

可分型指数矩阵的快速精细积分法

徐建新 郭巧荣 卿光辉

(中国民航大学航空工程学院, 天津 300300)

摘要 针对可分型矩阵的特性, 结合 2^N 类算法为可分型指数矩阵的计算提出一种快速精细积分法. 核心思想是: 利用可分型矩阵中的子矩阵进行分块计算; 增加 Taylor 展开式的保留项数, 减少迭代次数. 一方面, 程序实现简便, 另一方面, 数值算例表明: 对矩阵维数很大的可分型指数矩阵计算来说, 本文的快速精细积分法减少了计算量和存储量, 大大地提高了计算效率.

关键词 可分型指数矩阵, 2^N 类算法, 快速精细积分法, 子矩阵

引言

指数矩阵计算与许多理论和工程科学有关, 如控制论中的状态方程求解, 力学中动力学问题等, 但是指数矩阵计算一直是数学中公认的难题. Moler and Van Loan^[1] 提到了十九种计算方法, 并就多种方法的稳定性、适应性和计算精度等问题做了较详细的分析和讨论. 相对来说, Pade 方法被工程界广泛接受和运用, 例如著名软件 Matlab[®] 和 Mathematica[®] 中的指数矩阵计算均以 Pade 方法为核心, 该方法最明显的优点是速度快, 但涉及到矩阵求逆运算时, 一方面存在数值不稳定问题, 另一方面 Pade 方法不能处理逆矩阵不存在的问题, 因此限制了该方法的适应范围. 钟万勰教授在上个世纪 90 年代初提出了指数矩阵计算的精细积分法^[2]. 精细积分法中独特的构思和巧妙的算法设计使得计算结果精度非常高^[3], 该方法具有相容性、收敛性、稳定性、零振幅衰减率、零周期延长率及无超越性等优良特性^[4], 同时算法十分简单, 无矩阵求逆问题, 已经广泛应用在结构动力分析、优化控制、偏微分方程的精细求解、非稳态随机动力学等许多领域^[5]. 然而, 矩阵尺度很大时, 计算量很大, 并给数据存储量带来较大困难^[4].

可分型矩阵是比较常见的, 文献[6]指出: 弹性力学的哈密顿正则方程经过转换可以化为可分型的哈密顿正则方程, 即该微分方程的系数矩阵具有可分的特征, 可分型哈密顿系统在数值计算上有许多

优良特性, 应用也很广泛. 本文结合指数矩阵的定义—Taylor 展开式及精细积分法中的 2^N 类算法, 为可分型矩阵提出了一种快速的精细积分算法.

1 精细积分法的基本理论及计算过程

精细积分法主要是将注意力放在增量上, 而不是全量. 假设 A 为 $n \times n$ 阶矩阵, 为了避免舍入误差, 利用指数函数的加法定理:

$$T' = \exp(A\eta) = [\exp(A\eta/m)]^m \quad (1)$$

其中 m 为任意正整数, $m = 2^N$ (例如 $N = 20$, 则 $m = 1048576$).

由于 η 是一个小的时间区段, 则 $\tau = \eta/m$ 是一个非常小的时间区段. 因此对于 τ , 有

$$T = \exp(A\tau) = I_n + A\tau + \frac{1}{2!}A^2\tau^2 + \frac{1}{3!}A^3\tau^3 + \frac{1}{4!}A^4\tau^4 + \dots \quad (2)$$

文献[7]指出指数截断阶数 $L = 4$, 迭代次数 $N \geq 15$ 时精细积分法中指数矩阵计算可以认为接近其精确解答. 此时指数矩阵 T 与单位矩阵 I_n 相差不远, 因此可写为

$$\exp(A\tau) \approx (T_a + I_n) \\ T_a = A\tau + (A\tau)^2 \left[I_n + \frac{(A\tau)}{3} + \frac{(A\tau)^2}{12} \right] / 2 \quad (3)$$

其中 T_a 阵是一个小量矩阵.

