

# 大型结构子结构解法的并行算法

张汝清

(重庆大学工程力学系, 1990年2月20日收到)

## 摘 要

本文根据ELXSI机的并行环境, 提出了大型复杂结构静力和动力分析中子结构解法的并行算法, 并研制了相应的并行计算程序。

**关键词** 结构分析 子结构解法 并行算法

## 一、引 言

现在作结构分析计算一般使用的是串行机, 而各种分析方法、计算格式和相应的程序也都是串行的。

当前, 在许多大型复杂结构的分析计算中, 提出了许多大型和超大型的计算问题。解决这类问题要求具有运算速度快, 存储容量大的计算机。但是, 按传统的串行机的体系结构已不能装出高速计算机来, 于是, 近十多年来, 相继出现了从亿次/秒~100亿次/秒以上的并行机, 这就为结构分析实现并行计算, 解决大型复杂结构分析问题提供了物质条件。

ELXSI并行机是属于多机系统。从并行计算的观点来看, 它可实现程序级并行。即是把一个完整的程序被分割为一些可同时执行的独立部分, 每一部分可在一个CPU上运行。这样, 一个程序的若干部分可并行执行, 从而加快了整个程序的执行速度。

在这种系统中, 同时有各种指令在对不同的数据进行处理, 因此, 人们也把这种并行称之为“多指令流多数据流(MIMD)”的并行。

在程序必须按次序顺序执行时, 各CPU可以分别执行不同程序段。各个CPU既可单独作战, 又可协同工作, 使并行计算得以灵活实现。

显然, 在ELXSI机上作结构分析, 实现并行计算, 可将整个结构划分为若干子结构。每个子结构的计算, 可在一个CPU上运行。这样, 各个子结构可并行执行, 从而提高结构分析的执行速度。因此, 我们提出了静力分析中子结构解法的并行算法, 动力分析中模态综合子结构解法的并行算法, 并针对这些并行算法, 研制了静力和动力分析的软件。

## 二、静力分析子结构解法的并行算法

对于大型复杂结构分析问题, 通常采用子结构解法, 即先将结构划分为若干子结构进行局部分析, 然后综合组集再作整体分析, 这是一种先局部后整体的降阶方法。从子结构解法的另一种意义上讲, 对于任何结构都可任意地去排定求解方程的先后次序, 使得运算起来较

\* 创刊十周年暨一百期纪念特刊(I)论文。

为方便, 我们可利用这个特点来安排求解方程的次序, 划分相对独立并行无关的子结构, 就可将原来基于串行机的串行算法改成基于并行机的并行算法。即把每个子结构分配到各个 CPU 中, 各自独立地进行分解消元运算、实现子结构一级程序并行执行。在分解消元后, 综合组集, 得到了降阶的整体方程。这样, 既保留了子结构解法降阶的优点, 又可在并行机上实施, 这就是结构分析中子结构并行算法的基本思想。

对于一个大型结构, 设所划分的子结构为  $N$ , 每个子结构中, 又划分为若干单元, 未知量的编号, 从第一个子结构开始, 在编完各子结构后, 再编界面, 则组集后的集体方程系数矩阵为一加边对角块型, 即

$$\begin{bmatrix} K_{11} & & 0 & & K_{1M} & & U_1 \\ & K_{22} & & & K_{2M} & & U_2 \\ & & \ddots & & \vdots & & \vdots \\ 0 & & & \ddots & K_{NM} & & U_N \\ K_{1M}^T & K_{2M}^T & \cdots & K_{MM} & K_{MM} & & U_M \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_N \\ R_M \end{bmatrix} \quad (2.1)$$

下标  $M$  表示与界面相对应的量。根据三角分解和高斯消元的原理, 把第  $N$  个子结构分解后, 方程变为

$$\begin{bmatrix} D_{11}L_1^T & & 0 & & L_1^{-1}K_{1M} \\ & D_{22}L_2^T & & & L_2^{-1}K_{2M} \\ & & \ddots & & \vdots \\ 0 & & & D_{NN}L_N^T & L_N^{-1}K_{NM} \\ & 0 & & & K_{MM} - \sum_{i=1}^N K_{iM}^T K_{ii}^{-1} K_{iM} \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_N \\ U_M \end{bmatrix} = \begin{bmatrix} L_1^{-1}R_1 \\ L_2^{-1}R_2 \\ \vdots \\ L_N^{-1}R_N \\ R_M - \sum_{i=1}^N K_{iM}^T K_{ii}^{-1} R_i \end{bmatrix} \quad (2.2)$$

根据有限元的组集方法,  $K_{MM} = \sum_{i=1}^N K_{MM}^i$ , 这里,  $K_{MM}^i$  为第  $i$  个子结构对  $K_{MM}$  的贡献, 于是, 与界面对应的系数矩阵和右端项变为

$$\sum_{i=1}^N (K_{MM}^i - K_{iM}^T K_{ii}^{-1} K_{iM}), \quad \sum_{i=1}^N (R_M^i - K_{iM}^T K_{ii}^{-1} R_i)$$

