

大规模动力系统高精度增维精细积分方法快速算法

吴泽艳¹, 王立峰², 武哲^{1,2}

(1. 清华大学 航天航空学院, 北京 100084; 2 北京航空航天大学 航空科学与工程学院, 北京 100191)

摘要: 考虑高精度增维精细积分法求解大规模动力系统快速算法。为提高增维精细积分方法求解大规模动力系统精度, 将非齐次项近似为高阶多项式, 形成高精度增维精细积分方法; 为减少计算时间、提高计算效率, 提出高精度增维精细积分方法快速算法。算例表明, 通过提高非齐次项近似阶数可显著提高计算精度, 快速算法可使计算效率呈量级提高, 高精度快速算法适合大规模动力系统长时间推进计算。

关键词: 大规模动力系统; 增维精细积分; 高精度; 快速算法

中图分类号: O241.8 **文献标识码:** A

Fast algorithm for precise integration with high accuracy dimension expanding method with for large – scale dynamic systems

WU Ze-yan¹, WANG Li-feng², WU Zhe^{1,2}

(1. School of Aerospace, Tsinghua University, Beijing 100084, China;

2. School of Aeronautic Science and Technology, Beihang University, Beijing 100191, China)

Abstract: In order to improve the accuracy of precise integration with dimension expanding method for large scale dynamic systems, the non homogeneous items were approximated by high order polynomials. To reduce the time consuming of calculation, and improve the computational efficiency, a fast algorithm was derived. A numerical example shows that by increasing the approximate order of the non-homogeneous items, the computational accuracy and efficiency can be significantly improved. The algorithm is suited to the long time computation for large scale dynamic systems.

Key words: large-scale dynamic systems; precise integration with dimension expanding method; high accuracy; a fast algorithm

对线弹性结构动力学问题, 较成熟、常用的时域积分方法有 Newmark 方法、Runge-Kutta 方法。因计算稳定性及精度要求, 此两方法的积分步长均较小, 计算量较大。文献[1]提出的求解结构动力方程精细时程积分法求解线性齐次常微分方程, 能获得数值上逼近计算机精度的结果。而求解非齐次常微分方程, 该方法需对方程系数矩阵求逆计算, 当逆不存在时方法失效。

为充分发挥精细积分方法高精度功效, 正致力于研究避免矩阵求逆的精细积分方法, 发展方向为:

(1) 研究 Duhamel 项积分方法。采用数值积分方法对 Duhamel 项进行积分。张森文等^[2-3]利用辛普生积分公式、高斯积分公式求解 Duhamel 积分, 在精细积分方法基础上研究 Duhamel 项的精细积分方法。谭述君等^[4-5]对非齐次项多项式、指数函数、正/余弦函数

给出精确计算方法, 获得数值上逼近计算机精度结果。谭述君等^[6]基于 Duhamel 项的精细积分方法, 构造出求解非线性微分方程的数值算法。

(2) 研究增维精细积分方法。顾元宪等^[7-8]提出增维精细积分法。该方法可避免矩阵求逆, 但增维后系数矩阵是时变的, 只在时间步长非常小时方可近似为定常矩阵。文献[9-12]分别将非齐次项用 Taylor 多项式、Laguerre 多项式、Legendre 多项式及 Chebyshev 多项式展开, 再利用增维方法扩大系数矩阵, 并用精细积分算法求解新的微分方程组, 获得较好计算结果。文献[13]将增维精细积分方法用于非线性系统求解。

由于用增维精细积分方法时增维后系数矩阵是时变的, 如果每次递推计算前均需按精细积分法进行 $L = 20$ 次矩阵与矩阵相乘, 计算量大, 效率低, 对大规模动力系统更是如此。因此需研究增维精细积分方法的快速算法。任伟等^[14]提出将非齐次项作为常数时的快速算法。张继峰等^[15]在增维精细积分法基础上对矩阵进行分块计算, 考虑非齐次项特点, 减小矩阵维数, 实现

简化计算。将非齐次项近似为多项式研究中,文献[4]给出的递推计算公式精度高、算法快速。文献[16]以精细积分方法为基础,利用大规模动力系统矩阵的稀疏性及动力物理特性,分析矩阵指数的特殊结构,并给出计算大规模动力系统矩阵指数及动力响应的高效率方法。为进一步提高计算精度,本文研究将非齐次项近似为高次多项式(以二次多项式为例)基于增维精细积分的快速算法,获得与文献[4]相同结果。该算法不依赖于方程的物理特性,且具有高精度。

