

一种求解结构动力响应的并行解法

余天堂 姜弘道

(河海大学土木工程学院 南京 210098)

摘 要 利用直接积分法结合子结构技术和网络并行计算的特点 , 提出一种求解结构动力响应的网络并行解法 . 该方法在并行形成各子结构的劲度矩阵、质量矩阵、阻尼矩阵及荷载列阵后 , 并行进行各子结构的静凝聚 , 然后采用并行子结构预条件共轭梯度法迭代求解结点位移 , 从而确定结构的动响应 . 算例表明 , 随着子结构内点数和时间积分步数的增加 , 能获得较高的加速比 .

关键词 动力响应 ; 子结构 ; 网络并行计算 ; 加速比

中图号 O342

结构动力问题在数值分析中是一个重要领域 , 同时又由于它比结构静力问题更耗费机时 , 因此有不少学者都对这个领域内的并行计算进行过研究^[1~4] , 取得了一系列成果 . 并行计算的计算格式与编程取决于所用并行机的并行环境 , 以前的并行研究基本上都在巨型并行机上进行的 . 网络并行计算是近年来发展起来的一个重要并行计算方向 , 它具有巨大的计算潜能 , 良好的性能价格比和可扩展性 , 以及灵活的体系结构等优点 . 近年来 , 又出现了一批基于消息传递的并程序序设计软件平台 , 极大地方便了并行编程 , 同时保证了程序的可移植性^[5] .

本文利用子结构技术结合网络并行计算的特点提出一种求解复杂系统动力响应的并行直接积分法 . 该方法不必组集界面方程 , 而是采用并行子结构预条件共轭梯度法迭代求解结点位移 , 首先给界面结点一初始位移 , 然后回代求子结构内点位移 , 再修正界面结点位移和回代求内点位移 , 直至收敛 . 本文在基于 PVM (Parallel Virtual Machine) 的网络并行环境下实现了该方法 . 算例表明该方法在网络上能获得较高的加速比 .

1 子结构动力响应

对一个结构进行有限元离散处理后 , 可得到如下的动力控制方程^[6] :

$$M\ddot{\delta} + C\dot{\delta} + K\delta = F(t) \tag{1}$$

式中 $\ddot{\delta}$, $\dot{\delta}$, δ ——加速度、速度、位移 ; K , C , M , $F(t)$ ——结构的整体劲度矩阵、阻尼矩阵、质量矩阵、结构结点的荷载列阵 .

将方程 (1) 对时间离散化积分 (以 Wilson - θ 法为例) :

$$\hat{K}\delta_{t+\theta\Delta t} = \hat{F}_{t+\theta\Delta t} \tag{2}$$

$$\hat{K} = K + \frac{6}{\theta^2\Delta t^2}M + \frac{3}{\theta\Delta t}C \tag{3}$$

$$\begin{aligned} \hat{F}_{t+\theta\Delta t} = & F_t + \theta(F_{t+\theta\Delta t} - F_t) + M\left(\frac{6}{\theta^2\Delta t^2}\delta_t + \frac{6}{\theta\Delta t}\dot{\delta}_t + 2\ddot{\delta}_t\right) + \\ & C\left(\frac{3}{\theta\Delta t}\delta_t + 2\dot{\delta}_t + \frac{\theta\Delta t}{2}\ddot{\delta}_t\right) \end{aligned} \tag{4}$$

求解方程 (2) 就得到 $\delta_{t+\theta\Delta t}$. 从而 , 有

$$\ddot{\delta}_{t+\Delta t} = \frac{6}{\theta^3\Delta t^2}(\delta_{t+\theta\Delta t} - \delta_t) - \frac{6}{\theta^2\Delta t}\dot{\delta}_t + \left(1 - \frac{3}{\theta}\right)\ddot{\delta}_t \tag{5}$$

$$\dot{\delta}_{t+\Delta t} = \dot{\delta}_t + \frac{\Delta t}{2}(\ddot{\delta}_{t+\Delta t} + \ddot{\delta}_t) \quad (6)$$

$$\delta_{t+\Delta t} = \delta_t + \Delta t \dot{\delta}_t + \frac{\Delta t^2}{6}(\ddot{\delta}_{t+\Delta t} + 2\ddot{\delta}_t) \quad (7)$$

当 $\theta \geq 1.37$ 时, 这一隐式积分是无条件稳定的. 若采用 Newmark 积分法, 则只需改变上述计算式中的系数, 因此这两种积分法可在同一程序中实现.

