变质量系统的自适应李级数算法

邢誉峰^{①*}, 王宇楠^①, 潘忠文^②, 董锴^②

① 北京航空航天大学固体力学研究所, 北京 100191;

② 北京宇航系统工程研究所, 北京 100076

* E-mail: xingyf@buaa.edu.cn

收稿日期: 2014-02-25; 接受日期: 2014-03-21

国家自然科学基金(批准号: 11172028, 11372021)和国防 973 计划(批准号: 613133)资助项目

摘要 运载火箭是典型变质量系统, 其固有振动频率随着推进剂快速减少而迅速升高. 针对变质量系统, 本文提出了求解其变系数线性常微分动力学方程的自适应李级数直接积分算法, 给出了多自由度系统的矩阵递推格式; 若步长不变, 根据精度要求, 这种方法能够随着系统频率的升高自动增加阶次而保证精度不变; 也可以阶次不变, 自动减少步长.

关键词变质量系统
李级数算法
直接积分方法
自适应

1 引言

在线弹性和小变形范围内, 变质量系统由变系数线性微分方程来控制, 对于某一冻结状态或关键秒状态, 模态正交性和模态叠加方法是成立的. 对这种变系数线性系统, 若考虑时变动力学效应, 正交性及叠加方法不再成立. 时变运载火箭系统是典型的变系数线性系统, 在任意微小时段, 该系统可以等效成为定常线性系统, 其模态满足正交性并具有可叠加性. 因此, 研究运载火箭某一时刻的模态和频率以确定其姿态和刚度, 以及用模态来计算某一时刻的载荷, 其结果是有根据且值得信赖.

类似于时不变系统, 可以采用直接积分方法和模态叠加方法逐步递推计算变质量系统的动力学响应^[1]. 随着系统固有振动频率的升高, 通常需要逐渐减小步长或增加模态数来得到足够精确的数值结果. 为了使直接积分方法能够反映系统动态特性的变化,

邢誉峰和郭静^[2]发展了一种与结构动特性协同的自适应 Newmark 算法, 其算法参数可以随着系统频率的改变而改变. 当然, 从原则上来讲, 只要步长足有小, 一般直接积分方法也可以用来分析变质量系统的动态响应, 但要牺牲计算效率.

李级数方法^[3~6]克服了高阶时间导数计算的困难, 便于构造任意高阶次的动力学算法. 针对时变多自由度系统, 本文给出了基于李级数方法的矩阵递推格式, 根据误差容限能够自适应地判断算法的阶次或展开项数. 算例说明了该算法的高精度特性和实用性.

2 自适应李级数算法

变质量系统的动力学方程为^[1]

$$M\ddot{x} + (\dot{M} + C)\dot{x} + Kx = \dot{M}u + R(t), \quad (1)$$

引用格式: 邢誉峰, 王宇楠, 潘忠文, 等. 变质量系统的自适应李级数算法. 中国科学: 技术科学, 2014, 44: 532–536
Xing Y F, Wang Y N, Pan Z W, et al. Self-adaptive Lie series method for mass-variable systems (in Chinese). Sci Sin Tech, 2014, 44: 532–536, doi: 10.1360/N092014-00033

其中, $\dot{M}u$ 相当于推进剂减少产生的推力. 物理阻尼 $C\dot{x}$ 耗散暂态响应携带的能量. 推进剂对运载火箭纵横和扭转具有不同作用^[7,8], 推进剂消耗引起的阻尼效应 $\dot{M}\dot{x}$ 可以忽略不计. 因此下面仅考虑如下形式的多自由度系统控制微分方程

$$M\ddot{x} + Kx = F, \quad (2)$$

其中位移向量为 $x^T = [x_1 \ x_2 \ \cdots \ x_n]$, n 为系统自由度数, $F = \dot{M}u + R(t)$. 令速度向量 $\dot{x} = y$. 构造如下向量

$$\begin{aligned} z^T &= [z_1 \ \cdots \ z_n, z_{n+1} \ \cdots \ z_{2n}] \\ &= [x_1 \ \cdots \ x_n, y_1 \ \cdots \ y_n], \end{aligned} \quad (3)$$

方程(2)可以变为

$$\frac{dz}{dt} = f(z). \quad (4)$$

式中

$$f = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \quad (5)$$

$$f_1 = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = y, \quad f_2 = \begin{bmatrix} f_{n+1} \\ f_{n+2} \\ \vdots \\ f_{2n} \end{bmatrix} = B + Ax. \quad (6)$$

