

弹性接触问题参数变分原理 的有限元并行算法*

胡 宁 张汝清

(重庆大学工程力学系, 1990年11月26日)

摘 要

本文基于弹性接触问题的参数变分原理的有限元解法, 利用并行计算机的特性和并行处理结构, 建立了相应的并行算法。该算法从刚度阵的生成和组集, 静凝聚过程, 求应力过程等多方面实现了并行化。该算法在西安交通大学 ELXSI-6400 并行计算机上程序实现, 计算结果表明能有效地节省计算时间, 是一种分析接触问题的有效的并行算法。

关键词 并行计算 接触分析

一、引 言

并行计算是随着并行机的产生而发展起来的, 它是目前对大型和超大型问题进行分析的有力工具。并行机作为计算机发展史上的一个里程碑, 从根本上克服了串行机单指令流和单数据流的缺陷, 发展出了多数据多指令流 (MIMD) 和数组向量运算等不同类型用途广泛的运行结构形式, 从而使运算速度有了惊人的提高。当然, 如果按照通常使用串行机的方法使用并行机, 则并没有充分发挥并行机的运潜能力, 只有采用并行结构的并行算法和相应于计算机的一些高级指令和高级语言 (如向量 Fortran), 才能充分利用并行机的并行特性, 提高运算速度。

并行计算机速度的提高取决于是否利用有效的并行算法, 而并行算法的计算格式和编程取决于所用并行机的并行环境。目前存在的各种并行机所主要归纳为共享内存式 (Shared memory) 和信息传递式 (Message-passing) 两大类。西安交通大学配备的 ELXSI-6400 并行机, 该机属于多 CPU 的程序级的共享内存式并行机。所谓程序级并行是指子程序间的并行, 程序中可以并行的若干部份都以子程序的面目出现, 而这些子程序不象通常串行程序那样一个一个地依次调用, 而是同时被启动, 成为几个同时由多个 CPU 执行的并行进程。所谓共享内存是指各进程之间通讯和并发控制是以共享内存资源的形式实现的。从系统结构上讲, 该机采用总线结构。所有 CPU, 内存和外设都挂在 320Mbyte/每秒的高速总线

* 该文为西安交通大学国家结构强度与振动重点实验室基金资助项目。

上. 各个 CPU 中的进程由系统进程 PM (Prncess Maragar) 负责调度. 另外, 该机的并行方式主要有两种: (1)垂直方式, 它适用于同一程序处理多组独立的数据流的情况. (2)水平方式, 主要指不同程序处理同一数据流的情况.

运行程序通过利用用于并行处理的函数形式的指令来实现并行性能.

目前, 国外已就如何合理地利用并行机来进行工程结构的分析进行了广泛的探讨, 其中在有限元并行计算方面有两个主要分支, 一是有限迭代的并行算法, 其中 T. Hughes, R. Ferencz, J. Hallquist^[1], K. H. Law^[2] 等人在这方面进行过研究; 二是有限元的直接并行算法, 其中 Charbel Farhat, Edward Wilson^[3], Shun Doi, Shoichi Koyama^[4] 等人在这方面作过研究. 迭代并行算法通常是基于 Pcg 和 Jacobi 等迭代法提出来的; 直接并行算法一般是建立在子结构方法或并行同步控制的基础上. 目前线性有限元并行分析方面总的来看已比较成熟, 但在非线性分析方面, 广泛而深入的研究工作还未得到开展, 而象接触分析这类待定边界的非线性问题更是无人问津. 本文利用参数变分原理的接触问题分析方法, 提出了接触分析问题的并行分析方法.

