第 46 卷 第 3 期　　力 学 学 报　　Vol. 46，No. 3
2014 年 5 月　　Chinese Journal of Theoretical and Applied Mechanics　　May，2014

研究论文

统一格式的显式与隐式任意混合异步算法[1)]

张伟伟 *,†　金先龙 *,†,2)

*(上海交通大学机械与动力工程学院，上海 200240)

†(上海交通大学机械系统与振动国家重点实验室，上海 200240)

摘要 动力学问题的有限元分析需要在每一时步求解系统信息，相对于静力学问题，其计算量要大得多. 因而，提高计算效率，节省计算工作量是动力学求解方法研究的主要内容. 该文针对大型复杂动力学系统的高效求解问题，提出了一种基于 Newmark 离散格式的显式、隐式任意混合异步算法，根据整体系统不同局部的物理力学特性和求解精度要求，在空间域及时间域内对动力学系统方程进行多尺度求解. 该方法根据显式、隐式算法固有的信息传递机制，采取动态的可变边界处理方法，避免了异步边界上的误差积累；并通过对整体系统能量平衡的校验，动态地确定和修正仿真计算时步，可以有效地预防不稳定性的产生和发展. 数值算例表明：该算法能在保持较高的计算精度的同时，极大地降低计算资源消耗，因而具有一定的实用价值.

关键词 结构动力学，Newmark 离散，显式算法，隐式算法，任意混合，异步积分，算法稳定性

中图分类号：O242.2, O342　　**文献标识码**：A　　doi：10.6052/0459-1879-13-260

引 言

对工程实际中的动力学过程进行数值模拟，需要根据实际问题的固有属性来选择合理的积分算法. 当待求问题的非线性快速发展或响应中高频部分占主导时 (例如接触碰撞、波的传播等)，显式积分算法往往更受青睐 [1-3]；当待求问题为准线性问题或低频部分占主导时 (例如结构振动、冲击后的响应问题等)，无条件稳定的隐式算法则更加合适 [4-5]. 随着数值模拟方法的完善和计算机处理能力的进步，人们也需要更多地面对那些较为复杂的大规模问题，而这类问题往往是以上两种问题的组合，因此，研究显式、隐式混合求解算法是很有必要的.

事实上，选择合理积分方法的关键在于确保算法鲁棒性的同时提供足够的仿真精度，还要尽量提高计算效率. 显式算法由于计算稳定性的原因，需要采用较小的临界步长，但是，由于避免了迭代求解、显式算法不受收敛性的影响. 当待求问题属于高频成分占主导地位 (例如波的传播) 或相互作用时间极短的瞬态问题时，为了得到有意义的解答，必须采用较小的时间步长求解，这恰恰与显式算法步长受临界步长限制的要求是一致的. 反之，对于结构整体的动态响应问题，一般来讲，显式算法就显得不那么合适. 结构的动态响应通常是低频成分为主导，从计算精度上考虑，采用大步长的隐式算法是允许的，同时，结构动态响应问题的空间尺度往往较大，若计算时间步长太小，则整个过程的计算工作量将非常巨大，因此，对于结构动态响应问题，往往采用无条件稳定的隐式算法. 然而，隐式算法需要在每一时步进行矩阵求逆或迭代，耗费的计算资源较大. 在许多实际物理力学过程的不同阶段或不同区域，高频响应与低频响应总是分别出现，常常需要在数值计算中结合显式与隐式算法进行求解，以发挥不同算法的优势 [6-9]. 这时，又存在两种不同的情况：一种是需要在时间域内采用显式、隐式算法顺序求解；另一种情况是需要在不同空间域内采用不同算法.

对于第一种情况，较合理的处理方式是将显式和隐式算法整合到统一的程序中，根据待求问题的特征变化进行隐式算法和显式算法的切换，同时还需处理好显式、隐式切换时显式计算结果的数值震荡，最典型的应用案例便是冲压回弹分析 [10]. Noels 等 [11] 在该领域做了较多富有成效的工作. 一般情况下，类似于冲压、碰撞之类的瞬态冲击问题，适合采用显式算法求解，而随之而来的整体结构动态响应

2013–08–07 收到第 1 稿，2013–12–11 收到修改稿.

1) 国家高技术研究发展计划 (2012AA01AA307) 国家自然科学基金 (11072150, 61073088) 资助项目.

2) 金先龙, 教授，主要研究方向：结构动力学的数值计算方法及应用、并行计算. E-mail: jxlong@sjtu.edu.cn

或回弹过程，往往是低频的响应占主导地位，一般采用隐式求解. 传统的显式隐式序列求解往往需要人为地设定转换的时间节点，且加速度等信息都被重置为零. 自动的显式、隐式切换出现于文献 [12]，显式、隐式的切换通过显式算法和隐式算法求解一个时间步所需的 CPU 时间比率来确定. 同时，算法切换时的初始条件对于计算稳定性和收敛性也非常重要，一般通过能量平衡的处理方式 [13]. 目前，显式、隐式序列求解方法的发展已较为成熟，采用成熟的商业软件 (例如 ANSYS，LS-DYNA)，便可以方便地实现显式、隐式序列求解过程.

对于另一类问题，即空间上的显式、隐式分区计算，主要应用于大型复杂系统局部受冲击载荷的动力学响应或结构–介质相互作用的分析中. 采用有限单元法将动力学系统在空间域内离散后，还需要通过直接数值积分方法逐次求解. 传统数值积分方法的时间步长往往受局部单元的稳定性和精度要求所限制，导致整个模型都必须采取较小的时间步长，造成大量计算资源的浪费. 而在大型结构或耦合系统有限元模型中，通常可以将网格划分为具有不同时间步长要求的区域，为了提高计算效率，需设计可以在不同分区采用不同时间步长的积分算法. 这类研究始于 Belytschko，其初衷是模拟结构与介质的相互作用问题 [14-15]. 在不同介质分区，采用不同的离散格式，可以发挥不同算法的优势. 在后来的几十年中，出现了许多分区异步算法 [16-26]，主要包括显式子循环算法 [15,18-19,22]，隐式子循环算法 [20-21]，隐式 - 显式异步算法 [14,16-17] 以及这些算法的相关改进 [23-24] 和扩展应用 [25-26]. 在不同时步边界处理上，这些算法多数采用线性加速度或线性速度的假设，对边界节点进行插值处理，而这种方式往往被认为误差积累和不稳定性发展的根源，导致了计算结果存在较大误差以及稳定性条件苛刻等问题，给实际应用带来了极大的限制.

本文提出一种基于 Newmark 离散格式的显式、隐式任意混合异步算法，并通过结构在冲击载荷下的动态响应计算验证该方法的正确性和有效性. 该算法根据复杂动力系统各空间域的物理力学特性将模型进行分区异步求解，并根据显式、隐式算法固有的信息传递机制，采取动态的可变边界处理方法进行异步边界求解，能在保持较高的计算精度的同时，极大地降低计算资源消耗，因而具有一定的实用价值.

1 统一的显式隐式积分格式

1.1 动力学方程

一般情况下，结构或系统的动力学方程可写为

$$\boldsymbol{M}\boldsymbol{a}+\boldsymbol{C}\boldsymbol{v}+\boldsymbol{K}\boldsymbol{d}=\boldsymbol{Q}^{\text{ext}} \tag{1}$$

其中, $\boldsymbol{a}$, $\boldsymbol{v}$, $\boldsymbol{d}$ 分别为加速度、速度和位移向量，$\boldsymbol{M}$, $\boldsymbol{C}$, $\boldsymbol{K}$ 和 $\boldsymbol{Q}^{\text{ext}}$ 分别为结构的质量、阻尼、刚度矩阵和外界载荷向量，并分别由各自的单元矩阵和向量组集而成

$$\boldsymbol{M}=\sum_e \boldsymbol{M}^e=\sum_e\int_{V_e}\rho\boldsymbol{N}^{\mathrm{T}}\boldsymbol{N}\mathrm{d}V \tag{2}$$

$$\boldsymbol{C}=\sum_e \boldsymbol{C}^e=\sum_e\int_{V_e}\mu\boldsymbol{N}^{\mathrm{T}}\boldsymbol{N}\mathrm{d}V \tag{3}$$

$$\boldsymbol{K}=\sum_e \boldsymbol{K}^e=\sum_e\int_{V_e}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{B}\mathrm{d}V \tag{4}$$

$$\boldsymbol{Q}^{\text{ext}}=\sum_e \boldsymbol{Q}^e=\sum_e\Big(\int_{V_e}\boldsymbol{N}^{\mathrm{T}}\boldsymbol{f}\mathrm{d}V+\int_{S_T^e}\boldsymbol{N}^{\mathrm{T}}\boldsymbol{T}\mathrm{d}S\Big) \tag{5}$$

其中，$\boldsymbol{N}$ 和 $\boldsymbol{B}$ 分别为空间域离散所采用单元的形函数矩阵和应变矩阵；$\boldsymbol{f}$ 和 $\boldsymbol{T}$ 分别代表作用于单元节点上的体力和面力.

