

分类号_____

密级_____

UDC _____

编号_____

中国科学院研究生院 博士学位论文

多元多项式的近似因式分解和近似最大公因子的问题

杨争峰

指导教师 _____ 吴文俊院士，支丽红副研究员

_____ 中国科学院数学与系统科学研究院

申请学位级别 _____ 博士 _____ 学科专业名称 _____ 应用数学

论文提交日期 _____ 2006年4月 _____ 论文答辩日期 _____ 2006年5月

培养单位 _____ 中国科学院数学与系统科学研究院

学位授予单位 _____ 中国科学院研究生院

答辩委员会主席 _____

摘 要

本文主要讨论的是多元多项式的近似因式分解和有关单变元和多变元多项式的近似最大公因子的一些问题。我们所得到的主要结果包括三个方面：

第一，计算多变元多项式的近似因式分解。算法输入的多项式 f 在复数域上不可约，而且系数是不精确并带有未知的扰动。希望给多项式加一个小的扰动使得到的多项式在 $\mathbb{C}$ 上可约。我们将 W. Ruppert 和 S. Gao 算法推广到近似多元多项式的情形，用奇异值分解和 Gauss-Newton 优化来计算近似多项式因子。当多项式的系数相对误差在 10^{-3} 范围内，本文给出了多元多项式的近似因式分解算法。

第二：对于系数不精确的单变元和多变元多项式，计算它的近似最大公因子 (GCD) 问题可以转化为：给一个 (广义) Sylvester 矩阵，计算一个离它最近的低秩的 (广义) Sylvester 矩阵。这里，单变元多项式对应的是 Sylvester 矩阵，多变元多项式对应的是广义 Sylvester 矩阵。我们设计了基于罚函数的结构最小二乘范数 (STLN) 的迭代算法来计算近似最大公因子。另外，当输入中给了容许误差 ϵ 时，STLN 算法可以计算近似 ϵ -GCD。

第三，本文把计算两个多项式的近似最大公因子问题推广到计算多个多项式 (单变元和多变元) 近似最大公因子问题，而且还解决了计算多项式系数有线性限制的近似最大公因子问题。这里用的算法同样是 STLN 方法。对于单变元的情形，系数有线性限制的近似最大公因子问题可以应用到计算最接近的有 k 重根的奇异多项式问题，并且得到全局最优解。

关键词：近似因式分解，奇异值分解，Gauss-Newton 优化，(广义) Sylvester 矩阵，近似最大公因子，结构最小二乘范数，奇异多项式，线性限制，符号/数值算法。

Approximate Factorization of Multivariate Polynomials and Some Problems in Approximate Greatest Common Divisors

Zhengfeng Yang (majored in applied mathematics)

Directed by Professor Wu Wen Tsün and Associate Professor Lihong Zhi

Abstract

In this paper, we discuss the problem on approximate factorization of multivariate polynomials and approximate greatest common divisors(GCD) of univariate and multivariate polynomials. The main results in the thesis include three parts:

(1) About approximate factorization in many variables, the input to our algorithm is a multivariate polynomial, whose complex rational coefficients are considered imprecise with an unknown error that causes f to be irreducible over the complex number $\mathbb{C}$. We seek to perturb the coefficients by a small quantity such that the resulting polynomial factors over $\mathbb{C}$. One would like to minimize the perturbation in some selected distance measure. Our algorithms are based on a generalization of the differential forms introduced by W. Ruppert and S. Gao to many variables, and use singular value decomposition or structure total least squares approximation and Gauss-Newton optimization to numerically compute the approximate multivariate factors. We demonstrate on a large set of benchmark polynomials that our algorithms efficiently yield approximate factorizations within the coefficient noise that is substantial (10^{-3}).

(2) The task of determining the approximate greatest common divisor (GCD) of univariate and multivariate polynomials with inexact coefficients can be formulated as computing for a given (generalized) Sylvester matrix a new (generalized) Sylvester matrix of lower rank whose entries are near the corresponding entries of that input matrix. We solve the approximate GCD problem by a new method based on structured total least norm (STLN) algorithms, in our case for matrices with Sylvester structure. We present iterative algorithms that compute an approximate GCD and can certify an approximate ϵ -GCD when a tolerance ϵ is given on input. Each single iteration is carried out with a number of floating point operations that is of cubic order in the input degrees. We also demonstrate the practical performance of our algorithms on a diverse set of univariate or multivariate pairs of polynomials.

(3) We consider the problem of computing minimal real or complex deformations to the coefficients in a list of relatively prime real or complex multivariate polynomials such

that the deformed polynomials have a greatest common divisor (GCD) of a given degree at least k . In addition, we restrict the deformed coefficients by a given set of linear constraints, thus introducing the *linearly constrained approximate GCD* problem. We present an algorithm based on a version of the structured total least norm method and demonstrate on a diverse set of benchmark polynomials that the algorithm in practice computes globally minimal approximations. As an application of the linearly constrained approximate GCD problem we present an STLN-based method that computes to a real or complex polynomial the nearest real or complex polynomial that has a root of multiplicity at least k . We demonstrate that the algorithm in practice computes on the benchmark polynomials given in the literature the known globally optimal nearest singular polynomials. Our algorithms can handle, via randomized preconditioning, the difficult case when the nearest solution to a list of real input polynomials actually has non-real complex coefficients.

Keywords: approximate factorization, singular value decomposition, Gauss-Newton optimization, generalized Sylvester matrix, approximate greatest common divisor, structured total least norm(STLN), singular polynomials, linear constraints, hybrid/numerical algorithm .

目 录

摘 要	i
第一章 引言	1
§1.1 问题的背景	1
§1.2 多元多项式的近似因式分解的简介	2
§1.3 多项式的近似最大公因子的简介	3
§1.4 论文的主要内容	3
第二章 多元多项式的近似因式分解	5
§2.1 前言	5
§2.2 多元多项式近似除法和近似最大公因子	5
§2.2.1 多项式的近似除法	5
§2.2.2 近似最大公因子	6
§2.3 因式分解的算法和实验	11
§2.3.1 准确多元多项式绝对不可约因式分解	11
§2.3.2 近似多元多项式绝对不可约因式分解	14
§2.3.3 算法	16
§2.3.4 重因子情形	17
§2.3.5 近似因式分解的例子	18
§2.3.6 实验	20
第三章 结构低秩逼近 (广义) Sylvester 矩阵	23
§3.1 前言	23
§3.2 结构低秩逼近 Sylvester 矩阵	23
§3.2.1 预备知识	23
§3.2.2 STLN 解决有 Sylvester 结构的超定系统	28
§3.2.3 单变元的近似最大公因子算法和实验	31
§3.3 结构低秩逼近广义 Sylvester 矩阵	34
§3.3.1 预备知识	34
§3.3.2 用 STLN 解决有广义 Sylvester 结构的超定系统	37
§3.3.3 多变元近似最大公因子的算法和实验	40
§3.4 小结	43

第四章 多个有线性限制多项式的近似最大公因子和奇异多项式	45
§4.1 介绍	45
§4.2 转化为线性代数问题	46
§4.3 基于 STLN 的解	48
§4.3.1 输入系数有线性限制的解法	50
§4.3.2 实的输入计算复的最优	52
§4.4 算法和实验	53
§4.5 结论	58
第五章 结论与展望	59
致 谢	71

第一章 引言

§1.1 问题的背景

符号和数值混合计算的主要目标是结合符号和数值计算方法有效地解决一些数学问题。混合计算主要解决两类问题[23]：第一，对近似问题的求解。在日常生活中，测量的数据是有限精度的。因此，用符号计算中的算法是无法得到这类近似问题的解。第二，对于一些准确的问题，用符号计算中的算法效率很低，这时也需要用混合计算的方法去求解这些问题的近似解。

现在，多项式的符号数值算法（SNAP）在近似计算机代数中占有非常重要的地位。这是因为在揭示很多非线性问题所具有的难以得到的性质和特征过程中，SNAP 起到了重要的作用。混合计算中可以用的数值工具有奇异值分解，线性规划，最小二乘法，等等。近些年 SNAP 发展得很迅速。在 SNAP 方面，我们可以考虑几种类型问题。首先用近似最大公因子问题做为例子。

第一，全局最优问题。给定一对多项式，要找到一对离它们“最近”的有非平凡公因子的多项式。这里要求的是最近的，并且，对于不同的问题，“最近”所指的意思也不同。例如，给定一对多项式

$$\begin{aligned} f &= 1000x^{10} + x^3 - 1.0 \\ g &= 1.0x^2 - \frac{1}{100} \end{aligned}$$

想找到离它们最近的一对多项式 $\tilde{f}, \tilde{g}$ 。这里最近指的是 Euclidean 距离最近。用第四章算法计算得到多项式 $\tilde{f}, \tilde{g} \in \mathbb{C}[x]$:

$$\begin{aligned} \tilde{f} &= (x - 0.494155)(1000.0x^9 + 494.155x^8 + 244.189x^7 + 120.667x^6 + 59.6284x^5 \\ &\quad + 29.4659x^4 + 14.5612x^3 + 8.19650656855512948x^2 + 4.05239x + 2.00666) \\ \tilde{g} &= (x - 0.494155)(0.956139x + 0.383722) \end{aligned}$$

计算两对多项式之间的距离 $\|f - \tilde{f}\|_2^2 + \|g - \tilde{g}\|_2^2 = 0.042157867$ ，在第四章中将要说明这个距离是最近的距离。

第二， ϵ -近似问题。给定一对多项式，要找到一对有非平凡公因子的多项式，且要求这一对多项式离原来的一对多项式的距离不大于 ϵ 。还是用上面的例子，这里设 $\epsilon = 0.05$ 。很明显，上面找到一对多项式 $\tilde{f}, \tilde{g}$ 是这个问题的解。这里的问题要比全局最优问题条件要松一点，我们还可以找到另外一对多项式 $\hat{f}, \hat{g} \in \mathbb{C}[x]$ 也是这个

问题的解:

$$\begin{aligned}\hat{f} &= (x - 0.41019 - 0.29067i)(1000.x^9 + 410.191x^8 + 290.669ix^8 + 83.7678x^7 \\ &\quad + 238.459ix^7 + 122.163ix^6 - 34.952x^6 - 49.8461x^5 + 39.9505ix^5 + 1.89861ix^4 \\ &\quad - 32.0590x^4 - 13.7026x^3 - 8.5401ix^3 - 7.4870ix^2 - 2.13866ix^2 + 1.29957x \\ &\quad - 3.6947ix + 1.61022 - 1.1402i) \\ \hat{g} &= (x - 0.41019 - 0.29067i)(0.952118x - 0.001811ix + 0.315449 + 0.218002i)\end{aligned}$$

计算多项式之间的距离 $\|f - \hat{f}\|_2^2 + \|g - \hat{g}\|_2^2 = 0.0474087 < 0.050$ 。因此, 这对多项式是 ϵ -近似问题的解。

第三, 宽松的近似问题。给定一对多项式, 要找到“附近”的一对有非平凡公因子的多项式。因为对于一些近似问题, 我们无法找到全局最优解和 ϵ -近似解。

上面只是用近似最大公因子问题举个例子。我们可以把它推广到近似代数中的其它问题上去。也就是说, 近似多项式代数中的有下面三类问题值得考虑[51]。

1. 全局最优问题。给定一个多项式组 F , 要找到离这组多项式最近的一组多项式, 而且该多项式组满足某种性质 $\mathfrak{F}$ 。
2. ϵ -近似问题。给定一个多项式组 F , 要找到一组多项式满足某种性质 $\mathfrak{F}$, 且这两组多项式的距离不超过 ϵ 。
3. 宽松的近似问题。给定一个多项式组 F , 要在 F 附近找到一组多项式满足某种性质 $\mathfrak{F}$ 。

本文的第二章考虑的近似因式分解问题是上面所说的第三种类型的问题。而本文的第三章和第四章考虑的近似最大公因子问题就是上面所说的第一种类型的问题。关于这些问题, 将在后面的章节具体解释。

§1.2 多元多项式的近似因式分解的简介

近些年, 有很多的文章研究在实数域和复数域上的近似因式分解问题。文章[31] (第 6 部分) 第一个提出来需要对系数不是准确的多项式进行因式分解。另外, 文章[32; 33] 提出将不准确的多项式因式分解问题转化为一个优化问题。

多变元多项式的近似因式分解可以转化为关于多项式因子的待定系数的非线性优化问题。文章[26]给出了一个用参数最小二乘的算法计算最近的有全次数为 d 因子的多项式。文章[28; 66; 14; 6; 15; 64; 69]运用 Hensel 提升或者曲线插值等方法解决了输入的多项式离可以因式分解的多项式比较近的情形时的近似因式分解问题。

而对于任意给定的多项式, 我们第二章给出的算法可以得到给定多项式附近的一个可以分解的多项式。

还有一些相关的工作[4]是用浮点数算法来计算准确的因式分解问题, 这里输入的多项式的系数是准确的情形。文章[52; 34; 53]给出了输入多项式的不可约的半径。这些界不但对于多项式的数值不可分解很有用, 而且可以判断一个已经计算的近似因式分解是否不可约分解。

§1.3 多项式的近似最大公因子的简介

对于计算多项式的近似最大公因子, 有很多作者做了这方面的工作。文章[40]给出了一个多项式时间算法计算单变元多项式的近似最大公因子问题。文章[67; 54; 7; 13; 27; 3; 56; 65; 77; 10; 74; 75]考虑了单变元多项式的 ϵ -近似最大公因子问题和宽松的近似最大公因子问题。其中基于文章[10]的 Maple 程序包 SNAP 中的 QRGCD 程序就是这些算法中一个非常好的例子。

关于多变元多项式的近似最大公因子问题, 现在还没有多项式时间算法。到目前为止, 已经有了很多算法[55; 54; 7; 27; 79; 56; 65; 73; 76] 计算宽松的近似最大公因子问题。

在本文的第三章中, 希望计算得到近似最大公因子的全局最优问题的解。这一章用的是基于罚函数的结构最小二乘方法 (STLN)。理论上说, STLN 算法只能得到局部最优解。但是实验说明, 对于单变元的近似最大公因子问题, STLN 算法可以得到全局最优解。

在这里讨论的另一个问题是计算多项式的系数有线性限制的近似最大公因子问题。保持首一性是多项式系数有线性限制的一种情形。对于单变元的情形, 系数有线性限制的近似最大公因子问题可以应用于计算最接近的有 k 重根的奇异多项式问题。文章[41; 25; 81; 80]给出了其它的算法去求解这个问题。

§1.4 论文的主要内容

本文所要讨论的是计算多变元多项式的近似因式分解以及单变元和多变元多项式的近似最大公因子问题。论文的余下部分由下述章节构成:

在第二章中首先给出了一个多变元多项式的近似除法的算法和近似最大公因子算法。然后运用这两个算法, 给出了一个多变元多项式的近似因式分解算法。当输入的多项式离可以分解的多项式的距离比较远时, 我们的算法仍然能得到比较好的结果。而且可以用 Gauss-Newton 迭代优化这个结果。一般情况下, 迭代的次数小于 10 次。最后, 给出一个表来说明和以前的算法相比较的结果。

第三章主要给出了一个新的算法计算多项式的近似最大公因子问题，这一章考虑了单变元多项式和多变元多项式两种情形。我们把计算最近最大公因子问题转化为对（广义）Sylvester 矩阵的结构低秩逼近，用的是结构最小二乘范数的方法（STLN）。跟以前的算法相比，STLN 算法能得到更小的向后误差，而且迭代的次数比较少（一般情况下，只需要两三次）。

第四章主要讨论多项式的系数有线性限制条件的近似最大公因子问题。这一章把计算两个多项式的近似最大公因子算法推广到计算多个多项式的近似最大公因子算法。当然，最主要的是对于多项式系数之间满足线性关系等式或者不等式情形，我们都给出了有效的算法计算多项式近似最大公因子的问题。

对于单变元的情形，系数有线性限制的近似最大公因子问题可以应用于求解最接近的有 k 重根的奇异多项式问题，这是一个全局优化问题。和以前算法相比较，我们的算法可以通过更简便的计算得到最近的奇异多项式问题的全局最优解。在多变元的情形，计算最近的有重因子的多项式的无平方因子分解问题可以转化为求解系数有线性限制的多项式近似最大公因子问题，这对多项式的近似因式分解算法有非常重要的作用。

第五章总结我们所做的工作并且主要讨论了将来的研究方向和展望。

第二章 多元多项式的近似因式分解

§2.1 前言

当输入一个多项式时,由于数值计算或者物理的测量容易产生误差,这个多项式就会丢失了在准确意义上的可以分解的性质。在这部分我们考虑求多元多项式的复的近似因子。 $f(x_1, \dots, x_n) \in \mathbb{C}[x_1, \dots, x_n]$ 在复数 $\mathbb{C}$ 上绝对不可约,这里不可约是由于多项式 f 的系数有扰动造成的。我们用 $\|f\|$ 代表 f 的多项式系数向量的 Euclidean 长度,用 $f^{[\min]}$ 代表在 $\mathbb{C}$ 上可以分解的多项式而且满足 $\deg(f^{[\min]}) \leq \deg(f)$ 和 $\|f - f^{[\min]}\|$ 最小的,也就是说, $f^{[\min]}$ 是最近的可分解的多项式。这一章还给出了一个新的算法来计算因式分解: $\tilde{f} = f_1 \cdot f_2 \cdots f_r \in \mathbb{C}[x_1, \dots, x_n]$, 而且有 $\deg(\tilde{f}) \leq \deg(f)$, 使得 $\|\tilde{f} - f^{[\min]}\|$ 非常小。

文章[34]中的例 2 说明了 $f^{[\min]}$ 是与次数的定义有关。算法中的次数限制 $\deg(\tilde{f}) \leq \deg(f)$ 是对每个变元都有限制的,也就是 $\deg_{x_i}(\tilde{f}) \leq \deg_{x_i}(f), i = 1, \dots, n$ 。我们的算法将[16]中的准确的算法推广到计算近似多项式的因式分解。我们通过 Ruppert 矩阵的奇异值分解来确定因子的个数。另外还设计了新的算法来计算多变元多项式的近似最大公因子。

这一章文章还给出了很多近似多项式因式分解的例子,其中包括那些不可约范围内的例子(见小节 §2.3.6)。我们能在更短的时间内计算近似因式分解,并且得到了更小的误差。另外,我们还测试了一些随机生成的例子。实验结果表明我们的算法具有鲁棒性。

§2.2 多元多项式近似除法和近似最大公因子

在多元多项式近似绝对不可约分解算法中,需要有复多项式的近似最大公因子和复多项式的近似除法的算法。下面先考虑多项式的近似除法。

§2.2.1 多项式的近似除法

给定的多项式 f 与一个全次数是 d 的多项式相乘的运算是一个关于多项式 f

的系数的线性算子, 用卷积矩阵 $C^{[d]}f$ 表示。例如:

$$\overrightarrow{(a_2x + a_1y + a_0) \cdot (b_2x + b_1y + b_0)} = C^{[2]}a_2x + a_1y + a_0 \cdot \begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix} = \begin{bmatrix} a_2 & 0 & 0 \\ a_1 & a_2 & 0 \\ 0 & a_1 & 0 \\ a_0 & 0 & a_2 \\ 0 & a_0 & a_1 \\ 0 & 0 & a_0 \end{bmatrix} \cdot \begin{bmatrix} b_2 \\ b_1 \\ b_0 \end{bmatrix}.$$

另外, 卷积矩阵可以用其它种类的次数或者用多项式给定的支集来构造。但是, 为了方便, 文章中的次数都是表示多项式的全次数。

给定两个多项式 f 和 g , $\text{tdeg}(g) \geq \text{tdeg}(f)$, 而且 f 不能整除 g 。我们想对多项式的系数进行扰动, 使得 f 能整除 g 。改变多项式 g 的系数得到离 g 最近的多项式 $\tilde{g}$, 使得 f 整除 $\tilde{g}$ 。这可以通过计算一个最小二乘问题得到:

$$\min_{\text{tdeg}(q)=\text{tdeg}(g)-\text{tdeg}(f)} \|fq - g\|_2. \quad (2.1)$$