二、弹性接触问题参数变分原理及其有限元数学模型

参数变分原理是一种最近发展起来的求解结构非线性问题的新的变分方法. 它的主要思想是引入不参加变分的参数 λ 来控制系统的变化过程, 对于接触问题等非线性过程, 它的主要优点是每个加载步不需要进行平衡迭代. 钟万勰, 孙苏明^[5] 等人相继在这方面作过深入而广泛的研究工作. 对于参数最小势能原理, 弹性接触系统的总势能为

$$\begin{aligned} \Pi_c = & \int_{\Omega} \left[\frac{1}{2} \varepsilon^T D \varepsilon - b^T \cdot u \right] d\Omega + \int_{S_c} \left[\frac{1}{2} \Delta_c^T D_c \Delta_c - \lambda_*^T \cdot \rho_c \Delta_c \right] dS \\ & - \int_{S_p} p_*^T \cdot u dS \end{aligned} \quad (2.1)$$

这里 $\rho_c = \left(\frac{\partial}{\partial p_c} g^T \right)^T D_c$, λ_* 是不参加变分的参

变矢量. 对于如图 1. 所示的接触问题, 法向压力 p_n 和切向摩擦力 p_t 可以写成,

$$p_n = -E_n \cdot \beta(u_n) \quad \text{在 } L^2(S_c) \text{ 上} \quad (2.2)$$

$$\beta(u_n) = (-\Delta u_n)^+, \Delta u_n = u_n^{(2)} - u_n^{(1)} + \delta^*,$$

$$\phi^+(x) = \sup\{\phi(x), 0\} \in S_c \quad (2.3)$$

$$p_t = \begin{cases} -E_t \Delta u_t, & |p_t| < -\mu p_n \text{ 时,} \\ -\mu p_n \cdot \text{sign}(\Delta u_t), & |p_t| = -\mu p_n \text{ 时,} \end{cases} \quad \Delta u_t = u_t^{(2)} - u_t^{(1)} \quad (2.4)$$

上式中, S_c 为可能接触边界, δ^* 为初始间隙, μ 为摩擦系数, E_n 和 E_t 为接触点处接触元的法向和切向模量 (惩罚因子), 一般取 $E_n = E_t = E$, 并且 $E \rightarrow \infty$. 因此 (2.1) 式中对应于接触单元的势能项

$$\{\Delta_c\} = \{u_t^{(2)} - u_t^{(1)}, u_n^{(2)} - u_n^{(1)} + \delta^*\}^T \quad (2.5)$$

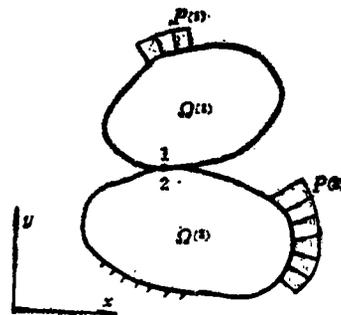


图 1

$$[D_c^e] = \begin{bmatrix} E & 0 \\ 0 & E \end{bmatrix} \quad (2.6)$$

故接触力 $\{p_c\} = \{p_\tau, p_n\}^T$ 与弹性接触位移 $\{\Delta_c^e\}$ 满足

$$\{p_c\} = [D_c^e] \{\Delta_c^e\} = [D_c^e] (\{\Delta_c^e\} - \{\Delta_c^e\}) \quad (2.7)$$

接触面各点满足单边约束条件和库伦摩擦条件

$$*f^{(3)} = p_n \leq 0 \quad (2.8)$$

$$*f^{(1)} = p_\tau + \mu p_n \leq 0 \quad (2.9)$$

$$*f^{(2)} = -p_\tau + \mu p_n \leq 0 \quad (2.10)$$

以上三式实际上是平面接触问题的滑动函数, 如 $*f^{(i)} = 0$, 则接触点对脱离或滑动。定义相应的滑动势函数以给出滑动方向的梯度

$$*g^{(1)} = p_\tau, \quad *g^{(2)} = -p_\tau, \quad *g^{(3)} = p_n \quad (2.11)$$