其中 $A = -M^{-1}K$, $B = M^{-1}F$. 记 B_i 和 A_i 分别为 B 和 A 的第 i 个元素和第 i 行向量. 根据复合求导法则, 可以得到李级数中的线性微分演化算子为

$$\begin{aligned} L(z) &= f_1(z) \frac{\partial}{\partial z_1} + \cdots + f_{2n}(z) \frac{\partial}{\partial z_{2n}} \\ &= z_{n+1} \frac{\partial}{\partial z_1} + \cdots + z_{2n} \frac{\partial}{\partial z_n} + (B_1 + A_1 x) \frac{\partial}{\partial z_{n+1}} + \cdots \\ &\quad + (B_n + A_n x) \frac{\partial}{\partial z_{2n}}. \end{aligned} \quad (7)$$

从(7)式可以得到如下规律

$$L^i x = L^{i-1} y = \begin{cases} y, & i=1, \\ A^{\frac{i-1}{2}} (B + Ax), & i \geq 2 \text{ 为偶数}, \\ A^{\frac{i-1}{2}} y, & i \geq 3 \text{ 为奇数}. \end{cases} \quad (8)$$

其中阶数 i 为正整数. 设时间步长为 h , 已知对应 $t_k = t_0 + kh$ 时刻 z^k 的值, 则李级数递推格式为

$$\begin{aligned} x^{k+1} &= \left[\sum_{i=1}^{\infty} \frac{h^{i-1}}{(i-1)!} L^{i-1} x \right]_{z=z^k} \\ &= \left[x + hy + \frac{1}{2!} h^2 (B + Ax) + \frac{1}{3!} h^3 Ay + \cdots \right]_{z=z^k}. \end{aligned} \quad (9)$$

对(9)式两端施加微分算子 L 得速度, 即

$$\begin{aligned} y^{k+1} &= \left[\sum_{i=1}^{\infty} \frac{h^{i-1}}{(i-1)!} L^{i-1} y \right]_{z=z^k} \\ &= \left[y + h(B + Ax) + \frac{1}{2!} h^2 Ay + \cdots \right]_{z=z^k}, \end{aligned} \quad (10)$$

式中 $k=0,1,2,\dots$ 表示时间步数. 若选取前 m 项(m 为偶数), 相当于算法为 $m-1$ 阶, 则 x^{k+1} 的截断误差主项为

$$R^{k+1} = O(h^m) = \frac{h^m}{(m)!} L^m x^k, \quad (11)$$

该误差可以准确计算. 综上所述, 李级数算法的递推公式为

$$\begin{cases} x^{k+1} = x^k + hy^k + \left[\sum_{i=3}^m \frac{h^{i-1}}{(i-1)!} L^{i-1} x \right]_{z=z^k}, \\ y^{k+1} = y^k + h(B + Ax^k) + \left[\sum_{i=3}^m \frac{h^{i-1}}{(i-1)!} L^{i-1} y \right]_{z=z^k}, \end{cases} \quad (12)$$

或

$$\begin{bmatrix} x \\ y \end{bmatrix}^{k+1} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}^k + \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}^{k+1}. \quad (13)$$

其中

$$\begin{aligned} J_{11} &= I + \sum_{i=2}^{m/2} \left[\frac{h^{2(i-1)}}{(2i-2)!} A^{i-1} \right], & J_{12} &= \sum_{i=1}^{m/2} \left[\frac{h^{2i-1}}{(2i-1)!} A^{i-1} \right], \\ J_{21} &= \sum_{i=1}^{m/2} \left[\frac{h^{2i-1}}{(2i-1)!} A^i \right], & J_{22} &= \sum_{i=1}^{m/2} \left[\frac{h^{2(i-1)}}{(2i-2)!} A^{i-1} \right], \\ q_1 &= \sum_{i=2}^{m/2} \left[\frac{h^{2(i-1)}}{(2i-2)!} A^{i-2} B \right], & q_2 &= \sum_{i=1}^{m/2} \left[\frac{h^{2i-1}}{(2i-1)!} A^{i-1} B \right]. \end{aligned}$$

在变质量系统中, 因质量变化, 李级数算法的误差也要随之变化. 由于李级数算法的误差可以准确

估计, 因此根据误差容限, 可以确定能够满足精度要求的 m .