1 精细积分方法^[1]

考虑齐次线性常微分方程初值问题:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{A}\mathbf{x} \\ \mathbf{x}(t_0) = \mathbf{x}_0 \end{cases} \quad (1)$$

式中: \mathbf{A} 为 n 阶常数矩阵; \mathbf{x} 为 n 维列向量。据常微分方程理论,式(1)的解为:

$$\mathbf{x} = e^{\mathbf{A}t} \mathbf{x}_0 \quad (2)$$

令时间步长为 τ ,一系列等步长 τ 的时刻为:

$$t_0 = 0, t_1 = \tau, \dots, t_k = k\tau, \dots$$

则:

$$\mathbf{x}(\tau) = \mathbf{x}_1 = \mathbf{T}\mathbf{x}_0, \mathbf{T} = e^{\mathbf{A}\tau} \quad (3)$$

及递推公式:

$$\mathbf{x}_k = \mathbf{T}\mathbf{x}_{k-1} \quad (4)$$

问题归结为式(3)中 \mathbf{T} 阵的计算。在较小时时间间隔 $[t_{k-1}, t_k]$ 内计算 \mathbf{T} 阵:

$$\mathbf{T} = e^{\mathbf{A}\tau} = (e^{\frac{\tau}{m}\mathbf{A}})^m \quad (5)$$

其中: $m = 2^L$ 。若 $L = 20$,则 $m = 1\,048\,576$ 。令 $\mu = \tau/m$,因 τ 很小, m 很大,故 μ 很小。在 μ 区间段内,有:

$$\begin{aligned} e^{\mu\mathbf{A}} &\approx \mathbf{I} + \mathbf{A}\mu + (\mathbf{A}\mu)^2/2 + \\ &(\mathbf{A}\mu)^3/6 + (\mathbf{A}\mu)^4/24 = \mathbf{I} + \mathbf{T}_a \end{aligned} \quad (6)$$

其中:

$$\begin{aligned} \mathbf{T}_a &= \mathbf{A}\mu + (\mathbf{A}\mu)^2/2 + (\mathbf{A}\mu)^3/6 + \\ &(\mathbf{A}\mu)^4/24 \end{aligned} \quad (7)$$

令:

$$\mathbf{T} = e^{\tau\mathbf{A}} = (\mathbf{I} + \mathbf{T}_a)^{2^L} = (\mathbf{I} + \mathbf{T}_a)^{2^{L-1}} (\mathbf{I} + \mathbf{T}_a)^{2^{L-1}} \quad (8)$$

且 $(\mathbf{I} + \mathbf{T}_b)(\mathbf{I} + \mathbf{T}_c) = \mathbf{I} + \mathbf{T}_b + \mathbf{T}_c + \mathbf{T}_b\mathbf{T}_c$,因此求解 \mathbf{T} 即等同于执行语句为:

$$\text{for}(i = 1; i < L; i++) \quad \mathbf{T}_a = 2\mathbf{T}_a + \mathbf{T}_a\mathbf{T}_a$$

循环结束时执行:

$$\mathbf{T} = \mathbf{I} + \mathbf{T}_a \quad (9)$$

由此获得 \mathbf{T} 阵。此即求解齐次常微分方程组初值问题的精细积分方法。

2 高精度增维精细积分方法快速算法

设系统刚度矩阵、阻尼矩阵、质量矩阵分别为 \mathbf{k} ,

\mathbf{C}, \mathbf{M} ,则结构动力学方程为:

$$\mathbf{M}\ddot{\mathbf{y}} + \mathbf{C}\dot{\mathbf{y}} + \mathbf{k}\mathbf{y} = \mathbf{f}(t)$$

式中: $\mathbf{f}(t)$ 为外力。方程可改写为:

$$\dot{\mathbf{v}} = \mathbf{H}\mathbf{v} + \mathbf{g}$$

$$\text{其中: } \mathbf{v} = \begin{Bmatrix} \mathbf{y} \\ \dot{\mathbf{p}} \end{Bmatrix}, \mathbf{H} = \begin{bmatrix} 0 & \mathbf{M}^{-1} \\ -\mathbf{K} & -\mathbf{C}\mathbf{M}^{-1} \end{bmatrix}, \mathbf{g} = \begin{Bmatrix} 0 \\ \mathbf{f} \end{Bmatrix}.$$

