

# 计算张量指数函数的广义逆张量 $\varepsilon$ -算法

顾传青<sup>1</sup> 唐鹏飞<sup>1</sup> 陈之兵<sup>2</sup>

**摘要** 张量指数函数已经广泛应用于工程领域. 本文得到了一种有效的张量广义逆, 并以此为基础构造了广义逆张量 Padé 逼近的一种  $\varepsilon$ -算法. 该算法可以编程实施递推的计算, 其特点是, 在计算过程中, 不必计算张量的乘积, 也不必计算张量的逆. 给出的计算张量指数函数的数值实验显示, 将本文的方法与目前通常使用的截断法进行比较, 在不降低逼近阶的条件下,  $\varepsilon$ -算法能很好地降低计算复杂度, 尤其是在张量的维数比较大的时候.

**关键词** 张量指数函数, 广义逆张量 Padé 逼近, 张量  $\varepsilon$ -算法, 张量指数函数的截断法

**引用格式** 顾传青, 唐鹏飞, 陈之兵. 计算张量指数函数的广义逆张量  $\varepsilon$ -算法. 自动化学报, 2020, 46(4): 744–751

**DOI** 10.16383/j.aas.c180002

## Generalized Inverse Tensor $\varepsilon$ -algorithm for Computing Tensor Exponential Function

GU Chuan-Qing<sup>1</sup> TANG Peng-Fei<sup>1</sup> CHEN Zhi-Bing<sup>2</sup>

**Abstract** Tensor exponent functions have been widely used in engineering. In this paper, an effective tensor generalized inverse is obtained, and based on it, an  $\varepsilon$ -algorithm of generalized inverse tensor Padé approximation is presented. The algorithm can be programmed to implement recursive calculations, which are characterized by the fact that it is not necessary to calculate the product of the tensors in the calculation nor to calculate the inverse of tensors. The numerical experiment of calculating tensor exponential function shows that compared with the commonly used truncation method, the  $\varepsilon$ -algorithm can reduce computational complexity well without reducing the approximation order, especially when the dimension of tensor is relatively large.

**Key words** Tensor exponential functions, generalized inverse tensor Padé approximation, tensor  $\varepsilon$ -algorithm, truncation method

**Citation** Gu Chuan-Qing, Tang Peng-Fei, Chen Zhi-Bing. Generalized inverse tensor  $\varepsilon$ -algorithm for computing tensor exponential function. *Acta Automatica Sinica*, 2020, 46(4): 744–751

近年来, 张量方法在控制理论及其应用的各个领域得到了广泛的应用, 如基于张量数据表示的深度学习模型<sup>[1]</sup>、基于非负张量分析的时序链路预测方法<sup>[2]</sup>、局部图像描述中基于结构张量的 HDO (Histograms of dominant orientations) 算法<sup>[3]</sup>、二维解析张量投票算法<sup>[4]</sup> 等.

张量动力系统已经广泛应用于 Volterra 系统识别<sup>[5]</sup>、张量乘积 TP (Tensor product) 模型变换<sup>[6]</sup>、人体动作识别<sup>[7–8]</sup> 和塑性模型<sup>[9]</sup> 等各个领域. 因此, 由于在张量常微分方程解中的关键作用, 张量指数

函数的计算已经成为一个重要的研究领域.

考虑如下常微分方程<sup>[9]</sup>

$$\begin{cases} \dot{\mathcal{Y}}(t) = \mathcal{A}\mathcal{Y}(t) \\ \mathcal{Y}(t_0) = \mathcal{Y}_0 \end{cases} \quad (1)$$

这里  $\mathcal{A}$  和  $\mathcal{Y}_0$  是给定张量, 一般是非对称的, 那么关于系统 (1) 的张量指数函数  $\exp(\cdot)$  有如下唯一解:

$$\mathcal{Y}(t) = \exp[(t - t_0)\mathcal{A}]\mathcal{Y}_0 \quad (2)$$

对于一般的张量  $\mathcal{A}$ , 它的指数函数可以表示为它的级数形式:

$$\exp(\mathcal{A}) = \sum_{n=0}^{\infty} \frac{1}{n!} \mathcal{A}^n$$

目前计算张量指数函数 (2) 通常使用的方法是截断法<sup>[10]</sup>, 其实质就是将张量指数函数展开成无穷级数后, 取前  $n_{\max}$  项, 得到近似解, 即:

$$\exp(\mathcal{A}) \approx \sum_{n=0}^{n_{\max}} \frac{1}{n!} \mathcal{A}^n$$

收稿日期 2018-01-03 录用日期 2018-04-06  
Manuscript received January 3, 2018; accepted April 6, 2018  
国家自然科学基金 (11371243), 上海市重点学科建设资助项目 (S30104) 资助  
Supported by National Natural Science Foundation of China (11371243), Key Disciplines of Shanghai Municipality (S30104)  
本文责任编辑 黎铭  
Recommended by Associate Editor LI Ming  
1. 上海大学数学系 上海 200444 2. 深圳大学数学与统计学院 深圳 518052  
1. Department of Mathematics, Shanghai University, Shanghai 200444 2. School of Mathematics and Statistics, Shenzhen University, Shenzhen 518052

张量指数函数选取的项数  $n_{\max}$  受限于所需的精度:

$$\frac{1}{n_{\max}!} \|\mathcal{A}^{n_{\max}}\| \leq \epsilon_{\text{tol}}$$

显然有截断法的精度与张量指数函数选取的项数  $n_{\max}$  有关, 保留的项数越多, 精度越高, 但是需要进行的张量乘积次数也就越多; 保留的项数越少, 计算量越少, 但是精度也就越低. 因此, 计算张量指数函数的截断法有待改进.

为研究上述问题, 本文首次定义了张量的一种广义逆, 并以此为基础构造了张量广义逆 Padé 逼近 GITPA (Generalized inverse tensor Padé approximation) 的一种  $\varepsilon$ -算法. GITPA 方法的优势在于: 在计算过程中, 不必用到张量的乘积, 也不要计算张量的逆, 另外, 该方法对奇异张量也是适用的. 目前在国内外, 关于计算张量的逆还没有找到一种比较可行的计算方法. 作为 GITPA 方法的一个重要应用, 本文在后面给出计算张量指数函数的数值实验, 来说明  $\varepsilon$ -算法的有效性.

本文组织如下: 第 1 节简单介绍本文用到的张量基础知识, 并定义张量的一种广义逆; 第 2 节, 首先给出广义逆张量 Padé 逼近的定义, 并以此为基础给出张量  $\varepsilon$ -算法; 第 3 节将  $\varepsilon$ -算法用来计算张量指数函数值, 并与通常使用的的级数截断法相比较; 最后给出简单的小结.

