和系统平衡方程得到更充分的满足. 虽然为提高接触计算精度而在局部采用了迭代技术，但算法仍然具有较高的效率，且与显式时间积分方案完全相容. 此外，通过构造专门的区域分解方案，实现了将现有为串行程序开发的搜索算法平滑移植到并行环境的目标. 数值算例表明，所提出的接触算法具有很好的并行性，在保证了接触问题并行计算精度的同时，取得了满意的并行效率.

关键词　有限元法，并行计算，接触算法，显式时间积分，区域分解

中图分类号：O034, O0246　　**文献标识码**：A　　**文章编号**：0459-1879(2007)03-0422-06

引　言

工程领域存在大量动态接触问题，如旋转机械中运动部件与固定部件的摩擦、建筑物在地震或爆炸荷载作用下的破坏等. 由于这类问题往往伴随有结构大变形和材料塑性等高度非线性特征，因此，在对其进行动力学有限元分析时，采用并行环境成为必然选择 [1]. 随着并行算法的发展，在不涉及接触问题的情况下，有限元并行计算已经取得极大成功，大多数工程问题均能够以超过 85% 的并行效率快速完成动力学过程模拟和分析. 但是，由于现有的接触算法大多是为串行计算程序所设计，并不具有内在并行性，从而导致现有有限元程序在用于接触问题分析时，很难取得比较高的并行效率 [2,3].

接触算法主要包括接触搜索算法和接触力算法. 接触搜索的目的是确定整个计算域中哪些单元进入接触状态. 接触搜索过程一般分为全局搜索和局部搜索. 全局搜索每隔若干时间步进行一次，目的是粗略地找出可能发生接触的潜在接触对. 局部搜索在每个时间步都需进行，任务是确定实际接触对. 全局搜索算法的关键是算法的执行效率，局部搜索算法的关键是计算精度. 目前有多种全局搜索算法 [4,5] 和局部算法 [6~8] 可供选择. 局部搜索算法因只涉及局部少数单元之间的几何关系计算，故可直接用于并行计算，但全局搜索算法因涉及多个相邻子域公共边界上的信息交换，因此，为串行程序开发的全局搜索算法一般不能直接用于并行计算. 为此，部分学者专门开发了针对有限元并行计算的全局搜索算法 [3,9]. 但这种做法很难保证在求解不同问题时总具有高的并行效率，且算法的维护成本较高. 目前的并行计算趋向于直接使用现有串行全局搜索算法，但前提是在区域分解及单元编号等方面进行大量附加计算 [1,10].

接触力算法的实质是让构形系统满足接触边界上无穿透约束条件. 常用接触力算法主要包括拉氏乘子法 [11~13] 和罚参数法 [11,14]. 拉氏乘子法要求精确满足接触界面无穿透的约束条件，是一种准确的接触力算法，但由于引入新未知量，使方程组不再解偶，从而使其与显式时间积分方案不相容. 而动态接触问题所包含的高频成份决定了有限元分析只能采用显式时间积分方案来完成，这样，罚参数法被广泛用于接触约束. 罚参数法通过引入罚参数与界面穿透量的乘积作为接触力，使无穿透的约束条件近似得到满足. 它与显式时间积分方案完全兼容，且简单易用，但当所引入的罚参数很大时可能导致方程组病态，对罚参数的选取目前尚无规则可循，只能靠使用者的经验来决定 [2]. 虽然有一些改进的

2006–08–28 收到第 1 稿，2007–01–12 收到修改稿.
1) 国家自然科学基金 (10372114) 资助项目.
2) E-mail: wangfj@cau.edu.cn

接触力算法用于显式时间积分方案，如小球算法 [15] 和防御节点法 [16] 等，但这些算法在用于并行计算时均存在一定问题，要么是切向接触力计算精度较低，要么是计算时间过长 [17].

为此，本文在现有算法的基础上，通过对单元分类及构造专门的区域分解算法，实现了在并行有限元程序中直接使用现有各种接触搜索算法的目的. 同时，提出了一种改进的接触力计算方法，它通过局部迭代使接触约束条件得到更充分满足，明显提高了接触力计算精度. 新的接触算法具有很好的内在并行性.