1.2 基于 Newmark 离散格式的隐式算法

在 $n\sim n+1$ 时步的间隔 Δt 内，Newmark 时间离散对位移、速度和加速度采用如下假设关系

$$\boldsymbol{d}_{n+1}=\boldsymbol{d}_n+\dot{\boldsymbol{d}}_n\Delta t+\Big[\Big(\frac{1}{2}-\beta\Big)\ddot{\boldsymbol{d}}_n+\beta\ddot{\boldsymbol{d}}_{n+1}\Big]\Delta t^2 \tag{6}$$

$$\dot{\boldsymbol{d}}_{n+1}=\dot{\boldsymbol{d}}_n+[(1-\gamma)\ddot{\boldsymbol{d}}_n+\gamma\ddot{\boldsymbol{d}}_{n+1}]\Delta t \tag{7}$$

$$\ddot{\boldsymbol{d}}_{n+1}=\frac{1}{\beta\Delta t^2}(\boldsymbol{D}_{n+1}-\boldsymbol{d}_n)-\frac{1}{\beta\Delta t}\dot{\boldsymbol{d}}_n-\Big(\frac{1}{2\beta}-1\Big)\ddot{\boldsymbol{d}}_n \tag{8}$$

式中，β 和 γ 为考虑精度和稳定性要求而调整的参数.

Newmark 隐式方法中，$n+1$ 时步的位移由 $t+\Delta t$ 时刻的运动方程得到，将式 (6) ~ 式 (8) 代入系统动力学方程则可得到关于 d_{n+1} 的方程

$$\begin{aligned}\Big(\boldsymbol{K}+\frac{1}{\beta\Delta t^2}\boldsymbol{M}+\frac{\gamma}{\beta\Delta t}\boldsymbol{C}\Big)\boldsymbol{d}_{n+1}=\boldsymbol{Q}_{n+1}+\\ \boldsymbol{M}\Big[\frac{1}{\beta\Delta t^2}\boldsymbol{d}_n+\frac{1}{\beta\Delta t}\dot{\boldsymbol{d}}_n+\Big(\frac{1}{2\beta}-1\Big)\ddot{\boldsymbol{d}}_n\Big]+\\ C\Big[\frac{\gamma}{\beta\Delta t}\boldsymbol{d}_n+\Big(\frac{\gamma}{\beta}-1\Big)\dot{\boldsymbol{d}}_n+\Big(\frac{\gamma}{2\beta}-1\Big)\Delta t\ddot{\boldsymbol{d}}_n\Big]\end{aligned} \tag{9}$$

即

$$\bar{\boldsymbol{K}}\boldsymbol{d}_{n+1}=\bar{\boldsymbol{Q}}_{n+1} \tag{10}$$

其中，$\bar{\boldsymbol{K}}$ 为广义刚度矩阵，$\bar{\boldsymbol{Q}}_{n+1}$ 为 $t+\Delta t$ 时刻的广义

载荷向量，表达式为

$$\left.\begin{aligned}&\bar{\boldsymbol{K}}=\boldsymbol{K}+\frac{1}{\beta\Delta t^2}\boldsymbol{M}+\frac{\gamma}{\beta\Delta t}\boldsymbol{C}\\&\bar{\boldsymbol{Q}}_{n+1}=\boldsymbol{Q}_{n+1}+\boldsymbol{M}\Big[\frac{1}{\beta\Delta t^2}\boldsymbol{d}_n+\frac{1}{\beta\Delta t}\dot{\boldsymbol{d}}_n+\Big(\frac{1}{2\beta}-1\Big)\ddot{\boldsymbol{d}}_n\Big]+\\&\qquad\boldsymbol{C}\Big[\frac{\gamma}{\beta\Delta t}\boldsymbol{d}_n+\Big(\frac{\gamma}{\beta}-1\Big)\dot{\boldsymbol{d}}_n+\Big(\frac{\gamma}{2\beta}-1\Big)\Delta t\ddot{\boldsymbol{d}}_n\Big]\end{aligned}\right\}\tag{11}$$

求解式 (10) 可得位移解 $\boldsymbol{d}_{n+1}$，回代至式 (7) 和式 (8) 即得速度和加速度.

研究表明：当 $\gamma\geqslant\dfrac{1}{2}$，$\beta\geqslant\dfrac{1}{4}\Big(\dfrac{1}{2}+\gamma\Big)^2$ 时，Newmark 隐式方法是无条件稳定的.

1.3 基于 Newmark 离散格式的显式预测 —— 校正算法

该算法仍采用 Newmark 隐式算法关于位移、速度和加速度的离散格式. 将式 (6) ~ 式 (7) 的位移和速度表达式写为预测–校正形式

$$\left.\begin{aligned}\boldsymbol{d}_{n+1}&=\tilde{\boldsymbol{d}}_{n+1}+\beta\Delta t^2\ddot{\boldsymbol{d}}_{n+1}\\\dot{\boldsymbol{d}}_{n+1}&=\tilde{\dot{\boldsymbol{d}}}_{n+1}+\gamma\Delta t\ddot{\boldsymbol{d}}_{n+1}\end{aligned}\right\}\tag{12}$$

式中，符号 “~” 表示由时刻 t 的速度和加速度值获得的 $t+\Delta t$ 时刻的位移和速度的预测值

$$\left.\begin{aligned}\tilde{\boldsymbol{d}}_{n+1}&=\boldsymbol{d}_n+\dot{\boldsymbol{d}}_n\Delta t+\Big(\frac{1}{2}-\beta\Big)\Delta t^2\ddot{\boldsymbol{d}}_n\\\tilde{\dot{\boldsymbol{d}}}_{n+1}&=\dot{\boldsymbol{d}}_n+(1-\gamma)\ddot{\boldsymbol{d}}_n\Delta t\end{aligned}\right\}\tag{13}$$

将预测值代入运动方程得

$$\boldsymbol{M}\ddot{\boldsymbol{d}}_{n+1}+\boldsymbol{C}\tilde{\dot{\boldsymbol{d}}}_{n+1}+\boldsymbol{K}\tilde{\boldsymbol{d}}_{n+1}=\boldsymbol{Q}_{n+1}\tag{14}$$

若采用集中质量矩阵，将式 (13) 代入式 (14) 则可以显式地求得 $t+\Delta t$ 时刻的加速度值 $\ddot{\boldsymbol{d}}_{n+1}$

$$\begin{aligned}\ddot{\boldsymbol{d}}_{n+1}=&\boldsymbol{M}^{-1}\boldsymbol{Q}_{n+1}-\boldsymbol{M}^{-1}\boldsymbol{C}[\dot{\boldsymbol{d}}_n+(1-\gamma)\ddot{\boldsymbol{d}}_n\Delta t]-\\&\boldsymbol{M}^{-1}\boldsymbol{K}\Big[\boldsymbol{d}_n+\dot{\boldsymbol{d}}_n\Delta t+\Big(\frac{1}{2}-\beta\Big)\Delta t^2\ddot{\boldsymbol{d}}_n\Big]\end{aligned}\tag{15}$$

将求得的加速度值回代至校正式 (12) 即得 $n+1$ 时步的位移和速度值. 可以证明，当阻尼不存在时，算法的稳定条件为 $\Omega_c=\sqrt{2/\gamma}$，当 $\gamma=1/2$ 时，条件与经典的中心差分法相同，且保持 2 阶精度，当 $\gamma\geqslant1/2$ 时，该算法可保持 1 阶精度 [27].

2 分区边界统一处理方式

2.1 显式及隐式算法的信息传递机制

以显式–隐式同步算法信息传递规律为例进行分析，如图 1 所示的二维有限元模型分区，下标 I 和 E 分别代表隐式和显式，只由显式算法计算的单元由 E 标识，至少与一个隐式单元相邻的节点用 I 标识. 显式、隐式区域的公共节点用 B 标识，而 B 为 I 的子集. 该模型刚度矩阵具有以下形式

$$\boldsymbol{K}=\begin{bmatrix}\boldsymbol{K}_{\rm EE}&\boldsymbol{K}_{\rm EB}&\boldsymbol{0}\\\boldsymbol{K}_{\rm BE}&\boldsymbol{K}_{\rm BB}&\boldsymbol{K}_{\rm BI}\\\boldsymbol{0}&\boldsymbol{K}_{\rm IB}&\boldsymbol{K}_{\rm II}\end{bmatrix}\tag{16}$$

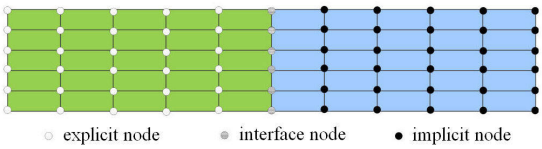


图 1 有限元模型分区示意图

Fig. 1 Schematic diagram of partitioned FE mesh

令

$$\left.\begin{aligned}&\boldsymbol{K}^{\rm E}=[\boldsymbol{K}_{\rm EE}],\quad\boldsymbol{K}^{\rm EI}=[\boldsymbol{K}_{\rm EB}\ \ \boldsymbol{0}]\\&\boldsymbol{K}^{\rm IE}=\begin{bmatrix}\boldsymbol{K}_{\rm BE}\\\boldsymbol{0}\end{bmatrix},\quad\boldsymbol{K}^{I}=\begin{bmatrix}\boldsymbol{K}_{\rm BB}&\boldsymbol{K}_{\rm BI}\\\boldsymbol{K}_{\rm IB}&\boldsymbol{K}_{\rm II}\end{bmatrix}\end{aligned}\right\}\tag{17}$$

则 $\boldsymbol{K}=\begin{bmatrix}\boldsymbol{K}^{\rm E}&\boldsymbol{K}^{\rm EI}\\\boldsymbol{K}^{\rm IE}&\boldsymbol{K}^{\rm I}\end{bmatrix}$，阻尼矩阵 $\boldsymbol{C}$ 的分块类似.