如果采用子结构法, 方程(2)则为各子结构的动力方程. 子结构的结点编号采用先内点后界面点的顺序, 则对第 s 个子结构, 其动力方程可写为

$$\begin{bmatrix} \hat{K}_{ii}^s & \hat{K}_{ib}^s \\ \hat{K}_{bi}^s & \hat{K}_{bb}^s \end{bmatrix} \begin{Bmatrix} \{\delta_i^s\}_{t+\theta\Delta t} \\ \{\delta_b^s\}_{t+\theta\Delta t} \end{Bmatrix} = \begin{Bmatrix} \{\hat{F}_i^s\}_{t+\theta\Delta t} \\ \{\hat{F}_b^s\}_{t+\theta\Delta t} \end{Bmatrix} \quad (8)$$

式中: 下标 i ——内点自由度; 下标 b ——界面点自由度; 上标 s ——子结构编号.

式(8)通过静凝聚后, 界面方程为:

$$K^* \{\delta_b\}_{t+\theta\Delta t} = \{F^*\}_{t+\theta\Delta t} \quad (9)$$

$$K^* = \sum_{s=1}^n \hat{K}_{bb}^s - \hat{K}_{bi}^s [\hat{K}_{ii}^s]^{-1} \hat{K}_{ib}^s \quad (10)$$

$$\{F^*\}_{t+\theta\Delta t} = \sum_{s=1}^n \{\hat{F}_b^s\}_{t+\theta\Delta t} - \hat{K}_{bi}^s [\hat{K}_{ii}^s]^{-1} \{\hat{F}_i^s\}_{t+\theta\Delta t} \quad (11)$$

n 为子结构的总数. 由式(9)求出 $\{\delta_b\}_{t+\theta\Delta t}$ 后代入式(8)就可得 $\{\delta_i\}_{t+\theta\Delta t}$, 由式(5)(6)(7)可求得 $\dot{\delta}_{t+\Delta t}$, $\delta_{t+\Delta t}$.

2 并行求解界面方程

本文采用预条件共轭梯度法(PCG)^[7]求解界面方程(9), 为了能并行求解, 需要把 PCG 法的各步计算转换成子结构上的计算. 下面给出一种 PCG 法并行求解方程(9)的过程.

取 K^* 的对角阵为预条件矩阵. 为了书写方便, 省略下标 $t + \theta\Delta t$.

a. 给界面结点位移 δ_b —初值 $\{\delta_b^s\}_0$ (一般取 $\{\delta_b^s\}_0 = \{0\}$), 计算

$$C_b^s = \text{diag} \hat{K}_{bb}^s - \text{diag}(\hat{K}_{bi}^s [\hat{K}_{ii}^s]^{-1} \hat{K}_{ib}^s)$$

$$\overline{C_b^s} = \sum_{j \in m} C_b^j + C_b^s \quad (m \text{ 为与第 } s \text{ 个子结构相邻的子结构域})$$

$$\{\delta_i^s\}_0 = -[\hat{K}_{ii}^s]^{-1}(\hat{K}_{ib}^s \{\delta_b^s\}_0 - \hat{F}_i^s)$$

$$\{r_b^s\}_0 = \hat{F}_b^s - (\hat{K}_{bi}^s \{\delta_i^s\}_0 + \hat{K}_{bb}^s \{\delta_b^s\}_0)$$

$$\{t_b^s\}_0 = [\overline{C_b^s}]^{-1} \{r_b^s\}_0$$

$$\{s_b^s\}_0 = \sum_{j \in m} \{t_b^j\}_0 + \{t_b^s\}_0$$

$$\{p_b^s\}_0 = \{s_b^s\}_0$$

式中: $\sum_{j \in m}$ ——局部通讯求和, $\sum$ ——全局通讯求和.

b. 对 $k = 0, 1, 2, \dots$ 重复执行

$$\{p_i^s\}_{k+1} = -[\hat{K}_{ii}^s]^{-1} \hat{K}_{ib}^s \{p_b^s\}_k$$

$$\{h_b^s\}_{k+1} = \hat{K}_{bb}^s \{p_b^s\}_k + \hat{K}_{bi}^s \{p_i^s\}_{k+1}$$

$$a_n^s = \{r_b^s\}_k^T \{s_b^s\}_k \quad a_d^s = \{p_b^s\}_k^T \{h_b^s\}_{k+1}$$

$$a_n = \sum a_n^s \quad a_d = \sum a_d^s \quad a = a_n / a_d$$

$$\{\delta_i^s\}_{k+1} = \{\delta_i^s\}_k + a \{p_i^s\}_{k+1} \quad \{\delta_b^s\}_{k+1} = \{\delta_b^s\}_k + a \{p_b^s\}_k$$

$$\Delta = \|\delta_{k+1} - \delta_k\|_2 \quad (\text{此处需进行全局通讯})$$

若 $\Delta < \epsilon$, 则终止.