## 1 一种张量广义逆

### 1.1 张量基础知识

本节介绍本文将要用到的张量基础知识. 张量是一种多维数组, 其中向量是一阶张量, 矩阵是二阶张量, 特别地, 一个  $p$  阶张量拥有  $p$  个下标, 是  $p$  个拥有独立坐标系的向量空间的外积 (张量积). 一个  $p$  阶  $n_1 \times n_2 \times \cdots \times n_p$  维张量可以表示为:

$$\mathcal{A} = (a_{i_1 i_2 \cdots i_p}) \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_p}$$

文献 [11] 提出了张量的切片方法. 对一个三阶张量, 可以固定其中任意一个下标, 从而得到该张量的一种表示形式. 设  $\mathcal{A} = (a_{i_1 i_2 i_3}) \in \mathbb{C}^{2 \times 2 \times 3}$ , 固定其第三个下标, 则张量可以表示为

$$\mathcal{A} = \left[ \begin{array}{c|c|c} \mathcal{A}_1 & \mathcal{A}_2 & \mathcal{A}_3 \end{array} \right] = \left[ \begin{array}{cc|cc|cc} a_{111} & a_{121} & a_{112} & a_{122} & a_{113} & a_{123} \\ a_{211} & a_{221} & a_{212} & a_{222} & a_{213} & a_{223} \end{array} \right]$$

下面定义两个三阶张量的  $t$ -积, 可以通过递归自然地推广到高阶张量的  $t$ -积.

**定义 1**<sup>[12]</sup>. (块循环矩阵) 令  $\mathcal{A} \in \mathbb{R}^{l \times p \times n}$ , 则  $\mathcal{A}$  的块循环矩阵被定义为

$$\text{bcirc}(\mathcal{A}) = \begin{bmatrix} \mathcal{A}_1 & \mathcal{A}_n & \mathcal{A}_{n-1} & \cdots & \mathcal{A}_2 \\ \mathcal{A}_2 & \mathcal{A}_1 & \mathcal{A}_n & \cdots & \mathcal{A}_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathcal{A}_n & \mathcal{A}_{n-1} & \cdots & \mathcal{A}_2 & \mathcal{A}_1 \end{bmatrix}_{ln \times pn}$$

定义一个展开算子:  $\text{unfold}(\cdot)$ <sup>[12]</sup>, 用以下方式将一个  $p \times m \times n$  的张量展开成一个  $pn \times m$  的矩阵:

$$\text{unfold}(\mathcal{B}) = \left[ \mathcal{B}_1^T \quad \mathcal{B}_2^T \quad \cdots \quad \mathcal{B}_n^T \right]^T$$

这里  $\text{fold}(\cdot)$ <sup>[12]</sup> 是它的逆算子, 它会将一个  $pn \times m$  的矩阵转化成  $p \times m \times n$  的张量. 因此,

$$\text{fold}(\text{unfold}(\mathcal{B})) = \mathcal{B}$$

**定义 2**<sup>[12-13]</sup>. (张量  $t$ -积) 令  $\mathcal{A}$  是一个  $l \times p \times n$  的张量,  $\mathcal{B}$  是一个  $p \times m \times n$  的张量, 则张量  $t$ -积  $\mathcal{A} * \mathcal{B}$  将得到一个  $l \times m \times n$  的张量, 定义如下:

$$\mathcal{A} * \mathcal{B} = \text{fold}(\text{bcirc}(\mathcal{A}) \cdot \text{unfold}(\mathcal{B}))$$

**例 1.** 设  $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{2 \times 2 \times 3}$ , 现固定其第三个下标, 分别产生

$$\mathcal{A} = \left[ \begin{array}{cc|cc|cc} 1 & 2 & 5 & 6 & 9 & 10 \\ 3 & 4 & 7 & 8 & 11 & 12 \end{array} \right]$$

$$\mathcal{B} = \left[ \begin{array}{cc|cc|cc} 1 & 2 & 4 & 3 & 1 & 0 \\ 3 & 4 & 2 & 1 & 0 & 1 \end{array} \right]$$

则由定义 2 得到

$$\mathcal{A} * \mathcal{B} = \text{fold} \left[ \left[ \begin{array}{ccc} \mathcal{A}_1 & \mathcal{A}_3 & \mathcal{A}_2 \\ \mathcal{A}_2 & \mathcal{A}_1 & \mathcal{A}_3 \\ \mathcal{A}_3 & \mathcal{A}_2 & \mathcal{A}_1 \end{array} \right] \cdot \left[ \begin{array}{c} \mathcal{B}_1 \\ \mathcal{B}_2 \\ \mathcal{B}_3 \end{array} \right] \right] = \text{fold} \left[ \begin{array}{c} \mathcal{A}_1 \mathcal{B}_1 + \mathcal{A}_3 \mathcal{B}_2 + \mathcal{A}_2 \mathcal{B}_3 \\ \mathcal{A}_2 \mathcal{B}_1 + \mathcal{A}_1 \mathcal{B}_2 + \mathcal{A}_3 \mathcal{B}_3 \\ \mathcal{A}_3 \mathcal{B}_1 + \mathcal{A}_2 \mathcal{B}_2 + \mathcal{A}_1 \mathcal{B}_3 \end{array} \right] = \left[ \begin{array}{cc|cc|cc} 68 & 53 & 40 & 49 & 72 & 81 \\ 90 & 75 & 62 & 71 & 94 & 103 \end{array} \right]$$

下面定义张量的范数. 令  $\mathcal{A} \in \mathbb{C}^{n_1 \times n_2 \times \cdots \times n_p}$ , 张量的范数就等于它所有元素平方和的平方根<sup>[11]</sup>, 即

$$\|\mathcal{A}\| = \sqrt{\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_p=1}^{n_p} a_{i_1 i_2 \cdots i_p}^2} \quad (3)$$

这和矩阵的 *Frobenius*-模是类似的. 两个大小相同的张量  $\mathcal{A}, \mathcal{B} \in \mathbf{C}^{n_1 \times n_2 \times \cdots \times n_p}$  的内积等于它们对应元素的乘积的和<sup>[11]</sup>, 即

$$(\mathcal{A}, \mathcal{B}) = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_p=1}^{n_p} a_{i_1 i_2 \cdots i_p} b_{i_1 i_2 \cdots i_p} \quad (4)$$

于是显然成立  $(\mathcal{A}, \mathcal{A}^*) = \|\mathcal{A}\|^2$ , 其中记号 “\*” 表示取复共轭.