式中: $\mathbf{p} = \mathbf{M}\mathbf{y}$ 为动量。

考虑初始条件的结构动力学方程可写为:

$$\frac{d\mathbf{v}}{dt} = \mathbf{H}\mathbf{v} + \mathbf{g}, \mathbf{v}(t_0) = \mathbf{v}_0 \quad (10)$$

式中: \mathbf{H} 为 n 阶常数矩阵; \mathbf{g} 仅为时间 t 的函数。

将 \mathbf{g} 在时刻 t_k 处作Taylor展开:

$$\begin{aligned} \mathbf{g}(t) &= \mathbf{g}(t_k) + \mathbf{g}'(t_k)(t - t_k) + \\ &\frac{1}{2}\mathbf{g}''(t_k)(t - t_k)^2 + o[(t - t_k)^3] \end{aligned} \quad (11)$$

略去式(11)中高阶项后代入式(10)得:

$$\frac{d\mathbf{U}}{dt} = \mathbf{A}\mathbf{U} \quad (12)$$

式中:

$$\mathbf{U} = \begin{bmatrix} \mathbf{v} \\ 1 \\ t - t_k \\ \frac{1}{2}(t - t_k)^2 \end{bmatrix}, \mathbf{A} = \begin{bmatrix} \mathbf{H} & \mathbf{g}_0 & \mathbf{g}_1 & \mathbf{g}_2 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (13)$$

$$\mathbf{g}_0 = \mathbf{g}(t_k), \mathbf{g}_1 = \mathbf{g}'(t_k), \mathbf{g}_2 = \mathbf{g}''(t_k)$$

式(12)在形式上表现为线性齐次常微分方程组,可用精细积分方法求解。但在不同时间段内矩阵 \mathbf{A} 不同,若每次按精细积分法做 $L = 20$ 次矩阵与矩阵相乘,计算量较大,效率较低。以下导出求解矩阵 \mathbf{T}_a 的快速算法。

由式(12)中矩阵 \mathbf{A} 的定义计算得:

$$\begin{aligned} \mathbf{A}^2 &= \begin{bmatrix} \mathbf{H}^2 & \mathbf{H}\mathbf{g}_0 + \mathbf{g}_1 & \mathbf{H}\mathbf{g}_1 + \mathbf{g}_2 & \mathbf{H}\mathbf{g}_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \\ \mathbf{A}^3 &= \begin{bmatrix} \mathbf{H}^3 & \mathbf{H}^2\mathbf{g}_0 + \mathbf{H}\mathbf{g}_1 + \mathbf{g}_2 & \mathbf{H}^2\mathbf{g}_1 + \mathbf{H}\mathbf{g}_2 & \mathbf{H}^2\mathbf{g}_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ \mathbf{A}^4 &= \begin{bmatrix} \mathbf{H}^4 & \mathbf{H}^3\mathbf{g}_0 + \mathbf{H}^2\mathbf{g}_1 + \mathbf{H}\mathbf{g}_2 & \mathbf{H}^3\mathbf{g}_1 + \mathbf{H}^2\mathbf{g}_2 & \mathbf{H}^3\mathbf{g}_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \end{aligned}$$

记:

$$\mathbf{P}_0 = \mu\mathbf{I} + (\mu^2/2)\mathbf{H} + (\mu^3/6)\mathbf{H}^2 + (\mu^4/24)\mathbf{H}^3 \quad (14a)$$

$$Q_0 = (\mu^2/2)I + (\mu^3/6)H + (\mu^4/24)H^2 \quad (14b)$$

$$R_0 = (\mu^3/6)I + (\mu^4/24)H \quad (14c)$$

$$a_0 = \mu \quad b_0 = \mu^2/2 \quad c_0 = \mu \quad (14d)$$

由(7)式有:

$$T_a = \begin{bmatrix} P_0H & P_0g_0 + Q_0g_1 + R_0g_2 & P_0g_1 + Q_0g_2 & P_0g_2 \\ 0 & 0 & 0 & 0 \\ 0 & a_0 & 0 & 0 \\ 0 & b_0 & c_0 & 0 \end{bmatrix} \quad (15)$$