对于任意的 i 时刻，动力学控制方程可写为

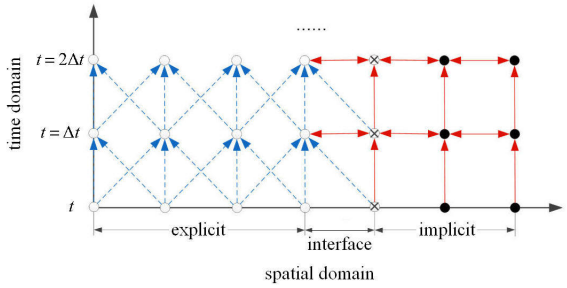
$$\begin{aligned}&\begin{bmatrix}\boldsymbol{M}^{\rm E}&\boldsymbol{0}\\\boldsymbol{0}&\boldsymbol{M}^{\rm I}\end{bmatrix}\begin{Bmatrix}\ddot{\boldsymbol{d}}_i^{\rm E}\\\ddot{\boldsymbol{d}}_i^{\rm I}\end{Bmatrix}+\begin{bmatrix}\boldsymbol{C}^{\rm E}&\boldsymbol{C}^{\rm EI}\\\boldsymbol{C}^{\rm IE}&\boldsymbol{C}^{\rm I}\end{bmatrix}\begin{Bmatrix}\dot{\boldsymbol{d}}_i^{\rm E}\\\dot{\boldsymbol{d}}_i^{\rm I}\end{Bmatrix}+\\&\begin{bmatrix}\boldsymbol{K}^{\rm E}&\boldsymbol{K}^{\rm EI}\\\boldsymbol{K}^{\rm IE}&\boldsymbol{K}^{\rm I}\end{bmatrix}\begin{Bmatrix}\boldsymbol{d}_i^{\rm E}\\\boldsymbol{d}_i^{\rm I}\end{Bmatrix}=\begin{Bmatrix}\boldsymbol{q}_i^{\rm E}\\\boldsymbol{q}_i^{\rm I}\end{Bmatrix}\end{aligned}\tag{18}$$

现推导已知 n 时刻，求解 $n+1$ 时刻的方程. 将 Newmark 预测显式格式代入式 (18) 的第 1 行得

$$\begin{aligned}&\boldsymbol{M}^{\rm E}\ddot{\boldsymbol{d}}_{n+1}^{\rm E}+\boldsymbol{C}^{\rm E}\tilde{\dot{\boldsymbol{d}}}_{n+1}^{\rm E}+\boldsymbol{C}^{\rm EI}\dot{\boldsymbol{d}}_{n+1}^{\rm I}+\boldsymbol{K}^{\rm E}\tilde{\boldsymbol{d}}_{n+1}^{\rm E}+\boldsymbol{K}^{\rm EI}\boldsymbol{d}_{n+1}^{\rm I}=\boldsymbol{q}_{n+1}^{\rm E}\Rightarrow\\&\ddot{\boldsymbol{d}}_{n+1}^{\rm E}=(\boldsymbol{M}^{\rm E})^{-1}(\boldsymbol{q}_{n+1}^{\rm E}-\boldsymbol{C}^{\rm E}\tilde{\dot{\boldsymbol{d}}}_{n+1}^{\rm E}-\boldsymbol{C}^{\rm EI}\dot{\boldsymbol{d}}_{n+1}^{\rm I}-\\&\boldsymbol{K}^{\rm E}\tilde{\boldsymbol{d}}_{n+1}^{\rm E}-\boldsymbol{K}^{\rm EI}\boldsymbol{d}_{n+1}^{\rm I})\end{aligned}\tag{19}$$

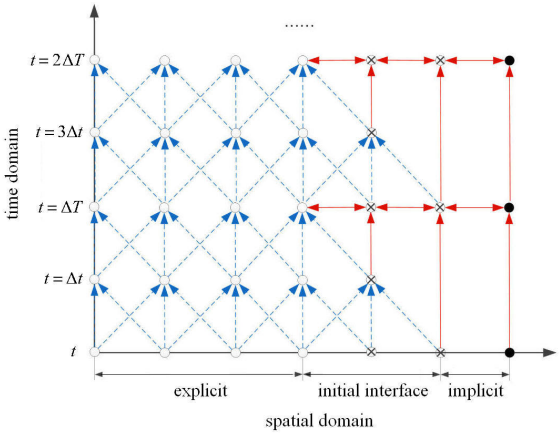
因为 $\boldsymbol{K}^{\rm EI}=[\boldsymbol{K}_{\rm EB}\ \ \boldsymbol{0}]$，$\boldsymbol{d}^{\rm I}=\{\boldsymbol{d}_{\rm BB},\boldsymbol{d}_{\rm II}\}'$，上式最后一项化为：$\boldsymbol{K}_{\rm EB}\{\boldsymbol{d}_{\rm BB}\}_{n+1}$；同理，上式倒数第 3 项则化为 $\boldsymbol{C}_{\rm EB}\{\dot{\boldsymbol{d}}_{\rm BB}\}_{n+1}$. 即：求解显式分区节点的下一时刻位移需用到该节点和相邻节点位移和速度值. 同理可推知：求解隐式分区节点的下一时刻位移需用到该节点和相邻节点同一时刻的相关信息.

从数学意义上讲：显式算法是指待求时步 ($t+\Delta t$ 时刻) 的未知量可由上一时步 (t 时刻) 的已知结果显式地表达，如式 (15) 所示；而隐式算法则是指待求时步 ($t+\Delta t$ 时刻) 的未知量也含有当前时刻的未知量，如式 (10) 所示. 以一维有限元模型为例，显式 - 隐式同步及异步 ($\Delta T = 2\Delta t$) 算法在时、空域内的信息传递机制见图 2.



(a) 相同时步

(a) Equal time steps



(b) 隐式时步 =2 × 显式时步

(b) Implicit time step=2 × explicit time step

图 2　显式–隐式混合算法信息传递

Fig. 2　Flow of information in mixed explicit–implicit algorithm

2.2 显式–隐式异步算法边界处理

从上文的分析可以看出：显式算法其实质是一种弱耦合算法，求解下一时刻的相关信息时，只需用到相临节点上一时步的相关信息；而隐式算法更多表现为一种强耦合算法，求解下一时刻的节点相关信息时，则需要耦合所有节点的相关信息. 因此，当隐式时步为显式时步的 m 倍时，显式算法需用到 m 层相邻节点信息. 设显式分区积分步长 Δt，而隐式分区积分步长为 $m\Delta t$. $m = 3$ 时 (隐式时步为显式 3 倍) 的节点分区如图 3 所示.