1 动态接触问题的数学描述

1.1 有限元公式

根据虚位移原理，将 t 时刻的平衡条件写成等效的积分形式如下 [16]

$$\int_{{}^tV}[\delta_t\boldsymbol{e}]^{\mathrm{T}}{}^t\tau\mathrm{d}V = \int_{{}_0V}[\delta^t\boldsymbol{u}]^{\mathrm{T}}\boldsymbol{b}\mathrm{d}V+\int_{{}_0S}[\delta^t\boldsymbol{u}]^{\mathrm{T}}\boldsymbol{T}\mathrm{d}S- \int_{{}^tV}[\delta^t\boldsymbol{u}]^{\mathrm{T}}\rho^t\ddot{\boldsymbol{u}}\mathrm{d}V+\int_{{}^tS_c}\delta([{}^t\boldsymbol{u}_c]^{\mathrm{T}}{}^t\boldsymbol{f}_c)\mathrm{d}S \quad (1)$$

上式从左至右分别为内力、体力、外力、惯性力和接触力的虚功. 对整个计算域进行单元离散后，引入应变位移关系、应力应变关系，可建立下列方程

$$\boldsymbol{M}\,{}^t\ddot{\boldsymbol{u}} = {}^t\boldsymbol{Q}-{}^t\boldsymbol{F}+{}^t\boldsymbol{F}_c \quad (2)$$

采用中心差分法，可建立如下时间递推公式 [17]

$${}^{t+\Delta t}\boldsymbol{u} = \boldsymbol{M}^{-1}[\Delta t^2({}^t\boldsymbol{Q}-{}^t\boldsymbol{F}+{}^t\boldsymbol{F}_c)+ 2\boldsymbol{M}{}^t\boldsymbol{u}-\boldsymbol{M}{}^{t-\Delta t}\boldsymbol{u}] \quad (3)$$

为了计算出接触力 ${}^t\boldsymbol{F}_c$，需要先通过接触搜索算法确定接触边界 S_c，然后使用接触力算法根据接触搜索结果计算接触力.

1.2 接触搜索算法

为完成接触搜索任务，作者早期专门开发了全局搜索算法 [5] 和局部搜索算法 [6]. 在本文中，可直接使用这些算法来进行并行计算，但前提是需要对单元进行附加类型定义，并通过区域分解算法保证必要的数据通讯. 具体细节参见 2.1 节，2.2 节和文献 [10,18].

1.3 接触力算法

对于确定的接触对，需要在法线和切线两个方向上计算接触力，然后才能使方程 (3) 可解. 假定节点 A 与单元面 B 在某一时刻形成接触对，且在点 C 发生接触. 对于接触点 C 处的法向，即可在单元面 B 上定义，也可在节点 A 所在的单元面上定义. 这里，选择前一种定义，这也是大多数文献遵从的约定 [11].

现考虑节点 A 与接触点 C 的法向运动，可列出这两点沿法线方向的各自运动方程 (节点 A 的物理量用上标 I 表示，接触点 C 的物理量用上标 II 表示)

$$M^{\alpha\,t}\ddot{u}_n^{\alpha} = {}^t\hat{F}_n^{\alpha}+{}^tR_{cn}^{\alpha},\quad \alpha=\mathrm{I},\mathrm{II} \quad (4)$$

其中：下标 n 代表法线方向；M 是节点质量，$\ddot{u}_n$ 是加速度，${}^t\hat{F}_n^{\alpha}$ 是除了接触力以外的节点内力和外力矢量和；${}^tR_{cn}^{\alpha}$ 是接触力. ${}^tR_{cn}^{\alpha}$ 可分解为

$${}^tR_{cn}^{\alpha} = \bar{R}_{cn}^{\alpha}+f_n^{\alpha} \quad (5)$$

式中，f_n^{α} 是由于本接触对所产生的接触力增量，$\bar{R}_{cn}^{\alpha}$ 是其它接触对中的接触力的贡献.

由于接触点 C 是落在单元面 B 上，因此，接触点 C 的质量、速度、加速度、节点内力、接触力等，均可借助单元面 B 的各节点物理量通过形函数来表征，例如，式 (4) 中接触点 C 的加速度可表示为

$$\ddot{u}_n^{\mathrm{II}} = \sum_{k=1}^{n}N_k\ddot{u}_{nk} \quad (6)$$

其中，求和的范围 n 为单元面 B 的节点数；$N_k,\ddot{u}_{nk}$ 为节点 k 的形函数、加速度. 式 (4) 中接触点 C 的质量 M^{II}，由下式计算

$$M^{\mathrm{II}} = \sum_{k=1}^{n}m_k,\quad m_k = M_kN_k\Big/\sum_{j=1}^{n}N_j^2 \quad (7)$$

式中，m_k 是节点 k 对接触点 C 的质量贡献，M_k 是节点 k 的质量.