那么, 接触面上塑性变形 (不可恢复变形) $\{\Delta_c^e\}$ 可由滑动规则决定

$$\Delta_c^e = \lambda_* \begin{pmatrix} \partial *g^T \\ \partial p_c \end{pmatrix} \quad (2.12)$$

这里 $\{*g\} = \{*g^{(1)}, *g^{(2)}, *g^{(3)}\}^T$, λ_* 为滑动参量, 表征滑动或脱离量的大小, 然后将式 (2.6), (2.7), (2.12) 式代入 (2.8), (2.9), (2.10) 式得,

$$\begin{cases} *f^{(1)} = E_\tau \cdot \Delta u_\tau + \mu E_n (\Delta u_n + \delta^*) - E_\tau \lambda_*^{(1)} \leq 0 \\ *f^{(2)} = -E_\tau \cdot \Delta u_\tau + \mu E_n (\Delta u_n + \delta^*) - E_\tau \lambda_*^{(2)} \leq 0 \\ *f^{(3)} = \Delta u_n + \delta^* - \lambda_*^{(3)} \leq 0 \end{cases} \quad (2.13)$$

其中 $\{\lambda_*\} = \{\lambda_*^{(1)}, \lambda_*^{(2)}, \lambda_*^{(3)}\}^T$, 若令

$$*\lambda^{(i)} = \lambda_*^{(i)} \quad (i=1, 2), \quad *\lambda^{(3)} = \lambda_*^{(3)} - \delta^* \quad (2.14)$$

则有

$$*\lambda^{(i)} \begin{cases} = 0 & (\text{当 } *f^{(i)} \leq 0) \\ \geq 0 & (\text{当 } *f^{(i)} = 0) \end{cases} \quad (2.15)$$

其中 $\{*\lambda\} = \{*\lambda^{(1)}, *\lambda^{(2)}, *\lambda^{(3)}\}^T$.

综合 (2.13), (2.14), (2.15) 式, 并引入松弛变量 $*v^{(i)} (i=1, 2, 3)$, 可以导出接触问题的互补条件,

$$\begin{cases} *f^{(i)}(u_\tau^{(\alpha)}, u_n^{(\alpha)}, *\lambda^{(i)}) + *v^{(i)} = 0 \\ *v^{(i)} \cdot *\lambda^{(i)} = 0, \quad *\lambda^{(i)}, *v^{(i)} \geq 0 \quad (i=1, 2, 3, \alpha=1, 2) \end{cases} \quad (2.16)$$

因此对于接触体 Ω 进行有限元离散后, 在接触边界上用接触单元把可能发生接触的物体相互联系起来, 共有 n_2 个接触单元, 且第 e 个接触元共有 $\bar{L}_e (> 0)$ 个滑动条件, 那么共有

$\text{NCCTS} = \sum_{e=1}^{n_2} \bar{L}_e$ 个接触状态, 并设弹性体内部的普通单元为 n_1 个, 将 (2.1) 式和 (2.16) 式进

行有限元离散得,

$$Kc = \frac{1}{2} \delta^T K \delta - \delta^T (\Phi \lambda_* + t) \quad (2.17)$$

$$c\delta - M\lambda_* - d + *v = 0 \quad (2.18)$$

$$*v^T \cdot *\lambda = 0, \quad *v, *\lambda \geq 0 \quad (2.19)$$

其中

$$K = \sum_{e=1}^{n_1} \int_{\Omega^e} B^T DB d\Omega + \sum_{e=1}^{n_2} \int_{S_c^e} N^T D_c^e N dS \quad (2.20)$$

$$\Phi = \sum_{e=1}^{n_2} \int_{S_c^e} \left(\left(\frac{\partial *g_e}{\partial p_c} \right)^T D_c^e N \right)^T dS \quad (2.21)$$

$$C = \sum_{e=1}^{n_2} \int_{S_c^e} \left(\left(\frac{\partial *f_e}{\partial p_c} \right)^T D_c^e N \right)^T dS \quad (2.22)$$