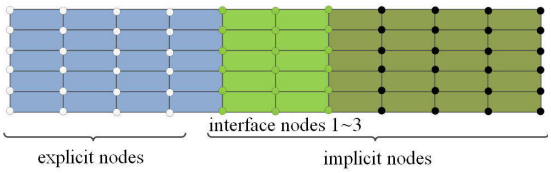


图 3　显式–隐式异步算法节点分区示意图

Fig. 3　Schematic diagram of node partition in multi-time step explicit–implicit algorithm

而对应的刚度矩阵有如下分块形式

$$\boldsymbol{K}=\begin{bmatrix} \boldsymbol{K}_{\mathrm{EE}} & \boldsymbol{K}_{\mathrm{EB}_1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{K}_{\mathrm{B}_1\mathrm{E}} & \boldsymbol{K}_{\mathrm{B}_1\mathrm{B}_1} & \boldsymbol{K}_{\mathrm{B}_1\mathrm{B}_2} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_{\mathrm{B}_2\mathrm{B}_1} & \boldsymbol{K}_{\mathrm{B}_2\mathrm{B}_2} & \boldsymbol{K}_{\mathrm{B}_2\mathrm{B}_3} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{K}_{\mathrm{B}_3\mathrm{B}_2} & \boldsymbol{K}_{\mathrm{B}_3\mathrm{B}_3} & \boldsymbol{K}_{\mathrm{B}_3\mathrm{I}} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{K}_{\mathrm{IB}_3} & \boldsymbol{K}_{\mathrm{II}} \end{bmatrix} \tag{20}$$

同上一节，令

$$\left.\begin{aligned} &\boldsymbol{K}^{\mathrm{E}}=[\boldsymbol{K}_{\mathrm{EE}}],\quad \boldsymbol{K}^{\mathrm{EI}}=[\boldsymbol{K}_{\mathrm{EB}_1}\ \ \boldsymbol{0}],\quad \boldsymbol{K}^{\mathrm{IE}}=\begin{bmatrix}\boldsymbol{K}_{\mathrm{B}_1\mathrm{E}}\\ \boldsymbol{0}\\ \boldsymbol{0}\\ \boldsymbol{0}\end{bmatrix} \\ &\boldsymbol{K}^{\mathrm{I}}=\begin{bmatrix} \boldsymbol{K}_{\mathrm{B}_1\mathrm{B}_1} & \boldsymbol{K}_{\mathrm{B}_1\mathrm{B}_2} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{K}_{\mathrm{B}_2\mathrm{B}_1} & \boldsymbol{K}_{\mathrm{B}_2\mathrm{B}_2} & \boldsymbol{K}_{\mathrm{B}_2\mathrm{B}_3} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_{\mathrm{B}_3\mathrm{B}_2} & \boldsymbol{K}_{\mathrm{B}_3\mathrm{B}_3} & \boldsymbol{K}_{\mathrm{B}_3\mathrm{I}} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{K}_{\mathrm{IB}_3} & \boldsymbol{K}_{\mathrm{II}} \end{bmatrix} \end{aligned}\right\} \tag{21}$$

则 $\boldsymbol{K}=\begin{bmatrix}\boldsymbol{K}^{\mathrm{E}} & \boldsymbol{K}^{\mathrm{EI}}\\ \boldsymbol{K}^{\mathrm{IE}} & \boldsymbol{K}^{\mathrm{I}}\end{bmatrix}$. 类似地可对 $\boldsymbol{C}$ 矩阵进行分块.

同理，可对自由度进行分块，令

$$\left.\begin{aligned} &\boldsymbol{d}^{\mathrm{E}}=\{\boldsymbol{d}_{\mathrm{EE}}\} \\ &\boldsymbol{d}^{\mathrm{I}}=\{\boldsymbol{d}_{\mathrm{B}_1\mathrm{B}_1}\boldsymbol{d}_{\mathrm{B}_2\mathrm{B}_2}\boldsymbol{d}_{\mathrm{B}_3\mathrm{B}_3}\boldsymbol{d}_{\mathrm{II}}\}' \end{aligned}\right\} \tag{22}$$

则 $\boldsymbol{d}=\{\boldsymbol{d}^{\mathrm{E}}\boldsymbol{d}^{\mathrm{I}}\}'$.

以 $m = 3$ 的情况为例，现假设第 n 时步的信息已知，推导 $n+m$ 时步的情形，算法的实现步骤如下：

(i) 采用预测公式 (13) 计算显式分区和所有边界节点的位移和速度的预测值.

(ii) 由平衡方程计算显式和所需边界节点的加速度，在每次由平衡方程求解加速度时，边界减少一层.

相应的递推公式如下

$$\left\{\ddot{\boldsymbol{d}}_{\mathrm{EE}}\right\}_{n+1}=\frac{1}{\boldsymbol{M}^{\mathrm{E}}}[\boldsymbol{q}_{n+1}^{\mathrm{E}}-\boldsymbol{C}^{\mathrm{E}}\tilde{\dot{\boldsymbol{d}}}_{n+1}^{\mathrm{E}}-\boldsymbol{C}_{\mathrm{EB}_1}\tilde{\dot{\boldsymbol{d}}}_{n+1}^{\mathrm{B}_1}-\boldsymbol{K}^{\mathrm{E}}\tilde{\boldsymbol{d}}_{n+1}^{\mathrm{E}}-\boldsymbol{K}_{\mathrm{EB}_1}\tilde{\boldsymbol{d}}_{n+1}^{\mathrm{B}_1}] \tag{23}$$

$$\left\{\ddot{\boldsymbol{d}}_{\mathrm{B}_1\mathrm{B}_1}\right\}_{n+1}=\frac{1}{\boldsymbol{M}^{\mathrm{B}_1}}\left[\boldsymbol{q}_{n+1}^{\mathrm{B}_1}-\boldsymbol{C}_{\mathrm{B}_1\mathrm{B}_1}\tilde{\dot{\boldsymbol{d}}}_{n+1}^{\mathrm{B}_1}-\boldsymbol{C}_{\mathrm{B}_1\mathrm{E}}\tilde{\dot{\boldsymbol{d}}}_{n+1}^{\mathrm{E}}-\boldsymbol{C}_{\mathrm{B}_1\mathrm{B}_2}\tilde{\dot{\boldsymbol{d}}}_{n+1}^{\mathrm{B}_2}-\boldsymbol{K}_{\mathrm{B}_1\mathrm{B}_1}\tilde{\boldsymbol{d}}_{n+1}^{\mathrm{B}_1}-\boldsymbol{K}_{\mathrm{B}_1\mathrm{E}}\tilde{\boldsymbol{d}}_{n+1}^{\mathrm{E}}-\boldsymbol{K}_{\mathrm{B}_1\mathrm{B}_2}\tilde{\boldsymbol{d}}_{n+1}^{\mathrm{B}_2}\right] \tag{24}$$

对于编号 $i\geqslant 2$ 的边界

$$\left\{\ddot{\boldsymbol{d}}_{\mathrm{B}_i\mathrm{B}_i}\right\}_{n+1}=\frac{1}{\boldsymbol{M}^{\mathrm{B}_i}}\left[\boldsymbol{q}_{n+1}^{\mathrm{B}_i}-\boldsymbol{C}_{\mathrm{B}_i\mathrm{B}_i}\tilde{\dot{\boldsymbol{d}}}_{n+1}^{\mathrm{B}_i}-\boldsymbol{C}_{\mathrm{B}_i\mathrm{B}_{i-1}}\tilde{\dot{\boldsymbol{d}}}_{n+1}^{\mathrm{B}_{i-1}}-\boldsymbol{C}_{\mathrm{B}_i\mathrm{B}_{i+1}}\tilde{\dot{\boldsymbol{d}}}_{n+1}^{\mathrm{B}_{i+1}}-\boldsymbol{K}_{\mathrm{B}_i\mathrm{B}_i}\tilde{\boldsymbol{d}}_{n+1}^{\mathrm{B}_i}-\boldsymbol{K}_{\mathrm{B}_i\mathrm{B}_{i-1}}\tilde{\boldsymbol{d}}_{n+1}^{\mathrm{B}_{i-1}}-\boldsymbol{K}_{\mathrm{B}_i\mathrm{B}_{i+1}}\tilde{\boldsymbol{d}}_{n+1}^{\mathrm{B}_{i+1}}\right] \tag{25}$$

依次类推，直至本步计算所需的所有边界节点.

(iii) 由校正公式 (12) 计算位移和速度.

(iv) 依次推进，直至 $t+m\Delta t$ 时刻.

(v) 将求得的显式分区的位移和速度值代入隐式分区运动方程，同时，用 $m\Delta t$ 替换原公式中的 Δt，求解得到隐式分区 (包括边界) 节点的位移. 对应的表达式如下

$$\hat{\boldsymbol{K}}\boldsymbol{d}_{n+m}^{I}=\hat{\boldsymbol{q}}_{n+m}^{I} \tag{26}$$

其中

$$\left.\begin{aligned}\hat{\boldsymbol{K}}&=\left(\boldsymbol{K}^{\mathrm{I}}+\frac{1}{\gamma m^2\Delta t^2}\boldsymbol{M}^{\mathrm{I}}+\frac{\beta}{\gamma m\Delta t}\boldsymbol{C}^{\mathrm{I}}\right)\\ \hat{\boldsymbol{q}}_{n+m}^{\mathrm{I}}&=\boldsymbol{q}_{n+m}^{\mathrm{I}}+\frac{\boldsymbol{M}^{\mathrm{I}}}{\gamma m^2\Delta t^2}\tilde{\boldsymbol{d}}_{n+m}^{\mathrm{I}}-\boldsymbol{C}^{\mathrm{I}}\tilde{\dot{\boldsymbol{d}}}_{n+m}^{\mathrm{I}}+\\ &\quad\frac{\beta}{\gamma m\Delta t}\boldsymbol{C}^{\mathrm{I}}\tilde{\boldsymbol{d}}_{n+m}^{\mathrm{I}}-\boldsymbol{C}^{\mathrm{IE}}\dot{\boldsymbol{d}}_{n+m}^{\mathrm{E}}-\boldsymbol{K}^{\mathrm{IE}}\boldsymbol{d}_{n+m}^{\mathrm{E}}\end{aligned}\right\}$$

则

$$\boldsymbol{d}_{n+m}^{\mathrm{I}}=\hat{\boldsymbol{K}}^{-1}\hat{\boldsymbol{q}}_{n+m}^{\mathrm{I}} \tag{27}$$

2.3 显式–显式异步算法边界处理

类似于上一小节所提出的显式 – 隐式异步算法，显式 – 显式异步算法仍然以大步长是小步长的 3 倍 ($m=3$) 为例，此时，边界单元要多一层，因为显式分区 2 同样要用到显式分区 1 的一层边界节点信息，此时的节点分区如图 4 所示.

