

非线性动力方程精细积分法的自适应步长研究

王海波*, 纪海潮

(中南大学 土木工程学院, 长沙 410075)

摘要:基于 Adams 显式和隐式预估公式实现对时间步长的自适应选择, 利用当前时刻 $v(t_k)$, 采用预估公式的两种形式(显式与隐式), 对 $v(t_{k+1})$ 进行两次预估, 利用两公式局部截断误差关系, 得出误差估计值 $\xi(t_{k+1})$, 并根据其大小自适应调节时间步长。将该思想应用于预估型(求解过程需要用到预估公式)精细积分算法中, 使精细积分算法的时间步长依赖于给定的每步误差限值, 提高计算精度, 且使算法具有很好的稳定性, 对刚度硬化和软化问题均有很好的效果。数值算例验证了本文思想的有效性与适用性。

关键词:非线性动力方程; 自适应步长; 精细积分法; 预估公式

中图分类号: O322; TU311.3

文献标志码: A

文章编号: 1007-4708(2021)03-0371-06

1 引言

钟万铨^[1]提出的精细积分方法, 为非线性系统的时程分析开辟了崭新的途径; 赵秋玲^[2]提出了一种非线性系统线性化的精细积分法; 张洵安等^[3]提出了一种精度较高的线性化迭代精细积分方法; 裘春航等^[4]将非线性部分用 j 次多项式来近似; 李金桥等^[5]先将非线性部分在 t_k 时刻展开成泰勒级数形式, 然后将非线性积分项采用泰勒展开法或高斯-勒让德数值积分法与精细积分法结合进行求解; 王海波等^[6]将精细积分法与 Adams-Bashforth-Moulton 多步法相结合, 提出了一种高精度和高效率的精细积分多步法; 李伟华等^[7]利用改进的欧拉法对状态变量进行预估及校正, 然后结合辛时间子域法对非线性方程进行求解; 江小燕^[8]根据 Hamilton 变作用定律构造了时空有限元矩阵, 提出了求解非线性动力方程的精细时空有限元法; 王海波等^[9]结合精细积分法和 Romberg 数值积分, 提出了一种避免状态矩阵求逆的高效精细积分单步法; 梅树立等^[10]利用龙贝格积分法的自适应性解决时间步长的选择问题, 提出了结构非线性动力方程的自适应精细积分算法; 李健等^[11]基于积分区间逐次半分的思想实现了任意时间步长的自适应求积。

本文基于 Adams 显式和隐式预估公式实现对

时间步长的自适应选择, 使得时间步长依赖于给定的误差限值。算例证明该思想可以应用于预估型(求解过程需要用到预估公式)精细积分算法, 能有效提高计算精度, 同时使得算法具有很好的稳定性, 能解决刚度软化和刚度硬化问题, 具有广泛的适用性。

2 自适应步长的实现原理

2.1 非线性动力方程

非线性动力方程可以表示为

$$\dot{\mathbf{v}} = \mathbf{H}\mathbf{v} + \mathbf{f}(\mathbf{v}, t) \quad (1)$$

利用文献[1]的指数矩阵可将其转化为如下同解积分方程,

$$\mathbf{v}(t_{k+1}) = e^{\mathbf{H}\Delta t} \mathbf{v}(t_k) + \int_{t_k}^{t_{k+1}} e^{\mathbf{H}(t_{k+1}-\tau)} \mathbf{f}(\mathbf{v}, \tau) d\tau \quad (2)$$

式中 $\mathbf{v}(t_{k+1})$ 和 $\mathbf{v}(t_k)$ 分别为所求解向量在时刻 t_{k+1} 和 t_k 的值。

对于式(2), 第一项可直接采用精细算法精确得到, 计算精度非常高, 因此误差的主要来源是对第二项的处理。对于第二项, 精细积分算法会用到预估公式, 如文献[6, 9](先对 $\mathbf{v}(t_{k+j/4})$ ($j=1, 2, 3, 4$) 进行预估, 再进行数值积分的一种精细积分算法)。

为数学上叙述方便, 对积分方程(3)进行讨论。

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \int_{x_n}^{x_{n+1}} \mathbf{f}(x, \mathbf{y}(x)) dx \quad (3)$$

2.2 Adams 显式与隐式预估公式

设由 $r+1$ 个已知的数据点 $(x_n, f_n), (x_{n-1},$

收稿日期: 2020-06-30; 修改稿收到日期: 2020-09-14.