根据卷积矩阵准确写出多项式 $\tilde{g}$,

$$\tilde{g} = C^{[\text{tdeg}(g)]}f (C^{[\text{tdeg}(g)]}f)^\dagger g$$

这里 $(C^{[\text{tdeg}(g)]}f)^\dagger$ 表示矩阵 $C^{[\text{tdeg}(g)]}f$ 的广义逆矩阵。在单变元的情形, 我们可以采用快速算法。这是因为最小二乘问题(2.1) 的系数矩阵有很小的位移秩, 所以快速算法可以有效地解决这个问题[78]。

上面方法的缺点是: 尽管它能计算近似除法问题, 但是它没有让多项式 f 和 g 的系数同时改变。

假设多项式 f 和 g 的系数都可以扰动, 那么近似除法问题是一个最小二乘问题(矩阵中的元素和等式右边的向量都可以扰动)。实际上, 因为矩阵 $C^{[d]}f$ 有特殊的结构, 所以近似除法可以转化为结构最小二乘问题。

§2.2.2 近似最大公因子

§2.2.2.1 算法的描述

当把 Gao 的算法推广到近似的情形时, 我们需要一个好的近似最大公因子算法。

首先给出一个简单的引理, 它是近似最大公因子的一个基础。

引理 2.2.1. 给定两个非零的多项式 $g, h \in \mathbb{C}[x_1, \dots, x_n]$ 。如果 $g_1 = g/\text{GCD}(g, h)$, $h_1 = h/\text{GCD}(g, h)$ 。那么方程

$$ug + vh = 0 \quad (2.2)$$

的所有解 $u, v \in \mathbb{C}[x_1, \dots, x_n]$ 必须是下面的形式:

$$u = h_1 q, \quad v = -g_1 q, \quad (2.3)$$

这里 $q \in \mathbb{C}[x_1, \dots, x_n]$ 。

证明. 式子 2.2 可以写成:

$$ug_1 \text{GCD}(g, h) + vh_1 \text{GCD}(g, h) = 0$$

那么上面的式子同样可以写成:

$$ug_1 + vh_1 = 0$$

由于 g_1 和 h_1 互素, 所以 g_1 和 h_1 可以分别整除 v 和 u 。 □

方程(2.2)是一个由多项式 u 和 v 的系数构成的线性系统。为了让它是有限系统, 这里需要限制 u 和 v 的次数。我们可以按照多项式的每个变元的次数, 也可以是多项式的全次数。这里用的是多项式的全次数 (tdeg), 而且要求:

$$\text{tdeg}(u) \leq \text{tdeg}(h) - 1, \quad \text{tdeg}(v) \leq \text{tdeg}(g) - 1. \quad (2.4)$$

那么, $\text{GCD}(g, h) = 1$ 当且仅当没有非零的多项式 u 和 v 是(2.2) 和(2.4)的解。 $\text{GCD}(g, h)$ 的次数和解空间的维数有关系。为了说明这一点, 首先, 全次数 $\leq d$ 的单项式 $x_1^{i_1} \cdots x_n^{i_n}$ 的个数是二项式数:

$$\beta(d, n) = \binom{d+n}{n}.$$

因此, u 有 $\beta(\text{tdeg}(h) - 1, n)$ 个系数, v 有 $\beta(\text{tdeg}(g) - 1, n)$ 个系数。这样, 线性系统的变元的个数就是

$$m = \beta(\text{tdeg}(g) - 1, n) + \beta(\text{tdeg}(h) - 1, n).$$

根据(2.3) 和(2.4), u 和 v 的所有解都被多项式 $q \in \mathbb{C}[x_1, \dots, x_n]$ 所决定, 而且

$$\text{tdeg}(q) \leq \ell - 1,$$

这里 ℓ 是 $\text{GCD}(g, h)$ 的全次数。因此, u 和 v 解空间的维数是等于

$$\beta(\ell - 1, n).$$

因此, 我们可以计算系数矩阵的秩 p , 然后从 $m - p = \beta(\ell - 1, n)$ 中得到 ℓ 。

对于单变元多项式, (2.2) 和 (2.4) 的系数矩阵就是 g 和 h 构造的通常意义的 Sylvester 矩阵。在文章[8]中, Sylvester 矩阵可以用来计算单变元多项式的近似最大公因子。

对于多变元多项式, 可以用广义 Sylvester 矩阵来表示 (2.2) 和 (2.4) 相对应的系数矩阵。同样, 它可以用来计算多变元多项式的最大公因子。这里要计算的最低次数的解也就是多项式的余因子 g_1 和 h_1 。

S_1 表示 g 和 h 的广义 Sylvester 矩阵。为了求 GCD, 需要找到 S_1 零空间的基。

在准确情形, 我们可以通过对零空间的基做高斯消去法得到 g 和 h 的余因子。然后做除法就能计算出 GCD。在数值情形, 有两个困难需要解决。

第一, 在准确的情形 S_1 可能是满秩的。第二, 用高斯消去法来求最低次数的解是不稳定的。

为了克服第一个困难, 奇异值分解可以用来找 (低) 秩为 p 的最近矩阵。最后 $m - p$ 个奇异向量构成最近低秩矩阵的零空间的基。我们用 S_1 的奇异值来确定数值秩 p 。当确定数值秩时, 需要给一个容许误差 ϵ 或者在奇异值中找一个跳跃, 即最大比例 σ_{i+1}/σ_i :

$$\sigma_m \geq \cdots \geq \sigma_{m-p-1} > \epsilon \geq \sigma_{m-p} \geq \cdots \geq \sigma_1.$$

另外一个方法是: 给多项式 g 和 h 除了一个变元外其它变元都赋值, 然后找到单变元多项式 GCD 的“最好”的次数。容许误差 ϵ 可以从因式分解问题中得到。当 ϵ 很小的时候, 这种方法也很有效。因为用赋值的方法得到的 Sylvester 矩阵的维数很小, 即使我们多算几次赋值, 这种方法还是要比算广义 Sylvester 矩阵奇异值快很多。因此, 在实际计算过程中我们用这种方法来确定数值秩。

对于第二个困难可以通过删掉 S_1 的一些行和列来解决。如果知道 S_1 的秩, 可以用更强的次数 (未知的多项式 u 和 v) 限制条件来构造线性系统 $ug + vh = 0$ 。因此, 得到一个新的广义 Sylvester 矩阵 S_k (S_1 的子矩阵), 而且 S_k 的零空间的维数为 1, 这里 k 是 $\text{GCD}(g, h)$ 的全次数。在这种情况下, 零空间中的基的个数为 1, 也就是余因子 g_1 和 h_1 的常数倍。这个零向量可以用迭代的方法[47] 来计算。当有了近似的余因子 g_1 和 h_1 , 近似最大公因子可以通过最小二乘法做近似除法得到。

算法 2.2.1. AMVGCD: 近似最大公因子

输入: g 和 $h \in \mathbb{C}[x_1, \dots, x_n]$ 。

输出: d , g 和 h 的非常数的近似最大公因子。

1. 确定 k , g 和 h 的近似最大公因子的次数, 可以用下面的两种方法中的一种:

- (a) 构造线性系统 $ug + vh = 0$ 生成的矩阵 $S = S_1(g, h)$, 这里 $g, h \in \mathbb{C}[x_1, \dots, x_n]$, 而且满足 $\text{tdeg}(u) < \text{tdeg}(h)$ 和 $\text{tdeg}(v) < \text{tdeg}(g)$ 。根据 S 的奇异值找到最大跳跃。然后从 S 的数值秩得到次数 k 。
- (b) 通过对 g 和 h 做几个任意的单变元的映射, 计算 *Sylvester* 矩阵的数值秩。然后确定次数 k 。
2. 用更强的两个次数限制 $\text{tdeg}(u) = \text{tdeg}(h) - k$ 和 $\text{tdeg}(v) = \text{tdeg}(g) - k$ 来构造线性系统, 重新得到广义 *Sylvester* 矩阵 $S_k(g, h)$, 记为 S 。新的矩阵 S 的零空间的维数为 1。
3. 计算 S 的最小的奇异值所对应的奇异向量, 给出 $[u, v]^T$ 的一个解。
4. 计算出 h 和 u (或者 g 和 v) 的近似商, 得到 d , 或者通过解决最小二乘问题 $\|h - du\|_2^2 + \|g + dv\|_2^2$ 得到 d 。

如果想指定容许误差, 那么只有第一步需要改变。在这种情形下, 可能得到 $k = 0$, 如果出现这种情况, 直接返回结果 $d = 1$, 这也就是说, g 和 h 在这个指定的容许误差下是近似相对互素的。

另外, 后面一章将要讨论怎样用结构最小二乘问题来计算近似最大公因子。

§2.2.2.2 算法的收敛

为了简单起见, 这部分只考虑两个变元 ($n = 2$) 的情形。设多项式 $\tilde{g}$ 和 $\tilde{h}$ 相对互素, 首先对两个多项式正规化, 也就是使得 $\|\tilde{h}\|_2 = \|\tilde{g}\|_2 = 1$ 。

假设 $\text{GCD}(g, h) = d \neq 1$, $\text{tdeg}(g) = \text{tdeg}(\tilde{g})$, $\text{tdeg}(h) = \text{tdeg}(\tilde{h})$, 设 $\|g - \tilde{g}\|_2 = \epsilon_1$, $\|h - \tilde{h}\|_2 = \epsilon_2$ 。我们能证明: 当 $\epsilon_1, \epsilon_2 \rightarrow 0$, 则 $\tilde{g}$ 和 $\tilde{h}$ 的近似最大公因子趋近 d 。

设 $S = S_k(g, h)$, 这里 k 使得 S 的秩为 p , 亏秩为 1。那么, $w = [u, v]$ 是 S 的零空间的基, 这里 $u = h/d$, $v = -g/d$, 不失一般性, 设 $\|u\|_2 = 1$ 。设 $\tilde{S} = S_k(\tilde{g}, \tilde{h})$ 。我们可以限制这两个 *Sylvester* 矩阵的距离:

$$\|S - \tilde{S}\|_2^2 \leq \|S - \tilde{S}\|_F^2 = a_1\epsilon_1^2 + a_2\epsilon_2^2 = \epsilon_3^2$$

这里

$$a_1 = \beta(\text{tdeg}(h) - k, 2) \text{ 和 } a_2 = \beta(\text{tdeg}(g) - k, 2)$$

只依赖 k 和 g, h 的次数。

计算 SVD 找到矩阵 M 使得:

$$\min_{\text{Rank}(M)=\text{Rank}(S)} \|\tilde{S} - M\|_2 = \sigma_{m-p}(\tilde{S}) \leq \epsilon_3$$

注意 M 不一定是 Sylvester 矩阵, 但是满足 $\|M - S\|_2 \leq 2\epsilon_3$, 并且 M 的亏秩为 1。设 $\tilde{w} = [\tilde{u}, \tilde{v}]$ 是由 M 的零空间得到的向量, $\|\tilde{u}\|_2 = 1$ 。在定理 6.4 [70] 中说明: 根据 ϵ_3 可以限制 w 和 $\tilde{w}$ 的距离, 使得当 $\epsilon_3 \rightarrow 0$ 时, $\epsilon_4 = \|w - \tilde{w}\|_2 \rightarrow 0$ 。这样因为足够小的 ϵ_1 和 ϵ_2 , 我们有 $\text{tdeg}(\tilde{u}) = \text{tdeg}(u)$ 。

另外需要用[20, 第 134-139 页]中的因子系数的限制: $\|f_1\|_2 \|f_2\|_2 \leq 2^{\sum_i (\deg_{x_i} f)} \|f\|_2$, $f = f_1 f_2$ 。用最小二乘方法可以求出多项式 $\tilde{d}$:

$$\epsilon_5 = \min_{\tilde{d}: \text{tdeg}(\tilde{d}) \leq \text{tdeg}(d)} \|\tilde{d}\tilde{u} - \tilde{h}\|_2$$

同时 ϵ_5 满足一个限制:

$$\begin{aligned} \epsilon_5 &= \|\tilde{d}\tilde{u} - \tilde{h}\|_2 \leq \|d(\tilde{u} - u) - (\tilde{h} - h)\|_2 \\ &\leq \|d(\tilde{u} - u)\|_1 + \epsilon_2 \leq \|d\|_1 \|\tilde{u} - u\|_1 + \epsilon_2 \\ &\leq \binom{k+2}{2}^{1/2} \|d\|_2 \binom{\text{tdeg}(h)-k+2}{2}^{1/2} \|\tilde{u} - u\|_2 + \epsilon_2 \\ &\leq a_4 \epsilon_4 + \epsilon_2 \end{aligned}$$

根据多变元多项式的因子系数的限制得到:

$$a_4 = 2^{\deg_x(h) + \deg_y(h)} \|h\|_2 \binom{k+2}{2}^{1/2} \binom{\text{tdeg}(h)-k+2}{2}^{1/2}$$

则:

$$\begin{aligned} \|(d - \tilde{d})u\|_2 &= \|h - \tilde{d}\tilde{u} - \tilde{d}(u - \tilde{u})\|_2 \\ &\leq \|h - \tilde{h} + (\tilde{h} - \tilde{d}\tilde{u})\|_2 + \|\tilde{d}(u - \tilde{u})\|_2 \\ &\leq \|h - \tilde{h}\|_2 + \|\tilde{h} - \tilde{d}\tilde{u}\|_2 + \|\tilde{d}(u - \tilde{u})\|_1 \\ &\leq \epsilon_2 + \epsilon_5 + \|\tilde{d}\|_1 \|u - \tilde{u}\|_1 \\ &\leq 2\epsilon_2 + a_4 \epsilon_4 + \binom{k+2}{2}^{1/2} \|\tilde{d}\|_2 \binom{\text{tdeg}(h)-k+2}{2}^{1/2} \epsilon_4, \end{aligned}$$

$\|\tilde{d}\|_2$ 可以用下面的式子进行限制:

$$\begin{aligned} \|\tilde{d}\|_2 &\leq 2^{\deg_x(h) + \deg_y(h)} \|\tilde{d}\tilde{u}\|_2 \\ &\leq 2^{\deg_x(h) + \deg_y(h)} (\|\tilde{h}\|_2 + \|\tilde{d}\tilde{u} - \tilde{h}\|_2) \\ &\leq 2^{\deg_x(h) + \deg_y(h)} (\|\tilde{h}\|_2 + \epsilon_2 + a_4 \epsilon_4). \end{aligned}$$

再次用多变元因子系数的限制可以得到:

$$\|d - \tilde{d}\|_2 \leq 2^{\deg_x(h) + \deg_y(h)} (2\epsilon_2 + a_5 \epsilon_4)$$

a_5 可以从上面两个式子得到。

因为 $\epsilon_4 \rightarrow 0$, 而且 a_5 是由 $\tilde{g}$ 和 $\tilde{h}$ 的次数和项数限定了, 所以当 $\epsilon_1, \epsilon_2 \rightarrow 0$ 时有 $\|d - \tilde{d}\|_2 \rightarrow 0$ 。在实际计算过程中, 计算得到的 $\tilde{d}$ 比上面所给的界小得多。

§2.3 因式分解的算法和实验

我们将 [16] 中的两个变元的算法推广到一个计算复多元多项式的近似因式分解。另外, 用 Gauss-Newton 方法优化近似因子。我们用 Maple 程序实现了算法, 并且给了一些例子。

§2.3.1 准确多元多项式绝对不可约因式分解

假设 f 不是常数, 而且 $\text{GCD}(f, f_{x_1}) = 1$, $f_{x_1} = \partial f / \partial x_1$, 也就是 f 是无平方因子的, 且在 $\mathbb{C}[x_2, \dots, x_n]$ 上没有因子。假设 f 可以分解为:

$$f = f_1 f_2 \cdots f_r, \quad (2.5)$$

$f_i \in \mathbb{C}[x_1, x_2, \dots, x_n]$ 是 $\mathbb{C}$ 上不同的不可约因子。定义

$$E_i = \frac{f}{f_i} \frac{\partial f_i}{\partial x_1} \in \mathbb{C}[x_1, x_2, \dots, x_n] \quad 1 \leq i \leq r. \quad (2.6)$$

那么

$$f_{x_1} = E_1 + E_2 + \cdots + E_r \text{ 和 } E_i E_j \equiv 0 \pmod{f} \text{ 对于所有的 } i \neq j. \quad (2.7)$$

下面给出因式分解算法中重要的绝对不可约的判定定理。首先, 引用文章[63] 中两个变元多项式不可约的判定定理:

引理 2.3.1. 设 $f \in \mathbb{C}[x, y]$ 次数为 (m, n) , 即 $\deg_x f = m$, $\deg_y f = n$ 。那么 f 是绝对不可约当且仅当方程

$$\frac{\partial}{\partial y} \left(\frac{g}{f} \right) = \frac{\partial}{\partial x} \left(\frac{h}{f} \right), \quad (2.8)$$

没有非零的解 $g, h \in \mathbb{C}[x, y]$, 这里 $\deg g \leq (m-1, n)$, $\deg h \leq (m, n-2)$ 。

根据引理2.3.1, 可以把不可约的判定推广到多变元多项式情形。

定理 2.3.1. 设 $f \in \mathbb{C}[x_1, x_2, \dots, x_n]$ 次数为 $(d_1, d_2, \dots, d_n)$, 即 $\deg_{x_i} f = d_i$ 。那么, f 是绝对不可约当且仅当方程

$$\frac{\partial}{\partial x_i} \left(\frac{g}{f} \right) = \frac{\partial}{\partial x_1} \left(\frac{h_i}{f} \right), \quad i = 2, \dots, n \quad (2.9)$$

没有非零的 $g, h_2, \dots, h_n \in \mathbb{C}[x_1, x_2, \dots, x_n]$ 满足

$$\deg g \leq (d_1 - 2, d_2, \dots, d_n),$$

$$\deg h_i \leq (d_1, d_2, \dots, d_i - 1, \dots, d_n), i = 2, \dots, n.$$

根据定理 2.3.1 的判定规则, 文章[34]给出了一些界来判定一个数值多项式是否是绝对不可约的。当这些界很小时, 则多项式 f 离可约多项式很近。下面解释如何用定理 2.3.1 进行因式分解。

首先, 对于两个变元的情形, 需要改变引理 2.3.1 中的 g 和 h 的次数条件:

$$\deg g \leq (m-1, n), \deg h \leq (m, n-1), \quad (2.10)$$

这样即使 f 不可约, $(g, h) = (f_x, f_y)$ 都是微分方程的解。下面给出一个引理:

引理 2.3.2. [16] 设 $f \in \mathbb{C}[x, y]$ 是非常数的多项式, 次数为 (m, n) , 而且有 $GCD(f, f_x) = 1$ 。定义

$$G = \{g \in \mathbb{C}[x, y] : h \in \mathbb{C}[x, y] \text{ 满足 (2.8) 和 (2.10)}\} \quad (2.11)$$

设 f 分解为不可约的多项式 (2.5)。那么 G 是 $\mathbb{C}$ 上维数为 r 的向量空间, 而且每个 $g \in G$ 都有形式: $g = \sum \lambda_i E_i$, 这里 $\lambda_i \in \mathbb{C}$ 。

上面的引理可以推广到多变元多项式的情形。首先要改变 g 和 h_i 的次数条件:

$$\begin{aligned} \deg g &\leq (d_1 - 1, d_2, \dots, d_n), \\ \deg h_i &\leq (d_1, d_2, \dots, d_i - 1, \dots, d_n), i = 2, \dots, n, \end{aligned} \quad (2.12)$$

当 f 是不可约的, $(f_{x_1}, f_{x_2}, \dots, f_{x_n})$ 都是微分方程 (2.9) 的解。因为微分在 $\mathbb{C}$ 上是线性的, 次数界 (2.12) 和方程 (2.9) 给出了一个关于 g 和 h_i 系数的线性系统, 这个系数矩阵称为 Ruppert 矩阵, 记为 $\text{Rup}(f)$ 。

定理 2.3.2. 设 $f \in \mathbb{C}[x_1, x_2, \dots, x_n]$ 是非常数的多项式, 次数为 $(d_1, d_2, \dots, d_n)$, 而且 $GCD(f, f_{x_1}) = 1$ 。定义

$$G = \{g \in \mathbb{C}[x_1, x_2, \dots, x_n] : h_2, \dots, h_n \in \mathbb{C}[x_1, x_2, \dots, x_n] \text{ 满足 (2.9) 和 (2.12)}\} \quad (2.13)$$

设 f 可以分解为不可约多项式的乘积 (2.5)。那么 G 是在 $\mathbb{C}$ 上维数为 r 的向量空间, 而且每个 $g \in G$ 有形式: $g = \sum_{i=1}^r \lambda_i E_i$, 这里 $\lambda_i \in \mathbb{C}$ 。

