Fast Algorithms Related to Approximate Greatest Common Divisors of Univariate Polynomials

Bingyu Li

Directed by **Prof. Zhuojun Liu, Associate Prof. Lihong Zhi**

Key Laboratory of Mathematics Mechanization

Institute of Systems Science

Academy of Mathematics and Systems Science

Chinese Academy of Sciences

Beijing 100080 **China**

April, 2006

摘　　要

具有有限精度的一元多项式的近似最大公因子的计算在控制理论, 网络理论和计算机辅助设计中都有着重要应用. 作为符号数值混合计算领域的一个分支，求解近似最大公因子问题的一个主要方法是把这一数值多项式代数问题转化成矩阵代数问题，而矩阵在数值计算中是可以用数值稳定的技巧来处理的. 位移结构矩阵是把数值计算和多项式代数计算联系起来的一个有效的桥梁.

位移结构矩阵的概念最早是由 Kailath, Kung 和 Morf 在 1979 年提出的. 数值计算中, 对于位移结构矩阵, 人们可以发展出相比于经典算法复杂度更低的快速算法; 因而, 基于位移结构矩阵我们可以更加高效地处理数值多项式代数中的问题.

本文研究与求解一元近似最大公因子相关的快速算法，并做了如下两方面工作：

1. 提出了一个求解 Sylvester 矩阵数值秩的快速算法，并讨论其在计算 Sylvester 矩阵的奇异值方面的应用. Sylvester 矩阵的数值秩与一元近似最大公因子的次数上界密切相关 [23]. 计算 Sylvester 矩阵的数值秩最稳定的方法是应用奇异值分解；但这样做的代价是需要关于 Sylvester 矩阵阶数立方次的运算量. 基于 Sylvester 矩阵 S 的位移结构，我们提出了用广义 Schur 算法实现快速 Cholesky 分解来计算 S^TS 的数值秩，进而得到 S 的数值秩的快速算法. 这一算法具有 Sylvester 矩阵阶数平方次的复杂度，相比于奇异值分解来说其复杂度降了一阶.

2. 用快速算法求解了 Sylvester 矩阵的结构低秩逼近问题. 在文献 [53] 中，作者提出通过求解 Sylvester 矩阵的结构低秩逼近矩阵，来计算所给多项式的极小扰动的多项式，其具有次数不低于给定正整数的最大公因子. 这一算法可以直接应用于求解近似最大公因子. 这一算法的复杂度取决于算法中求解一系列最小二乘问题的运算量，利用经典的算法，其复杂度为 $O(st^2)$, 这里 s 和 t 分别代表最小二乘问题系数阵的行数和列数. 本文中基于最小二乘问题系数阵的位移结构，应用低秩的生成子矩阵，用平方阶的复杂度 $O(s^2)$ 求解了这一问题.

关键词: 近似最大公因子, 位移结构矩阵, 快速算法，数值秩，结构低秩逼近.

Fast Algorithms Related to Approximate Greatest Common Divisors of Univariate Polynomials

Bingyu Li (majored in Computer Algebra)
Directed by **Prof. Zhuojun Liu, Associate Prof. Lihong Zhi**

Abstract

The computation of approximate greatest common divisors (GCD) of univariate polynomials given with limited accuracy is of importance in control theory, network theory and computer aided design. As a class of fundamental problem in the field of symbolic-numeric hybrid computations, the problem of approximate GCD is expected to be resolved by transforming such a problem of numerical polynomial algebra into a problem on matrices, which can be treated by numerical techniques ensuring numerically stability in numerical computations. The notion of displacement structured matrices is an effective bridge to build a link between numerical computations and polynomial algebra computations.

The notion of displacement structured matrix was firstly introduced by Kailath, Kung and Morf in 1979. In numerical computations, with respect to displacement structured matrices, it is possible to develop fast algorithms with much lower complexity compared to those classic algorithms; As a result, based on displacement structured matrices, it is possible to solve problems in numerical polynomial algebra much more efficiently.

In this thesis, we investigate fast algorithms related to the computation of approximate GCD of univariate polynomials and propose two new fast algorithms as follows:

1. We derive a fast rank-revealing algorithm for the Sylvester matrix and discuss its application on the computation of singular values of the Sylvester matrix. The

numerical rank of the Sylvester matrix is closely related to an upper bound on the degree of approximate GCD [23]. As to the computation of numerical rank of the Sylvester matrix, the most reliable method is by applying the singular value decomposition (SVD). However, the manipulation of SVD involves cubic amount of work and is costly. Considering the displacement structure of the Sylvester matrix S, we propose to reveal the numerical rank of S through pursuing numerical rank of S^TS, which is efficiently achieved by performing fast Cholesky factorization based on the generalized Schur algorithm. Such a rank-revealing algorithm is of quadratic complexity.

2. We implement fast low rank approximation of a Sylvester matrix. For two given polynomials, in [53] the authors proposed to find minimal perturbed polynomials with a GCD, which is of degree not less than a given integer, through computing low rank approximation of their Sylvester matrix. This algorithm can be directly used to compute approximate GCD. For their algorithm, the overall computation time depends on solving a sequence least squares problems (LSP). The complexity of the classic algorithms for solving LSP is $O(st^2)$, here s and t denote the row dimensions and column dimensions of those least squares systems. Using a displacement structured matrix approach, we can solve LSP with quadratic complexity $O(s^2)$.

Keywords: approximate greatest common divisor, displacement structured matrix, fast algorithm, numerical rank, structured low rank approximation.

目　录

本文主要记号

$\mathbb{R}$	: 实数域
$\mathbb{C}$	: 复数域
$\mathbb{R}^{m\times n}$	: $m\times n$的实矩阵全体
$\mathbf{0}$	: 零矩阵
$\gcd(a,b)$	: 一元多项式$a(x)$和 $b(x)$ 的最大公因子
deg	: 多项式的次数
$\Vert a(x)\Vert_2$	: 一元多项式 $a(x)$ 的系数向量 2- 范数
$\sigma_i=\sigma_i(\cdot)$	: 矩阵的第 i 大的奇异值
$\Vert\cdot\Vert$	: 矩阵的 2- 范数
$\Vert\cdot\Vert_F$	: 矩阵的 Frobenius- 范数
$\kappa(\cdot)$	: 矩阵的 2- 范数条件数
$\mathrm{rank}(\cdot)$	: 矩阵的秩
Z_i	: 一个$i\times i$的下移位矩阵
I_i	: 一个$i\times i$的单位矩阵
$A(i,:)$	: 矩阵A的第i行
$A(:,j)$	: 矩阵A的第j列
λ_i	: 矩阵的特征值
$\mathrm{diag}(a_1,\ldots,a_i)$	: 由标量$a_1,\ldots,a_i$构成的对角矩阵
$\mathrm{diag}(A_1,\ldots,A_i)$	: 由矩阵$A_1,\ldots,A_i$构成的块对角矩阵
u	: 机器精度
$\hat{\cdot}$	: 一个计算的量

注 0.0.1 本文中的算法复杂度指的是算法包含的运算量，即算法中加，减，乘，除运算次数的总和.

第一章　引　　言

本章首先介绍位移结构矩阵的定义，并给出几个基本位移结构矩阵的例子，然后回顾位移结构矩阵在一元多项式代数中，特别是在求解最大公因子问题中已有的研究方法和结果. 最后介绍本文研究的问题和主要结果.

§1.1　位移结构矩阵的概念

设 i 是一个正整数，一个 $i\times i$ 的下移位矩阵 Z_i 是具有如下形状的矩阵：

$$Z_i = \begin{bmatrix} 0 & & & \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{bmatrix}, \tag{1.1.1}$$

即 Z_i 的第一次对角线元素为 1, 其他元素为 0.

一个 $n\times n$ 实对称矩阵 R 的**位移**定义如下：

$$\nabla R = R - ZRZ^T, \tag{1.1.2}$$

其中 Z 是一个 $n\times n$ 的下三角矩阵. Z 的选取依赖于矩阵 R，例如，如果 R 是一个 Toeplitz 阵，那么选取 Z 为一个下移位矩阵 Z_n. 进一步的，设 ∇R 的秩为 r, 如果 r 小于 n 并且其与 R 的阶数 n 无关，那么称 R 为关于位移 (1.1.2) 的一个**结构矩阵**，而称 r 为 R 的**位移秩**. 对称的 ∇R 总可以分解成如下形式：

$$\nabla R = R - ZRZ^T = GJG^T, \tag{1.1.3}$$

其中 G 是一个 $n\times r$ 的矩阵，而 J 是一个符号矩阵：

$$J = \begin{bmatrix} I_p & \mathbf{0} \\ \mathbf{0} & -I_q \end{bmatrix}, p+q=r,$$

其中 p 和 q 分别代表 ∇R 的正的特征值的个数和负的特征值的个数.

设 $z_i, i = 1, \ldots, n$ 是 Z 的对角元素，那么方程 (1.1.3) 可以唯一地确定 R, 只要 Z 满足：

$$1 - z_i z_j \neq 0.$$

（对于更广的情形的讨论可见文献 [52]）. 而矩阵 G 称为 R 的**生成子矩阵**，矩阵对 (G, J) 称为 R 的**生成子对**. 这是基于这样一个事实：在 Z 的定义下，方程 (1.1.3) 的唯一解 R 完全可以用 G 和 J 来确定.

位移结构矩阵的概念也可以扩展到非对称矩阵 R 上. 此时，R 的**位移**要由两个下三角矩阵 Z 和 A 来定义，

$$\nabla R = R - ZRA^T. \tag{1.1.4}$$

通常把具有低秩为 r 的非对称**位移**∇R 分解成

$$\nabla R = R - ZRA^T = GB^T, \tag{1.1.5}$$

这里 G 和 B 都是 $n \times r$ 的矩阵，称为 R 的**生成子**矩阵. 类似的，位移方程 (1.1.5) 可以唯一地确定 R, 只要 Z 和 A 的对角元素满足：

$$1 - z_i a_j \neq 0.$$

位移结构的概念最早是由 Kailath, Kung 和 Morf 在文 [49, 50] 中提出的. 对于矩阵的位移，可以用 Stein 型位移算子来定义，如上 (1.1.3) 和 (1.1.4) 所定义的位移，其中相应的结构矩阵称为 Toeplitz 型 (Toeplitz-like) 矩阵；位移还可以用 Sylvester 型位移算子来定义，此时位移 $\triangle R = ZR - RA^T$ [43]. Sylvester 型位移算子一般用来定义 Hankel 型 (Hankel-like) 矩阵的位移. 最早是 Heinig 和 Rost 在文 [43] 中详细地研究了 Sylvester 型位移算子所定义的位移结构并提出，除了 Hankel 矩阵以外，Cauchy 矩阵和 Vandermonde 矩阵也都属于 Hankel 型矩阵. 而实际上，所有 Toeplitz 型和 Hankel 型矩阵又都是更广的 Toeplitz+Hankel 型 (Toeplitz-plus-Hankel) 矩阵的特殊情形，详见文献 [52].

位移结构概念的提出使得含有平方阶元素的结构矩阵可以由只含有线性阶元素的生成子矩阵来确定. 基于此，关于结构矩阵的算法，如矩阵向量乘法，矩阵三角分解，矩阵求逆等可以从结构矩阵的生成子矩阵出发来实现. 这样发展起来的算法，与直接从原矩阵出发所设计的经典算法比需要更低的复杂度，故成为快速算法.

位移结构矩阵的一个非常重要的性质是，**位移结构由结构矩阵的 Schur 补所继承**. 给定一个矩阵 R,

$$R = \begin{bmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{bmatrix}, \tag{1.1.6}$$

其中 R_{11} 是 R 的一个顺序主子矩阵，并且是可逆的，那么 $S = R_{22} - R_{21}R_{11}^{-1}R_{12}$ 称为 R 关于 R_{11} 的**Schur 补**. 结构矩阵的这一性质被广泛地应用于结构矩阵快速算法中，比如，广义 Schur 算法 [52], 一元 Bézout 矩阵的结构矩阵算法 [13], 基于分而治之策略的结构矩阵快速算法 [16, 65, 27, 70, 77].

注 1.1.1 位移结构矩阵的生成子矩阵不是唯一的. 以一个对称结构矩阵 R 为例，事实上，任取一个 J-**正交矩阵** Θ, 即 Θ 满足 $\Theta J \Theta^T = J$, 如果 G 是一个满足 *(1.1.3)* 的生成子矩阵，那么 $G\Theta$ 亦然.

注 1.1.2 与实数矩阵类似的，位移结构矩阵的定义也可以推广到 *Hermitian* 和非 *Hermitian* 的复矩阵上 *[52].*

1. 几个基本的结构矩阵的位移结构

位移结构的概念很容易由对称 Toeplitz 矩阵描述出来. 设 R 代表一个对称的 Toeplitz 矩阵，$R = [\ t_{|i-j|}\ ]_{i,j=1}^n$, $t_0 = 1$, 容易验证 $R - Z_nRZ_n^T$ 的秩为 2 (除非所有的 $t_i, i \neq 0$ 为零)，可以找到 R 的一个生成子对 (G, J) 满足：

$$R - Z_nRZ_n^T = \begin{bmatrix} 1 & 0 \\ t_1 & t_1 \\ \vdots & \vdots \\ t_{n-1} & t_{n-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_1 & t_1 \\ \vdots & \vdots \\ t_{n-1} & t_{n-1} \end{bmatrix}^T = GJG^T. \tag{1.1.7}$$

类似的，设 R 是一个非对称的 Toeplitz 矩阵，$R = [\ t_{i-j}\ ]_{i,j=1}^n$, 那么 $R - Z_nRZ_n^T$ 的秩为 2 而 R 的生成子矩阵 (G, B) 可以按如下构造：

$$R - Z_nRZ_n^T = \begin{bmatrix} t_0 & 1 \\ t_1 & 0 \\ \vdots & \vdots \\ t_{n-1} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & t_{-1} \\ \vdots & \vdots \\ 0 & t_{-n+1} \end{bmatrix}^T = GB^T. \tag{1.1.8}$$

特别的，如果一个矩阵 R 使得 $R - Z_n R Z_n^T$ 的秩为 2, 那么称这个矩阵为**拟 Toeplitz** (quazi-Toeplitz) 矩阵. 例如，Toeplitz 矩阵的逆矩阵就是一个拟 Toeplitz 矩阵 [52].

下面介绍几个 Hankel 型位移结构矩阵. 设 H 是一个 Hankel 矩阵，$H = [\, h_{i+j}]_{i,j=0}^{n-1}$, 那么

$$Z_n H - H Z_n^T = \begin{bmatrix} 0 & 1 \\ h_0 & 0 \\ \vdots & \vdots \\ h_{n-2} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -h_0 \\ \vdots & \vdots \\ 0 & -h_{n-2} \end{bmatrix}^T = GB^T.$$

需要注意的是，H 中的元素 $\{h_{n-1}, \ldots, h_{2n-2}\}$ 并未出现在如上的生成子矩阵中，所以直接用这里的生成子矩阵无法恢复出 H. 文 [52] 中讨论了几个小的技巧来克服这个困难.

Cauchy 型矩阵是在研究无约束插值问题和系统理论时常遇到的一类矩阵. 一个 $n \times n$ 的 Cauchy 型矩阵 R 的元素有如下形式：

$$R = \left[\frac{\mathbf{u}_i \mathbf{v}_j^T}{f_i - a_j} \right]_{i,j=0}^{n-1},$$

其中 $\mathbf{u}_i$ 和 $\mathbf{v}_j$ 代表两个 $1 \times r$ 的行向量，$\{f_i, a_j\}$ 是实的标量并且对所有的 i, j 满足 $f_i - a_j \neq 0$. 当 $r = 1$, $\mathbf{u}_i = 1 = \mathbf{v}_j$, $i, j = 0, \ldots, n-1$, 得到的矩阵是 Cauchy 矩阵. 容易验证，Cauchy 型矩阵 R 满足

$$FR - RA^T = \begin{bmatrix} \mathbf{u}_0 \\ \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_{n-1} \end{bmatrix} \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_{n-1} \end{bmatrix}^T,$$

其中

$$F = \mathrm{diag}(f_0, \ldots, f_{n-1}), \;\; A = \mathrm{diag}(a_0, \ldots, a_{n-1}),$$

故其位移秩为 r. 特别的，当 R 是 Cauchy 矩阵时，其位移秩为 1.

最后一个例子是 Vandermonde 矩阵，$V = [x_i^j]_{i,j=0}^{n-1}$, 其位移秩为 1: 定义 $A =$

$\mathrm{diag}(1/x_0, \ldots, 1/x_{n-1})$ (假设 $x_i \neq 0$), 那么

$$AV - VZ_n^T = \begin{bmatrix} 1/x_0 \\ 1/x_1 \\ \vdots \\ 1/x_{n-1} \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}^T.$$

注 1.1.3 *Toeplitz* 矩阵, *Hankel* 矩阵, *Cauchy* 矩阵和 *Vandermonde* 矩阵的复合, 如矩阵相乘, 矩阵的逆, 仍然是位移结构矩阵. 特别地, 前面已经提到, *Toeplitz* 矩阵的逆是称为拟 *Toeplitz* 矩阵的结构矩阵. 此外, *Hankel* 矩阵的逆是一个 *Bézout* 矩阵 *[92]*. *Bézout* 矩阵是一个 *Hankel* 型 *(Hankel-like)* 位移结构矩阵 *[13]*. 设 $a(x)$ 和 $b(x)$ 是两个次数分别为 n 和 m 的一元多项式, 这里 $n \geq m$, 那么 $a(x)$ 和 $b(x)$ 的 *Bezoutian* 是如下定义的二元多项式:

$$\frac{a(x)b(y) - b(y)a(x)}{x - y} = \sum_{0 \leq i,j \leq n-1} \theta_{i,j} x^i y^j;$$

而矩阵 $B(a, b) = (\theta_{i,j})$, $i, j = 0, \ldots, n-1$, 则称为 $a(x)$ 和 $b(x)$ 的 *Bézout* 矩阵.

§1.2　结构矩阵在一元多项式代数中的一些应用

符号数值混合计算是近十几年发展的比较活跃的一个研究方向. 如何充分应用数值计算中处理矩阵的技巧, 以至高效并数值稳定地处理数值多项式代数中的问题, 是符号数值混合计算领域所要研究的一个主要内容. 实际上, 多项式代数和数值计算之间往往由一些位移结构矩阵所联系着, 比如: Toeplitz 矩阵, Hankel 矩阵, Cauchy 矩阵和 Vandermonde 矩阵. 充分利用这些矩阵的位移结构, 有助于我们更加高效地处理多项式代数中的问题.

迄今为止, 结构矩阵在快速求解一元多项式代数的诸多基本问题中都有了重要应用, 如最大公因子问题 [12, 14, 13, 73, 77, 94], 有理插值问题 [1, 2, 8, 15, 83, 83, 39, 38, 69, 70], 代数解码问题 [71, 72] 等. 有趣的是, 以上四类基本的位移结构矩阵还可以相互转化 [74, 77]. 这样, 有一个直接的好处是, 可以发展统一的结构矩阵快速算法 [52, 77]. 此外, Vandmonde 矩阵和 Cauchy 矩阵的转化关系已被应用于快速求解多项式插值和

多点赋值问题，详见 [77, 82, 89, 35, 36, 4]. 基于 Cauchy 矩阵的位移结构在列交换和行交换下保持不变的特点，Toeplitz 矩阵与 Cauchy 矩阵的转化关系已被应用于快速而稳定的求解系数阵为 Toeplitz 矩阵的线性方程组和最小二乘问题 [34, 42, 33].

在求解多项式最大公因子问题中结构矩阵的应用主要有以下三个方面：

第一，Sylvester 矩阵的应用. 给出两个次数分别为 m 和 n 的一元多项式 $a(x), b(x) \in \mathbb{R}[x]$, $a(x) = \sum_{i=0}^{m} a_i x^i$, $b(x) = \sum_{i=0}^{n} b_i x^i$, $a(x)$ 和 $b(x)$ 的结式矩阵或 Sylvester 矩阵是 $S \in \mathbb{R}^{(m+n)\times(m+n)}$:

$$S = S(a,b) = \begin{bmatrix} a_m & \cdots & a_1 & a_0 & & \\ & \ddots & & \ddots & \ddots & \\ & & a_m & \cdots & a_1 & a_0 \\ b_n & \cdots & b_1 & b_0 & & \\ & \ddots & & \ddots & \ddots & \\ & & b_n & \cdots & b_1 & b_0 \end{bmatrix}. \tag{1.2.1}$$

Sylvester 矩阵是多项式代数中很重要的一类位移结构矩阵：

定理 1.2.1 *[94] Sylvester 矩阵 S 是一个拟 Toeplitz 矩阵，即：位移 $S - Z_{m+n} S Z_{m+n}^T$ 的秩为 2.*

Sylvester 矩阵与一元多项式最大公因子密切相关. 特别是，在文 [62] 中证明了，两个一元多项式最大公因子的系数可以从 Sylvester 矩阵 QR 分解中上三角矩阵 R 的最后一个非零行读出. 应用这一结果，文 [94] 中提出了一个求解一元近似最大公因子的快速算法，这一快速算法依赖于对 Sylvester 矩阵 QR 分解的快速实现，这里的 Sylvester 矩阵是给定多项式的 Sylvester 矩阵，或者是在变换 $x \to 1/x$ 下得到的互反多项式的 Sylvester 矩阵 [24]. 具体说来，基于 Sylvester 矩阵的拟 Toeplitz 矩阵结构，文 [94] 中应用稳定的广义 Schur 算法 [20] 来快速实现如下位移结构矩阵的三角分解

$$M_5 = \begin{bmatrix} S^T S & S^T \\ S & \mathbf{0} \end{bmatrix},$$

以得到 S 的 QR 分解，这里 S 代表一个 Sylvester 矩阵. 这样，整个近似最大公因子算法的复杂度依赖于实现 Sylvester 矩阵 QR 分解的复杂度因而为给定多项式次数和的平方次；这比应用经典的 QR 分解算法 (如：Householder 变换和 Givens 旋转) 的复杂度低一阶.

第二, Be$\acute{z}$out 矩阵和 Hankel 矩阵的应用. 在文 [13] 中, 作者把文 [12] 中关于 Hankel 矩阵的结果推广到了任一常数域和 B$\acute{e}$zout 矩阵的情形, 用两个多项式的 B$\acute{e}$zout 矩阵 $B(a,b)$ 给出了最大公因子计算的矩阵表示形式, 证明了最大公因子可以由 $JB(a,b)J$ 的核 (kernel) 中的一个向量来确定, 其中 J 代表一个置换矩阵, 其反对角线元素为 1, 其他元素为零. 基于以上想法及 B$\acute{e}$zout 矩阵的位移结构, 文 [13] 中设计出了一个求解最大公因子的并行算法, 需要 $O(\log_n^2)$ 并行步, $n^2 p_{\mathbf{F}}(n) q_{\mathbf{F}}(n)/\log n$ 的处理器, 其中 n 代表两个多项式次数中较大的次数, 如果域 $\mathbf{F}$ 支持快速傅立叶变换 FFT, 那么 $p_{\mathbf{F}}(n)=1$, 否则 $p_{\mathbf{F}}(n)=\log\log n$; 如果域 $\mathbf{F}$ 中可以对 $n!$ 做除法, 那么 $q_{\mathbf{F}}(n)=1$, 否则, $q_{\mathbf{F}}(n)=n$.

第三, Pad$\acute{e}$ 逼近的应用. 给出两个次数分别为 m 和 n 的一元多项式 $a(x)$ 和 $b(x)$, 其最大公因子 $g(x)$ 满足等式:

$$g(x)=\frac{a(x)}{\omega(x)}=\frac{b(x)}{z(x)}, \tag{1.2.2}$$

其中 $(\omega(x),z(x))$ 是幂级数 $h(x)=\sum_{i=0}^{\infty}h_i x^i=\frac{a(x)}{b(x)}$ 的 (m,n) Pad$\acute{e}$ 逼近. 这里, 不失一般性的, 可以假设 $b(0)\neq 0$. 基于以上结果, 文 [73, 76] 提出, $a(x)$ 和 $b(x)$ 的最大公因子可以通过计算如下三个多项式得到:

1. $h_N(x)=h(x) \bmod x^{N+1}$, $N=m+n$,
2. $\omega(x)$ 和 $z(x)$, 即 $h_N(x)$ 的 (m,n) Pad$\acute{e}$ 逼近,
3. (1.2.2) 中的 $g(x)$.

给定一个正数 ε, 应用上述想法计算 $a(x)$ 和 $b(x)$ 的 ε- 最大公因子时, 需要对第 2, 3 步做一些修改 [73, 76], 这样得到的算法类似于求解近似最大公因子的子结式方法 [75]. 当应用扩展欧几里德算法来计算 Pad$\acute{e}$ 逼近时, 整个算法的复杂度为 $O(n\log^2 n)$[17, 14, 75]. 计算 Pad$\acute{e}$ 逼近比较稳定的方法是应用其矩阵表示形式, 即求解一个 Toeplitz 矩阵为系数阵的线性方程组 [17, 14]; 这样的方程组可以根据 [77] 中相应的快速算法来求解.