显式 – 显式异步算法的主要流程亦与上一小节所述的显式–隐式异步算法相似. 唯一不同之处在于需要在小步长的显式分区多增加一层边界节点，在求解大步长分区时需用到该边界节点上一主时间步的相关信息.

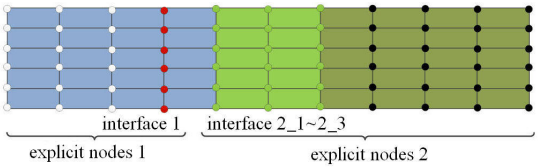


图 4 显式–显式异步算法节点分区示意图

Fig. 4 Schematic diagram of node partition in multi-time step explicit-explicit algorithm

实现流程上，显式–显式异步算法的前 4 步主要流程与上一小节所述的显式–隐式异步算法相同，而对应的第 (v) 计算步则为：采用式 (23) 计算大步长显式分区所有节点的加速度，用到的是属于小步长分区的边界层节点信息，进而通过校正公式可计算大步长显式分区内相关节点的速度与加速度.

3 实现流程的统一

从上文的分析中可以发现：根据边界节点与大小步长的对应关系，本文提出的显式–隐式或显式–显式异步算法，在实现流程上非常相近. 如果能够将关键步骤的算法模块化，便可以实现显式与隐式异步算法的任意组合，从而增强算法的实用性和应用范围.

在具体应用时，只需要在前处理时充分考虑显式、隐式分区以及边界的设置和分布，便能够方便的实现显式与隐式算法的任意组合异步计算. 对于特别关注的模型性细节或碰撞发生区域采用相对精细的网格划分，同时采用小步长积分求解；对于那些非关注的模型区域或响应以低频为主的模型分区，则可以采用较粗糙的网格离散，在时域积分计算时也可以采用较大步长的显式或隐式算法进行求解. 最终，将整体模型的结果数据进行整合与输出. 整体算法的主要实现流程如图 5 所示.

4 稳定性判断与能量平衡

时域逐步积分算法稳定性分析主要有 2 种方法：一种是通过分析整体系统能量变化，另一种则是通过分析数值积分逼近算子的谱半径. 能量法中，数值积分算法的稳定性条件为：对于积分过程的任意时间步 n，存在关于计算结果的范数 S_n，满足条件：$S_n\leqslant\kappa S_0$，其中 κ 为正常数. 该范数往往含有自变量导数的平方项，对应于固体力学的能量形式，因而称为能量法.

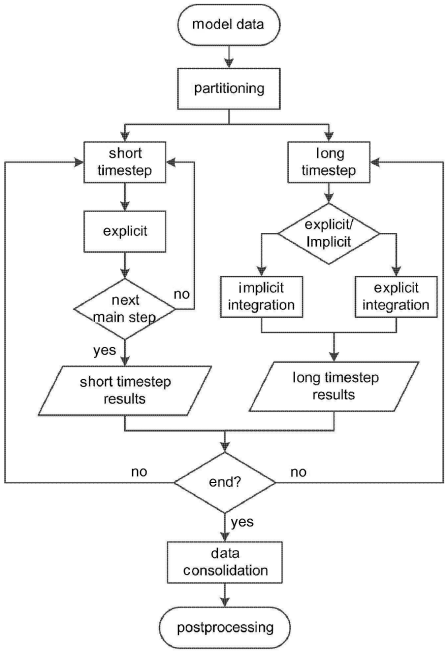


图 5 显式–隐式任意混合异步算法主要流程图

Fig. 5 Flowchart diagram of the arbitrarily mixed implicit–explicit asynchronous algorithm

文献 [14] 采用能量法分析了梯形公式–中心差分混合算法的稳定性条件. 设 λ^* 为显式区域内所有网格的最大特征值，则混合算法的最大容许步长需满足 $\Delta t \leqslant 2/\lambda^*$，然而，这也是显式中心差分方法的稳定性条件. 文献 [27] 通过能量法详细证明和分析了 Newmark 系列算法和由此发展而来的混合算法的稳定性条件. 混合算法的稳定性条件一般受各计算子域内算法稳定性的约束. 同理，亦可以用类似方法证明本文所述混合异步算法亦具有类似稳定性特征，此处不再赘述.

通常情况下，对积分算法的稳定性分析以及临界时间步长判断，都是在线性假设的前提下推导得出的 [27]. 目前，尚没有合适的稳定性原理可以涵盖到工程中所遇到的全部非线性现象，例如接触–冲击，断裂等. 即使是满足上文所述的稳定性条件，不稳定性也可能产生和发展，相对于线性问题，非线性问题的不稳定解答有时更不容易被识别. 一些由非线性诱发的不稳定，例如几何硬化 (材料是线弹性的)，这种不稳定引起的结果是局部指数的增长，从而导致了塑性行为. 而塑性响应又相应的软化了结构，降低了波速，从而使积分又得到稳定. 这种现象称为抑制失稳，抑制失稳会导致可能远远超过真实情况的位移，但是通过跟踪结果却很难发现. 特别是对于显式、隐式的混合异步算法，要保证数值求解过程中的绝对稳定，则变得异常困难.

然而，任何不稳定的结果都会导致伪能量生成，从而导致能量守恒的破坏. 因此，在非线性计算中，通过校核能量平衡可以实时地检验计算的稳定性，也有利于更好地察觉不稳定的出现. 计算过程中，可以通过与积分方法阶次类似的数值方法对时间积分得到能量，例如梯形法则. 内能和外力功的表达式分别可近似如下

$$W_{n+1}^{\mathrm{int}} = W_n^{\mathrm{int}} + \frac{\Delta t_{n+1}}{2}(\boldsymbol{v}_n)^{\mathrm{T}}(\boldsymbol{f}_n^{\mathrm{int}} + \boldsymbol{f}_{n+1}^{\mathrm{int}}) = W_n^{\mathrm{int}} + \frac{1}{2}\Delta \boldsymbol{d}^{\mathrm{T}}(\boldsymbol{f}_n^{\mathrm{int}} + \boldsymbol{f}_{n+1}^{\mathrm{int}}) \tag{28}$$

$$W_{n+1}^{\mathrm{ext}} = W_n^{\mathrm{ext}} + \frac{\Delta t_{n+1}}{2}(\boldsymbol{v}_{n+1})^{\mathrm{T}}(\boldsymbol{f}_n^{\mathrm{ext}} + \boldsymbol{f}_{n+1}^{\mathrm{ext}}) = W_n^{\mathrm{ext}} + \frac{1}{2}\Delta \boldsymbol{d}^{\mathrm{T}}(\boldsymbol{f}_n^{\mathrm{ext}} + \boldsymbol{f}_{n+1}^{\mathrm{ext}}) \tag{29}$$

其中，$\Delta \boldsymbol{d} = \boldsymbol{d}_{n+1} - \boldsymbol{d}_n$. 对于线弹性系统，时步 n 的内部节点力可通过下式进行计算

$$\boldsymbol{f}_n^{\mathrm{int}} = \boldsymbol{K}\boldsymbol{d}_n \tag{30}$$

而对于非线性系统，内部节点力计算需要在整个求解域进行积分计算

$$\boldsymbol{f}_n^{\mathrm{int}} = \int_{\Omega} \boldsymbol{B}^{\mathrm{T}}\boldsymbol{\sigma}_n \mathrm{d}\Omega \tag{31}$$

在单元或积分点的水平上，也可以计算内能

$$W_{n+1}^{\mathrm{int}} = W_n^{\mathrm{int}} + \frac{1}{2}\sum_e \Delta \boldsymbol{D}_e^{\mathrm{T}}(\boldsymbol{f}_n^{\mathrm{int,e}} + \boldsymbol{f}_{n+1}^{\mathrm{int,e}}) \tag{32}$$

系统动能的计算式为

$$W_n^{\mathrm{kin}} = \frac{1}{2}(\boldsymbol{v}_n)^{\mathrm{T}}\boldsymbol{M}\boldsymbol{v}_n \tag{33}$$

根据保守系统能量守恒原理，动力学方程求解过程中的各能量需满足下式

$$\left|W^{\mathrm{kin}} + W^{\mathrm{int}} - W^{\mathrm{ext}}\right| \leqslant \varepsilon \max\left(W^{\mathrm{kin}}, W^{\mathrm{int}}, W^{\mathrm{ext}}\right) \tag{34}$$

其中，ε 为一很小的误差容许极限，一般量级为 10^{-2}.

