应用数学和力学，第13卷第4期(1992年4月)
Applied Mathematics and Mechanics

应用数学和力学编委会编
重庆出版社出版

一种迭代格式的有限元并行算法*

胡 宁　　张汝清

(重庆大学工程力学系，1990年10月11日收到)

摘　要

本文提出了一种求解有限元方程的迭代格式的并行算法。该方法在线性代数方程迭代解法的基础上，引进并行运算步骤；并且运用加权残数方法，通过选择适当的权函数，推导了该并行算法的有限元基本格式。该方法在西安交通大学ELXSI-6400并行计算机上程序实现，计算结果表明它能有效地提高运算速度，减少计算时间，是一种有效的求解大型结构有限元方程的并行算法。

关键词　并行算法　有限元方法　迭代解法　加权残值法

一、引　　言

现在广泛应用的常规的有限元方法及其应用软件都是在单CPU的串行计算机上实现的。由于计算机硬件领域的飞速发展和对工程结构分析所提出的越来越高的要求，为了更好地利用新一代多CPU并行计算机的潜在计算能力，人们正在各个计算机应用领域作出不懈的努力以改变现存的计算模式来适应这一计算机发展的新的趋势。目前，这种适应于并行计算机的并行算法及其应用的研究已成为数值计算领域内一个潜力很大并迅速发展的独立分支。国外除创刊了专业性的并行计算的国际杂志以外，还召开了多次并行计算的国际会议。在计算力学领域内，并行计算的研究和推广的速度更为惊人，目前，国外已出现了一些在特定的并行机上实现的商业化的大型有限元结构分析软件。

目前，相应于多CPU并行机发展起来的有限元方程的并行求解方法主要可归纳为两大类：一类是直接的并行解法，Charbel Farhat, Edward Wilson[1]和D. Goehlich, L. Komzsik, R. E. Fulton[2]等人都作过这方面的工作；另一类是迭代格式的并行解法，G. F. Carey, E. Barragy, R. Mclay, M. Sharma[3], R. B. King, V. Sonnad[4], T. J. R. Hughes, I. Levit, J. Winget[5]等人在这一领域进行过研究。直接解法的主要优点是数值解的稳定性很好，但最大的弱点是要引入同步控制，将会浪费宝贵的CPU时间；迭代算法的主要优点是不需要引入太多的同步运算，有些迭代并行算法甚至为完全异步控制，因此如果在算法的稳定性好的情况下，迭代算法比直接算法具有更大的优势。目前，主要的迭代并行算法一般都是对串行算法，加共轭梯度方法(conju

* 西安交通大学国家结构振动与强度重点实验室国家基金资助项目。

gate gradient method)[3][4][5]，进行改进．本文从Jacobi块迭代法出发，推导了基本上异步控制的有限元方程的并行解法，并且从加权残数法出发导出了并行有限元解法的基本格式．

二、ELXSI—6400并行机系统

并行计算的计算格式和编程方式取决于所用并行机的并行环境．目前存在的各种类型的并行机可主要归纳为共享内存(shared memory)和信息传递式(message-passing)两大类．ELXSI—6400并行计算机属于程序级的共享内存式并行机，程序中可以并行的若干部份都以子程序的面目出现．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的形式实现的．

从系统结构上讲，该机采用总线结构．CPU，内存，外设都挂在320M byte/s的高速总线上．如图1所示，系统进程及用户进程可分布于所有CPU上，不同CPU上运行的进程