因为 T_a 很小, 当它与单位阵 I_n 相加时, 就会成为尾数, 在计算机的舍入操作中会被舍去, 影响计算精度^[8]. 所以, 至关重要的一点是指数矩阵的存

储是式(3)中的 T_a , 而不是 $(T_a + I_n)$.

为了计算 $\exp(A\eta)$, 有

$$\exp(A\eta) \approx (I_n + T_a)^{2^{(N-1)}} \times (I_n + T_a)^{2^{(N-1)}} \quad (4)$$

这种分解一直做下去, 共 N 次. 其次应注意, 对任意矩阵 T_a, T_c 有

$$(I_n + T_b) \times (I_n + T_c) = I_n + T_b + T_c + T_b \times T_c \quad (5a)$$

当 T_b, T_c 很小时, 不应加上 I_n 后再执行乘法.

将 T_b, T_c 都看成为 T_a , 因此式(4)的 N 次乘法在计算机中相当于以下的语句:

$$\text{for } (iter = 0; iter < N; iter + +) \quad T_a = 2T_a + T_a \times T_a \quad (5b)$$

当该语句循环结束后, 再执行

$$\exp(A\eta) \approx I_n + T_a \quad (6)$$

执行 N 次乘法后, T_a 已不再是很小的矩阵了, 这个加法已没有严重的舍入误差.

很明显, 关于 T_a 迭代计算涉及到矩阵乘法运算, 当 N 值较大时, 对计算效率有很大影响.

2 可分型指数矩阵的快速精细积分法的基本理论及过程

在精细积分法中, 若 A 阵是满阵, 当 A 阵的维数非常大时, 矩阵相乘计算量及存储量随维数迅速增长, 当迭代次数 N 值较大时, 对计算效率有很大的影响. 因此文献[4]考虑用子矩阵的办法, 将指数矩阵计算转化为各子矩阵的运算.

文献[9]指出, 精细积分法中 N 值的选取与 Taylor 展开式项数的选取上值得商榷, 因为这些参数选取与算法的效率和精度有直接的关系. 文献[10]提出适当增加保留项数比单纯增加迭代次数的效果更好, 更有利于精度的提高和逼近过程的加速实现. 在这一指导思想下, 下面结合可分型矩阵的特性和精细积分法的核心思想给出关于可分型矩阵的快速精细积分法.

选择取 m 的原则为: 假设 $m = 2^N$, 在算法实现上取 N 使

$$\|A\tau\| = \frac{\|A\eta\|}{2^N} \leq \frac{1}{2} \quad (7)$$

现设可分矩阵为

$$A = \begin{bmatrix} 0 & C \\ D & 0 \end{bmatrix} \quad (8)$$

其中矩阵 C 和 D 阶数是 A 阵阶数的一半, 即 $\frac{n}{2}$ 阶,

这里的 n 为偶数.

Liou^[11]指出若 δ 为事先规定的容许误差, 则截断项数 q 由下式决定.

$$\|T_q(A\tau) - e^{A\tau}\| \leq \left(\frac{\|A\tau\|^{q+1}}{(q+1)!} \right) \times \left(\frac{1}{1 - \|A\tau\|/(q+2)} \right) \leq \delta \quad (9)$$

上式可化简为

$$\frac{\|A\tau\|^{q+1}}{(q+1)!} \leq \delta \quad (10)$$

下面以截断项数 q 是偶数为例说明本文可分型指数矩阵快速精细积分法的计算技巧:

$$\exp(A\tau) \approx I_n + A\tau + \frac{(A\tau)^2}{2!} + \frac{(A\tau)^3}{3!} + \frac{(A\tau)^4}{4!} + \dots + \frac{(A\tau)^{q-1}}{(q-1)!} + \frac{(A\tau)^q}{q!} \quad (11)$$