证明. 这个证明可以直接由引理 2.3.2 推广得到。首先, 对于 $g = E_k = \frac{f}{f_k} \frac{\partial f_k}{\partial x_1}$, 设 $h_i = \frac{f}{f_k} \frac{\partial f_k}{\partial x_i}$, $i = 2, \dots, n$ 。那么 $(g, h_2, \dots, h_n)$ 满足 (2.12) 和

$$\frac{\partial}{\partial x_i} \left(\frac{g}{f} \right) = \frac{\partial}{\partial x_i} \left(\frac{1}{f_k} \frac{\partial f_k}{\partial x_1} \right) = \frac{\partial}{\partial x_1} \left(\frac{1}{f_k} \frac{\partial f_k}{\partial x_i} \right) = \frac{\partial}{\partial x_1} \left(\frac{h_i}{f} \right),$$

$i = 2, \dots, n$, 而且 $E_1, \dots, E_r \in G$ 。因为 $E_1, \dots, E_r$ 满足 (2.7), 他们在 $\mathbb{C}$ 上线性无关。因此 $\dim_{\mathbb{C}} G \geq r$ 。

设 $g \in G$ 且 $h_2, \dots, h_n \in \mathbb{C}[x_1, x_2, \dots, x_n]$ 满足 (2.9) 和 (2.12)。我们需要证明 g 是 $E_1, \dots, E_r$ 在 $\mathbb{C}$ 上的线性组合。因为 $\text{GCD}(f, f_{x_1}) = 1$, f 在 $\mathbb{C}(x_2, \dots, x_n)$ 的代数闭包里没有重因子。设

$$f = u_{d_1} \prod_{i=1}^{d_1} (x_1 - c_i), c_i \in \overline{\mathbb{C}(x_2, \dots, x_n)}.$$

因为 $\deg_{x_1} g < \deg_{x_1} f$ 和 $\deg_{x_1} h_j \leq \deg_{x_1} f$, 我们由分式分解有:

$$\frac{g}{f} = \sum_{i=1}^{d_1} \frac{a_i}{x_1 - c_i}, \quad \frac{h_j}{f} = \sum_{i=1}^{d_1} \frac{b_{ji}}{x_1 - c_i} + h_j^*,$$

这里 $\deg_{x_1} h_j^* = 0$,

$$a_i = g(c_i, x_2, \dots, x_n) / f_{x_1}(c_i, x_2, \dots, x_n). \quad (2.14)$$

因为

$$\begin{aligned} \frac{\partial}{\partial x_j} \left(\frac{g}{f} \right) &= \sum_{i=1}^{d_1} \left(\frac{1}{x_1 - c_i} \frac{\partial a_i}{\partial x_j} + \frac{a_i}{(x_1 - c_i)^2} \frac{\partial c_i}{\partial x_j} \right), \\ \frac{\partial}{\partial x_1} \left(\frac{h_j}{f} \right) &= \sum_{i=1}^{d_1} \frac{-b_{ij}}{(x_1 - c_i)^2}. \end{aligned}$$

方程 (2.9) 表示 $\frac{\partial a_i}{\partial x_j} = 0$, 也就是 $a_i \in \mathbb{C}$, $i = 1..d_1$ 。如果 c_i 和 c_j 在 $\overline{\mathbb{C}(x_2, \dots, x_n)}$ 上代数共轭的, 根据 (2.14), a_i 和 a_j 同样代数共轭的。因为 $a_i, a_j \in \mathbb{C}$, 可以得到 $a_i = a_j$ 。

因此, 对于 c_i 在对应于 f 的不可约因子在 $\mathbb{C}(x_2, \dots, x_n)$ 上的同一个共轭类里, a_i 是个常数。跟[16]一样, 有

$$\frac{g}{f} = \sum_{i=1}^r \lambda_i \frac{1}{f_i} \frac{\partial f_i}{\partial x_1},$$

这里 $\lambda_i \in \mathbb{C}$ 。因此, 每一个 $g \in G$ 有形式 $g = \sum_{i=1}^r \lambda_i E_i$ 且 $\dim_{\mathbb{C}} G = r$ 。 $\square$

接着说明怎样从线性空间 G 来分解多项式 f 。对于二元多项式分解, 文章[16]有如下引理:

引理 2.3.3. 设 $g_1, \dots, g_r$ 是 G 在 $\mathbb{C}$ 上的基。选取 $s_i \in S \subset \mathbb{C}$, 对于 $1 \leq i \leq r$, s_i 是任意取得且线性无关的。设 $g = \sum_{i=1}^r s_i g_i$ 。则存在唯一的矩阵 $A = [a_{i,j}] \in \mathbb{C}^{r \times r}$ 使得

$$gg_i \equiv \sum_{j=1}^r a_{i,j} g_j f_x \pmod{f} \in \mathbb{C}(y)[x] \quad (2.15)$$

而且, 设 $E_g(x) = \det(Ix - A)$ 是 A 的特征多项式。那么,

$$f = \prod_{\lambda \in \mathbb{C}: E_g(\lambda)=0} \text{GCD}(f, g - \lambda f_x) \quad (2.16)$$

给出了 f 在 $\mathbb{C}$ 上绝对不可约分解的概率不小于 $1 - r(r-1)/(2|S|)$ 。

上面的引理可以推广到多变元多项式的情形:

定理 2.3.3. 设 $g_1, \dots, g_r$ 是 G 在 $\mathbb{C}$ 上的基。选取 $s_i \in S \subset \mathbb{C}$, 对于 $1 \leq i \leq r$, s_i 是任意取得且线性无关的。设 $g = \sum_{i=1}^r s_i g_i$ 。则存在唯一的矩阵 $A = [a_{i,j}] \in \mathbb{C}^{r \times r}$ 使得

$$gg_i \equiv \sum_{j=1}^r a_{i,j} g_j f_{x_1} \pmod{f} \in \mathbb{C}(x_2, \dots, x_n)[x_1]. \quad (2.17)$$

而且, 设 $E_g(x) = \det(Ix - A)$ 是 A 的特征多项式。那么,

$$f = \prod_{\lambda \in \mathbb{C}: E_g(\lambda)=0} \text{GCD}(f, g - \lambda f_{x_1}) \quad (2.18)$$

给出了 f 在 $\mathbb{C}$ 上绝对不可约分解的概率不小于 $1 - r(r-1)/(2|S|)$ 。

§2.3.2 近似多元多项式绝对不可约因式分解

为了把[16]中的准确因式分解算法推广到近似情形, 以下几个问题需要解决:

1. 计算多元多项式的近似最大公因子: $\text{GCD}(f, g - \lambda_i f_{x_1})$,
2. 约化多项式 f , 使得多项式 f 和 f_{x_1} 近似相对互素,
3. 确定 G 的数值意义上的维数。
4. 计算 A_g 使其没有重的特征值。

对于第一个问题, 上面一小节已经给出了一个比较好的算法来计算多变元多项式的近似最大公因子。对于第二个问题, 同样可以用近似最大公因子来解决: 首先计算出 f 和 f_{x_1} 的近似最大公因子。然后用近似除法, $f/\text{GCD}(f, f_{x_1})$ 把多项式变成 $\text{GCD}(f, f_{x_1}) = 1$ 的情形。关于这些细节将在 §2.3.4 部分介绍。

为了解决第三个问题, 我们对 $\text{Rup}(f)$ 矩阵做奇异值分解来确定 G 的数值维数。设 σ_i 是 $\text{Rup}(f)$ 第 i 个奇异值。如果容许误差 ϵ 已经给定, 则 G 的数值维数为 r , 这里 r 满足

$$\cdots \geq \sigma_{r+2} \geq \sigma_{r+1} > \epsilon \geq \sigma_r \geq \cdots \geq \sigma_1.$$

然而, 如果 f 的系数的相对误差是未知的, 那就很难给出与相对误差相对应的容许误差 ϵ 。这时, 可以根据奇异值的最大跳跃来指定容许误差。也就是, 选择

$\epsilon = \sigma_r$ 使得 σ_{r+1}/σ_r 尽可能大。文章[34]指出, 奇异值 σ_r 小于等于 f 和最近的有 r 个不可约因子的多项式 $\tilde{f}$ 的距离。也就是:

$$\min_{\substack{\deg \tilde{f}=(d_1, \dots, d_n) \\ \dim \text{Nullspace}(\text{Rup}(\tilde{f}))=r}} \|\text{Rup}(f) - \text{Rup}(\tilde{f})\|_2 \geq \sigma_r.$$

这个指定的容许误差 σ_r 可以作为后面求多变元多项式近似最大公因子的输入的容许误差。

注 2.3.1. 理论上, 可以用保持结构的低秩逼近方法 (SPLRA) [57] 来得到离 $\text{Rup}(f)$ 最近的矩阵 $\tilde{R}$, 而且 $\tilde{R}$ 亏秩为 r 。因为 $\tilde{R}$ 保持了 $\text{Rup}(f)$ 的结构, 所以 $\tilde{R}$ 所对应的多项式 $\tilde{f}$ 是最近的有 r 个绝对不可约因子的多项式。然而, SPLRA 方法应用到例子上, 计算得到的结果并不理想, 希望在以后的工作中能进一步研究探讨。在这里仍然对 $\text{Rup}(f)$ 用奇异值分解来找到离它最近的亏秩的矩阵。

下面讨论第四个问题。假设从 $\text{Rup}(f)$ 的最后 r 个奇异值所对应的奇异向量已经得到 G 的近似基。容易得到 $\|\text{Rup}(f)g_i\|_2 \leq \sigma_i \leq \sigma_r$ 。因此, 向量 g_i 构成了在容许误差 σ_r 下的 G 的近似基。根据定理 2.3.3 来构造矩阵 A_g 。任意取一些复数 $s_1, \dots, s_r \in S \subset \mathbb{C}$, 而且设 $g = \sum_{i=1}^r s_i g_i$ 。然后用任意取的值 $\alpha_i \in \mathbb{C}$ 给 x_i 赋值, 使赋值后的多项式 $f(x_1, \alpha_1, \dots, \alpha_n)$ 是无平方因子的。矩阵 A_g 可以通过下面两步得到: 首先, 对于所有的 $1 \leq i, j \leq r$, 并且用单变元的近似除法 (见 §2.2.1 部分) 对 gg_i 和 $g_j f_{x_1}$ 用 f (用 $x_k = \alpha_k$ 对多项式赋值) 做剩余。然后解决一个最小二乘问题:

$$\min \|\text{rem}(gg_i - (a_{i,1}g_1 f_{x_1} + \dots + a_{i,r}g_r f_{x_1}), f)\|_2$$

计算出未知量 $a_{i,j}$ 的值。计算出矩阵 A_g 的特征值, 然后找出它们之间最小的距离:

$$\min_dist(g) = \min\{|\lambda_i - \lambda_j|, \quad 1 \leq i < j \leq r\}.$$

如果距离很小也就是说 A_g 数值上有重的特征值。这时需要重新选一组数 s_i , 继续找没有重特征值的 A_g 。实际上, 定理 2.3.3 已经说明了有很大的概率得到没有重特征值的 A_g 。因此算几次任意的 g , 然后用对应于最大的 $\min_dist(g)$ 的 g 进行计算可以得到较好的结果。在 Gao 的准确算法中, 用 A_g 在代数扩域上计算最大公因子来得到绝对不可约因子。因为我们的算法都是在复数域上计算的, 而且 A_g 所有的特征值都是复数。因此, 没有必要在代数扩域上计算, 而是直接在复数域上计算。根据 §2.2 部分的方法, 对 A_g 的每一个特征值 λ_i 计算多变元多项式的近似最大公因子 $\tilde{f}_i = \text{GCD}(f, g - \lambda_i f_{x_1})$, 也就得到了 f 在 $\mathbb{C}$ 上的近似因式分解: $f \approx \prod_{i=1}^r \tilde{f}_i$ 。

我们可以采取一些办法来优化所计算的多项式的近似不可约分解。首先, 我们计算出一个比例常数 c , 使得近似因式分解的向后误差达到最小:

$$\min_{c \in \mathbb{C}} \|f - c \prod_{i=1}^r \tilde{f}_i\|_2 / \|f\|_2.$$

因式分解的结果通过用 Gauss-Newton 迭代方法解决一个最小二乘问题来进一步地提高精度。相应的优化问题是找到一个由式子 $F(v_1, \dots, v_r) = f$ 构造的非线性系统的最小二乘解, 这里 $v_i \in \mathbb{C}[x_1, \dots, x_n]$, 而且

$$F(v_1, \dots, v_r) = \begin{bmatrix} C^{\text{tdeg}(v_2 \cdots v_r)} v_1 \dots C^{\text{tdeg}(v_r)} v_{r-1} v_r \end{bmatrix}$$

$C^{[k]}v$ 是在 §2.2.1 小节中描述的卷积矩阵。关于方法的细节, 请见文章[35]。

§2.3.3 算法

算法 2.3.1. AFMP: 多变元多项式的近似因式分解

输入: $f \in \mathbb{C}[x_1, \dots, x_n]$ 使得 f 和 f_{x_1} 是近似互素的, 而且在 $\mathbb{C}[x_2, \dots, x_n]$ 上没有近似因子 (见 §2.3.4 一节)。

输出: 多项式的近似因式分解和最优的比例常数 c 。

设 S 是复数上有限的一组数, 而且 $|S| \geq \text{tdeg}(f)^2$ 。

1. 计算近似零空间的基:

- (a) 构造矩阵 $\text{Rup}(f)$;
- (b) 计算 Ruppert 矩阵的奇异值分解, 找到最后 $\text{tdeg}(f) + 1$ 个奇异值 σ_i ;
- (c) 找到奇异值中的最大跳跃, 然后确定 G 的数值维数 r , 这里假设 $r \geq 2$;
- (d) 从矩阵 $\text{Rup}(f)$ 的最后 r 个右奇异向量构造 G 的一组基 $g_1, \dots, g_r$ 。

2. 计算 A_g 使得它的特征值相互之间的距离比较远:

- (a) 对变元 $x_i = \alpha_i$ 用 S 中任意的值进行赋值, 而且要求不能改变 f 的次数和无平方因子的性质;
- (b) k 从 1 到 p (在实际计算中, $p = 4$ 算得比较好) 做以下步骤:
 - i. 任意取 $s_{i,k} \in S$, 设 $\bar{g}_k = \sum_{i=1}^r s_{i,k} g_i$
 - ii. 计算 $a_{i,j,k}$ 使得下面的余式最小:

$$\min \| \text{rem}(\bar{g}_k g_i - \sum_{j=1}^r a_{i,j,k} g_j f_{x_1}, f) \|_2;$$

iii. 计算 $A_{\bar{g}_k}$ 的特征值 $\lambda_{i,k}, 1 \leq i \leq r$, 这里 $[A_{\bar{g}_k}]_{i,j} = [a_{i,j,k}]$. 设 $\min_dist_k = \min_{1 \leq i < j \leq r} \{|\lambda_{i,k} - \lambda_{j,k}|\}$;

(c) 设 $g = \bar{g}_k$, 这里 $\min_dist_k$ 是最大的。

3. 通过计算近似公因子得到因子:

对 $1 \leq i \leq r$, 我们在 $\mathbb{C}[x_1, \dots, x_n]$ 计算近似最大公因子: $GCD(f, g - \lambda_i f_{x_1})$ 。

4. 优化因式分解的结果:

(a) 应用 Gauss-Newton 迭代来优化近似因子;

(b) 计算 $\min_{c \in \mathbb{C}} \|f - c \prod_{i=1}^r f_i\|_2 / \|f\|_2$ 。

注 2.3.2. 关于在第三步中的近似最大公因子的计算, 有两个方面需要说明。第一, 为了求出多变元多项式的最大公因子的次数, 我们知道 AMVGCD 算法 (§2.2.2.1 小节) 中第一步给出的容许误差 ϵ , 如果用这个容许误差映射到单变元多项式的方法来确定多变元多项式的最大公因子的次数, 这种方法是最快的。由于映射后的单变元多项式有很大的范数, 所以用 σ_r 做为容许误差来计算最大公因子并不准确。第二, AFMP 算法并没有用 Gauss-Newton 方法来优化计算得到的近似最大公因子。而只是用在因式分解算法中。因为在做实验时发现: 优化近似最大公因子, 然后再优化因式分解, 这样所得的结果跟只对因式分解优化计算的结果一样。

注 2.3.3. AFMP 算法并不能保证得到的因子是不可约的。例如, 输入的多项式离可以分解的多项式并不是很近, 这样用近似最大公因子的算法得到的因子可能离可约的多项式很近。当然, 我们始终可以得到近似不可约的因子: 首先判断近似因子是不是近似不可约[34; 51], 如果近似可约, 可以用 AFMP 算法对近似因子再计算一次。

§2.3.4 重因子情形

f 离有重因子的多项式很近时, AFMP 算法无法得到好的结果。这与准确算法无法计算有重因子的情形有关。在这种情形时, 矩阵 $\text{Rup}(f)$ 的亏秩很大而且与因子的个数无关 (至少与无平方因子情形不一样)。

我们可以利用近似最大公因子算法计算 f 和 f_x 的近似最大公因子, 然后计算 f 和最大公因子的近似商: f_{sqfr} ; 用我们的算法计算出 $f_{\text{sqfr}} \approx f_1 \cdots f_r$ 的近似不可约分解。最后, 通过计算 $\alpha_{i,j} = \sigma_1(S_1(f_i, \partial_{x,j} f))$ 之间的跳跃来确定每个因子的幂次。

如果离它最近的多项式的因子是无平方因子的, 则 f 认为是近似无平方因子的。虽然我们不能计算最近可约多项式的因子, 但是可以用 $\text{Rup}(f)$ 的奇异值[34]给出离最近可约多项式的界。同样, $S_1(f, f_{x_1})$ 的奇异值给出了离最近的有重因子的多

项式的距离界。如果这两个界很接近时,需要用这两种方式(有平方因子方法情形和无平方因子情形)进行因式分解,然后用向后误差小的那一个。

在[74]文章中,他们给出了基于多变元多项式的近似最大公因子的算法。在实验中,这两种方法的效果差不多(在下面的表中比较了文章[76]中的例子)。

§2.3.5 近似因式分解的例子

例 2.3.1. [52] 我们用 AFMP 算法来分解这个例子:

$$f := (x^2 + yx + 2y - 1)(x^3 + y^2x - y + 7) + 0.2x.$$

因为 $\deg f = (4, 4, 4)$, Ruppert 矩阵的维数是 37×26 。矩阵的最后几个奇异值是:

$$\dots, 0.0475177, 0.0180165, 0.00168900, 0.203908 \times 10^{-10}$$

从第二个最小奇异值开始,最大的跳跃是 $0.0180165/0.00168900 = 10.6669$ 。因此, $r = 2$, f 可以认为有两个不可约因子。从最后两个右奇异向量得到 G 的基:

$$\begin{aligned} g_1 &= -0.222225x^4 - 0.249186yx^3 - 0.00669638x^3 - 0.0402726y^2x^2 - 0.541602yx^2 \\ &\quad + 0.265256x^2 - 0.0670575y^3x - 0.00117838y^2x - 0.0515876yx + 0.341504x \\ &\quad - 0.179952y^3 + 0.0653902y^2 + 0.157991y + 0.0120672, \\ g_2 &= 0.186723x^4 + 0.149379yx^3 - 0.146533 \cdot 10^{-9}x^3 + 0.112034y^2x^2 + 0.224068yx^2 \\ &\quad - 0.112034x^2 + 0.0746893y^3x - 0.0746893yx + 0.522825x \\ &\quad + 0.0746893y^3 - 0.0746893y^2 + 0.261412y + 0.00746893. \end{aligned}$$

任意取一个线性组合: $g = \frac{49}{50}g_1 + \frac{29}{40}g_2$, 设 $y = 1$ 。计算矩阵 $A = [a_{i,j}]$

$$\begin{bmatrix} -0.757129 & 1.166496 \\ 0.726481 & 0.537448 \end{bmatrix}.$$

和 A 的两个特征值: $\lambda_1 = -1.23519$, $\lambda_2 = 1.01551$ 。然后计算 $f_i = \text{GCD}(f, g - \lambda_i f_x)$, $i = 1, 2$, 得到 f 的两个因子:

$$\begin{aligned} f_1 &= -0.227274x^3 + 0.000400937yx^2 + 0.224906y - 1.57196 - 0.226899y^2x \\ &\quad - 0.00914627yx + 0.000416589y^3 + 0.0100343x^2 + 0.00747927x + 0.000416589y^2 \\ f_2 &= -0.442980x^2 - 0.0121738x - 0.447473yx - 0.887806y - 0.00098981y^2 + 0.444049. \end{aligned}$$

通过计算得到它的向后误差是: $\|f - 10.03713f_1f_2\|_2/\|f\|_2 = 0.00950792$ 。应用 Gauss-Newton 方法进行优化后得到的因子为:

$$\begin{aligned}
f_1 &= -0.718794x^3 + 0.00621037yx^2 + 0.702111y - 4.979600 - 0.713639y^2x \\
&\quad + 0.00376984yx + 0.000296705y^3 + 0.0146338x^2 - 0.00958225x + 0.00261478y^2 \\
f_2 &= -1.401574x^2 - 0.0424485x - 1.405118yx - 2.814096y + 0.00521123y^2 + 1.405785.
\end{aligned}$$

优化后的向后误差是: $\|f - f_1 f_2\|_2 / \|f\|_2 = 0.0010246453$.