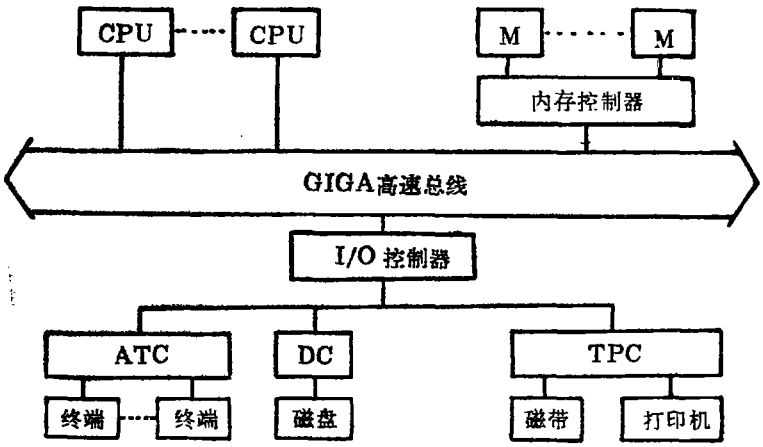

图 1

可以共享内存单元和磁盘文件．系统中提供了用于并行处理的函数形式的指令，程序设计者利用这些指令来实现运行程序的并行性能．

三、迭代格式的有限元并行算法

1. Jacobi块迭代法的扩展形式

有限元方程的基本形式如下：

$$[K]\{A\}=\{F\} \tag{3.1}$$

这一线性代数方程存在着许多迭代解法[6]，我们从Jacobi块迭代法出发，将$[K]$分成许多对角块和非对角块，其中任一块的迭代格式为

$$[K_{ii}]\{A_i^q\}=-\sum_{j=1}^{n}[K_{ij}]_{j\neq i}\{A_j^{q-1}\}+\{F_i\} \qquad (i=1,\ 2,\ \cdots,\ n) \tag{3.2}$$

上式中q为迭代次数，将形式改写为便于运算的形式，令

$$\left.\begin{aligned}&[K_{ii}]\{A_i^1\}=\{F_i\} \qquad (i=1,2,\cdots,n)\\&\{A_i^0\}=0 \qquad\qquad\quad (i=1,2,\cdots,n)\end{aligned}\right\} \tag{3.3}$$

将(3.2)改写成

$$[K_{ii}]\{A_i^q\}=-\sum_{j=1}^{i}[K_{ij}]_{j\neq i}\{A_j^{q-1}\}+[K_{ii}]\{A_i^1\}+\sum_{j=1}^{n}[K_{ij}]_{j\neq i}\{A_j^q\}$$

$$=-\sum_{j=1}^{n}[K_{ij}]_{j\neq i}\{A_j^{q-1}\}+\Big(-\sum_{j=1}^{n}[K_{ij}]_{j\neq i}\{A_j^1\}+[K_{ii}]\{A_i^1\}$$

$$+\sum_{j=1}^{n}[K_{ij}]_{j\neq i}\{A_j^0\}\Big)+\sum_{j=1}^{n}[K_{ij}]_{j\neq i}\{A_j^1\} \tag{3.4}$$

令 $$[K_{ii}]\{A_i^2\}=-\sum_{j=1}^{n}[K_{ij}]_{j\neq i}\{A_j^1\}+\Big([K_{ii}]\{A_i^1\}+\sum_{j=1}^{n}[K_{ij}]_{j\neq i}\{A_j^0\}\Big) \tag{3.5}$$

代入(3.4)式，可得

$$[K_{ii}]\{A_i^q\}=-\sum_{j=1}^{n}[K_{ij}]_{j\neq i}\{A_j^{q-1}\}+[K_{ii}]\{A_i^2\}+\sum_{j=1}^{n}[K_{ij}]_{j\neq i}\{A_j^1\} \tag{3.6}$$

重复(3.4)和(3.5)式的过程多次，最后可得

$$[K_{ii}]\{A_i^q\}=-\sum_{j=1}^{n}[K_{ij}]_{j\neq i}\{A_j^{q-1}\}+\Big([K_{ii}]\{A_i^{q-1}\}+\sum_{j=1}^{n}[K_{ij}]_{j\neq i}\{A_j^{q-2}\}\Big) \tag{3.7}$$