设 $A' = A\tau$, 则可设 $A' = \begin{bmatrix} 0 & C' \\ D' & 0 \end{bmatrix} = \begin{bmatrix} 0 & C' \\ D' & 0 \end{bmatrix}$,

$$\begin{aligned} \exp(A') \approx & I_n + A' + \frac{(A')^2}{2!} + \frac{(A')^3}{3!} + \frac{(A')^4}{4!} + \dots \\ & \dots + \frac{(A')^{q-1}}{(q-1)!} + \frac{(A')^q}{q!} = I_n + \begin{bmatrix} 0 & C' \\ D' & 0 \end{bmatrix} + \\ & \frac{1}{2!} \begin{bmatrix} C'D' & 0 \\ 0 & D'C' \end{bmatrix} + \frac{1}{3!} \begin{bmatrix} 0 & C'D'C' \\ D'C'D' & 0 \end{bmatrix} + \\ & \frac{1}{4!} \begin{bmatrix} (C'D')^2 & 0 \\ 0 & (D'C')^2 \end{bmatrix} + \dots + \\ & \frac{1}{(q-1)!} \begin{bmatrix} 0 & (C'D')^{\frac{q}{2}-1} C' \\ (D'C')^{\frac{q}{2}-1} D' & 0 \end{bmatrix} + \\ & \frac{1}{q!} \begin{bmatrix} (C'D')^{\frac{q}{2}} & 0 \\ 0 & (D'C')^{\frac{q}{2}} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} + \\ & \begin{bmatrix} T_1 & T_2 \\ T_3 & T_4 \end{bmatrix}, q \in \{2, 4, 6, \dots, 2k\}, k \geq 1 \quad (12) \end{aligned}$$

其中:

$$\begin{aligned} T_1 &= C'D'\alpha \\ \alpha &= \frac{1}{2!}I + \frac{1}{4!}C'D' + \frac{1}{6!}(C'D')^2 + \dots + \\ & \frac{1}{(q-2)!}(C'D')^{\frac{q}{2}-2} + \frac{1}{q!}(C'D')^{\frac{q}{2}-1} \\ T_2 &= \beta C' \\ \beta &= I + \frac{1}{3!}C'D' + \frac{1}{5!}(C'D')^2 + \dots + \\ & \frac{1}{(q-3)!}(C'D')^{\frac{q}{2}-2} + \frac{1}{(q-1)!}(C'D')^{\frac{q}{2}-1} \end{aligned}$$

$$T_3 = D'\beta$$

$$T_4 = D'\alpha C' \tag{13}$$

式(12)采用了精细积分法中的表达形式,单位阵 I_n 在累加与乘法过程中不进行计算,因而使展开式的计算舍入误差减小.另外,计算式(13)中的 α 和 β 时,我们只需计算和存储一次 $C'D'$, $(C'D')^2, \dots, (C'D')^{\frac{n}{2}-2}, (C'D')^{\frac{n}{2}-1}$, 并采用由高阶量(小值)向低阶量(大值)反向累加的逆向算法,把矩阵的幂从高次幂开始依次与相对应的系数相乘得到的结果累加,以此来控制计算机加、乘法运算造成的舍入误差.

3 实例分析

一般说来,在实际应用过程中,选取容许误差 $\delta = 10^{-15}$, 要求较高.如取 $\delta = 10^{-8}$, 则可使计算结果精度保持至少6位^[9].现在取 $\delta = 10^{-k}$ ($k = 10, 11, \dots, 15$), 时间步长 $\eta = 0.01s$, 可分型矩阵中的非零元素 $e \in [-1, 1]$.图1~图6给出了精细积分法、Pade方法和本文方法的计算时间对比.

