应用数学和力学，第16卷 第11期（1995年11月）
Applied Mathematics and Mechanics

应用数学和力学编委会编
重庆出版社出版

瞬态动力问题的逐单元矩阵分裂和逐步积分方法*

王怀忠

（上海大学，上海市应用数学和力学研究所，上海 200072）

（潘立宙推荐，1994年7月7日收到）

摘 要

本文对瞬态动力问题，结合逐步积分方法提出了一类广义的矩阵分裂和逐单元松弛算法，摆脱了有限元法通常需形成总体刚度矩阵，总体质量矩阵和求解大型稀疏方程组的工作，理论分析和计算实例表明，本文的广义矩阵分裂是最优的分裂方案。本文的算法物理意义明确，便于编写程序推广应用。

关键词 有限元法 逐步积分 矩阵分裂 逐单元松弛

一、引 言

用有限单元和逐步积分方法相结合分析动荷载作用下结构与介质的动力相互作用时，需要进行空间和时间的双重离散。

逐步积分法是将所关心的时间域离散为一系列短的时步Δt，在每个时步的起点和终点满足动力平衡条件，并以一个设定的积分格式为依据，近似地计算时步内体系的运动，一个时步终点的运动状态作为下一个时步的初始条件，这样逐步递推可求得整个过程的解。在运用逐步积分法计算时人们通常采用位移为基本变量，本文参照[5]采用加速度为基本变量，这样也避免了导出变量误差病态积累。

有限单元法作为Ritz-Galerkin法的推广，是将具有物理意义的空间区域离散为许多小的单元，在每个单元中采用多项式作为试探函数，单元之间在边界处满足一定的连续性。每个单元可形成刚度矩阵，质量矩阵和荷载矢量，并按节点位移的定位关系集总形成总体刚度矩阵，质量矩阵和荷载矢量。通过求解以总体刚度矩阵和质量矩阵为系数矩阵的大型稀疏方程组可得问题的解。

形成总体系数矩阵和求解大型稀疏方程组需要很大的工作量和很大的内存量，因而成为制约有限元法应用的一个重要因素。为摆脱这两项工作，本文引入定位矩阵并由此导出单元系数矩阵同总体系数矩阵之间集总和分裂的运算关系，推广了[3]中收敛速度较快的广义矩阵分裂逐块松弛算法，实现了逐单元松弛迭代求解。

二、空间离散及逐单元矩阵分裂

对边界Γ包围的线弹性介质Ω，不计材料阻尼，考虑惯性力影响，可得平衡微分方程：

$$-S^TDSu+\rho\ddot{u}=f \qquad (\text{在}\Omega\times(0,T)\text{内}) \tag{2.1}$$

边界条件为：

$$u=\bar{u} \qquad (\text{在}\Gamma_1\text{上}) \tag{2.2a}$$

$$Hu=\bar{\sigma} \qquad (\text{在}\Gamma_2\text{上}) \tag{2.2b}$$

$\Gamma=\Gamma_1\cup\Gamma_2$，初值条件为：

$$u_0=u|_{t=0},\ \dot{u}_0=\dot{u}|_{t=0} \tag{2.2c}$$

S为应变算子，D为弹性矩阵，H为边界应力算子.

(2.1)～(2.2)一般不能得到封闭的解析解，考虑其弱形式，(2.1)～(2.2)等价于：

求$u\in H^1_\bar{u}(\Omega)$使$u-\bar{u}\in H^1_0(\Omega)$且满足：

$$B(u,v)+(\rho\ddot{u},v)=(f,v)+(\bar{\sigma},v) \qquad (\forall v\in H^1_0(\Omega)) \tag{2.3}$$

其中

$$B(u,v)=\int_\Omega(Su)^TD(Sv)dx \tag{2.4}$$

$$(\rho\ddot{u},v)=\int_\Omega\ddot{u}^T\rho v dx \tag{2.5}$$

$$(f,v)=\int_\Omega u^Tf dx \tag{2.6}$$

$$(\bar{\sigma},\ v)=\int_{\Gamma_2}u^T\bar{\sigma}ds \tag{2.7}$$

首先对$u(x,t)$进行空间有限元离散，令：

$$u(x,t)\approx u_h(t)=u^i_h \tag{2.8}$$

其中 u^i_h为分片多项式：

$$u^i_h=N_id^i_i \qquad (\forall x\in e(i),i=1,\cdots,n_e) \tag{2.9}$$

$$d^i_i=Q_id^t \tag{2.10}$$

式中 N_i，Q_i，d^i_i分别为单元i的形函数矩阵，定位矩阵和节点位移向量，d^t为总体位移向量，$e(i)$为单元i所占区域，n_e为总单元数．以下除特别注明外，简记$\sum_{i=1}^{n_e}$为$\sum$．由(2.3)～(2.10)得：

$$\sum\int_{e(i)}(SN_iQ_i)^TD(SN_iQ_i)d^tdx+\sum\int_{e(i)}(N_iQ_i)^T\rho N_iQ_i\ddot{a}^tdx$$
$$=\sum\int_{e(i)}(N_iQ_i)^Tfdx+\sum\int_{s(i)}(N_iQ_i)^T\bar{\sigma}ds \tag{2.11}$$