其中，

$$[K_{ii}]\{A_i^{q-1}\}=-\sum_{j=1}^{n}[K_{ij}]_{j\neq i}\{A_j^{q-2}\}+\Big([K_{ii}]\{A_i^{q-2}\}+\sum_{j=1}^{n}[K_{ij}]_{j\neq i}\{A_j^{q-3}\}\Big) \tag{3.8}$$

可以看出(3.5)，(3.7)，(3.8)为同一形式的递推公式，在方程(3.7)中，令

$$\{r_i^q\}=\sum_{j=1}^{n}[K_{ij}]_{j\neq i}\{A_j^{q-1}\}-\sum_{j=1}^{n}[K_{ij}]_{j\neq i}\{A_j^{q-2}\} \tag{3.9}$$

代入(3.7)式可得

$$[K_{ii}]\{A_i^q\}=[K_{ii}]\{A_i^{q-1}\}-\{r_i^q\} \tag{3.10}$$

如令

$$\{A_i^q\}=\{A_i^{q-1}\}+\{\Delta_i^q\} \tag{3.11}$$

则方程(3.10)最后写成

$$[K_{ii}]\{\Delta_i^q\}=-\{r_i^q\} \tag{3.12}$$

方程(3.9)则变成

$$\{r_i^q\}=\sum_{j=1}^{n}[K_{ij}]_{j\neq i}\{\Delta_j^{q-1}\} \tag{3.13}$$

因此相应于Jacobi迭代解法的变换形式由方程(3.11)，(3.12)，(3.13)式可知，整个迭代过程为

$$\begin{cases} [K_{ii}]\{\Delta_i^q\}=-\{r_i^q\} & (3.14)\\ \{A_i^q\}=\{A_i^{q-1}\}+\{\Delta_i^q\} & (3.15)\\ \{r_i^{q+1}\}=\sum_{j=1}^{n}[K_{ij}]_{j\ i}\{\Delta_j^q\} & (3.16)\\ \{r_i^0\}=\{F_i\} & (3.17)\end{cases} \quad (i=1,2,\cdots,n)$$

上面的迭代格式很容易改写成并行格式．首先，将分块数n定为CPU的个数，然后将n个$[K_{ii}]$放到各CPU中进行分解，这一步为并行执行，分解后的$[K_{ii}]$留着后面的回代用，然后将上面(3.14)到(3.17)的第i个迭代过程放到第i个CPU中，成为第i个并行进程，这样总有n个并行迭代进程．每个过程主要有下面4个方面：(1)在方程(3.14)中，通过前后回代求得$\{\Delta_i^q\}$；(2)在方程(3.15)中更新位移矢量；(3)接收其它迭代过程传递来的$\{\Delta_i^q\}$，求下一步的$\{r_i^{q+1}\}$；(4)传递$\{\Delta_i^q\}$到其它并行迭代进程．上面4步中的(1)，(2)两步，是i进程中完全独立于其它进程的运算步，而(3)，(4)两步则与其它进程有关，所传递和接收信息的过程可以通过共享内存实现．

2. 加权残值法推导的有限元并行迭代格式

对区域$\Omega(x)$，线性边值问题可表示成，在希尔伯特空间(Hilbert space)求解$u(x)$，并使

$$a(u,\ w)=(f,\ w)\qquad w\in H(\Omega) \tag{3.18}$$

有限元方法是通过有限维子空间来逼近$H(\Omega)$空间寻求近似解．伽辽金方法是通过选择一子空间Φ，使在Φ中找到φ，使$a(\varphi,\ w)=(f,\ w)$，$w\in\Phi$成立．再通过由形函数构成的一组基$\{B_i\}_1^N$(N为φ的维数)来构造φ，