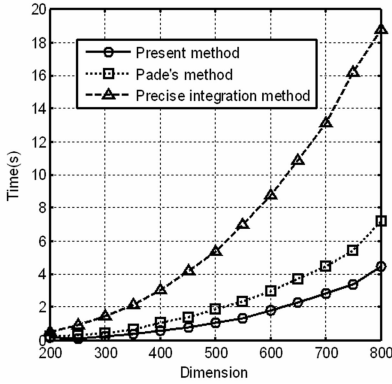


图1 $\delta = 10^{-10}$, 三种方法比较

Fig. 1 $\delta = 10^{-10}$ Comparison of the three methods

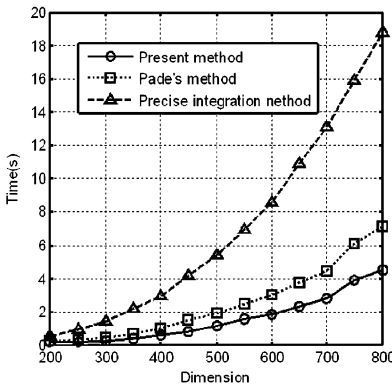


图2 $\delta = 10^{-11}$, 三种方法比较

Fig. 2 $\delta = 10^{-11}$, Comparison of the three methods

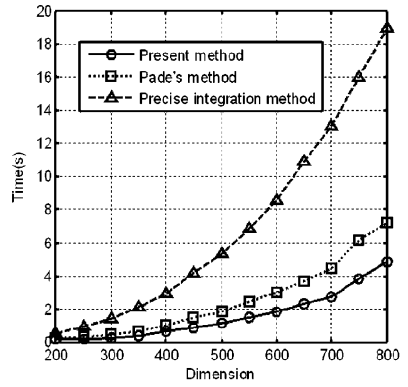


图3 $\delta = 10^{-12}$, 三种方法比较

Fig. 3 $\delta = 10^{-12}$, Comparison of the three methods

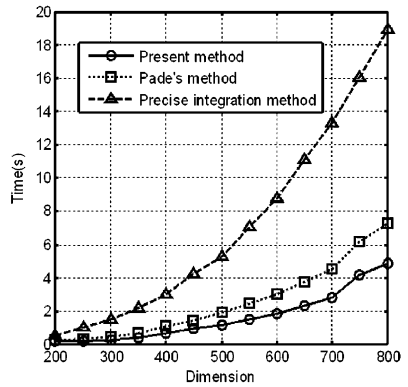


图4 $\delta = 10^{-13}$, 三种方法比较

Fig. 4 $\delta = 10^{-13}$, Comparison of the three methods

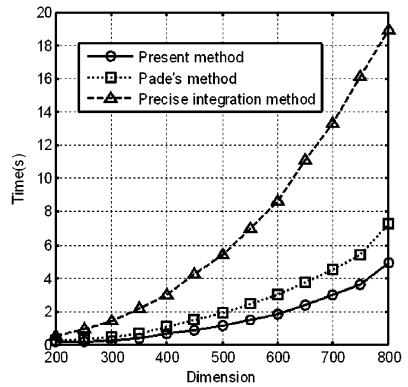


图5 $\delta = 10^{-14}$, 三种方法比较

Fig. 5 $\delta = 10^{-14}$, Comparison of the three methods

从上述图形可以看出给定精确度和确定的时间步长情况下,本文方法的计算速度比较快,尤其是计算矩阵维数较大时(如 Hamilton 正则方程的有限元半解析法中需要计算高维的可分型指数矩阵),效果非常显著.以图6为例说明,在误差精度 $\delta = 10^{-15}$ 的情况下,计算800维的随机指数矩阵,本文方法、Pade方法和精细积分法所用的平均时间

分别为 4.8626s, 7.2282s, 18.9346s.