其中 $s(i)=\Gamma_2\cap\Gamma(i)$，$\Gamma(i)$为单元$i$边界．对每个单元$i$有：

$$K_i=\int_{e(i)}(SN_iQ_i)^TD(SN_iQ_i)dx \tag{2.12}$$

$$M_i=\int_{e(i)}(N_iQ_i)^T\rho N_iQ_idx \tag{2.13}$$

$$F_i^t=\int_{e(i)}(N_iQ_i)^T f\,dx+\int_{S(i)}(N_iQ_i)^T\bar{\sigma}\,ds \tag{2.14}$$

形成总体刚度矩阵，总体质量矩阵和总体荷载向量的运算为：

$$K=\sum Q_i^T K_i Q_i \qquad M=\sum Q_i^T M_i Q_i \qquad F^t=\sum Q_i^T F_i^t \tag{2.15}$$

则(2.11)可表示为常见的大型微分方程组：

$$Kd^t+Ma^t=F^t \tag{2.16}$$

由(2.10)(2.15)可将(2.16)化为：

$$\sum Q_i^T K_i d_i^t+\sum Q_i^T M_i a_i^t=\sum Q_i^T F_i^t \tag{2.17}$$

令 $K_{ij}=Q_jQ_i^T K_i \qquad M_{ij}=Q_jQ_i^T M_i \qquad F_{ij}^t=Q_jQ_i^T F_i^t$

将方程(2.17)左乘Q_j得逐单元分裂矩阵形式的微分方程组：

$$\sum K_{ij}d_i^t+\sum M_{ij}a_i^t=\sum F_{ij}^t \qquad (j=1,\cdots,n_e) \tag{2.18}$$

显然(2.18)是(2.16)的分裂形式。

注记：

1. Q_i为单元i的定位矩阵，它反映了单元位移向量与总体位移向量的对应关系。
2. 若单元i，j相邻，d_i^t与d_j^t有重叠项，$Q_iQ_i^T\neq 0$.
3. 若单元i，j不相邻，d_i^t与d_j^t无重叠项，$Q_iQ_i^T=0$，所以方程(2.18)只有很少的非零项.
4. $Q_iQ_i^T=I_i$，所以$K_{ii}=K_i$，$M_{ii}=M_i$.

三、时间离散及逐步积分法

对$u(x,t)$进行时间离散，将u_h^t进一步近似为u_h^n：

$$u_h^t\approx u_h^n \qquad (t=n\Delta t) \tag{3.1}$$

$$u_h^n=N_id_i^n \qquad (\forall x\in e(i),i=1,\cdots,n_e) \tag{3.2}$$

方程(2.16)和(2.18)分别化为(3.3)和(3.4)：

$$Kd^n+Ma^n=F^n \tag{3.3}$$

$$\sum K_{ij}d_i^n+\sum M_{ij}a_i^n=\sum F_{ij}^n \qquad (j=1,\cdots,n_e) \tag{3.4}$$

(3.3)和(3.4)为微分方程组，可采用逐步积分法求解，不妨以Newmark格式(3.5)为例来说明，

$$\left.\begin{aligned} d^{n+1}&=d^n+v^n\Delta t+[(0.5-\alpha)a^n+\alpha a^{n+1}]\Delta t^2\\ v^{n+1}&=v^n+[(1-\delta)a^n+\delta a^{n+1}]\Delta t\end{aligned}\right\} \tag{3.5}$$

令：$b_0=\Delta t\delta$，$b_1=\Delta t^2\alpha$，$b_2=\Delta t(1-\delta)$，$b_3=\Delta t$，$b_4=\Delta t^2(0.5-\alpha)$；

采用加速度为基本变量，将(3.5)代入(3.4)得：

$$\begin{aligned}&\sum(b_1K_{ij}+M_{ij})a_i^n\\&\quad=\sum F_{ij}^n-\sum K_{ij}(d_i^{n-1}+b_3v_i^{n-1}+b_4a_i^{n-1}) \qquad (j=1,\cdots,n_e)\end{aligned} \tag{3.6}$$