$$M = \sum_{e=1}^{n_2} \int_{S_c^e} \left(\left(\frac{\partial *f_e}{\partial p_c} \right)^T D_c^e \frac{\partial *g_e^T}{\partial p_c} \right) dS \quad (2.23)$$

$$d = - \sum_{e=1}^{n_2} \int_{S_c^e} *f_{0e} dS \quad (2.24)$$

$$t = \sum_{e=1}^{n_1} \left(\int_{\Omega^e} N^T b d\Omega + \int_{S_p^e} N^T p_* dS \right) \quad (2.25)$$

最后, 根据(2.17), (2.18), (2.19)式, 由Kuhn-Tucker条件得

$$\begin{cases} *v - (M - CK^{-1}\phi)\lambda_* = -CK^{-1}t + d \\ * \lambda^T \cdot *v = 0, * \lambda, *v \geq 0 \\ * \lambda = * \lambda + \delta * \end{cases} \quad (2.26)$$

因此问题最后转化为对(2.26)式互补问题的求解. 对于(2.8)~(2.11)式的三个滑动方向来说, 各接触单元的 C, Φ, M 阵分别为

$$[C_e] = \begin{bmatrix} -E_\tau & -\mu E_n \\ E_\tau & -\mu E_n \\ 0 & -E_n \end{bmatrix}, [\Phi_e]^T = \begin{bmatrix} -E_\tau & 0 \\ E_\tau & 0 \\ 0 & -E_n \end{bmatrix} \quad (2.27)$$

$$[M_e] = \begin{bmatrix} E_\tau & -E_\tau & \mu E_n \\ -E_\tau & E_\tau & \mu E_n \\ 0 & 0 & E_n \end{bmatrix} \quad (2.28)$$

另外, (2.20)式的组集过程中, 由于接触单元对应的位移为局部坐标下的相对位移, 因此要将目标接触体(Ω_1)中含接触点的单元刚度阵对应接触点自由度的元素旋转到接触点局部坐标下, 并利用(2.5)式转换到主接触点位移和相对位移下. 互补问题(2.26)式的求解主要归结为求刚度阵的逆, 但是由于惩罚因子 $E \rightarrow \infty$ 的存在, 会使方程出现病态, 所以必须作一些处理, 消去惩罚因子. 设 ND 为结构总独立位移向量 δ 的维数, Nd_n 和 Nd_r 分别为普通位移向量 δ_s 和接触相对位移向量 δ_r 的维数, 于是,

$$\{\delta\}^T = \{\delta_s; \delta_r\} \quad (2.29)$$

其中 δ_s 的维数为 Nd_n , δ_r 的维数为 Nd_r , 所以结构总刚度阵可表为

$$[K] = \begin{bmatrix} K_{11} & K_{12} \\ \cdots & \cdots \\ K_{21} & K_{22} + EK'_{12} \end{bmatrix} \begin{matrix} \uparrow \\ Nd_n \\ \downarrow \\ \uparrow \\ Nd_r \\ \downarrow \end{matrix} \quad (2.30)$$

这里 $E[K'_{22}]$ 为与接触单元有关的刚度阵。且总的 C, Φ, M 阵为,

$$[C] = \begin{bmatrix} 0 & EC' \\ \leftarrow Nd_n \rightarrow & \leftarrow Nd_r \rightarrow \end{bmatrix}, [\Phi] = \begin{bmatrix} 0 & E\Phi' \\ \leftarrow Nd_n \rightarrow & \leftarrow Nd_r \rightarrow \end{bmatrix} \quad (2.31)$$

$$[M] = E[M'] = E \begin{bmatrix} M'_1 & & & \\ & M'_2 & & \\ & & \ddots & \\ & & & M'_{n_2} \end{bmatrix} \quad (2.32)$$