## 1.2 张量广义逆

参考复数、向量和矩阵的倒数, 容易得到以下结论:

1) 若  $b$  是一个复数,  $b \in \mathbf{C}$ , 则  $bb^* = |b|^2$ ,  $1/b = b^{-1} = b^*/|b|^2$ ;

2) 若  $\mathbf{v}$  是一个向量,  $\mathbf{v} \in \mathbf{C}^n$ , 则  $\mathbf{v} \cdot \mathbf{v}^* = |\mathbf{v}|^2$ ,  $1/\mathbf{v} = \mathbf{v}^{-1} = \mathbf{v}^*/|\mathbf{v}|^2$  (见 Graves-Morris<sup>[14]</sup>);

3) 若  $A = a_{ij}, B = b_{ij} \in \mathbf{C}^{s \times t}$ , 则  $A \cdot B = \sum_{i=1}^s \sum_{j=1}^t (a_{ij} b_{ij}) \in \mathbf{C}$ ,  $1/A = A_r^{-1} = A^*/\|A\|^2$  (见顾传青<sup>[15-17]</sup>). 因此, 下面张量的广义逆可以看作是向量和矩阵的推广:

**定义 3.** 令  $\mathcal{A}, \mathcal{B} \in \mathbf{C}^{n_1 \times n_2 \times \cdots \times n_p}$ , 其内积为

$$(\mathcal{A}, \mathcal{B}) = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_p=1}^{n_p} a_{i_1 i_2 \cdots i_p} b_{i_1 i_2 \cdots i_p}$$

$\mathcal{A} \cdot \mathcal{A}^* = \|\mathcal{A}\|^2$ , 则张量  $\mathcal{A}$  的广义逆被定义为

$$\mathcal{A}_r^{-1} = \frac{1}{\mathcal{A}} = \frac{\mathcal{A}^*}{\|\mathcal{A}\|^2}, \quad \mathcal{A} \neq 0, \quad \mathcal{A} \in \mathbf{C}^{n_1 \times n_2 \times \cdots \times n_p} \quad (5)$$

其中张量的范数由式 (3) 给出.

**引理 1.** 令  $\mathcal{A}, \mathcal{B} \in \mathbf{C}^{n_1 \times n_2 \times \cdots \times n_p}$ ,  $\mathcal{A}, \mathcal{B} \neq 0$  且  $b \in \mathbf{R}, b \neq 0$ , 则下列关系成立:

1)  $\frac{b}{\mathcal{A}} = \frac{1}{\mathcal{B}} \iff \mathcal{A} = b\mathcal{B}$ ;

2)  $(\mathcal{A}_r^{-1})_r^{-1} = \mathcal{A}, (b\mathcal{A})_r^{-1} = \frac{1}{b}\mathcal{A}_r^{-1}$ .

**证明.** 根据定义 3, 结论 2) 是显然的. 现在证明结论 1). 因为  $\mathcal{A}, \mathcal{B} \neq 0$ , 从  $\mathcal{A} = b\mathcal{B}$ , 容易推得  $\frac{b}{\mathcal{A}} = \frac{1}{\mathcal{B}}$ . 利用定义 3, 得到引理 1 的左端,  $\frac{b\mathcal{A}^*}{\|\mathcal{A}\|^2} = \frac{\mathcal{B}^*}{\|\mathcal{B}\|^2}$ , 即

$$\mathcal{A}^* = \frac{\|\mathcal{A}\|^2}{b\|\mathcal{B}\|^2} \mathcal{B}^*, \quad \mathcal{A} = \frac{\|\mathcal{A}\|^2}{b\|\mathcal{B}\|^2} \mathcal{B}$$

因此

$$\|\mathcal{A}\|^2 = \mathcal{A} \cdot \mathcal{A}^* = \frac{\|\mathcal{A}\|^4 \|\mathcal{B}\|^2}{b^2 \|\mathcal{B}\|^4}$$

即  $b^2 = \frac{\|\mathcal{A}\|^2}{\|\mathcal{B}\|^2}$ , 所以  $\mathcal{A} = b\mathcal{B}$ . □

## 2 张量广义逆 $\varepsilon$ -算法

### 2.1 广义逆张量 Padé 逼近的定义

设  $f(t)$  是给定的张量多项式, 其系数为张量, 即

$$f(t) = C_0 + C_1 t + C_2 t^2 + \cdots + C_n t^n + \cdots \quad (6)$$

$$C_i = (c_i^{(i_1 i_2 \cdots i_p)}) \in \mathbf{C}^{n_1 \times n_2 \times \cdots \times n_p}, \quad t \in \mathbf{C}$$

**定义 4.** 令  $\mathbf{C}^{n_1 \times n_2 \times \cdots \times n_p}[t]$  是一个  $p$  阶张量的多项式集合, 维数分别为  $n_1 \times n_2 \times \cdots \times n_p$ . 一个张量多项式  $\mathcal{A}(t) = (a_{i_1 i_2 \cdots i_p}(t)) \in \mathbf{C}^{n_1 \times n_2 \times \cdots \times n_p}[t]$  是  $m$  阶的, 表示为  $\partial \mathcal{A}(t) = m$ , 若  $\partial(a_{i_1 i_2 \cdots i_p}(t)) \leq m$  对于所有  $i_i = 1, 2, \cdots, n_i, 1 \leq i \leq p$  都成立, 且  $\partial(a_{i_1 i_2 \cdots i_p}(t)) = m$  对于某些  $i_i = 1, 2, \cdots, n_i, 1 \leq i \leq p$  成立.

**定义 5.** 定义  $[\frac{n}{2k}]$  型广义逆张量 Padé 逼近 (GITPA) 是一个张量有理函数

$$\mathcal{R}(t) = \frac{\mathcal{P}(t)}{q(t)} \quad (7)$$

其中,  $\mathcal{P}(t)$  是一个张量多项式,  $q(t)$  是一个数量多项式, 满足以下条件:

1)  $\partial \mathcal{P}(t) \leq n, \partial q(t) = 2k$ ;

2)  $q(t) \parallel \|\mathcal{P}(t)\|^2$ ;

3)  $q(t)f(t) - \mathcal{P}(t) = O(t^{n+1})$ ;

其中,  $\mathcal{P}(t) = (p_{(i_1 i_2 \cdots i_p)}(t)) \in \mathbf{C}^{n_1 \times n_2 \times \cdots \times n_p}$ , 其范数  $\|\mathcal{P}(t)\|^2$  由式 (3) 给出, 而整除性条件 2) 表明分母数量多项式  $q(t)$  能够整除分子张量多项式  $\mathcal{P}(t)$  范数的平方.