基金项目: 国家自然科学基金(50908230)资助项目.

作者简介: 王海波* (1974-), 男, 博士, 副教授

(E-mail: haibarg@163.com).

$f_{n-1}), \dots, (x_{n-r}, f_{n-r})$ 构造插值多项式 $p_r(x)$, 其中 $f_k = f(x_k, y_k)$, $x_k = x_0 + kh$ (h 为时间步长), 运用插值公式有

$$l_j(x) = \prod_{k=0, k \neq j}^r \frac{x - x_{n-k}}{x_{n-j} - x_{n-k}} \quad (4)$$

$$P_r(x) = \sum_{j=0}^r f_{n-j} l_j(x) \quad (5)$$

得到 $r+1$ 阶 Adams 显式公式,

$$y_{n+1} = y_n + h \sum_{j=0}^r \alpha_{rj} f_{n-j} \quad (6)$$

$$\alpha_{rj} = \frac{1}{h} \int_{x_n}^{x_{n+1}} l_j(x) dx = \int_0^1 \prod_{k=0, k \neq j}^r \frac{t+k}{k-j} dt \quad (j=0, 1, \dots, r)$$

同理, 由 r 个已知的数据点 $(x_n, f_n), (x_{n-1}, f_{n-1}), \dots, (x_{n-r+1}, f_{n-r+1})$ 以及一个未知的数据点 (x_{n+1}, f_{n+1}) , 构造插值多项式 $p_r(x)$, 就可以得到 $r+1$ 阶 Adams 隐式公式。

2.3 Adams 公式的局部误差

$y(x_{n+1})$ 为真实值, y_{n+1} 为基于预估公式对 $y(x_{n+1})$ 的预估值, 误差函数 $T_{n+1} = y(x_{n+1}) - y_{n+1}$ 。构造如下具有 p 阶精度的预估公式:

$$y_{n+1} = \alpha_0 y_n + \alpha_1 y_{n-1} + \dots + \alpha_r y_{n-r} + h(\beta_{-1} f_{n+1} + \beta_0 f_n + \dots + \beta_r f_{n-r}) \quad (7)$$

$$T_{n+1} = y(x_{n+1}) - \left[\sum_{k=0}^r \alpha_k y(x_{n-k}) + h \sum_{k=-1}^r \beta_k y'(x_{n-k}) \right] \quad (8)$$

式(8)右端各项在点 x_n 处作 Taylor 展开, 整理得

$$T_{n+1} = (1 - \sum_{k=0}^r \alpha_k) y(x_n) + \sum_{j=1}^p \frac{h^j}{j!} \left[1 - \sum_{k=1}^r (-k)^j \alpha_k - j \sum_{k=1}^r (-k)^{j-1} \beta_k \right] y^{(j)}(x_n) + \frac{h^{p+1}}{(p+1)!} \left[1 - \sum_{k=1}^r (-k)^{p+1} \alpha_k - (p+1) \sum_{k=1}^r (-k)^p \beta_k \right] \times y^{(p+1)}(x_n) + O(h^{p+2}) \quad (9)$$

当参数 α_k 和 β_k 满足式(10)时, 预估公式达到 p 阶精度。

$$\begin{cases} \sum_{k=0}^r \alpha_k = 1 \\ \sum_{k=1}^r (-k)^j \alpha_k + j \sum_{k=1}^r (-k)^{j-1} \beta_k = 1 \quad (j=1, 2, \dots, p) \end{cases} \quad (10)$$