例 2.3.2. AFMP 算法分解一个三个变元的多项式(文章[33]的例子):

$$\begin{aligned}
f := & 81x^4 + 72x^2y^2 + 0.002x^2z^2 - 648x^2 + 16y^4 + 0.001y^2z^2 \\
& - 288y^2 + 1296 - 648.003z^4 - 0.007z^2.
\end{aligned}$$

上面的多项式是下面的多项式的乘积:

$$(9x^2 + 4y^2 + 25.45596z^2 - 36)(9x^2 + 4y^2 - 25.45578z^2 - 36),$$

因为 $\deg f = (4, 4, 4)$, 矩阵 $\text{Rup}(f)$ 的维数是: 168×60 . 矩阵的最后几个奇异值是:

$$\dots, 198.661, 145.253, 0.868 \times 10^{-10}, 0.431 \times 10^{-12}.$$

从第二个最小奇异值开始, 最大的跳跃是:

$$145.253 / (0.868 \times 10^{-10}) = 0.167 \times 10^{13}.$$

因此, 可以得到 $r = 2$, 也就是说, f 是离有两个不可约因子的多项式很近. 从最后两个右奇异向量计算出 G 的基:

$$\begin{aligned}
g_1 &= 0.000151524784x^3 - 0.000606279x + 0.000067324xy^2 + 0.233157269xz^2, \\
g_2 &= 0.108724346x^3 - 0.434897383x + 0.048321932xy^2 - 0.000322604xz^2.
\end{aligned}$$

任意取一个线性组合 $g = g_1 + g_2$, 设 $y = 2$ 和 $z = -1$. 然后计算矩阵 $A = [a_{i,j}]$:

$$\begin{bmatrix} 0.000336771 & 0.000192632 \\ 0.000335102 & 0.000335302 \end{bmatrix}$$

和 A 的两个特征值: $\lambda_1 = 0.000590107$, $\lambda_2 = 0.000081966$. 计算 $f_i = \text{GCD}(f, g - \lambda_i f_x)$, $i = 1, 2$, 得到 f 的两个因子:

$$\begin{aligned}
f_1 &= 406.598x^2 + 180.710y^2 - 1150.03z^2 - 1626.39, \\
f_2 &= 406.596x^2 + 180.709y^2 + 1150.04z^2 - 1626.39.
\end{aligned}$$

然后找到最优的比例常数 $c = 0.00048996$ 使得向后误差 $\|f - c f_1 f_2\|_2$ 很小, 得到的因子为:

$$\sqrt{c} f_1 = 9.000015518 x^2 + 4.000009094 y^2 - 25.45583592 z^2 - 36.00004257,$$

$$\sqrt{c} f_2 = 8.999984448 x^2 + 3.999990906 y^2 + 25.45597017 z^2 - 35.99995743,$$

它的向后误差为: $\|f - c f_1 f_2\|_2 / \|f\|_2 = 4.67 \times 10^{-13}$ 。同样应用 Gauss-Newton 迭代, 找到优化后的因式分解:

$$f_1 = 9.000015552 x^2 + 4.000009094 y^2 - 25.455835924 z^2 - 36.000042565,$$

$$f_2 = 8.999984448 x^2 + 3.999990906 y^2 + 25.455970172 z^2 - 35.99995743,$$

它的向后误差是: $\|f - f_1 f_2\|_2 / \|f\|_2 = 3.23 \times 10^{-14}$ 。

§2.3.6 实验

我们用 Maple 实现了 AFMP 算法。在表 2.1 中, 用 Maple 程序计算了文章[66; 52; 9; 6]中的例子和随机生成的例子。计算例子的机器配置是 Pentium 4 2.0 GHz, 而且用的是 Maple 10 和 $Digits = 14$ 。

这里 *coeff. error* 表示输入的扰动, 也就是表示给可以分解的多项式加了 2-范数的相对的系数误差。backward error 和 *bkwd. err. w/ iter.* 这两个都是相对误差, 也就是 $\|f - \prod_i \tilde{f}_i\|_2 / \|f\|_2$ 。time(sec) 表示计算所需要的时间; 由于计算给定的多项式近似因式分解过程中, 算法中有一些步骤随机选了变量, 所以表中的时间有所变化; 一般情况下, 花在 Gauss-Newton 迭代的时间不到总时间的 10%; *iters* 这一列表示的是 Gauss-Newton 迭代的次数。*impr.* 这一列表示的是: 用优化所能提高结果的倍数。

从表 2.1 中的实验可以看到用 Gauss-Newton 优化能把结果提高将近一个阶。例如, 在例 9 中, 没有优化的向后误差达到 $2.37e-1$, 用优化后我们能得到一个更小的向后误差: $1.06e-1$ 。有些时候当 $\|noise\| \approx 10^{-1}$ 的情形, 尽管在计算过程中运用 Gauss-Newton 迭代方法优化近似因子, 但是还是很难得到理想的结果。

我们可以计算每个因式分解的向前误差。通过它, 就知道计算的结果和没有加扰动的原始多项式之间的 Euclidean 距离。在所做的试验中, 除了例 9, 向后误差和向前误差都是一个阶的 (因为例 9 算出来的因子的次数分别为 4 和 5, 然而原始多项式的因子的次数分别为 3, 3, 3)。

在表 2.1 中需要说明几点:

- 例 1 是文章[52] 中的例子, 他们算出的向后误差是 0.000753084。尽管他们算出的向后误差比我们的稍微小了一点, 但是他们的算法没有推广到可以解决一般的近似因式分解问题, 另外, 他们所花的时间也没有指出;

表 2.1 AFMP 算法的结果

<i>Ex.</i>	$\deg(f_i)$	<i>coeff. error</i>	<i>backward error</i>	<i>bkwd err. w/ iter</i>	<i>time(sec)</i>	<i>iters</i>	<i>impr.</i>
1	2,3	10^{-2}	1.08e-2	1.02e-3	7.764	7	10.6×
2	5,5	10^{-13}	1.07e-12	1.18e-13	6.813	2	9.0×
3	10,10	10^{-7}	9.95e-7	2.87e-7	157.09	3	3.4×
4	7,8	10^{-9}	1.94e-8	2.38e-9	50.222	16	8.2×
5	3,3,3	0	1.24e-13	6.44e-14	19.517	1	2.4×
6	6,6,10	10^{-5}	1.47e-4	7.24e-6	1329.4	4	20.3×
7	9,7	10^{-4}	2.18e-4	7.07e-5	74.157	4	3.1×
8	4,4,4,4,4	10^{-5}	3.34e-3	8.56e-6	5345.5	4	390.6×
9	3,3,3	10^{-1}	8.03e-1	1.06e-1	33.062	16	7.6×
10	12,7,5	10^{-5}	3.16e-4	8.02e-6	1766.7	4	39.4×
11	12,7,5	10^{-5}	7.77e-5	7.66e-6	2737.6	4	10.2×
12	12,7,5	10^{-3}	5.82e-3	7.66e-4	4288.7	6	7.6×
13	$5,(5)^2$	10^{-5}	6.84e-5	6.52e-6	46.751	3	10.5×
14	$(5)^3,3,(2)^4$	10^{-10}	2.60e-8	3.93e-9	136.39	2	6.6×
15	5,5	10^{-5}	1.55e-5	7.91e-6	559.30	3	2.0×
15a	2,2	10^{-5}	4.62e-13	3.23e-14	2.871	2	14.4×
15b	2,3	10^{-2}	7.44e-4	3.78e-4	6.687	4	2.0×
16	18,18	10^{-6}	4.50e-6	6.65e-7	5945.9	3	6.8×
17	18,18	10^{-6}	4.03e-6	6.61e-7	10348.	3	6.1×
18	6,6	10^{-7}	2.97e-7	5.10e-8	31.829	2	3.8×

- 例 2 和例 3 都是文章[66]的例子; Sasaki 的算法在机器 SPARC 5 (CPU: microSPARC II, 70 MHz) 上花了 430ms 和 2080ms, 他们计算的向后误差分别为 10^{-9} 和 10^{-5} ;
- 例 4 是文章[9]中的例子; 他们计算的向后误差是 0.47×10^{-4} , 而我们算出的向后误差是: 2.38×10^{-9} (时间在文章中并没有提);
- 例 5 是文章[6]的例子, 这个例子是次数为 9 的准确因式分解。两个算法计算出同样的向后误差, 时间在文章[6]中没有提到;
- 例 6 到例 13 和例 15 到例 17 的因子都是系数为 $-5 \leq c \leq 5$ 范围内的整数, 然后给因子相乘后的多项式加扰动: 我们给的容许扰动是 10^{-e} , 然后任意取次

数与因子乘积的次数相同的扰动多项式，扰动多项式的项数为因子乘积的项数的25%（例 10 为 5%，例 17 为 99%），系数在 $[-10^e, 10^e]$ 范围内；最后，给扰动乘以一个比例常数使得多项式的相对误差是 10^{-e} ；

- 例 10, 例 11 和例 12 是相同的多项式但是加了不同的扰动；
- 例 14 是在文章[74]中的 ASFF 例子；它有非平凡的浓度和重因子。计算得到的向前误差是 10^{-8} ，与 Zeng 的向前误差很接近。
- 例 15, 15a, 15b 是有三个变元的例子；15a 是文章[33]中的例子，关于这个例子的细节见 §2.3.5 部分；例 15b 是文章[28]中的例子，文章中指出：用他们的算法和优化过后得到的因式分解的向后误差是： $5.72e-4$ ；
- 例 18 是一个复系数的多项式，因子的实部和虚部的系数都是在 $[-5, 5]$ 范围内的整数，扰动是加了实部和虚部的所有项。

另外，我们算了 Jan Verschelde 教授给的工程上关于 Stewart-Gough 平台的四个例子[69]。输入的多项式有两个或者三个变元，次数为 12，而且多项式的系数误差特别小（ 10^{-16} ），近似因子的重数为 1, 3, 5。多项式的向前误差很接近机器误差，因此采用了稀疏数值插值方法 [21] 来计算近似因子。我们所用的时间不超过 200秒，比他们的解法要快得多。而且向后误差不大于 $7.62 \cdot 10^{-9}$ 。当然，这里我们还用了稀疏插值程序 [21]来计算。

第三章 结构低秩逼近（广义）Sylvester 矩阵

§3.1 前言

这一章要解决下面这个问题：

问题 3.1.1. 输入两个多项式 $f, g \in \mathbb{C}[y_1, y_2, \dots, y_r]$ ，全次数分别为： $\text{tdeg}(f) = m$ 和 $\text{tdeg}(g) = n$ 。给定一个正整数 k 且 $k \leq \min(m, n)$ ，计算 $\Delta f, \Delta g \in \mathbb{C}[y_1, y_2, \dots, y_r]$ 使得 $\text{tdeg}(\Delta f) \leq m$ ， $\text{tdeg}(\Delta g) \leq n$ ， $\text{tdeg}(\text{GCD}(f + \Delta f, g + \Delta g)) \geq k$ 而且 $\|\Delta f\|_2^2 + \|\Delta g\|_2^2$ 是最小的。

文章[67; 54; 7; 13; 27; 3; 56; 65] 解决了 ϵ -近似最大公因子问题。先假设一个误差估计 ϵ (输入的量)，输出为两个多项式 $\Delta f, \Delta g$ 使得 $\|\Delta f\|_2^2 + \|\Delta g\|_2^2 \leq \epsilon$ ，然后得到 $\text{GCD}(f + \Delta f, g + \Delta g)$ 。

在这章中，我们分单变元和多变元两种情况考虑。问题 3.1.1 可以转化为结构最小二乘问题：保持（广义）Sylvester 结构的低秩逼近。这里单变元多项式对应的是 Sylvester 矩阵，多变元多项式对应的是广义 Sylvester 矩阵。求解这个最小二乘问题有很多方法，这里采用了结构最小二乘范数 (STLN) 算法[57]，也就是计算超定线性系统 $AX \approx B$ 的近似解 $(A + E)X = B + H$ ，且最小扰动 $[E H]$ 保持结构。通过实验证明，STLN 的方法很有效。这一章我们将要阐述怎样把 STLN（范数为 2）方法应用到（广义）Sylvester 子矩阵来解决上面的问题。对于单变元的情形，我们还用迭代-映射算法 [59] 解决了上面的问题。但是跟 STLN 方法相比，它的迭代次数比较多，且向后误差比较大。

文章还给出了一些用 STLN 方法计算的例子。从实验中可以看出 STLN 方法只需要几次迭代而且向后误差比文章[76]中的基于 SVD 和 Gauss-Newton 方法算法的向后误差小很多。

§3.2 结构低秩逼近 Sylvester 矩阵

§3.2.1 预备知识

首先证明了最小问题 3.1.1 解的存在性。设 $f, g \in \mathbb{C}[x] \setminus \{0\}$ ， $\deg(f) = m$ 和 $\deg(g) = n$ 。也就是

$$\begin{aligned} f &= a_m x^m + a_{m-1} x^{m-1} + \dots + a_1 x + a_0, \quad a_m \neq 0, \\ g &= b_n x^n + b_{n-1} x^{n-1} + \dots + b_1 x + b_0, \quad b_n \neq 0. \end{aligned}$$

定理 3.2.1. 设 k 是一个正整数且 $k \leq \min(m, n)$ 。则存在 $\hat{f}, \hat{g} \in \mathbb{C}[x]$, $\deg(\hat{f}) \leq m$, $\deg(\hat{g}) \leq n$ 且 $\deg \text{GCD}(\hat{f}, \hat{g}) \geq k$ 使得对所有 $\tilde{f}, \tilde{g} \in \mathbb{C}[x]$, $\deg(\tilde{f}) \leq m$, $\deg(\tilde{g}) \leq n$ 且 $\deg \text{GCD}(\tilde{f}, \tilde{g}) \geq k$, 我们有:

$$\|\hat{f} - f\|_2^2 + \|\hat{g} - g\|_2^2 \leq \|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2$$

证明. 见文章[36]. □

注 3.2.1. 当限制输入的多项式和扰动多项式的系数是实数时, 定理 3.2.1 仍然成立。需要注意的是: 对于输入实的多项式, 最优的复解可能比最优实解要好一些。例如, 给两个多项式 $f = x^2 + 1$ 和 $g = x^2 + 2$ 且设 $k = 1$, 则最优的实解为: $\hat{f} = 0.723598x^2 + 1.170810$, $\hat{g} = 1.170822x^2 + 1.894436$ 。我们发现 $\deg(\text{GCD}(\hat{f}, \hat{g})) = 2 > k$, 并且 $\|\hat{f} - f\|_2^2 + \|\hat{g} - g\|_2^2 = 0.145898$ 。然而, 存在更近的一对复的多项式有公共根:

$$\begin{aligned} \hat{\hat{f}} &= (0.81227 - 0.0000016019i)x^2 \\ &\quad + (0.00000038156 - 0.14812i)x \\ &\quad + 1.1169 - 0.00000039519i \end{aligned}$$

和

$$\begin{aligned} \hat{\hat{g}} &= (1.1220 - 0.00000076026i)x^2 \\ &\quad + (0.0000011810 + 0.096265i)x \\ &\quad + 1.9240 + 0.0000013787i \end{aligned}$$

$\|\hat{\hat{f}} - f\|_2^2 + \|\hat{\hat{g}} - g\|_2^2 = 0.1007615$ 。而且, 最近的多项式不是唯一的。第二组解是:

$$\begin{aligned} \hat{\hat{f}} &= (0.81227 - 0.0000010696i)x^2 \\ &\quad + (0.0000018180 + 0.14812i)x \\ &\quad + 1.1169 - 0.00000035348i \end{aligned}$$

和

$$\begin{aligned} \hat{\hat{g}} &= (1.1220 + 0.00000034875i)x^2 \\ &\quad + (0.0000010267 - 0.096263i)x \\ &\quad + 1.9240 + 0.00000055845i. \end{aligned}$$

□

设 $S(f, g)$ 是 f 和 g 的 Sylvester 矩阵。则 f 和 g 的最大公因子的次数是与 S 的亏秩是相等的。所以, 有下面的等价式:

$$\min_{\deg(\text{GCD}(\tilde{f}, \tilde{g})) \geq k} \|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2 \iff \min_{\dim \text{Nullspace}(\tilde{S}) \geq k} \|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2, \quad (3.1)$$

这里 $\tilde{S}$ 是由 $\tilde{f}$ 和 $\tilde{g}$ 产生的 Sylvester 矩阵, $\deg \tilde{f} = \tilde{m} \leq m$, $\deg \tilde{g} = \tilde{n} \leq n$ 。需要注意的是: $\tilde{f} = 0$ 或者 $\tilde{g} = 0$, Sylvester 矩阵 $\tilde{S}$ 没有定义。如果一个多项式是零, 需要假设 $\text{Rank}(\tilde{S}) \leq \tilde{n} + \tilde{m} - k$ (因为 GCD 是其它非零的多项式)。如果 $\tilde{f} = \tilde{g} = 0$, 则设 $\text{Rank}(\tilde{S}) = \infty$ 。

第 k 个 Sylvester 矩阵的子矩阵 $S_k \in \mathbb{C}^{(m+n-k+1) \times (m+n-2k+2)}$ 是由矩阵 S 删掉最后 $k-1$ 行和 f, g 各自系数的最后 $k-1$ 列后得到的子矩阵。为了方便起见, Sylvester 矩阵的第 k 个子矩阵记为 k -th Sylvester 矩阵。

$$S_k = \begin{bmatrix} a_m & & & b_n & & \\ a_{m-1} & \ddots & & b_{n-1} & \ddots & \\ \vdots & \ddots & a_m & \vdots & \ddots & b_n \\ a_0 & & a_{m-1} & b_0 & & b_{n-1} \\ & \ddots & \vdots & & \ddots & \vdots \\ & & a_0 & & & b_0 \end{bmatrix},$$

$\underbrace{\hspace{10em}}_{n-k+1} \quad \underbrace{\hspace{10em}}_{m-k+1}$

对于 $k=1$ 的情况, $S_1 = S$ 是 Sylvester 矩阵。文章[18]说明了 S 和它的 k -th Sylvester 矩阵 S_k 之间有一个很强的关系。下面给出一个定理[18]:

定理 3.2.2. 给定两个单变元多项式 $f, g \in \mathbb{C}[x]$, $\deg(f) = m$, $\deg(g) = n$ 和 $1 \leq k \leq \min(m, n)$ 。 $S(f, g)$ 是 f 和 g 的 Sylvester 矩阵, S_k 是 f 和 g 的 k -th Sylvester 矩阵。则下面的两个结论是等价的:

(a) $\text{Rank}(S) \leq m + n - k$

(b) S_k 的亏秩大于等于 1

有了上面的定理, 式子(3.1) 可以变化为:

$$\min_{\deg(\text{GCD}(\tilde{f}, \tilde{g})) \geq k} \|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2 \iff \min_{\dim \text{Nullspace}(\tilde{S}_k) \geq 1} \|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2, \quad (3.2)$$

这里 $\tilde{S}_k$ 是由 $\tilde{f}$ 和 $\tilde{g}$ 的 k -th Sylvester 矩阵, $\tilde{f}$ 和 $\tilde{g}$ 的次数满足: $\deg \tilde{f} \leq m$ 和 $\deg \tilde{g} \leq n$ 。