此外, 我们知道在有限精度计算下, 多项式形式的扩展欧几里德 (快速) 算法是数值不稳定的. 在文 [77] 中, 作者把多项式形式的扩展欧几里德算法转化成了以子结式矩阵为系数阵的结构矩阵线性方程组的 (快速) 求解问题, 从而在一定程度上保证了数值稳定性. 而在文 [13] 中作者把扩展欧几里德算法转化成了一个关于一元 B$\acute{e}$zout

矩阵的结构矩阵问题，从数值稳定的角度说，这一算法更有优越性 [77].

关于结构矩阵在一元多项式代数中更广的应用可参见文献 [52, 75, 77].

§1.3 本文的研究工作

本文围绕近似最大公因子的求解问题，着重对具有 Sylvester 型结构的位移结构矩阵问题，做了如下研究：

1. 在第三章中, 我们提出了一个求解 Sylvester 矩阵数值秩的快速算法，并讨论其在计算 Sylvester 矩阵的奇异值方面的应用. Sylvester 矩阵的数值秩与一元近似最大公因子的次数上界密切相关 [23]. 计算 Sylvester 矩阵的数值秩最稳定的方法是计算 Sylvester 矩阵的奇异值分解；但这样做的代价是需要 $O((m+n))^3$ 的运算量，这里 $m+n$ 代表 Sylvester 矩阵的阶数. 基于 Sylvester 矩阵的位移结构，我们在本章提出了用广义 Schur 算法实现快速 Cholesky 分解来计算 S^TS 的数值秩，进而得到 S 的数值秩的快速算法. 这一算法具有复杂度 $O((m+n))^2$, 相比于奇异值分解来说复杂度低一阶. 本章的工作如下：

(1) *引入一类 Hankel 型 Sylvester 矩阵，其与 Sylvester 矩阵有相似的性质. 并且找到一个构造 Sylvester 矩阵和 Hankel 型 Sylvester 矩阵的技巧，使得在很大程度上可以避免广义 Schur 算法的过早停止.*

(2) *对基于广义 Schur 算法的快速 Cholesky 分解做了向后误差分析. 这为应用 Cholesky 分解求解数值秩的算法做了必要的准备.*

(3) *根据 Raleigh 商矩阵的理论，应用截断的 Cholesky 因子来计算 Sylvester 矩阵的某些奇异值得到了小的相对误差.*

(4) *在 Maple 软件上实现了本章所提出的快速算法.*

2. 在第四章中，我们用快速算法求解了 Sylvester 矩阵的结构低秩逼近问题. 我们的工作是用快速算法实现了文 [53] 中所给算法的核心部分内容，即求解关于位移结构矩阵的最小二乘问题。忽略位移结构，用经典的算法求解这一问题需要复杂度为 $O(st^2)$, 这里 s 和 t 分别代表最小二乘问题系数阵的行数和列数: 设给出的多项式的次数分别为 m 和 n, 那么 $s=2m+2n-k+3$, $t=2m+2n-2k+3$. 通过分析最小二乘问题系数阵的位移结构，应用其低秩的生成子矩阵，我们用平方阶的复杂度 $O(s^2)$ 求

解了这一问题. 本章的工作如下：

(1) 分析最小二乘问题系数阵的位移结构, 并构造在有限精度计算下具有小的误差的生成子矩阵.

(2) 提出最小二乘问题的快速解法, 分析算法的合理性, 并对算法做了向前误差分析.

(3) 在 *Maple* 软件上实现了本章所提出的快速算法, 并与文 [53] 中经典算法的结果进行了比较.

3. 在第五章中, 我们对本文做了简单的总结, 并对今后的工作做了展望.

第二章　预备知识

本章简单介绍了广义 Schur 算法, 及其与本文相关的一些基本定义和定理. 广义 Schur 算法有着非常丰富的应用，例如它可以快速实现具有位移结构的 Hermitian 矩阵和非 Hermitian 矩阵的三角分解 [52]. 为简便起见，这里只介绍本文所需要的如何应用广义 Schur 算法来实现对称结构矩阵的三角分解，这一对称结构矩阵满足形如 (1.1.3) 的位移方程，并且其中的三角矩阵 Z 是严格下三角的.

§2.1　广义 Schur 算法

设一个 $n\times n$ 实对称位移结构矩阵 R 是**强正规**的，即：R 的所有顺序主子矩阵都是满秩的，并且 R 满足如下的位移方程：

$$R - ZRZ^T = G\begin{bmatrix} I_p & \mathbf{0} \\ \mathbf{0} & -I_q \end{bmatrix} G^T, p+q=r, Z \text{ 是严格下三角的}, \tag{2.1.1}$$

那么应用如下定理描述的广义 Schur 算法可以得到 R 的形如 LDL^T 的三角分解，这里 L 是一个下三角矩阵，D 是一个符号矩阵.

定理 2.1.1 *[52] (广义 Schur 算法) 给出一个 $n\times n$ 的强正规矩阵 $R\in\mathbb{R}^{n\times n}$，其满足位移方程 (2.1.1). 那么 R 关于其 $i-1$ 阶顺序主子矩阵的 Schur 补 S_i 也都是结构矩阵，并且 S_i 满足如下的位移方程*

$$S_i - Z^{(i)}S_iZ^{(i)^T} = G_iJG_i^T, \tag{2.1.2}$$

其中 $Z^{(i)}$ 是从 $Z^{(i-1)}$ 中去掉第一行和第一列后得到的 $(n-i+1)\times(n-i+1)$ 子矩阵，而生成子矩阵 G_i 由下面过程递归构造：设 $S_1=R$, $G_1=G$, $Z^{(1)}=Z$, 对 $i\geq 1$ 进行递归.

(1) 设在第 i 步我们有 G_i 和 $Z^{(i)}$. 设 g_i 代表 G_i 的第一行.

(2) 选择一个 J- 正交矩阵 Θ_i，它使得 $g_i\Theta_i$ 中只含有一个非零元素 δ_i: 当 $g_iJg_i^T>0$ 时, 我们称为正步，δ_i 出现在前面的 p 个位置中；当 $g_iJg_i^T<0$ 时，我们称为负

步, δ_i 出现在后面的 q 个位置中 (R 的强正规性保证了 $g_iJg_i^T=0$ 的情形不会出现). 现不妨设 δ_i 出现在第 j 个位置:

$$g_i\Theta_i = [\ 0\ \cdots 0\ \underset{(j)}{\delta_i}\ 0\ \cdots 0\ ]. \tag{2.1.3}$$

(3) 按照如下步骤计算 G_{i+1}:

1. 用 Θ_i 去乘 G_i;
2. 对于 $G_i\Theta_i$, 除了第 j 列外, 其他列保持不变;
3. 用 $\Phi_i = Z^{(i)}$ 乘以 $G_i\Theta_i$ 的第 j 列;
4. 这样计算得到的就是 G_{i+1}.

用矩阵来描述上述过程如下:

$$\begin{bmatrix} \mathbf{0} \\ G_{i+1} \end{bmatrix} = \Phi_i G_i \Theta_i \begin{bmatrix} \mathbf{0}_{j-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0}_{r-j} \end{bmatrix} + G_i\Theta_i \begin{bmatrix} I_{j-1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I_{r-j} \end{bmatrix}. \tag{2.1.4}$$

(4) 三角分解 $R = LDL^T$ 的因子由下式确定: 设 l_i 和 d_i 分别代表 L 的第 i 列和 D 的第 i 个对角元, 那么

$$l_i = \begin{bmatrix} \mathbf{0} \\ \bar{l}_i \end{bmatrix}, \quad \bar{l}_i = G_i\Theta_i \begin{bmatrix} \mathbf{0} \\ 1 \\ \mathbf{0} \end{bmatrix} (j), \quad d_i = J_{jj}. \tag{2.1.5}$$

在完成了 n 步上述递归步后, 广义 Schur 算法得到了三角分解:

$$R = LDL^T = \sum_{i=1}^{n} d_i l_i l_i^T, \tag{2.1.6}$$

这一过程需要的浮点运算量为 $O(rn^2)$. 相对应的, 应用高斯消元法则需要 $O(n^3)$ 的运算量. 可见, 当 r 是 R 的远比 n 小得多的位移秩时, 广义 Schur 算法实现三角分解的运算量比高斯消元法低一个数量级. 当 R 是正定矩阵时, LDL^T 分解即成为 Cholesky 分解. 在应用广义 Schur 算法来计算 Cholesky 分解时, 不会遇到负步, 这是因为 R 的正定性保证了定理 2.1.1 中 (2) 的 $g_iJg_i^T<0$ 的情形不会出现.

定理 2.1.1 中由第 i 个生成子矩阵 G_i 化成的矩阵 $G_i\Theta_i$，叫做 G_i 的**合适形式** (proper form)，我们把它记做 $\bar{G}_i$；相应的，把 $g_i\Theta_i$ (2.1.3) 叫做 g_i 的合适形式. 注意到 J- 正交矩阵 Θ_i 满足：$\Theta_i J\Theta_i^T = J$, 我们有

$$\bar{G}_i J\bar{G}_i^T = G_i\Theta_i J\Theta_i^T G_i^T = G_i J G_i^T, \tag{2.1.7}$$

所以 $\bar{G}_i$ 也是第 i 个 Schur 补 S_i 的生成子矩阵. 这样的 J- 正交矩阵 Θ_i 总可以保证存在，关于 J- 正交矩阵的构造详见文献 [52] 中的第 4.4 节.

不失一般性，我们可以选择应用这样的 J- 正交矩阵 Θ_i, 它使得 (2) 中 $g_i\Theta_i$ 的非零元素 δ_i 在 $g_iJg_i^T > 0$ 时出现在 $g_i\Theta_i$ 的第一个位置；在 $g_iJg_i^T < 0$ 时出现在 $g_i\Theta_i$ 的第 r 个位置. 需要注意的是，在实际应用广义 Schur 算法时，有限精度下的计算使得等式 (2.1.7) 并不能严格成立；我们只能尽可能地使其在一个小的误差范围下成立. 如文 [19, 20] 中所述，通过采取适当的方法去构造和应用 J- 正交变换 Θ_i 可以保证这一点. 为此我们循文 [20] 中的做法构造 Θ_i 为两个正交变换和一个初等 Hyperbolic 旋转的复合. 具体来说，在实现正步时，

(1) 构造一个正交变换 $\Theta_{i,1}$ (如 *Householder* 变换)，将 G_i 的前 p 列变换到合适形式，即：$g_i\begin{bmatrix}\Theta_{i,1} & \mathbf{0}\\ \mathbf{0} & I_q\end{bmatrix}$ 的前 p 个元素中只有一个非零元素，并出现在第一个位置.

(2) 构造一个正交变换 $\Theta_{i,2}$ (如 *Householder* 变换)，将 G_i 的后 q 列变换到合适形式，即：$g_i\begin{bmatrix}\Theta_{i,1} & \mathbf{0}\\ \mathbf{0} & I_q\end{bmatrix}\begin{bmatrix}I_p & \mathbf{0}\\ \mathbf{0} & \Theta_{i,2}\end{bmatrix}$ 的后 q 个元素中只有一个非零元素，并出现在最后一个位置.

(3) 应用一个初等 *Hyperbolic* 旋转 $\Theta_{i,3}$ 消去 $g_i\begin{bmatrix}\Theta_{i,1} & \mathbf{0}\\ \mathbf{0} & I_q\end{bmatrix}\begin{bmatrix}I_p & \mathbf{0}\\ \mathbf{0} & \Theta_{i,2}\end{bmatrix}$ 中最后一个位置上的非零元素.

(4) $\Theta_{i,1}$, $\Theta_{i,2}$ 和 $\Theta_{i,3}$ 的复合最终把 G_i 变化成了合适形式：

$$\bar{G}_i = G_i\begin{bmatrix}\Theta_{i,1} & \mathbf{0}\\ \mathbf{0} & I_q\end{bmatrix}\begin{bmatrix}I_p & \mathbf{0}\\ \mathbf{0} & \Theta_{i,2}\end{bmatrix}\Theta_{i,3}. \tag{2.1.8}$$

负步的实现具有完全类似的形式，只需要把 (3) 改为：

(3) 应用一个初等 *Hyperbolic* 旋转 $\Theta_{i,3}$ 消去 $g_i \begin{bmatrix} \Theta_{i,1} & \mathbf{0} \\ \mathbf{0} & I_q \end{bmatrix} \begin{bmatrix} I_p & \mathbf{0} \\ \mathbf{0} & \Theta_{i,2} \end{bmatrix}$ 中第一个位置上的非零元素.

一个 2×2 初等 Hyperbolic 旋转 Θ 把一个实的行向量 $[\ a\ \ b\ ]$ 变换成：

$$[\ a\ \ b\ ]\,\Theta = \left[\ \pm\sqrt{a^2-b^2}\ \ 0\ \right] \text{ 如果 } |a|>|b|, \tag{2.1.9}$$

$$[\ a\ \ b\ ]\,\Theta = \left[\ 0\ \ \pm\sqrt{b^2-a^2}\ \right] \text{ 如果 } |a|<|b|. \tag{2.1.10}$$

设 $[\ a'\ \ b'\ ]$ 代表 (2.1.9) 和 (2.1.10) 中由 $[\ a\ \ b\ ]$ 变换后的行向量，那么在这两种情况下都有：

$$[\ a'\ \ b'\ ]\,J\,[\ a'\ \ b'\ ]^T = [\ a\ \ b\ ]\,J\,[\ a\ \ b\ ]^T, \tag{2.1.11}$$

其中 $J=\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$. 如下构造的 J- 正交变换 $\Theta(\Theta J\Theta^T=J)$ 就是一个初等 Hyperbolic 旋转变换，它把行向量 $[\ a\ \ b\ ]$ 分别变换成 (2.1.9) 或 (2.1.10) 中等式右端的行向量:

$$\Theta = \frac{1}{\sqrt{1-\rho^2}}\begin{bmatrix} 1 & -\rho \\ -\rho & 1 \end{bmatrix},\ \text{这里 } \rho=\frac{b}{a},\ a\neq 0,\ |a|>|b|, \tag{2.1.12}$$

$$\Theta = \frac{1}{\sqrt{1-\rho^2}}\begin{bmatrix} 1 & -\rho \\ -\rho & 1 \end{bmatrix},\ \text{这里 } \rho=\frac{a}{b},\ a\neq 0,\ |a|<|b|. \tag{2.1.13}$$

要保证等式 (2.1.7) 在一个小的误差意义下成立，直接应用初等 Hyperbolic 旋转是不明智的做法. 这主要是因为有限精度计算中 $\|\Theta_{i,3}\|$ 可能会很大 [11, 19, 20]. 关于应用初等 Hyperbolic 旋转的稳定方法，可以采用 [19] 中提出的 OD 步骤 (OD-procedure). 这一实现方法数值稳定的特性是基于，将应用一个初等 Hyperbolic 旋转通过一系列正交矩阵和对角矩阵的作用来实现. 这依赖于任何一个形如 (2.1.12) 的 Hyperbolic 旋转都有如下的特征分解：

$$\Theta = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}\begin{bmatrix} \sqrt{\frac{a+b}{a-b}} & 0 \\ 0 & \sqrt{\frac{a-b}{a+b}} \end{bmatrix}\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}\frac{1}{\sqrt{2}}, \tag{2.1.14}$$

其中，矩阵

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix} \tag{2.1.15}$$

是正交矩阵. 具体的说，用 OD 步骤来实现形如 (2.1.12) 的 Hyperbolic 旋转 Θ 可由下面算法描述. 关于形如 (2.1.13) 的 Hyperbolic 旋转有类似的实现方法.

算法 2.1.1 (OD 步骤) [19] 给出一个形如 (2.1.12) 的 Hyperbolic 旋转 Θ 和一个初始行向量 $[\ x\ y\ ]$. 设 $[\ x'''\ y'''\ ]$ 是由 $[\ x\ y\ ]$ 经 Hyperbolic 旋转 Θ 作用后得到的行向量，那么 $[\ x'''\ y'''\ ]$ 可以按如下步骤计算：

$$[\ x'\ y'\ ] \leftarrow [\ x\ y\ ]\begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix},$$

$$[\ x''\ y''\ ] \leftarrow [\ x'\ y'\ ]\begin{bmatrix} \frac{1}{2}\sqrt{\frac{a+b}{a-b}} & 0 \\ 0 & \frac{1}{2}\sqrt{\frac{a-b}{a+b}} \end{bmatrix},$$

$$[\ x'''\ y'''\ ] \leftarrow [\ x''\ y''\ ]\begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}.$$

关于稳定地应用初等 Hyperbolic 旋转的方法，除了上述 OD 步骤以外还可以参见文 [7, 11] 中提出的混合 - 递进 (mixed-downdating) 步骤, 或者 [19] 中提出的 H 步骤 (H-procedure).

注 2.1.2 定理 *2.1.1* 构造性地证明了，一个结构矩阵的 *Schur* 补仍然是结构矩阵. 应用广义 *Schur* 算法可以递归地求出每个 *Schur* 补的生成子矩阵，进而得到原矩阵的三角分解因子.

注 2.1.3 从定理 *2.1.1* 可以看出，两个 *Schur* 补 S_i 和 S_{i+1} 可以通过 *Schur* 约化步联系起来 *[20]*:

$$S_i - d_i \bar{l}_i \bar{l}_i^T = \begin{bmatrix} 0 & \mathbf{0} \\ \mathbf{0} & S_{i+1} \end{bmatrix}, \tag{2.1.16}$$

其中 $\bar{l}_i$ 代表 l_i 的非零部分.

§2.2 基本定义和定理

1. 基本定义

关于一元多项式近似最大公因子的计算, 已有比较丰富的研究结果 [86, 68, 23, 54, 29, 41, 55, 18, 76, 24, 93, 53]. 在这些工作中对于近似最大公因子的定义主要有以下几种. 首先是 A.Schönhage 于 1985 年 [86] 提出的拟最大公因子 (quasi-GCD) 的概念及其求解算法. 这一算法假设输入多项式的系数的精度可以随着对输出结果的准确度的要求而无限增长. 其次是 N. Kamarkar 和 Y. N. Lakshman 把近似最大公因子问题归结为优化问题, 于 1996 年 [54] 提出的伪最大公因子的概念. 那么与这两者不同, 本文中我们假设给出的两个一元多项式 $a(x)$, $b(x)$, 其系数具有有限精度, 而多项式 $a(x)$ 和 $b(x)$ 的近似最大公因子, 指的是这两个多项式的 ε- 最大公因子 [23], 其中 ε 是一个给定的正数.

定义 2.2.1 *[23, 91]* 给定一个正数 ε, 我们称 $d(x)$ 是 $a(x)$ 和 $b(x)$ 的 ε- 最大公因子, 如果它是满足

$$d = \gcd(a^*, b^*),\ \|a - a^*\|_2 \leq \varepsilon,\ \|b - b^*\|_2 \leq \varepsilon \tag{2.2.1}$$

的次数最高的多项式, 其中 a^* 和 b^* 分别是 次数不超过 $\deg(a(x))$ 和 $\deg(b(x))$ 的多项式.

根据上面的定义, 两个多项式的近似最大公因子是在其扰动容许范围内 (即 ε 邻域内) 的两个多项式的准确最大公因子. 同时, 两个多项式的 ε- 最大公因子不是唯一的.

例 2.2.2 给出一元多项式 $a(x) = x^2 + 3.999x + 4$ 和 $b(x) = x + 2$, 及扰动容许度 $\varepsilon = 0.001$. 可以看出, 扰动多项式 $a(x)$ 为: $a + \Delta a = a(x) + 0.001x = x^2 + 4.000x + 4$, 我们得到 $\gcd(a + \Delta a, b) = x + 2$. 于是, $x + 2$ 是 $a(x)$ 和 $b(x)$ 的一个 ε- 最大公因子.

2. 基本定理

为了后文的需要, 我们简要介绍矩阵计算中奇异值分解的基本知识.

定理 2.2.3 *[30]* 设 A 为一实 $m \times n$ 矩阵，则必存在正交矩阵

$$U = [\mathbf{u}_1, \ldots, \mathbf{u}_m] \in \mathbb{R}^{m\times m}, V = [\mathbf{v}_1, \ldots, \mathbf{v}_n] \in \mathbb{R}^{n\times n},$$

使得

$$U^T AV = \mathrm{diag}(\sigma_1, \ldots, \sigma_p) = \Sigma \in \mathbb{R}^{m\times n}, p = \min\{m, n\},$$

其中 $\sigma_1 \geq \cdots \geq \sigma_p \geq 0$.

定理中的 σ_i 称为 A 的第 i 大的奇异值, 相应的, $\mathbf{u}_i$ 和 $\mathbf{v}_i$ 分别称为 A 的第 i 个左奇异向量和第 i 个右奇异向量, $1 \leq i \leq \min\{m, n\}$. 通过比较 $AV = U\Sigma$ 和 $A^T U = V\Sigma^T$ 的对应列可以得到, 对 $1 \leq i \leq \min\{m, n\}$ 有,

$$A\mathbf{v}_i = \sigma_i \mathbf{u}_i,\ A^T\mathbf{u}_i = \sigma_i \mathbf{v}_i.$$

进一步的，如果 $\sigma_1 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_p = 0$, 那么 A 的秩为 r. 这里 A 的秩指的是 A 的值域空间的维数，即

$$\mathrm{rank}(A) := \dim(\mathrm{span}\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}),$$

其中 $A = [\mathbf{a}_1, \ldots, \mathbf{a}_n]$ 是 A 的一个列分划.

设 $A \in \mathbb{R}^{m\times n}$, 那么

$$\begin{aligned} \|A\|_F^2 &= \sigma_1^2 + \cdots + \sigma_p^2,\ p = \min\{m, n\}, \\ \|A\|_2 &= \sigma_1, \\ \min_{\mathbf{x}\neq 0} \frac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_2} &= \sigma_n \quad (m \geq n). \end{aligned}$$

奇异值分解最有价值的一个方面是它可以帮我们很好地处理矩阵数值秩的概念. 在数值计算中，人们通常关心矩阵的 ε- 秩. 给定一个矩阵 A, 对给定的正数 ε, 通常定义矩阵 A 的 ε- 秩为

$$\mathrm{rank}(A, \varepsilon) := \min_{\|A-B\|_2 \leq \varepsilon} \mathrm{rank}(B).$$

矩阵的数值 ε- 秩可以由奇异值分解很好地刻画, 这是因为奇异值可以表明一个给定矩阵与比其秩低的矩阵的靠近程度：

定理 2.2.4 *[30] 设 $A \in \mathbb{R}^{m\times n}$ 的奇异值分解由定理 2.2.3 给出. 如果 $k < r = \operatorname{rank}(A)$, 且*

$$A_k = \sum_{i=1}^{k} \sigma_i \mathbf{u}_i \mathbf{v}_i^T,$$

那么

$$\min_{\operatorname{rank}(B)=k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}.$$

根据定理 2.2.4, 如果 $r_\varepsilon = \operatorname{rank}(A, \varepsilon)$, 那么

$$\sigma_1 \geq \cdots \geq \sigma_{r_\varepsilon} > \varepsilon \geq \sigma_{r_\varepsilon+1} \geq \cdots \geq \sigma_p,\ p = \min\{m, n\}.$$

第三章　求解 Sylvester 矩阵数值秩的一个快速算法

本章提出了一个求解 Sylvester 矩阵数值秩的快速算法. 这一快速算法基于 S^TS 或 H^TH 的快速 Cholesky 分解，其中，S 和 H 分别代表 Sylvester 矩阵和一个 Hankel 型的 Sylvester 矩阵. 而快速 Cholesky 分解的实现是利用广义 Schur 算法 [19]. 算法复杂度为 $O((m+n)^2)$, 这里 $m+n$ 代表 Sylvester 矩阵的阶数. 数值结果表明，这一快速算法对于求解那些具有低的亏秩的 Sylvester 矩阵的数值秩是比较有效的.