$$\varphi=\sum_{i=1}^{N}A_iB_i=\{A\}^T\{B\}$$

最后问题变成

$$a(\varphi,B_l)=(f,B_l)\qquad (l=1,2,\cdots,N) \tag{3.19}$$

然后利用构造许多数学问题近似解的映射方法[7]设有界映射P，它是一个定义于正交线性空间，并满足$P^2=P$的线性算子．线性空间中的φ可表示成$\varphi=P\varphi+Q\varphi$，其中$Q=I-P$，$P\varphi$称为$\varphi$的映射，$Q\varphi$为相应的余值或残值．考虑基$\{B_j\}_1^N$的所有非空子集的集合（含$(2^N-1)$个元素），并由$\{b^{(i)}\}$表示，任一$b^{(i)}$都具有形式$\{B_{l_1},\ B_{l_2},\cdots,B_{l_s}\}$，其中$l_1,l_2,\cdots,l_s$为1到$N$的整数，维数为$s$的每个$b^{(i)}$复盖$\Phi$中一子空间$\Phi_i$．因此对每个$b^{(i)}$，$\varphi$的映射为

$$P\varphi=\sum_{\substack{j=1\\ B_j\notin b^{(i)}}}^{N}A_jB_j \tag{3.20}$$

上式定义的映射的集合为γ，其中每个P_i有特性，$\varphi=P_i\varphi+Q_i\varphi$，如设$P_i\varphi\neq 0$，故有$\|Q_i\varphi\|<\|\varphi\|$，其中$\|\varphi\|=\sum_{i=1}^{N}|A_i|$．考虑一$\varphi\in\Phi$和$P_i\in\gamma$，故有$\varphi=P_i\varphi+Q_i\varphi$，而$Q_i\varphi\in\Phi$可以由另

一个子空间Φ_j和映射$P_j\in\gamma$继续分解，有

$$\varphi=P_i\varphi+P_j(Q_i\varphi)+Q_j(Q_i\varphi) \tag{3.21}$$

并且有$\|Q_j(Q_i\varphi)\|<\|Q_i\varphi\|<\|\varphi\|$，这意味着残值变小，由$E_{i+1}\varphi$代表残值$E_i\varphi$映射于$\Phi_i$后的残值，有

$$E_i\varphi=P_i(E_i\varphi)+(I-P_i)E_i\varphi=P_iE_i\varphi+E_{i+1}\varphi \tag{3.22}$$

上式结合(3.21)式，连续施行一组$\{P_1,P_2,\cdots,P_n\}$映射得

$$\varphi=\sum_{i=1}^{n}P_iE_i\varphi+E_{n+1}\varphi=\varphi_n+E_{n+1}\varphi \tag{3.23}$$

其中φ_n为具有残值$E_{n+1}\varphi$的φ的近似值，其表达式为

$$\varphi_n=P_1\varphi+P_2(I-P_1)\varphi+\cdots+P_n(I-P_{n-1})\cdots(I-P_1)\varphi \tag{3.24}$$

$$\varepsilon_n=E_{n+1}\varphi=(I-P_n)(I-P_{n-1})\cdots(I-P_1)\varphi \tag{3.25}$$

如果$\{P_1,\cdots,P_n\}$满足$P_jE_i\varphi\neq0$和$P_j\neq P_jP_i$，那么当$n\to\infty$时有，$\lim\limits_{n\to\infty}\varphi_n=\varphi$，$\lim\limits_{n\to\infty}\varepsilon_n=0$。将方程(3.23)改写为迭代格式，有

$$\varphi_n=(I-P_n)\varphi_{n-1}+P_n\varphi \tag{3.26}$$

而有限元格式(3.19)可写成

$$T_l\varphi=r_l\qquad(l=1,2,\cdots,N) \tag{3.27}$$

这里T_l为Φ上一线性函数，$T_l\varphi=a(\varphi,B_l)$，$r_l=(f,B_l)$由方程(3.24)，$\varphi$由$\varphi_n$逼近，有

$$\lim_{n\to\infty}T_l\varphi_n=r_l\qquad(l=1,2,\cdots,N) \tag{3.28}$$

而 $\varphi=\varphi_n+\varepsilon_n=Q_n\varphi_{n-1}+P_n\varphi+\varepsilon_n$，代入(3.27)式有

$$T_l\{P_n\varphi\}=r_l-T_lQ_n\varphi_{n-1}-T_l\varepsilon_n \tag{3.29}$$

因此求解满足方程(3.28)的$\{\bar{\varphi}_n\}$的摄动方程为

$$T_l(P_n\eta_n)=r_l-T_l(Q_n\bar{\varphi}_{n-1}),\quad\bar{\varphi}_n=(I-P_n)\bar{\varphi}_{n-1}+P_n\eta_n \tag{3.30}$$