如果用 STLN [61; 60; 57; 43]方法解下面的超定系统:

$$A_k \mathbf{x} \approx \mathbf{b}_k, \quad (3.3)$$

这里 $S_k = [\mathbf{b}_k \ A_k]$, 也就是说, $\mathbf{b}_k$ 是 S_k 的第一列而 A_k 是 S_k 的剩余列。然后我们得到具有 k -th Sylvester 结构的最小的扰动矩阵 $[\mathbf{h}_k \ E_k]$ 使得

$$\mathbf{b}_k + \mathbf{h}_k \in \text{Range}(A_k + E_k).$$

因此, $\tilde{S}_k = [\mathbf{b}_k + \mathbf{h}_k, A_k + E_k]$ 是保持 k -th Sylvester 结构的解 (这里要求多项式的最高项的系数为非零的), 且满足 $\dim \text{Nullspace}(\tilde{S}_k) \geq 1$ 。

下面的例子和定理说明了我们取第一列构造超定系统 (3.14) 的原因。

例 3.2.1. 给定两个多项式:

$$\begin{aligned} f &= x^2 + x = x(x+1), \\ g &= x^2 + 4x + 3 = (x+3)(x+1), \end{aligned}$$

S 是 f 和 g 的 Sylvester 矩阵,

$$S = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 4 & 1 \\ 0 & 1 & 3 & 4 \\ 0 & 0 & 0 & 3 \end{bmatrix}$$

S 的亏秩是 1。按两种方式来构造超定系统:

$$S = [\hat{A}_1 \ \hat{\mathbf{b}}_1] = [\bar{\mathbf{b}}_1 \ \bar{A}_1],$$

这里 $\hat{\mathbf{b}}_1$ 是 S 的最后一列, 而 $\bar{\mathbf{b}}_1$ 是 S 的第一列。

超定系统

$$\hat{A}_1 \mathbf{x} = \hat{\mathbf{b}}_1$$

没有解, 而超定系统

$$\bar{A}_1 \mathbf{x} = \bar{\mathbf{b}}_1$$

有准确的解: $\mathbf{x} = [-3, 1, 0]^T$ 。

定理 3.2.3. 给定单变元的多项式 $f, g \in \mathbb{C}[x]$, $\deg(f) = m, \deg(g) = n$ 和一个正整数 $k \leq \min(m, n)$ 。假设 S_k 是 f 和 g 的 k -th Sylvester 矩阵。假设 $S_k = [\mathbf{b}_k \ A_k]$, 这里 $\mathbf{b}_k$ 是 S_k 的第一列, 而 A_k 是 S_k 的最后 $n + m - 2k + 1$ 列。则有:

$$\dim \text{Nullspace}(S_k) \geq 1 \iff A_k \mathbf{x} = \mathbf{b}_k \text{ 有解.} \quad (3.4)$$

证明. “ $\Leftarrow$ ” : 设 $A_k \mathbf{x} = \mathbf{b}_k$ 有解, 那么 $\mathbf{b}_k \in \text{Range}(A_k)$ 。因为 $\mathbf{b}_k$ 是 S_k 的第一列, 则 $S_k = [\mathbf{b}_k \ A_k]$ 的亏秩至少为 1。

“ $\Rightarrow$ ” : 假设 $S_k = [\mathbf{b}_k \ A_k]$ 的亏秩至少为 1。把向量 $[x^{n+m-k}, \dots, x, 1]$ 乘到方程 $A_k \mathbf{x} = \mathbf{b}_k$ 的两边, 可以得到下面的式子:

$$[x^{n-k-1}f, \dots, f, x^{m-k}g, x^{m-k-1}g, \dots, g]\mathbf{x} = x^{n-k}f. \quad (3.5)$$

(3.5) 的解 $\mathbf{x}$ 对应到多项式 u, v 的系数, 次数满足 $\deg(u) \leq n - k - 1, \deg(v) \leq m - k$, 而且还满足:

$$x^{n-k}f = u f + v g.$$

设 $d = \text{GCD}(f, g)$, $f_1 = f/d, g_1 = g/d$ 。因为 $\dim \text{Nullspace}(S_k) \geq 1$, 且有 $\deg(d) \geq k$ 和 $\deg(f_1) \leq m - k, \deg(g_1) \leq n - k$ 。用 g_1 除 x^{n-k} , 得到商 q 和余式 p 使得

$$x^{n-k} = q g_1 + p,$$

这里 $\deg(q) \leq \deg(d) - k, \deg(p) \leq n - k - 1$ 。接着, 我们来检验

$$u = p, v = q f_1,$$

是 (3.5) 的解, 因为 $\deg(u) \leq n - k - 1$,

$$\deg(v) = \deg(q) + \deg(f_1) \leq \deg(d) - k + \deg(f_1) \leq m - k,$$

而且

$$v g + u f = f_1 q d g_1 + p f = f q g_1 + f p = f x^{n-k}. \quad \square$$

接下来需要说明给定任意一个 Sylvester 矩阵, 如果它的元素可以任意扰动, 那么总有可能找到具有 k -th Sylvester 结构 (这里要求首项非零) 的矩阵 $[\mathbf{h}_k \ E_k]$, 使得 $\mathbf{b}_k + \mathbf{h}_k \in \text{Range}(A_k + E_k)$, 这里 $\mathbf{b}_k$ 是 S_k 的第一列, A_k 是 S_k 的剩余列。

定理 3.2.4. 给定正整数 m, n 和 k , $k \leq \min(m, n)$, 则存在 Sylvester 矩阵 $S \in \mathbb{C}^{(m+n) \times (m+n)}$, 且矩阵 S 的秩为 $m + n - k$ 。

证明. 对于任意的 m 和 n , 我们始终可以构造多项式 $f, g \in \mathbb{C}[x]$ 使得 $\deg(f) = m$, $\deg(g) = n$, $\text{GCD}(f, g)$ 的次数为 k 。由 f, g 生成的 Sylvester 矩阵 S 。 $\square$

推论 3.2.1. 给定正整数 m, n , $k \leq \min(m, n)$, 和 k -th Sylvester 矩阵 $S_k = [\mathbf{b}_k A_k]$, 这里 $A_k \in \mathbb{C}^{(m+n-k+1) \times (m+n-2k+1)}$, $\mathbf{b}_k \in \mathbb{C}^{(m+n-k+1) \times 1}$, 我们总是可以找到有 k -th Sylvester 结构的扰动矩阵 $[\mathbf{h}_k E_k]$ 使得 $\mathbf{b}_k + \mathbf{h}_k \in \text{Range}(A_k + E_k)$ 。

§3.2.2 STLN 解决有 Sylvester 结构的超定系统

这部分说明怎样来求解超定系统

$$A_k \mathbf{x} \approx \mathbf{b}_k, \quad (3.6)$$

这里 $A_k \in \mathbb{C}^{(m+n-k+1) \times (m+n-2k+1)}$, $\mathbf{b}_k \in \mathbb{C}^{(m+n-k+1) \times 1}$, $S_k = [\mathbf{b}_k A_k]$ 是 k -th Sylvester 矩阵。根据定理 3.2.4 和推论 3.2.1, 始终存在 k -th Sylvester 结构的扰动 $[\mathbf{h}_k E_k]$ 使得 $(\mathbf{b}_k + \mathbf{h}_k) \in \text{Range}(A_k + E_k)$ 。下面我们详细说明怎样用 STLN 找到最小的解。

首先, S_k 的保持 k -th Sylvester 结构的扰动矩阵

$$[\mathbf{h}_k E_k] = \begin{bmatrix} z_1 & & & z_{m+2} & & \\ z_2 & \ddots & & z_{m+3} & \ddots & \\ \vdots & \ddots & & \vdots & \ddots & z_{m+2} \\ z_{m+1} & & z_2 & z_{m+n+2} & & z_{m+3} \\ & \ddots & \vdots & & \ddots & \vdots \\ & & z_{m+1} & & & z_{m+n+2} \end{bmatrix}$$

$\underbrace{\hspace{10em}}_{n-k+1} \quad \underbrace{\hspace{10em}}_{m-k+1}$

可以用一个向量 $\mathbf{z} \in \mathbb{C}^{(m+n+2) \times 1}$ 表示出来。这个向量设为:

$$\mathbf{z} = [z_1, z_2, \dots, z_{m+n+1}, z_{m+n+2}]^T.$$

因为 $\mathbf{h}_k$ 是上面矩阵的第一列, 所以可以定义一个矩阵 P_k 为:

$$P_k = \begin{bmatrix} \mathbf{I}_{m+1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{C}^{(m+n-k+1) \times (m+n+2)}, \quad (3.7)$$

这里 $\mathbf{I}_{m+1}$ 是一个 $(m+1) \times (m+1)$ 是单位矩阵, 而且使得 $\mathbf{h}_k = P_k \mathbf{z}$ 。

下面要求解一个有等式限制的最小二乘问题:

$$\min_{\mathbf{z}, \mathbf{x}} \|\mathbf{z}\|_2, \quad \mathbf{r} = \mathbf{0}, \quad (3.8)$$

这里结构剩余 $\mathbf{r}$ 是:

$$\mathbf{r} = \mathbf{r}(\mathbf{z}, \mathbf{x}) = \mathbf{b}_k + \mathbf{h}_k - (A_k + E_k)\mathbf{x}.$$

结构的最小问题(3.8)可以用罚值的方法 [1]来求解, 那么 (3.8) 可以转化为:

$$\min_{\mathbf{z}, \mathbf{x}} \left\| \begin{array}{c} w\mathbf{r}(\mathbf{z}, \mathbf{x}) \\ \mathbf{z} \end{array} \right\|_2, \quad w \gg 1, \quad (3.9)$$

这里 w 是介于 10^8 和 10^{10} 之间的一个很大的罚值。在文章 [1; 72] 中已经说明: 尽管有很大的罚值, 基于 Givens 旋转的算法能得到精确的结果。

按照 [60; 57], 我们对 $\mathbf{r}(\mathbf{z}, \mathbf{x})$ 用线性逼近来求解最小问题。设 $\Delta\mathbf{z}$ 和 $\Delta\mathbf{x}$ 分别表示 $\mathbf{z}$ 和 $\mathbf{x}$ 的很小变化, 而 ΔE 表示相对应于 E_k 的变化。则对 $\mathbf{r}(\mathbf{z} + \Delta\mathbf{z}, \mathbf{x} + \Delta\mathbf{x})$ 的一阶逼近为:

$$\begin{aligned} \mathbf{r}(\mathbf{z} + \Delta\mathbf{z}, \mathbf{x} + \Delta\mathbf{x}) &= \mathbf{b}_k + P_k(\mathbf{z} + \Delta\mathbf{z}) - (A_k + E_k + \Delta E)(\mathbf{x} + \Delta\mathbf{x}) \\ &\approx \mathbf{b}_k + P_k\mathbf{z} - (A_k + E_k)\mathbf{x} + P_k\Delta\mathbf{z} - (A_k + E_k)\Delta\mathbf{x} - \Delta E\mathbf{x} \\ &= \mathbf{r} + P_k\Delta\mathbf{z} - (A_k + E_k)\Delta\mathbf{x} - \Delta E\mathbf{x}. \end{aligned}$$

引进一个 Sylvester 结构的矩阵 $Y_k \in \mathbb{C}^{\mu \times \nu}$, 这里 $\mu = m + n - k + 1$, $\nu = m + n + 2$, 使得

$$Y_k \mathbf{z} = E_k \mathbf{x} \quad \text{这里} \quad \mathbf{x} = [x_1, x_2, \dots, x_{m+n-2k+1}]^T. \quad (3.10)$$

有了上面的准备, 将 $\mathbf{z} = \Delta\mathbf{z}$ 和 $E_k = \Delta E$ 代入 (3.10), 那么(3.9) 可以由下面的式子来逼近:

$$\min_{\Delta\mathbf{x}, \Delta\mathbf{z}} \left\| \begin{bmatrix} w(Y_k - P_k) & w(A_k + E_k) \\ \mathbf{I}_{m+n+2} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{z} \\ \Delta\mathbf{x} \end{bmatrix} + \begin{bmatrix} -w\mathbf{r} \\ \mathbf{z} \end{bmatrix} \right\|_2. \quad (3.11)$$

根据 Sylvester 矩阵的性质, 我们给出一种新的方法来构造矩阵 Y_k 。对于 $f, g, E, \mathbf{z}$ 和 $\mathbf{x}$ 还是按照上面的方式假设。把一个向量

$$\mathbf{v} = [x^{m+n-k}, x^{m+n-k-1}, \dots, x^2, x^1, 1] \in \mathbb{C}[x]^{m+n-k+1}$$

乘到方程(3.10)的两边, 则得到多项式的等式:

$$\mathbf{v} Y_k \mathbf{z} = \mathbf{v} E_k \mathbf{x}.$$

对于向量 $\begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix}$, 可以写成:

$$\mathbf{v} E_k \mathbf{x} = \mathbf{v} [\mathbf{h}_k, E_k] \hat{\mathbf{x}} = \hat{g}_1 \hat{u}_1 + \hat{g}_2 \hat{u}_2, \quad (3.12)$$

这里 $\hat{g}_1$ 是次数为 m 的多项式, 且由 $\mathbf{z}$ 的子向量产生的:

$$[z_1, z_2, \dots, z_{m+1}],$$

同样 $\hat{g}_2$ 是次数为 n 的多项式, 也是由 $\mathbf{z}$ 的子向量产生的:

$$[z_{m+2}, z_{m+3}, \dots, z_{m+n+2}],$$

$\hat{u}_1$ 是次数为 $n-k-1$ 的多项式, 是由 $\hat{\mathbf{x}}$ 的子向量产生:

$$[0, x_1, x_2, \dots, x_{n-k}],$$

$\hat{u}_2$ 是次数为 $m-k$ 的多项式, 也是由 $\hat{\mathbf{x}}$ 的子向量产生:

$$[x_{n-k+1}, x_{n-k+2}, \dots, x_{m+n-2k+1}].$$

从 x 的幂次和变元 z_i 组成的上面的线性系统(3.12) 的系数矩阵就得到了矩阵 Y_k :

$$Y_k = \begin{bmatrix} 0 & & & x_{n+1-k} & & \\ x_1 & \ddots & & x_{n+2-k} & \ddots & \\ \vdots & \ddots & 0 & \vdots & \ddots & x_{n+1-k} \\ x_{n-k} & & x_1 & x_{m+n+1-2k} & & x_{n+2-k} \\ & \ddots & \vdots & & \ddots & \vdots \\ & & x_{n-k} & & & x_{m+n+1-2k} \end{bmatrix}.$$

$\underbrace{\hspace{10em}}_{m+1} \quad \underbrace{\hspace{10em}}_{n+1}$

例 3.2.2. 假设 $m = n = 3$, $k = 2$, 则 $S_2 = [\mathbf{b}_2 \ A_2]$, 这里

$$A_2 = \begin{bmatrix} 0 & b_3 & 0 \\ a_3 & b_2 & b_3 \\ a_2 & b_1 & b_2 \\ a_1 & b_0 & b_1 \\ a_0 & 0 & b_0 \end{bmatrix}, \quad \mathbf{b}_2 = \begin{bmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \\ 0 \end{bmatrix},$$

$$P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$Y_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & x_2 & 0 & 0 & 0 \\ x_1 & 0 & 0 & 0 & x_3 & x_2 & 0 & 0 \\ 0 & x_1 & 0 & 0 & 0 & x_3 & x_2 & 0 \\ 0 & 0 & x_1 & 0 & 0 & 0 & x_3 & x_2 \\ 0 & 0 & 0 & x_1 & 0 & 0 & 0 & x_3 \end{bmatrix}.$$

从上面的例子很容易看出(3.11)中的系数矩阵是 Toeplitz 块结构。因此, 我们可以应用快速最小二乘方法。关于这方面的细节, 见文章 [46]。

§3.2.3 单变元的近似最大公因子算法和实验

下面的算法是计算问题 3.1.1 的近似解:

算法 3.2.1. AppSylv-k

输入: 单变元多项式 $f, g \in \mathbb{C}[x]$, 次数分别为 $m \geq n$, S 是由 f, g 生成的 Sylvester 矩阵, 一个正整数 $1 \leq k \leq n$ 和容许误差 tol 。

输出: 多项式 $\tilde{f}$ 和 $\tilde{g}$ 使得 $\dim \text{Nullspace}(S(\tilde{f}, \tilde{g})) \geq k$ 且 Euclidean 距离 $\|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2$ 达到最小。

1. 构造 k -th Sylvester 矩阵 S_k , 设 $\mathbf{b}_k$ 为 S_k 的第一列, A_k 为 S_k 的剩余列。设 $E_k = \mathbf{0}$, $\mathbf{h}_k = \mathbf{0}$ 。
2. 从 $\min \|A_k \mathbf{x} - \mathbf{b}_k\|_2$ 和 $\mathbf{r} = \mathbf{b}_k - A_k \mathbf{x}$ 计算 $\mathbf{x}$ 。构造矩阵 P_k 和 Y_k 。
3. 迭代

$$(a) \min_{\Delta \mathbf{x}, \Delta \mathbf{z}} \left\| \begin{bmatrix} w(Y_k - P_k) & w(A_k + E_k) \\ \mathbf{I}_{m+n+2} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{z} \\ \Delta \mathbf{x} \end{bmatrix} + \begin{bmatrix} -w\mathbf{r} \\ \mathbf{z} \end{bmatrix} \right\|_2.$$

(b) 设 $\mathbf{x} = \mathbf{x} + \Delta \mathbf{x}$, $\mathbf{z} = \mathbf{z} + \Delta \mathbf{z}$.

(c) 根据 $\mathbf{z}$ 构造 E_k 和 $\mathbf{h}_k$, 从向量 $\mathbf{x}$ 构造 Y_k . 设 $A_k = A_k + E_k$, $\mathbf{b}_k = \mathbf{b}_k + \mathbf{h}_k$, $\mathbf{r} = \mathbf{b}_k - A_k \mathbf{x}$.