当 $r=3, p=4$, 取 $\beta_{-1}=0, \alpha_1=\alpha_2=\alpha_3=0$, 由式(10)可得出其他待定参数的方程组, 解之得

$$y_{n+1} = y_n + \frac{h}{24} (55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) \quad (11)$$

$$T_{n+1} = \frac{251}{720} h^5 y^{(5)}(x_n) + O(h^6) \quad (12)$$

式(11)是四阶精度的 Adams 显式公式, 式(12)是其截断误差。同理可以得出各阶精度 Adams 公式的局部截断误差主项, 列入表 1。

表 1 局部截断误差主项

Tab. 1 Local truncation error main term

r	0	1	2	3
Adams 显式公式	$\frac{1}{2} h^2 y''(x_n)$	$\frac{5}{12} h^3 y'''(x_n)$	$\frac{3}{8} h^4 y^{(4)}(x_n)$	$\frac{251}{720} h^5 y^{(5)}(x_n)$
Adams 隐式公式	$-\frac{1}{2} h^2 y''(x_n)$	$-\frac{1}{12} h^3 y'''(x_n)$	$-\frac{1}{24} h^4 y^{(4)}(x_n)$	$-\frac{19}{720} h^5 y^{(5)}(x_n)$

2.4 误差估计及其优点

使用 Adams 显式公式进行预估, 将其结果作为已知的数值点代入 Adams 隐式公式, 再利用 Adams 显式与隐式公式的局部误差公式, 即可得到对一个时间步长的误差估计, 下面以四阶精度的 Adams 显式公式和隐式公式为例。

$$y_{n+1}^p = y_n + \frac{h}{24} (55f_n - 59f_{n-1} + 37f_{n-2} - 9f_{n-3}) \quad (13)$$

$$y_{n+1}^c = y_n + \frac{h}{24} [9f(x_{n+1}, y_{n+1}^p) + 19f_n - 5f_{n-1} - f_{n-2}] \quad (14)$$

$$\begin{cases} y(x_{n+1}) - y_{n+1}^p \approx \frac{251}{720} h^5 y^{(5)}(x_n) \\ y(x_{n+1}) - y_{n+1}^c \approx -\frac{19}{720} h^5 y^{(5)}(x_n) \end{cases} \quad (15)$$

$$y(x_{n+1}) - y_{n+1}^p \approx -\frac{251}{270} (y_{n+1}^p - y_{n+1}^c) \quad (16)$$

$$\xi'(x_{n+1}) = -\frac{251}{270} (y_{n+1}^p - y_{n+1}^c) \quad (17)$$

$$\xi(x_{n+1}) = \max[\text{abs}(\xi'(x_{n+1}))] \quad (18)$$

$$y(x_{n+1}) = y_{n+1}^p + \xi'(x_{n+1}) \quad (19)$$

式中 $y(x_{n+1})$ 为该时间步长最终对 $y(x_{n+1})$ 预估的结果, $\xi(x_{n+1})$ 为对该时间步长的误差估计。

计算时,可调节计算步长,使 $\xi(x_{n+1}) < a$ 。其中 a 为基于实际问题给定的误差控制阈值,即每一时间步要求达到的计算精度。该思想具有以下优点。

(1) 通过选择 $r+1$ 阶 Adams 公式及误差限值 a , 理论上可以实现任意精度的计算结果。

(2) $\xi(x_{n+1}) < a$ 为算法提供一个判断语句,来实现对计算时间步长的自适应选择,从而满足精度的要求。

(3) 式(14~19)的系数均是小于1的数,在进行重复计算时,使得计算误差不会累计扩大,具有良好的稳定性。

3 自适应时间步长精细积分算法

在精细积分算法中,预估环节应用本文第2节的内容,即可实现自适应时间步长,本节以四阶精度的 Adams 公式进行说明。

3.1 流程图及参数分析

a 的取值决定求解的时间步,进而影响求解精度与效率,在给定 a 值时,要注意每步局部误差的大小与采用的 Adams 公式的精度有关,如对于四阶精度的 Adams 公式,其局部误差的大小是时间步长的5次方。在选定 a 值时,要根据实际问题允许的误差范围,再结合采用的 Adams 公式的精度及所希望的时间步长范围综合考虑。

b 为根据结构非线性程度控制时间步长缩小的参数,可以在不同的时段内设置不同的 b 。当函数的阶数越高,就需要越多的控制点才能准确描绘出曲线。求解非线性动力方程得到时程曲线,进而得到位移-时间的函数关系,当某一时间段内的函数阶数较高时,即非线性程度较高时,可以取较小的 b 值,结果是所取时间步长会较小,该时间步长内的函数阶数就会相应降低,预估的结果更加准确。