§3.1　引　言

设一元多项式 $a(x), b(x) \in \mathbb{R}[x]$, $a(x) = a_m x^m + a_{m-1}x^{m-1} + \cdots + a_1 x + a_0, a_m \neq 0$, $b(x) = b_n x^n + b_{n-1}x^{n-1} + \cdots + b_1 x + b_0, b_n \neq 0$. $a(x)$ 和 $b(x)$ 的 Sylvester 矩阵是 $S(a,b) \in \mathbb{R}^{(m+n)\times(m+n)}$:

$$S(a,b) = \underbrace{\begin{bmatrix} a_m & & & & b_n & & & \\ a_{m-1} & a_m & & & b_{n-1} & b_n & & \\ \vdots & a_{m-1} & \ddots & & \vdots & b_{n-1} & \ddots & \\ a_1 & \vdots & \ddots & a_m & b_1 & \vdots & \ddots & b_n \\ a_0 & a_1 & & a_{m-1} & b_0 & b_1 & & b_{n-1} \\ & a_0 & \ddots & \vdots & & b_0 & \ddots & \vdots \\ & & \ddots & a_1 & & & \ddots & b_1 \\ & & & a_0 & & & & b_0 \end{bmatrix}}_{n \qquad\qquad m}. \tag{3.1.1}$$

假设给出的两个多项式 $a(x)$ 和 $b(x)$ 的系数是具有有限精度的, 并给定一个正数 ε，它规定了这两个多项式的扰动容许范围. 计算两个多项式的 $\varepsilon-$ 最大公因子是近十几年符号数值混合计算中一个热门的研究课题. 有多个文献从不同的角度提出了 $\varepsilon-$ 最大公因子的算法，如 [23, 29, 94, 93, 53, 24]. 对于这些算法, 预先得到 $\varepsilon-$ 最大公因子次数

的信息是很重要的. 下面的引理表明, Sylvester 矩阵的 ε- 秩与 $\varepsilon-$ 最大公因子的次数上界密切相关.

引理 3.1.1 *[23] 设 $\sigma_i, i=1,\ldots,m+n$ 是 Sylvester 矩阵 $S(a,b)$ 的奇异值, 并且 $\sigma_1 \geq \cdots \geq \sigma_r > \varepsilon\sqrt{m+n} > \varepsilon \geq \sigma_{r+1} \geq \cdots \geq \sigma_{m+n}$ (换句话说, Sylvester 矩阵的数值 $\varepsilon-$ 秩为 r), 那么如果 $\hat{d}$ 是 $a+\triangle a$ 和 $b+\triangle b$ 的最大公因子, 并且 $\deg(\hat{d}) \geq m+n-r+1$, 那么或者有 $\|\triangle a\|_2 > \varepsilon$ 或者有 $\|\triangle b\|_2 > \varepsilon$ 成立, 这里 $\|\cdot\|_2$ 代表多项式的系数向量 2-范数.*

对于计算一个矩阵的数值 $\varepsilon-$ 秩, 矩阵的奇异值分解无疑是最可靠的方法; 但这样做的代价是复杂度比较高. 比如应用到 Sylvester 矩阵上时, 所需的浮点运算量为 $O((m+n)^3)$. 在文 [63] 中提出了一个部分奇异值分解的算法, 这一算法是牛顿迭代的一个巧妙应用. 给出一个矩阵和一个奇异值的临界值 δ, 这一算法可以计算出来小于 δ 的那些奇异值以及这些奇异值所对应的奇异向量. 当应用到一个数值秩为 r 的 Sylvester 矩阵上时, 如果跳跃 σ_r/σ_{r+1} 是明显的 (如：10^3), 那么通常在三次迭代后, 这一算法可以达到其作用. 然而, 由于这一算法在初始步时需要对输入的矩阵做一个满的 QR 分解, 使得整个算法的复杂度仍为输入矩阵阶数的立方次.

鉴于 Sylvester 矩阵是一个拟 Toeplitz 矩阵 [94], 本章提出了一个求解 Sylvester 矩阵数值秩的快速算法，算法复杂度为 $O((m+n)^2)$. 这一快速算法依赖于对 $T=S^TS$ 或 H^TH 的快速 Cholesky 分解，其中 H 是一个 Hankel 型的 Sylvester 矩阵. 我们引入 Hankel 型的 Sylvester 矩阵是为了在不需要选主元的情况下, 仍然可以得到满意的 Cholesky 分解. 而快速 Cholesky 分解的实现，是通过应用一个修正的数值稳定的广义 Schur 算法 [19, 20].

具体说来，我们旨在通过实现 T 的 Cholesky 分解来求解 S^TS 的数值 ε^2- 秩进而得到 Sylvester 矩阵的 $\varepsilon-$ 秩. 这一想法来源于一个基本的事实：Sylvester 矩阵的数值 $\varepsilon-$ 秩与 S^TS 的数值 ε^2- 秩是相等的. 现假设 $\hat{R}_i$ 是完成 i 步广义 Schur 算法后得到的一个 Cholesky 因子,

$$\hat{R}_i = \begin{bmatrix} \hat{R}_{11}^{(i)} & \hat{R}_{12}^{(i)} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

这里 $\hat{R}_{11}^{(i)}$ 是一个 $i\times i$ 上三角矩阵. 那么当 $\|T-\hat{R}_r^T\hat{R}_r\|_2$ 足够小时, 我们称 r 是 Sylvester 矩阵的备选的数值 $\varepsilon-$ 秩.

循文献 [11, 19, 20, 45, 46] 中误差分析的技巧，我们对于快速 Cholesky 分解进行了向后误差分析. 这为我们在计算 Sylvester 矩阵的数值秩时设定 Cholesky 分解的停止准则做了必要的准备.

本章其余各节安排如下: §3.2 我们分析了 S^TS 和 H^TH 的位移结构, 此外，通过讨论半正定矩阵 S^TS 和 H^TH 的顺序主子矩阵的性质, 找到了一个构造 Sylvester 矩阵和 Hankel 型 Sylvester 矩阵的技巧，使得在一定程度上可以避免广义 Schur 算法的过早停止; §3.3 我们进行一些误差分析；§3.4 给出求解 Sylvester 矩阵数值秩的快速算法；§3.5 给出数值实验结果.

§3.2 避免广义 Schur 算法的过早停止

我们已经知道，Sylvester 矩阵 S 是一个拟 Toeplitz 矩阵的位移结构矩阵. 在下面的定理中，我们揭示了 S^TS 的位移结构.

定理 3.2.1 *[94]* S^TS 关于位移 $S^TS - ZS^TSZ^T$ 的位移秩不超过 4，这里

$$Z = \mathrm{diag}(Z_n, Z_m).$$

事实上，我们可以构造生成子对 (G, J), $J = \begin{bmatrix} I_2 & \mathbf{0} \\ \mathbf{0} & -I_2 \end{bmatrix}$, 使得

$$S^TS - ZS^TSZ^T = GJG^T, \tag{3.2.1}$$

这里 $G = [\mathbf{x}_1\ \mathbf{x}_2\ \mathbf{x}_3\ \mathbf{x}_4]$, 其中

$$\begin{aligned}
\mathbf{x}_1 &= S^TS(:,1)/\|S(:,1)\|, \\
\mathbf{x}_2 &= S^TS(:,n+1)/\|S(:,n+1)\|, \quad \mathbf{x}_2[1] = 0, \\
\mathbf{x}_3 &= \mathbf{x}_1, \quad \text{除了} \quad \mathbf{x}_3[1] = 0, \\
\mathbf{x}_4 &= \mathbf{x}_2, \quad \text{除了} \quad \mathbf{x}_4[n+1] = 0.
\end{aligned}$$

这样，我们可以把广义 Schur 算法直接作用于生成子对 (G, J) 来实现 S^TS 的快速 Cholesky 分解.

设一个 Sylvester 矩阵 S 的数值 $\varepsilon-$ 秩为 r. 当应用广义 Schur 算法来实现 S^TS 的 Cholesky 分解时，广义 Schur 算法却可能会由于 S^TS 的第一个 $r \times r$ 主子矩阵不是充分正定而不能顺利地完成前 r 个正步.

例 3.2.2 *考虑*

$$a(x) = 2x^2 + 3x - x^4 - 2x^3,$$
$$b(x) = 3x + 2 + x^2,$$

可以验证, Sylvester 矩阵 $S(a,b)$ 的秩为 5, 而 $S(a,b)^T S(a,b)$ 的第一个 5×5 主子矩阵的秩为 4. 当应用广义 Schur 算法来计算 $S(a,b)^T S(a,b)$ 的 Cholesky 分解时, 算法顺利完成了前四个正步却在实现第五步时中断.

对于上面的例子, 广义 Schur 算法提前中断的原因就是, $S(a,b)^T S(a,b)$ 的第一个 5×5 主子矩阵是秩亏的. 对于这样的半正定矩阵 $S(a,b)^T S(a,b)$, 可以应用矩阵计算理论中选主元的方法去实现 Cholesky 分解而得到 $R_5^T R_5$, 其中

$$R_5 = \begin{bmatrix} R_{11}^{(5)} & R_{12}^{(5)} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \tag{3.2.2}$$

这里 $R_{11}^{(5)}$ 是一个 5×5 满秩的上三角矩阵. 此时, $R_5^T R_5$ 是矩阵 $\Pi^T S(a,b)^T S(a,b)\Pi$ 的 Cholesky 分解, 其中 Π 是一个置换矩阵. 可以看出, 位移结构会因为选主元而被破坏掉, 因此本文中我们避免应用这一方法. 我们发现, 如果构造 Sylvester 矩阵为 $S(b,a)$, 那么 $S^T(b,a)S(b,a)$ 的第一个 5×5 主子矩阵是满秩的, 此时应用广义 Schur 算法, 前五个正步都顺利完成, 可以计算出 $S^T(b,a)S(b,a)$ 的形如 (3.2.2) 的 Cholesky 因子.

一般地, 我们有如下定理:

定理 3.2.3 *设 $S(a,b)$ 是一元 m 次多项式 $a(x)$ 和一元 n 次多项式 $b(x)$ 的 Sylvester 矩阵. 假定*

$$a(x) = x^p(a_m x^{m-p} + \cdots + a_p),\ a_p \neq 0,$$
$$b(x) = x^q(b_n x^{n-q} + \cdots + b_q),\ b_q \neq 0.$$

如果 $0 \le p \le q$ 并且 $\mathrm{rank}(S) = m+n-d$ 其中 $p \le d \le \min\{m,n\}$, 那么 $S(a,b)$ 的前 $m+n-d$ 个列向量必是列满秩的.

证明 注意到 $\mathrm{rank}(S) = m+n-d$ 蕴涵着 $\gcd(a,b)$ 的次数为 d. 假设 $S(a,b)$ 的前 $m+n-d$ 列不是列满秩的, 根据下面的等式

$$(x^{n-1}a, x^{n-2}a, \ldots, a,\ x^{m-1}b, \ldots, b) = (x^{m+n-1}, \ldots, x^m, \ldots, 1)S(a,b),$$

必存在不全为零的常数 s_i, t_j 使得

$$(s_{n-1}x^{n-1} + \cdots + s_0)a + (t_{m-1}x^{m-1} + \cdots + t_d x^d)b = 0. \tag{3.2.3}$$

设 $a' = a/\gcd(a,b)$, $b' = b/\gcd(a,b)$, 那么 $\gcd(a',b') = 1$, a' 和 b' 的次数分别为 $m-d$ 和 $n-d$. 同时，由 (3.2.3) 有

$$(s_{n-1}x^{n-1} + \cdots + s_0)a' + x^d(t_{m-1}x^{m-d-1} + \cdots + t_d)b' = 0. \tag{3.2.4}$$

由于 $p \le q$, 所以 $x \nmid a'$. 这样由 (3.2.4) 可以得到

$$a' \mid (t_{m-1}x^{m-d-1} + \cdots + t_d)b'.$$

注意到 $\gcd(a',b') = 1$, 我们有 $a'|(t_{m-1}x^{m-d-1} + \cdots + t_d)$, 而 $\deg(a') = m-d$, 所以 $t_j = 0, j = d, \ldots, m-1$. 进而由 (3.2.3) 得到，$s_i = 0, i = 0, \ldots, n-1$, 这与前面 s_i, t_j 不全为零的假设产生矛盾. 因此，$S(a,b)$ 的前 $m+n-d$ 列必是列满秩的. ■

基于上面的定理，为了避免广义 Schur 算法在实现 S^TS 的 Cholesky 分解过程中过早停止，我们构造 Sylvester 矩阵为 $S(a,b)$, 其中 $a(x)$ 相比于 $b(x)$ 有更少的近似为零的尾项. 但是，当给出的两个多项式都有比较小的尾项系数时，将很难判定哪一个多项式有更少的小尾项. 为了处理这样的多项式对，我们下面引入 Hankel 型的 Sylvester 矩阵.

1. Hankel 型的 Sylvester 矩阵

设 P_i 代表一个 $i \times i$ 置换矩阵：

$$P_i = \begin{bmatrix} & & 1 \\ & \iddots & \\ 1 & & \end{bmatrix},$$

这里 i 代表一个正整数. 所谓的 Hankel 型的 Sylvester 矩阵定义如下：

$$\begin{aligned} H(a,b) &= S(a,b)\ \mathrm{diag}\left(P_n, P_m\right) \\ &= \begin{bmatrix} & & & a_m & & & & b_n \\ & & \iddots & a_{m-1} & & & \iddots & b_{n-1} \\ & a_m & \iddots & \vdots & & b_n & \iddots & \vdots \\ a_m & a_{m-1} & & a_1 & b_n & b_{n-1} & & b_1 \\ a_{m-1} & \vdots & & a_0 & b_{n-1} & \vdots & & b_0 \\ \vdots & a_1 & \iddots & & \vdots & b_1 & \iddots & \\ a_1 & a_0 & & & b_1 & b_0 & & \\ a_0 & & & & b_0 & & & \end{bmatrix}. \\ & \qquad \underbrace{\qquad\qquad\qquad\qquad}_{n}\ \underbrace{\qquad\qquad\qquad\qquad}_{m} \end{aligned}$$

Hankel 型 Sylvester 矩阵与 Sylvester 矩阵有相似的性质.

命题 3.2.4 设 $H = U\Sigma V^T$ 是 $H(a,b)$ 的奇异值分解，这里 U 和 V 是两个正交矩阵，$\Sigma = \mathrm{diag}\left(\sigma_1(H), \sigma_2(H), \ldots, \sigma_{m+n}(H)\right)$ 是一个由 $H(a,b)$ 的奇异值所构成的对角矩阵，那么 $S = U\Sigma(PV)^T$ 是 $S(a,b)$ 的奇异值分解, 其中 $P = \mathrm{diag}\left(P_n, P_m\right)$.

证明 从 Hankel 型 Sylvester 矩阵的定义易证. ∎

推论 3.2.5 *Hankel* 型 *Sylvester* 矩阵 $H(a,b)$ 与 *Sylvester* 矩阵 $S(a,b)$ 有相同的秩.

定理 3.2.6 H^TH 关于位移 $H^TH - ZH^THZ^T$ 的位移秩不超过 4，这里

$$Z = \mathrm{diag}\left(Z_n, Z_m\right).$$

证明 我们可以构造一个生成子对 (G, J) 使得

$$H^TH - ZH^THZ^T = GJG^T,$$

这里 $J = \begin{bmatrix} I_2 & \mathbf{0} \\ \mathbf{0} & -I_2 \end{bmatrix}$, $G = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4]$, 其中

$$
\begin{aligned}
\mathbf{x}_1 &= H^T H(:,1)/\|H(:,1)\|, \\
\mathbf{x}_2 &= H^T H(:,n+1)/\|H(:,n+1)\|, \quad \mathbf{x}_2[1] = 0, \\
\mathbf{x}_3 &= \mathbf{x}_1, \quad \text{除了} \quad \mathbf{x}_3[1] = 0, \\
\mathbf{x}_4 &= \mathbf{x}_2, \quad \text{除了} \quad \mathbf{x}_4[n+1] = 0.
\end{aligned}
$$

■

定理 3.2.7 *设 $H(a,b)$ 是一元 m 次多项式 $a(x)$ 和一元 n 次多项式 $b(x)$ 的 Hankel 型 Sylvester 矩阵. 如果 $H(a,b)$ 的秩为 $m+n-d$, 这里 $0 < d \le \min\{m,n\}$, 那么 $H(a,b)$ 的前 $m+n-d$ 个列向量必是列满秩.*

证明 注意到 $H(a,b)$ 的秩为 $m+n-d$ 蕴涵着 $\gcd(a,b)$ 次数为 d. 假设 $H(a,b)$ 的前 $m+n-d$ 个列向量是线性相关的, 那么根据下列等式

$$
\begin{aligned}
&\left[a, x\,a, \ldots, x^{n-1}\,a,\ b, \ldots, x^{m-d-1}\,b, \ldots, x^{m-1}\,b\right] \\
= &\left[x^{m+n-1}, \ldots, x^m, \ldots, 1\right] H(a,b),
\end{aligned}
$$

必存在 $m+n-d$ 个不全为零的常数 s_i, t_j, 使得

$$
(s_0 + s_1 x + \cdots + s_{n-1}x^{n-1})a + (t_0 + t_1 x + \cdots + t_{m-d-1}x^{m-d-1})b = 0. \tag{3.2.5}
$$

设 $a' = a/\gcd\,(a,b)$, $b' = b/\gcd\,(a,b)$, 那么 $\gcd(a',b') = 1$,

$$
(s_0 + s_1 x + \cdots + s_{n-1}x^{n-1})a' + (t_0 + t_1 x + \cdots + t_{m-d-1}x^{m-d-1})b' = 0. \tag{3.2.6}
$$

于是,

$$
a' \mid (t_0 + t_1 x + \cdots + t_{m-d-1}x^{m-d-1})b'.
$$

注意到 $\gcd(a',b') = 1$, 我们得到 $a' \mid (t_0 + t_1 x + \cdots + t_{m-d-1}x^{m-d-1})$, 而 $\deg(a') = m-d$, 所以 $t_j = 0, j = 0, \ldots, m-d-1$. 同时, 从 (3.2.5) 我们又可以得到 $s_i = 0, i = 0, \ldots, n-1$, 这样就与前面 s_i, t_j 不全为零的假设产生矛盾. 因此, $H(a,b)$ 的前 $m+n-d$ 列必是列满秩的. ■

这一定理告诉我们，对于秩为 r 的 Hankel 型 Sylvester 矩阵 H, H^TH 的第一个 $r \times r$ 主子矩阵总是满秩的. 我们可以直接应用广义 Schur 算法来实现 H^TH 的 Cholesky 分解. 尽管如此，我们在实际应用时，对于 Hankel 型 Sylvester 矩阵的构造并不是随意的；我们总是构造 Hankel 型 Sylvester 矩阵为 $H(a,b)$, 其中 a_m 占有比较大的权重，即 $|a_m|/\|a\|_2$ 比较大. 事实上，从定理 3.2.7 的证明中可以看出来，当多项式 $a(x)$ 首项系数很小时，多项式 a' 的次数可以近似地小于等于 $m-d-1$, 根据 (3.2.6), 这样会使得 $H(a,b)$ 的前 $m+n-d$ 列近似线性相关.

注 3.2.8 *给出两个多项式 $a(x)$ 和 $b(x)$, 对于形如 (1.2.1) 的 Sylvester 矩阵 S, 也有类似于定理 3.2.7 的结论成立 (见附录一的定理 A.0.1). 这样, 对于秩为 r 的 S, S^TS 的 r 阶顺序主子矩阵是正定的. 但在有限精度计算下，文献 [24] 中的结果告诉我们，当给出的两个多项式有绝对值大于 1 的公共根时，应用数值计算中经典的 QR 分解算法无法得到关于 S 的秩的正确信息. 详细讨论可见文献 [24, 91, 94]. 数值测试表明，应用 S^TS 的 Cholesky 分解通常遇到类似的问题；相比较而言，应用形如 (3.1.1) 的 Sylvester 矩阵和 Hankel 型 Sylvester 矩阵来做 Cholesky 分解则好得多. 这是本文引入 Hankel 型 Sylvester 矩阵的一个原因. 下面是一个简单的例子：*

例 ***3.2.9*** *[94]*

$$\begin{aligned} a(x) &= 0.02077971692\, x^{13} \;+\; 0.09350872615\, x^{12} + \cdots + 0.1655883694\, x^2 \\ &\quad -0.09935302169x + 0.000000006520463574, \\ b(x) &= 0.03804013712\, x^{11} - \; 0.1616705828\, x^{10} + \cdots + 0.5663225421\, x^2 \\ &\quad + 0.03233411610\, x \;- 0.1940046900, \end{aligned}$$

这里 $\|a\|_2 = \|b\|_2 = 1$. 设定 Digits $= 16$, 用 Maple 9.5 算得 Sylvester 矩阵的奇异值：

$$\ldots, 0.037,\ 0.029,\ \mathbf{0.42e-8},\ 0.22e-8,\ 0.43e-12, \tag{3.2.7}$$

其中小的奇异值 $\mathbf{0.42e-8}$ 体现出了 Sylvester 矩阵的数值秩信息. 文 [94] 中计算得到一个三次的近似最大公因子；而 -5.787684 是 $a(x)$ 和 $b(x)$ 的一个 (近似) 公共根.

我们分别构造 Sylvester 矩阵 $S(b,a)$ 和 Hankel 型 Sylvester 矩阵 $H(b,a)$, 用广义 Schur 算法来实现 $S^T(b,a)S(b,a)$ 和 $H^T(b,a)H(b,a)$ 的 Cholesky 分解, 在完成 21 个正步时, 用 $\hat{R}_{21}$ 代表得到的秩为 21 的 Cholesky 因子, 分别测试 $\|S^T(b,a)S(b,a) - \hat{R}_{21}^T\hat{R}_{21}\|$

和 $\|H^T(b,a)H(b,a)-\hat{R}_{21}^T\hat{R}_{21}\|$ 而得到 $0.6e-13$ 和 $0.8e-13$，它们充分表现出了 *Sylvester* 矩阵数值秩亏为 3 的信息. 而应用 *(1.2.1)* 中的 *Sylvester* 矩阵 S 来计算，相应的范数则为 $0.1e-1$. 显然没有给出正确的秩的信息.

§3.3 误差分析

1. 快速 Cholesky 分解的向后误差分析

本节中假定 $T=A^TA$, 这里 $A=H(a,b)$ 或者 $S(a,b)$. (G,J) 是根据定理 3.2.1 或 3.2.6 计算得来的 T 的一个生成子对，Z 是其位移方程中的下三角矩阵. 假设直接应用广义 Schur 算法到 (G,J,Z) 上，在顺利完成 r 步正步后我们得到 T 的（截断的）Cholesky 分解：$\hat{T}_r=\hat{R}_r^T\hat{R}_r$. 循文献 [11, 19, 20, 45, 46] 中的误差分析方法，本节对 T 的快速 Cholesky 分解进行了向后误差分析.

设 $\hat{G}_i$ 是实现广义 Schur 算法第 i 步时的生成子矩阵，$\hat{\hat{G}}_i$ 是由 $\hat{G}_i$ 计算得到的具有合适形式的生成子矩阵，即：在 $\hat{\hat{G}}_i$ 的第一行中只有一个非零元素，且其在 $\hat{\hat{G}}_i$ 的第一列. 并假设 $\hat{\hat{G}}_i$ 是利用两个正交变换和一个初等 Hyperbolic 旋转的作用而得来，而这个初等 Hyperbolic 旋转的实现采用的是 OD 步骤 (见算法 2.1.1). 由于形成生成子矩阵 $G_1=G$ 时的初始误差相对于整个算法的舍入误差来说是微小的，所以循 [19, 20] 中的思路，我们假设 $\hat{G}_1=G_1$.