按照列高三角分解原理, 在作分解过程中, 每个子结构对应的系数, 只与界面有关, 而与其它子结构无关。因此, 对于任意的第  $i$  个子结构, 相应于上面的分解和消元, 有

$$\begin{bmatrix} D_{ii}L_i^T & L_i^{-1}K_{iM} \\ 0 & K_{MM}^i - K_{iM}^T K_{ii}^{-1} K_{iM} \end{bmatrix} \begin{bmatrix} U_i \\ U_M^i \end{bmatrix} = \begin{bmatrix} L_i^{-1}R_i \\ R_M^i - K_{iM}^T K_{ii}^{-1} R_i \end{bmatrix} \quad (2.3)$$

上式就是对子结构内部进行三角分解, 而对界面进行消元的结果。将各子结构消元后的界面

系数按一定的顺序组集, 得

$$\left[ K_{MM} - \sum_{i=1}^N K_{iM}^T K_{ii}^{-1} K_{iM} \right] [U_M] = \left[ R_M - \sum_{i=1}^N K_{iM}^T K_{ii}^{-1} R_i \right] \quad (2.4)$$

这也称为静凝聚过程。由此, 静凝聚方法解出 $[U_M]$ , 再返回式(2.3)中解出 $[U_i]$ 。

静力分析中子结构解法并行算法的操作步骤是

1. 将结构划分为 $N$ 个子结构, 按照先内部后界面的方法进行结点编号, 输入有关的信息。
2. 各CPU独立形成各子结构的刚度矩阵, 即

$$\begin{bmatrix} K_{ii} & K_{iM} \\ K_{Mi} & K_{MM} \end{bmatrix} \quad (i=1, 2, \dots, N)$$

3. 各CPU对各子结构并行执行分解消元, 即

$$\begin{bmatrix} D_{ii} L_i^T & L_i^{-1} K_{iM} \\ 0 & K_{MM}^i - K_{iM}^T K_{ii}^{-1} K_{iM} \end{bmatrix} \begin{bmatrix} U_i \\ U_M^i \end{bmatrix} = \begin{bmatrix} L_i^{-1} R_i \\ R_M^i - K_{iM}^T K_{ii}^{-1} R_i \end{bmatrix} \quad (i=1, 2, \dots, N)$$

4. 组集聚缩后的整体方程组, 得

$$K_{MM}^* = K_{MM}^0, \quad R_M^* = R_M^0$$

$$K_{MM}^* = K_{MM}^0 + (K_{MM}^i - K_{iM}^T K_{ii}^{-1} K_{iM})$$

$$R_M^* = R_M^0 + (R_M^i - K_{iM}^T K_{ii}^{-1} R_i) \quad (i=1, 2, \dots, N)$$

5. 求解整体方程组  $U_M = K_{MM}^{*-1} R_M^*$

6. 各CPU对各子结构回代求解

$$U_i = (L_i^T)^{-1} D_{ii}^T (L_i^{-1} R_i - L_i^{-1} K_{iM} U_M^i)$$

### 三、子结构模态综合解法的并行算法

动态子结构模态综合法是结构动力分析中十分有效的方法。现有的计算格式和相应的软件是适用于串行机的。仍仿照上述静力分析中子结构并行化的思想, 我们提出了固定界面子结构模态综合解法的并行算法。

设将结构划分为 $N$ 个子结构, 其划分的方法与静力分析中的子结构划分方法相同。

若对每个子结构中的结点位移, 分为内部结点位移和界面结点位移, 则对于子结构 $i$ 的自由振动方程为

$$\begin{bmatrix} M_{ii} & M_{iM} \\ M_{Mi} & M_{MM} \end{bmatrix} \begin{bmatrix} \ddot{U}_i \\ \ddot{U}_M^i \end{bmatrix} + \begin{bmatrix} K_{ii} & K_{iM} \\ K_{Mi} & K_{MM} \end{bmatrix} \begin{bmatrix} U_i \\ U_M^i \end{bmatrix} = \begin{bmatrix} 0 \\ R_M^i \end{bmatrix} \quad (3.1)$$

固定界面模态综合法认为, 在求解子结构的主模态时, 假定界面坐标是固定的, 即有附加约束。在固定界面下, 子结构内部振动主模态为 $[\phi_i]$ , 释放界面结点所得到的约束模态 $[\phi_M]$ , 即

$$[U_i] = [\phi_i \quad \phi_M] \begin{bmatrix} X_i \\ U_M^i \end{bmatrix} \quad (3.2)$$

这里  $[\phi_i]$  可从固定界面下的振动方程求得, 那从方程

$$[K_{ii}][\phi] = \lambda[M_{ii}][\phi] \quad (3.3)$$

中求得的全部或部分 $[\phi]$ 所组成。而 $[\phi_M]$ 是约模态，它可从静力约束方程

$$[K_{ii}]\{U_i\} + [K_{iM}]\{U_M^i\} = \{0\} \quad (3.4)$$

中求得，即  $[\phi_M] = -[K_{ii}]^{-1}[K_{iM}]$

于是，有 
$$\begin{bmatrix} U_i \\ U_M^i \end{bmatrix} = \begin{bmatrix} \phi_i & -K_{ii}^{-1}K_{iM} \\ 0 & I \end{bmatrix} \begin{bmatrix} X_i \\ U_M^i \end{bmatrix} \quad (3.5)$$