直到 $(\|\Delta \mathbf{x}\|_2 \leq \text{tol} \text{ 和 } \|\Delta \mathbf{z}\|_2 \leq \text{tol})$ 。

4. 根据 $\mathbf{b}_k$ 和 A_k , 输出 $\tilde{f}$ 和 $\tilde{g}$ 。

给定一个容许误差 ϵ , 算法 AppSylv-k 可以用来计算多项式 f 和多项式 g 的 ϵ -GCD, 这里设次数分别为 m 和 n , 且 $m \geq n$ 。我们用 AppSylv-k 算法从 $k = n \leq m$ 开始计算极小值 $\mathcal{N} = \|\tilde{f} - f\|_2 + \|\tilde{g} - g\|_2$ 且 $\text{Rank}(S(\tilde{f}, \tilde{g})) \leq m + n - k$ 。如果 $\mathcal{N} < \epsilon$, 可以从 $S_k(\tilde{f}, \tilde{g})$ [18; 74] 计算 ϵ -GCD; 否则, 逐渐减小 k 来重复应用算法 AppSylv-k。另外一个方法是根据对 Sylvester 矩阵 $S(f, g)$ 的奇异值分解来测试 ϵ -GCD 的次数, 找到 ϵ -GCD 的次数的上界, 关于细节见文章[7; 13]。因此, 算法不是从 $k = n$ 而是从 $k = r$ 作为最大公因子的最高次数来测试, 然后计算 ϵ -GCD。

例 3.2.3. 下面给出 Karmarkar 和 Lakshman 文章[40]中的一个例子, 给定多项式 f, g :

$$\begin{aligned} f &= x^2 - 6x + 5 = (x - 1)(x - 5), \\ g &= x^2 - 6.3x + 5.72 = (x - 1.1)(x - 5.2), \end{aligned}$$

想找到最小的多项式扰动 Δf 和 Δg , 使得多项式 $f + \Delta f$ 和 $g + \Delta g$ 有一个相同的根。按照两种情形考虑这个问题: 多项式的首项系数可以扰动和首项系数不可以扰动。

情形 1: 首项系数可以扰动

对多项式 f, g 应用算法 AppSylv-k, 设 $k = 1$ 和 $\text{tol} = 10^{-3}$ 。通过三次迭代, 找到多项式 $\tilde{f}$ 和 $\tilde{g}$;

$$\begin{aligned} \tilde{f} &= 0.9850x^2 - 6.0029x + 4.9994, \\ \tilde{g} &= 1.0150x^2 - 6.2971x + 5.7206, \end{aligned}$$

距离是:

$$\mathcal{N} = \|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2 = 0.0004663$$

$\tilde{f}$ 和 $\tilde{g}$ 的公共根是: 5.09890429。

情形 2: 首项系数不可以扰动

对多项式 f, g 应用算法 $AppSylv-k$, 同样设 $k = 1$ 和 $tol = 10^{-3}$ 。通过三次迭代, 找到多项式 $\tilde{f}$ 和 $\tilde{g}$;

$$\tilde{f} = x^2 - 6.0750x + 4.9853,$$

$$\tilde{g} = x^2 - 6.2222x + 5.7353,$$

最小距离是

$$\mathcal{N} = \|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2 = 0.01213604583$$

$\tilde{f}$ 和 $\tilde{g}$ 的公共根是: 5.0969478。

在文章 [40] 中, 我们限制扰动的多项式是首一的。Karmarkar 和 Lakshman 计算得到的最小扰动是 0.01213605293, 所对应的公共根是 5.096939087。

<i>Ex.</i>	<i>m, n</i>	<i>k</i>	<i>it.</i> (<i>Chu</i>)	<i>it.</i> (<i>STLN</i>)	<i>error</i> (<i>Zeng</i>)	<i>error</i> (<i>STLN</i>)	σ_k	$\tilde{\sigma}_k$
1	2, 2	1	4.73	1.90	1.89e-4	2.87e-5	3.53e-3	10 ⁻⁹
2	3, 3	2	8.49	1.98	1.36e-3	1.05e-4	8.21e-3	10 ⁻⁹
3	5, 4	3	11.44	2.00	1.00e-3	1.25e-4	1.01e-2	10 ⁻⁹
4	5, 5	3	13.64	2.00	7.43e-4	1.25e-4	9.57e-3	10 ⁻⁹
5	6, 6	4	23.07	2.00	1.46e-3	1.41e-4	9.64e-3	10 ⁻⁹
6	8, 7	4	32.64	2.00	6.53e-4	1.31e-4	8.04e-3	10 ⁻⁹
7	10, 10	5	43.12	2.00	1.61e-3	2.01e-4	1.21e-2	10 ⁻⁹
8	14, 13	7	58.16	2.00	1.23e-3	2.52e-4	1.51e-2	10 ⁻⁹
9	28, 28	10	161.74	2.00	2.61e-3	3.41e-4	1.48e-2	10 ⁻¹⁰
10	65, 65	15	633.64	2.00	6.19e-3	5.50e-4	1.90e-2	10 ⁻⁹

表 3.1 AppSylv-k 算法的结果 (单变元)

在表 3.1 中, 我们对一些用 Maple 10 随机生成的例子应用 AppSylv-k 算法计算 ϵ -GCD, 这里 $Digits = 14$ 。对于每个例子中的 (m, n) , 计算了 50 个随机生成的例子, 然后取它们的平均值。

对于每个例子, 两个多项式的互素部分和 GCD 的系数都是在 $-10 \leq c \leq 10$ 的任意整数, 然后给多项式加扰动。扰动是这样构造的: 取相对容许扰动为 10^{-e} , 然后任取一个扰动多项式的系数在 $-10 \leq c \leq 10$ 的范围内, 且次数与乘积多项式相

同, 最后, 给扰动多项式乘上一个常数比例, 使得相对误差为: 10^{-e} 。这里 m 和 n 分别表示多项式 f 和 g 的次数; k 是 f 和 g 的近似最大公因子的次数; $it. (Chu)$ 表示用 Chu 的方法[59]需要的迭代, 而 $it. (STLN)$ 表示用 AppSylv-k 算法需要的迭代; $error (Zeng)$ 表示用 Zeng 的算法[74]计算的距离 $\|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2$, 而 $error (STLN)$ 表示用 AppSylv-k 算法得到的距离 $\|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2$; σ_k 和 $\tilde{\sigma}_k$ 分别表示矩阵 $S(f, g)$ 和 $S(\tilde{f}, \tilde{g})$ 的倒数第 k 个奇异值。

§3.3 结构低秩逼近广义 Sylvester 矩阵

§3.3.1 预备知识

设多项式 $f, g \in \mathbb{C}[y_1, y_2, \dots, y_r]$, 且全次数为 $\text{tdeg}(f) = m$ 和 $\text{tdeg}(g) = n$, 也就是:

$$\begin{aligned} f &= a_{0,\dots,0,m}y_r^m + a_{0,\dots,0,1,m-1}y_{r-1}y_r^{m-1} + \dots + a_{m-1,1,0,\dots,0}y_1^{m-1}y_2 + a_{m,0,\dots,0}y_1^m \\ &\quad + \dots + a_{0,\dots,0,1}y_r + \dots + a_{0,1,0,\dots,0}y_2 + a_{1,0,\dots,0}y_1 + a_{0,0,\dots,0}, \\ g &= b_{0,\dots,0,n}y_r^n + b_{0,\dots,0,1,n-1}y_{r-1}y_r^{n-1} + \dots + b_{n-1,1,0,\dots,0}y_1^{n-1}y_2 + b_{n,0,\dots,0}y_1^n \\ &\quad + \dots + b_{0,\dots,0,1}y_r + \dots + b_{0,1,0,\dots,0}y_2 + b_{1,0,\dots,0}y_1 + b_{0,0,\dots,0}. \end{aligned}$$

和单变元多项式的情形一样, 首先给出一个定理说明最小问题 3.1.1 有解。

定理 3.3.1. 设 k 是正整数且 $k \leq \min(m, n)$ 。那么存在一对多项式 $\hat{f}, \hat{g} \in \mathbb{C}[y_1, y_2, \dots, y_r]$, $\text{tdeg}(\hat{f}) \leq m$, $\text{tdeg}(\hat{g}) \leq n$, 且 $\text{tdeg}(\text{GCD}(\hat{f}, \hat{g})) \geq k$ 使得对于所有的 $\tilde{f}, \tilde{g} \in \mathbb{C}[y_1, y_2, \dots, y_r]$, $\text{tdeg}(\tilde{f}) \leq m$, $\text{tdeg}(\tilde{g}) \leq n$ 且 $\text{tdeg}(\text{GCD}(\tilde{f}, \tilde{g})) \geq k$, 则有:

$$\|\hat{f} - f\|_2^2 + \|\hat{g} - g\|_2^2 \leq \|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2.$$

证明. 见文章 [37]. □

跟上面的记号一样, 全次数 $\leq k$ 单项式 $y_1^{i_1} \cdots y_r^{i_r}$ 的个数是二项式数:

$$\beta(k, r) = \binom{k+r}{r}. \quad (3.13)$$

假设 $S(f, g)$ 是广义 Sylvester 矩阵。根据引理 2.2.1, $S(f, g)$ 的亏秩是 $\beta(k-1, r)$ 。因此, 问题 3.1.1 可以转化为:

$$\begin{aligned} &\min_{\text{tdeg}(\text{GCD}(\tilde{f}, \tilde{g})) \geq k} \|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2 \\ \iff &\min_{\dim \text{Nullspace}(\tilde{S}) \geq \beta(k-1, r)} \|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2, \end{aligned}$$

这里 $\tilde{S}$ 是由多项式 $\tilde{f}$ 和多项式 $\tilde{g}$ 生成的广义 Sylvester 矩阵, $\text{tdeg}(\tilde{f}) = \tilde{m} \leq m$ 和 $\text{tdeg}(\tilde{g}) = \tilde{n} \leq n$ 。需要注意的是: $\tilde{f} = 0$ 或者 $\tilde{g} = 0$, 广义 Sylvester 矩阵 $\tilde{S}$ 没有定义。如果一个多项式是零, 需要假设 $\dim \text{Nullspace}(\tilde{S}) \geq \beta(k-1, r)$ (因为 GCD 是其它非零的多项式)。如果 $\tilde{f} = \tilde{g} = 0$, 假设 $\text{Rank}(\tilde{S}) = \beta(m-1, r) + \beta(n-1, r)$ 。

和单变元多项式一样, 构造 k -th 广义 Sylvester 矩阵 $S_k(f, g)$:

$$S_k = \left(\underbrace{\begin{array}{cccc} a_{0,\dots,0,m} & 0 & \dots & 0 \\ a_{0,\dots,0,1,m-1} & a_{0,\dots,0,m} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ a_{m,0,\dots,0} & a_{m-1,1,\dots,0} & \dots & 0 \\ 0 & a_{m,0,\dots,0} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ a_{0,\dots,0,1} & 0 & \dots & 0 \\ a_{0,\dots,0,1,0} & a_{0,\dots,0,1} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & a_{0,0,\dots,0} \end{array}}_{\beta(n-k,r)} \mid \underbrace{\begin{array}{cccc} b_{0,\dots,0,n} & 0 & \dots & 0 \\ b_{0,\dots,0,1,n-1} & b_{0,\dots,0,n} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ b_{n,0,\dots,0} & b_{n-1,1,0,\dots,0} & \dots & 0 \\ 0 & b_{n,0,\dots,0} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ b_{0,\dots,0,1} & 0 & \dots & 0 \\ b_{0,\dots,0,1,0} & b_{0,\dots,0,1} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & b_{0,\dots,0} \end{array}}_{\beta(m-k,r)} \right)$$

对于 $k=1$, $S_k = S$ 是 f 和 g 的广义 Sylvester 矩阵。由引理 2.2.1, 可以得到下面的定理:

定理 3.3.2. 给定 $f, g \in \mathbb{C}[y_1, y_2, \dots, y_r]$, $\text{tdeg}(f) = m, \text{tdeg}(g) = n$, $1 \leq k \leq \min(m, n)$ 。设 $S(f, g)$ 是 f 和 g 的广义 Sylvester 矩阵, S_k 是 f 和 g 的 k -th 广义 Sylvester 矩阵, 那么 $\text{tdeg}(\text{GCD}(f, g)) \geq k$ 当且仅当 S_k 的亏秩大于等于 1。

有了上面的定理, 问题 3.1.1 还可以转化为:

$$\min_{\text{tdeg}(\text{GCD}(\tilde{f}, \tilde{g})) \geq k} \|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2 \iff \min_{\dim \text{Nullspace}(\tilde{S}_k) \geq 1} \|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2$$

这里 $\tilde{S}_k$ 是由 f 和 g 生成的 k -th 广义 Sylvester 矩阵, $\text{tdeg}(\tilde{f}) \leq m$ 和 $\text{tdeg}(\tilde{g}) \leq n$ 。

如果用 STLN [61; 60; 57; 43] 方法解下面的超定系统

$$A_k \mathbf{x} \approx \mathbf{b}_k, \quad (3.14)$$

这里 $\mathbf{b}_k$ 是 S_k 中的一列而 A_k 是 S_k 的剩余列, 那么可以得到最小的有 k -th 广义 Sylvester 结构的扰动 $\mathbf{h}_k$ 和 E_k 使得

$$\mathbf{b}_k + \mathbf{h}_k \in \text{Range}(A_k + E_k).$$

因此, $\mathbf{b}_k + \mathbf{h}_k$ 和 $A_k + E_k$ 得到的 $\tilde{S}_k$ 是有 k -th 广义 Sylvester 结构且 $\dim \text{Nullspace}(\tilde{S}_k) \geq 1$ 的解。

假设 S_k 的亏秩大于等于 1, 下面阐述怎样选列 $\mathbf{b}_k$ 使得 $A_k \mathbf{x} = \mathbf{b}_k$ 有解。首先, 我们计算非零的向量 $\mathbf{w}$ 使得

$$S_k \mathbf{w} = \mathbf{0}. \quad (3.15)$$

给变元选一个序: $y_1 \prec y_2 \prec \cdots \prec y_r$, 多项式按分字典序重新排列。如果把向量

$$\mathbf{y} = [y_r^{n+m-k}, y_{r-1}y_r^{n+m-k-1}, \dots, y_1^{n+m-k}, \dots, y_1, 1] \quad (3.16)$$

乘到矩阵 S_k 上, 那么有

$$\mathbf{y} S_k = [y_r^{n-k}f, y_{r-1}y_r^{n-k-1}f, \dots, f, y_r^{m-k}g, y_{r-1}y_r^{m-k-1}g, \dots, g].$$

线性方程 (3.15) 转化为多项式形式:

$$uf + vg = 0, \quad \text{tdeg}(u) \leq n - k, \quad \text{tdeg}(v) \leq m - k, \quad (3.17)$$

这里非零的多项式 u 和 v 的系数是由向量 $\mathbf{w}$ 生成的。取 u 的非零项 $c_{\alpha_1 \dots \alpha_r} y_1^{\alpha_1} \cdots y_r^{\alpha_r}$ 。 S_k 中的列 $\mathbf{b}_k$ 可以使得

$$\mathbf{y} \mathbf{b}_k = y_1^{\alpha_1} \cdots y_r^{\alpha_r} f. \quad (3.18)$$

设 A_k 是由 S_k 中的剩余列组成, 超定系统 $A_k \mathbf{x} = \mathbf{b}_k$ 始终有解。事实上, 式子 (3.17) 可以写成:

$$\frac{(u - c_{\alpha_1 \dots \alpha_r} y_1^{\alpha_1} \cdots y_r^{\alpha_r})f + vg}{-c_{\alpha_1 \dots \alpha_r}} = y_1^{\alpha_1} \cdots y_r^{\alpha_r} f. \quad (3.19)$$

多项式 $u - c_{\alpha_1 \dots \alpha_r} y_1^{\alpha_1} \cdots y_r^{\alpha_r}$ 没有项 $y_1^{\alpha_1} \cdots y_r^{\alpha_r}$ 。式 (3.19) 可以转变成矩阵形式, 其中多项式系数矩阵为 A_k 和右边的向量为 $\mathbf{b}_k$ 。

假设输入的多项式是不准确的, 则不能得到准确的解使得 $S_k \mathbf{w} = \mathbf{0}$ 。这里可以用奇异值分解计算 S_k 近似的零向量 $\tilde{\mathbf{w}}$ 。由向量 $\tilde{\mathbf{w}}$ 生成的非零的多项式 $\tilde{u}$ 和 $\tilde{v}$ 满足

$$\tilde{u}f + \tilde{v}g \approx 0, \quad \text{tdeg}(\tilde{u}) \leq n - k, \quad \text{tdeg}(\tilde{v}) \leq m - k.$$

设 $\tilde{c}_{\alpha_1 \dots \alpha_r} y_1^{\alpha_1} \cdots y_r^{\alpha_r}$ 是 $\tilde{u}$ 的项且 $|\tilde{c}_{\alpha_1 \dots \alpha_r}| = \|\tilde{u}\|_\infty$ 。取 S_k 的列 $\mathbf{b}_k$ 满足 (3.18)。类似 (3.19), 就说明了 $A_k \mathbf{x} = \mathbf{b}_k$ 可以从多项式的系数得到近似解:

$$\frac{(\tilde{u} - \tilde{c}_{\alpha_1 \dots \alpha_r} y_1^{\alpha_1} \cdots y_r^{\alpha_r})}{-\tilde{c}_{\alpha_1 \dots \alpha_r}}, \quad \frac{\tilde{v}}{-\tilde{c}_{\alpha_1 \dots \alpha_r}}.$$

下面说明对于给定的 k -th 广义 Sylvester 矩阵, 如果矩阵中的任意元素都可以扰动, 那么总是能找到有广义 Sylvester 结构的 $\mathbf{h}_k$ 和 E_k 使得 $\mathbf{b}_k + \mathbf{h}_k \in \text{Range}(A_k + E_k)$, 这里 $\mathbf{b}_k$ 是 S_k 中的某一列, A_k 是 S_k 的剩余列。

定理 3.3.3. 给定正整数 m, n 和 k , $k \leq \min(m, n)$ 。那么, 总是存在亏秩为 1 的 k -th 广义 Sylvester 矩阵 $S_k \in \mathbb{C}^{\beta(m+n-k, r) \times (\beta(m-k, r) + \beta(n-k, r))}$ 。

证明. 类似单变元的定理 3.2.4。 $\square$

推论 3.3.1. 给定正整数 m, n 和 k , $k \leq \min(m, n)$ 。对于任意的 k -th 广义 Sylvester 矩阵 $S_k = [A_{k1} \mathbf{b}_k A_{k2}]$ 和 $S_k \in \mathbb{C}^{\beta(m+n-k, r) \times (\beta(m-k, r) + \beta(n-k, r))}$, 这里 $\mathbf{b}_k$ 是 S_k 的某一列, $A_k = [A_{k1} A_{k2}]$ 是 S_k 的剩余列, 那么总能找到具有 k -th 广义 Sylvester 结构的 $[\mathbf{h}_k, E_k]$ 使得 $\mathbf{b}_k + \mathbf{h}_k \in \text{Range}(A_k + E_k)$ 。

§3.3.2 用 STLN 解决有广义 Sylvester 结构的超定系统

这一小节阐述怎样用 STLN [57; 60; 61] 求解超定系统

$$A_k \mathbf{x} \approx \mathbf{b}_k \quad (3.20)$$

这里 $S_k = [A_{k1} \mathbf{b}_k A_{k2}]$ 是 k -th 广义 Sylvester 矩阵, $\mathbf{b}_k$ 是 S_k 中的一列, $A_k = [A_{k1} A_{k2}]$ 是 S_k 中的剩余列。我们计算最小的扰动 $\mathbf{h}_k$ 和 $E_k = [E_{k1} E_{k2}]$ 使得

$$\mathbf{b}_k + \mathbf{h}_k \in \text{Range}(A_{k1} + E_{k1} A_{k2} + E_{k2}),$$

也就是, $\tilde{S}_k = [A_{k1} + E_{k1} \mathbf{b}_k + \mathbf{h}_k A_{k2} + E_{k2}]$ 是有 k -th 广义 Sylvester 结构且 $\dim \text{Nullspace}(\tilde{S}_k) \geq 1$ 。

为了简单说明问题, 下面假设 $\mathbf{b}_k$ 是 S_k 的第一列, A_k 是 S_k 的剩余列。

向量 $\mathbf{z} \in \mathbb{C}^{(\beta(m, r) + \beta(n, r)) \times 1}$ 表示 k -th 广义 Sylvester 矩阵 $[\mathbf{h}_k E_k]$ 的自由元素:

$$\mathbf{z} = [z_1, z_2, \dots, z_{\beta(m, r) + \beta(n, r) - 1}, z_{\beta(m, r) + \beta(n, r)}]^T.$$

存在一个矩阵 $P_k \in \mathbb{C}^{\beta(m+n-k, r) \times (\beta(m, r) + \beta(n, r))}$ 使得

$$\mathbf{h}_k = P_k \mathbf{z}.$$

矩阵 P_k 里包含了一些单位矩阵块和零矩阵块。例如, 当 $r = 2$, 则有:

$$P_k = \left(\begin{array}{ccc|c} \mathbf{Q}_{m+1} & & & \mathbf{0} \\ & \mathbf{Q}_m & & \\ & & \ddots & \\ & & & \mathbf{Q}_1 \\ \hline & \mathbf{0} & & \mathbf{0} \end{array} \right) \quad (3.21)$$

这里 $\mathbf{Q}_i = \begin{pmatrix} \mathbf{I}_i \\ \mathbf{0} \end{pmatrix}$, $i = 1, 2, \dots, m+1$, $\mathbf{I}_i$ 是 $i \times i$ 单位矩阵, $\mathbf{0}$ 是 $(n-k) \times i$ 零矩阵。

下面求解一个有等式限制的最小二乘问题:

$$\min_{\mathbf{z}, \mathbf{x}} \|\mathbf{z}\|_2, \quad \mathbf{r} = \mathbf{0}, \quad (3.22)$$

这里结构剩余 $\mathbf{r}$ 是

$$\mathbf{r} = \mathbf{r}(\mathbf{z}, \mathbf{x}) = \mathbf{b}_k + \mathbf{h}_k - (A_k + E_k)\mathbf{x}.$$

跟单变元情形一样, 结构最小二乘问题 (3.22) 可以用罚值方法 [1] 去解决, (3.22) 变为:

$$\min_{\mathbf{z}, \mathbf{x}} \left\| \begin{bmatrix} w\mathbf{r}(\mathbf{z}, \mathbf{x}) \\ \mathbf{z} \end{bmatrix} \right\|_2, \quad w \gg 1, \quad (3.23)$$