执行一次 $T_a = 2T_a + T_a T_a$ 计算得:

$$T_a^{(1)} = \begin{bmatrix} P_1H & P_1g_0 + Q_1g_1 + R_1g_2 & P_1g_1 + Q_1g_2 & P_1g_2 \\ 0 & 0 & 0 & 0 \\ 0 & a_1 & 0 & 0 \\ 0 & b_1 & c_1 & 0 \end{bmatrix} \quad (16)$$

式中:

$$\left. \begin{aligned} P_1 &= P_0HP_0 + 2P_0 \\ Q_1 &= P_0HQ_0 + a_0P_0 + 2Q_0 = P_0HQ_0 + c_0P_0 + 2Q_0 \\ R_1 &= P_0HR_0 + a_0Q_0 + b_0P_0 + 2R_0 \\ a_1 &= 2a_0 \quad b_1 = a_0c_0 + 2b_0 \quad c_1 = 2c_0 \end{aligned} \right\} \quad (17)$$

比较式(15)、(16)可知,每执行一次更新计算所得新矩阵 T_a 形式不变。由此得计算 T_a 的快速算法:

$$\left. \begin{aligned} \text{for}(i = 1; i \leq L; i++) \\ P_i &= P_{i-1}HP_{i-1} + 2P_{i-1} \\ Q_i &= P_{i-1}HQ_{i-1} + a_{i-1}P_{i-1} + 2Q_{i-1} \\ R_i &= P_{i-1}HR_{i-1} + a_{i-1}Q_{i-1} + b_{i-1}P_{i-1} + 2R_{i-1} \\ a_i &= 2a_{i-1}, \quad b_i = a_{i-1}c_{i-1} + 2b_{i-1}, \quad c_i = 2c_{i-1} \end{aligned} \right\} \quad (18)$$

式(18)循环结束后,得:

$$T_a^{(L)} = \begin{bmatrix} P_LH & P_Lg_0 + Q_Lg_1 + R_Lg_2 & P_Lg_1 + Q_Lg_2 & P_Lg_2 \\ 0 & 0 & 0 & 0 \\ 0 & a_L & 0 & 0 \\ 0 & b_L & c_L & 0 \end{bmatrix} \quad (19)$$

由式(6)得: $T_{(n+3)(n+3)} = I_{(n+3)(n+3)} + T_a^{(L)}$

从而有:

$$v_{k+1} = (P_LH + I_{n \times n})v_k + P_Lg_0 + Q_Lg_1 + R_Lg_2 \quad (20)$$

式(20)即将非齐次项近似为二次多项式后的递推计算公式。递推计算前,先计算 P_L, Q_L, R_L, P_LH 并保存。同文献[4],乘法计算量亦为 $4L$ 次矩阵与矩阵相乘。此后的每个时间步,每次递推计算只需 4 次矩阵与向量相乘运算,避免 $L = 20$ 次矩阵与矩阵相乘,计算量由 $L(n+3)^3$ 减小到 $4n^2$,效率提高显著。其中 n 为

矩阵 H 的阶数。若需进一步提高计算精度,可保留式(11)中更多 Taylor 展开项,亦能获得类似结果。

3 算例

3.1 算例 1

考虑动力系统:

$$\frac{dx}{dt} = Hx + f(t), \quad x(t_0) = x_0 = 0$$

其中:

$$H = \begin{bmatrix} c & b & & \\ a & c & b & \\ & \cdot & \cdot & \cdot \\ & & a & c & b \\ & & & a & c \end{bmatrix}_{n \times n}$$

记:

$$\lambda_i = c + 2[\text{sgn}(b)]\sqrt{abc}\cos\frac{i\pi}{n+1}$$

$$r_i = \begin{bmatrix} \left(\frac{a}{b}\right)^{\frac{1}{2}} \sin\frac{i\pi}{n+1} & \left(\frac{a}{b}\right) \sin\frac{2i\pi}{n+1} & \dots \\ \left(\frac{a}{b}\right)^{\frac{j}{2}} \sin\frac{j\pi}{n+1} & \dots & \left(\frac{a}{b}\right)^{\frac{n}{2}} \sin\frac{ni\pi}{n+1} \end{bmatrix}^T$$