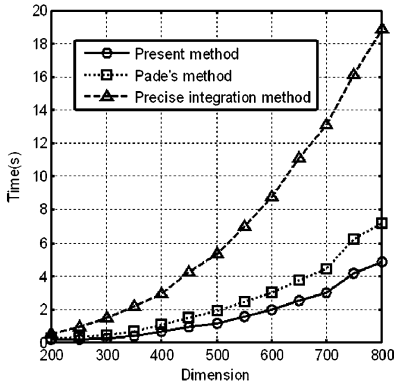


图 6 $\delta = 10^{-15}$, 三种方法比较

Fig. 6 $\delta = 10^{-15}$, Comparison of the three methods

文献[12]指出精细积分法是条件稳定的,综合考虑稳定性、精度和计算工作量,判断指数截断阶数 $L=4$ 时,精细积分法的总体效果最好,另外迭代次数 $N \geq 15$ 时精细积分法中指数矩阵计算可认为接近其精确解答^[7],故在精细积分法中的代码中选取了 $L=4, N=16$,计算精度完全满足误差精度要求.实际上,精细积分法在矩阵和积分步长满足一定条件时,其计算结果可以达到计算机所能表示的满精度.本文方法和 Pade 方法采用同样的原理选取截断项数和迭代次数,可以说两种方法的计算精度是相同的.虽然 Pade 方法、本文的方法和精细积分法($L=4, N=16$)的计算精度均在同一数量级内,但是本文方法和 Pade 方法的计算精度与精细积分法的计算精度相比有一定差距.在一般工程计算中,计算精度往往只需达到一定要求,计算效率是主要的,那么对可分型指数矩阵计算采用本文的快速精细积分法比较经济.

4 计算量分析

不论是可分型矩阵还是其它形式的方阵,迭代算法会使式(5)中的非零元素快速增加,使得矩阵乘法的计算量增大.若不考虑加法所用时间,则维数为 $2n$ 的矩阵相乘,其计算量为 $O(8 \times n^3)$.若可分型矩阵的维数为 $2n$,精细积分法中选取 $L=4, N=16$,则计算量约为 $O(152 \times n^3)$.通常情况下可分型指数矩阵截断项数 q 不会超过 20 项,所以当计算式(13)中的 α 和 β 时,依次计算矩阵的幂至矩阵的最高次幂 $(C'D')^{\frac{q}{2}-1}$ 最多需要 9 次 n 维的矩阵相乘.根据式(13),计算 T_n 阵时,计算量最多约为

$O(14 \times n^3)$,经过 N 次循环迭代后,则可分型指数矩阵的计算量约为 $O((14 + 8N) \times n^3)$,上述实例表明,迭代次数 N 最多不会超过 10 次,故本文方法计算量最多为 $O(94 \times n^3)$,约为传统精细积分法计算量的 61.8%,可见计算量的降低是十分显著的.Pade 方法和本文的快速精细积分法是采用同样原理选取截断项数和迭代次数,但是求逆计算增加了计算量,所以说 Pade 方法在计算速度上比本文的方法慢.

实际上,在相同的计算精度要求下,本文方法和传统精细积分法取相同迭代步数 N 和截断阶数 L (假设 $L=4$),传统精细积分法的计算量约为 $O((24 + 8N) \times n^3)$,而本文方法的计算量约为 $O((6 + 8N) \times n^3)$,相比较而言,本文方法还是比传统精细积分法的计算量小很多.所以说,当计算维数很大的可分型指数矩阵时,这种方法很好的解决了这一难题.

5 结论

本文在精细积分法基础上,结合可分型矩阵的特性和精细积分法的核心思想,为可分型指数矩阵计算提出一种快速算法,程序实现简便.在给定的精度和确定的时间步长情况下,验证了快速精细积分法在处理大型问题时可以显著节约时间.

本文的方法仍然具有精细积分法的原有优点,不需要矩阵求逆计算,在处理系统矩阵 A 奇异或接近奇异时的优越性是十分明显的.