在 §2.1 中我们已提到，有限精度的计算使得等式 $\hat{\hat{G}}_iJ\hat{\hat{G}}_i^T=\hat{G}_iJ\hat{G}_i^T$ 不能严格成立，现在设

$$\begin{bmatrix}\mathbf{0}\\ \hat{\hat{G}}_i\end{bmatrix}J\begin{bmatrix}\mathbf{0}\\ \hat{\hat{G}}_i\end{bmatrix}^T-\begin{bmatrix}\mathbf{0}\\ \hat{G}_i\end{bmatrix}J\begin{bmatrix}\mathbf{0}\\ \hat{G}_i\end{bmatrix}_i^T=N_i,\ i=1,\ldots,r. \tag{3.3.1}$$

为了便于进行误差分析，我们在上式中给矩阵 $\hat{G}_i$ 和 $\hat{\hat{G}}_i$ 上添加了 $i-1$ 个零向量行，使得这样构成的矩阵具有 $m+n$ 行. 把上面的 r 个等式做和，经过整理后有

$$\hat{T}_r-Z\hat{T}_rZ^T+\begin{bmatrix}\mathbf{0}\\ \hat{G}_{r+1}\end{bmatrix}J\begin{bmatrix}\mathbf{0}\\ \hat{G}_{r+1}\end{bmatrix}^T=\sum_{i=1}^{r}N_i+T-ZTZ^T. \tag{3.3.2}$$

注意到 Z 是幂零的 (假定 $m \geq n$，那么 $Z^m = \mathbf{0}$), 从上式我们得到

$$\hat{T}_r + \sum_{k=0}^{m-1} Z^k \begin{bmatrix} \mathbf{0} \\ \hat{G}_{r+1} \end{bmatrix} J \begin{bmatrix} \mathbf{0} \\ \hat{G}_{r+1} \end{bmatrix}^T (Z^k)^T = T + \sum_{k=0}^{m-1} Z^k \left(\sum_{i=1}^{r} N_i \right) (Z^k)^T. \tag{3.3.3}$$

设

$$\hat{T}_s = \sum_{k=0}^{m-1} Z^k \begin{bmatrix} \mathbf{0} \\ \hat{G}_{r+1} \end{bmatrix} J \begin{bmatrix} \mathbf{0} \\ \hat{G}_{r+1} \end{bmatrix}^T (Z^k)^T, \tag{3.3.4}$$

$$E = \sum_{k=0}^{m-1} Z^k \left(\sum_{i=1}^{r} N_i \right) (Z^k)^T, \tag{3.3.5}$$

我们有

$$\hat{T}_r + \hat{T}_s = T + E, \tag{3.3.6}$$

其中 $\hat{T}_s = \begin{bmatrix} \mathbf{0} & \\ & \hat{S}_{r+1} \end{bmatrix}$,

$$\hat{S}_{r+1} = \sum_{k=0}^{m-1} Z^{(r+1)^k} \hat{G}_{r+1} J \hat{G}_{r+1}^T Z^{(r+1)^{k^T}}, \tag{3.3.7}$$

这里 $Z^{(r+1)}$ 是去掉 Z 的前 r 行和前 r 列后得到的下三角矩阵. 注意到 $Z^{(r+1)}$ 是严格下三角的，$\hat{S}_{r+1}$ 是下列位移方程的唯一解：

$$\hat{S}_{r+1} - Z^{(r+1)} \hat{S}_{r+1} Z^{(r+1)^T} = \hat{G}_{r+1} J \hat{G}_{r+1}^T. \tag{3.3.8}$$

进一步地，由 $\hat{T}_r$ 的定义，(3.3.6) 表明 $\hat{S}_{r+1}$ 是 $T+E$ (关于其 r 阶顺序主子矩阵) 的准确的 Schur 补，而 $\hat{T}_s - E$ 是快速 Cholesky 分解过程产生的向后误差矩阵. 我们现在来估计向后误差 $\|\hat{T}_s - E\|$ 的一个上界.

首先，循文献 [11, 20] 中的思路，我们推导出

$$\|E\| \leq c_1 u \left(6r^3 \|\hat{R}_r^T\|^2 + (24r+4)\|T\| \right) + O(u^2), \tag{3.3.9}$$

这里 c_1 代表一个低阶的关于 m 的多项式，u 代表机器精度. 因为

$$\|\hat{R}_r^T\|^2 = \|\hat{T}_r\| \leq \|T\| + \|E\| + \|\hat{T}_s\|, \tag{3.3.10}$$

我们进而得到

$$\|E\| \leq \frac{c_1 u}{1-6r^3c_1u}\Big((6r^3+24r+4)\|T\|+6r^3\|\hat{T}_s\|\Big)+O(u^2). \tag{3.3.11}$$

这样，我们可以先把向后误差做如下界定：

$$\|T-\hat{T}_r\| \leq \|E\|+\|\hat{T}_s\| \leq \frac{c_1u(6r^3+24r+4)\|T\|+\|\hat{T}_s\|}{1-6r^3c_1u}. \tag{3.3.12}$$

为了界定 $\|\hat{T}_s\|$, 即 $\|\hat{S}_{r+1}\|$, 我们把 T 写成

$$T=\tilde{T}+\Delta T, \tag{3.3.13}$$

这里 $\tilde{T}$ 是离 T 最近的一个秩为 r 的半正定矩阵. 具体说来，假定

$$T=V\operatorname{diag}\left(\sigma_1(T),\ldots,\sigma_{n+m}(T)\right)V^T \tag{3.3.14}$$

是 T 的奇异值分解，那么 $\tilde{T}$ 取为如下矩阵：

$$\tilde{T}=V\operatorname{diag}\left(\sigma_1(T),\ldots,\sigma_r(T),0,\ldots,0\right)V^T, \tag{3.3.15}$$

这样

$$\|\Delta T\|=\sigma_{r+1}(T). \tag{3.3.16}$$

设 $F=\Delta T+E$, 那么 $T+E=\tilde{T}+F$, 注意到 $\hat{S}_{r+1}$ 是 $T+E$ 的 Schur 补，那么 $\hat{S}_{r+1}$ 也是 $\tilde{T}+F$ 的 Schur 补. 现把 $\tilde{T}$ 和 F 按如下分块：

$$\tilde{T}=\begin{bmatrix}\tilde{T}_{11} & \tilde{T}_{12}\\ \tilde{T}_{12}^T & \tilde{T}_{22}\end{bmatrix},\quad F=\begin{bmatrix}F_{11} & F_{12}\\ F_{12}^T & F_{22}\end{bmatrix}, \tag{3.3.17}$$

这里 $\tilde{T}_{11},F_{11}\in\mathbb{R}^{r\times r}$, 循文献 [45] 中的思路我们有：

引理 3.3.1 假设 $\tilde{T}_{11}$ 是可逆的，那么

$$\hat{S}_{r+1}=F_{22}-\tilde{W}^TF_{12}-F_{12}^T\tilde{W}+\tilde{W}^TF_{11}\tilde{W}+O(\|F\|^2), \tag{3.3.18}$$

其中

$$\tilde{W}=\tilde{T}_{11}^{-1}\tilde{T}_{12}. \tag{3.3.19}$$

证明 $\tilde{T}_{11}$ 的可逆性保证了 $\tilde{T}$ 的 Schur 补 $\tilde{T}_{22}-\tilde{T}_{12}^T\tilde{T}_{11}^{-1}\tilde{T}_{12}$ 是有定义的. 于是 $\tilde{T}+F$ 关于其顺序主子矩阵 $\tilde{T}_{11}+F_{11}$ 的 Schur 补 $\hat{S}_{r+1}$ 可以按如下等式计算:

$$\begin{aligned}\hat{S}_{r+1} &= \tilde{T}_{22}+F_{22}-(\tilde{T}_{12}^T+F_{12}^T)(\tilde{T}_{11}+F_{11})^{-1}(\tilde{T}_{12}+F_{12})\\ &= \tilde{T}_{22}-\tilde{T}_{12}^T\tilde{T}_{11}^{-1}\tilde{T}_{12}+F_{22}-\tilde{T}_{12}^T\tilde{T}_{11}^{-1}F_{12}-F_{12}^T\tilde{T}_{11}^{-1}\tilde{T}_{12}+\tilde{T}_{12}^T\tilde{T}_{11}^{-1}F_{11}\tilde{T}_{11}^{-1}\tilde{T}_{12}+O(\|F\|^2)\\ &= F_{22}-\tilde{T}_{12}^T\tilde{T}_{11}^{-1}F_{12}-F_{12}^T\tilde{T}_{11}^{-1}\tilde{T}_{12}+\tilde{T}_{12}^T\tilde{T}_{11}^{-1}F_{11}\tilde{T}_{11}^{-1}\tilde{T}_{12}+O(\|F\|^2).\end{aligned}$$

第三个等式成立是因为 $\tilde{T}$ 的 Schur 补 $\tilde{T}_{22}-\tilde{T}_{12}^T\tilde{T}_{11}^{-1}\tilde{T}_{12}=\mathbf{0}$. 把 $\tilde{T}_{11}^{-1}\tilde{T}_{12}$ 代替成 $\tilde{W}$ 就得到了引理中最后的等式. ■

基于引理 3.3.1, 经过简单的计算我们得到：

$$\|\hat{T}_s\|=\|\hat{S}_{r+1}\|\le(1+\|\tilde{W}\|)^2\|F\|+O(\|F\|^2),$$

从而,

$$\|\hat{T}_s\|\le(1+\|\tilde{W}\|)^2(\|\Delta T\|+\|E\|)+O(\|F\|^2).$$

结合 (3.3.11), 我们得到了

$$\|\hat{T}_s\|\le\Omega\left(\frac{c_1u(6r^3+24r+4)\|T\|}{1-6r^3c_1u}+\|\Delta T\|\right)(1+\|\tilde{W}\|)^2+O(\|F\|^2), \tag{3.3.20}$$

这里 $\Omega=\left(1-\frac{6r^3c_1u}{1-6r^3c_1u}(1+\|\tilde{W}\|)^2\right)^{-1}$. 最后，用 (3.3.20) 和 (3.3.12) 我们就得到了下面的定理：

定理 3.3.2 *设 $T=\tilde{T}+\Delta T$ 是一个如 (3.3.13) 所定义的对称矩阵. 把 $\tilde{T}$ 和 ΔT 按照 (3.3.17) 进行分块. 假定*

$$\frac{\|\Delta T_{11}\|}{\|T_{11}\|}=\theta,$$

T_{11} 是对称正定的，并且满足：

$$2\theta\kappa_2(T_{11})<1; \tag{3.3.21}$$

再假定

$$\frac{6r^3c_1u}{1-6r^3c_1u}(1+\|\tilde{W}\|)^2<1/2, \tag{3.3.22}$$

那么顺利地完成广义 Schur 算法的前 r 步后我们会有：

$$\|T-\hat{R}_r^T\hat{R}_r\|\le\frac{c_1u\zeta}{1-6r^3c_1u}+\frac{2}{1-6r^3c_1u}\left(\frac{c_1u\zeta}{1-6r^3c_1u}+\|\Delta T\|\right)(1+\|\tilde{W}\|)^2+O(\|F\|^2), \tag{3.3.23}$$

其中

$$\tilde{W} = \tilde{T}_{11}^{-1}\tilde{T}_{12},$$
$$\zeta = (6r^3 + 24r + 4)\|T\|.$$

证明 (3.3.21) 中的假设保证了 $\tilde{T}_{11}$ 的可逆性. 事实上,

$$\|T_{11}^{-1}\Delta T_{11}\| \le \|T_{11}^{-1}\|\|\Delta T_{11}\| = \kappa_2(T_{11})\theta < \frac{1}{2},$$

于是由 T_{11} 的可逆性我们推出了 $\tilde{T}_{11} = T_{11} - \Delta T_{11}$ 的可逆性. 这样我们可以应用引理 3.3.1 的结果及其相关分析.

在顺利完成广义 Schur 算法的前 r 步后我们得到 (3.3.6) 和 (3.3.12), 以及 $T+E = \tilde{T} + F$ 的一个 Schur 补 $\hat{S}_{r+1}$. 在应用引理 3.3.1 后又得到 (3.3.20). 现在把 (3.3.20) 和 (3.3.12) 结合起来有:

$$\|T - \hat{R}_r^T\hat{R}_r\| \le \frac{c_1 u\zeta}{1-6r^3c_1u} + \frac{\Omega}{1-6r^3c_1u}\left(\frac{c_1u\zeta}{1-6r^3c_1u} + \|\Delta T\|\right)(1+\|\tilde{W}\|)^2 + O(\|F\|^2), \tag{3.3.24}$$

其中

$$\Omega = \left(1 - \frac{6r^3c_1u}{1-6r^3c_1u}(1+\|\tilde{W}\|)^2\right)^{-1}.$$

最后应用假设 (3.3.22) 就得到了定理中最后的结果. ■

注 3.3.3 根据定理 *3.3.2*, 应用广义 *Schur* 算法对 T 进行快速 *Cholesky* 分解的向后误差大小依赖于 $\|\tilde{W}\|$ 的大小. 向后误差可以近似地界定为:

$$\|T - \hat{R}_r^T\hat{R}_r\| \lessapprox (1+\|\tilde{W}\|)^2\|\Delta T\|. \tag{3.3.25}$$

对于全选主元的 Cholesky 分解, $\|\tilde{W}\|$ 可以有一个很好的上界 [45]. 然而, 在本文快速 Cholesky 分解的实现过程中, 为了避免破坏矩阵 T 的位移结构我们不可以进行选主元, 因而我们只能给出与文 [46] 中引理 10.12 类似的一个上界.

定理 3.3.4 *对于定理 3.3.2 中所定义的 $\tilde{W}$, 我们有*

$$\|\tilde{W}\| \le \sqrt{2\|T_{11}^{-1}\|\|T\|}. \tag{3.3.26}$$

证明 注意到 $\tilde{T}$ 关于 $\tilde{T}_{11}$ 的 Schur 补等于 $\mathbf{0}$, 即: $\tilde{T}_{22} - \tilde{T}_{12}^T \tilde{T}_{11}^{-1} \tilde{T}_{12} = \mathbf{0}$, 我们有

$$\|\tilde{W}\| = \left\|\tilde{T}_{11}^{-1}\tilde{T}_{12}\right\| = \left\|\tilde{T}_{11}^{-1/2}\tilde{T}_{11}^{-1/2}\tilde{T}_{12}\right\| \leq \left\|\tilde{T}_{11}^{-1}\right\|^{1/2} \left\|\tilde{T}_{12}^T\tilde{T}_{11}^{-1}\tilde{T}_{12}\right\|^{1/2} = \sqrt{\|\tilde{T}_{11}^{-1}\|\|\tilde{T}_{22}\|}.$$

进一步地，基于定理 3.3.2 中的假设，我们有

$$\|T_{11}^{-1}\Delta T_{11}\| \leq \|T_{11}^{-1}\|\|\Delta T_{11}\| = \kappa_2(T_{11})\frac{\|\Delta T_{11}\|}{\|T_{11}\|} < 1/2,$$

从而

$$\|\tilde{T}_{11}^{-1}\| \leq \frac{\|T_{11}^{-1}\|}{1 - \|T_{11}^{-1}\Delta T_{11}\|} < 2\|T_{11}^{-1}\|.$$

同时，由 $\tilde{T}$ 的定义我们有：

$$\|\tilde{T}_{22}\| \leq \|\tilde{T}\| = \|T\|,$$

这样我们就得到了 $\|\tilde{W}\| \leq \sqrt{2\|T_{11}^{-1}\|\|T\|}$.

■

注 3.3.5 上面的定理描述了 $\|\tilde{W}\|$ 的一个上界，这个上界在一定程度上依赖于 $\sqrt{\kappa_2(T_{11})}$. 事实上，不难构造出 $\|\tilde{W}\|$ 很大的例子来，此时 $\|\tilde{W}\|$ 接近于定理中描述的上界. 然而，在 §*3.4* 我们将要描述的算法 *HSylRRA* 中，我们基于 §*3.2* 中定理 *3.2.3* 和定理 *3.2.7* 的结果来构造矩阵 A, $A = S(a,b)$ 或者 $H(a,b)$. 实验结果表明，对于这样的 $T = A^T A$, $\|\tilde{W}\|$ 在很多情况下远小于这个上界.

2. 逼近奇异值时的一个误差界

假设通过实现 $T = A^T A$ 的快速 Cholesky 分解我们得到了 $\hat{R}_r^T \hat{R}_r$, 其中 $\hat{R}_r$ 是截断的 Cholesky 因子, $A = S$ 或 H. 经典的矩阵特征值理论告诉我们，$\sigma_r(\hat{R}_r)$ 和 $\hat{R}_r$ 的零空间基底向量可以分别用于逼近 $\sigma_r(A)$ 和 $\sigma_{r+1}(A)$.

定理 3.3.6 *假定在顺利完成 r 步广义 Schur 算法后得到的误差为 $\bar{\varepsilon}$, 即*

$$\left\|T - \hat{R}_r^T \hat{R}_r\right\| \leq \bar{\varepsilon}, \tag{3.3.27}$$

那么

$$\left|\sigma_i(A) - \sigma_i(\hat{R}_r)\right| \leq \bar{\varepsilon}/\sigma_r(A), \; i = 1, \ldots, r. \tag{3.3.28}$$

进一步地，假定 $V_r = [\mathbf{v}_{r+1}, \ldots, \mathbf{v}_{n+m}]$ 和 $W_r = [\mathbf{w}_{r+1}, \ldots, \mathbf{w}_{n+m}]$ 是两个由标准正交列向量所构成的矩阵，其中 V_r 满足

$$A^T A V_r = V_r \Lambda_r, \text{ 这里 } \Lambda_r = \mathrm{diag}\left(\sigma_{r+1}^2(A), \ldots, \sigma_{n+m}^2(A)\right), \tag{3.3.29}$$

而 W_r 由 $\hat{R}_r$ 的零空间基底向量构成，那么

$$\|\sin\Theta(V_r, W_r)\| \le \sqrt{m+n-r}\ \bar{\varepsilon}/\sigma_r^2(A), \tag{3.3.30}$$

$$|\|AW_r\| - \sigma_{r+1}(A)| \le \frac{4\|A\|^2 (m+n-r)\bar{\varepsilon}^2}{\sigma_r^4(A)\sigma_{r+1}(A)}. \tag{3.3.31}$$

证明 应用文献 [47] 中推论 7.3.8 到 (3.3.27) 我们得到

$$\left|\sigma_i(A^T A) - \sigma_i(\hat{R}_r^T \hat{R}_r)\right| = \left|\sigma_i^2(A) - \sigma_i^2(\hat{R}_r)\right| \le \bar{\varepsilon},\ i = 1, 2, \ldots, r.$$

于是，就有不等式 (3.3.28):

$$\left|\sigma_i(A) - \sigma_i(\hat{R}_r)\right| \le \bar{\varepsilon}/\left(\sigma_i(A) + \sigma_i(\hat{R}_r)\right) \le \bar{\varepsilon}/\sigma_i(A) \le \bar{\varepsilon}/\sigma_r(A),\ i = 1, 2, \ldots, r.$$

为了证明不等式 (3.3.30) 我们要考虑到

$$\sigma_j(\hat{R}_r) = 0,\ j = r+1, \ldots, m+n.$$

现定义

$$\bar{\delta} = \min_{i,j}\left|\sigma_i^2(A) - \sigma_j^2(\hat{R}_r)\right|,\ i = 1, \ldots, r,\ j = r+1, \ldots, m+n.$$

那么：

$$\bar{\delta} = \sigma_r^2(A).$$

同时，设 $\bar{R}$ 代表下列残数矩阵：

$$\bar{R} = A^T A W_r - W_r D_r, \text{ 其中 } D_r = \mathrm{diag}\left(\sigma_{r+1}^2(\hat{R}_r), \ldots, \sigma_{m+n}^2(\hat{R}_r)\right),$$

我们有

$$\begin{aligned}
\|\bar{R}\| &= \left\|A^T A W_r\right\| \\
&\le \left\|A^T A W_r - \hat{R}_r^T \hat{R}_r W_r\right\| + \left\|\hat{R}_r^T \hat{R}_r W_r\right\| \\
&\le \left\|A^T A - \hat{R}_r^T \hat{R}_r\right\| \|W_r\| \\
&\le \bar{\varepsilon}.
\end{aligned}$$

因此, 根据矩阵特征值扰动理论 [87] 我们得到

$$\|\sin\Theta(V_r,W_r)\| \le \sqrt{m+n-r}\left\|\bar{R}\right\|/\bar{\delta} \le \sqrt{m+n-r}\ \bar{\varepsilon}/\sigma_r^2(A).$$

现在来估计 $\|HW_r\|$. 基于 Rayleigh 商矩阵的特征值理论 [60, 64] 我们得到

$$\begin{aligned}
&\left|\|AW_r\|^2-\sigma_{r+1}^2(A)\right| \\
=& \left|\lambda_1(W_r^TA^TAW_r)-\lambda_1(V_r^TA^TAV_r)\right| \\
\le& 4\left\|A^TA\right\|\|\sin\Theta(V_r,W_r)\|^2,
\end{aligned}$$

结合不等式 (3.3.30), 我们有

$$\left|\|AW_r\|^2-\sigma_{r+1}^2(A)\right| \le 4\left\|A^TA\right\|(m+n-r)\bar{\varepsilon}^2/\sigma_r^4(A).$$

最后得到：

$$|\|AW_r\|-\sigma_{r+1}(A)| \le \frac{4\|A\|^2(m+n-r)\bar{\varepsilon}^2}{\sigma_r^4(A)\sigma_{r+1}(A)}.$$

■

根据定理 3.3.6, 当 $\bar{\varepsilon}$ 是一个小的量, 而 $\sigma_r(A)=\sigma_r(S)$ 比较大时, $\sigma_r(\hat{R}_r)$ 和 $\|AW_r\|$ 分别是 $\sigma_r(S)$ 和 $\sigma_{r+1}(S)$ 的很好的逼近.

§3.4 求解数值秩的快速算法的描述

我们应用广义 Schur 算法来实现 T 的 Cholesky 分解, 这里 $T=A^TA$, 而 A 等于 $S(a,b)$ 或者 $H(a,b)$. 广义 Schur 算法直接作用于 T 的生成子对 (G,J) 和下三角矩阵 Z, 其实现 Cholesky 分解所需的浮点运算量为 $O((m+n)^2)$. 设 $\hat{R}_i$ 是完成广义 Schur 算法的前 i 个正步而得到的（截断的）Cholesky 因子,

$$\hat{R}_i=\begin{bmatrix}\hat{R}_{11}^{(i)} & \hat{R}_{12}^{(i)}\\ 0 & 0\end{bmatrix},$$

其中 $\hat{R}_{11}^{(i)}$ 是一个 $i\times i$ 的上三角矩阵. 假定完成广义 Schur 算法的前 $\hat{r}$ 个正步后, $\|T-\hat{R}_{\hat{r}}^T\hat{R}_{\hat{r}}\|$ 变得足够小，那么我们就令广义 Schur 算法停止，而称 Sylvester 矩阵的数值

秩为 $\hat{r}$. 在算法中我们设定了一个关于 Cholesky 分解向后误差大小的临界值 γ，当 $\|T-\hat{R}_{\hat{r}}^T\hat{R}_{\hat{r}}\|\le\gamma$ 时我们就认为 $\|T-\hat{R}_{\hat{r}}^T\hat{R}_{\hat{r}}\|$ 是足够小的.

本质上，我们希望通过实现 T 的 Cholesky 分解来计算 S^TS 的 ε^2- 秩，进而得到 S 的数值 $\varepsilon-$ 秩. 根据数值秩的定义，我们要在 T 的 ε^2 邻域内找秩最低的亏秩矩阵，从而 γ 应取为 ε^2; 而实际上，基于第 3.3.1 节的向后误差分析和 (3.3.25) 式，当 Sylvester 矩阵的数值 $\varepsilon-$ 秩为 r 时，计算 T 的 Cholesky 分解 $\hat{R}_r^T\hat{R}_r$ 所具有的向后误差近似地被 $\varepsilon^2(1+\|\tilde{W}\|)^2$ 所界定. 这里注意到，当 Sylvester 矩阵的数值 $\varepsilon-$ 秩为 r 时，

$$\varepsilon^2\ge\sigma_{r+1}(T)=\|\Delta T\|.$$

这说明我们需要在 T 的比 ε^2 大的邻域内找秩为 r 的半正定矩阵 $\hat{R}_r^T\hat{R}_r$. 因此我们在算法中设定 γ 为 ε^2 的一个合适的倍数.

正如文 [45, 85] 中所讨论的，我们没有必要计算误差 $\|T-\hat{R}_i^T\hat{R}_i\|$，我们完全可以把 γ 直接设定在 Schur 补范数 $\|\hat{S}_{i+1}\|$ 上，这里

$$\hat{S}_{i+1}=\sum_{k=0}^{m-1}Z^{(i+1)^k}\hat{G}_{i+1}J\hat{G}_{i+1}^TZ^{(i+1)^{k^T}},\tag{3.4.1}$$

它由下列位移方程所确定：

$$\hat{S}_{i+1}-Z^{(i+1)}\hat{S}_{i+1}Z^{(i+1)^T}=\hat{G}_{i+1}J\hat{G}_{i+1}^T,\tag{3.4.2}$$

其中 $Z^{(i+1)}$ 是从 Z 中去掉前 i 行和前 i 列后得到的下三角矩阵. 事实上，与关系式 (3.3.6) 和 (3.3.11) 同理我们可以得到，$T-\hat{R}_i^T\hat{R}_i=E+\hat{T}_s^{(i+1)}$, 其中 $\|E\|=O(u)(\|T\|+\|\hat{T}_s^{(i+1)}\|)$, $\hat{T}_s^{(i+1)}=\begin{bmatrix}\mathbf{0} & \\ & \hat{S}_{i+1}\end{bmatrix}$, 所以完成 i 步广义 Schur 算法时向后误差的大小可以由 $\|\hat{S}_{i+1}\|=\|\hat{T}_s^{(i+1)}\|$ 来近似. 而利用 $\hat{S}_{i+1}$ 的生成子表示形式 (3.4.1)，我们可以高效地计算出每一个 Schur 补 $\hat{S}_{i+1}$, $i\ge 1$ [52, 92].

算法 HSylRRA

输入: $a(x)$, $b(x)-$ 两个一元多项式，$\varepsilon-$ 给定的一个正数，$\gamma-$ Cholesky 分解向后误差大小的临界值，

输出: $\hat{r}-$ Sylvester 矩阵 S 的备选的数值秩，$\hat{R}_{\hat{r}}-$ T 的截断的 Cholesky 因子，$A-$ Sylvester 矩阵或者 Hankel 型 Sylvester 矩阵.