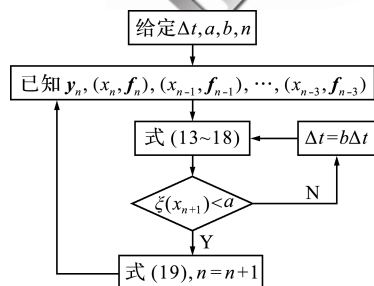


图1 算法流程

Fig. 1 Data flow diagram

3.2 文献[6]自适应步长

为便于说明基于 Adams 显式和隐式预估公式

实现对时间步长的自适应选择思想在精细积分算法中的应用,选择以文献[6]为基础,其步骤简单,思路清晰。

采用四阶精度的 Adams 公式对式(2)的 $v(t_{k+1})$ 进行预估,

$$P(t_{k+1}) = e^{H\Delta t} v(t_k) + w_k \quad (20)$$

$$w_k = \frac{\Delta t}{24} [55F(t_k) - 59F(t_{k-1}) + 37F(t_{k-2}) - 9F(t_{k-3})] \quad (21)$$

$$F(t_{k-i}) = f[v(t_{k-i}), t_{k-i}] \cdot \prod_{j=0}^i E_j \quad (i=0, 1, 2, 3) \quad (22)$$

$$E_j = e^{H(t_{k-j+1}-t_{k-i})} \quad (j=0, 1, 2, 3) \quad (23)$$

式中 $E_j (j=0, 1, 2, 3)$ 采用精细算法^[1]精确得到。

$$m(t_{k+1}) = P(t_{k+1}) + \frac{251}{270} [C(t_k) - P(t_k)] \quad (24)$$

$$\text{令 } F(t_{k+1}) = f[m(t_{k+1}), t_{k+1}],$$

$$C(t_{k+1}) = e^{H\Delta t} v(t_k) + w'_k \quad (25)$$

$$w'_k = \frac{\Delta t}{24} [9F(t_k) + 19F(t_{k-1}) - 5F(t_{k-2}) + F(t_{k-3})] \quad (26)$$

$$\xi(t_{k+1}) = \max \left\{ \text{abs} \left\{ \frac{19}{270} [C(t_{k+1}) - P(t_{k+1})] \right\} \right\} \quad (27)$$

给定误差限值 a , 根据 $\xi(t_{k+1})$ 进行判断,

(1) 当 $\xi(t_{k+1}) \leq a$ 时

$$v(t_{k+1}) = C(t_{k+1}) - \frac{19}{270} [C(t_{k+1}) - P(t_{k+1})] \quad (28)$$

进入下一时间步的计算。

(2) 当 $\xi(t_{k+1}) > a$ 时, 令

$$\Delta t = b\Delta t, E_0 = e^{H\Delta t} \quad (29, 30)$$

返回式(20)重新进行计算。

式(24)的 $(C_k - P_k)$ 存在起步问题, 对 $v(t_5)$ 进行计算时, 令 $C_4 = 0, P_4 = 0$, 其做法对结果的影响可以忽略不计。

本方法为多步法, 需要用单步法计算表头。

4 数值算例

4.1 算例1

刚度软化问题^[12]。已知非线性结构动力方程 $\ddot{x} + 4\pi^2 \times 5^{-t} x = 0$, 初始条件为 $x_0 = 0.1, \dot{x}_0 = 0$, 设 $x = v_1, \dot{x} = v_2$, 相应的一阶微分方程组为

$$\begin{Bmatrix} \dot{v}_1 \\ \dot{v}_2 \end{Bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{Bmatrix} v_1 \\ v_2 \end{Bmatrix} + \begin{Bmatrix} 0 \\ (-1 - 4\pi^2 \cdot 5^{-t}) v_1 \end{Bmatrix}$$

注意, 等步长算法是文献[6]提出的一次预估-

校正精细积分多步法($\Delta t=0.1$ s);变步长算法是以文献[6]提出的一次预估-校正精细积分多步法为基础的自适应步长精细积分算法($a=0.0001$ m, $b=0.5$, $\Delta t=0.1$ s)。