然后求 (2.30) 式逆的二阶近似可得

$$[K]^{-1} = \begin{bmatrix} V_1 & V_2 \\ V_3 & V_4 \end{bmatrix} \quad (2.33)$$

$$V_1 = K_{11}^{-1} + \frac{1}{E} K_{11}^{-1} K_{12} K_{21} K_{11}^{-1} + O\left(\frac{1}{E^2}\right) \quad (2.34)$$

$$V_2 = -\frac{1}{E} K_{11}^{-1} K_{12} + O\left(\frac{1}{E^2}\right) \quad (2.35)$$

$$V_3 = -\frac{1}{E} K_{21} K_{11}^{-1} - \frac{1}{E^2} K_{21} K_{11}^{-1} K_{12} K_{21} K_{11}^{-1} + \frac{1}{E^2} K_{22} K_{21} K_{11}^{-1} + O\left(\frac{1}{E^3}\right) \quad (2.36)$$

$$V_4 = \frac{1}{E} I - \frac{1}{E^2} (K_{22} - K_{21} K_{11}^{-1} K_{21}) + O\left(\frac{1}{E^3}\right) \quad (2.37)$$

最后

$$\lim_{E \rightarrow \infty} (M - CK^{-1}\Phi) = C' K_r \Phi' \quad (2.38)$$

$$\lim_{E \rightarrow \infty} (CK^{-1}t) = C' t_r \quad (2.39)$$

其中

$$K_r = K_{22} - K_{21} K_{11}^{-1} K_{12} \quad (2.40)$$

$$t_r = t_2 - K_{21} K_{11}^{-1} t_1 \quad (2.41)$$

$$\text{这里, } \{t\}^T = \begin{bmatrix} t_1 & t_2 \\ \leftarrow Nd_n \rightarrow & \leftarrow Nd_r \rightarrow \end{bmatrix} \quad (2.42)$$

因此互补问题 (2.26) 式转化为

$$\begin{cases} *v - (C' K_r \Phi') = -C' t_r + C' K_r \delta^* + d \\ * \lambda^T \cdot *v = 0, * \lambda, *v \geq 0 \end{cases} \quad (2.43)$$

最后利用 Lemke 法对 (2.43) 式最终求解。

三、弹性接触问题参数变分原理解法的并行实施

1. 刚度阵的并行生成和组集

一个好的并行算法总是尽可能的使并行成份在总运算量中所占的比重最大, 以尽可能的提高运算速度。虽然从有限元分析的全过程来看, 刚度阵的形成与组集所耗费的时间并不很多, 但是对于单元量和结点数很大的大型结构, 这部份运算的并行实现所产生的效益仍是十分可观的。刚度阵的并行生成是将结构分成对应于 CPU 个数的不同子区, 为了使每个 CPU 的工作负载平衡, 各个子区的单元数应尽量相差不大。如图 2 所示, 划分两个子区, 每个子

区含两个普通单元,一个接触单元.图中粗实线为接触区,花线为子区交界区.在并行生成单元刚度阵之前,必须建立并行进程共享的数据结构,它含有结点坐标,单元结点编号,单元材料号等数据信息.总刚的组集的并行实施是个十分困难的问题,因为组集过程从本质上讲是串行的,特别是当不同单元的刚度信息要组集到总刚中同一地址中去时(overlapping),如果是并行执行,很显然得不到正确的结果.文[7]曾利用图论中的着色问题对并行组集刚度阵进行过讨论,但是必须对结点重新进行编号,由于接触问题的结点编号是有特定顺序的,因此这种方法势必导致对结点编号和刚度阵中元素位置的反复调整.这里我们利用并行机特殊功能建立了两种并行组集方法.