设有限单元数为N_e，$\{N\}$代表局部基矢量，$[C]$为整体几何阵，$\{A\}$为φ的参数构成的矢量，因此Φ的每一φ可表示成$\varphi=\{N\}^T[C]\{A\}$。$\{d\}$代表自由度矢量，$\{\tilde{d}^{(n)}\}$代表被映射P_n复盖的自由度，故有$\{\tilde{d}^{(n)}\}=[D^{(n)}]\{d\}$，$[D^{(n)}]$是每一行只有一非零元素（单位值）构成的矩阵，映射$P_n$可表示成

$$P_n\varphi=\{N\}^T[C][D^{(n)}]^T[D^{(n)}]\{A\} \tag{3.31}$$

设$\{A_n\}$为φ在第n步近似值φ_n的参数构成的矢量，

$$\varphi_n=\{N\}^T[C]\{A_n\} \tag{3.32}$$

故φ_n的映射P_m可表示成

$$P_m\varphi_n=\{N\}^T[C][D^{(m)}]^T[D^{(m)}]\{A_n\} \tag{3.33}$$

将(3.31)，(3.32)，(3.33)式代入(3.26)式得迭代逼近方程

$$\{A_n\}=\{A_{n-1}\}+[D^{(n)}]^T\{\Delta^{(n)}\} \tag{3.34}$$

其中 $\{\Delta^{(n)}\}=[D^{(n)}](\{A\}-\{A_{n-1}\})$

设一个边值问题的表达式如下，

$$A(\psi)=L\psi+p=0\qquad\text{对区域}\Omega \tag{3.35}$$

$$B(\psi)=M\psi+q=0\qquad\text{对边界}\Gamma \tag{3.36}$$

设加权函数为W^l和$\overline{W}^l$ $(l=1,2,\cdots,N)$，由伽辽金方法，

$$\int_\Omega W^l L\varphi d\Omega+\int_\Gamma \overline{W}^l M\varphi d\Gamma=-\int_\Omega W^l p d\Omega-\int_\Gamma \overline{W}^l q d\Gamma \tag{3.37}$$

摄动方程(3.30)式变成，

$$\int_\Omega W_n^l LP_n\eta_n d\Omega+\int_\Gamma \overline{W}_n^l MP_n\eta_n d\Gamma=-\int_\Omega W_n^l L(I-P_n)\bar{\varphi}_{n-1}d\Omega-\int_\Omega W_n^l p d\Omega$$

$$-\int_\Gamma \overline{W}_n^l M(I-P_n)\bar{\varphi}_{n-1}d\Gamma-\int_\Gamma \overline{W}_n^l q d\Gamma \qquad (l=1,2,\cdots,N) \tag{3.38}$$

我们将加权函数选成如下形式，

$$[W_n^l]=[D^{(n)}][C]^T\{N\},\ [\overline{W}_n^l]=[D^{(n)}][C]^T\{\bar{N}\} \tag{3.39}$$