### 2.2 张量 $\varepsilon$ -算法

给定张量多项式 (6), 根据张量广义逆 (5), 定义张量  $\varepsilon$ -算法如下:

$$\varepsilon_{-1}^{(j)} = 0, \quad j = 0, 1, 2, \cdots \quad (8)$$

$$\varepsilon_0^{(j)} = \sum_{i=0}^j C_i z^i, \quad j = 0, 1, 2, \cdots \quad (9)$$

$$\varepsilon_{k+1}^{(j)} = \varepsilon_{k-1}^{(j+1)} + (\varepsilon_k^{(j+1)} - \varepsilon_k^{(j)})^{-1}, \quad j, k \geq 0 \quad (10)$$

类似于矩阵情况, 根据式 (8)~(10), 上述张量  $\varepsilon$ -算法组成了一个称为  $\varepsilon$ -表的二维数组:

$$\begin{matrix} \varepsilon_{-1}^{(0)} & & & & & \\ & \varepsilon_0^{(0)} & & & & \\ \varepsilon_{-1}^{(1)} & & \varepsilon_1^{(0)} & & & \\ & \varepsilon_0^{(1)} & & \varepsilon_2^{(0)} & & \\ \varepsilon_{-1}^{(2)} & & \varepsilon_1^{(1)} & & \cdots & \\ & \vdots & & \vdots & & \\ \vdots & & \vdots & & & \end{matrix}$$

中

$$\begin{aligned} \varepsilon_0^{(0)} &= \mathcal{I}, \varepsilon_0^{(1)} = \mathcal{I} + \mathcal{A}t \\ \varepsilon_1^{(0)} &= (\varepsilon_0^{(1)} - \varepsilon_0^{(0)})^{-1} = \frac{1}{\mathcal{A}t} \\ \varepsilon_0^{(-1)} &= 0, \varepsilon_1^{(-1)} = (\varepsilon_0^{(0)} - \varepsilon_0^{(-1)})^{-1} = \frac{1}{\mathcal{I}} \end{aligned}$$

格式 II:  $\varepsilon_2^0 = [\frac{2}{2}]_e = \varepsilon_0^{(1)} + (\varepsilon_1^{(1)} - \varepsilon_1^{(0)})^{-1}$ , 其中

$$\begin{aligned} \varepsilon_1^{(1)} &= (\varepsilon_0^{(2)} - \varepsilon_0^{(1)})^{-1} = \frac{2}{\mathcal{A}^2 t^2} \\ \varepsilon_0^{(2)} &= \mathcal{I} + \mathcal{A}t + \frac{\mathcal{A}^2 t^2}{2} \end{aligned}$$

格式 III:  $\varepsilon_2^1 = [\frac{3}{2}]_e = \varepsilon_0^{(2)} + (\varepsilon_1^{(2)} - \varepsilon_1^{(1)})^{-1}$ , 其中

$$\begin{aligned} \varepsilon_1^{(2)} &= (\varepsilon_0^{(3)} - \varepsilon_0^{(2)})^{-1} = \frac{6}{\mathcal{A}^3 t^3} \\ \varepsilon_0^{(3)} &= \mathcal{I} + \mathcal{A}t + \frac{\mathcal{A}^2 t^2}{2} + \frac{\mathcal{A}^3 t^3}{6} \end{aligned}$$

中

格式 IV:  $\varepsilon_4^{-1} = [\frac{3}{4}]_e = \varepsilon_2^{(0)} + (\varepsilon_3^{(0)} - \varepsilon_3^{(-1)})^{-1}$ , 其中

$$\begin{aligned} \varepsilon_3^{(0)} &= \varepsilon_1^{(1)} + (\varepsilon_2^{(1)} - \varepsilon_2^{(0)})^{-1} \\ \varepsilon_3^{(-1)} &= \varepsilon_1^{(0)} + (\varepsilon_2^{(0)} - \varepsilon_2^{(-1)})^{-1} \end{aligned}$$

格式 V:  $\varepsilon_4^0 = [\frac{4}{4}]_e = \varepsilon_2^{(1)} + (\varepsilon_3^{(1)} - \varepsilon_3^{(0)})^{-1}$ , 其中

$$\varepsilon_3^{(1)} = \varepsilon_1^{(2)} + (\varepsilon_2^{(2)} - \varepsilon_2^{(1)})^{-1}$$

格式 VI:  $\varepsilon_4^1 = [\frac{5}{4}]_e = \varepsilon_2^{(2)} + (\varepsilon_3^{(2)} - \varepsilon_3^{(1)})^{-1}$ , 其中

$$\varepsilon_3^{(2)} = \varepsilon_1^{(3)} + (\varepsilon_2^{(3)} - \varepsilon_2^{(2)})^{-1}$$

例 2. 设张量  $\mathcal{A} \in \mathbf{R}^{2 \times 2 \times 2}$ , 其张量指数函数为

$$\begin{aligned} \exp(\mathcal{A}t) &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} + \\ &\begin{bmatrix} 0 & 1 & 0 & 2 \\ 0 & -2 & 0 & -1 \end{bmatrix} t + \\ &\begin{bmatrix} 0 & -2 & 0 & -\frac{5}{2} \\ 0 & \frac{5}{2} & 0 & 2 \end{bmatrix} t^2 + \end{aligned}$$

其中, 根据  $\varepsilon$ -算法产生的元素  $\varepsilon_{k+1}^{(j)}$ , 它的上标  $j$  表示斜行数, 下标  $k+1$  表示列数. 每个元素的产生与一个菱形有关, 例如, 由  $\varepsilon_0^{(1)}, \varepsilon_1^{(0)}, \varepsilon_1^{(1)}, \varepsilon_2^{(0)}$  所组成的菱形, 现在要计算  $\varepsilon_2^{(0)}$ , 由式 (10), 将  $\varepsilon_1^{(1)}$  减去  $\varepsilon_1^{(0)}$ , 再取广义逆 (3), 然后将结果加上  $\varepsilon_0^{(1)}$ , 从而得到  $\varepsilon_2^{(0)}$ .