参 考 文 献

- 1 Moler C, Loan C V. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Rev*, 1978, 20(4): 801 ~ 836
- 2 钟万勰. 结构动力方程的精细时程积分. 大连理工大学学报, 1994, 34(2): 131 ~ 136 (Zhong W X. On precise time-integration method for structural dynamic equation. *Journal of Dalian University of Technology*, 1994, 34(2): 131 ~ 136(in Chinese))
- 3 顾元宪,陈颺松. 瞬态热传导方程精细积分方法中对称性的利用. 力学与实践, 2000, 22(5): 19 ~ 22 (Gu Y X, Chen B S. The symmetry of precise time integration for transient heat conduction. *Mechanics in Engineering*, 2000, 22(5): 19 ~ 22(in Chinese))

- 4 陈飏松,顾元宪. 瞬态热传导方程的子结构精细积分方法. 应用力学学报,2001,18(1): 14~19 (Chen B S, Gu Y X. Substructures precise time integration method for transient heat conduction. *Chinese Journal of Applied Mechanics*, 2001,18(1): 14~19(in Chinese))
- 5 向宇,黄玉盈,曾革委. 精细时程积分法的误差分析与精度设计. 计算力学学报,2002,19(3): 276~280 (Xiang Y, Huang Y Y, Zeng G W. Error analysis and accuracy design for the precise time-integration method. *Chinese Journal of Computational Mechanics*, 2002,19(3): 276~280(in Chinese))
- 6 郑宇. 变分法、无穷维 Hamilton 系统及其在弹性力学中的应用. 大连:大连理工大学,1994 (Zheng Y. Variational method and infinite dimensional Hamiltonian systems with its applications in elasticity. Dalian: Dalian University of Technology, 1994)
- 7 汪梦甫,区达光. 精细积分方法的评估与改进. 计算力学学报,2004,21(6): 728~733 (Wang M P, Qu D G. Assessment and improvement of precise time step integration method. *Chinese Journal of Computational Mechanics*, 2004,21(6): 728~733(in Chinese))
- 8 侯秀慧,邓子辰,黄立新. 桥梁结构移动荷载识别的辛精细积分算法. 动力学与控制学报,2008,6(1): 66~72 (Hou X H, Deng Z C, Huang L X. An improved symplectic precise integration method for moving load identification of bridge structure. *Journal of Dynamics and Control*, 2008,6(1): 66~72(in Chinese))
- 9 张洪武,钟万勰. 矩阵指数计算算法讨论. 大连理工大学学报,2000,40(5): 522~525 (Zhang H W, Zhong W X. Discussion about numerical computation of the matrix exponential. *Journal of Dalian University of Technology*, 2000,40(5): 522~525(in Chinese))
- 10 董聪,丁李粹. 动力学系统精细算法的逼近机理与误差上界. 计算力学学报,1999,16(3): 253~260 (Dong C, Ding L C. Approximate mechanism and upper error limit of precise computation for dynamics system. *Chinese Journal of Computational Mechanics*, 1999,16(3): 253~260(in Chinese))
- 11 M L Liou. A Novel Method of Evaluating Transit Response. Proc, IEEE, 1966, 54, 20~23
- 12 赵丽滨,张建宇,王寿梅. 精细积分方法的稳定性和精度分析. 北京航空航天大学学报,2000,26(5):569~572 (Zhao L B, Zhang J Y, Wang S H. Stability and precision analysis for precise integration method. *Journal of Beijing University of Aeronautics and Astronautics*, 2000,26(5): 569~572(in Chinese))

A FAST PRECISE INTEGRATION METHOD FOR THE SEPARABLE EXPONENTIAL MATRIX

Xu Jianxin Guo Qiaorong Qing Guanghui

(College of Aeronautical Engineering, Civil Aviation University of China, Tianjin 300300, China)

Abstract Based on the characteristics of the separable exponential matrix and the 2^N -type algorithm, a fast precise integration method for the separable exponential matrix was proposed. The key idea is that the block-matrix is computed based on the submatrix of the separable matrix, thus increasing the reserved item numbers in the Taylor series expansion and decreasing iterative times. On the one hand, the programming is easy. On the other hand, the numerical examples show that the method presented for the large dimension separable exponential matrix can reduce the computation and storage amount and greatly improve the calculation efficiency.

Key words separable exponential matrix, 2^N -type algorithm, fast precise integration method, submatrix