定义如下误差

$$\frac{\|\mathbf{x}_0^{k+1} - \mathbf{x}^{k+1}\|_2}{\|\mathbf{x}_0^{k+1}\|_2} \approx \frac{\|\mathbf{R}^{k+1}\|_2}{\|\mathbf{x}^{k+1} + \mathbf{R}^{k+1}\|_2} \leq e, \quad (14)$$

其中 e 为误差容限, $\mathbf{x}_0^{k+1} \approx \mathbf{x}^{k+1} + \mathbf{R}^{k+1}$ 为待求的精确解, $\mathbf{x}_0^{k+1}$ 由(9)式计算, 而 $\mathbf{x}^{k+1}$ 和 $\mathbf{R}^{k+1}$ 分别由(12)和(11)式计算. 根据(13)式可以确定算法的阶次 m . 值得指出的是, 在使用李级数算法时, m 将由误差 e 和时间步长 h 来确定. 一个更严格的误差为

$$\text{abs}(\mathbf{R}^{k+1}) \leq \text{abs}(e\mathbf{x}^{k+1} + e\mathbf{R}^{k+1}). \quad (15)$$

(14)式相当于用向量长度定义的总体误差准则, (15)式相当于用向量每个元素定义的局部误差准则.

对于单自由度系统, 令 $F=0$, 则 $B=0$, 且 $A = -k/m = -\omega_0^2$, 这些参数对应方程 $\ddot{x} + \omega_0^2 x = 0$. 此时, (13)式变成

$$\mathbf{z}^{k+1} = \mathbf{J}\mathbf{z}^k, \quad (16)$$

其中

$$\mathbf{J} = \begin{bmatrix} J_{11} & \frac{1}{\omega} J_{12} \\ -\omega J_{12} & J_{11} \end{bmatrix}, \quad (17)$$

而

$$J_{11} = \sum_{i=1}^{m/2} \left[\frac{\tau^{2(i-1)} (-1)^{i-1}}{(2i-2)!} \right], \quad J_{12} = \sum_{i=1}^{m/2} \left[\frac{\tau^{2i-1} (-1)^{i-1}}{(2i-1)!} \right], \quad (18)$$

式中, $\tau = \omega h$. 雅可比矩阵 J 的特征值为

$$\lambda_{1,2} = J_{11} \pm i J_{12} = r(\cos \theta \pm i \sin \theta), \quad (19)$$

式中幅值和相位分别为

$$r = \sqrt{J_{11}^2 + J_{12}^2}, \quad (20)$$

$$\theta = \arccos \frac{J_{11}}{r}, \quad (21)$$

单步相位误差为

$$\phi = \tau - \theta = \tau - \arccos \frac{J_{11}}{r}. \quad (22)$$

上式有意义的条件是 $0 < J_{11}/r < 1$, 也就是要求 $\tau < \pi/2$. 值得强调的是, J_{11} 和 J_{12} 分别为 $\cos \tau$ 和 $\sin \tau$ 级数展开的前 $m/2$ 项; 对任何 τ , 只要 m 足够大, 结果都是收敛的. 或者说, 对单自由度系统, 当 m 趋于无穷大时, $J_{11} = \cos \tau$ 和 $J_{12} = \sin \tau$, 此时结果为解析解.

3 时变系统的模态叠加法

为了用模态叠加方法计算时变系统的动态响应, 需要将时间坐标分成若干小时间区间或单元. 每个时间单元的结束状态即为下一个相邻时间单元的初始状态. 若存在外部激励, 则选择时间单元中点或结束时刻的激励值并作为常数代入 Duhamel 积分中进行计算, 即可得到单元内动态响应. 这种模态叠加方法称为递推模态叠加方法. 位移和速度响应为

$$\mathbf{x}^{k+1} = \sum_{j=1}^n \boldsymbol{\phi}_j^{k+1} \left[a \cos \omega_j^{k+1} h + b \sin \omega_j^{k+1} h + c(1 - \cos \omega_j^{k+1} h) \right], \quad (23)$$

$$\dot{\mathbf{x}}^{k+1} = \sum_{j=1}^n \boldsymbol{\phi}_j^{k+1} \omega_j^{k+1} (b \cos \omega_j^{k+1} h - a \sin \omega_j^{k+1} h + c \sin \omega_j^{k+1} h), \quad (24)$$