此外，如果整体动力学系统非常庞大，则能量的平衡计算可以在每个子区域内进行，相邻子区域间的内力可视为作用在子区域上的外力.

5 算例与分析

5.1 算例验证

对于简单结构，其固有频率和振型往往可通过解析方式求得，进而根据其振型的正交性，通过坐标变换在各主坐标内解析地求解解耦后的动力学方

程，最后通过叠加得到整体系统的响应. 因此，本文以均匀简支梁的弯曲振动为例，对所提出异步算法的计算精度进行校验，并对两种异步算法的计算效率、能量平衡和稳定性展开分析.

如图 6 所示的简支梁结构，在其长度的 1/4 处 (A 点) 受一阶跃力作用，本文分别通过模态迭加法，LS-DYNA 中心差分显式算法，显式–隐式异步算法以及显式 – 显式异步算法计算其响应，并进行对比分析. 模型的几何尺寸、主要材料参数、载荷及边界条件见图 6.

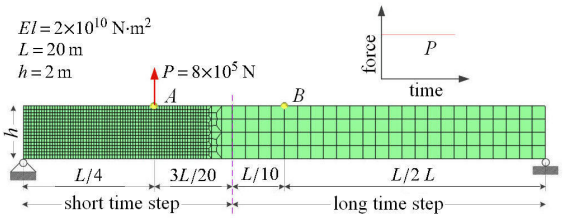


图 6 简支梁强迫振动分析模型

Fig. 6 Forced vibration analysis model of simply supported beam

由于显式 – 显式异步算法的稳定性受各计算域内单元最小特征尺寸影响. 在空间域内粗网格区域的最小单元的特征尺寸约为精细网格区域尺寸的 4 倍，因此，在初步的对比中，两种异步算法中的 m 都取为 4. 不同算法求解得到的梁中点处 (B 点) 的振动位移响应如图 7 所示. 1/2 倍梁的一阶弯曲振动周期时刻 ($t \approx 0.65$ s) 梁的变形以及轴向应力分布如图 8 所示. 不同数值方法的计算结果间只存在微小偏差，表明了本文所提出的统一格式显式、隐式任意混合异步算法具有较高的计算精度.

图 9 为两种异步算法在 $m = 4$ 时的能量变化统计结果. 不难发现：在较小的异步倍数 (m) 下，两种

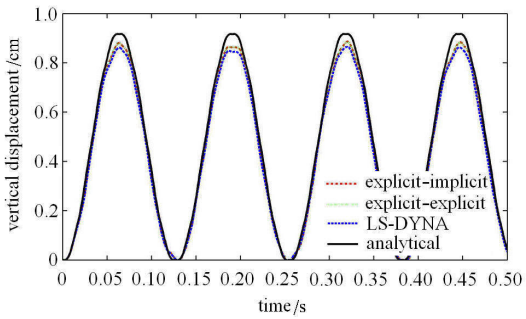


图 7 简支梁中点位移响应

Fig. 7 Displacement response at the middle point of the simply supported beam

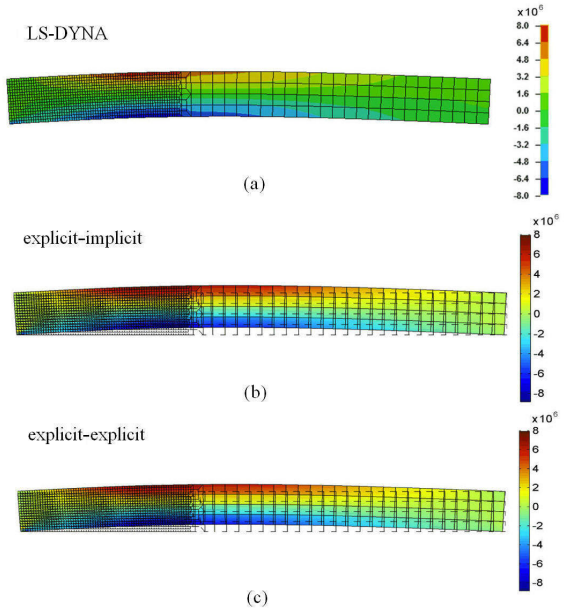


图 8 简支梁变形及轴向应力 (Pa)

Fig. 8 Deformation and axial stress (Pa) of the simply supported beam

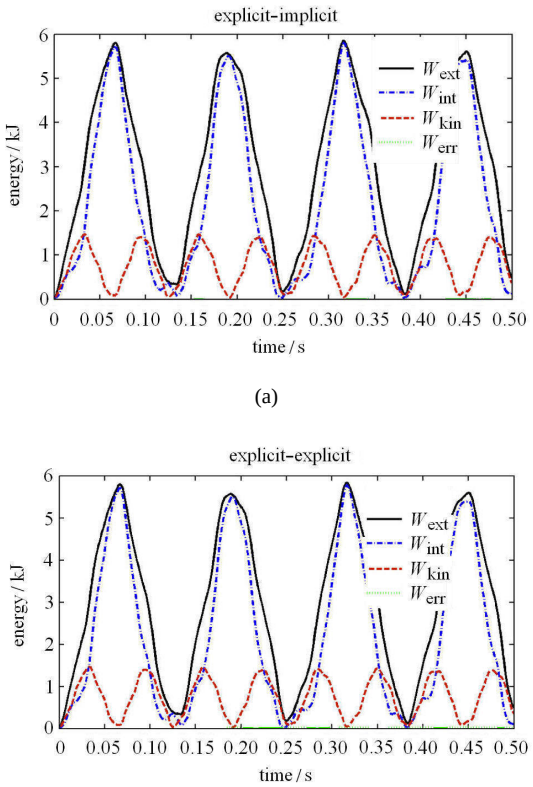


图 9 两种异步算法能量统计

Fig. 9 Energy curves of the two kinds of asynchronous algorithms

异步算法的能量误差曲线基本与坐标横轴重合，整体系统的能量平衡关系仅存在微小的误差，即：外力功与系统所储存的动能和势能之和进行守恒的相互

转化，没有出现伪能量的产生和发展. 因此，在满足各计算域内所有单元最小时间步长要求的前提下，本文所提出的统一格式异步算法体现出了较好的求解稳定性.

然而，值得一提的是，因为隐式算法较之显式算法具有更宽松的稳定性条件，显式–隐式异步计算可采用更大的步长比. 换一种角度来看，可以将隐式分区整体视为一超级单元参与显式计算，而隐式计算域的计算为超级单元的单元求解. 因而隐式超级单元整体需满足显式计算的稳定性条件. 本文同时测试了当 $m = 10, 20$ 时的显式–隐式异步算法计算效果. 图 10 和图 11 分别显示了梁中点位移时程曲线和系统能量变化情况. 不难发现，当 m 值大幅提高时，算法的计算精度并没有出现明显下降.

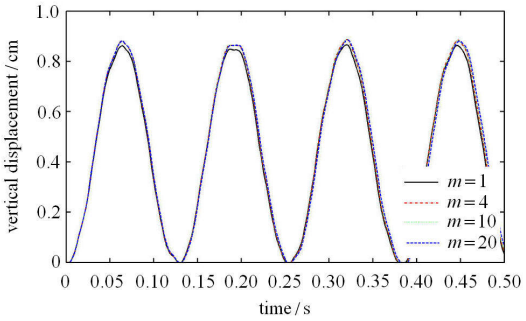


图 10　不同 m 取值时显式–隐式异步计算结果

Fig. 10　Explicit–implicit computation results with different m values

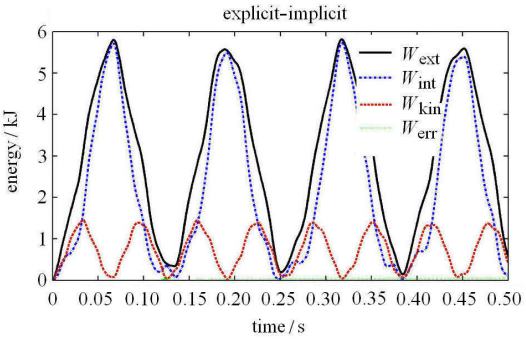


图 11　$m = 20$ 时，显式–隐式异步算法能量统计

Fig. 11　Energy curve of the explicit–implicit asynchronous algorithm with $m = 20$

本文提出的显式、隐式任意混合的异步算法，不但可以在空间域内对动力学系统进行多尺度离散，还可以在时域内采用多尺度积分，从而大大地节约计算资源，提高求解效率. 以全精细网格 (尺度与图 6 精细区域相同)，统一的时间尺度计算所需求解时间为基准 (100%)，采用显式、隐式混合异步算法在不同 m 值时求解相同物理过程所需时间如表 1 所示. 采用多尺度的离散网格，同时在时域积分上采用不同的步长尺度，大大地缩短了求解相同物理问题所需的计算时间. 当面对实际问题时，需要关注的细节往往在一些结构的局部，因而大步长的计算域更大. 采用大步长计算的节点越多，则理论上获得的计算效率提升也就越大.