1. 初始化
   (a) 构造合适的 S 和 H.
      i. 设 $\bar{a} := a, \bar{b} := b$ 或者 $\bar{a} := b, \bar{b} := a$ 使得 $\frac{|lc(\bar{a})|}{\|\bar{a}\|_2} \geq \frac{|lc(\bar{b})|}{\|\bar{b}\|_2}$, 这里 $lc(\cdot)$ 代表一个多项式的首项系数;
      ii. 假定 $a_p \neq 0$ 和 $b_q \neq 0$ 分别是多项式 a 和 b 的第一个非零末系数. 如果 $p \leq q$ 那么设 $\tilde{a} := a, \tilde{b} := b$ 或者 $\tilde{a} := b, \tilde{b} := a$ 使得 $\frac{|\tilde{a}_p|}{\|\tilde{a}\|_2} \geq \frac{|\tilde{b}_p|}{\|\tilde{b}\|_2}$; 否则设 $\tilde{a} := b, \tilde{b} := a, p := q$.
   (b) 如果 $\frac{|lc(\bar{a})|}{\|\bar{a}\|_2} \geq \frac{|\tilde{a}_p|}{\|\tilde{a}\|_2}$ 那么设 $A := H(\bar{a}, \bar{b})$; 否则, 设 $A := S(\tilde{a}, \tilde{b})$.
   (c) 设 $T := A^T A$, 根据定理 3.2.1 或定理 3.2.6 来构造 T 的生成子矩阵对 (G, J) 和下三角矩阵 Z.
2. 实现 T 的 Cholesky 分解来求秩
   (a) 把广义 Schur 算法作用于 (G, J, Z) 来实现 T 的 Cholesky 分解, 同时计算 Schur 补 $\hat{S}_{i+1}$, $i > 1$.
   (b) 当 $\|\hat{S}_{\hat{r}+1}\| \leq \gamma$ 时令广义 Schur 算法停止.
   (c) 输出 Sylvester 矩阵的数值秩 $\hat{r}$ 和 T 的 Cholesky 因子 $\hat{R}_{\hat{r}}$.

注 3.4.1 注意到当 Sylvester 矩阵的数值秩为 r 时, 对于 $\gamma \geq \sigma_r(T)$, 在 T 的 γ 邻域内可能会有秩为 $r-1$ 的秩亏矩阵, 这是我们所不希望的, 所以一个合适的 γ 应该满足 $\gamma < \sigma_r(T)$. 结合下一节中的数值结果, 我们的算法要求 $\sigma_r(S)$ 与 $\sigma_{r+1}(S)$ 之间有一个比较大的跳跃, 通常要求 $\sigma_r(S)/\sigma_{r+1}(S)$ 大于等于 10^2.

注 3.4.2 一般来说我们没有必要计算所有的 Schur 补. 我们可以根据预先的关于数值秩的信息来选择计算一部分 Schur 补.

正如在第 3.3.2 节中所提到的, 我们可以通过计算 $\sigma_{\hat{r}}(\hat{R}_{\hat{r}})$ 和 $\|AW_{\hat{r}}\|$ 分别得到 $\sigma_{\hat{r}}(S)$ 和 $\sigma_{\hat{r}+1}(S)$ 的近似值, 这里 $W_{\hat{r}}$ 代表 $\hat{R}_{\hat{r}}$ 的标准正交零空间基底所构成的矩阵. 本文中应用文 [63] 中的算法来计算 $\sigma_{\hat{r}}(\hat{R}_{\hat{r}})$ 和 $\hat{R}_{\hat{r}}$ 的一个标准正交零空间基底. 因为 $\hat{R}_{\hat{r}}$ 是上三角矩阵, 所以这一过程所需的浮点运算量为 $O((m+n)^2)$.

§3.5 实验结果

我们在 Maple9.5 上实现了本章所提出的快速算法. 在本节中我们给出应用算

法 HSylSRRA 来求解 Sylvester 矩阵数值秩的实验结果，并给出应用这一算法求解 Sylvester 矩阵的某些奇异值时的实验结果. 这里我们设 Digits = 16.

表 3.1: 数值秩的计算结果 (I)

(m,n,d)	$\gamma=10^4\varepsilon^2$			$\gamma=10^5\varepsilon^2$			$\gamma=10^6\varepsilon^2$		
	ε_{aver}	case	num.	ε_{aver}	case	num.	ε_{aver}	case	num.
$(71,56,11)$	0.1006	$\hat{r}=r$	39	0.0099	$\hat{r}=r$	47	0.000101	$\hat{r}=r$	48
		$\hat{r}<r$	2		$\hat{r}<r$	1		$\hat{r}<r$	1
		$\hat{r}>r$	9		$\hat{r}>r$	2		$\hat{r}>r$	1
$(68,53,8)$	0.085	$\hat{r}=r$	46	0.0090	$\hat{r}=r$	48	0.000087	$\hat{r}=r$	47
		$\hat{r}<r$	1		$\hat{r}<r$	2		$\hat{r}<r$	0
		$\hat{r}>r$	3		$\hat{r}>r$	0		$\hat{r}>r$	3
$(80,78,3)$	0.064	$\hat{r}=r$	46	0.0068	$\hat{r}=r$	48	0.000065	$\hat{r}=r$	50
		$\hat{r}<r$	4		$\hat{r}<r$	2		$\hat{r}<r$	0
		$\hat{r}>r$	0		$\hat{r}>r$	0		$\hat{r}>r$	0
$(43,38,8)$	0.066	$\hat{r}=r$	46	0.0067	$\hat{r}=r$	45	0.000067	$\hat{r}=r$	49
		$\hat{r}<r$	1		$\hat{r}<r$	0		$\hat{r}<r$	0
		$\hat{r}>r$	3		$\hat{r}>r$	5		$\hat{r}>r$	1

部分 (I)：

在这一部分我们测试了 600 个多项式对. 所测试的多项式构造如下：在每一个多项式对中，互素部分和最大公因子都是随机生成的多项式，其系数是区间 $-10\leq c\leq 10$ 中的随机整数. 然后对这样构造的多项式再加以扰动. 至于扰动的构造，我们选择相对扰动为 10^{-e}: 先随机选择一个与乘积多项式次数相同，系数在区间 $[-10^e, 10^e]$ 的多项式，然后我们规范化这个扰动多项式以使得其相对扰动为 10^{-e}. 对于每个多项式对，我们令比较大的那个扰动多项式范数为这组多项式的扰动容许度 ε. 我们将要用算法 HSylSRRA 来计算每一个这样多项式对所构成的 Sylvester 矩阵的 ε- 秩.

在表 3.1 和表 3.2 中，(m,n,d) 代表所测试的多项式的次数及其构造的最大公因子的次数. 对每一个 (m,n,d) 我们构造了三组多项式对, 每一组包含 50 个多项式对，都受到相等大小的相对扰动; 这三组多项式对我们取的相对扰动分别为：$10^{-4},10^{-5},10^{-7}$.

表 3.2: 奇异值的相对误差结果 (I)

(m,n,d)	ε_{aver}	$err\text{-}\hat{\sigma}_r$ $err\text{-}\hat{\sigma}_{r+1}$	ε_{aver}	$err\text{-}\hat{\sigma}_r$ $err\text{-}\hat{\sigma}_{r+1}$	ε_{aver}	$err\text{-}\hat{\sigma}_r$ $err\text{-}\hat{\sigma}_{r+1}$
$(71,56,11)$	0.1006	$0.33e-1$ $0.147e-2$	0.0099	$0.195e-1$ $0.34e-4$	0.000101	$0.22e-1$ $0.33e-6$
$(68,53,8)$	0.085	$0.43e-1$ $0.24e-3$	0.0090	$0.23e-1$ $0.20e-4$	0.000087	$0.39e-1$ $0.17e-6$
$(80,78,3)$	0.064	$0.32e-1$ $0.24e-5$	0.0068	$0.32e-1$ $0.14e-6$	0.000065	$0.24e-1$ $0.63e-6$
$(43,38,8)$	0.066	$0.36e-1$ $0.20e-2$	0.0067	$0.30e-1$ $0.28e-5$	0.000067	$0.25e-1$ $0.91e-6$

我们报告了每一组多项式对的扰动容许度 ε 的平均值，记为 ε_{aver}. 如表 3.1 所示，我们取 Cholesky 分解向后误差大小的临界值 γ 为 ε^2 乘以倍数：$10^4, 10^5, 10^6$. 我们用 $\hat{r}$ 代表应用算法 HSylSRRA 所计算的 Sylvester 矩阵的数值秩，用 r 代表通过奇异值分解得到的 Sylvester 矩阵的数值秩，即：Sylvester 矩阵的奇异值满足 $\sigma_r(S) > \varepsilon \geq \sigma_{r+1}(S)$. 我们在表 3.1 中报告了 $\hat{r}$ 和 r 的比较结果.

在表 3.1 中，对于 $\hat{r} = r$ 的例子，Schur 补 $\hat{S}_{r+1}$ 都有相对小的范数，相比于 Schur 补 $\hat{S}_r$ 来说，它们的大小下降得很明显. $\|\tilde{W}\|$ 的平均值小于 1000. 对于 $\hat{r} < r$ 的情形，广义 Schur 算法没有进行完前 r 步就停止；这样例子中的一部分，特别是对于 $d = 3$ 的例子，$\|\tilde{W}\|$ 很小，而算法提前停止的原因是由于 γ 取得过大 (见注 3.4.1). 至于 $\hat{r} > r$ 的例子，$\tilde{W}$ 的范数则都比较大，例如，大于 $O(10^3)$.

值得一提的是，对于表 3.1 中 $d = 3$ 的三组例子，我们总有 $\|\tilde{W}\| = O(100)$; 对于 $\varepsilon_{aver} = 0.000065$ 这一组例子，取 $\gamma = 10^5\varepsilon^2$ 我们得到与表中所示完全相同的结果.

对于 $\hat{r} = r$ 的情形，我们继续应用算法 HSylSRRA 的计算结果，按 §3.4 所描述的方法来计算 Sylvester 矩阵的第 r 大和第 $r+1$ 大的奇异值，并把计算得来的结果分别记为 $\hat{\sigma}_r$ 和 $\hat{\sigma}_{r+1}$. 我们测试了 $\hat{\sigma}_r$ 和 $\hat{\sigma}_{r+1}$ 的相对误差：$\frac{|\hat{\sigma}_r - \sigma_r(S)|}{\sigma_r(S)}$ 和 $\frac{|\hat{\sigma}_{r+1} - \sigma_{r+1}(S)|}{\sigma_{r+1}(S)}$, 这里 $\sigma_r(S)$ 和 $\sigma_{r+1}(S)$ 是应用奇异值分解得到的 Sylvester 矩阵的奇异值. 在表 3.2 中我们报告了相对误差的平均值，分别记为 err-$\hat{\sigma}_r$ 和 err-$\hat{\sigma}_{r+1}$. 如表 3.2 中相对误差所表明，

表 3.3: 数值秩与奇异值的计算结果 (II)

(m,n,d)	γ	ε_{aver}	case	num.	err-$\hat{\sigma}_r$	err-$\hat{\sigma}_{r+1}$
$(71,61,11)$	$10^5\varepsilon^2$	0.00196	$\hat{r}=r$	28	$0.43e-1$	$0.97e-2$
			$\hat{r}<r$	6		
			$\hat{r}>r$	16		
$(68,58,8)$	$10^5\varepsilon^2$	0.00166	$\hat{r}=r$	38	$0.14e-1$	$0.28e-2$
			$\hat{r}<r$	3		
			$\hat{r}>r$	9		
$(78,78,3)$	$10^4\varepsilon^2$	0.0012	$\hat{r}=r$	47	$0.83e-2$	$0.106e-1$
			$\hat{r}<r$	3		
			$\hat{r}>r$	0		

作为 $\sigma_r(S)$ 和 $\sigma_{r+1}(S)$ 的近似值，$\hat{\sigma}_r$ 至少具有正确的数量级，而 $\hat{\sigma}_{r+1}$ 则具有多位正确的有效位数.

部分 (II)：

这一部分我们测试了 150 个多项式对. 所测试的多项式对是先按照 (I) 中 描述的方法构造出一个多项式对，然后把这样的多项式对作为互素多项式对，继续应用 (I) 中描述的方法构造得到的. 很明显，这样构造的例子，相比于表 3.1 中 用到的例子，注 3.3.5 中所说的 $\sqrt{\kappa_2(T_{11})}$ 要大得多. 对于每一个 “互素” 多项式对，我们分别乘以次数为 $d=11,8,3$ 的最大公因子后再作以扰动，取的相对扰动均为 10^{-6}. 对于每个多项式对，我们仍然记比较大的那个扰动多项式范数为这组多项式的扰动容许度 ε. 根据次数 d 的不同我们把 150 个多项式对分成了三组.

至于 “互素” 多项式对的构造，所乘的最大公因子均是次数为 5 的随机多项式，所受到的相对扰动均为 10^{-2}. 为了防止最后得到的多项式（相比于 (I) 中构造的多项式）系数膨胀得太大，我们在应用 (I) 中描述的多项式构造方法时，把随机生成多项式的区间 $-10\le c\le 10$ 改为 $-5\le c\le 5$.

我们把计算 Sylvester 矩阵数值 ε- 秩的结果和计算奇异值的结果都在表 3.3 中给出. 表 3.3 中所用到的符号与表 3.1 和表 3.2 中含义相同. γ 是为每次 Cholesky 分解停止所设定的向后误差大小临界值；ε_{aver} 表明了每一组中多项式对所受的扰动大小.

err-$\hat{\sigma}_r$ 和 err-$\hat{\sigma}_{r+1}$ 表明了 $\hat{\sigma}_r$ 和 $\hat{\sigma}_{r+1}$ 逼近 $\sigma_r(S)$ 和 $\sigma_{r+1}(S)$ 的精度. 相比于表 3.2 中结果, $\hat{\sigma}_{r+1}$ 的逼近精度没有那么高, 从第 3.3.2 节中的误差分析可以看出, 这主要是由于这里的三组例子中, $\sigma_r(S)$ 都有相对小的值.

如表 3.3 中所示, 对于 $d = 11$ 这组例子, 计算数值秩的准确程度相比于表 3.1 中的情形有所下降, 这体现出了 $\|\tilde{W}\|$ 有所增长的结果, 尽管 $\|\tilde{W}\|$ 相比于 $\sqrt{\kappa_2(T_{11})}$ 通常还是小很多. 而对于 $d = 8$ 和 $d = 3$ 的两组例子, 数值秩的计算结果依然是令人满意的. 特别是, 对于 $d = 3$ 这组例子, 相比于表 3.1 中的例子, $\|\tilde{W}\|$ 基本没有明显的增长, 我们仍然有 $\|\tilde{W}\| = O(100)$.

第四章　快速求解 Sylvester 矩阵的结构低秩逼近问题

本章用快速算法求解了 Sylvester 矩阵的结构低秩逼近问题. 在文 [53] 中, 作者提出了一个算法, 通过构造给定多项式 Sylvester 矩阵的结构低秩逼近矩阵, 来计算给定多项式的极小扰动的多项式对, 其具有次数不低于给定正整数的准确最大公因子. 对这一算法来说, 计算量主要依赖于求解一系列最小二乘问题. 基于这些最小二乘问题系数阵的位移结构, 我们在本章提出一个快速求解算法, 其具有的复杂度比经典算法低一个数量级.

§4.1　前　言

一元多项式近似最大公因子的计算有着丰富的结果 [23, 28, 29, 41, 76, 24, 93, 53]. 两个一元多项式的近似最大公因子可以定义为这两个多项式给定邻域内的多项式的次数最高的准确最大公因子. 因此, 求解一元多项式的近似最大公因子问题可以先找到这对邻近的多项式, 然后求解其最大公因子.

在文 [53] 中作者提出了一个迭代算法, 通过求解给定多项式 Sylvester 矩阵的结构低秩逼近矩阵, 来计算给定多项式对的极小扰动的多项式对, 其具有次数不低于给定正整数的准确最大公因子. 这一算法可以直接应用于求解近似最大公因子. 对此, 我们可以应用第四章中提出的快速算法, 通过快速求解 Sylvester 矩阵的数值秩来设定一个有意义的, 文 [53] 中算法所需提前给定的正整数. 特别地, 可以通过快速计算 Sylvester 矩阵数值 ε- 秩满秩而提前测试两个多项式在 ε- 邻域内互素, 即在这一邻域内不存在具有非平凡最大公因子的多项式对; 而此时如果直接应用文 [53] 中算法 去计算近似最大公因子, 就会付出无意义的劳动.

对文 [53] 中算法来说, 计算量主要依赖于求解一系列最小二乘问题的运算量. 文 [53] 中应用经典的求解最小二乘问题的算法来实现这一过程, 其复杂度为 $O(st^2)$, 其中 s 和 t 分别代表最小二乘问题系数阵的行数和列数. 在本章中, 我们通过分析最小二乘问题系数阵的位移结构, 提出了一个具有平方阶复杂度 $O(s^2)$ 的快速算法.

本章其余各节内容安排如下：§4.2 我们简要介绍文 [53] 中求解 Sylvester 矩阵结构低秩逼近的算法以引出我们所要解决的问题；§4.3 首先我们分析了最小二乘问题系数阵的位移结构，然后提出求解最小二乘问题的快速算法，最后对算法进行向前误差分析；§4.4 我们用 Maple 软件实现了快速算法，在这一节中给出实验结果并与经典算法的结果进行比较.

§4.2 问题的提出

给出两个多项式 a, $b \in \mathbb{R}[x]$, 其中 $a = a_m x^m + \cdots + a_1 x + a_0$, $b = b_n x^n + \cdots + b_1 x + b_0$, $a_m \neq 0, b_n \neq 0$. 记 S 为 a 和 b 的 Sylvester 矩阵. a 和 b 的扰动分别记为：$\Delta a = \Delta a_m x^m + \cdots + \Delta a_1 x + \Delta a_0$, $\Delta b = \Delta b_n x^n + \cdots + \Delta b_1 x + \Delta b_0$. 现在考虑这样一个极小扰动问题：**极小化 $\|\Delta a\|_2^2 + \|\Delta b\|_2^2$，同时保证 $a + \Delta a$ 和 $b + \Delta b$ 具有次数大于等于给定正整数的最大公因子.**

给定 a 和 b 的 Sylvester 矩阵 S，消去 S 的下面 $k-1$ 行，再分别消去 S 中 a 和 b 的系数列的后 $k-1$ 列，得到的一个子矩阵称为第 k 个 Sylvester 子矩阵，记为 $S_k \in \mathbb{R}^{(m+n-k+1)\times(m+n-2k+2)}$：

$$S_k = \begin{array}{c} \left[\begin{array}{ccccccccccc} a_m & 0 & \cdots & 0 & 0 & b_n & 0 & \cdots & 0 & 0 \\ a_{m-1} & a_m & \cdots & 0 & 0 & b_{n-1} & b_n & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & a_0 & a_1 & 0 & 0 & \cdots & b_0 & b_1 \\ 0 & 0 & \cdots & 0 & a_0 & 0 & 0 & \cdots & 0 & b_0 \end{array}\right] \\ \underbrace{\hphantom{a_{m-1}\ a_m \cdots 0\ a_0}}_{n-k+1}\underbrace{\hphantom{b_{n-1}\ b_n \cdots 0\ b_0}}_{m-k+1} \end{array}. \tag{4.2.1}$$

现把 S_k 分块成：$S_k = [\mathbf{a}\ A_k]$, 其中 $\mathbf{a}$ 是 S_k 的第一列，A_k 由 S_k 的后 $m+n-2k+1$ 列构成.

两个多项式的扰动 Δa 和 Δb 可以由一个 $m+n+2$ 维的向量 $\mathbf{d}$ 来表示：

$$\mathbf{d} = [d_1, d_2, \ldots, d_{m+n+1}, d_{m+n+2}]^T.$$

那么与 $S_k = [\mathbf{a}, A_k]$ 相对应的，下面用 $[\Delta\mathbf{a}\, D_k]$ 来表示第 k 个 Sylvester 子矩阵的结构扰动：

$$[\Delta\mathbf{a}\, D_k] = \left[\begin{array}{ccccc|cccccc} d_1 & 0 & \cdots & 0 & 0 & d_{m+2} & 0 & \cdots & 0 & 0 \\ d_2 & d_1 & \cdots & 0 & 0 & d_{m+3} & d_{m+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & d_{m+1} & d_m & 0 & 0 & \cdots & d_{m+n+2} & d_{m+n+1} \\ 0 & 0 & \cdots & 0 & d_{m+1} & 0 & 0 & \cdots & 0 & d_{m+n+2} \end{array}\right], \qquad (4.2.2)$$

$$\underbrace{\qquad\qquad\qquad}_{n-k+1}\;\underbrace{\qquad\qquad\qquad\qquad}_{m-k+1}$$

这里 $\Delta\mathbf{a}$ 是此矩阵的第一列.

我们有下面的几个定理：

定理 4.2.1 *[37] 给出一元多项式 $a(x), b(x) \in \mathbb{R}[x]$, 其次数分别为 m 和 n. 设 $S(a,b)$ 是 $a(x)$ 和 $b(x)$ 的 Sylvester 矩阵，S_k 是 $a(x)$ 和 $b(x)$ 的第 k 个 Sylvester 子矩阵，其中 $1 \le k \le \min(m,n)$, 那么下面的两个论断是等价的：*

(1) $\mathrm{rank}(S) \le m+n-k$.

(2) S_k *的亏秩大于或等于 1.*

定理 4.2.2 *[53] 给出一元多项式 $a(x), b(x) \in \mathbb{R}[x]$, 其次数分别为 m 和 n. k 是一个正整数，$k \le \min(m,n)$. S_k 是 $a(x)$ 和 $b(x)$ 的第 k 个 Sylvester 矩阵. 把 S_k 分块成 $S_k = [\mathbf{a}\ A_k]$, 其中 $\mathbf{a}$ 是 S_k 的第一列，A_k 是 S_k 的后 $m+n-2k+1$ 列构成的子矩阵. 那么有：*

$$dim\ Nullspace(S_k) \ge 1 \Longleftrightarrow A_k\mathbf{x} = \mathbf{a}\ \text{有解}.$$

定理 4.2.3 *[53] 给出正整数 m, n 和 k, $k \le \min(m,n)$, 那么一定可以找到一个 Sylvester 矩阵 $S \in \mathbb{R}^{(m+n)\times(m+n)}$ 其秩为 $m+n-k$.*

根据上述定理，给出两个多项式 a 和 b 及一个正整数 k, 总可能找到一个第 k 个 Sylvester 子矩阵的结构扰动矩阵 $[\Delta\mathbf{a}\, D_k]$ 使得 $\mathbf{a}+\Delta\mathbf{a} \in \mathrm{Range}(A_k+D_k)$. 这样前面所述的多项式的极小扰动问题可以用下面的具有等式限制条件的最小二乘问题来描述：

$$\min_{\mathbf{d},\mathbf{x}} \|\mathbf{d}\|_2, \quad \text{使得}\ \mathbf{r} = 0, \qquad (4.2.3)$$

其中结构残数向量 $\mathbf{r}$ 由下式定义：

$$\mathbf{r} = \mathbf{a} + \Delta\mathbf{a} - (A_k + D_k)\mathbf{x}.$$

应用罚函数方法 [3], (4.2.3) 可以转化为

$$\min_{\mathbf{d},\mathbf{x}} \left\| \begin{bmatrix} w\mathbf{r} \\ \mathbf{d} \end{bmatrix} \right\|_2, \tag{4.2.4}$$

其中 w 是一个大的罚函数值，如 10^{10}.

把向量 $\Delta\mathbf{a}$ 和 $D_k\mathbf{x}$ 表示为

$$\Delta\mathbf{a} = P_k\,\mathbf{d},\ D_k\mathbf{x} = X_k\mathbf{d},$$

其中

$$P_k = \begin{bmatrix} I_{m+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix},$$

$$X_k = \begin{bmatrix} 0 & & & x_{n+1-k} & & \\ x_1 & \ddots & & x_{n+2-k} & \ddots & \\ \vdots & \ddots & 0 & \vdots & \ddots & x_{n+1-k} \\ x_{n-k} & & x_1 & x_{m+n+1-2k} & & x_{n+2-k} \\ & \ddots & \vdots & & \ddots & \vdots \\ & & x_{n-k} & & & x_{m+n+1-2k} \end{bmatrix},$$
$$\underbrace{\hphantom{x_{n-k}\quad\ddots\quad x_1}}_{m+1}\ \underbrace{\hphantom{x_{m+n+1-2k}\quad\ddots\quad x_{m+n+1-2k}}}_{n+1}$$

I_{m+1} 是一个阶数为 $m+1$ 的单位矩阵，(4.2.4) 转化成了下面的最小二乘问题：

$$\min_{\Delta\mathbf{x},\Delta\mathbf{d}} \left\| \begin{bmatrix} w(X_k - P_k) & w(A_k + D_k) \\ I_{m+n+2} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{d} \\ \Delta\mathbf{x} \end{bmatrix} + \begin{bmatrix} -w\mathbf{r} \\ \mathbf{d} \end{bmatrix} \right\|_2,$$

这里 I_{m+n+2} 是阶数为 $m+n+2$ 的单位矩阵.