($i = 1, 2, \dots, n$)

$$\Phi(t) = [e^{\lambda_1 t} r_1, \dots, e^{\lambda_n t} r_n]$$

$$f(t) = \Phi(t)[1, \dots, 1]^T$$

由于 λ_i, r_i 为矩阵 H 的特征值及特征向量,故方程有解析解为:

$$x(t) = \Phi(t)(t-t_0)[1, \dots, 1]^T$$

式中: n 代表方程组规模。

取 $a = b = 1.0, c = -2.0, \Delta t = 0.01, n = 100$ 。为显示算法的精度,在每个时间段分别将非齐次项近似为常数及一、二次多项式,递推计算 100 步,相对误差见图 1。由图 1 看出,计算精度随近似阶的提高显著提高。

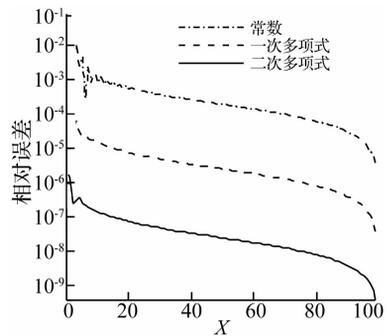


图 1 相对误差
Fig. 1 Relative error

为显示快速算法的高效性,分别用 MATLAB 的 ode45 函数、经典增维精细积分法与快速算法求解该动力系统。ode45 基于 Runge-Kutta 法,可自适应调节步长,精度设置为 10^{-7} 。经典增维精细积分法与快速算法均将非齐次项近似为二次多项式,时间步长为 0.01。计算规模 n 取不同值时用三种方法在时间方向递推计算 1 s 的耗时见表 1。其中快速算法的准备时间指递推前计算式(20)中各矩阵的时间。由表 1 看出,随计算规模的增大,与经典增维精细积分算法相比,快速算法的计算时间呈量级减少;快速算法耗时主要集中在递推计算前的数据准备阶段,计算时间少于 ode45。因此推进计算时间越长,快速算法的优越性越显著。

表 1 计算时间对比(单位 s)
Tab.1 Calculate time

计算规模 n	100	200	500	1 000
ode45	<1	2	86	842
经典增维算法	11	76	1 010	23 010
快速				
准备时间	<1	2	31	769
递推时间	<1	<1	6	21
总时间	<1	<3	37	790

3.2 算例 2^[4]

设弦的张力为 T ,用 $4n + 3$ 个节点将弦等分为 $(4n + 3) + 1$ 段,每个节点布置一个质量 $m/(4n + 3)$,不计弦自重;在弦的 $1/4, 1/2, 3/4$ 处分别受正/余弦函数形式的外部激励,见图 2。

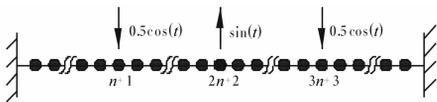


图 2 振动模型

Fig.2 Model for vibration

记 $k = T/L$,振动方程为:

$$M\ddot{x} + k\dot{x} = f_1 \sin(\omega t) + f_2 \cos(\omega t)$$

其中:系统参数为:

$$M = \frac{1}{4n + 3} \begin{bmatrix} m & & & & \\ & m & & & \\ & & \ddots & & \\ & & & m & \\ & & & & m \end{bmatrix}_{(4n+3) \times (4n+3)}$$

$$K = \begin{bmatrix} 2k & -k & & & \\ -k & 2k & -k & & \\ & & \ddots & & \\ & & & -k & 2k & -k \\ & & & -k & 2k \end{bmatrix}_{(4n+3) \times (4n+3)}$$

$$f_1 = [\underbrace{0 \cdots 0}_n \quad 0 \quad \underbrace{0 \cdots 0}_n \quad 1 \quad \underbrace{0 \cdots 0}_n \quad 0 \quad \underbrace{0 \cdots 0}_n]^T$$

$$f_2 = [\underbrace{0 \cdots 0}_n \quad -0.5 \quad \underbrace{0 \cdots 0}_n \quad 0 \quad \underbrace{0 \cdots 0}_n \quad -0.5 \quad \underbrace{0 \cdots 0}_n]^T$$