定理 1. (恒等定理) 利用张量广义逆 (5), 根据  $\varepsilon$ -算法 (8)~(10) 构造  $\varepsilon_{2k}^{(j)}$ , 如果在计算过程中没有出现分母为零的情形, 则广义逆 Padé 逼近  $[\frac{j+2k}{2k}]_f$  存在, 且成立恒等式:

$$\varepsilon_{2k}^{(j)} = \left[ \frac{j+2k}{2k} \right]_f, \quad j, k \geq 0 \quad (11)$$

证明. 该证明与矩阵情况类似, 可以参考文献 [15]. □

设张量指数函数为

$$\exp(\mathcal{A}t) = \mathcal{I} + \mathcal{A}t + \frac{1}{2!} \mathcal{A}^2 t^2 + \frac{1}{3!} \mathcal{A}^3 t^3 + \dots \quad (12)$$

给出计算张量指数函数 (12) 的张量  $\varepsilon$ -算法如下:

算法 1. (计算张量指数函数的  $\varepsilon$ -算法):

输入: 张量  $\mathcal{A}$ 、自变量  $t$  和需要逼近的阶数  $j_{\max}$  和  $k_{\max}$  (偶数) 的值.

1) 计算  $\varepsilon$ -表的第一列,  
 $\varepsilon_{-1}^{(j)} = 0, j = 0, 1, 2, \dots, j_{\max} + 1.$

2) 计算  $\varepsilon$ -表的第二列,  
 $\varepsilon_0^{(j)} = \sum_{i=0}^j \frac{1}{i!} \mathcal{A}^i t^i, j = 0, 1, 2, \dots, j_{\max}.$

3) 逐列计算  $\varepsilon$ -表的第三列至第  $k_{\max} + 2$  列,  
 $\varepsilon_{k+1}^{(j)} = \varepsilon_{k-1}^{(j+1)} + (\varepsilon_k^{(j+1)} - \varepsilon_k^{(j)})^{-1}, j, k \geq 0.$

输出: 计算结果  $\varepsilon_{2k}^{(j)} = [\frac{j+2k}{2k}]_e$ , 即为所求张量指数的  $[\frac{j+2k}{2k}]$  型广义逆 Padé 逼近, 其中在第 2) 步计算  $\mathcal{A}^i$  时用到定义 1 给出的张量  $t$ -积.

### 2.3 张量 $\varepsilon$ -算法的常用格式

给定张量指数函数 (12), 下面给出张量  $\varepsilon$ -算法的几个常用格式:

格式 I:  $\varepsilon_2^{-1} = [\frac{1}{2}]_e = \varepsilon_0^{(0)} + (\varepsilon_1^{(0)} - \varepsilon_1^{(-1)})^{-1}$ , 其

$$\begin{aligned} & \left[ \begin{array}{cc|cc} 0 & \frac{13}{6} & 0 & \frac{7}{3} \\ 0 & -\frac{7}{3} & 0 & -\frac{13}{6} \end{array} \right] t^3 + \\ & \left[ \begin{array}{cc|cc} 0 & -\frac{5}{3} & 0 & -\frac{41}{24} \\ 0 & \frac{41}{24} & 0 & \frac{5}{3} \end{array} \right] t^4 + \dots = \\ & \mathcal{I} + \mathcal{A}t + \frac{1}{2!}\mathcal{A}^2t^2 + \frac{1}{3!}\mathcal{A}^3t^3 + \\ & \frac{1}{4!}\mathcal{A}^4t^4 + \dots \end{aligned} \tag{13}$$

计算  $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$  型 GITPA.

由常用格式 II 得

$$\begin{aligned} \varepsilon_2^{(0)} &= \begin{bmatrix} 2 \\ 2 \end{bmatrix}_e = \varepsilon_0^{(0)} + (\varepsilon_1^{(0)} - \varepsilon_1^{(-1)})^{-1} = \\ \mathcal{I} + \mathcal{A}t + \frac{1}{\frac{\mathcal{A}^2t^2}{2} - \frac{1}{\mathcal{A}t}} &= \\ \frac{\begin{bmatrix} a_1 & a_2 & 0 & a_3 \\ 0 & a_1 - a_3 & 0 & -a_2 \end{bmatrix}}{a_1} &= \frac{\mathcal{P}_2(t)}{q_2(t)} \end{aligned}$$

其中

$$\begin{aligned} a_1 &= 41 + 140t + 125t^2 \\ a_2 &= 40t^2 + 41t \\ a_3 &= 155t^2 + 82t \end{aligned}$$

由定义 5,  $\varepsilon_2^{(0)} = \frac{\mathcal{P}_2(t)}{q_2(t)}$  是式 (13) 的  $\begin{bmatrix} 2 \\ 2 \end{bmatrix}$  型 GITPA, 满足以下条件:

- 1)  $\partial\mathcal{P}_2(t) = 2, \partial q_2(t) = 2;$
- 2)  $q_2(t) \parallel \mathcal{P}_2(t) \parallel^2;$  这里

$$\begin{aligned} \|\mathcal{P}_2(t)\|^2 &= 2(a_1^2 + a_2^2 + a_3^2) - 2a_1a_3 = \\ &= 2(a_1^2 + 205t^2a_1 - a_1a_3) = \\ &= 2a_1(a_1 + 205t^2 - a_3) \end{aligned}$$

- 3)  $q_2(t)\exp(\mathcal{A}t) - \mathcal{P}_2(t) = O(t^3).$

### 3 数值实验

本节计算张量  $\varepsilon$ -算法的近似值和精确值的误差, 并将本文的方法与计算张量指数函数的截断法进行比较. 首先给出目前通常使用的无穷序列截断法:

**算法 2.** (无穷序列截断法<sup>[10]</sup>)

输入: 张量  $\mathcal{A}$ 、自变量  $t$  和误差限  $\epsilon_{\text{tol}}$  的值.

- 1) 初始化  $n = 0$  和  $\exp(\mathcal{A}t) := \mathcal{I}.$
- 2)  $n := n + 1.$
- 3) 计算  $\frac{t^n}{n!}$  和  $\mathcal{A}^n.$

4) 将该项计算结果加到

$$\exp(\mathcal{A}t) := \exp(\mathcal{A}t) + \frac{t^n}{n!}\mathcal{A}^n$$

5) 判断是否停止, 若

$$\frac{\|\mathcal{A}^n\|t^n}{n!} < \epsilon_{\text{tol}}$$

则停止, 否则回到第 2) 步.