这里 w 是一个介于 10^8 和 10^{10} 之间的很大的数。同样, 最小问题可以通过对结构剩余 $\mathbf{r}(\mathbf{z}, \mathbf{x})$ 用线性逼近来解决。设 $\Delta\mathbf{z}$, $\Delta\mathbf{x}$ 分别代表 $\mathbf{z}$ 和 $\mathbf{x}$ 的小的变化, ΔE 表示相应 E_k 的变化。那么对 $\mathbf{r}(\mathbf{z} + \Delta\mathbf{z}, \mathbf{x} + \Delta\mathbf{x})$ 的一阶逼近是:

$$\begin{aligned} \mathbf{r}(\mathbf{z} + \Delta\mathbf{z}, \mathbf{x} + \Delta\mathbf{x}) &= \mathbf{b}_k + P_k(\mathbf{z} + \Delta\mathbf{z}) - (A_k + E_k + \Delta E)(\mathbf{x} + \Delta\mathbf{x}) \\ &\approx \mathbf{b}_k + P_k\mathbf{z} - (A_k + E_k)\mathbf{x} + P_k\Delta\mathbf{z} - (A_k + E_k)\Delta\mathbf{x} - \Delta E\mathbf{x} \\ &= \mathbf{r} + P_k\Delta\mathbf{z} - (A_k + E_k)\Delta\mathbf{x} - \Delta E\mathbf{x}. \end{aligned}$$

引入一个类似广义 Sylvester 结构的矩阵 $Y_k \in \mathbb{C}^{\mu \times \nu}$, 这里 $\mu = \beta(m+n-k, r)$, $\nu = \beta(m, r) + \beta(n, r)$, 使得

$$Y_k \mathbf{z} = E_k \mathbf{x} \quad \text{这里} \quad \mathbf{x} = [x_1, x_2, \dots, x_{\beta(m-k)+\beta(n-k)-1}]^T. \quad (3.24)$$

将 $\mathbf{z} = \Delta\mathbf{z}$ 和 $E_k = \Delta E$ 代入 (3.24), 那么 (3.23) 可以由下面的式子来逼近:

$$\min_{\Delta\mathbf{x}, \Delta\mathbf{z}} \left\| \begin{bmatrix} w(Y_k - P_k) & w(A_k + E_k) \\ \mathbf{I}_{\beta(m,r)+\beta(n,r)} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{z} \\ \Delta\mathbf{x} \end{bmatrix} + \begin{bmatrix} -w\mathbf{r} \\ \mathbf{z} \end{bmatrix} \right\|_2. \quad (3.25)$$

注 3.3.1. 为了计算式子 (3.25) 的最小二乘解, 我们运用 QR 分解算法。在文章 [2] 中说明基于自比例快速旋转方法的 QR 算法 (即使采用了列主元的方法) 能得到准确的结果。设

$$M = \begin{bmatrix} w(Y_k - P_k) & w(A_k + E_k) \\ \mathbf{I}_{\beta(m,r)+\beta(n,r)} & \mathbf{0} \end{bmatrix}.$$

因此, 在 M 不是满秩的情形, 仍然可以采用选列主元的 QR 分解方法求最小问题 (3.25)。

下面, 同样给出构造矩阵 Y_k 的方法. 对于 $f, g, E, \mathbf{z}$ 和 $\mathbf{x}$ 还是如上所设. 一个向量 $\mathbf{y}$ 乘到方程 (3.24) 的两边, 得到多项式的等式:

$$\mathbf{y} Y_k \mathbf{z} = \mathbf{y} E_k \mathbf{x}.$$

设 $\hat{\mathbf{x}} = \begin{bmatrix} 0 \\ \mathbf{x} \end{bmatrix}$, 则有:

$$\mathbf{y} E_k \mathbf{x} = \mathbf{y} [\mathbf{h}_k, E_k] \hat{\mathbf{x}} = \hat{g}_1 \hat{u}_1 + \hat{g}_2 \hat{u}_2, \quad (3.26)$$

这里 $\hat{g}_1$ 是由 $\mathbf{z}$ 的子向量生成的多项式, 且全次数为 m , 这个子向量为:

$$[z_1, z_2, \dots, z_{\beta(m,r)}],$$

$\hat{g}_2$ 也是由 $\mathbf{z}$ 的子向量生成的多项式, 且全次数为 n , 这个子向量为:

$$[z_{\beta(m,r)+1}, z_{\beta(m,r)+2}, \dots, z_{\beta(m,r)+\beta(n,r)}],$$

$\hat{u}_1$ 是由 $\hat{\mathbf{x}}$ 的子向量生成的多项式, 且全次数为 $n-k$, 这个子向量为:

$$[0, x_1, x_2, \dots, x_{\beta(n-k,r)-1}],$$

$\hat{u}_2$ 也是由 $\hat{\mathbf{x}}$ 的子向量生成的多项式, 且全次数为 $m-k$, 这个子向量为:

$$[x_{\beta(n-k,r)}, x_{n-k+2}, \dots, x_{\beta(m-k,r)+\beta(n-k,r)-1}].$$

根据 x 的幂次数和变元 z_i 组成的上面的线性系统(3.26) 的系数矩阵就得到了矩阵 Y_k , 下面用一个简单的例子说明矩阵 Y_k 的结构.

例 3.3.1. 设 $k=1$, 矩阵 S 是由二元多项式 f 和 g 生成的, 这里

$$\begin{aligned} f &= a_{00} + a_{01}y + a_{10}x + a_{02}y^2 + a_{11}xy + a_{20}x^2, \\ g &= b_{00} + b_{01}y + b_{10}x + b_{02}y^2 + b_{11}xy + b_{20}x^2. \end{aligned}$$

$$A_1 = \begin{bmatrix} 0 & 0 & b_{02} & 0 & 0 \\ a_{02} & 0 & b_{11} & b_{02} & 0 \\ a_{11} & 0 & b_{20} & b_{11} & 0 \\ a_{20} & 0 & 0 & b_{20} & 0 \\ 0 & a_{02} & b_{01} & 0 & b_{02} \\ a_{01} & a_{11} & b_{10} & b_{01} & b_{11} \\ a_{10} & a_{20} & 0 & b_{10} & b_{20} \\ 0 & a_{01} & b_{00} & 0 & b_{01} \\ a_{00} & a_{10} & 0 & b_{00} & b_{10} \\ 0 & a_{00} & 0 & 0 & b_{00} \end{bmatrix}, \quad b_1 = \begin{bmatrix} a_{02} \\ a_{11} \\ a_{20} \\ 0 \\ a_{01} \\ a_{10} \\ 0 \\ a_{00} \\ 0 \\ 0 \end{bmatrix}.$$

设 $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5]^T$, 那么

$$Y_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & x_3 & 0 & 0 & 0 & 0 & 0 \\ x_1 & 0 & 0 & 0 & 0 & 0 & x_4 & x_3 & 0 & 0 & 0 & 0 \\ 0 & x_1 & 0 & 0 & 0 & 0 & 0 & x_4 & x_3 & 0 & 0 & 0 \\ 0 & 0 & x_1 & 0 & 0 & 0 & 0 & 0 & x_4 & 0 & 0 & 0 \\ x_2 & 0 & 0 & 0 & 0 & 0 & x_5 & 0 & 0 & x_3 & 0 & 0 \\ 0 & x_2 & 0 & x_1 & 0 & 0 & 0 & x_5 & 0 & x_4 & x_3 & 0 \\ 0 & 0 & x_2 & 0 & x_1 & 0 & 0 & 0 & x_5 & 0 & x_4 & 0 \\ 0 & 0 & 0 & x_2 & 0 & 0 & 0 & 0 & 0 & x_5 & 0 & x_3 \\ 0 & 0 & 0 & 0 & x_2 & x_1 & 0 & 0 & 0 & 0 & x_5 & x_4 \\ 0 & 0 & 0 & 0 & 0 & x_2 & 0 & 0 & 0 & 0 & 0 & x_5 \end{bmatrix}.$$

我们可以看出矩阵 Y_1 还是具有拟的广义 *Sylvester* 结构。

§3.3.3 多变元近似最大公因子的算法和实验

下面的算法是计算问题 3.1.1 的近似解。

算法 3.3.1. 输入: 一对多变元多项式 $f, g \in \mathbb{C}[y_1, y_2, \dots, y_r]$, 全次数分别为 $m \geq n$, 一个正整数 $1 \leq k \leq n$ 和容许误差 tol 。

输出: 多项式 $\tilde{f}$ 和 $\tilde{g}$, 使得 $\dim \text{Nullspace}(S(\tilde{f}, \tilde{g})) \geq \beta(k-1, r)$ 且 *Euclidean* 长度 $\|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2$ 最小。

1. 构造 k -th 广义 *Sylvester* 矩阵 S_k , 计算 S_k 的近似右奇异向量, 按照前面的方法选出一列 $\mathbf{b}_k$ 和 S_k 的剩余列组成的矩阵 A_k 。设 $E_k = \mathbf{0}$, $\mathbf{h}_k = \mathbf{0}$ 。
2. 根据 $\min \|A_k \mathbf{x} - \mathbf{b}_k\|_2$ 计算 $\mathbf{x}$ 和剩余 $\mathbf{r} = \mathbf{b}_k - A_k \mathbf{x}$ 。按照上面一节, 构造矩阵 P_k 和 Y_k 。
3. 迭代

$$(a) \min_{\Delta \mathbf{x} \Delta \mathbf{z}} \left\| \begin{bmatrix} w(Y_k - P_k) & w(A_k + E_k) \\ I_{\beta(m,r)+\beta(n,r)} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{z} \\ \Delta \mathbf{x} \end{bmatrix} + \begin{bmatrix} -w\mathbf{r} \\ \mathbf{z} \end{bmatrix} \right\|_2.$$

(b) 设 $\mathbf{x} = \mathbf{x} + \Delta \mathbf{x}$, $\mathbf{z} = \mathbf{z} + \Delta \mathbf{z}$ 。

(c) 从 $\mathbf{z}$ 构造矩阵 E_k 和 $\mathbf{h}_k$, 从 $\mathbf{x}$ 构造矩阵 Y_k 。更新 $A_k = A_k + E_k$, $\mathbf{b}_k = \mathbf{b}_k + \mathbf{h}_k$, $\mathbf{r} = \mathbf{b}_k - A_k \mathbf{x}$,
直到 $(\|\Delta \mathbf{x}\|_2 \leq tol \text{ 和 } \|\Delta \mathbf{z}\|_2 \leq tol)$ 。

4. 根据 $\mathbf{b}_k$ 和 A_k 输出多项式 $\tilde{f}$ 和 $\tilde{g}$

例 3.3.2. 给定多项式

$$\begin{aligned} p &= 3 + y + 5x + 5y^2 + xy + 2x^2, \\ f &= p \times (3 + 4y + 3x) + 0.04y + 0.03x + 0.01xy + 0.04y^3, \\ g &= p \times (2 + y + 5x) + 0.06 + 0.02xy + 0.06y^2 + 0.04x^3, \end{aligned}$$

和 $\epsilon = 10^{-2}$ 。计算 f 和 g 的 ϵ -GCD:

设 $tol = 10^{-3}$ 和 $k = 2$, $\beta(1, 2) = 3$, S 是 f 和 g 的广义 Sylvester 矩阵。 S 的最后几个奇异值为

$$\dots, 19.5963, 16.4921, 13.3136, 0.08342, 0.04921, 0.03893.$$

对 2-th 广义 Sylvester 矩阵 S_2 应用算法 3.3.1。

计算出 S_2 的最小的奇异值所对应的奇异向量 $\mathbf{w}$:

$$\mathbf{w} = [-0.1257, -0.6244, -0.2504, 0.5008, 0.3744, 0.3750]^T.$$

根据 $\mathbf{w}$ 中元素的最大绝对值, 取第二列为 $\mathbf{b}_2$ 。经过三次迭代, 得到矩阵 $\tilde{S}_2$:

$$\tilde{S}_2 = \begin{pmatrix} 20.036 & 0 & 0 & 5.008 & 0 & 0 \\ 19.001 & 20.036 & 0 & 25.994 & 5.008 & 0 \\ 11.020 & 19.001 & 0 & 6.991 & 25.994 & 0 \\ 6.012 & 11.020 & 0 & 10.015 & 6.991 & 0 \\ 0 & 6.012 & 0 & 0 & 10.015 & 0 \\ 19.019 & 0 & 20.036 & 11.048 & 0 & 5.008 \\ 26.013 & 19.019 & 19.001 & 12.008 & 11.048 & 25.994 \\ 21.007 & 26.013 & 11.020 & 28.998 & 12.008 & 6.991 \\ 0 & 21.007 & 6.012 & 0 & 28.998 & 10.015 \\ 15.025 & 0 & 19.019 & 5.015 & 0 & 11.048 \\ 24.002 & 15.025 & 26.013 & 25.035 & 5.015 & 12.008 \\ 0 & 24.002 & 21.007 & 0 & 25.035 & 28.998 \\ 9.007 & 0 & 15.025 & 6.028 & 0 & 5.015 \\ 0 & 9.007 & 24.002 & 0 & 6.028 & 25.035 \\ 0 & 0 & 9.007 & 0 & 0 & 6.028 \end{pmatrix}.$$

$\tilde{S}_2$ 的奇异值为:

$$52.930, 47.759, 19.582, 15.201, 7.776 \times 10^{-13}.$$

从矩阵 $\tilde{S}_2$ 可以得到多项式 $\tilde{f}, \tilde{g}$:

$$\begin{aligned}\tilde{f} &= 9.007 + 15.025y + 24.002x + 19.019y^2 + 26.013xy \\ &\quad + 21.007x^2 + 20.036y^3 + 19.001xy^2 + 11.020x^2y + 6.012x^3, \\ \tilde{g} &= 6.028 + 5.015y + 25.035x + 11.048y^2 + 12.008xy \\ &\quad + 28.998x^2 + 5.008y^3 + 25.994xy^2 + 6.991x^2y + 10.015x^3.\end{aligned}$$

最小的扰动是:

$$\|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2 = 0.005684.$$

因为 S 的倒数第四到第六个奇异值是在 $O(10)$ 阶里, 则 f 和 g 如果有次数为 3 的 GCD 所需要的扰动最少是同一阶的。因此对于 $\epsilon = 10^{-2}$, 我们说 f 和 g 的 ϵ - GCD 最高次数为 2。而且, 根据上一章的多变元多项式的近似最大公因子算法从 $\tilde{S}_2$ 中可计算出 GCD :

$$\tilde{p} = y^2 + 0.19941yx + 0.40066x^2 + 0.20006y + 0.99911x + 0.60001.$$

ϵ - GCD 的向前误差为:

$$\|p - 5\tilde{p}\|_2 / \|p\|_2 = 0.001033.$$

<i>Ex.</i>	m, n	k	<i>it.</i>	<i>error1</i>	<i>error2</i>	<i>error3</i>	σ_k	$\tilde{\sigma}_k$
1	3, 3	2	1	$6.20e - 10$	$3.89e - 10$	$2.96e - 10$	$1.45e - 5$	10^{-8}
2	7, 7	2	1	$6.92e - 9$	$1.54e - 9$	$1.31e - 9$	$1.87e - 5$	10^{-8}
3	2, 2	1	2	$6.88e - 3$	$9.01e - 4$	$3.39e - 4$	$1.10e - 2$	10^{-9}
4	3, 2	2	2	$1.17e - 3$	$4.00e - 4$	$3.51e - 4$	$1.48e - 2$	10^{-9}
5	3, 3	2	1	$9.06e - 6$	$4.48e - 6$	$4.16e - 6$	$2.31e - 3$	10^{-8}
6	6, 4	3	2	$2.01e - 5$	$1.88e - 5$	$9.73e - 6$	$2.05e - 3$	10^{-8}
7	6, 4	3	2	$2.01e - 3$	$1.88e - 3$	$9.72e - 4$	$2.05e - 2$	10^{-8}
8	10, 10	5	3	$4.51e - 3$	$2.64e - 3$	$2.20e - 3$	$3.61e - 2$	10^{-7}
9	15, 15	6	2	$1.57e - 4$	$8.35e - 5$	$5.90e - 5$	$5.36e - 3$	10^{-7}

表 3.2 算法3.3.1的结果 (两个变元)

在表 3.2 中, 我们对 Maple 10 中且 $Digits = 14$ 任意生成的两个变元的多项式用算法3.3.1计算 ϵ - GCD 。这里 m 和 n 分别表示 f 和 g 的全次数; k 是 f 和 g 的近似最大公因子次数; *it.* 表示用算法3.3.1 所需的迭代次数; *error1* 表示用 Zeng [74] 的

算法计算的扰动 $\|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2$; $error2$ 表示用上一章多变元多项式的近似最大公因子的算法计算出的扰动 $\|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2$; 而 $error3$ 表示用算法 3.3.1 计算的扰动 $\|\tilde{f} - f\|_2^2 + \|\tilde{g} - g\|_2^2$; σ_k 和 $\tilde{\sigma}_k$ 分别表示 $S(f, g)$ 和 $S(\tilde{f}, \tilde{g})$ 的最后第 $\beta(k-1, r)$ 个奇异值。

§3.4 小结

这一章给出了基于 STLN 方法计算两个单变元和多变元多项式的最大公因子的算法。算法 (单变元和多变元) 的复杂度是依赖于第三步中迭代的次数和每次迭代的复杂度。如果初始值很好, 第三步的迭代很快就收敛。关于这一点, 可以从表 3.1 和表 3.2 中看出。对于每次迭代, 我们应用了 Givens 旋转。对于单变元多项式, 根据检验第三步中的最小问题的矩阵的维数, 可以得到每步迭代所需要的计算量小于 $(4m + 4n - k + 6)(2m + 2n - 2k + 3)^2$, 这里 $m \geq n \geq k$ 表示给定多项式和它们的近似最大公因子的次数; 对于多变元多项式, 得到算法的复杂度为 $O\left(\binom{m+n-k+r}{r} \binom{m+r}{r}^2\right)$, 这里 $m \geq n \geq k$ 表示给定多项式和它们的近似最大公因子的全次数, r 是多项式变元的个数。因为最小二乘问题中的矩阵是结构矩阵, 他们有很低的位移秩[30]。对于单变元的情形, 在文章[46]中, 我们应用了快速算法计算最小问题。这样, 把算法的复杂度减少到关于给定多项式次数的平方大小。关于细节, 见[46]。

第四章 多个有线性限制多项式的近似最大公因子和奇异多项式

§4.1 介绍

这一章做的工作有以下几个方面：第一，主要解决实数域或复数域上的多个 ($s \geq 2$) 多项式近似最大公因子问题。同样，这里还是应用 STLN[57]方法来计算近似最大公因子问题。第二，我们计算系数有线性限制的多项式的近似最大公因子。这些限制可以使输入的多项式的系数保持某些性质，例如：首一的或者稀疏的。同样可以使得输入多项式的系数满足一些线性关系（等式或者不等式）。另外还给出了一个适合计算有线性限制的多个多项式（单变元和多变元）近似最大公因子的 STLN 优化的线性系统，该系统的维数比没有线性限制情形时的线性系统的维数要小。第三，应用有线性限制的近似最大公因子算法来计算最近的奇异多项式，即多项式有重根，重数大于等于 2 [81]。

通过计算一些例子说明 STLN 算法可以计算近似最大公因子和最近的奇异多项式的全局最优解。尽管基于最小二乘方法的 STLN 算法不能保证全局最小，然而实验的例子说明了 STLN 算法得到的解与文章 [41; 25; 81; 80]中的全局最优解一致。

STLN 算法能处理输入多项式离有非平凡的公因子或者有 k 重根的多项式的距离很远的情形，尤其是当输入的多项式离满足线性限制的多项式很远的情形。而且，STLN 算法还能处理输入多项式的系数是实数的，但是离它们最近有非平凡公因子或者奇异的多项式的系数是复数的情形。这时，STLN 算法通过给初始的向量加一个随机的扰动来计算复的最小解。STLN 方法还采用了 Lagrangian 乘子的方法计算初始的向量。