表 1　求解时间统计

Table 1　CPU time needed for different approaches

Model and solving method			Time consumed/s	Percentage/%
single-scale	—	I	128.6	100.0
	—	E	116.9	90.90
multi-scale	$m = 1$	I	90.6	70.45
		E	83.5	64.93
	$m = 4$	E-I	76.2	59.25
		E-E	72.3	56.22
	$m = 10$	E-I	66.8	51.94
	$m = 20$	E-I	63.4	49.30

5.2　应用案例

塔楼–裙楼结构是一种常见的建筑形式，其低层为商业中心，高层为办公楼、公寓和住宅等，差异化的建筑布局极大地节约了日趋紧张的城市土地. 作为一种土地集约化的建筑形式，这类建筑也越来越受到市场和开发商的认同. 本文以塔楼–裙楼结构在爆炸冲击波作用下的动态响应为例，将异步算法应用到复杂动力学系统的求解中，以进一步验证算法的实用性.

对于由商业裙楼和塔楼组成的复合建筑结构，在爆炸冲击波作用下，其低层，往往直接承受冲击作用，主要分析结构的变形或破坏等，因而，在进行动态响应分析时宜采用小步长的的显式算法；而对于其高层，则主要分析其整体结构的晃动及层间相对位移等，因此，大步长的隐式算法则显得更加合适.

一般来讲，爆炸物在地面或近地爆炸，冲击波会在地面的反射作用下得到迅速加强，并以半球形的形态向空间扩散，直至与结构物发生作用. 爆炸冲击波与结构的作用是一个相当复杂的动力学过程，冲击波会在极短的时间内对结构的正面、侧面、背面及顶部造成冲击. 在冲击波入射角为 0° 的结构正面，受到的载荷远大于结构的其他部位. 研究中主要用冲击波压力、超压峰值、冲量以及持续时间等参数来描述爆炸冲击波作用. 仿真计算时，可将爆炸冲击作用保守地简化为对结构正面的右三角脉冲载荷，保留其峰值和持续时间特性 [28-30]. 参考文献 [30]，

本文采用的爆炸冲击波作用工况设置如表 2 所示，以等效节点载荷的形式加载，结构阻尼比取为 5%，以瑞利阻尼的形式加载. 混凝土材料的参数取为：密度 2 500 kg/m^3，弹性模量 3.0×10^{10} Pa，泊松比 0.2. 采用显式–隐式异步算法求解，整体有限元仿真模型以及显式–隐式分区如图 12 示.

表 2 爆炸冲击波作用参数设置

Table 2 Blast load considered in dynamic analysis

Load case	Mass/kg	Distance/m	Reflected overpressure/kPa	Impulse/(kPa · ms)	Positive phase duration/ms
I	100	10	846	1 543	9.7
II	200	10	1 699	2 582	12.1
III	500	10	4 250	5 172	17.5
IV	500	15	1 254	3 096	16.2
V	500	20	535	2 186	18.4

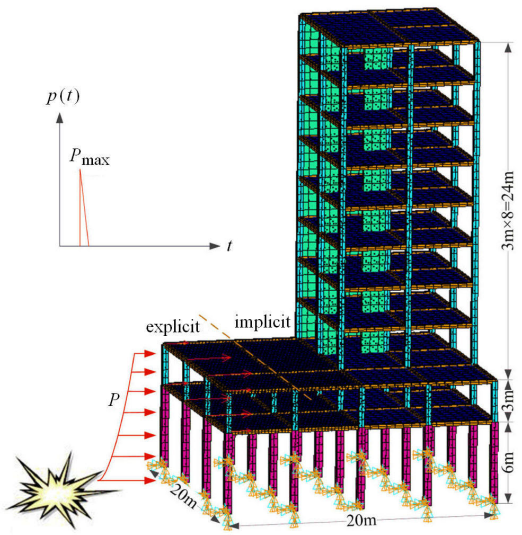


图 12 框架–剪力墙结构有限元模型

Fig. 12 FE model of the frame–shear wall structure

图 13 为工况 I 裙楼及塔楼顶部在冲击波作用方向的位移时程曲线，在爆炸冲击波作用下，建筑结构在短时间产生了剧烈的晃动，且塔楼部分的晃动位移明显大于直接受冲击载荷的裙楼部分，其最大位移达到了 0.125 m. 因此，有必要对建筑在各工况载荷下的位移响应进行进一步分析.

对于建筑结构在地震或其他冲击作用下的响应评估，层间位移角是较为重要的参数指标. 图 14 对比了 5 种工况下建筑结构最大层间位移角分布情况，结果表明：建筑结构响应受爆炸物质量以及爆炸点

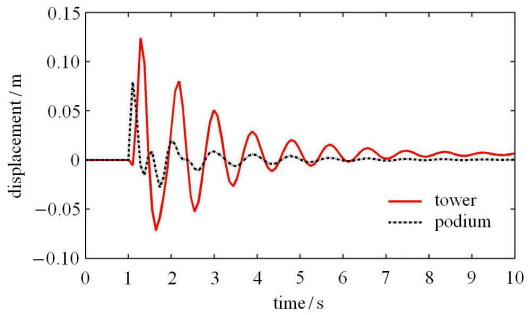


图 13 裙楼及塔楼顶部曲线

Fig. 13 Displacement time history at the podium and the top of tower

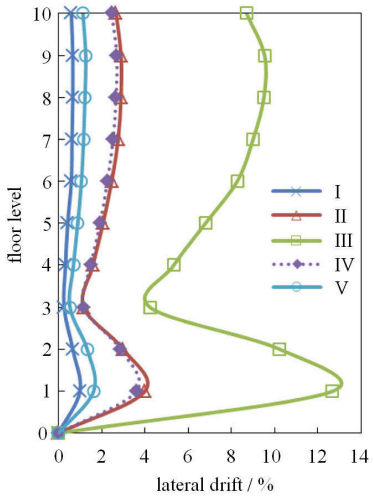


图 14 结构最大层间位移角

Fig. 14 Maximum inter-story drift ratio of the structure

距离影响显著. 工况 II 和 IV 时，结构的最大层间位移角接近 4%. 而工况 III 载荷下的结构变形远远大于其他工况下的的响应，建筑的最大层间位移角甚至已超过 10%，会对建筑物产生较大损害. 同时，从各工况响应数据不难发现：爆炸冲击波在极短时间内向结构物传递了极大的冲击作用，对建筑物的损害甚至超过了相关文献中描述的地震对建筑物的影响. 因此，当爆炸事故发生后，应当对相关建筑物的屈服破坏情况进行检测和评估，排除安全隐患. 同时，也很有必要将数值模拟技术与结构在爆炸载荷下的损伤评估理论结合起来，展开进一步的深入研究.

同时，为了验证所提出算法的计算效率，本文分别采用传统的中心差分算法与本文所提出的显式–隐式异步算法求解了该动力学有限元模型. 通过计算机物理核所消耗的 CPU 时间来统计不同 m 取值时的计算资源消耗，其结果如表 3 所示. 由于冲击载荷只发生在整体动力学模型的局部 (显式分区)，

因而，在整体模型的大部分区域 (隐式分区) 内采用较大时间步长，大大地节约了计算时间. 当隐式时步为显式时步的 10 倍时，总的 CPU 时间消耗减少了约 60%. 这也印证了上文中 “采用大步长计算的节点越多，则理论上获得的计算效率提升也就越大” 这一论述. 因此，如果能将算法合理的并行化，并对显式 – 隐式计算域进行分区优化，以达到负载均衡，必能在求解超大规模问题是更好地发挥本算法的优势，也将是进一步的研究重点.

表 3 CPU 计算时间消耗

Table 3 CPU time consumed

Multiple of timesteps	Explicit	$m = 2$	$m = 5$	$m = 10$
CPU time/s	40 433.7	24 939	18 998	15 937
percentage	100.0%	61.7%	47.0%	39.4%

6 结论与展望

本文根据 Newmark 离散格式显式、隐式算法的固有属性，提出一种适合于在有限元不同分区采用不同时间步长的显式、隐式任意混合算法. 根据不同计算域的物理力学特性和计算精度要求，同时在空间域和时域对模型采用不同的尺度进行数值模拟，可大幅度提高计算效率，减少存储资源消耗.

不同物理力学属性的动力学系统需要合理的选择数值求解方法 (显式或隐式)，系统不同部分的求解精度要求决定了有限元离散网格的尺度，而该尺度又制约了时间积分的步长. 基于统一格式的显式、隐式任意混合异步算法则解决了这一难题.

分区后的异步算法稳定性步长受各计算域内的单元尺寸等复杂因素影响，特别是在非线性计算中，推导合理的稳定临界步长变的尤为困难. 通过能量平衡统计的方法可更好地发现伪能量的产生和不稳定的发展，实时地调整计算步长有助于得到更为合理的计算结果.