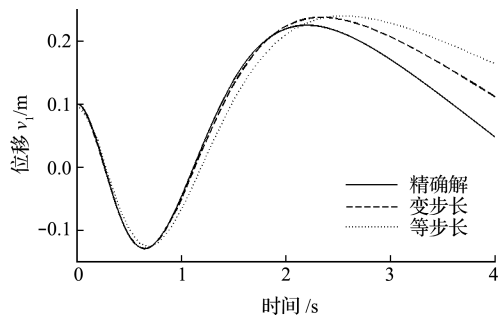


图2 位移时程曲线
Fig.2 Displacement

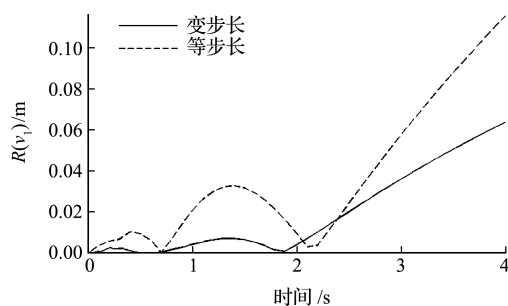


图3 解的绝对误差
Fig.3 Displacement error

图2中,精确解是指积分步长缩小至能计算出有效位数(一般取小数点后8位)相同的计算结果,下同。

表2 各时间段包含时间节点的个数
Tab.2 Number of time nodes contained in each time period

时间段	0 s~1 s	1 s~2 s	2 s~3 s	3 s~4 s
变步长算法	30	12	10	10
等步长算法	13	10	10	10

从图2和图3可以看出,变步长算法的精度比等步长算法的精度高;由表2可知,在0 s~1 s的时间段内,变步长算法的时间步长小于等步长算法的时间步长;证明了基于显式和隐式预估公式实现对时间步长的自适应选择的思路对预估型精细积分算法的有效性,根据给定的误差限值来决定时间步长的大小,即根据非线性程度的大小决定时间步长的大小,从而提高非线性程度较大时间段内的精度,进一步提高整体的计算精度。

4.2 算例2

刚度硬化问题^[12]。已知非线性结构动力方程

$\ddot{x} + \frac{4\pi^2}{9} \times 5^{\frac{3}{2}t} x = 0$, 初始条件为 $x_0=0.1$, $\dot{x}_0=0$, 设 $x=v_1$, $\dot{x}=v_2$, 相应的一阶微分方程组为

$$\begin{cases} \dot{v}_1 \\ \dot{v}_2 \end{cases} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{Bmatrix} v_1 \\ v_2 \end{Bmatrix} + \begin{Bmatrix} 0 \\ (-1 - \frac{4\pi^2}{9} \cdot 5^{\frac{3}{2}t}) v_1 \end{Bmatrix}$$

本节的等步长算法是文献[9]提出的精细积分法($\Delta t=0.02$ s);变步长算法是以文献[9]提出精细积分法为基础的自适应步长精细积分算法($a=0.0001$ m, $b=0.5$, $\Delta t=0.02$ s)(文献[9]所述精细积分法在每个时间步长中需要对 $v(t_{k+j/4})$ ($j=1,2,3,4$) 进行预估,可得出 $\xi(t_{k+j/4})$ ($j=1,2,3,4$),本文令 $\xi(t_{k+1}) = \xi(t_{k+1/4}) + \xi(t_{k+1/2}) + \xi(t_{k+3/4}) + \xi(t_{k+1})$ 。