输出:  $\exp(\mathcal{A}t).$

**例 3.** 设张量  $\mathcal{A} \in \mathbf{R}^{2 \times 2 \times 2}, \mathcal{A}$  中的元素为

$$a_{121} = \frac{1}{2}, a_{221} = -\frac{2}{3}, a_{122} = -\frac{1}{2}, a_{222} = \frac{2}{3}$$

其余均为 0, 它的指数函数展开式为

$$\begin{aligned} \exp(\mathcal{A}t) &= \left[ \begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array} \right] + \\ & \left[ \begin{array}{cc|cc} 0 & \frac{1}{2} & 0 & \frac{2}{3} \\ 0 & -\frac{2}{3} & 0 & -\frac{1}{2} \end{array} \right] t + \\ & \left[ \begin{array}{cc|cc} 0 & -\frac{1}{3} & 0 & -\frac{15}{72} \\ 0 & \frac{15}{72} & 0 & \frac{1}{3} \end{array} \right] t^2 + \\ & \left[ \begin{array}{cc|cc} 0 & \frac{19}{144} & 0 & \frac{43}{324} \\ 0 & -\frac{43}{324} & 0 & -\frac{19}{144} \end{array} \right] t^3 + \\ & \left[ \begin{array}{cc|cc} 0 & -\frac{25}{648} & 0 & -\frac{128}{3315} \\ 0 & \frac{128}{3315} & 0 & \frac{25}{648} \end{array} \right] t^4 + \dots = \\ & \mathcal{I} + \mathcal{A}t + \frac{1}{2!}\mathcal{A}^2t^2 + \frac{1}{3!}\mathcal{A}^3t^3 + \\ & \frac{1}{4!}\mathcal{A}^4t^4 + \dots \end{aligned} \tag{14}$$

用  $\varepsilon$ -算法和截断法计算该指数函数.

#### 3.1 $\begin{bmatrix} 4 \\ 4 \end{bmatrix}$ 型 GITPA 的数值实验

由常用格式 V 计算  $\begin{bmatrix} 4 \\ 4 \end{bmatrix}$  型 GITPA, 数值实验结果见表 1.

表 1 中由常用格式 V 计算的近似值记作  $A$  值, 根据截断法即算法 2 取指数函数展开式 (14) 前 15 项计算得到的结果为精确值, 记作  $E$  值. 表 1 中分别列出张量下标为 (1, 2, 1), (2, 2, 1), (1, 2, 2), (2, 2, 2) 在点 0.2, 0.4, 0.6, 0.8, 1 相应的近似值与精确值. 表 1 中最后一列表示近似值与精确值残差的范数的平方, 记作  $RES$ , 其计算公式如下:

$$RES(t) = \left\| \exp(\mathcal{A}t) - \left[ \frac{\mathcal{A}}{4} \right]_{e^{\mathcal{A}t}}(t) \right\|^2$$

其中, 模范数  $\|\cdot\|$  由式 (3) 定义.

表 1 显示, 张量  $\varepsilon$ -算法得到的近似值能够达到较高的计算精度. 可以发现, 自变量的值越接近 0, 张量  $\varepsilon$ -算法的逼近效果越好, 计算结果基本符合理论预期.

### 3.2 两个算法数值实验的比较

首先, 在张量指数函数 (14) 中令  $t = 2$ , 取前 15 项计算得到的精确值为

$$\begin{bmatrix} 1 & 0.3098 & 0 & 0.5932 \\ 0 & 0.4068 & 0 & -0.3098 \end{bmatrix}$$

根据算法 1 分别计算  $\left[\frac{2}{2}\right], \left[\frac{4}{4}\right], \left[\frac{6}{6}\right]$  型 GITPA, 并

与截断算法 2 进行比较, 得到的数值实验结果见表 2.

表 2 显示,  $\left[\frac{6}{6}\right]$  型 GITPA 计算的数值结果与截断法取前 13 项相加的数值结果的精确度相近, 由此说明本文给出的张量  $\varepsilon$ -算法是有效的.

### 3.3 两个算法计算复杂度的比较

下面从计算复杂度的角度来对两种算法进行比较. 由于需要自乘, 前两维的维数必须相等. 对于一个  $l \times l \times n$  的三阶张量  $\mathcal{A}$ , 1 次张量  $t$ -积的计算复杂度为  $O(l^3 n^2)$ ; 而 1 次张量的广义逆等价于 2 次张量的数乘计算, 其计算复杂度为  $O(l^2 n)$ . 因此, 随着张量维数的增大, 计算张量  $t$ -积显然要比计算张量的广义逆付出更大的成本.

表 1  $\left[\frac{4}{4}\right]$  型 GITPA-算法数值实验

Table 1 The numerical experiment of  $\left[\frac{4}{4}\right]$  type GITPA-algorithm

| $x$ |       | (1, 2, 1)  | (2, 2, 1)  | (1, 2, 2)  | (2, 2, 2)   | $RES$                  |
|-----|-------|------------|------------|------------|-------------|------------------------|
| 0.2 | $E$ 值 | 0.08766299 | 0.87955329 | 0.12044671 | -0.08766299 | $5.69 \times 10^{-13}$ |
|     | $A$ 值 | 0.08766327 | 0.87955283 | 0.12044717 | -0.08766327 |                        |
| 0.4 | $E$ 值 | 0.15420167 | 0.78130960 | 0.21869040 | -0.15420167 | $3.74 \times 10^{-10}$ |
|     | $A$ 值 | 0.15420895 | 0.78129804 | 0.21870196 | -0.15420895 |                        |
| 0.6 | $E$ 值 | 0.20408121 | 0.70078192 | 0.29921808 | -0.20408121 | $1.40 \times 10^{-8}$  |
|     | $A$ 值 | 0.20412606 | 0.70071136 | 0.29928864 | -0.20412606 |                        |
| 0.8 | $E$ 值 | 0.24081224 | 0.63444735 | 0.36555265 | -0.24081224 | $1.63 \times 10^{-7}$  |
|     | $A$ 值 | 0.24096630 | 0.63420702 | 0.36579298 | -0.24096630 |                        |
| 1   | $E$ 值 | 0.26715410 | 0.57953894 | 0.42046106 | -0.26715410 | $1.01 \times 10^{-6}$  |
|     | $A$ 值 | 0.26753925 | 0.57894247 | 0.42105753 | -0.26753925 |                        |