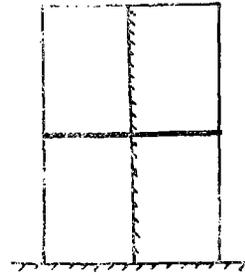


图 2

(1) 利用“信号量”(Semaphore)和“锁”(lock)进行并行组集.该方法首先设置一个共享的变量,它存放了各并行进程即将组集进总刚 A 数组中的地址.“信号量”用于进行同步控制,“锁”用于避免不同进程对同一数据区进行操作.在同步控制各进程中放入总刚的地址后,判断是否有相同的地址,如有,调用“锁”变量进行控制,如没有,则并行组集.其流程如图3所示.该方法的主要优点是对不同组集地址时,不需调用“锁”变量可并行组集,主要弱点在于在组集循环中每次调用“信号量”进行同步控制,需要花费一定的CPU时间.

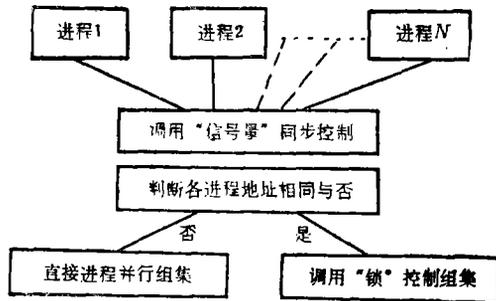


图 3

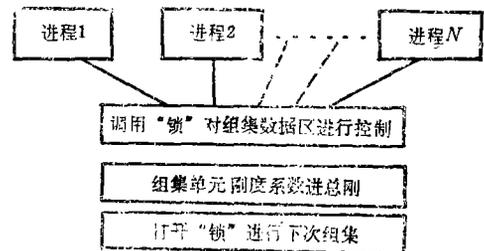


图 4

用“信号量”进行同步控制,节省了时间;主要弱点在于每次循环都在调用“锁”变量(不管不同进程是否对一数据区进行操作)控制组集过程,花费了不必要的时间.

两种组集方法各有优点,对于图2的情况,第一种方法的CPU时间为0.662秒,第二种方法的CPU时间为0.641秒,计算时间包含了并行生成刚度阵的时间.

2. 高斯消元的并行静凝聚方法

从前面的分析可以看出,计算中最耗费时间的部份是(2.40)和(2.41)式的静凝聚部份,因此这部份的并行求解是整个运算过程的关键.文献[3],[6]曾讨论过 LDL^T 分解的并行实施方法,这里我们提出一种高斯消元的并行方法来对(2.40)和(2.41)式进行计算.

(1) K_{11} 的并行消元

设并行机共有 N_p 个 CPU, 并设刚度阵的等带宽存储, 其在各 CPU 中的分配存储方式如图 5 所示, 从图中可以看出第 i 个 CPU 存放的行为 $(i, i+N_p, i+2N_p, \dots)$. 其并行消元方式主要有以下几步,

a. 任选一 CPU 进行主元行的处理, 即如对第 i 行有

$$K(i, j) = K(i, j) / K(i, 1) \quad j=1, \dots, \text{NBHC}, i=1, \dots, \text{NN} \quad (3.1)$$

其中 $K(i, 1)$ 为等带宽存储时第 i 行的主元, NBHC 为带宽值, NN 为方程阶数. 这一步之后各 CPU 进程必须利用“信号量”(semaphore) 设置一同步点, 为的是保证将当前第 i 行的主元值和行号等信息传递到其它并行子进程中.

b. N_p 个进程在接到主元信息之后, 对图 5 中所示的各自存放的行并行进行消元, 即对第 I 个 CPU 有,

$$K(k, j) = K(k, j) - K(i, 1+k-i) \cdot K(i, j+k-i) \cdot K(i, 1) \quad (3.2)$$

其中 $k=i+N_p, \dots, \text{NN}$ (以 N_p 为增量), $j=1, \dots, \text{NBHC}$, i 为第 a 步传入的主元行号. 这一步完成后, 各并行子进程都必须设置一同步点, 使各子进程这一步同时结束. 然后再返回 a 步进行下一行的消元.