采用差分法积分 (4) 式，得到

$${}^{t+\Delta t}u_n^{\alpha} = \Delta t^{t-\Delta t}\dot{U}_n^{\alpha}+\frac{(\Delta l)^2}{M^{\alpha}}({}^t\hat{F}_n^{\alpha}+\bar{R}_{cn}^{\alpha}+f_n^{\alpha}) \quad (8)$$

其中，${}^{t-\Delta t}\dot{U} = \dfrac{1}{\Delta t}({}^tu-{}^{t-\Delta t}u)$. 现引入接触约束条件

$${}^{t+\Delta t}g_n = {}^tg_n+{}^{t+\Delta t}u_n^{\mathrm{II}}-{}^{t+\Delta t}u_n^{\mathrm{I}} = 0 \quad (9)$$

式中，g_n 为节点 A 与接触点 C 的距离 (或贯入量). 上式表示在时刻 $t+\Delta t$ 无穿透. 将式 (8) 代入式 (9), 并考虑节点 A 和点 C 的接触力大小相等、方向相反，有

$$f_n^{\mathrm{I}}=\frac{M^{\mathrm{I}}M^{\mathrm{II}}}{M^{\mathrm{I}}+M^{\mathrm{II}}}\Big[\frac{{}^{t}\hat{F}_n^{\mathrm{II}}+\bar{R}_{cn}^{\mathrm{II}}}{M^{\mathrm{II}}}-\frac{{}^{t}\hat{F}_n^{\mathrm{I}}+\bar{R}_{cn}^{\mathrm{I}}}{M^{\mathrm{I}}}+\frac{{}^{t-\Delta t}\dot{U}_n^{\mathrm{II}}}{\Delta t}-\frac{{}^{t-\Delta t}\dot{U}_n^{\mathrm{I}}}{\Delta t}-\frac{{}^{t}g_n}{(\Delta t)^2}\Big] \tag{10}$$

这里的 f_n^{I} 便是本接触对在节点 A 处产生的接触力增量. 接触点 C 处的接触力增量按 $f_n^{\mathrm{II}}=-f_n^{\mathrm{I}}$ 计算. 然后，将其按单元面 B 中各节点的质量贡献分配到节点上去，即

$$f_{nk}=\frac{m_k}{M^{\mathrm{II}}}f_n^{\mathrm{II}} \tag{11}$$

式中 f_{nk} 为单元面 B 各节点得到的法向接触力增量. 考虑其它接触对的影响后所最终得到的接触力，将在 2.3 节介绍.

假定节点 A 和单元面 B 在点 C 黏滞，按计算法向接触力的同样办法可得到切向接触力增量

$$f_{t\xi}^{\mathrm{I}}=\frac{M^{\mathrm{I}}M^{\mathrm{II}}}{M^{\mathrm{I}}+M^{\mathrm{II}}}\Big[\frac{{}^{t}\hat{F}_{t\xi}^{\mathrm{II}}+\bar{R}_{ct\xi}^{\mathrm{II}}}{M^{\mathrm{II}}}-\frac{{}^{t}\hat{F}_{t\xi}^{\mathrm{I}}+\bar{R}_{ct\xi}^{\mathrm{I}}}{M^{\mathrm{I}}}+\frac{{}^{t-\Delta t}\dot{U}_{t\xi}^{\mathrm{II}}}{\Delta t}-\frac{{}^{t-\Delta t}\dot{U}_{t\xi}^{\mathrm{I}}}{\Delta t}-\frac{{}^{t}g_{t\xi}}{(\Delta t)^2}\Big] \tag{12}$$

$$f_{t\eta}^{\mathrm{I}}=\frac{M^{\mathrm{I}}M^{\mathrm{II}}}{M^{\mathrm{I}}+M^{\mathrm{II}}}\Big[\frac{{}^{t}\hat{F}_{t\eta}^{\mathrm{II}}+\bar{R}_{ct\eta}^{\mathrm{II}}}{M^{\mathrm{II}}}-\frac{{}^{t}\hat{F}_{ty}^{\mathrm{I}}+\bar{R}_{ct\eta}^{\mathrm{I}}}{M^{\mathrm{I}}}+\frac{{}^{t-\Delta t}\dot{U}_{t\eta}^{\mathrm{II}}}{\Delta t}-\frac{{}^{t-\Delta t}\dot{U}_{t\eta}^{\mathrm{I}}}{\Delta t}-\frac{{}^{t}g_{t\eta}}{(\Delta t)^2}\Big] \tag{13}$$

式中，下标 ξ 和 η 分别表示接触点处两个切线方向.