文 [53] 中提出了下面的迭代的算法来求解多项式的极小扰动问题：

算法 AppSylv-k

输入 - 由多项式 $a(x)$ 和 $b(x)$ 生成的 Sylvester 矩阵 S，这里 $\deg(a)=m$, $\deg(b)=n$, 且 $m\geq n$; 给出一个整数 k, $1\leq k\leq n$ 和一个容许量 tol.

输出 - 两个多项式 $\tilde{a}$ 和 $\tilde{b}$, 满足 $\operatorname{rank}(S(\tilde{a},\tilde{b}))\leq m+n-k$, 且欧几里德距离 $\|\tilde{a}-a\|_2^2+\|\tilde{b}-b\|_2^2$ 被约化到极小值.

1. 构造第 k 个 Sylvester 子矩阵 S_k, 取 S_k 的第一列记为 $\mathbf{a}$, 取 S_k 的后 $m+n-2k+1$ 列形成矩阵 A_k. 设 $D_k=\mathbf{0}$, $\Delta\mathbf{a}=\mathbf{0}$.
2. 求解最小二乘问题 $\min\|A_k\mathbf{x}-\mathbf{a}\|_2$, 计算 $\mathbf{r}=\mathbf{a}-A_k\mathbf{x}$. 按照前面描述的方法构造 P_k 和 X_k.
3. 重复计算：
   (a) 求解最小二乘问题：
$$\min_{\Delta\mathbf{x},\,\Delta\mathbf{d}}\left\|\begin{bmatrix} w(X_k-P_k) & w(A_k+D_k)\\ I_{m+n+2} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\Delta\mathbf{d}\\ \Delta\mathbf{x}\end{bmatrix}+\begin{bmatrix}-w\mathbf{r}\\ \mathbf{d}\end{bmatrix}\right\|_2. \tag{4.2.5}$$
   (b) 设 $\mathbf{x}=\mathbf{x}+\Delta\mathbf{x}$, $\mathbf{d}=\mathbf{d}+\Delta\mathbf{d}$.
   (c) 用 $\mathbf{d}$ 构造矩阵 $[\Delta\mathbf{a}\,D_k]$, 用 $\mathbf{x}$ 构造矩阵 X_k. 设 $A_k=A_k+D_k$, $\mathbf{a}=\mathbf{a}+\Delta\mathbf{a}$, $\mathbf{r}=\mathbf{a}-A_k\mathbf{x}$.
   直到 $(\|\Delta\mathbf{x}\|_2\leq tol$ 且 $\|\Delta\mathbf{d}\|_2\leq tol)$
4. 输出多项式 $\tilde{a}$ 和 $\tilde{b}$: $\tilde{a}$ 和 $\tilde{b}$ 从 $[\mathbf{a}\,A_k]$ 构造得来.

可见, 算法 AppSylv-k 的复杂度依赖于算法第 3 步求解一系列形如 (4.2.5) 的最小二乘问题的复杂度. 设 (4.2.5) 中系数阵的行数和列数分别为 s,t, 那么 $s=2m+2n-k+3$, $t=2m+2n-2k+3$. 文 [53] 中应用经典的最小二乘求解算法，其复杂度为 $O(st^2)$. 我们在下一节中通过分析最小二乘问题系数阵的位移结构，提出一个具有平方阶复杂度 $O(s^2)$ 的快速算法.

给定一个正数 ε, 文 [53] 中把算法 AppSylv-k 继续用于计算两个多项式 $a(x)$ 和 $b(x)$ 的 ε- 最大公因子, 这里假设 $a(x)$ 和 $b(x)$ 的次数分别为 m 和 n, 并且 $m\geq n$. 基本想法是，从 $k=n\leq m$ 开始，应用算法 AppSylv-k 来计算极小扰动多项式 $\tilde{a},\tilde{b}$ 及 $\aleph=\sqrt{\|\tilde{a}-a\|_2^2+\|\tilde{b}-b\|_2^2}$, 其中 $\operatorname{rank}(S(\tilde{a},\tilde{b}))\leq m+n-k$. 如果 $\aleph\leq\varepsilon$, 那么从矩阵 $S_k(\tilde{a},\tilde{b})$ 出发依 [37, 93] 中的方法计算 ε- 最大公因子; 否则，把 k 值减 1 重复上述

过程. 文 [53] 中给出的另一个方法是，根据文 [23] 中的结果，通过计算 Sylvester 矩阵 $S(a,b)$ 的奇异值分解找到一个 ε- 最大公因子的次数上界 d. 这样，可以从 $k=d$ 开始，而不必从 $k=n$ 开始去计算 ε- 最大公因子. 根据本文第三章的结果，我们可以应用算法 HSylRRA 求解 Sylvester 矩阵的 ε- 秩而快速得到 ε- 最大公因子的一个次数上界.

§4.3 求解最小二乘问题的快速算法

1. 系数阵的位移结构

现在我们记最小二乘系统 (4.2.5) 的系数矩阵为 M,

$$M=\begin{bmatrix} w(X_k-P_k) & w(A_k+D_k) \\ I_{m+n+2} & \mathbf{0} \end{bmatrix},$$

同时简记未知向量和右端向量为：$\mathbf{y}=\begin{bmatrix}\Delta\mathbf{d}\\ \Delta\mathbf{x}\end{bmatrix}$, $\mathbf{z}=\begin{bmatrix}w\mathbf{r}\\ -\mathbf{d}\end{bmatrix}$, 那么 (4.2.5) 变成了

$$\min_{\mathbf{y}}\|M\mathbf{y}-\mathbf{z}\|_2\,. \tag{4.3.1}$$

为了找到一个求解最小二乘问题 (4.3.1) 的快速算法，我们首先分析这个系统系数矩阵的位移结构：

定理 4.3.1 M 是一个位移秩不超过 4 的位移结构矩阵：关于移位块矩阵

$$F_1=\operatorname{diag}\left(Z_{m+n-k+1},Z_{m+n+2}\right),\ F_2=\operatorname{diag}\left(Z_{m+1},Z_{n+1},Z_{n-k},Z_{m-k+1}\right),$$

我们有

$$M-F_1MF_2^T=\left[\mathbf{u}_1,\mathbf{u}_2,\mathbf{u}_3,\mathbf{u}_4\right]\left[\mathbf{e}_1,\mathbf{e}_{m+2},\mathbf{e}_{m+n+3},\mathbf{e}_{2n+m-k+3}\right]^T,$$

其中 $\mathbf{e}_i$ 代表 $2m+2n-2k+3$ 阶单位矩阵的第 i 列，而且

$$\begin{aligned} \mathbf{u}_1 &= M(:,1), & \mathbf{u}_2 &= M(:,m+2), \\ \mathbf{u}_3 &= M(:,m+n+3), & \mathbf{u}_4 &= M(:,2n+m-k+3). \end{aligned}$$

对于系数阵是 Toeplitz 矩阵的最小二乘问题，已有许多基于快速 QR 分解的快速算法，如 [25, 67, 21, 6, 26, 80, 84]. 但这些算法的数值稳定性问题一直都有所争议，

对于系数阵是病态矩阵的最小二乘问题，大多数算法在求解的精度上都受到很大影响. 基于修正的半正规方程方法，文献 [78] 中提出了一个新的快速 QR 分解的算法，通过这个算法算得的 QR 分解中的 R 因子具有更好的精确性，甚至对于某些相当病态的 Toeplitz 矩阵这一算法同样适用. 因此，相比于原有的那些最小二乘问题的快速求解算法，这一算法可以应用于求解更广范围的最小二乘问题. 然而，对于一些象 (4.3.1) 这样的，系数阵有超过三个的很小的奇异值的最小二乘问题，[78] 中的算法可能也会失去作用 [5, 78]. 除了上面提到的，文献 [33] 中也提出了一个最小二乘问题的快速求解算法，这个算法是基于一个求解 Cauchy 型最小二乘问题的快速算法 [34, 51, 31, 42] 之上的. 关于这样的算法是否适合于求解 (4.3.1) 这样的最小二乘系统，现在还在研究中. 在本章中，我们推广文 [20] 中求解 Toeplitz 线性方程组的快速算法，得到了一个求解最小二乘问题 (4.3.1) 的快速算法.

2. 求解最小二乘问题 (4.3.1) 的快速算法描述

对于最小二乘问题 (4.3.1)，我们记 $\mathbf{y}_{\mathrm{LS}}$ 为其最小二乘解，记 $\mathbf{r}_{LS}$ 为最小残数向量：$\mathbf{r}_{LS} = M\mathbf{y}_{\mathrm{LS}} - \mathbf{z}$, 那么 $\mathbf{y}_{LS}$ 适合下面这个线性方程组：

$$\begin{bmatrix} M^TM & M^T \\ M & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ -\mathbf{z} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{z} + \mathbf{r}_{LS} \end{bmatrix}. \tag{4.3.2}$$

现记 Q 为 M 的 QR 分解中的正交矩阵，我们把 Q 分块成：$Q = [Q_1, Q_2]$, 其中 $Q_2 \in \mathbb{R}^{(2m+2n-k+3)\times k}$, 那么

$$\|\mathbf{r}_{LS}\|_2 = \|Q_2^T\mathbf{z}\|_2 = \left\| Q_2^T \begin{bmatrix} w\mathbf{r} \\ -\mathbf{d} \end{bmatrix} \right\|_2. \tag{4.3.3}$$

当矩阵 M 的上半块矩阵 $M(1..m+n-k+1,:)$ 占有大的权重（如 $O(10^{10})$）时, 正交矩阵 Q 的子块 $Q_2(m+n-k+2..2m+2n-k+3,:)$ 中的元素都是 $O(1)$ 的，而子块 $Q_2(1..m+n-k+1,:)$ 中的元素都是接近于零的. 因此，由 (4.3.3) 推导知，$\|\mathbf{r}_{LS}\|_2$ 相比于 $\|\mathbf{z}\|_2$ 来说要小得多，即：

$$\|\mathbf{r}_{LS}\|_2 \ll \|\mathbf{z}\|_2 .$$

这启发我们可以利用下面的系统去计算 (4.3.1) 的一个近似最小二乘解：

$$\begin{bmatrix} M^TM & M^T \\ M & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ -\mathbf{z} \end{bmatrix} \approx \begin{bmatrix} \mathbf{0} \\ \mathbf{z} \end{bmatrix}. \tag{4.3.4}$$

这样自然涉及到系统 (4.3.4) 的系数阵的三角分解，对此，我们将应用广义 Schur 算法来实现.

现把 M 和 $\mathbf{z}$ 进行规范化：

$$M := M/\|M\|_F,\ s.t.\ \|M\| \leq 1, \tag{4.3.5}$$

$$\mathbf{z} := \mathbf{z}/\|M\|_F, \tag{4.3.6}$$

其中 $\|M\|_F$ 是 M 的 Frobenius 范数. 由于大的罚函数值 w, 在做了规范化之后，M 的左下角子块（原来是单位阵 I_{m+n+2}）将由接近于零的元素构成. 这必然导致 M^TM 的数值秩亏. 同时，注意到 M 不是方阵，系统 (4.3.4) 的系数阵总是奇异的. 为了完成广义 Schur 算法，我们构造一个邻近的矩阵 T：

$$T = \begin{bmatrix} M^TM + \alpha I^{(1)} & M^T \\ M & -\beta I^{(2)} \end{bmatrix}, \tag{4.3.7}$$

其中 $\alpha I^{(1)}, \beta I^{(2)}$ 是两个单位阵乘以一个小的倍数得到，以使得 $M^TM + \alpha I^{(1)}$ 是充分正定的，这将保证广义 Schur 算法正步的顺利进行；同时，使得 T 关于主子矩阵 $M^TM + \alpha I^{(1)}$ 的 Schur 补是负定的，这将保证广义 Schur 算法负步的顺利进行.

我们把 T 的位移结构揭示如下：

定理 4.3.2 T 是一个位移结构矩阵，关于位移 $T - ZTZ^T$ 其位移秩不超过 10, 这里

$$Z = \operatorname{diag}\left(Z_{m+1}, Z_{n+1}, Z_{n-k}, Z_{m-k+1}, Z_{m+n-k+1}, Z_{m+n+2}\right).$$

事实上，我们可以构造一个 T 的生成子对 (G, J), 使得

$$T - ZTZ^T = GJG^T,$$

这里 $J = \operatorname{diag}(I_4, -I_6)$, 而 G 是一个 $(4m+4n-3k+6) \times 10$ 的矩阵. 我们在文 [58] 中，把生成子矩阵 G 直接用 T 的列来表示，这样构造的 G, 某些元素的数量级是 $1/w^2$, w

是一个大的罚函数值. 这从数值稳定的角度来说，是不合适的. 下面我们构造一个新的生成子矩阵，其元素与 M 中元素有相同的数量级，仅为 $1/w$. 现用 $\mathbf{g}_i, i=1,\ldots,10$ 来代表这个生成子矩阵 G 的列. 定义 $t_1,\ldots,t_4$ 为：

$$t_1=\|M(:,1)\|_2^2+\alpha,\qquad t_2=\|M(:,m+2)\|_2^2+\alpha,$$
$$t_3=\|M(:,m+n+3)\|_2^2+\alpha,\ t_4=\|M(:,2n+m-k+3)\|_2^2+\alpha.$$

设

$$\begin{aligned}
\mathbf{c}_1 &= \left[M^T(1,:)M,\ M^T(1,:)\right]^T+\alpha I(:,1),\\
\mathbf{c}_2 &= \left[M^T(m+2,:)M,\ M^T(m+2,:)\right]^T+\alpha I(:,m+2),\\
\mathbf{c}_3 &= \left[M^T(m+n+3,:)M,\ M^T(m+n+3,:)\right]^T+\alpha I(:,m+n+3),\\
\mathbf{c}_4 &= \left[M^T(2n+m-k+3,:)M,\ M^T(2n+m-k+3,:)\right]^T+\alpha I(:,2n+m-k+3),
\end{aligned}$$

其中 I 代表 $4m+4n-3k+6$ 阶的单位矩阵. 那么

$$\begin{aligned}
\mathbf{g}_1 &= \mathbf{c}_1/\sqrt{t_1},\\
\mathbf{g}_2 &= \mathbf{c}_2/\sqrt{t_2},\quad \text{除了}\quad \mathbf{g}_2[1]=0,\\
\mathbf{g}_3 &= \mathbf{c}_3/\sqrt{t_3},\quad \text{除了}\ \mathbf{g}_3[1]=0,\ \mathbf{g}_3[m+2]=0,\\
\mathbf{g}_4 &= \mathbf{c}_4/\sqrt{t_4},\quad \text{除了}\ \mathbf{g}_4[1]=0,\ \mathbf{g}_4[m+2]=0,\ \mathbf{g}_4[m+n+3]=0,\\
\mathbf{g}_5 &= \left[0,\mathbf{g}_1^T(2:4m+4n-3k+6)\right]^T,\\
\mathbf{g}_6 &= \left[\mathbf{g}_2^T(1:m+1),0,\mathbf{g}_2^T(m+3:4m+4n-3k+6)\right]^T,\\
\mathbf{g}_7 &= \left[\mathbf{g}_3^T(1:m+n+2),0,\mathbf{g}_3^T(m+n+4:4m+4n-3k+6)\right]^T,\\
\mathbf{g}_8 &= \left[\mathbf{g}_4^T(1:2n+m-k+2),0,\mathbf{g}_4^T(2n+m-k+4:4m+4n-3k+6)\right]^T,\\
\mathbf{g}_9 &= [\ \underbrace{0,\ \ldots,\ 0,\ 0}_{2m+2n-2k+3},\ \sqrt{\beta},\ 0,\ \ldots,\ 0\ ]^T,\\
\mathbf{g}_{10} &= [\ \underbrace{0,\ \ldots,\ 0,\ 0}_{3m+3n-3k+4},\ \sqrt{\beta},\ 0,\ \ldots,\ 0\ ]^T.
\end{aligned}$$

应用 M 的奇异值分解，循 [20] 中的思路可以证明，通过作用于生成子对 (G, J)，完成广义 Schur 算法的 $2m+2n-2k+3$ 个正步和 $2m+2n-k+3$ 个负步后，我们

可以得到 T 的向后稳定的三角分解:

$$\begin{bmatrix} \hat{R}^T & 0 \\ \hat{Q} & \hat{D} \end{bmatrix} \begin{bmatrix} \hat{R} & \hat{Q}^T \\ 0 & -\hat{D}^T \end{bmatrix}, \tag{4.3.8}$$

这里 $\hat{R}$ 是上三角矩阵, $\hat{D}$ 是下三角矩阵. 进一步的, 应用得到的三角分解 (4.3.8), 我们可以求解下列增广系统

$$T \begin{bmatrix} \mathbf{y} \\ \xi \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{z} \end{bmatrix},$$

而得到

$$(T+H) \begin{bmatrix} \hat{\mathbf{y}} \\ \hat{\xi} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{z} \end{bmatrix}, \tag{4.3.9}$$

这里 $\|H\| = O(u)$, u 代表机器精度. 解向量 $\begin{bmatrix} \hat{\mathbf{y}} \\ \hat{\xi} \end{bmatrix}$ 通过下列回代步得来:

$$\begin{bmatrix} \hat{R} & \check{Q}^T \\ 0 & -\hat{D}^T \end{bmatrix}^{-1} \begin{bmatrix} \hat{R}^T & 0 \\ \hat{Q} & \hat{D} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{0} \\ \mathbf{z} \end{bmatrix},$$

而 $\hat{\mathbf{y}}$ 根据下式计算得来:

$$\hat{R}^{-1}\hat{Q}^T\hat{D}^{-T}\hat{D}^{-1}\mathbf{z}. \tag{4.3.10}$$

我们就把 $\hat{\mathbf{y}}$ 看作是系统 (4.3.1) 的一个最小二乘近似解.

在应用广义 Schur 算法来计算 T 的三角分解时, 假设, 在每个递归步中我们构造 J- 正交矩阵为两个 Householder 旋转变换和一个初等 Hyperbolic 旋转的复合, 并且这个初等 Hyperbolic 旋转的实现我们采用 OD 形式 (见算法 2.1.1). 这样, 按上述过程计算三角分解所需的浮点运算次数为:

$$\frac{47}{2}(4m+4n-3k+6)^2 + \frac{121}{2}(4m+4n-3k+6).$$

计算近似解 $\hat{\mathbf{y}}$(4.3.10) 需要用到两个向后回代, 一个向前回代, 和一个矩阵向量乘法, 因而所需浮点运算次数为 [30]:

$$2(2m+2n-k+3)^2 + (2m+2n-2k+3)^2 + 2(2m+2n-k+3)(2m+2n-2k+3).$$

这样，在去掉低阶项后，整个算法需要

$$\frac{51}{2}s^2 + 49st + \frac{49}{2}t^2$$

次浮点运算，其中 $s = 2m + 2n - k + 3$, $t = 2m + 2n - 2k + 3$. 相比之下，[53] 中的求解需要 $2st^2 - \frac{2}{3}t^3$ 次浮点运算. 所以，本文的实现方法具有更低的复杂度，特别是当 k 不是很大的时候.

注 4.3.3 在构造生成子矩阵 G 时，利用 M 的位移结构，可以把计算 $\mathbf{c}_i, i = 1, \ldots, 4$ 时的矩阵向量乘法转化成卷积的计算，进而利用快速傅立叶变换 FFT 得以快速实现. 事实上，根据定理 *4.3.1,*

$$M^T - F_2 M^T F_1^T = [\mathbf{e}_1, \mathbf{e}_{m+2}, \mathbf{e}_{m+n+3}, \mathbf{e}_{2n+m-k+3}][\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4]^T. \tag{4.3.11}$$

注意到 F_1 和 F_2 是幂零矩阵，我们有

$$M^T = L_1 N_1^T + L_2 N_2^T + L_3 N_3^T + L_4 N_4^T,$$

这里

$$L_j = [\mathbf{v}_j, F_2\mathbf{v}_j, \ldots, F_2^m \mathbf{v}_j, \mathbf{0}, \ldots, \mathbf{0}],\ N_j = \left[\mathbf{u}_j, F_1\mathbf{u}_j, \ldots, F_1^{m+n-k}\mathbf{u}_j\right],$$

其中 $\mathbf{v}_j (j = 1, \ldots, 4)$ 依次代表 *(4.3.11)* 中的四个单位向量. 这样，矩阵 M^T 与向量的乘法可以转化成几个 *Toeplitz* 矩阵与向量的乘法运算，因而可以转化成卷积计算.

3. 误差分析

本节我们为近似最小二乘解 $\hat{\mathbf{y}}$ 推导出一个相对向前误差界. 本节中 $\kappa(\cdot)$ 代表一个矩阵的 2- 范数条件数. 定义 $l = m + n - k + 1$, $l' = m + n - k + 2$. 对任意整数 i, $1 \le i \le l + l'$, 我们简记 σ_i 为 M 的第 i 大的奇异值.