表 2 算法 1 和算法 2 的数值实验比较

Table 2 The numerical experiment comparison of Algorithm 1 and Algorithm 2

| $\left[\frac{j+2k}{2k}\right]$ | 张量 $\varepsilon$ -算法 |           |           |           | $n_{\max}$ | $\sum_{n=0}^{n_{\max}} \frac{1}{n!} \mathcal{A}^n t^n$ |           |           |           |
|--------------------------------|----------------------|-----------|-----------|-----------|------------|--------------------------------------------------------|-----------|-----------|-----------|
|                                | $a_{121}$            | $a_{221}$ | $a_{122}$ | $a_{222}$ |            | $a_{121}$                                              | $a_{221}$ | $a_{122}$ | $a_{222}$ |
| $\left[\frac{2}{2}\right]$     | 0.4235               | 0.3513    | 0.6487    | -0.4235   | 1          | 1.0000                                                 | -0.3333   | 1.3333    | -1.0000   |
| $\left[\frac{4}{4}\right]$     | 0.3049               | 0.4141    | 0.5859    | -0.3049   | 2          | -0.3333                                                | 1.0556    | -0.0556   | 0.3333    |
| $\left[\frac{6}{6}\right]$     | 0.3098               | 0.4068    | 0.5932    | -0.3098   | 3          | 0.7222                                                 | -0.0062   | 1.0062    | -0.7222   |
| -                              | -                    | -         | -         | -         | 4          | 0.1049                                                 | 0.6116    | 0.3884    | -0.1049   |
| -                              | -                    | -         | -         | -         | 5          | 0.3931                                                 | 0.3234    | 0.6766    | -0.3931   |
| -                              | -                    | -         | -         | -         | 6          | 0.2810                                                 | 0.4355    | 0.5645    | -0.2810   |
| -                              | -                    | -         | -         | -         | 7          | 0.3184                                                 | 0.3981    | 0.6019    | -0.3184   |
| -                              | -                    | -         | -         | -         | 8          | 0.3075                                                 | 0.4090    | 0.5910    | -0.3075   |
| -                              | -                    | -         | -         | -         | 9          | 0.3103                                                 | 0.4062    | 0.5938    | -0.3103   |
| -                              | -                    | -         | -         | -         | 10         | 0.3097                                                 | 0.4069    | 0.5931    | -0.3097   |
| -                              | -                    | -         | -         | -         | 11         | 0.3098                                                 | 0.4067    | 0.5933    | -0.3098   |
| -                              | -                    | -         | -         | -         | 12         | 0.3098                                                 | 0.4068    | 0.5932    | -0.3098   |

根据张量  $\varepsilon$ -算法计算  $\begin{bmatrix} 6 \\ 6 \end{bmatrix}$  型 GITPA 所使用的是张量指数函数 (14) 的前 7 项, 需要进行 5 次张量  $t$ -积和 21 次广义逆计算, 而达到相同精度, 截断法需要取前 13 项, 需要进行 11 次张量  $t$ -积和 12 次数乘运算. 两种算法的计算复杂度比较见表 3. 从表 3 可以看出: 在计算过程中, 张量  $\varepsilon$ -算法计算复杂度为  $5l^3n^2 + 42l^2n$ , 截断法计算复杂度为  $11l^3n^2 + 12l^2n$ .

表 3 算法 1 和算法 2 的计算复杂度分析

Table 3 The analysis of computational complexity of Algorithm 1 and Algorithm 2

|          | 计算一次<br>复杂度 | $\varepsilon$ -算法: $\begin{bmatrix} 6 \\ 6 \end{bmatrix}$ 型<br>计算 | 截断法: 取 13 项<br>计算   |
|----------|-------------|-------------------------------------------------------------------|---------------------|
| $t$ -积运算 | $l^3n^2$    | 5                                                                 | 11                  |
| 数乘运算     | $l^2n$      | 21                                                                | 12                  |
| 范数运算     | $l^2n$      | 21                                                                | 0                   |
| 总和       |             | $5l^3n^2 + 42l^2n$                                                | $11l^3n^2 + 12l^2n$ |

**例 4.** 为了比较两种算法所需的时间, 在 Matlab 中选取 'seed' = 1 : 100, 随机生成张量  $\mathcal{A}$  如下:  $3 \times 3 \times 3, 10 \times 10 \times 10, 20 \times 20 \times 20, 30 \times 30 \times 30, 40 \times 40 \times 40$  各 100 个张量, 分别使用算法 1 和算法 2 计算其张量指数函数 (12), 其中, 本文给出的算法 1 计算的是  $\begin{bmatrix} 6 \\ 6 \end{bmatrix}$  型 GITPA, 算法 2 计算的是指数函数展开的前 13 项之和. 两个算法计算所需时间见表 4, 表中数据均是由内存 8 GB, 主频 2.50 GHz, 操作系统 Windows 10, 处理器 Inter(R) Core(TM) i5-7300HQ 的笔记本电脑的 Matlab (R2015b) 得到.

表 4 不同维数下两种算法的运行时间表 (s)

Table 4 The consuming time of two algorithms in different dimensions (s)

| 张量维数                     | 张量 $\varepsilon$ -算法运行时间 | 截断法运行时间   |
|--------------------------|--------------------------|-----------|
| $3 \times 3 \times 3$    | 1.755262                 | 1.103832  |
| $10 \times 10 \times 10$ | 2.272663                 | 2.164860  |
| $20 \times 20 \times 20$ | 3.831094                 | 4.805505  |
| $30 \times 30 \times 30$ | 6.419004                 | 10.545785 |
| $40 \times 40 \times 40$ | 15.814063                | 30.012744 |

图 1 显示, 算法 1 计算  $\begin{bmatrix} 6 \\ 6 \end{bmatrix}$  型 GITPA 与算法 2 即截断法计算前 13 项之和相比, 在维数较低的情况下 (维数低于  $10 \times 10 \times 10$ ), GITPA 的运行时间略高于截断法; 当维数较高时, 本文的算法 1 逐渐显示出优越性, 可以很好地降低计算时间.

## 4 结论

目前, 国内外还没有看到计算张量的逆和广义逆的有效算法, 本文提出的一种张量广义逆即定

义 3 是一个实用的计算方法, 它是同一类型的向量广义逆 (Graves-Morris<sup>[14]</sup>) 和矩阵广义逆 (顾传青<sup>[15-16, 18]</sup>) 在张量上的推广. 在该张量广义逆的基础上, 本文得到了计算张量指数函数 (12) 的张量  $\varepsilon$ -算法. 从计算张量指数函数 (14) 的数值实验来看, 与目前通用的无穷序列截断法相比较, 在计算精度和计算复杂度上具有一定的优势, 在张量维数比较大时, 这个优势会更加明显. 下面的研究工作从两个方面进行, 一是用本文提出的广义逆张量 Padé 逼近 (GITPA) 方法来考虑控制论中的模型简化问题, 二是对张量  $\varepsilon$ -算法的稳定性进行探讨.