现引入如下定义

$$\Phi=|f_t|-\mu|\boldsymbol{f}_n|=\sqrt{f_{t\xi}+f_{t\eta}}-\mu|\boldsymbol{f}_n| \tag{14}$$

其中，μ 为摩擦系数，可按文献 [14] 中的方法计算. 然后，我们可做如下判断：若 $\Phi<0$, 则黏滞状态成立，取 $f_{t\xi}$, $f_{t\eta}$ 为摩擦力增量；若 $\Phi\geq 0$, 节点 A 与单元 B 处于滑动状态，此时由 $f_{t\xi}$, $f_{t\eta}$ 可确定滑动方向，取 μf_n 为摩擦力增量的值.

考虑到相邻接触对之间是有相互影响的，即一个接触对内的接触力并不只与本接触对有关，它还要受到相邻接触对的影响. 因此，需要通过迭代过程计及其它接触对的影响. 迭代在一个时间步内对整个系统内的所有接触对进行. 迭代计算步骤如下：

(1) 令迭代计算次数 $j=1$;

(2) 对系统中的每个接触对，分别按式 (10)~(14) 计算法向接触力增量和摩擦力增量；

(3) 对每个单元面，在其上所有接触对循环，令 $(\bar{\boldsymbol{R}}_{cn})^{(j)}=(\bar{\boldsymbol{R}}_{cn})^{(j)}+\boldsymbol{f}_n$, $(\bar{\boldsymbol{R}}_{ct})^{(j)}=(\bar{\boldsymbol{R}}_{ct})^{(j)}+\boldsymbol{f}_t$;

(4) 对于求得的 $\bar{\boldsymbol{R}}_c^{\mathrm{II}}$, 按所在单元各节点的质量贡献分配到各节点上去. 若接触力向量 $\bar{\boldsymbol{R}}_{cn}^{\alpha}$ 中的某个值大于 0, 令其为 0, 以防接触对中出现拉应力状态；

(5) 若 $|(\boldsymbol{R}_c^{\mathrm{I}})^{(j)}-(\boldsymbol{R}_c^{\mathrm{I}})^{(j-1)}|/|(\boldsymbol{R}_c^{\mathrm{I}})^{(j)}|<\delta$ 或循环次数 j 大于某个提前规定的限制值，则结束迭代，否则 $j=j+1$ 回到第 (2) 步继续迭代.

2 接触问题的并行计算

2.1 子域及单元类型定义

并行计算总是在子域内进行的. 在本文中，每个子域由一个各面与坐标面平行的立方体来限定，如图 1 所示. 为了控制区域分解的频率，引入了子域缓冲区的概念. 缓冲区的一半落在子域之内，另一半落在子域边界之外. 图中示出了子域 6 的缓冲区. 这样，每个子域有 3 个立方体区与之相关联：子域区、内区、外区，见图 1.

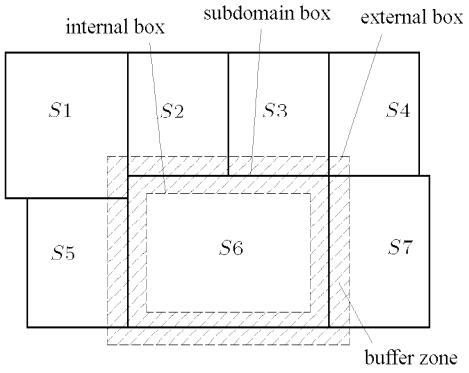

图 1 子域、缓冲区及其对应的子域区、内区和外区

Fig.1 Subdomains, buffer zones and their subdomain box, internal box and external box

缓冲区的尺寸取决于所求解问题的类型. 它对区域分解的效率及每个时间步的通讯开销有不同影响. 大的缓冲区将导致在每个子域中有更多的单元，从而将增大计算量与通讯量，但另一方面，将使必须进行区域分解操作的频率下降. 根据作者的经验，将系统中最小单元尺寸的一半，取作缓冲区的宽度是比较合适的.