将(3.31)，(3.32，(3.33，(3.39)式代入(3.38，得到映射刚度方程，

$$[K^{(n)}]\{\Delta^{(n)}\}=-\{\Delta r^{(n)}\} \tag{3.40}$$

其中，

$$\left.\begin{aligned}
&[K^{(n)}]=[D^{(n)}][K][D^{(n)}]^T,\ \{\Delta r^{(n)}\}=[D^{(n)}]\{r^{(n)}\}\\
&\{r^{(n)}\}=\{r^{(n-1)}\}+[K][D^{(n-1)}]^T\{\Delta^{(n-1)}\}\\
&[K]=[C]^T\left\{\int_\Omega\{N\}L\{N\}^T d\Omega+\int_\Gamma\{\bar{N}\}M\{\bar{N}\}^T d\Gamma\right\}[C]\\
&\{r^{(0)}\}=[C]^T\{f_\Omega\}+[C]^T\{f_\Gamma\}\\
&\{f_\Omega\}=\int_\Omega\{N\}p d\Omega,\ \{f_\Gamma\}=\int_\Gamma\{\bar{N}\}q d\Gamma
\end{aligned}\right\} \tag{3.41}$$

再结合方程(3.34)，得最后的迭代格式，

$$\left.\begin{aligned}
&\{\Delta r^{(n)}\}=[D^{(n)}]\{r^{(n)}\},\ [K^{(n)}]\{\Delta^{(n)}\}=-\{\Delta r^{(n)}\}\\
&\{r^{n+1}\}=\{r^n\}+[K][D^{(n)}]^T\{\Delta^{(n)}\},\ \{A^{(n)}\}=\{A^{(n-1)}\}+[D^{(n)}]^T\{\Delta^{(n)}\}\\
&\{r^{(0)}\}=[C]^T\{f_\Omega\}+[C]^T\{f_\Gamma\},\ \{\Delta^{(0)}\}=0
\end{aligned}\right\} \tag{3.42}$$

令第 i 个映射阵可表示成

$$[D_i]_{n\tilde{d}_i\times n_d}=[[0]_{n\tilde{d}_i\times n\tilde{d}_1}|[0]_{n\tilde{d}_i\times n\tilde{d}_2}|\cdots|[I]_{n\tilde{d}_i\times n\tilde{d}_i}|\cdots|[0]_{n\tilde{d}_i\times n\tilde{d}_{N_p}}] \tag{3.43}$$

N_p为映射个数，$n\tilde{d}_i$为第 i 个映射的维数，n_d是所有近似值的个数，映射阵同时满足

$$\{[D_1]^T\ [D_2]^T\ \cdots\ [D_i]^T\ \cdots\ [D_{N_p}]^T\}=[I]_{n_d\times n_d} \tag{3.44}$$

令$[D^{(n)}]=[D_i]$，$[D^{(n-1)}]=[D_j]$，并利用方程(3.44)，则方程(3.41)可表示成

$$\{r_i^{(n)}\}=\{r_i^{(n-1)}\}+\sum_j[K_{ij}]\{\Delta_j^{(n-1)}\} \tag{3.45}$$

其中，$\{r_i\}=[D_i]\{r\}$，$[K_{ij}]=[D_i][K][D_j]^T$，$\{\Delta r_i^{(n)}\}=\{r_i^{(n)}\}$，同时，

$$\{A_i^{(n)}\}=\{A_i^{(n-1)}\}+\{\Delta_i^{(n)}\} \qquad (i=1,2,\cdots,N_p) \tag{3.46}$$

总刚度阵被分块成

$$[K]=\begin{bmatrix}
[K_{11}] & [K_{12}] & \cdots & [K_{1i}] & \cdots & [K_{1N_p}]\\
[K_{21}] & [K_{22}] & \cdots & [K_{2i}] & \cdots & [K_{2N_p}]\\
\vdots & & & & & \\
[K_{N_p1}] & [K_{N_p2}] & \cdots & [K_{N_pi}] & \cdots & [K_{N_pN_p}]
\end{bmatrix} \tag{3.47}$$

每一分块体系的迭代格式为

$$[K_{ii}]\{\Delta_i^q\}=-\{r_i^q\} \tag{3.48}$$

$$\{A_i^q\}=\{A_i^{q-1}\}+\{\Delta_i^q\} \tag{3.49}$$

$$\{r_j^{(q+1)}\}=[K_{ji}]_{j\neq i}\{\Delta_i^q\} \tag{3.50}$$

上面(3.50)式是由于$\{r_j^q\}+[K_{ji}]\{\Delta_i^q\}=0$，并由(3.45 推导而得.可以看出，上面诸式和前面由Jacobi 块迭代法推得的扩展形式完全一样，其并行计算的实施步骤也和前面所述一样.