(2) 右端项的处理

从 (2.40) 和 (2.41) 式可以看出, 我们要处理的右端项主要是 K_{12} 的各列和 t_1 . 其在各 CPU 中的存储方式如图 6 所示, 即第 i 个 CPU 存放 K_{12} 和 t_1 对应的行为 $(i, i+N_p, i+2N_p, \dots)$. 其并行处理方式主要有下面几步,

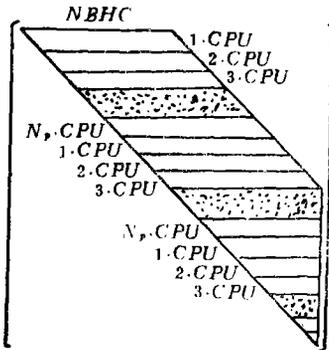


图 5

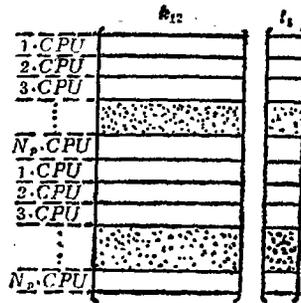


图 6

a. 用上面 K_{11} 消元过程中 a 步选定的 CPU 进行当前行处理

$$\begin{cases} K_{12}(i, j) = K_{12}(i, j) / K(i, 1) \\ t_1(i) = t_1(i) / K(i, 1) \end{cases} \quad (3.3)$$

其中 i 为当前行 ($i=1, \dots, \text{NN}$), j 为从 1 到 N_k 的整数, N_k 为 K_{12} 的列数, $K(i, 1)$ 为当前行的主元. 和上面 K_{11} 消元的 a 步一样, 在这里要进行同步控制.

b. N_p 个子进程在接到主元信息之后, 对图 6 中所示的各自存放的行并行处理, 即对第 I 个 CPU 有

$$\begin{cases} K_{12}(k, j) = K_{12}(k, j) - K(i, 1+k-i) \cdot K(i, j) \cdot K(i, 1) \\ t_1(k) = t_1(k) - t_1(i) \cdot K(i, 1+k-i) \cdot K(i, 1) \end{cases} \quad (3.4)$$

其中 $k=i+N_p, \dots, \text{NN}$ (以 N_p 为增量), $j=1, \dots, N_k$, i 为第 a 步传入的当前行号. 同上面

的 b 步一样, 这儿也必须设置同步点, 在各并行进程完成后, 再返回 a. 步进行下步。

(3) 回代过程的并行处理

回代过程由与前后顺序相关, 所以一个单独的回代过程很难并行完成。但是由于我们需要进行的是对 N_k+1 个列向量 (K_{12} 的 N_k 个列向量和 t_1 向量)。因此我们可以将 N_k+1 分成 N_p 份, 分配到 N_p 个 CPU 中, 由 N_p 个 CPU 并行对自己分配的各列进行回代。最后通过将 (2.40) 和 (2.41) 式中的 K_{21} 分成 N_p 个行块, 每一行块放在不同的 CPU 中以执行与 $K_{11}^{-1}K_{12}$ 和 $K_{11}^{-1}t_1$ 的矩阵积, 这样 (2.40) 和 (2.41) 就可以最终并行生成。

3. 在完成上面的静凝聚过程之后, 如果这时接触元很多, 可以按照前面讨论的高斯并行消元法并稍加以改进, 完成并行的 Lemke 分析。

4. 在求得各点位移后, 接前面讨论的并行生成刚度阵的方法, 将单元分成 N_p 批, 并行生成单元高斯点应力。

上面讨论了算法中的各并行步, 然后按照 ELXSI 公司提供的并行操作内部函数, 完成编程。

四、算 例

为了对该并行算法的效率进行分析, 我们分别用调通的串行与并行方法对一算例进行计算, 并对计算时间作了比较。算例如图 7 所示, 两弹性体接触。其中共有 5 个接触元, 在接触元不变的情况下, 图 8 反映了两种算法随结构总体自由度变化的情况。从图中可以看出,