为了在并行计算程序中直接使用为串行程序所开发的接触搜索算法，在一个子域中的所有单元被划分成内部单元、界面单元和外部单元 [18]. 如果一个单元完全落在子域的内区，则被定义为内部单

元；如果单元与子域缓冲区相交且其中心落在子域区之内，则被定义为界面单元；所有其它单元被定义为外部单元．对于节点也做类似定义，分别对应于内部节点和外部节点．有关细节参见文献 [18].

2.2 区域分解

区域分解的目的是将整个物理问题划分成若干子问题，分别在各节点机上进行并行计算．由于动态接触状态的大量出现，使得区域分解算法变得比较复杂．在本文中，使用了作者早期提出的双层次区域分解方案 [10]，即将区域分解分为低层次和高层次两种．在低层次的区域分解中，系统中所有子域位置随各子问题的更新而更新；而在高层次区域分解中，只有子问题本身被更新，对应子域的空间绝对位置并不更新．这样，整个系统的分区 (partitioning) 过程在高层次区域分解过程中就可省去．至于何时进行低层次或高层次的区域分解，取决于两个参数：系统中最大的累加位移与各处理器间的负载不平衡度．考虑到接触计算占有限元总计算时间的 40% 以上 [16]，因此，在划分子域时，各节点机的负载主要按接触对列表 (数目) 来估算．一般来讲，可使用全局搜索所生成的接触对列表来估算计算量．关于双层次动态区域分解的详细算法可参见文献 [10].

2.3 并行计算过程

在采用基于中心差分法求解接触问题时，在每个时间步均需进行接触计算、单元内力计算、外载计算、时间积分、构形更新等．本文采用了目前比较流行的主从模式 [19] 实现并行计算，系统中包含一个主处理器和若干个从处理器．主处理器负责操纵各种必要的串行计算，包括子域的分解及重分解工作．每个从处理器储存了相应子域上的子问题的全部数据，并进行该问题的数值模拟．由于在本文 2.1 节将单元类型进行了重新定义，因此从处理器工作时就象一个运行串行程序代码的普通处理器一样，主要的变化是多了一些通讯功能以便与其它从处理器或主处理器交换数据．

在每个时间步内，主处理器按下列过程进行并行计算：

(1) 接收来自于各从处理器所发送的最大累加位移；

(2) 根据总体最大累加位移及各处理器的负载不平衡度决定是否必须进行区域分解；

(3) 若要进行区域分解，收集来自于各从处理器的数据，然后进行区域分解，最后将新的子域数据发送到各从处理器．

在每个时间步内，从处理器按下列过程工作：

(1) 根据系统当前状态，决定是否调用接触的全局搜索算法寻找子域内的潜在接触对；

(2) 调用局部搜索算法确定实际接触对；

(3) 调用接触力算法计算子域内各接触对的接触力；

(4) 针对子域内的所有内部有限单元和界面单元，计算内力、外力；

(5) 针对子域内的所有节点，进行节点力的组集；

(6) 与其它从处理器交换界面单元与外部单元的节点力；

(7) 对子域内的内部节点进行时间积分 (即更新节点的当前位置)；

(8) 与其它从处理器交换界面单元节点的位移与速度；

(9) 向主处理器发送最大累加位移；

(10) 接收来自于主处理器的一个标志数，用以说明是否在本时间步进行区域分解；

(11) 如果要进行区域分解，向主处理器发送必要的数据，并接收在区域分解之后由主处理器发送过来的重新分布的数据．

3 数值算例

为了评价本文所提出的接触算法的并行效果，本文对一吉普车车架与刚性墙相撞的动态过程进行了并行计算．车架整体几何形状见文献 [17]，车架前端局部网格如图 2 所示．前端横梁中部集中质量为 900 kg, 用以模拟发动机；车架后端集中质量为 400 kg, 用以模拟乘员及其它重量．整个车架共划分 7 208 个 4 节点单元，系统总自由度为 35 950．并行计算环境为 PC 机群，运行 LINUX 操作系统，有限