参数 $k=2, m=1, \omega=3$,系统初始值为:

$$x(0) = 0, \dot{x}(0) = 0$$

将方程降阶后获得与式(10)相同形式:

$$\frac{d}{dt} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & -M^{-1}K \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0 \\ M^{-1}[f_1 \sin(\omega t) + f_2 \cos(\omega t)] \end{bmatrix}$$

上式初始条件为: $x(0) = 0, y(0) = 0$ 。

为显示算法精度,在每个时间段分别将非齐次项近似为常数及一、二次多项式,取 $\Delta t = 0.01, n = 10$,递推计算 100 步,以常微分方程组的小步长精细积分为参考计算各算法误差:

$$\frac{d}{dt} \begin{bmatrix} x \\ y \\ \sin(\omega t) \\ \cos(\omega t) \end{bmatrix} = \begin{bmatrix} 0 & I & 0 & 0 \\ 0 & -M^{-1}K & M^{-1}f_1 & M^{-1}f_2 \\ 0 & 0 & 0 & \omega \\ 0 & 0 & -\omega & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \sin(\omega t) \\ \cos(\omega t) \end{bmatrix}$$

该参考解精度已达 10^{-15} 量级。各算法绝对误差见图 3。由图 3 看出,计算精度随近似阶的提高显著提高。

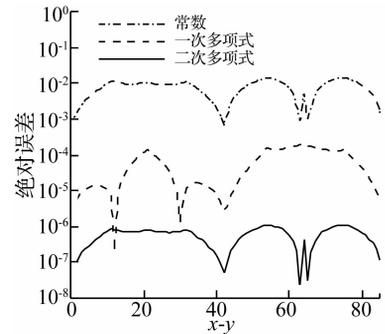


图 3 绝对误差

Fig.3 Absolute error

为显示快速算法的高效性,分别用 MATLAB 的 ode45 函数、经典增维精细积分法与快速算法求解该动力系统。ode45 精度设置为 10^{-6} 。经典增维精细积分法与快速算法均将非齐次项近似为二次多项式,时间步长为 0.01。表 2 为 n (计算规模为 $2(4n + 3)$) 取不同值时用三种方法在时间方向递推计算 5 s 的时间消耗。由表 2 可得与算例 1 相同结论。若时间方向递推时间少于 5 s,则 ode45 计算时间少于快速算法。限于篇幅,不进行时间对比。因此,快速算法适合大规模动力系统长时间递推计算。

表2 计算时间对比(单位 s)

Tab.2 Calculate time

<i>n</i>	10	25	50	100
ode45	2	8	52	439
经典增维算法	10	199	2 007	1 129 × 50
快速 算 法	准备时间	<1	14	398
	递推时间	<1	1	4
	总时间	<1	15	402

4 结 论

(1) 本文通过将非齐次项近似为关于时间的高阶多项式, 形成高精度增维精细积分方法, 提高增维精细积分方法求解动力系统精度; 推导出高精度增维精细积分方法的快速算法计算公式, 获得与已有文献相同结果。快速算法与动力系统物理特性无关, 具有一般性。

(2) 通过求解计算规模可调节的动力学问题, 对比 MATLAB ode45、快速算法与经典增维精细积分方法计算时间表明, 所提快速算法具有高精度性、高效性, 适合大规模结构动力学问题长时间推进计算。

参 考 文 献

[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.

[2] 张森文, 曹开彬. 计算结构动力响应的状态方程直接积分法[J]. 计算力学学报, 2000, 17(1): 94-97.
ZHANG Sen-wen, CAO Kai-bin. Direct integration of state equation method for dynamic response of structure [J]. Chinese Journal of Computational Mechanics, 2000, 17(1): 94-97.

[3] 汪梦甫, 周锡元. 结构动力方程的高斯精细时程积分法[J]. 工程力学, 2004, 21(4): 13-16.
JIANG Meng-fu, ZHOU Xi-yuan. Gauss precise time-integration of structural dynamic analysis [J]. Engineering Mechanics, 2004, 21(4): 13-16.

[4] 谭述君, 钟万勰. 非齐次动力方程 Duhamel 项的精细积分[J]. 力学学报, 2007, 39(3): 374-381.
TAN Shu-jun, ZHONG Wan-xie. Precise integration method for Duhamel terms arising from non-homogenous dynamic systems [J]. Acta Mechanica Sinica, 2007, 39(3): 374-381.