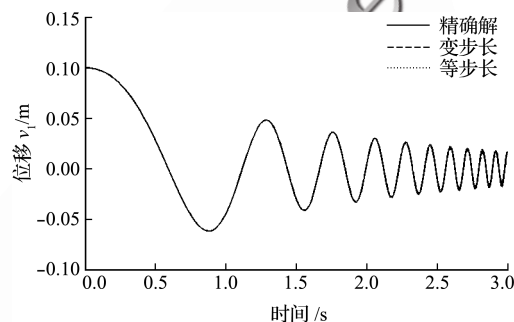


图4 位移时程曲线
Fig.4 Displacement

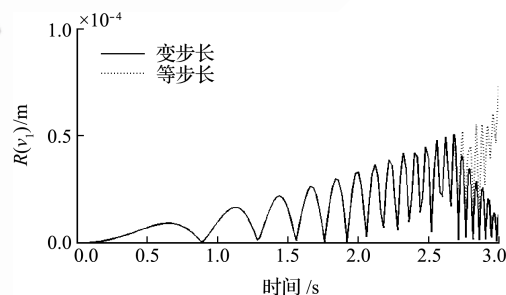


图5 解的绝对误差
Fig.5 Displacement error

表3 各时间段包含时间节点的个数
Tab.3 Number of time nodes contained in each time period

时间段	0 s~0.5 s	0.5 s~1 s	1 s~1.5 s	1.5 s~2 s	2 s~2.5 s	2.5 s~3 s
变步长算法	29	25	25	25	25	88
等步长算法	29	25	25	25	25	25

算例1是刚度软化问题,算例2是刚度硬化问题,而且算例1与算例2分别以文献[6,9]提出的预估型精细积分法为基础,两个算例均有很好的效果,故基于Adams显式和隐式预估公式实现对时间步长的自适应选择的思路具有广泛的适用性。

5 结 论

(1) 本文将基于 Adams 显式和隐式预估公式实现对时间步长的自适应选择思想应用于精细求积算法中,使得算法的计算精度依赖于给定的误差限值,故能根据非线性程度的大小控制时间步长的大小,有效提高非线性程度较大的时间段内的精度,从而提高整体的计算精度。对于初始时间段内非线性程度较大的问题,该思想更加有效。

(2) 应用基于显式和隐式预估公式实现对时间步长的自适应选择的的思想能使精细积分算法具有很好的稳定性。

(3) 本文思想应用于精细积分算法中,能解决刚度硬化和刚度软化等问题,具有广泛的适用性。

参考文献(References):