引理 4.3.4 设 $\mathbf{f}$ 是一个满足下式的向量：

$$\begin{bmatrix} M^T M + \alpha I^{(1)} & M^T \\ M & -\beta I^{(2)} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{y}} \\ \hat{\xi} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{z} \end{bmatrix} + \mathbf{f}, \tag{4.3.12}$$

我们为 $\frac{\|\hat{\mathbf{y}} - \mathbf{y}_{LS}\|_2}{\|\mathbf{y}_{LS}\|_2}$ 推导出一个上界，其近似的形式为：

$$\frac{\beta + \alpha\beta\kappa^2(M)}{\|M\|^2 - \alpha\beta\kappa^2(M)} + \frac{\kappa(M) + \beta\kappa^2(M)}{\|M\|^2 - \alpha\beta\kappa^2(M)} \frac{\|\mathbf{f}\|_2}{\|\mathbf{y}_{LS}\|_2}. \tag{4.3.13}$$

证明 把 $\mathbf{f}$ 分块为：$\left[\mathbf{f}_1^T, \mathbf{f}_2^T\right]^T$, $\mathbf{f}_1 \in \mathbb{R}^{(2m+2n-2k+3)\times 1}$, $\mathbf{f}_2 \in \mathbb{R}^{(2m+2n-k+3)\times 1}$, 那么从 (4.3.12) 我们有：

$$\begin{cases} \left(M^T M + \alpha I^{(1)}\right)\hat{\mathbf{y}} + M^T \hat{\xi} = \mathbf{f}_1, \\ M\hat{\mathbf{y}} - \beta\hat{\xi} = \mathbf{z} + \mathbf{f}_2. \end{cases}$$

消去 $\hat{\xi}$ 我们得到

$$\beta\left(M^T M + \alpha I^{(1)}\right)\hat{\mathbf{y}} + M^T M\hat{\mathbf{y}} = M^T\mathbf{z} + M^T\mathbf{f}_2 + \beta\mathbf{f}_1. \tag{4.3.14}$$

注意到

$$M^T\mathbf{z} = M^T M\mathbf{y}_{LS},$$

通过简单的计算我们有：

$$\left(\beta M^T M + \alpha\beta I^{(1)} + M^T M\right)(\mathbf{y}_{LS} - \hat{\mathbf{y}}) = \left(\beta M^T M + \alpha\beta I^{(1)}\right)\mathbf{y}_{LS} - M^T\mathbf{f}_2 - \beta\mathbf{f}_1, \tag{4.3.15}$$

以及

$$\begin{aligned} \mathbf{y}_{LS} - \hat{\mathbf{y}} = & \left(\beta I^{(1)} + \alpha\beta(M^T M)^{-1} + I^{(1)}\right)^{-1}\left(\beta I^{(1)} + \alpha\beta(M^T M)^{-1}\right)\mathbf{y}_{LS} \\ & - \left(\beta I^{(1)} + \alpha\beta(M^T M)^{-1} + I^{(1)}\right)^{-1}\left(M^\dagger\mathbf{f}_2 + \beta(M^T M)^{-1}\mathbf{f}_1\right). \end{aligned}$$

于是，

$$\begin{aligned} \frac{\|\mathbf{y}_{LS} - \hat{\mathbf{y}}\|_2}{\|\mathbf{y}_{LS}\|_2} &\le \frac{\beta + \alpha\beta\left\|(M^T M)^{-1}\right\|}{1 - \beta - \alpha\beta\left\|(M^T M)^{-1}\right\|} + \frac{\left\|M^\dagger\right\| + \beta\left\|(M^T M)^{-1}\right\|}{1 - \beta - \alpha\beta\left\|(M^T M)^{-1}\right\|}\frac{\|\mathbf{f}\|_2}{\|\mathbf{y}_{LS}\|_2} \\ &= \frac{\beta\|M\|^2 + \alpha\beta\kappa^2(M)}{(1-\beta)\|M\|^2 - \alpha\beta\kappa^2(M)} + \frac{\kappa(M)\|M\| + \beta\kappa^2(M)}{(1-\beta)\|M\|^2 - \alpha\beta\kappa^2(M)}\frac{\|\mathbf{f}\|_2}{\|\mathbf{y}_{LS}\|_2}, \end{aligned}$$

这里我们假设

$$\alpha\beta\|(M^T M)^{-1}\| < 1, \tag{4.3.16}$$

并用到下面等式：

$$\left\|M^\dagger\right\|\|M\| = \kappa(M),\ \left\|(M^T M)^{-1}\right\|\|M\|^2 = \kappa^2(M).$$

注意到

$$\|M\| \le 1,\ \beta \ll 1,$$

我们得到了引理中最后的不等式. ■

引理 4.3.5 把 $\mathbf{z}$ 划分成：$\left[\mathbf{z}_l^T, \mathbf{z}_{l'+k}^T\right]^T$，$\mathbf{z}_l \in \mathbb{R}^{l\times 1}$, $\mathbf{z}_{l'+k} \in \mathbb{R}^{(l'+k)\times 1}$，那么对于 *(4.3.7)* 中所定义的 T 我们有：

$$\left\| T^{-1}\begin{bmatrix}\mathbf{0}\\ \mathbf{z}\end{bmatrix}\right\|_2 \leq \left(\frac{4}{\sigma_l}+\frac{2}{w\beta}+\frac{1}{w\sigma_{l+l'}}\right)\|\mathbf{z}_l\|_2+\left(\frac{2}{\beta}+\frac{3}{\sigma_{l+l'}}\right)\|\mathbf{z}_{l'+k}\|_2 .$$

证明 设

$$T^{-1}=\begin{bmatrix}B_{11} & B_{12}\\ B_{12}^T & B_{22}\end{bmatrix},$$

那么

$$\begin{aligned}
B_{11} &= \beta M^{\dagger}\left(M\left[M^TM+\alpha I^{(1)}\right]^{-1}M^T+\beta I^{(2)}\right)^{-T}M\left(M^TM+\alpha I^{(1)}\right)^{-1},\\
B_{12} &= \left(M^TM+\alpha I^{(1)}\right)^{-1}M^T\left(M\left[M^TM+\alpha I^{(1)}\right]^{-1}M^T+\beta I^{(2)}\right)^{-1},\\
B_{22} &= -\left(M\left[M^TM+\alpha I^{(1)}\right]^{-1}M^T+\beta I^{(2)}\right)^{-1}.
\end{aligned}$$

令 $M=U\begin{bmatrix}\Sigma\\ \mathbf{0}_{k\times(l+l')}\end{bmatrix}V^T$ 是 M 的奇异值分解，这里 U,V 都是正交矩阵，Σ 是一个由 M 的奇异值所构成的对角矩阵. Σ 可以写成：

$$\Sigma=\begin{bmatrix}\Sigma_l & \\ & \Sigma_{l'}\end{bmatrix},$$

其中

$$\Sigma_l=\operatorname{diag}\left(\sigma_1,\ldots,\sigma_l\right),\ \Sigma_{l'}=\operatorname{diag}\left(\sigma_{l+1},\ldots,\sigma_{l+l'}\right).$$

把 U,V 分块成

$$U=\begin{bmatrix}U_{11} & U_{12}\\ U_{21} & U_{22}\end{bmatrix},\ V=\left[V_1,V_2\right],$$

其中 $U_{11}\in\mathbb{R}^{l\times l}$, $V_1\in\mathbb{R}^{(l+l')\times l}$. 再定义对角矩阵

$$\begin{aligned}
\Lambda_l &= \left(\left[I_l+\alpha\Sigma_l^{-2}\right]^{-1}+\beta I_l\right)^{-1},\\
\Lambda_l' &= \left(\Sigma_l+\alpha\Sigma_l^{-1}\right)^{-1}\Lambda_l,\\
\Lambda_{l'} &= \left(\Sigma_{l'}+\alpha\Sigma_{l'}^{-1}\right)^{-1}\left(\left[I_{l'}+\alpha\Sigma_{l'}^{-2}\right]^{-1}+\beta I_{l'}\right)^{-1},\\
\Lambda_{l'+k} &= \begin{bmatrix}\left(\left[I_{l'}+\alpha\Sigma_{l'}^{-2}\right]^{-1}+\beta I_{l'}\right)^{-1} & \\ & \frac{1}{\beta}I_k\end{bmatrix},
\end{aligned}$$

这里 $I_l, I_{l'}, I_k$ 分别代表阶数为 l, l', k 的单位矩阵. 那么可以验证有

$$B_{12} = \left[V_1\Lambda_l' U_{11}^T + V_2\left[\Lambda_{l'}\ \mathbf{0}\right]U_{12}^T \quad V_1\Lambda_l' U_{21}^T + V_2\left[\Lambda_{l'}\ \mathbf{0}\right]U_{22}^T\right],$$
$$B_{22} = -\begin{bmatrix} U_{11}\Lambda_l U_{11}^T + U_{12}\Lambda_{l'+k}U_{12}^T & U_{11}\Lambda_l U_{21}^T + U_{12}\Lambda_{l'+k}U_{22}^T \\ U_{21}\Lambda_l U_{11}^T + U_{22}\Lambda_{l'+k}U_{12}^T & U_{21}\Lambda_l U_{21}^T + U_{22}\Lambda_{l'+k}U_{22}^T \end{bmatrix}.$$

容易证明下面的不等式成立:

$$\|\Lambda_l\|_2 \le 1 + 1/\sigma_l,\ \|\Lambda_l'\|_2 \le 1/\sigma_l,\ \|\Lambda_{l'}\|_2 \le 1/\sigma_{l+l'},\ \|\Lambda_{l'+k}\|_2 \le 1 + 1/\beta, \tag{4.3.17}$$

在最后一个不等式的证明中我们假设:

$$\alpha\beta < \sigma_{l+l'}^2, \quad \text{亦即} \quad \alpha\beta\|(M^TM)^{-1}\| < 1.$$

同时, 当 M 矩阵的上半子块占有大的权重 $(O(w))$ 时, 我们有下面的不等式:

$$\|U_{12}\|, \|U_{21}\| \le 1/w.$$

这样我们得到了

$$\begin{aligned}\left\| T^{-1}\begin{bmatrix}\mathbf{0}\\ \mathbf{z}\end{bmatrix}\right\|_2 &\le \|B_{12}\mathbf{z}\|_2 + \|B_{22}\mathbf{z}\|_2 \\ &\le \left(\frac{4}{\sigma_l} + \frac{2}{w\beta} + \frac{1}{w\sigma_{l+l'}}\right)\|\mathbf{z}_l\|_2 + \left(\frac{2}{\beta} + \frac{3}{\sigma_{l+l'}}\right)\|\mathbf{z}_{l'+k}\|_2 .\end{aligned}$$

■

引理 4.3.6 *如前面所定义, 设 H 是满足 (4.3.9) 的向后误差矩阵, 并假定 $\gamma\|H\| < 1$, 其中*

$$\gamma = \beta/\sigma_{l+l'}^2 + 2/\sigma_{l+l'} + 1/\beta + 1,$$

那么对于引理 4.3.4 中所定义的向量 $\mathbf{f}$ 我们有:

$$\|\mathbf{f}\|_2 \le \frac{\|H\|}{1-\gamma\|H\|}\left[\left(\frac{4}{\sigma_l} + \frac{2}{w\beta} + \frac{1}{w\sigma_{l+l'}}\right)\|\mathbf{z}_l\|_2 + \left(\frac{2}{\beta} + \frac{3}{\sigma_{l+l'}}\right)\|\mathbf{z}_{l'+k}\|_2\right].$$

证明 由 $\mathbf{f}$ 和 H 的定义，我们可以得到：

$$\begin{aligned}\mathbf{f} &= -H\begin{bmatrix}\hat{\mathbf{y}}\\ \hat{\xi}\end{bmatrix}\\ &= -H\,(T+H)^{-1}\begin{bmatrix}\mathbf{0}\\ \mathbf{z}\end{bmatrix}\\ &= -H\left(I+T^{-1}H\right)^{-1}T^{-1}\begin{bmatrix}\mathbf{0}\\ \mathbf{z}\end{bmatrix}.\end{aligned}$$

于是，

$$\begin{aligned}\|\mathbf{f}\|_2 &\le \frac{\|H\|}{1-\|T^{-1}H\|}\left\|T^{-1}\begin{bmatrix}\mathbf{0}\\ \mathbf{z}\end{bmatrix}\right\|_2\\ &\le \frac{\|H\|}{1-\|T^{-1}\|\,\|H\|}\left\|T^{-1}\begin{bmatrix}\mathbf{0}\\ \mathbf{z}\end{bmatrix}\right\|_2,\end{aligned}$$

这里假设 $\|T^{-1}\|\|H\|<1$.

注意到

$$\begin{aligned}B_{11} &= V\begin{bmatrix}\beta\Sigma_l^{-1}\Lambda_l' & \\ & \beta\Sigma_{l'}^{-1}\Lambda_{l'}\end{bmatrix}V^T,\\ B_{12} &= V\begin{bmatrix}\Lambda_l' & & \mathbf{0}_{l\times k}\\ & \Lambda_{l'} & \mathbf{0}_{l'\times k}\end{bmatrix}U^T,\\ B_{22} &= -U\begin{bmatrix}\Lambda_l & \\ & \Lambda_{l'+k}\end{bmatrix}U^T,\end{aligned}$$

从 (4.3.17) 我们有

$$\begin{aligned}\left\|T^{-1}\right\| &\le \|B_{11}\|+\|B_{12}\|+\|B_{22}\|\\ &\le \beta/\sigma_{l+l'}^2+2/\sigma_{l+l'}+1/\beta+1.\end{aligned}$$

再应用引理 4.3.5 及假设 $\gamma\|H\|<1$，我们得到了本引理中的不等式. ■

定理 4.3.7 设 $\hat{\mathbf{y}}$ 是 *(4.3.9)* 中线性系统的解向量:

$$(T+H)\begin{bmatrix}\hat{\mathbf{y}}\\ \hat{\xi}\end{bmatrix}=\begin{bmatrix}\mathbf{0}\\ \mathbf{z}\end{bmatrix},$$

假定 $\alpha\beta<\sigma_{l+l'}^2$ 及 $\gamma\|H\|<1$, 我们可以得到向前相对误差 $\frac{\|\hat{\mathbf{y}}-\mathbf{y}_{LS}\|_2}{\|\mathbf{y}_{LS}\|_2}$ 的一个上界, 形为:

$$\frac{\beta+\alpha\beta\kappa^2(M)}{\|M\|^2-\alpha\beta\kappa^2(M)}+\frac{\kappa(M)+\beta\kappa^2(M)}{\|M\|^2-\alpha\beta\kappa^2(M)}\frac{\|H\|\ \delta}{(1-\gamma\|H\|)\|\mathbf{y}_{LS}\|_2}, \tag{4.3.18}$$

这里

$$\begin{aligned}\gamma &= \beta/\sigma_{l+l'}^2+2/\sigma_{l+l'}+1/\beta+1,\\ \delta &= \left(\frac{4}{\sigma_l}+\frac{2}{w\beta}+\frac{1}{w\sigma_{l+l'}}\right)\|\mathbf{z}_l\|_2+\left(\frac{2}{\beta}+\frac{3}{\sigma_{l+l'}}\right)\|\mathbf{z}_{l'+k}\|_2.\end{aligned}$$

证明 此结论可直接应用引理 4.3.4, 4.3.5 和 4.3.6 得到. ■

注 4.3.8 为了顺利完成广义 *Schur* 算法我们所取的 α,β 应满足 *[20]* 中所提出的下界. 本质上, α,β 可以取得比机器精度稍微大一些. 除此之外, 基于本节中的误差分析, α,β 的选取应使得误差上界 *(4.3.18)* 尽可能的小. 从这一角度说, 定理 *4.3.7* 中所用到的假设 $\alpha\beta<\sigma_{l+l'}^2$ 是一个自然的假设, 因为这个假设近似等价于误差上界 *(4.3.18)* 小于 1 的一个必要条件: $\alpha\beta\kappa^2(M)<1$.

注 4.3.9 在实际应用中, 在经过正规化 *(4.3.5)* 后, 下面的不等式成立:

$$\|\mathbf{z}_l\|_2<1,\ \|\mathbf{z}_{l'+k}\|_2<1/w;$$

进一步地, 当所给的整数 k *(4.2.5)* 是两个多项式的近似最大公因子的次数上界时, 通常我们有

$$\delta<1.$$

§4.4 实验结果

我们用 Maple9.5 实现了本章所提出的快速算法. 在下表中, 我们给出一些实验结果. 我们把本文中所描述的求解最小二乘问题 (4.3.1) 的快速算法嵌入到文 [53] 中的算法 AppSylv-k, 来计算给定的两个多项式具有次数不低于给定正整数的最大公因

表 4.1: 数值结果比较

Ex	*m,n*	*k*	*error* (*classic*)	*error* (*new fast*)	*cond.num* (*LS Prob.*)	*for.err.* (*LS Prob.*)	$\hat{\varepsilon}(\hat{\mathbf{y}})$ (*LS Prob.*)
1	$2,2$	1	$5.59933e{-}3$	$5.59933e{-}3$	$0.109e13$	$0.461e{-}3$	$0.238e{-}12$
2	$3,3$	2	$1.07129e{-}2$	$1.07129e{-}2$	$0.148e13$	$0.434e{-}3$	$0.176e{-}13$
3	$5,4$	3	$1.56146e{-}6$	$1.56146e{-}6$	$0.209e13$	$0.143e{-}2$	$0.143e{-}13$
4	$5,5$	3	$1.34138e{-}8$	$1.34318e{-}8$	$0.232e13$	$0.664e{-}3$	$0.452e{-}13$
5	$6,6$	4	$1.96333e{-}10$	$1.96333e{-}10$	$0.239e13$	$0.182e{-}3$	$0.448e{-}13$
6	$8,7$	4	$1.98415e{-}16$	$1.98416e{-}16$	$0.312e13$	$0.322e{-}3$	$0.896e{-}14$
7	$10,10$	5	$1.51551e{-}12$	$1.51552e{-}12$	$0.545e13$	$0.272e{-}2$	$0.598e{-}13$
8	$14,13$	7	$2.61818e{-}4$	$2.61819e{-}4$	$0.697e13$	$0.163e{-}1$	$0.112e{-}12$
9	$28,28$	10	$2.54575e{-}4$	$3.54600e{-}4$	$0.151e14$	$0.992e{-}1$	$0.512e{-}14$
10	$50,50$	30	$9.35252e{-}6$	$9.40237e{-}6$	$0.302e14$	0.134	$0.168e{-}13$

子所需要的极小扰动. 同时，与算法 AppSylv-k 的计算结果进行比较. 这里我们设定：Digits $= 15$, $\alpha = \beta = 10^{-14}$.

数值实验中，所测试的多项式是按如下描述来构造的：每一个例子中，对相同的 (m, n) 我们构造了 50 个随机情形的多项式对，然后报告算得结果的平均值. 对每个例子, 互素部分和最大公因子都是随机构造的多项式, 其系数是区间 $-10 \le c \le 10$ 中的随机整数. 然后对这样构造的多项式再加以扰动. 至于扰动的构造, 我们取一个与乘积多项式次数相同，系数在区间 $[-10^e, 10^e]$ 的随机多项式. 最后，我们规范化这个扰动多项式以使得其相对扰动为 10^{-e}，这里 e 是一个给定的正整数.

在下表中, m, n 代表给定多项式 a 和 b 的全次数；k 是给定的一个正整数；*error (classic)* 和*error (new fast)*分别代表应用文 [53] 中的算法 AppSylv-k 和本文的算法所计算得的多项式极小扰动 $\|\tilde{a}-a\|_2^2+\|\tilde{b}-b\|_2^2$；*cond.num.*代表所解最小二乘系统 (4.3.1) 的系数阵的 2- 范数条件数；而*for.err.* 表示近似最小二乘解（相对于应用 [53] 中的算法计算得来的解）的相对误差. 在表格的最后一列，我们报告了最小二乘问题的向后扰动大小 $\hat{\varepsilon}(\hat{\mathbf{y}})$.

具体地说，$\hat{\varepsilon}(\hat{\mathbf{y}})$ 是向后扰动 δM 的一个近似的 Frobenius 范数上界，这里 $\hat{\mathbf{y}}$ 是下面这个最小二乘系统的准确的最小二乘解：

$$\min_{\mathbf{y}} \|(M+\delta M)\mathbf{y}-\mathbf{z}\|_2. \tag{4.4.1}$$

$\hat{\varepsilon}(\hat{\mathbf{y}})$ 与文 [88, 46] 中所提出的可能的最小的向后扰动 $\varepsilon(\hat{\mathbf{y}})$ 相差至多一个不超过 2 的因子，$\hat{\varepsilon}(\hat{\mathbf{y}})$ 可以按如下方式计算 [32]: 如前面所定义，$M = U \begin{bmatrix} \Sigma \\ \mathbf{0} \end{bmatrix} V^T$ 是 M 的奇异值分解. 假定 $\hat{\mathbf{y}} \neq \mathbf{0}$，并且令

$$\hat{\mathbf{r}} = \mathbf{z} - M\hat{\mathbf{y}} = U \begin{bmatrix} \hat{\mathbf{r}}_1 \\ \hat{\mathbf{r}}_2 \end{bmatrix},\ \hat{\mathbf{r}}_1 \in \mathbb{R}^{(2m+2n-2k+3)\times 1},\ \hat{\mathbf{r}}_2 \in \mathbb{R}^{k\times 1},\ \eta = \frac{\|\hat{\mathbf{r}}\|_2}{\|\hat{\mathbf{y}}\|_2}.$$

那么

$$\hat{\varepsilon}(\hat{\mathbf{y}}) = \min(\eta, \tilde{\sigma}),\ \text{这里}\ \tilde{\sigma} = \sqrt{\frac{\hat{\mathbf{r}}_1^T \Sigma^2 (\Sigma^2+\eta^2 I)^{-1}\hat{\mathbf{r}}_1}{\|\hat{\mathbf{r}}_2\|_2^2/\eta^2 + \eta^2 \hat{\mathbf{r}}_1^T(\Sigma^2+\eta^2 I)^{-2}\hat{\mathbf{r}}_1}}.$$

如上表所述，小的向后扰动 $\hat{\varepsilon}(\hat{\mathbf{y}})$ 表明计算得来的近似最小二乘解是向后稳定的. 所计算的多项式的极小扰动与按文 [53] 中算法算得的结果至少有相同的数量级；特别地，在前八个例子中，我们所计算的结果与按文 [53] 中算法所得来的结果有多位相同的有效数字. 当然，正如后两个例子所表明的，计算结果的精度受到了 M 的大的条件数所影响.

第五章　结论与展望

本文基于位移结构矩阵的观点，围绕一元近似最大公因子的计算问题做了如下两方面研究：

1. 提出了一个求解 Sylvester 矩阵数值秩的快速算法，并讨论了其在计算 Sylvester 矩阵奇异值方面的应用. 基于 Sylvester 矩阵 S 的位移结构，我们用广义 Schur 算法实现 $T = S^TS$ 或 H^TH 的快速 Cholesky 分解来计算 S^TS 的数值秩，进而得到 S 的数值秩. 其中 H 是我们引入的一类 Hankel 型 Sylvester 矩阵. 我们找到一个技巧来构造 S 和 H 以保证广义 Schur 算法顺利进行. 通过对 Cholesky 分解做向后误差分析，我们可以设定一个向后误差大小的临界值来判定 Cholesky 分解何时停止而得到数值秩.

整个算法需要 Sylvester 矩阵阶数平方次的运算量；相比于奇异值分解算法其复杂度低了一阶. 我们用 Maple 软件实现了算法；数值结果表明，对具有低的亏秩的 Sylvester 矩阵我们的算法是比较有效的.

2. 用快速算法求解了 Sylvester 矩阵的结构低秩逼近问题. Sylvester 矩阵的结构低秩逼近问题来源于一元多项式近似最大公因子的求解问题，文 [53] 中所给算法的核心步骤是求解一个关于位移结构矩阵的最小二乘问题。忽略位移结构，用经典的算法求解需要复杂度为 $O(st^2)$, 这里 s 和 t 分别代表最小二乘问题系数阵的行数和列数. 基于最小二乘问题系数阵的位移结构，应用低秩的生成子矩阵，我们用平方阶的复杂度 $O(s^2)$ 求解了这一问题.

对于发展符号数值混合计算领域中的位移结构矩阵算法，有待研究的问题还有很多. 例如，文献 [53] 中讨论的多项式的系数是复数，而本文第四章中给出的 Sylvester 矩阵的结构低秩逼近问题的快速算法，考虑的只是实数的情形，那么对于复数情形的实现是我们下一步的任务. 此外，如何发挥位移结构矩阵在求解关于一元多个数值多项式，及多元数值多项式的问题中的作用，还是一个崭新的研究领域. 甚至对已经有所进展的问题，还有必要探讨是否有更合适的多项式代数问题到位移结构矩阵算法的转换方法，进而更高效，更稳定的解决数值多项式代数的问题.

附录一 §3.2 中几个定理的证明

定理 A.0.1 *设 $S(a,b)$ 是一元 m 次多项式 $a(x)$ 和一元 n 次多项式 $b(x)$ 的形如 (1.2.1) 的 Sylvester 矩阵. 如果 $S(a,b)$ 的秩为 $m+n-d$, 这里 $0<d\le\min\{m,n\}$, 那么 $S(a,b)$ 的前 $m+n-d$ 个列向量必是列满秩.*

证明：当 $d=\min\{m,n\}$ 时定理的结论显然成立. 当 $d<\min\{m,n\}$ 时，$S(a,b)$ 的前 $m+n-d$ 个列向量所构成的子矩阵具有如下形状：

$$\begin{bmatrix}
a_m & \cdots\cdots & a_0 & & & \\
\ddots & & \ddots & & & \\
 & a_m & \cdots\cdots & a_0 & & \\
 & \ddots & & \ddots & & \\
 & & \ddots & \cdots\cdots\cdots & a_0 \\
 & & & \ddots & \vdots \\
 & & & a_m & \cdots & a_d \\
b_n & \cdots\cdots & b_0 & & & \\
\ddots & & \ddots & & & \\
 & b_n & \cdots\cdots & b_0 & & \\
 & \ddots & & \ddots & & \\
 & & \ddots & \cdots\cdots\cdots & b_0 \\
 & & & \ddots & \vdots \\
 & & & b_n & \cdots & b_d
\end{bmatrix}
\begin{matrix}
\left.\vphantom{\begin{matrix}a\\a\end{matrix}}\right\}d \\
\left.\vphantom{\begin{matrix}a\\a\\a\\a\\a\end{matrix}}\right\}n-d \\
\left.\vphantom{\begin{matrix}a\\a\end{matrix}}\right\}d \\
\left.\vphantom{\begin{matrix}a\\a\\a\\a\\a\end{matrix}}\right\}m-d
\end{matrix}
\quad . \tag{A.0.1}$$

从这一矩阵中去掉前 d 列，再分别去掉 $a(x)$ 和 $b(x)$ 系数行的前 d 行，得到的子矩阵是一个子结式矩阵 [77]. 进一步地，由 $S(a,b)$ 的秩为 $m+n-d$ 可知，$\gcd(a,b)$ 的次数为 d, 从而这一子结式矩阵必是满秩的 [40, 66]; 结合 (A.0.1) 中矩阵的形状，并根据 $a_m\neq 0$ (或者 $b_n\neq 0$), 可以推断 (A.0.1) 中的矩阵必是列满秩的.

我们进一步利用子结式与最大公因子的关系（详细结论可见文献 [40, 66] 的相关章节)，给出了定理 3.2.3 和定理 3.2.7 的另一个证明.