记：

$$M_{ij}^*=b_1K_{ij}+M_{ij}$$

$$(F_{ij}^n)^*=F_{ij}^n-K_{ij}(d_i^{n-1}+b_3v_i^{n-1}+b_4a_i^{n-1})$$

则方程(3.6)化为：

$$\sum M_{ij}^*a_i^n=\sum(F_{ij}^n)^* \qquad (j=1,\cdots,n_e) \tag{3.7}$$

同样，将(3.5)代入(3.3)可得：

$$M^{*}a^{n}=(F^{n})^{*} \tag{3.8}$$

其中：

$$M^{*}=b_1K+M$$

$$(F^{n})^{*}=F^{n}-K(d^{n-1}+b_3v^{n-1}+b_4a^{n-1})$$

显然(3.7)是(3.8)的分裂形式。

逐步解线性代数方程组(3.7)或(3.8)，再将a_i^n代入(3.4)即得定解问题的解。

注记：

1. 一般情况下，$|K_{ii}|=0$。
2. 只要$|M_{ij}|\neq 0$，（事实上条件还可以再放宽）则：M_{ij}*对称正定。

四、逐元松弛算法

常规的方法是解大型方程组(3.8)，本文以(3.7)的方式分裂(3.8)的系数矩阵，再逐元松弛所对应的子块进行计算。所分裂的各子块之间有重叠部分，而且各子块相重叠部分之和等于总体系数矩阵对应位置的值。解线性代数方程组(3.7)的逐元松弛算法为：

任给一初始猜测：$A^{n(0)}=(a_1^{n(0)},a_2^{n(0)},\cdots,a_{ne}^{n(0)})$，作序列$\{A^{n(k)}\}_0^{\infty}$：

$$M_{jj}^{*}\,a_j^{n(k+1)}=M_{jj}^{*}\,\bar{a}_j^{n(k)}+\omega R_j^{n(k+1)} \qquad (j=1,\cdots,n_e;\ k=1,2,\cdots) \tag{4.1}$$

其中 ω为松弛因子，$0<\omega<2$，残向量：

$$R_j^{n(k+1)}=\sum(F_{i,j}^{n})^{*}-\left(\sum_{i=1}^{j}M_{ij}^{*}\ \bar{a}_i^{n(k+1)}-\sum_{i=j+1}^{ne}M_{ij}^{*}\,\bar{a}_i^{n(k)}\right) \tag{4.2}$$

式(4.1)(4.2)中，对未知数向量冠以“－”，是因为相邻单元子块的未知数向量有重叠项。计算中对未知数向量的重叠项及时调整是加快收敛速度的关键。

式(4.1)本质是逐块地调整$A_j^{n(k)}$使残向量$R_j^{n(k)}$逐渐趋向零。

由于M_{jj}^{*}对称正定，所以(4.1)可解且收敛。

子块之间重叠部分越多，每松弛一次计算量就越大，但收敛速度越快；反之，子块之间重叠部分越少，每松弛一次计算量就越小，但收敛速度越慢。怎样才能最优地分裂，这是一个很重要的问题。本文的分裂方案可以说是最优的方案之一。这是因为：

1. 只有单元i，j相邻，d_i^n与d_j^n才有重叠项，且$Q_iQ_i^T\neq 0$。所以(3.7)的非零项很少，这样在保证一定重叠量的同时，很有效地将计算量减少到最低。

2. 各子块实际上是单元和相邻单元的刚度矩阵，质量矩阵和荷载向量，计算中避免了形成总体系数矩阵和总体荷载向量，本文算法物理意义明确，便于编写程序。

3. 各子块的阶数较低且稳定，便于采用有效的方法解低阶的子方程组，使用较小的计算机内存也能解大型问题。

4. 运动方程逐步积分过程中，每一个时步的解为下一个时步提供了较近似的初始迭代值，也使收敛加快。

五、数值结果

为了便于同解析解比较，这里以一维波的传播问题为例。图1为有限元网格和计算示意图。Newmark格式参数；$\alpha=0.25$，$\delta=0.50$；松弛因子取$\omega=1.25$；总时步数 NSTEP=

120；荷载为$\bar{\sigma}(t)=50(1.0-\cos(\varphi t))$，$0\leqslant\varphi t\leqslant\pi$，其他计算参数见表1.