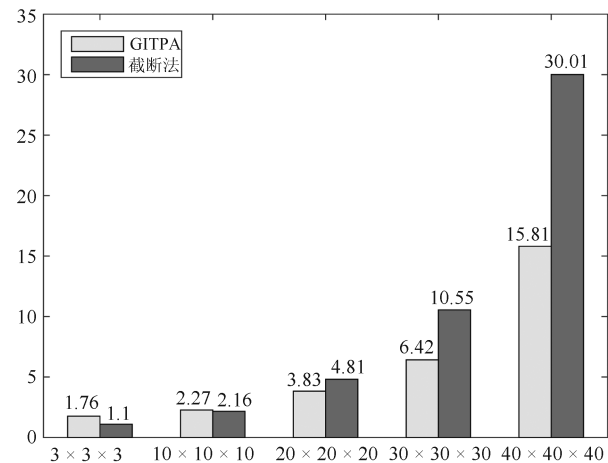


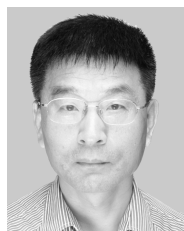
图 1 不同维数下两种算法运行时间直方图 (s)

Fig. 1 The time-consuming comparison histogram of two algorithms in different dimensions (s)

## References

- Chen Wei-Hong, An Ji-Yao, Li Ren-Fa, Li Wan-Li. Review on deep-learning-based cognitive computing. *Acta Automatica Sinica*, 2017, **43**(11): 1886–1897 (陈伟宏, 安吉尧, 李仁发, 李万里. 深度学习认知计算综述. *自动化学报*, 2017, **43**(11): 1886–1897)
- Wang Shou-Hui, Yu Hong-Tao, Huang Rui-Yang, Ma Qing-Qing. A temporal link prediction method based on motif evolution. *Acta Automatica Sinica*, 2016, **42**(5): 735–745 (王守辉, 于洪涛, 黄瑞阳, 马青青. 基于模体演化的时序链路预测方法. *自动化学报*, 2016, **42**(5): 735–745)
- Hu Yang, Zhang Dong-Bo, Duan Qi. An improved rotation-invariant HDO local description for object recognition. *Acta Automatica Sinica*, 2017, **43**(4): 665–673 (胡扬, 张东波, 段琪. 目标鲁棒识别的抗旋转 HDO 局部特征描述. *自动化学报*, 2017, **43**(4): 665–673)
- Lin Hong-Bin, Shao Yan-Chuan, Wang Wei. The 2D analytical tensor voting algorithm. *Acta Automatica Sinica*, 2016, **42**(3): 472–480 (林洪彬, 邵艳川, 王伟. 二维解析张量投票算法研究. *自动化学报*, 2016, **42**(3): 472–480)
- Batselier K, Chen Z M, Wong N. A tensor network Kalman filter with an application in recursive MIMO Volterra system identification. *Automatica*, 2017, **84**: 17–25

- 6 Liu X D, Yu Y, Li Z, Iu H H C, Fernando T. An efficient algorithm for optimally reshaping the TP model transformation. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2017, **64**(10): 1187–1191
- 7 Lee C S. Human action recognition using tensor dynamical system modeling. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops. Honolulu, HI, USA: IEEE, 2017. 1935–1939
- 8 Koniusz P, Cherian A, Porikli F. Tensor representations via kernel linearization for action recognition from 3D skeletons. In: Proceedings of European Conference on Computer Vision. Cham, Switzerland: Springer, 2016. 37–53
- 9 De Souza Neto E A. The exact derivative of the exponential of an unsymmetric tensor. *Computer Methods in Applied Mechanics and Engineering*, 2001, **190**(18–19): 2377–2383
- 10 de Souza Neto E A, Perić D, Owen D R J. *Computational Methods for Plasticity: Theory and Applications*. United Kingdom: Wiley, 2008.
- 11 Kolda T G, Bader B W. Tensor decompositions and applications. *SIAM Review*, 2009, **51**(3): 455–500
- 12 Kilmer M E, Braman K, Hao N, Hoover R C. Third-order tensors as operators on matrices: a theoretical and computational framework with applications in imaging. *SIAM Journal on Matrix Analysis and Applications*, 2013, **34**(1): 148–172
- 13 Kilmer M E, Martin C D, Perrone L. A Third-order Generalization of the Matrix SVD as a Product of Third-order Tensors. Tech. Report TR-2008-4, Tufts University, USA, 2008
- 14 Graves-Morris P R. Vector valued rational interpolants I. *Numerische Mathematik*, 1983, **41**(3): 331–348
- 15 Gu C Q. Generalized inverse matrix Padé approximation on the basis of scalar products. *Linear Algebra and its Applications*, 2001, **322**(1–3): 141–167
- 16 Gu C Q. A practical two-dimensional thiele-type matrix Padé approximation. *IEEE Transactions on Automatic Control*, 2003, **48**(12): 2259–2263
- 17 Gu Chuan-Qing. New algorithm for matrix exponentials by Padé approximation. *Acta Automatica Sinica*, 1999, **25**(1): 94–99  
(顾传青. 关于矩阵指数的 Padé 逼近新算法. 自动化学报, 1999, **25**(1): 94–99)
- 18 Gu Chuan-Qing. Matrix  $\varepsilon$ -algorithm and  $\eta$ -algorithm for Samelson inverse and application to matrix exponentials. *Journal on Numerical Methods and Computer Applications*, 1999, **20**(3): 223–230  
(顾传青. 基于 Samelson 逆的矩阵  $\varepsilon$ -算法和  $\eta$ -算法及对矩阵指数的应用. 数值计算与计算机应用, 1999, **20**(3): 223–230)



**顾传青** 上海大学数学系教授. 主要研究方向为数值代数, 张量与矩阵计算, 有理逼近及其在控制论应用. 本文通信作者. E-mail: cqgu@shu.edu.cn

(**GU Chuan-Qing** Professor in the Department of Mathematics, Shanghai University. His research interest covers numerical algebra, tensor and matrix calculation, rational approximation and its application in control theory. Corresponding author of this paper.)



**唐鹏飞** 上海大学数学系硕士研究生. 主要研究方向为张量 Padé 逼近及其在控制论中应用.

E-mail: soccer.fly@163.com

(**TANG Peng-Fei** Master student in the Department of Mathematics, Shanghai University. His research interest covers tensor Padé approximation and its application in control theory.)



**陈之兵** 深圳大学数学与统计学院教授. 主要研究方向为矩阵计算, 数值逼近及其应用. E-mail: chenzb@szu.edu.cn

(**CHEN Zhi-Bing** Professor at the School of Mathematics and Statistics, Shenzhen University. His research interest covers matrix calculation, numerical approximation and its application.)