- [1] 钟万勰. 结构动力方程的精细时程积分法[J]. 大连理工大学学报, 1994, **34**(2): 131-136. (ZHONG Wan-xie. On precise time-integration method for structural dynamics[J]. *Journal of Dalian University of Technology*, 1994, **34**(2): 131-136. (in Chinese))
- [2] 赵秋玲. 非线性动力学方程的精细积分法[J]. 力学与实践, 1998, **20**(6): 24-26 (ZHAO Qiu-ling. An accurate integration method for solving nonlinear dynamics problems[J]. *Mechanics in Engineering*, 1998, **20**(6): 24-26. (in Chinese))
- [3] 张洵安, 姜节胜. 结构非线性动力方程的精细积分算法[J]. 应用力学学报, 2000, **17**(4): 164-168. (ZHANG Xun-an, JIANG Jie-sheng. The precise integration algorithm for nonlinear dynamic equations of structures[J]. *Chinese Journal of Applied Mechanics*, 2000, **17**(4): 164-168. (in Chinese))
- [4] 裘春航, 吕和祥, 钟万勰. 求解非线性动力学方程的分段直接积分法[J]. 力学学报, 2002, **34**(3): 369-378. (QIU Chun-hang, LÜ He-xiang, ZHONG Wan-xie. On segmented-direct-integration method for nonlinear dynamics equations[J]. *Acta Mechanica Sinica*, 2002, **34**(3): 369-378. (in Chinese))
- [5] 李金桥, 于建华. 非线性动力方程避免状态矩阵求逆的级数解[J]. 四川大学学报(工程科学版), 2004, **36**(4): 26-30. (LI Jin-qiao, YU Jian-hua. Series solution of nonlinear dynamics equations avoiding calculating the inversion of the state matrix[J]. *Journal of Sichuan University (Engineering Science Edition)*, 2004, **36**(4): 26-30. (in Chinese))
- [6] 王海波, 余志武, 陈伯望. 非线性动力分析避免状态矩阵求逆的精细积分多步法[J]. 振动与冲击, 2008, **27**(4): 105-107. (WANG Hai-bo, YU Zhi-wu, CHEN Bo-wang. Precise integration multi-step method for nonlinear dynamic equations to avoid calculating inverse of state matrix[J]. *Journal of Vibration and Shock*, 2008, **27**(4): 105-107. (in Chinese))
- [7] 李伟华, 王 堉, 罗 恩. 求解非线性结构动力方程的预估校正—辛时间子域法[J]. 计算力学学报, 2014, **31**(4): 453-458. (LI Wei-hua, WANG Yu, LUO En. Predictor-corrector symplectic time-subdomain algorithm for nonlinear dynamic equations [J]. *Chinese Journal of Computational Mechanics*, 2014, **31**(4): 453-458. (in Chinese))
- [8] 江小燕, 王建国. 非线性动力方程的精细时空有限元方法[J]. 工程力学, 2014, **31**(1): 23-28. (JIANG Xiao-yan, WANG Jian-guo. The precise space-time finite element method for nonlinear dynamic equation [J]. *Engineering Mechanics*, 2014, **31**(1): 23-28. (in Chinese))
- [9] 王海波, 陈 晋, 李少毅. 用于非线性动力分析的一种高效精细积分单步法[J]. 振动与冲击, 2017, **36**(15): 158-162, 229. (WANG Hai-bo, CHEN Jin, LI Shao-yi. An efficient precise integration single-step method for the nonlinear dynamic analysis[J]. *Journal of Vibration and Shock*, 2017, **36**(15): 158-162, 229. (in Chinese))
- [10] 梅树立, 张森文, 徐加初, 等. 非线性动力学方程的自适应精细积分[J]. 暨南大学学报(自然科学版), 2005, **26**(3): 319-323. (MEI Shu-li, ZHANG Sen-wen, XU Jia-chu, et al. Adaptive precise integration algorithm for nonlinear dynamical equation [J]. *Journal of Jinan University (Natural Science)*, 2005, **26**(3): 319-323. (in Chinese))
- [11] 李健, 高广军, 张 洁. 离散精细时程积分的自适应求积算法研究[J]. 应用力学学报, 2014, **31**(4): 502-505. (LI Jian, GAO Guang-jun, ZHANG Jie. Research of adaptive quadrature algorithm for discrete precise time integration method [J]. *Chinese Journal of Applied Mechanics*, 2014, **31**(4): 502-505. (in Chinese))
- [12] 崔雪娜, 王煥定. 刚度解析表达时高阶单步法的研究[J]. 地震工程与工程振动, 2006, **26**(4): 63-67. (CUI Xue-na, WANG Huan-ding. Research on a high order single step method for analytic expression of stiffness [J]. *Earthquake Engineering and Engineering Vibration*, 2006, **26**(4): 63-67. (in Chinese))

Study on adaptive step size of precise intrgration method for nonlinear dynamic equations

WANG Hai-bo*, JI Hai-chao

(School of Civil Engineering, Central South University, Changsha 410075, China)

Abstract: In this paper, the adaptive selection of time steps is realized based on Adams explicit and implicit estimation formulas. For the state equation used for a nonlinear dynamic system, given $v(t_k)$ at the current time instant, two forms of the prediction formula (explicit and implicit) were used to respectively estimate the truncation errors, and the obtain $\xi(t_{k+1})$ was taken as the error of this time step, and the value of the time step was adjusted according to the value of the result. This idea is applied to the precise integration algorithm of the predictive type (the solution process needs to use the prediction formula), making the time step of the precise integration algorithm depend on the given error limit of each step, improving the calculation accuracy, and making the algorithm have good stability, and have good effect on the stiffness hardening and softening problems. Numerical examples demonstrate the validity and applicability of the proposed ideas.

Key words: nonlinear dynamic equations; adaptive step; precise integration; estimation formula

引用本文/Cite this paper:

王海波, 纪海潮. 非线性动力方程精细积分法的自适应步长研究[J]. 计算力学学报, 2021, **38**(3): 371-376.

WANG Hai-bo, JI Hai-chao. Study on adaptive step size of precise intrgration method for nonlinear dynamic equations [J]. *Chinese Journal of Computational Mechanics*, 2021, **38**(3): 371-376.