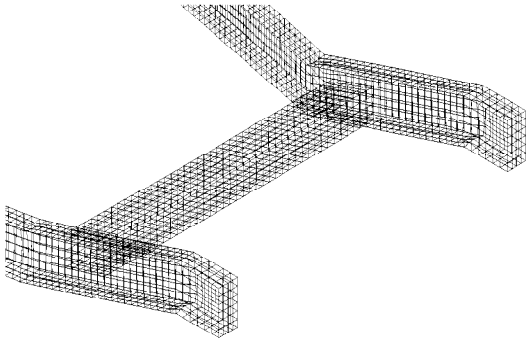

图 2　车架首部的有限元网格模型

Fig.2　The mesh model of vehicle front part

元程序采用基于 MPI 的软件包 MPICH 作为消息传递平台. 图 3 给出了车架前端在碰撞开始后 65.8 ms 时构形, 可以明显地看出车架前端发生的接触情况, 与实验结果 [17] 吻合. 表 1 给出了采用 4 个处理器进行计算时, 并行效率随着碰撞时程的变化情况, 从中可以看出, 附着碰撞发生过程中接触面积的不断增大, 并行效率是不断提高的.

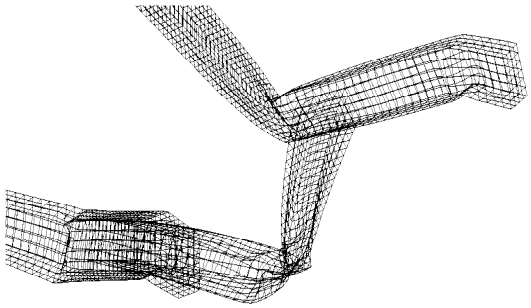

图 3 车架首部在碰撞 65.8 ms 后的变形情况

Fig.3 The deformation of vehicle front part at time 65.8 ms

表 1 并行效率与碰撞时程的关系 (采用 4 个节点机计算)

Table 1 Parallel-execution efficiency with time elapse of impact

Elapsed time for impact/s	10	20	30	40	50	60	70
Parallel – execution efficiency/%	68.4	69.2	70.1	72.2	75.8	79.3	84.4

为了评价该接触算法对大规模接触问题的适用性, 采用该算法对文献 [17] 给出的一个由 65 051 个单元组成、总自由度为 323 045 的汽车模型进行了耐撞性分析. 图 4 给出了分别采用 1~8 个处理器进行并行计算时的并行效率. 可看出, 即使在 8 个处理器下, 该系统仍具有较高的并行效率, 超过了 85%.

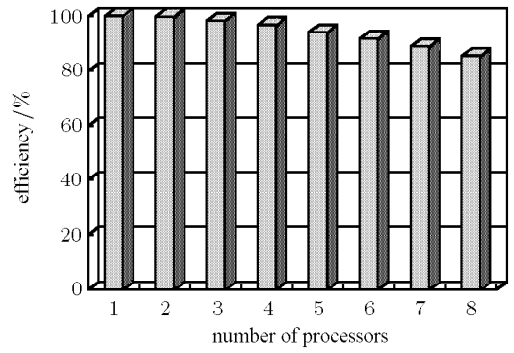

图 4 不同节点机数目时的并行效率

Fig.4 Parallel-excution efficiency when executing on different processors

4 结 论

本文提出了一种可用于有限元并行计算的接触算法. 该算法具有以下特点:

(1) 新的接触力算法基于局部拉氏乘子法, 虽然是一种局部迭代算法, 但与显式有限元时间积分方案完全相容. 在提供较罚参数法更高计算精度的同时, 也不失高效性.

(2) 通过在区域分解算法中对位于子域界面的单元进行特殊分类处理, 使得现有为串行程序所开发的接触全局搜索算法和局部搜索算法, 不需任何修改便可直接用于并行计算.

(3) 新的接触算法非常适合于在并行环境下分析动态接触问题. 虽然程序在并行机群环境下调试通过, 但算法也完全适合于 MPP 等其它并行环境.

参 考 文 献

1 Har J, Futon RE. A parallel finite element procedure for contact-impact problems. *Engineering with Computers*. 2003, 19(2-3): 67~84

2 Malone JG, Johnson NL. A parallel finite-element contact/impact algorithm for nonlinear explicit transient analysis, Part1: The search algorithm and contact mechanics. *International Journal for Numerical Methods in Engineering*, 1994, 37(4): 559~590

3 Attaway SW, Hendrickson BA, Plimpton SJ. Parallel contact detection algorithm for transient solid dynamics simulations using PRONTO3D. *Computational Mechanics*, 1998, 22(2): 143~159

4 Feng YT, Owen DRJ. An augmented spatial digital tree algorithm for contact detection in computational mechanics. *International Journal for Numerical Methods in Engineering*, 2002, 55: 159~176