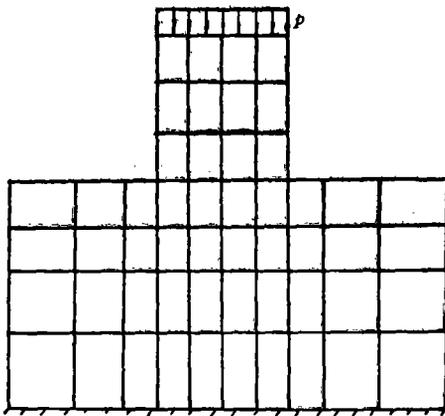


图 7

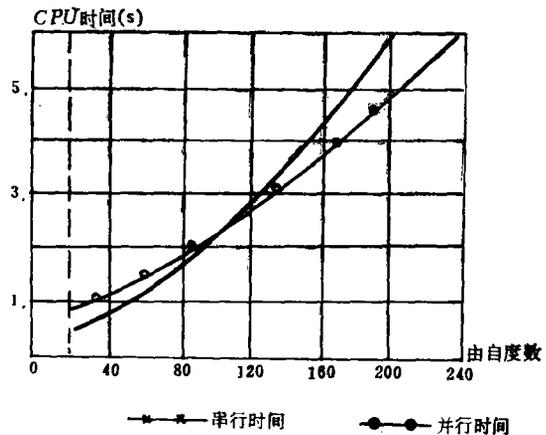


图 8

随着结构自由度的增加, 在一次加载的情况下, 所相对节约的时间也越多(图 8 中为 2 个 CPU 的计算时间情况, 因为西安交通大学的 ELXSI-6400 机目前仅配备两个 CPU)。

五、结 论

从上面的分析可以看出, 充分利用并行机的特性建立起来的弹性接触问题参数变分原理有限元算法的并行处理方法是一种有效的分析结构接触问题的方法。

参 考 文 献

- [1] Hughes, T. J. R., I. Lewit and J. Winget, An element-by-element solution algorithm for problems of structural and solid mechanics, *Computer Method in Applied Mechanics and Engineering* (1983), 241—254.
- [2] Law, K.H., A parallel finite element solution method, *Comp. Struct.*, **23** (6) (1986), 845—858.
- [3] Charbel Farhat and Edward Wilson, A parallel active column equation solver, *Com. Struct.*, **28** (2) (1988), 289—304.
- [4] Doi, Shun and Shoichi Koyama, A parallel computation technique for finite element method, *System Computers Controls* **13** (2) (1982).
- [5] 孙苏明, 参数二次规划法的研究及其工程结构分析应用 (大连理工大学博士论文) (1988).
- [6] Goehlich, D., L. Komzsik and R. E. Fulton, Application of a parallel equation solver to static FEM solution, *Comp. Struct.*, **31** (2) (1989), 121—129.
- [7] 胡宁、张汝清, 并行的有限元刚度阵组集方法, 工程力学 (待发表)。

A Finite Element Parallel Algorithm of the Parametric Variational Principle for Elastic Contact Problems

Hu Ning Zhang Ru-qing

(Department of Engineering Mechanics, Chongqing University, Chongqing)

Abstract

Based on the finite element solution of the parametric variational principle of elastic contact problem, a corresponding parallel algorithm has been created by utilizing the specialities of parallel computer and the architecture of concurrent processing in this paper. In this algorithm, the parallelisms have been realized in the processes of creation and assembly of stiffness matrix, of the static condensation, of the solution of stresses and in many other aspects. The programme of this algorithm has been realized on ELXSI-6400 parallel computer of Xi'an Jiaotong University. The results of computation show that the computational time can be saved efficiently and it is an effective parallel algorithm for the analyses of contact problems.

Key words parallel algorithm, contact analysis