四、并行程序设计

并行程序的设计取决于并行机的特性. ELXSI—6400机的程序并行实现主要依靠并行子程序的同时执行实现的. 各并行子程序（并行进程）之间的同步和互斥依靠进程之间的信息传递. 该机提供了两种并行方式：

（1） 垂直方式，它适用于同一程序处理多组独立的数据的情况.

（2） 水平方式，主要指不同程序处理同一组数据的情况.

在确定了程序结构和并行环境之后，利用 ELXSI 公司提供的并行语句，最后编写程序. ELXSI公司提供的并行语句包括：建立若干子进程，撤消所有子进程，为并行进程设置共享的内存页，建立"锁"和"信号量"等多种功能，为程序设计者提供了方便. 图 2 为程序的流程图，值得注意的是方程(3.16)求下一步的残值$\{r_i^{q+1}\}$时由于要用到其它进程所计算的$\{\Delta_j^q\}$，因此在此需引入同步功能，ELXSI机的同步过程由信号量MT$SIGNALSEMAPHORE和MT$WAITONSEMAPHORE两种功能实现.

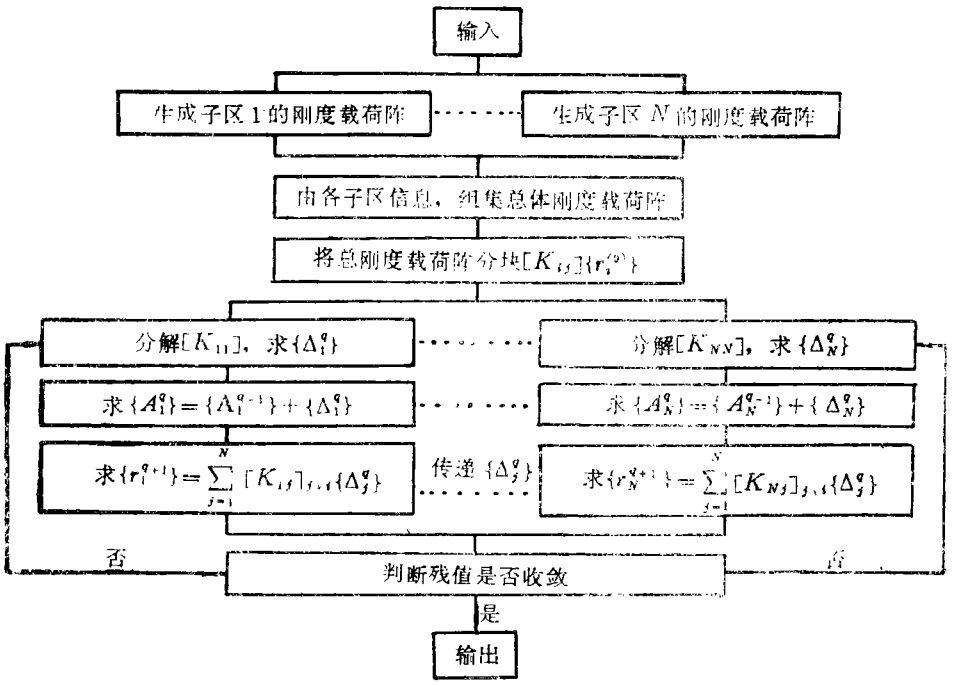

图 2

五、计 算 试 例

为了对该方法的效率进行分析，可将串行与并行方法对同一试例分析时所耗的CPU 时

间进行比较．试例如图3所示，图4反映了两种方法随结构自由度变化的情况，并同在两个CPU情况节省时间的理想曲线进行了比较（西安交通大学ELXSI机目前自配备了两个CPU）．从图中可以看出，随着结构自由度的上升，所相对节约的时间也越多，越接近理想线．