$$\{r_b^s\}_{k+1} = \{r_b^s\}_k - a \{h_b^s\}_{k+1}$$

$$\{t_b^s\}_{k+1} = [\overline{C_b^s}]^{-1} \{r_b^s\}_{k+1}$$

$$\begin{aligned} \{s_b^s\}_{k+1} &= \sum_{j \in m} \{t_b^j\}_{k+1} + \{t_b^s\}_{k+1} \\ \gamma_n^s &= -a \{h_b^s\}_{k+1}^T \{s_b^s\}_{k+1} \quad \gamma_n = \sum \gamma_n^s \quad \gamma = \gamma_n / a_n \\ \{p_b^s\}_{k+1} &= \gamma \{p_b^s\}_k + \{s_b^s\}_{k+1} \end{aligned}$$

ϵ 为容许收敛误差,一般取 $\epsilon = 10^{-6}$.至此,所有结点 $t + \theta \Delta t$ 时刻的位移都已知了.

由上述计算步骤可知,总体刚度矩阵和荷载列阵并不需形成,各微机都是同步进行的,如果各微机上的任务均衡,则不会出现空闲机.同时可知,只有在进行与界面结点有关的计算时才需进行相互间的通讯,且通讯的数据只与界面结点有关.所以为了减少通讯量,应尽可能减少界面结点数.上述迭代格式中由于是先给界面结点一初始位移,再回代求出内点位移,因此这种迭代的收敛速度比先给所有结点一初始位移要快.

3 并行子结构动力响应

将结构划分为 n 个子结构,再将 n 个子结构的数据分送给网上 n 台可用微机,下面给出并行子结构法求动力响应的过程.

- a. n 台微机并行形成 n 个子结构的 K^s, M^s, C^s , 给出初始值 $\{\delta^s\}_0, \{\dot{\delta}^s\}_0, \{\ddot{\delta}^s\}_0$, 形成各子结构的有效刚度矩阵 $\hat{K}^s$.
 - b. 对每一时间步计算: 计算各子结构 $t + \Delta t$ 时刻的有效荷载列阵 $\{\hat{F}^s\}_{t+\theta\Delta t}$; 采用并行子结构预条件共轭梯度法求解 $t + \theta \Delta t$ 时刻的位移, 计算 $t + \Delta t$ 时刻的位移、速度及加速度. 直至结束.
- 上述各步均可并行进行.

4 并程序序设计

本文使用的并行环境为总线结构的以太网,网络计算环境采用 PVM. 通过调用 PVM 提供的接口函数,可方便地编写并程序序. PVM 提供的接口函数包括进程控制、虚拟机动态配置、消息传递及任务组等函数^[5].

本文采用主从式 (Master-Slave) 并行处理模式,这种计算模式有一个控制程序称为主进程 (Master),负责进程的生成、初始化、收集并输出计算结果,其余的从进程 (Slave) 执行实际计算,其负载或者是由主进程分配 (有静态分配与动态分配两种),或者是由进程本身分配. 一个问题的求解,由主进程和从进程共同完成. 进程的生成、管理及相互间的通讯等并行过程通过调用 PVM 库函数实现. 由于网络上的通讯开销大,因此本文采用子结构级的粗粒度并行方式.

5 算 例

设有一悬臂梁结构 (图 1), 作用有如图所示的激振力 $p(t)$, 以 PVM 作为网络并行计算的并行环境, 用本文所给的方法计算其动响应. 将结构划分为两个子结构 (如图 1), 使用以太网上两台微机进行并行计算, 采用平面八结点等参单元, 加速比 $P = T_1/T_2, T_1$ 为串行计算时间, T_2 为并行计算时间.

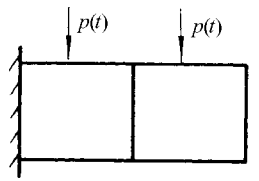


图 1 结构

由图 2 可知随着结构自由度的增加 (内点自由度占总自由度比例的增加), 实际 Fig.1 Structural diagram 并行时间也就越接近理想并行时间. 同时也可看出, 随着时间积分步数的增加, 所节约的时间也会增加.