[5] 富明慧, 刘作秋, 林敬华. 一种广义精细积分法[J]. 力学学报, 2007, 39(5): 672-677.
FU Ming-hui, LIU Zuo-qiu, LIN Jing-hua. A generalized precise time step integration method [J]. Acta Mechanica Sinica, 2007, 39(5): 672-677.

[6] 谭述君, 高 强, 钟万勰. Duhamel 项的精细积分方法在非线形微分方程数值求解中的应用[J]. 计算力学学报, 2010, 27(5): 752-758.
TAN Shu-jun, GAO Qiang, ZHONG Wan-xie. Applications of Duhamel term's precise integration method in solving nonlinear differential equations [J]. Chinese Journal of Computational Mechanics, 2010, 27(5): 752-758.

[7] 顾元宪, 陈彪松, 张洪武. 结构动力方程的增维精细积分法[J]. 力学学报, 2000, 32(4): 447-456.
GU Yuan-xian, CHEN Biao-song, ZHANG Hong-wu. Precise time-integration with dimension expending method [J]. Acta Mechanica Sinica, 2000, 32(4): 447-456.

[8] Gu Y X, Chen B S, Zhang H W, et al. Precise time-integration method with dimensional expanding for structural dynamic equations [J]. AIAA Journal, 2001, 39(12): 2394-2399.

[9] 向 宇, 黄玉盈, 黄健强. 一种新型齐次扩容精细积分法[J]. 华中科技大学学报(自然科学版), 2002, 30(11): 74-76.
XIANG Yu, HUANG Yu-ying, HUANG Jian-qiang. A method of homo-genization of high precision direct integration [J]. Journal Huazhong Univ. of Sci. & Tech. (Nature Science Edition), 2002, 30(11): 74-76.

[10] 冯 凤, 周 钢. 半无限时域上任意右端激励的长效精细算法[J]. 上海工程技术大学学报, 2005, 19(3): 215-219.
FENG Feng, ZHOU Gang. Long acting HHPD with arbitrary right impetus in semi-infinite time domain [J]. Journal of Shanghai University of Engineering Science, 2005, 19(3): 215-219.

[11] 时小红, 周 钢, 付召华. 基于 Legendre 多项式函数系的齐次扩容精细算法[J]. 计算力学学报, 2005, 22(3): 355-338.
SHI Xiao-hong, ZHOU Gang, FU Zhao-hua. A homogeneous high precise direct integration based on Legendre polynomial series [J]. Chinese Journal of Computational Mechanics, 2005, 22(3): 355-338.

[12] 付召华, 周 钢, 罗 顺, 等. 基于 Chebyshev 多项式函数系的齐次扩容精细算法[J]. 东华大学学报, 2006, 32(2): 46-49.
FU Shao-hua, ZHOU Gang, LUO Shun, et al. A homogenized high precision direct integration based on Chebyshev polynomial series [J]. Journal of Donghua University, 2006, 32(2): 46-49.

[13] 张素英, 邓子辰. 非线性动力方程的增维精细积分法[J]. 计算力学学报, 2003, 20(4): 421-426.
ZHANG Su-ying, DENG Zi-chen. Increment dimensional precise integration method for nonlinear dynamic equation [J]. Journal of Computational Mechanics, 2003, 20(4): 421-426.

[14] 任 伟, 杜铁钧. 定常结构动力方程增维精细积分法求解的注记[J]. 杭州电子科技大学学报, 2005, 25(1): 41-43.
REN Wei, DU Tie-jun. A note on increment dimensional precise integration method for invariant dynamic equation [J]. Journal of Hangzhou Dianzi University, 2005, 25(1): 41-43.

[15] 张继锋, 邓子辰. 结构动力方程的增维分块精细积分法[J]. 振动与冲击, 2008, 27(12): 88-90.
ZHANG Ji-feng, DENG Zi-chen. Dimensional increment and partitioning precise integration method for structural dynamic equation [J]. Chinese Journal of Vibration and Shock, 2008, 27(12): 88-90.

[16] 高 强, 吴 锋, 张洪武, 等. 大规模动力系统改进的快速精细积分方法[J]. 计算力学学报, 2011, 28(4): 493-498.
GAO Qiang, WU Feng, ZHANG Hong-wu, et al. A fast precise integration method for large-scale dynamic structures [J]. Chinese Journal of Computational Mechanics, 2011, 28(4): 493-498.