利用上式对方程(3.1)作动力变换，并假定 $M_{Mi} = M_{iM} = 0$ ，得到

$$\begin{bmatrix} I & -\phi_i^T M_{ii} K_{ii}^{-1} K_{iM} \\ \text{对称} & M_{MM} + K_{Mi} K_{ii}^{-1} M_{ii} K_{ii}^{-1} K_{iM} \end{bmatrix} \begin{bmatrix} \dot{X}_i \\ \dot{U}_M^i \end{bmatrix} + \begin{bmatrix} [\omega_i^2] & 0 \\ 0 & K_{MM} - K_{Mi} K_{ii}^{-1} K_{iM} \end{bmatrix} \begin{bmatrix} X_i \\ U_M^i \end{bmatrix} = \begin{bmatrix} 0 \\ R_M^i \end{bmatrix} \quad (3.6)$$

全部子结构组集后的整体方程为

$$\begin{bmatrix} I & -\phi_1^T M_{11} K_{11}^{-1} K_{1M} \\ & I & -\phi_2^T M_{22} K_{22}^{-1} K_{2M} \\ & & \dots & \vdots \\ & & I & -\phi_N^T M_{NN} K_{NN}^{-1} K_{NM} \\ \text{对称} & M_{MM} + \sum_{i=1}^N K_{Mi} K_{ii}^{-1} M_{ii} K_{ii}^{-1} K_{iM} \end{bmatrix} \begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \\ \vdots \\ \dot{X}_N \\ \dot{U}_M \end{bmatrix} + \begin{bmatrix} [\omega_1^2] & & & & \\ & [\omega_2^2] & & & \\ & & \dots & & \\ & & & [\omega_N^2] & \\ & & & & K_{MM} - \sum_{i=1}^N K_{Mi} K_{ii}^{-1} K_{iM} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \dots \\ X_N \\ U_M \end{bmatrix} = \{0 \ 0 \ \dots \ 0 \ 0\}^T \quad (3.7)$$

上标 $T$ 表示转置，简记为

$$[M]\{\dot{X}\} + [K]\{X\} = \{0\} \quad (3.8)$$

这样，在并行机上实施运算时，仍把各子结构分配到各自的CPU中，执行并行运算。

子结构模态综合法并行算法的实施步骤是

1. 将结构划分为 $N$ 个子结构，按照先内部后界面的顺序进行编号，输入有关信息。
2. 各CPU并行形成子结构的刚度矩阵与质量矩阵，有

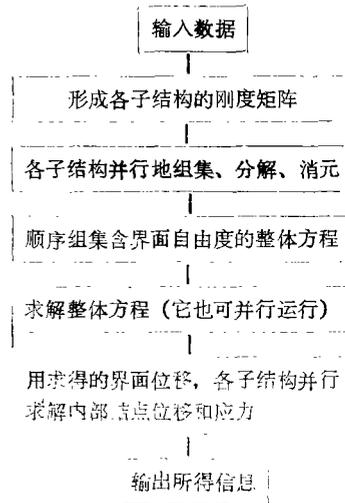
$$\begin{bmatrix} M_{ii} & M_{iM} \\ M_{Mi} & M_{MM} \end{bmatrix}, \quad \begin{bmatrix} K_{ii} & K_{iM} \\ K_{Mi} & K_{MM} \end{bmatrix}$$

3. 各CPU对子结构并行执行分解消元

$$\begin{bmatrix} D_{ii} L_i^T & L_i^{-1} K_{iM} \\ 0 & K_{MM} - K_{Mi}^T K_{ii}^{-1} K_{iM} \end{bmatrix}$$



在的串行性质（因不同的子结构的信息将不可避免地要组集到同一整体数组中），所以，并行程序中的组集功能将与串行程序一样。根据上面介绍的基本准则，我们设计了可在 ELXSI 计算机系统上实施的并行程序，此程序可由下面的简图略作说明。



显然，在这种子结构解法中还存在许多影响并行效率的方面。例如，各子结构的计算工作量应尽可能地一致，这可以减少那些提前完成计算的 CPU 的闲置时间。所以，网络的划分在并行计算中起着很重要的作用。

### 参 考 文 献

- [1] 西安交通大学计算中心, 《ELXSI 6400 上机使用手册》(内部资料)(1989)。
- [2] 张汝清、董 明, 《结构计算程序设计》, 重庆出版社(1989)。
- [3] 张汝清等, 《计算结构动力学》, 重庆大学出版社(1987)。

## Parallel Computational Algorithm of Substructure Method of Large-Scale Structure Analysis

Zhang Ru-qing

(Department of Engineering Mechanics, Chongqing University, Chongqing)

### Abstract

In this paper, according to the parallel environment of ELXSI computer, a parallel solving process of substructure method in static and dynamic analyses of large-scale and complex structure has been put forward and the corresponding parallel computational program has been developed.

**Key words** structural analysis, substructural solution, parallel algorithm