5 Wang FJ, Cheng JG, Yao ZH. A contact searching algorithm for finite element analysis of contact-impact problems. *Acta Mechanica Sinica*, 2000, 16(4): 374~382

6 Wang FJ, Cheng JG, Yao ZH. FFS contact searching algorithm for dynamic finite element analysis. *International Journal for Numerical Methods in Engineering*, 2001, 52(7): 655~672

7 Belytschko T, Daniel WJT, Ventura G. A monolithic smoothing-gap algorithm for contact-impact based on the signed distance function. *International Journal for Numerical Methods in Engineering*, 2002, 55: 101~125

8 Cirak F, West M. Decomposition contact response (DCR) for explicit finite element dynamics. *International Journal for Numerical Methods in Engineering*, 2005, 64: 1078~1110

9 Oishi A, Yoshimura S, Yagawa G. Domain decomposition based parallel contact algorithm and its implementation to

explicit finite element analysis. *JSME International Journal, Series A: Solid Mechanics and Material Engineering*, 2002, 45(2): 123∼130

10 Wang FJ, Feng YT, Owen DRJ. Parallelisation for finite-discrete element analysis on distributed-memory environment. *International Journal of Computational Engineering Science*, 2004, 5(1): 1∼23

11 Hallquist JO, Goudreau GL, Benson DJ. Sliding interfaces with contact-impact in large-scale Lagrangian computations. *International Journal for Numerical Methods in Engineering*, 1985, 51: 107∼137

12 Hughes TJR, Taylor RL, Sackman JL, et al. A finite element method for a class of contact-impact problems. *Computer Methods in Applied Mechanics and Engineering*, 1976, 8: 249∼276

13 Carpenter NJ, Taylor RL, Katona MG. Lagrange constraints for transient finite-element surface-contact. *International Journal for Numerical Methods in Engineering*, 1991, 32: 103∼128

14 Wriggers P, Vu VT, Stein E. Finite-element formulation of large deformation impact-contact problems with friction. *Computers and Structures*, 1990, 37: 319∼331

15 Belytschko T, Neal MO. Contact-impact by the pinball algorithm with penalty and Langrangian methods. *International Journal for Numerical Methods in Engineering*, 1991, 31: 547∼572

16 Zhong ZH. Finite Element Procedures for Contact-Impact Problems. Oxford: Oxford University Press, 1993

17 王福军. 冲击接触问题有限元法并行计算及其工程应用. [博士论文]. 北京：清华大学， 2000 (Wang Fujun. Parallel computation of contact-impact problems with FEM and its engineering application. [Ph D thesis]. Beijing: Tsinghua University, 2000 (in Chinese))

18 Wang FJ, Feng YT, Owen DRJ. Interprocessor communication schemes in parallel finite-discrete element analysis on PC cluster. *Engineering Computations*, 2003, 20(8): 1065∼1084

19 Krysl P, Belytschko T. Object-oriented parallelization of explicit structural dynamics with PVM. *Computers and Structures*, 1998, 66(2-3): 259∼273

A CONTACT ALGORITHM FOR PARALLEL COMPUTATION OF FEM [1)]

Wang Fujun*,[2)]　Wang Liping*　Cheng Jiangang†　Yao Zhenhan†

*(*College of Water Conservancy & Civil Engineering, China Agricultural University, Beijing,* 100083, *China*)

†(*School of Aerospace, Tsinghua University, Beijing* 100084, *China*)

Abstract　A general contact algorithm for parallel finite element simulation of dynamic contact problem is presented in this paper. The local Lagrange multiplier approach is used in the new contact algorithm. Since the impenetrability condition and the interaction of adjacent contact pairs are all considered in this algorithm, the contact constraint and equilibrium equations of the system are better satisfied as compared with the traditional penalty method. Although some local iterations have to be done, the algorithm has a high computational efficiency, and is totally consistent with the explicit time integration. In addition, the existing contact search algorithms developed for serial finite element programs could be transfered into the parallel finite element program without any modification by the design of a special domain decomposition scheme. Numerical examples show that the new contact algorithm is effective for parallel finite element computations. The new algorithm ensures not only a good simulation accuracy, but also a high parallel-execution efficiency.

Key words　finite element method, parallel computation, contact algorithm, explicit time integration, domain decomposition

Received 28 August 2006, revised 12 January 2007.

1) The project supported by the National Natural Science Foundation of China (10372114).

2) E-mail: wangfj@cau.edu.cn