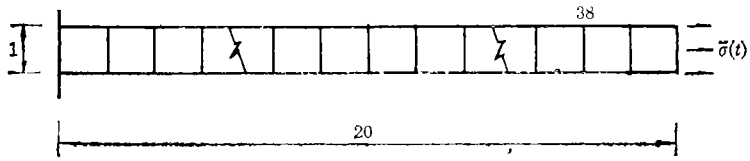

图1 有限元网格和计算示意图

表 1 计 算 参 数

E	υ	ρ	φ	Δt	$[\Sigma R]$
100.0	0.0	100.0	0.2π	0.50	0.001

表中$[\sum R]$为容许残差，参数实行了无量纲化．计算中只考虑沿x轴方向的运动

为节省篇幅，仅将节点38的加速度数值解与解析解的比较示于图2中，图3为时步1～4的$\sum R$随迭代次数IITER增加收敛至零的过程．

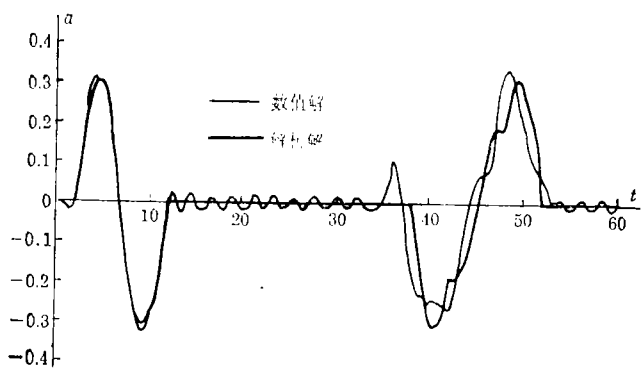

图2 节点38加速度时程曲线

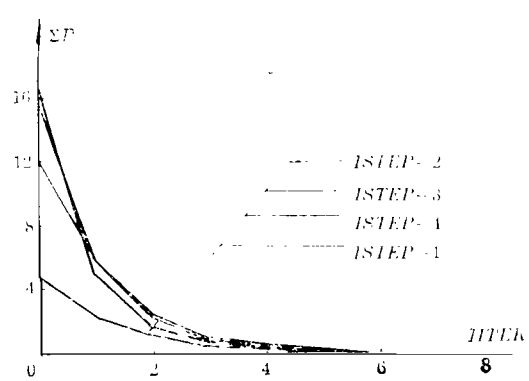

图3 ΣR收敛过程图

计算结果表明，在入射脉冲完全通过某一节点前，该节点的加速度的数值解与解析解基本重合，在其余时间段数值解有较小的振荡幅度，速度和位移的数值解与解析解相比误差较小，以至难以分辨出来．

图3表明，残差$\sum R$随迭代次数IITER 增加而呈负指数规律收敛至零，收敛速度较快，本例一般需10次迭代即满足精度要求，记载的最大迭代次数12次，最小迭代次数4次。

计算中采用用双精度实型使计算机舍入误差可忽略不计，采用了加速度为基本变量避免导出变量误差病态积累，且由于迭代收敛速度较快从而可有效地控制残差，所以误差主要是时间离散和空间离散带来的。

六、结 论

本文对瞬态动力问题，结合逐步积分方法提出了一类广义的矩阵分裂和逐单元松弛算法，摆脱了有限元法通常需要形成总体刚度矩阵，总体质量矩阵和求解大型稀疏方程组的繁重工作。该方法在保证一定子块重叠量以加速收敛的同时，很有效地将计算量减少到最低限度，便于采用有效的方法解低阶的子方程组，使用较小的计算机内存也能解大型问题，本文算法物理意义明确，便于编写程序推广应用。由于实现了逐元计算，有很大的灵活性，从而为时间—空间离散协调的工作奠定了基础。

参 考 文 献

[1] 钱伟长《变分法与有限元》（上），科学出版社(1980)。

[2] K. J. Bathe, *Finite Element Procedures in Engineering Analysis*, Prentice-Hall Inc. (1982).

[3] 康立山，全惠云等编，《数值解高维偏微分方程的分裂法》，上海科学技术出版社(1990)。

[4] O. C. Zienkiewicz, Computational mechanics today, *Int. J. Numer. Methods Eng.* 34(1992): 9—33.

[5] 王怀忠、赵毅，加速度为基本变量的逐步积分法，《中国博士后首届学术大会论文集》，国防工业出版社(1993)。

Element-by-Element Matrix Decomposition and Step-by-Step Integration Method for Transient Dynamic Problems

Wang Huaizhong

(Shanghai University of Technology, Shanghai Institute of Applied Mathematics and Mechanics, Shanghai 200072)

Abstract

In this paper, a general matrix decomposition scheme as well as an element-by-element relaxation algorithm combined with step-by-step integration method is presented for transient dynamic problems. Thus the finite element method can be free from forming global stiffness matrix, global mass matrix as well as solving large-scale sparse squations. Theory analysis and numerical results show that the present matrix decomposition scheme is the optimal one. The present algorithm has clear physical meaning and can be easily applied to finite element codes.

Key words finite element method, step-by-step integration, matrix decomposition, element-by-element relaxation