在粗网格区域，显式 – 隐式异步算法较之显式 – 显式异步算法具有较为宽松的稳定性条件，可以采用更大的时间步长. 但两种方法具有不同的实用范围，可互为补充. 同时，采用大步长计算的节点越多，则理论上获得的计算效率提升也就越大.

由于本算法采用不断变化的边界节点来处理大、小步长分区间的耦合，一定程度上增加了前处理的难度，因而，进一步的研究方向将是如何更加高效的处理不同尺度空间域、时域的耦合边界.

参 考 文 献

1 Woelke P, Abboud N, Tennant D, et al. Ship impact study: Analytical approaches and finite element modeling. *Shock and Vibration*, 2012, 19(4): 515-525

2 Xu HJ, Liu Y Q, Zhong W. Three-dimensional finite element simulation of medium thick plate metal forming and springback. *Finite Elements in Analysis and Design*, 2012, 51: 49-58

3 Firat M, Karadeniz E, Yenice M, et al. Improving the accuracy of stamping analyses including springback deformations. *Journal of Materials Engineering and Performance*, 2013, 22(2): 332-337

4 Behzad M, Alvandi M, Mba D, et al. A finite element-based algorithm for rubbing induced vibration prediction in rotors. *Journal of Sound and Vibration*, 2013, 332(21): 5523-5542

5 Kacimi A E, Woodward P K, Laghrouche O, et al. Time domain 3D finite element modelling of train-induced vibration at high speed. *Computers & Structures*, 2013, 118:66-73.

6 Kim J, Kang SJ, Kang BS. A comparative study of implicit and explicit FEM for the wrinkling prediction in the hydroforming process. *The International Journal of Advanced Manufacturing Technology*, 2003, 22(7-8): 547-552

7 Oliver J, Huespe AE, Cante JC. An implicit/explicit integration scheme to increase computability of non-linear material and contact/friction problems. *Computer Methods in Applied Mechanics and Engineering*, 2008, 197(21-24): 1865-1889

8 Cai Y, Li G, Wang H, et al. Development of parallel explicit finite element sheet forming simulation system based on GPU architecture. *Advances in Engineering Software*, 2012, 45(1): 370-379

9 Hadoush A, Boogaard AHVD. Efficient implicit simulation of incremental sheet forming. *International Journal for Numerical Methods in Engineering*, 2012, 90(5): 597-612

10 李光耀，王琥，杨旭静等. 板料冲压成形工艺与模具设计制造中的若干前沿技术. 机械工程学报, 2010, 46(10): 31-39 (Li Guangyao, Wang Hu, Yang Xujing, et al. Some new topics on process design and mould manufacture for sheet metal forming. *Journal of Mechanical Engineering*, 2010, 46(10): 31-39 (in Chinese))

11 Noels L, Stainier L, Ponthot JP. Combined implicit/explicit algorithms for crashworthiness analysis. *International Journal of Impact Engineering*, 2004, 30(8-9): 1161-1177

12 Noels L, Stainier L, Ponthot JP, et al. Combined implicit-explicit algorithms for non-linear structural dynamics. *Revue Européenne des Éléments*, 2002, 11(5): 565-591

13 Noels L, Stainier L, Ponthot JP. Energy conserving balance of explicit time steps to combine implicit and explicit algorithms in structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 2006, 195(19-22): 2169-2192

14 Belytschko T, Mullen R. Stability of explicit-implicit mesh partitions in time integration. *International Journal for Numerical Methods in Engineering*, 1978, 12(10): 1575-1586

15 Belytschko T, Yen HJ, Mullen R. Mixed methods for time integration. *Computer Methods in Applied Mechanics and Engineering*, 1979, 17-18(2): 259-275

16 Liu WK, Belytschko T. Mixed-time implicit–explicit finite elements for transient analysis. *Computers and Structures*, 1982, 15(4): 445-450

17 Belytschko T, Liu WK, Smolinski P. Multi-stepping implicit–explicit procedures in transient analysis. In: Proceedings of the In-

ternational Conference on Innovative Methods for Nonlinear Problems. Swansea, U.K.: Pineridge Press International Ltd. 1984

18 Smolinski P. An explicit multi-time step integration method for second order equations. *Computer Methods in Applied Mechanics and Engineering*, 1992, 94(1): 25-34

19 Smolinski P. Subcycling integration with non-integer time steps for structural dynamics problems. *Computers & Structures*, 1996, 59(2): 273-281

20 Daniel WJT. The subcycled Newmark algorithm. *Computational Mechanics*, 1997, 20(3): 272-281

21 Smolinski P, Wu YS. An implicit multi-time step integration method for structural dynamics problems. *Computational Mechanics*, 1998, 22(4): 337-343

22 Wu YS, Smolinski P. A multi-time step integration algorithm for structural dynamics based on the modified trapezoidal rule. *Computer Methods in Applied Mechanics and Engineering*, 2000, 187(3-4): 641-660

23 Daniel WJT. A partial velocity approach to subcycling structural dynamics. *Computer Methods in Applied Mechanics and Engineering*, 2003, 192(3-4): 375-394

24 高晖，李光耀，钟志华等. 汽车碰撞计算机仿真中的子循环法分析. 机械工程学报, 2005, 41(11): 98-101 (Gao Hui, Li Guangyao, Zhong Zhihua, et al. Analysis of subcycling algorithms for computer simulation of crashworthiness. *Chinese Journal of Mechanical Engineering*, 2005, 41(11): 98-101 (in Chinese))

25 缪建成，朱平，陈关龙等. 多柔体系统响应计算的子循环计算方法研究. 力学学报, 2008, 40(4): 511-519(Miao Jiancheng, Zhu Pin, Chen Guanlong, et al. Study on sub-cycling algorithm for flexible multi-body system. *Chinese Journal of Theoretical and Applied Mechanics*, 2008, 40(4): 511-519 (in Chinese))

26 Pugal D, Solin P, Kim KJ, et al. Modeling ionic polymer-metal composites with space-time adaptive multimesh hp-FEM. *Communications in Computational Physics*, 2012, 11: 249-270

27 Hughes TJR. The Finite Element Method: Linear Static and Dynamic Finite Element Analysis. Mineola, NY: Dover Publications Inc, 2000

28 Remennikov AM. A review of methods for predicting bomb blast effects on buildings. *Journal of Battlefield Technology*, 2003, 6(3): 5-10

29 Ngo T, Mendis P, Gupta A, et al. Blast loading and blast effects on structures—An overview. *Electronic Journal of Structural Engineering*, 2007, 7(Special Issue: Loading on Structures): 76-91

30 Saatcioglu M, Ozbakkaloglu T, Naumoski N, et al. Response of earthquake-resistant reinforced-concrete buildings to blast loading. *Canadian Journal of Civil Engineering*, 2009, 36(8): 1378-1390

(责任编辑: 周冬冬)

AN ARBITRARILY MIXED EXPLICIT-IMPLICIT ASYNCHRONOUS INTEGRATION ALGORITHM BASED ON UNIFORM DISCRETIZATION FORMAT [1)]

Zhang Weiwei[*,†] Jin Xianlong[*,†,2)]

[*](*School of Mechanical Engineering*，*Shanghai Jiao Tong University*，*Shanghai* 200240，*China*)

[†](*State Key Laboratory of Mechanical System and Vibration*，*Shanghai Jiao Tong University*，*Shanghai* 200240，*China*)

Abstract Dynamical finite element method requires solving system information at each time step, and the computational effort is much larger than solving the static ones. Thus, to improve computational efficiency and save computational effort is one the of the main research content in dynamics. The present paper introduces an arbitrarily mixed explicit-implicit asynchronous integration algorithm based on uniform Newmark discretization format, for the efficiently solving of the large and complex dynamic systems. The overall dynamical system can be partitioned into different parts according to the physical and mechanical properties, as well as the requirements of solution accuracy, and the system equation can be solved in multi-scale both at the space domain and time domain. According to the inherent message passing mechanisms of the explicit and implicit algorithm, a variable boundary treatment method was adopted to avoid the accumulation of errors at the asynchronous boundary. The simulation time steps were dynamically determined and corrected according to the energy balance checking, which can effectively prevent the emergence and development of the instability. Numerical example shows that the proposed algorithm can greatly reduce the consumption of computing resources while maintaining high accuracy, thus it has a high practical value.

Key words structural dynamics, Newmark discretization, explicit algorithm, implicit algorithm, arbitrarily mixed, asynchronous integration, stability

Received 7 August 2013, revised 11 December 2013.

1) The project was supported by the National High Technology Research and Development Program of China (2012AA01AA307) and the National Natural Science Foundation of China (11072150，61073088).

2) Jin Xianlong, professor, research interests: numerical method and application in structural dynamics, parallel computation. E-mail: jxlong@sjtu.edu.cn