定理 3.2.3 的证明：Sylvester 矩阵 $S(a,b)$ 的秩为 $m+n-d$ 蕴涵着 $\gcd(a,b)$ 的次数为

d. 考虑第 $d+1$ 个 Sylvester 子矩阵 $S_{d+1} \in \mathbb{R}^{(m+n-d)\times(m+n-2d)}$(其构造可见 (4.2.1))：

$$S_{d+1} = \underbrace{\begin{bmatrix} a_m & & \\ a_{m-1} & \ddots & \\ \vdots & \ddots & a_m \\ \vdots & & a_{m-1} \\ a_0 & \ddots & \vdots \\ & \ddots & \vdots \\ & & a_0 \end{bmatrix}}_{n-d} \underbrace{\begin{bmatrix} b_n & & \\ b_{n-1} & \ddots & \\ \vdots & \ddots & b_n \\ \vdots & & b_{n-1} \\ b_0 & \ddots & \vdots \\ & \ddots & \vdots \\ & & b_0 \end{bmatrix}}_{m-d}, \tag{A.0.2}$$

由 $\deg(\gcd(a,b)) = d$ 的假设知，S_{d+1} 是列满秩的 [40, 66]. 在 S_{d+1} 的前 $n-d$ 个列向量后面追加上 d 列 $a(x)$ 的系数向量，使其成为 $S(a,b)$ 的前 $m+n-d$ 个列向量所构成的子矩阵，结合定理中的假设

$$a(x) = x^p(a_m x^{m-p} + \cdots + a_p),\ a_p \neq 0, \tag{A.0.3}$$

$$b(x) = x^q(b_n x^{n-q} + \cdots + b_q),\ b_q \neq 0, \tag{A.0.4}$$

这一子矩阵必有如下形状：

$$\underbrace{\begin{bmatrix} a_m & & \\ \vdots & \ddots & \\ \vdots & & \ddots \\ \vdots & & \vdots \\ a_p & & \\ & \ddots & \vdots \\ & & \ddots \\ & & \end{bmatrix}}_{n-d} \underbrace{\begin{bmatrix} \\ \\ \\ a_m \\ \vdots \\ \vdots \\ \vdots \\ a_p \end{bmatrix}}_{d} \underbrace{\begin{bmatrix} b_n & \ddots & \\ \vdots & & b_n \\ b_q & & \vdots \\ & \ddots & \vdots \\ & & b_q \\ & & \end{bmatrix}}_{m-d}, \tag{A.0.5}$$

这样由 S_{d+1} 的列满秩及 $0 \le p \le q$ 的假设不难推出这样的子矩阵必定是列满秩的. □

定理 3.2.7 的证明: Hankel 型 Sylvester 矩阵 H 的秩为 $m+n-d$ 意味着 $\gcd(a,b)$ 的

次数为 d. 考虑 Hankel 型 Sylvester 矩阵 $H(a,b)$ 的前 $m+n-d$ 列所构成的子矩阵：

$$\begin{bmatrix} & & a_m & & \\ & & \iddots\ \vdots & & \\ & a_m & a_0 & & b_n \\ & \iddots\ \vdots & \iddots & & \iddots\ \vdots \\ a_m & a_0 & & b_n & b_0 \\ \vdots\ \iddots & & & \vdots\ \iddots & \\ a_0 & & & b_0 & \end{bmatrix}, \tag{A.0.6}$$
$$\underbrace{\hphantom{a_m\ a_0}}_{n-d}\ \underbrace{\hphantom{a_0}}_{d}\ \underbrace{\hphantom{b_n\ b_0}}_{m-d}$$

消去其上面 d 行，并消去 $a(x)$ 系数列的后 d 列，把得到的子矩阵记为 H_{d+1}：

$$H_{d+1} = \begin{bmatrix} & a_m & & b_n \\ & \iddots a_{m-1} & & \iddots b_{n-1} \\ a_m & \vdots & b_n & \vdots \\ a_{m-1} & \vdots & b_{n-1} & \vdots \\ \vdots & \iddots a_0 & \vdots & \iddots b_0 \\ \vdots & \iddots & \vdots & \iddots \\ a_0 & & b_0 & \end{bmatrix}, \tag{A.0.7}$$
$$\underbrace{\hphantom{a_{m-1}\ a_{m-1}}}_{n-d}\ \underbrace{\hphantom{b_{n-1}\ b_{n-1}}}_{m-d}$$

那么 $H_{d+1} \in \mathbb{R}^{(m+n-d)\times(m+n-2d)}$, 并且，$H_{d+1}$ 与第 $d+1$ 个 Sylvester 子矩阵 S_{d+1} 有如下关系：

$$H_{d+1} = S_{d+1} \begin{bmatrix} P_{n-d} & \\ & P_{m-d} \end{bmatrix}. \tag{A.0.8}$$

这样，由 $\deg(\gcd(a,b)) = d$ 可知 H_{d+1} 是列满秩的；再注意到 $a_m \neq 0$, 就可以推出 (A.0.6) 式中的矩阵必是列满秩的. $\square$

参考文献

[1] A. C. Antoulas, J. A. Ball, J. Kang, and J. C. Willems, *On the solution of the minimal rational interpolation problem*, Lin. Alg. Appl., 137/138 (1990), pp. 511–573.

[2] A. C. Antoulas and J. C. Willems, *Minimal rational interpolation and Prony's method*, in Analysis and Optimization of Systems. A. Bensoussan and J. L. Lions, eds., Lecture Notes in Control and Information Sciences, Vol. 144, Springer-Verlag, Berlin, 1990, pp. 297–306.

[3] A. A. Anda and H. Park, *Fast plane with dynamic scaling*, SIAM J. Matrix Anal. Applic., 15 (1994), pp. 162–174.

[4] J. Ambrosiano, L. Greengard, and V. Rokhlin, *The fast multipole method for gridless particle simulation*, Computer Physics Communications, 48 (1988), pp. 115–125.

[5] Ä. Björck, *Stability analysis of the method of semi-normal equations for least squares problems*, Lin. Alg. Appl., 88/89 (1987), pp. 31–48.

[6] A. W. Bojanczyk, R. P. Brent, and F. R. de Hoog, *QR factorization of Toeplitz matrices*, Numer. Math., 49 (1986), pp. 81–94.

[7] A. W. Bojanczyk, R. P. Brent, and F. R. de Hoog, *A note on downdating the Cholesky factorization*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 210–221.

[8] J. A. Ball, I. Gohberg, and L. Rodman, *Two-sided Lagrange-Sylvester interpolation problems for rational matrix functions*, in Proceedings of Symposia in Pure Mathematics, Part I, Vol. 51. American Mathematical Society, Providence, RI, 1990, pp. 17–83.

[9] S. Barnett, *Greatest common divisor of several polynomials*, in Proc. Camb. Phil. Soc, vol. 70, 1971, pp. 263–268.

[10] S. Barnett, *Polynomials and linear control systems*, Marcel Dekker, New York, 1983.

[11] A. W. Bojanczyk, R. P. Brent, F. R. de Hoog, and D. R. Sweet, *On the stability of the Bareiss and related Toeplitz factorization algorithm*, SIAM J. Matrix Anal. Applc., 16 (1995), pp. 40–57.

[12] D. Bini and L. Gemignani, *On the Euclidean scheme for polynomials having inter-*

laced real zeros, in Proc. 2nd Ann. ACM Symp. on Parallel Algorithms and Architectures, pp. 254–258. ACM Press, New York, 1990.

[13] D. Bini and L. Gemignani, *Fast parallel computation of the polynomial remaider sequence via Bézout and Hankel matrices*, SIAM J. Comput., 24 (1995), pp. 63–77.

[14] D. Bini and V. Y. Pan, *Polynomial and matrix computations, vol. 1: Fundamental algorithms*. Birkhäuser, Boston, Massachusetts, 1994.

[15] T. Boros, A. H. Sayed, and T. Kailath, *Structured matrices and unconstrained rational interpolation*, Lin. Alg. Appl., 203/204 (1994), pp. 155–188.

[16] R. Bitmead and B. Anderson, *Asymptotically fast solution of Toeplitz and related systems of linear equations*, Lin. Alg. Appl., 34 (1980), pp. 103–116.

[17] R. P. Brent, F. G. Gustavson, and D. Y. Y. Yun, *Fast solution of Toeplitz systems of equations and computation of Padé approximations*, J. of Algorithms, 1 (1980), pp. 259–295.

[18] B. Beckermann and G. Labahn, *When are two numerical polynomials relatively prime?*, J. Symbolic Comput., 26 (1998), pp. 677–689. Special issue on Symbolic Numeric Algebra for Polynomials.

[19] S. Chandrasekaran and A. H. Sayed, *Stabilizing the generalized Schur algorithm*, SIAM J. Matrix Anal. Applc., 17 (1996), pp. 950–983.

[20] S. Chandrasekaran and A. H. Sayed, *A fast stable solver for nonsymmetric Toeplitz and quasi-Toeplitz systems of linear equations*, SIMAX, 19 (1998), pp. 107–139.

[21] J. Chun, T. Kailath, and H. Lev-Ari, *Fast parallel algorithms for QR and triangular factorization*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 899–913.

[22] P. Chin, R. M. Corless, and G. F. Corliss, *Optimization strategies for the approximate GCD problem*, in Proc. 1998 Internat. Symp. Symbolic Algebraic Comput. (ISSAC1998), pp. 228–235. ACM Press, New York, 1998.

[23] R. M. Corless, P. M. Gianni, B. M. Trager, and S. M. Watt, *The singular value decomposition for polynomial systems*, in Proc. 1995 Internat. Symp. Symbolic Algebraic Comput. (ISSAC1995), pp. 195–207.

[24] R. M. Corless, S. M. Watt, and L. Zhi, *QR factoring to compute the GCD of univariate approximate polynomials*, IEEE Transactions on Signal Processing, 52 (2004), pp. 3394–3402.

[25] G. Cybenko, *A general orthogonalization technique with applications to time series analysis and signal processing*, Math.Comp., 40 (1983), pp. 323–336.

[26] G. Cybenko, *Fast Toeplitz orthogonalization using inner products*, SIAM J. Sci. Stat. Comput., 8 (1987), pp. 734–740.

[27] J. Chun and T. Kailath, *Divide-and-conquer solutions of least squares problems for matrices with displacement structure*, SIAM J. Matrix Anal. Applic., 12 (1991), pp. 128–145.

[28] I. Z. Emiris, A. Galligo, and H. Lombardi, *Numerical univariate polynomial GCD*, AMS-SIAM Summer Seminar in Applied Math.: Mathematics of Numerical Analysis. Park City, Utah, 1995, (J. Renegar, M. Shub, S. Smale editors), Lectures in Applied Math., 32, pp. 323–344, Amer. Math. Soc., Providence, RI., 1996.

[29] I. Z. Emiris, A. Galligo, and H. Lombardi, *Certified approximate univariate GCDs*, J. Pure and Applied Algebra, 117/118 (1997), pp. 229–251. Special issue on Algorithms for Algebra.

[30] G. H. Golub and C. F. V. Loan, *Matrix computations*. Johns Hopkins University Press, 3nd edition, 1996.

[31] M. Gu, *Stable and efficient algorithms for structured systems of linear equations*, LBL Report LBL-37690, Lawrence Berkeley National Laboratory, September 1995.

[32] M. Gu, *Backward perturbation bounds for linear least squares problems*, SIAM J. Matrix Anal. Applic., 20 (1998), pp. 363–372.

[33] M. Gu, *New fast algorithms for structured linear least squares problems*, SIAM J. Matrix Anal. Applic., 20 (1998), pp. 244–269.

[34] I. Gohberg, T. Kailath, and V. Olshevsky, *Fast Gaussian elimination with partial pivoting for matrices with displacement structure*, Math. Comp., 64 (1995), pp. 1557–1576.

[35] A. Gerasoulis, *A fast algorithm for the multiplication of generalized Hilbert matrices with vectors*, Math. of Comp., 50 (1987), pp. 179–188.

[36] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulation*, J. Comput. Physics, 73 (1987), pp. 325–348.

[37] S. Gao, E. Kaltofen, J. May, Z. Yang, and L. Zhi, *Approximate factorization of multivariate polynomials via differential equations*, in Proc. 2004 Internat. Symp.

Symbolic Algebraic Comput. (ISSAC2004), pp. 167–174.

[38] I. Gohberg and V. Olshevsky, *Fast algorithm for matrix Nehari problem*, in Proceedings of MTNS-93, Systems and Networks: Mathematical Theory and Applications, v.2, Invited and Contributed Papers. edited by U. Helmke, R. Mennicken and J. Sauers, Academy Verlag, 1994, pp. 687–690.

[39] I. Gohberg and V. Olshevsky, *Fast state space algorithms for matrix Nehari and Nehari-Takagi interpolation problems*, Integral Equations and Operator Theory, 20 (1994), pp. 44–83.

[40] J. von zur Gathen and J. Gerhard, *Modern computer algebra.* Cambrige University Press, 1999.

[41] V. Hribernig and H. J. Stetter, *Detection and validation of clusters of polynomials zeros*, J. Symbolic Comput., 24 (1997), pp. 667–681.

[42] G. Heinig, *Inversion of generalized Cauchy matrices and other classes of structured matrices*, in Linear algebra in signal processing, IMA volumes in mathematics and its applications, vol. 69, 1994, pp. 95–114.

[43] G. Heinig and K. Rost, *Algebraic methods for Toeplitz-like matrices and operators*, Akademie-Verlag, Berlin and Birkhäuser, 1984.

[44] G. Heinig and K. Rost, *Algebraic methods for Toeplitz-like matrices and operators*, Operator Theory, 13 (1984), pp. 109–127.

[45] N. J. Higham, *Analysis of the Cholesky decomposition of a semi-definite matrix.* In Reliable Numerical Computation, (M. G. Cox and S. J. Hammarling, eds.), 161–185, Oxford University Press, 1990.

[46] N. J. Higham, *Accuracy and stability of numerical algorithms.* Society for Industrial and Applied Mathematics, Philadelphia, 1996.

[47] R. A. Horn and C. R. Johnson, *Matrix analysis*, 1985. Cambrige University Press.

[48] C. P. Jeannerod and L. Zhi, *Computing low rank generators of Sylvester matrix embeddings.* Manuscript, 2002.

[49] T. Kailath, S. Y. Kung, and M. Morf, *Displacement ranks of a matrix*, Bull. Amer. Math. Soc., 1 (1979), pp. 769–773.

[50] T. Kailath, S. Y. Kung, and M. Morf, *Displacement ranks of matrices and linear equations*, J. Math. Anal. Appl., 68 (1979), pp. 395–407.

[51] T. Kailath and V. Olshevsky, *Symmetric and Bunch-Kaufman pivoting for Cauchy-like matrices with applications to Toeplitz-like matrices. 1995.*

[52] T. Kailath and A. H. Sayed, *Displacement structure: theory and applications*, SIAM Review, 37 (1995), pp. 297–386.

[53] E. Kaltofen, Z. Yang, and L. Zhi, *Structured low rank approximation of a Sylvester matrix.* In Dongming Wang and Lihong Zhi, editors, International Workshop on Symbolic-Numeric Computation Proceedings, pp. 188-201, 2005.

[54] N. K. Karmarkar and Y. N. Lakshman, *Approximate polynomials greatest common divisions and nearest singular polynomials*, in Proc. 1996 Internat. Symp. Symbolic Algebraic Comput. (ISSAC1996), pp. 35–42.

[55] N. K. Karmarkar and Y. N. Lakshman, *On approximate GCDs of univariate polynomials*, J. Symbolic Comput., 26 (1998), pp. 653–666. Special issue on Symbolic Numeric Algebra for Polynomials.

[56] G. V. Laureano, *An elementary proof of Barnett's theorem about the greatest common divisor of several univariate polynomials*, Lin. Alg. Appl., 247 (1996), pp. 185–202.

[57] B. Li, Z. Liu, and L. Zhi, *A structured rank-revealing method for Sylvester matrix.* Submitted to J. CAM (Journal of Computational and Applied Mathematics), 2005.

[58] B. Li, Z. Yang, and L. Zhi, *Fast low rank approximation of a Sylvester matrix by structured total least norm*, J. JSSAC (Journal of Japan Society for Symbolic and Algebraic Computation), 11 (2005), pp. 165–174.

[59] B. Li, Z. Liu, and L. Zhi, *Fast implementation of low rank approximation of a Sylvester matrix.* Accepted for publication in SNC book "Symbolic-Numeric Computation" (a book for 2005 International Workshop on Symbolic-Numeric Computation), edited by Dongming Wang and Lihong Zhi, to be published by Birkhauser, Basel Boston, 2006.

[60] R. Li, *On eigenvalues of a Rayleigh quotient matrix*, Lin. Alg. Appl., 169 (1992), pp. 249–255.

[61] P. Lancaster and M. Tismenetsky, *The theory of matrices.* Academic Press, New York, 1985.

[62] M. A. Laidacker, *Another theorem relating Sylvester's matrix and the greatest common divisor*, Math. Magazine, 42 (1969), pp. 126–128.

[63] T. Y. Li and Z. Zeng, *A rank-revealing method and its application.* Manuscript available at http://www.neiu.edu/z̃zeng/papers. htm, 2003.

[64] X. Liu and Y. Xu, *On the Rayleigh quotient matrix (in chinese)*, Math. Numer. Sinica, 12 (1990), pp. 208–213.

[65] M. Morf, *Doubling algorithms for Toeplitz and related equations*, in Proc. IEEE Internat. Conf. on Acoustics, Speech and Signal Processing. Denver, IEEE Press, Piscataway, NJ, 1980, pp.954–959.

[66] B. Mishra, *Algorithmic algebra.* Science Press. Springer Verlag, 2001.

[67] J. G. Nagy, *Fast inverse QR factorization for Toeplitz matrices*, SIAM J. Sci. Stat. Comput., 14 (1993), pp. 1174–1183.

[68] M. T. Noda and T. Sasaki, *Approximate GCD and its application to ill-conditioned algebraic equations*, J. Comput. Appl. Math., 38 (1991), pp. 335–351.

[69] V. Olshevsky, *Pivoting for structured matrices with applications, 1997.* http://www.cs.gsu.edu/m̃atvro.

[70] V. Olshevsky and V. Y. Pan, *A unified superfast algorithm for boundary rational tangential interpolation problem and for inversion and factorization of dense structured matrices*, in Proc. 39th Annual IEEE Symp. on Foundation of Computer Science, pp. 192–201. IEEE Computer Society Press, 1998.

[71] V. Olshevsky and M. A. Shokrollahi, *A displacement approach to efficient decoding of Algebraic-Geometric codes*, in Proceedings of 31th Annual Symposium on Theory of Computing (STOC'99), pp. 235–244. ACM Press, New York, May 1999.

[72] V. Olshevsky and M. A. Shokrollahi, *A displacement approach to efficient decoding of Algebraic-Geometric codes*, in Proceedings of 14th International Symposium on Networks and Systems (MTNS'2000). University of Perpignan, Perpignan, France, 2000.

[73] V. Y. Pan, *Approximate polynomial GCDs, Padé approximation, polynomial zeros and bipartite graphs*, in Proc. 9nd Ann. ACM-SIAM Symp. on Discrete Algorithms, pp. 68–77. ACM Press, New York, and SIAM Publications, Philadelphia, 1998.

[74] V. Y. Pan, *Computations with dense structured matrices*, in Proc. 1989 Internat. Symp. Symbolic Algebraic Comput. (ISSAC1989), pp.34–42, ACM Press, New York, and Math. of Comp. 55, 191, pp.179–190, 1990.

[75] V. Y. Pan, *Some recent algebraic/numerical algorithms*, Electronic Proceedings of IMACS/ACA'98.

[76] V. Y. Pan, *Numerical computation of a polynomial GCD and extensions*, Information and Computation, 167 (2001), pp. 71–85.

[77] V. Y. Pan, *Structured matrices and polynomials: unified superfast algorithms.* Birkhäuser, Boston, Springer New York, 2001.

[78] H. Park and L. Eldén, *Stability analysis and fast algorithms for triangularization of Toeplitz matrices*, Numer. Math., 76 (1997), pp. 383–402.

[79] H. Park, L. Zhang, and J. B. Rosen, *Low rank approximation of a Hankel matrix by structured total least norm*, BIT, 39 (1999), pp. 757–779.

[80] S. Qiao, *Hybrid algorithm for fast Toeplitz orthogonalization*, Numer. Math., 53 (1988), pp. 351–366.

[81] J. B. Rosen, H. Park, and J. Glick, *Total least norm formulation and solution for structured problems*, SIAM J. Matrix Anal. Applic., 17 (1996), pp. 110–128.

[82] V. Rokhlin, *Rapid solution of integral equations of classical potential theory*, J. Comput. Physics, 60 (1985), pp. 187–207.

[83] A. H. Sayed, T. Kailath, H. Lev-Ari, and T. Constantinescu, *Recursive solutions of rational interpolation problems via fast matrix factorization*, Integral Equations and Operator Theory, 20 (1994), pp. 84–118.

[84] D. R. Sweet, *Fast Toeplitz orthogonalization*, Numer. Math., 43 (1984), pp. 1–21.

[85] M. Stewart, *Cholesky factorization of semi-definite Toeplitz matrices*, Lin. Alg. Appl., 254 (1997), pp. 497–526.

[86] A. Schönhage, *Quasi-GCD computations*, J. Complexity, 1 (1985), pp. 118–137.

[87] J. Sun, *Matrix perturbation analysis (in Chinese).* Science Press, 1987.

[88] B. Waldén, R. Karlson, and J.-G. Sun, *Optimal backward perturbation bounds for the linear least square problem*, Numer. Lin. Alg. Appl., 2 (1995), pp. 271–286.

[89] A. F. Ware, *Fast approximate Fourier transform for irregularly spaced data*, SIAM Review, 40 (1998), pp. 838–856.

[90] 王东明, 夏壁灿, *计算机代数*. 清华大学出版社，北京，2004.

[91] 杨路，李志斌，侯晓荣，陈发来，夏壁灿，支丽红, *符号计算选讲（王东明主编）*. 清华大学出版社，北京，2003.

[92] Z. Xu, K. Zhang, and Q. Lu, *The fast algorithm for solving Toeplitz matrices (in Chinese)*. Xibei Industry University Press, 1999.

[93] Z. Zeng, *The approximate GCD of inexact polynomials. part I: a univariate algorithm.* Manuscript, 2004.

[94] L. Zhi, *Displacement structure in computing approximate GCD of univariate polynomials*, in Proc. Sixth Asian Symposium on Computer Mathematics (ASCM 2003), 2003, pp. 288–298.

在学期间发表文章目录

[1] B. Li, Z. Yang and L. Zhi. Fast low rank approximation of a Sylvester matrix by structured total least norm. J.JSSAC (Journal of Japan Society for Symbolic and Algebraic Computation), 11 (2005), pp. 165–174.

[2] B. Li, Z. Liu and L. Zhi. Fast implementation of low rank approximation of a Sylvester Matrix. Accepted for publication in SNC book “Symbolic-Numeric Computation” (a book for 2005 International Workshop on Symbolic-Numeric Computation), edited by Dongming Wang and Lihong Zhi, to be published by Birkhauser, Basel Boston, 2006.

[3] B. Li, Z. Liu and L. Zhi. A structured rank-revealing method for Sylvester matrix. MM Research Preprints, 24 (2004), pp.151–165. Submitted to J.CAM (Journal of Computational and Applied Mathematics)

致　　谢

值此论文完成之际，作者谨在此向多年来给予我关心和帮助的老师，同学，朋友和家人表示衷心地感谢!

首先特别感谢我的导师刘卓军研究员和支丽红副研究员. 两位导师三年来对我学习和科学研究上的莫大指导，把我引入数学科学研究的道路上来, 是两位导师不断地鼓励和教诲，使我对科学研究信心更充足，兴趣更浓厚. 本学位论文是在两位导师的悉心指导下完成的. 定稿前，两位导师于百忙中仔细地审阅了全文，并提出了许多宝贵的意见. 在此论文完成之际，谨向两位导师表示衷心的感谢.

感谢吴文俊院士. 虽然没有直接受到吴先生的指导，但吴先生八十多高龄依然精力充沛，思维敏捷，治学严谨的数学大师的风范深深感染着我.

感谢数学机械化研究中心的石赫研究员，高小山研究员，李洪波研究员，李子明研究员，王定康副研究员，杜宏副研究员，马玉杰助理研究员，闫振亚助理研究员，冯如勇助理研究员. 他们在历次学术活动中传授的人生感悟、治学体会, 令作者记忆深刻. 特别感谢李子明研究员对本文定理 A.0.1 的证明所提供的帮助. 同时感谢周代珍老师和王莎莎小姐平时对我的诸多帮助和照顾.

感谢计算数学所白中治研究员和曹礼群研究员，与两位老师的讨论让我受益匪浅.

在数学机械化中心的学习和生活中，我得到了许多同学的帮助，他们是杨争峰，徐荣华，张贵林，柴凤娟，申立勇，程进三，王培宏，郑大彬，张明波，张宁，袁春明，孟晓辉，吴晓丽，李宾，孙冬霞，龙红亮，秦龙，黄雷，熊涛，刘枫，孙维昆，唐春明，罗勇，谭作文，张艳硕，张帅，周成雄，陈颖，王灯山和谢正等人，与你们共度的愉快时光给我留下了许多美好回忆.

感谢我的硕士导师盛中平教授，硕士期间盛老师的指导让我接触到计算机代数这个活跃的研究领域，这对我后来选择计算机代数作为自己的研究方向起着重要的作用.

感谢我的每一位亲人，是你们无私的爱激励我不断努力. 特别感谢我的父亲和妹妹，你们的关怀和爱护是我坚强的后盾. 特别感谢我的爱人扶先辉，这三年紧张的学习生活中是你对我的鼓励和支持，让我有更坚强的信心完成学业.