式中

$$a = \frac{(\boldsymbol{\phi}_j^{k+1})^T \mathbf{M}^{k+1} \mathbf{x}^k}{M_{pj}^{k+1}}, \quad b = \frac{(\boldsymbol{\phi}_j^{k+1})^T \mathbf{M}^{k+1} \dot{\mathbf{x}}^k}{\omega_j^{k+1} M_{pj}^{k+1}},$$

$$c = \frac{P_j^{k+1}}{M_{pj}^{k+1} (\omega_j^{k+1})^2}.$$

4 数值分析

考虑一个均匀简支梁, 其力学参数为: 初始单位长度质量 $\rho A_0 = 1000$, 弯矩刚度 $EI = 10^5$, 梁长 $L = 100$. 将梁等分为 12 个三次单元. 初始条件为零, 简谐激励 $f = \sin(8t)$ 作用在梁的中点. 不用考虑质量减少产生的推力. 单位长度质量衰减规律为 $\rho A = (1 - 0.9t/200) \rho A_0$, 时间步长为 0.05.

图 1 仅给出了用递推模态叠加法得到的梁中点位移曲线, 这是因为难以从图像上区分模态叠加法和李级数法结果的差别(图 2). 从图 1 可以看出, 随着质量减少, 系统各阶固有频率升高, 而激励频率位于基频和最高阶固有频率之间, 低阶频率在响应中贡

图 1 简支梁中点位移($e=10^{-3}$) 图 2 局部放大的简支梁中点位移($e=10^{-3}$)

献逐渐增大, 因此位移响应幅值逐渐增大.

图 3 给出了李级数阶次随着精度要求的增加而逐渐增加的情况; 图 4 和 5 分别给出了幅值和相位差的相应变化情况. 不难看出, 随着质量减少, 若步长不变, 为了保证精度, 李级数算法阶次需要增加, 且算法阶次增加一次, 幅值和相位误差就相应大幅度减少.

5 总结

针对多自由度系统, 本文推导了李级数直接积分算法的递推格式, 该算法适用于时变和时不变系统. 李级数算法具有如下特点.

1) 李级数方法把对时间导数的计算问题转化为对状态变量的求导问题, 因此随着算法阶次升高, 不存在高阶时间导数的计算困难和误差大等问题.

图 3 (网络版彩图)李级数算法阶次

图 4 (网络版彩图)幅值误差

图 5 (网络版彩图)单步相位误差

2) 李级数算法的截断误差可以准确计算, 据此可以准确判断满足精度要求的算法阶次.

3) 李级数算法虽然求解状态方程, 但在实际应

用过程中, 并不需要矩阵扩维.

4) 李级数算法不需求刚度矩阵逆矩阵, 仅需要

求质量矩阵逆矩阵, 这与中心差分方法类似, 具有高效率特征.

参考文献

- 1 邢誉峰, 谢珂, 潘忠文. 变质量系统振动分析的两种方法. 北京航空航天大学学报, 2013, 39: 858–862
- 2 邢誉峰, 郭静. 与结构动特性协同的自适应 Newmark 方法. 力学学报, 2012, 44: 904–911
- 3 Steinberg S. Lie series and nonlinear ordinary differential equations. J Math Anal Appl, 1984, 101: 39–63
- 4 张素英, 邓子辰. 非线性动力学方程的李级数解法及应用. 动力学与控制学报, 2004, 2: 13–20
- 5 邢誉峰, 冯伟. 李级数算法与 Runge-Kutta 法. 振动工程学报, 2007, 20: 519–522
- 6 邢誉峰, 冯伟. 李级数算法和显式辛算法的相位分析. 计算力学学报, 2009, 26: 167–171
- 7 邢誉峰, 潘忠文, 杨阳. 蒙皮加筋圆柱壳弯曲频率的三种计算模型. 北京航空航天大学学报, 2012, 38: 438–443
- 8 潘忠文, 邢誉峰, 杨阳. 蒙皮加筋圆柱壳扭转频率的三种计算模型. 北京航空航天大学学报, 2011, 37: 1156–1159

Self-adaptive Lie series method for mass-variable systems

XING YuFeng¹, WANG YuNan¹, PAN ZhongWen² & DONG Kai²

¹ Institute of Solid Mechanics, Beihang University, Beijing 100191, China;

² Beijing Institute of Astronautical Systems Engineering, Beijing 100076, China

Launch vehicles are the reprehensive of mass-variable systems, whose frequencies increase quickly with the decrease of propellant. Self-adaptive Lie series method is proposed for solving the linear ordinary governing equations with time-varying coefficients, the recursive scheme in matrix form is also given. If keeping time step size constant, the order of the proposed method changes automatically with the requirements of accuracy, and if keeping order constant, the time step size can also change automatically.

mass-variable system, Lie series method, direct integration method, self-adaptive

doi: 10.1360/N092014-00033