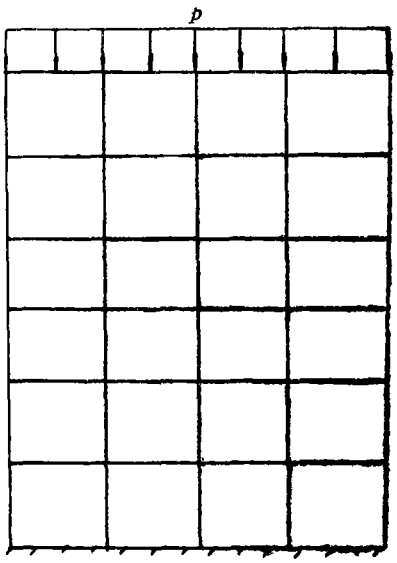

图 3

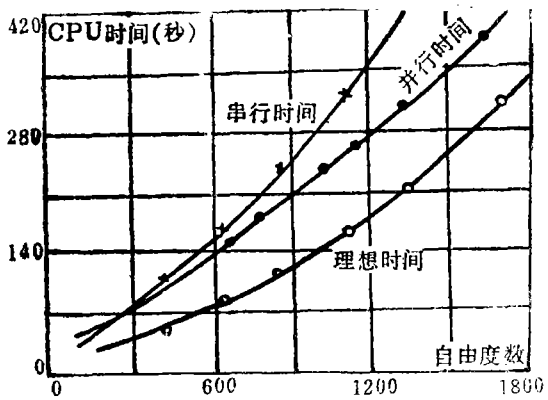

图 4

六、结 论

上面我们分别从Jacobi块迭代法和加权残值法出发，推导了迭代格式的有限元并行算法，计算分析表明该算法能够有效地提高运算速度，节省计算时间，是一种有效的大型结构有限元并行分析方法．

参 考 文 献

[1] Farhat, Charbel and Edward Wilson, A parallel active column equation solver, *Computers & Structures*, **28**(2) (1988), 289—304.

[2] Goehlich, D., L. Komzsik and R. E. Fulton, Application of a parallel equation solver to static FEM problems, *Computers & Structures*, **31**(2) (1989), 121—129.

[3] Carey, G. F., E. Barragy, R. Mclay and M. Sharma, Element-by-element vector and parallel computation, *Communications in Applied Numerical Method*, **4** (1988), 299—307.

[4] King, R. B. and V. Sonnad, Implementation of an element-by-element solution algorithm for the finite element method on a coarse-grained parallel computer, *Computer Methods in Applied Mechanics and Engineering*, **65** (1987), 47—59.

[5] Hughes, T. J. R., I. Levit and J. Winget, An element-by-element solution algorithm for problems of structural and solid mechanics, *Computer Methods in Applied Mechanics and Engineering* (1983), 241—254.

[6] Hageman, Louis A. and David M. Young, *Applied Iterative Methods*, Academic Press (1982).

[7] Krasnosel'skll, G. M., et al., *Approximate Solution of Operator Equations*, Wolters-Noordhoff Publishing, Groningen, The Netherlands (1972).

An Iterative Parallel Algorithm of Finite Element Method

Hu Ning Zhang Ru-qing

(*Department of Engineering Mechanics, Chongqing University, Chongqing*)

Abstract

In this paper, a parallel algorithm with iterative form for solving finite element equation is presented. Based on the iterative solution of linear algebra equations, the parallel computational steps are introduced in this method. Also by using the weighted residual method and choosing the appropriate weighting functions, the finite element basic form of parallel algorithm is deduced. The program of this algorithm has been realized on the ELXSI—6400 parallel computer of Xi'an Jiaotong University. The computational results show the operational speed will be raised and the CPU time will be cut down effectively. So this method is one kind of effective parallel algorithm for solving the finite element equations of large-scale structures.

Key words parallel algorithm, finite element method, iterative solution, weighted residual method