6 结 论

本文所给的子结构并行处理的结构动力响应解法具有以下优点: 编程简单, 可移植性强, 获得的加速比高. 根据本文的理论编制的程序, 可移植到以 PVM 为并行环境的并行机上运行. 由前面的介绍可知, 不需进行通讯的部分与串程序一样, 因此可直接将串程序中计算 K, M, C 等部分移植过来. 由算例可知, 随着子结构内点数和时间积分步数的增加, 该方法可大大地节约计算时间. 任务的划分在并行计算中十分重要, 它直接影响并行效果, 因此应尽量平衡各微机上的任务.

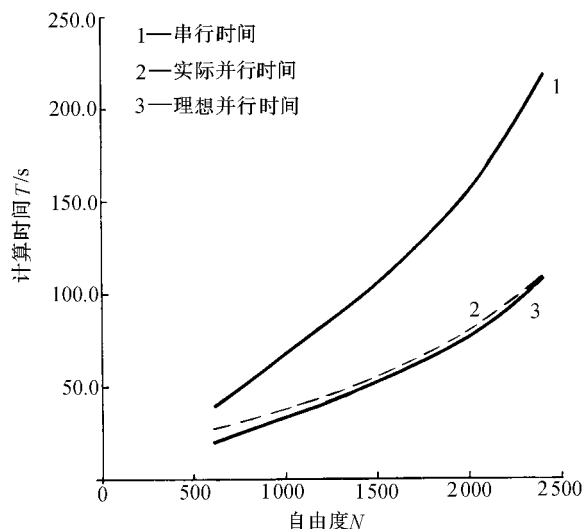


图2 自由度与计算时间的关系

Fig.2 Freedom vs computing time

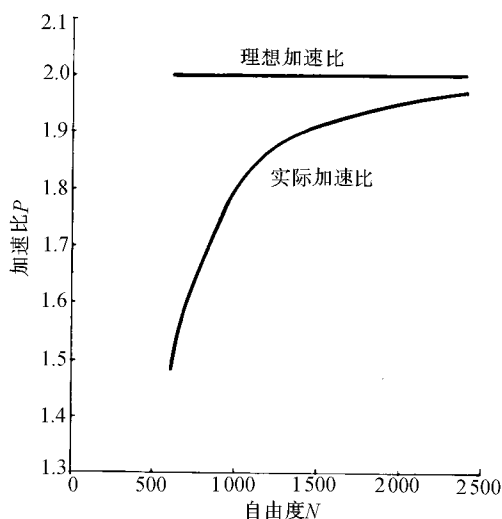


图3 自由度与加速比的关系

Fig.3 Freedom vs speedup ratio

参 考 文 献

- 1 Noor A, Lambiotte J. Finite element dynamic analysis on CDC STAR-100 computers. Computers & Structures, 1978; 8(8): 621 ~ 632
- 2 Olaf Storaasli, Jonathan Ransom. Structural dynamic analysis on a parallel computer: the finite element machine. Computers & Structures, 1987; 26(4): 551 ~ 559
- 3 Chiang K N, Fulton R E. Parallel transient finite element analysis. Computers & Structures, 1992; 42(5): 733 ~ 739
- 4 Synn S Y, Fulton R E. The concurrent element level processing for nonlinear dynamic analysis on a massively parallel computer. Computing Systems in Engineering, 1995; 1(3): 285 ~ 293
- 5 孙家昶, 张林波. 网络并行计算与分布式编程环境. 北京: 科学出版社, 1997. 36 ~ 113
- 6 张汝清. 并行计算结构力学. 重庆: 重庆大学出版社, 1993. 280 ~ 286
- 7 刘万勋. 大型稀疏线性方程组的解法. 北京: 国防工业出版社, 1981. 132 ~ 134

A Parallel Solution of Dynamic Response of Structure

Yu Tiantang Jiang Hongdao

(College of Civil Engineering, Hohai Univ., Nanjing 210098)

Abstract Based on Wilson- θ and Newmark integration methods, combined with the technique of substructure, in consideration of the characteristics of network parallel computation, a network parallel algorithm for solving dynamic response problems is presented in this paper. In the method, the stiffness matrix, the mass matrix, the damp matrix and the loading array of each substructure are formed in parallel, then the nodal displacements are obtained with the parallel substructure preconditioned conjugate gradient method, and thus the dynamic response is determined. The computation shows that with the method better speedup ratio can be achieved with the increase of the inner nodal number of the substructure and time integrating steps.

Key words dynamic response; substructure; network parallel computation; speedup ratio