在近似最大公因子问题中，输入为多项式 $f_1, \dots, f_s$ 和一个正整数 k ，变元为 $y_1, \dots, y_r$ ，多项式的系数为实数或者复数。输出为公因子 g 和 $f_1^*, \dots, f_s^*$ ，系数为实数或复数，并且 $\deg(g) \geq k$ ， $\deg(\tilde{f}_i) \leq \deg(f_i)$ ，这里 $\tilde{f}_i = gf_i^*$ ， $1 \leq i \leq s$ ；还使得 $\|\Delta f_1\| + \dots + \|\Delta f_s\|$ 为最小，这里 $\Delta f_i = \tilde{f}_i - f_i$ ， $1 \leq i \leq s$ 。

对于多变元多项式，我们有多种次数定义。文章[34]说明用每个变元的次数比全次数能得到更近的解。关于改变的多项式 Δf_i 的范数的定义同样会影响最优解。通常情况下用的是 Euclidean 范数。文章[26]引用了不同的范数 l_p 。同样一个重要的问题是多项式的系数域。在上一章中的注 3.2.1中已经说明：对于多项式 $f_1 = x^2 + 2$ 和 $f_2 = x^2 + 1$ ，离最近的有公共根的一对复的多项式比离最近的有公共根的一对实的多项式要近（Euclidean 范数）。以前很多算法都是计算实的最优解。而 STLN 能从实的输入出发找到复的最优解，只是算法需要更多次的迭代。

最后，全局最优解存在但不唯一。定理3.2.1和定理3.3.1 说明了近似最大公因子

问题始终存在最近的全局复的和实的最近解(2范数)。当输入的是实的多项式,如果得到了最近的复的最优解,那么它的共轭同样是最优的解。同样,也可能存在多个实的最优解(注 3.2.1)。事实上,文章 [81]说明了计算最近的奇异多项式时存在多个最优解。对于近似因式分解,文章[34]也说明了多个最优解存在。文章 [57, 小节 4.3]说明了最近的奇异 Toeplitz/Hankel 矩阵问题也存在无穷个最优解。对于近似最大公因子问题,多项式 $f_1 = x + 1$ 和 $f_2 = x - 1$, 离它们最近(2范数)的有公共根的多项式有无穷对,也就是, $\tilde{f}_1 = \frac{-\sigma + i\tau + 1}{\sigma^2 + \tau^2 + 1}(x - \sigma - i\tau)$ 和 $\tilde{f}_2 = \frac{\sigma - i\tau + 1}{\sigma^2 + \tau^2 + 1}(x - \sigma - i\tau)$, 这里 $i = \sqrt{-1}$, $\|\tilde{f}_1 - f_1\|_2^2 + \|\tilde{f}_2 - f_2\|_2^2 = 2$, $\sigma, \tau \in \mathbb{R}$ 。

对于给定多项式,我们计算最近的有 k 重根的奇异多项式。文章[81]给出了有多项式时间的算法计算最近的有 k 重根的奇异多项式。在[25, 小节 6.3]中,通过给输出多项式的系数加一些线性等式限制条件,就可以用计算近似最大公因子算法计算最近的奇异多项式。

§4.2 转化为线性代数问题

我们将多项式的 GCD 问题转化为线性代数的问题,然后应用结构最小二乘方法。我们把上一章的近似最大公因子的式子推广到 s 个多项式的情形。在这里,用全次数来定义多项式的次数。当然,其它的次数定义也适用。下面先给出一个引理。

引理 4.2.1. 设 $f_1, \dots, f_s \in F[y_1, \dots, y_r] \setminus \{0\}$, 这里 F 是任意的数域, 设 $d_i = \text{tdeg}(f_i)$ 和 $k \leq d_i$, 对于所有的 i 有 $1 \leq i \leq s$ 。那么 $\text{tdeg}(\text{GCD}(f_1, \dots, f_s)) \geq k$ 当且仅当存在多项式 $u_1, \dots, u_s \in F[y_1, \dots, y_r]$ 使得

$$u_1 \neq 0, \quad \forall j, 2 \leq j \leq s: u_j f_1 + u_1 f_j = 0, \quad \forall i, 1 \leq i \leq s: \text{tdeg}(u_i) \leq d_i - k. \quad (4.1)$$

证明. (4.1)表示一个事实是: 在唯一的因式分解环 $F[y_1, \dots, y_r]$ 上, 最大公因子能在分式 f_j/f_1 中约化掉。□

条件(4.1)可以使得对于 u_i 的未知系数形成齐次的线性系统。取 f_i 中次数最低的为左侧的多项式, 这样使得生成的系数矩阵有最小的维数。下面研究系数矩阵的结构。假设次数为 l 的多项式 u 的系数向量为 $\tilde{\mathbf{u}}$, 则得到多项式 $u \cdot f$ 的系数向量为: $C^{[l]}(f) \cdot \tilde{\mathbf{u}}$ 。在多变元的情形, 如果 $\text{tdeg}(f) = m$, 则 $C^{[l]}(f)$ 的维数为 $\binom{l+m+r}{r} \times \binom{l+r}{r}$ 。则 (4.1) 的系数矩阵为:

$$S_k(f_1, \dots, f_s) = \begin{bmatrix} C^{[d_2-k]}(f_1) & \mathbf{0} & \dots & \mathbf{0} & C^{[d_1-k]}(f_2) \\ \mathbf{0} & C^{[d_3-k]}(f_1) & & \mathbf{0} & C^{[d_1-k]}(f_3) \\ \vdots & & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & C^{[d_s-k]}(f_1) & C^{[d_1-k]}(f_s) \end{bmatrix} \quad (4.2)$$

矩阵 $S_k(f_1, \dots, f_s)$ 就是一个关于多个多项式 $f_1, \dots, f_s$ 的广义 Sylvester 矩阵。

引理 4.2.2. 设 $f_1, \dots, f_s$ 和 k 如引理 4.2.1 中所设。那么 $\text{tdeg}(\text{GCD}(f_1, \dots, f_s)) \geq k$ 当且仅当在 (4.2) 中的矩阵 $S_k(f_1, \dots, f_s)$ 的亏秩大于等于 1。

证明. 充分性: 直接从引理 4.2.1 得到, 因为 $u_1 \neq 0$ 产生一个非零的列是 $S_k(f_1, \dots, f_s)$ 的零向量。必要性: 因为 (4.2) 中前面的 $s-1$ 个块列组成一组线性无关的列向量, 所以 $S_k(f_1, \dots, f_s)$ 中右零空间中的向量对应于 (4.1) 中的多项式 u_1 一定有非零的元素。然后我们再应用引理 4.2.1 就证明了必要条件。□

根据引理 4.2.2, 给出下面的一个引理。这样就可以把一个多项式有 k 重因子问题转化为多项式的 GCD 问题。

引理 4.2.3. 设 $f(y) \in F[y]$ 是特征为 0 的域 F 上次数为 n 的多项式, 设 k 是重数且 $2 \leq k \leq n$ 。用 $f^{[i]} = d^i f / dy^i$ 记为 f 的 i 次导数。那么下面的条件是等价的:

(i) 存在一个多项式 $h(y) \in F[y]$ 且 $\deg(h) \geq 1$, 使得 h^k 是 f 的因子。

(ii) $\deg(\text{GCD}(f^{[0]}, \dots, f^{[k-1]})) \geq 1$ 。

(iii) 矩阵

$$S_k^{\text{sing}}(f) = \begin{bmatrix} C^{[n-1]}(f^{[k-1]}) & \mathbf{0} & \dots & \mathbf{0} & C^{[n-k]}(f^{[0]}) \\ \mathbf{0} & C^{[n-2]}(f^{[k-1]}) & & \mathbf{0} & C^{[n-k]}(f^{[1]}) \\ \vdots & & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & C^{[n-k+1]}(f^{[k-1]}) & C^{[n-k]}(f^{[k-2]}) \end{bmatrix} \quad (4.3)$$

的亏秩大于等于 1。

证明. 这个引理可以很快地从这个事实上得到: $f = h_1 h_2^2 \cdots h_m^m$, 这里 h_i 无平方因子且两两互素, 那么 $\text{GCD}(f, df/dy) = h_2 h_3^2 \cdots h_m^{m-1}$ (见 [19])。□

在第二章中的近似因式分解算法, 近似重因子要分离出来。引理 4.2.3 有一个关于多变元多项式的推论。可以通过 STLN 方法把近似重因子除掉, 然后计算近似无重因子多项式的绝对不可约分解。这对近似因式分解有非常重要的作用。

推论 4.2.1. 设 $f(y_1, \dots, y_r) \in F[y_1, \dots, y_r]$ 是特征为 0 的域 F 上的多项式, 并且对于变元 y_1 的次数为 n 。设 k 是重数且 $2 \leq k \leq n$ 。假设 f 对于 y_1 是本原的, 也就

是说 f 在 $F[y_2, \dots, y_r]$ 上没有因子。那么存在多项式 $h(y_1, \dots, y_r) \in F[y_1, \dots, y_r]$ 且 $\deg(h) \geq 1$ 使得 h^k 是 f 的因子当且仅当矩阵

$$S_{y_1, k}^{sing}(f) = \begin{bmatrix} C^{[n-1]}(\frac{\partial^{k-1}}{\partial y_1^{k-1}} f) & \mathbf{0} & \dots & \mathbf{0} & C^{[n-k]}(f) \\ \mathbf{0} & C^{[n-2]}(\frac{\partial^{k-1}}{\partial y_1^{k-1}} f) & & \mathbf{0} & C^{[n-k]}(\frac{\partial}{\partial y_1} f) \\ \vdots & & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & C^{[n-k+1]}(\frac{\partial^{k-1}}{\partial y_1^{k-1}} f) & C^{[n-k]}(\frac{\partial^{k-2}}{\partial y_1^{k-2}} f) \end{bmatrix} \quad (4.4)$$

的亏秩大于等于 1。

证明. 我们在系数域为 $L = F(y_2, \dots, y_r)$ 上应用引理 4.2.3。因为已经假设 f 对变元 y_1 是本原的, 根据 Guass 引理[19], 所有在 $L[y_1]$ 中的因子对应于 $F[y_1, \dots, y_r]$ 中的因子, 而且关于变元 y_1 的次数大于等于 1。□

§4.3 基于 STLN 的解

计算矩阵 S 的结构逼近矩阵, 使得逼近矩阵是亏秩的, 这里 b 是从矩阵 S 中选的一列, 而 A 是由剩余列组成的。一般考虑的结构矩阵有对称的, 稀疏的, 循环的, Toeplitz, Hankel 等矩阵。前面一章考虑的是 (广义) Sylvester 矩阵。这一章考虑怎样把 STLN 算法推广应用到计算多个多项式的广义 Sylvester 矩阵的结构低秩逼近, 并且应用于计算近似最大公因子问题。

设 $S(\zeta) = [A_1(\zeta) \mid b(\zeta) \mid A_2(\zeta)]$ 和 $A(\zeta) = [A_1(\zeta) \mid A_2(\zeta)]$ 。这里矩阵 S , A 和向量 b 都是通过向量 ζ 参数表示。在 (4.2) 中的这种形式 $S = S_k(f_1, \dots, f_s)$, 参数向量 ζ 包含多项式 $f_1, \dots, f_s$ 的系数; 在情形 $S = S_k^{sing}(f)$ 或 $S = S_{y_1, k}^{sing}(f)$, 参数向量 ζ 就是多项式 f 的系数向量。我们需要解下面两个保持结构的最小二乘问题:

$$\min_{\Delta \mathbf{c} \in \mathbb{R}^\nu} \|\Delta \mathbf{c}\| \text{ 或 } \min_{\Delta \mathbf{c} \in \mathbb{C}^\nu} \|\Delta \mathbf{c}\| \text{ 满足 } A(\mathbf{c} + \Delta \mathbf{c})\mathbf{x} = b(\mathbf{c} + \Delta \mathbf{c}), \text{ 对于某个向量 } \mathbf{x}, \quad (4.5)$$

这里 $\mathbf{c}$ 是初始的系数向量¹。这里 $\|\cdot\|$ 可以是 2-, 1- 和 ∞ -范数。象第三章一样, 我们取第一个奇异向量中的绝对值最大的那个元素所对应的那一列作为 b 。

STLN 算法首先设 $\Delta \mathbf{c} = \mathbf{z} = \mathbf{0}$, 用 $A(\mathbf{c})\mathbf{x} \approx b(\mathbf{c})$ 的非结构最小二乘解做为 $\mathbf{x}$ 的初始值。然后用迭代同时优化 $\mathbf{x}$ 和 $\mathbf{z}$: 更新的 $\mathbf{x} + \Delta \mathbf{x}$ 和 $\mathbf{z} + \Delta \mathbf{z}$ 满足 (4.5), 也就是:

$$\min \|\mathbf{z} + \Delta \mathbf{z}\| \text{ 满足 } A(\mathbf{c} + \mathbf{z} + \Delta \mathbf{z})(\mathbf{x} + \Delta \mathbf{x}) = b(\mathbf{c} + \mathbf{z} + \Delta \mathbf{z}).$$

然而, 这个优化问题有很多局部最优解, 经典的取初始值的方法[57]是不够的。因此, 我们将基于 Lagrangian 乘子的取初始值方法 [43, 小节 4.5.3]应用到计算近似

¹在上一章, 矩阵 $A(\mathbf{c})$ 用 A 表示, 扰动矩阵 $A(\Delta \mathbf{c})$ 用 E 表示, 扰动向量 $\Delta \mathbf{c}$ 用 $\mathbf{z}$ 表示。

最大公因子问题上。因为矩阵和向量的乘积 $S(\zeta)\xi$ 可以表示成乘积多项式的和，因此，存在类似的 Sylvester 矩阵 $H(\xi)$ 使得 $H(\xi)\zeta = S(\zeta)\xi$ 。假设 $S(\mathbf{c})$ 的第一个奇异向量是 $\mathbf{v}$ ；计算出向量 $\mathbf{z}$ ：

$$\mathbf{z} = -H(\mathbf{v})^{Tr}(H(\mathbf{v})H(\mathbf{v})^{Tr})^{-1}S(\mathbf{c})\mathbf{v}. \quad (4.6)$$

设 $b(\mathbf{c})$ 是 $\mathbf{v}$ 中绝对值最大的元素所对应的第 t 列，通过对向量 $\mathbf{v}$ 正规化且使得 $\mathbf{v}[t] = -1$ ，计算向量 $\mathbf{x}$ ，也就是，

$$\mathbf{x} = \left[-\frac{\mathbf{v}[1]}{\mathbf{v}[t]}, \dots, -\frac{\mathbf{v}[t-1]}{\mathbf{v}[t]}, -\frac{\mathbf{v}[t+1]}{\mathbf{v}[t]}, \dots \right]^{Tr}. \quad (4.7)$$

代入 (4.6) 我们有 $-S(\mathbf{z})\mathbf{v} = -H(\mathbf{v})\mathbf{z} = S(\mathbf{c})\mathbf{v}$ ，因此， $S(\mathbf{c} + \mathbf{z})\mathbf{v} = 0$ ，或者 $A(\mathbf{c} + \mathbf{z})\mathbf{x} = b(\mathbf{c} + \mathbf{z})$ ，这就是我们所需要的。

因为我们的参数化是线性的，这样可以对结构剩余做一阶逼近：

$$\begin{aligned} \mathbf{r}(\mathbf{z} + \Delta\mathbf{z}, \mathbf{x} + \Delta\mathbf{x}) &= b(\mathbf{c} + \mathbf{z} + \Delta\mathbf{z}) - A(\mathbf{c} + \mathbf{z} + \Delta\mathbf{z})(\mathbf{x} + \Delta\mathbf{x}) \\ &= b(\mathbf{c} + \mathbf{z}) + b(\Delta\mathbf{z}) - (A(\mathbf{c} + \mathbf{z}) + A(\Delta\mathbf{z}))(\mathbf{x} + \Delta\mathbf{x}) \\ &\approx b(\mathbf{c} + \mathbf{z}) - A(\mathbf{c} + \mathbf{z})\mathbf{x} + b(\Delta\mathbf{z}) - A(\mathbf{c} + \mathbf{z})\Delta\mathbf{x} - A(\Delta\mathbf{z})\mathbf{x} \\ &= \mathbf{r}(\mathbf{z}, \mathbf{x}) + b(\Delta\mathbf{z}) - A(\mathbf{c} + \mathbf{z})\Delta\mathbf{x} - A(\Delta\mathbf{z})\mathbf{x}. \end{aligned}$$

因为 $b(\zeta)$ 中的元素都可以由 ζ 表示，所以有一个常数矩阵 P 使得 $b(\zeta) = P\zeta$ 。而且，通过把 $H(\xi)$ 中对应于右边向量 $-b$ 的元素从 -1 改变为 0 ，得到第二个类似 Sylvester 矩阵 $Y(\xi)$ ，使得 $Y(\xi)\zeta = A(\zeta)\xi$ ，也就是 $A(\Delta\mathbf{z})\mathbf{x} = Y(\mathbf{x})\Delta\mathbf{z}$ 。因此，得到一个新的剩余量的一次逼近，表示为：

$$\mathbf{r}(\mathbf{z} + \Delta\mathbf{z}, \mathbf{x} + \Delta\mathbf{x}) \approx \mathbf{r}(\mathbf{z}, \mathbf{x}) + P\Delta\mathbf{z} - A(\mathbf{c} + \mathbf{z})\Delta\mathbf{x} - Y(\mathbf{x})\Delta\mathbf{z}. \quad (4.8)$$

将罚值 $w \gg 1$ 应用于剩余[1] 上，最小问题

$$\min_{\Delta\mathbf{z}, \Delta\mathbf{x}} \left\| \begin{matrix} w\mathbf{r}(\mathbf{z} + \Delta\mathbf{z}, \mathbf{x} + \Delta\mathbf{x}) \\ \mathbf{z} + \Delta\mathbf{z} \end{matrix} \right\|$$

有一阶的迭代更新[57, 方程 (2.9)]：

$$\min_{\Delta\mathbf{x}, \Delta\mathbf{z}} \left\| \begin{bmatrix} w(Y(\mathbf{x}) - P) & wA(\mathbf{c} + \mathbf{z}) \\ \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{z} \\ \Delta\mathbf{x} \end{bmatrix} + \begin{bmatrix} -w\mathbf{r}(\mathbf{z}, \mathbf{x}) \\ \mathbf{z} \end{bmatrix} \right\| \quad (4.9)$$

当 $\|\Delta\mathbf{x}\|$ 和 $\|\Delta\mathbf{z}\|$ 小于给定的容许误差时，迭代 $\mathbf{x} = \mathbf{x} + \Delta\mathbf{x}$ 和 $\mathbf{z} = \mathbf{z} + \Delta\mathbf{z}$ 停止。同样，(4.9) 可以转化为有线性方程限制的最小二乘问题（见小节 §4.3.1 和小节 §4.5）。

对于 1-范数和 ∞ -范数, 这个问题可以用线性规划来求解 [25, 小节 8.3]。然而, 如果最优问题考虑的是复数的情形 ((4.5)第二种情形), 实数部分和复数部分需要分开。那么有 $A(\zeta) = A(\zeta_R + i\zeta_I) = A(\zeta_R) + iA(\zeta_I)$, 这里 ζ_R 和 ζ_I 是 ζ 的实数部分和虚数部分, 且 $i = \sqrt{-1}$ 。同样, 这里要把剩余向量的实部和虚部分开, 更新的剩余量 (4.8)可以写成

$$\begin{aligned} & \mathbf{r}(\mathbf{z} + \Delta\mathbf{z}, \mathbf{x} + \Delta\mathbf{x}) \\ & \approx \mathbf{r}_R(\mathbf{z}, \mathbf{x}) + i\mathbf{r}_I(\mathbf{z}, \mathbf{x}) + P\Delta\mathbf{z}_R + iP\Delta\mathbf{z}_I \\ & \quad - (A(\mathbf{c}_R + \mathbf{z}_R) + iA(\mathbf{c}_I + \mathbf{z}_I))(\Delta\mathbf{x}_R + i\Delta\mathbf{x}_I) - (Y(\mathbf{x}_R) + iY(\mathbf{x}_I))(\Delta\mathbf{z}_R + i\Delta\mathbf{z}_I) \\ & = \mathbf{r}_R(\mathbf{z} + \Delta\mathbf{z}, \mathbf{x} + \Delta\mathbf{x}) + i\mathbf{r}_I(\mathbf{z} + \Delta\mathbf{z}, \mathbf{x} + \Delta\mathbf{x}). \end{aligned}$$

迭代的式子 (4.9) 可以转化为一个实的优化问题:

$$\min_{\Delta\mathbf{x}_R, \Delta\mathbf{x}_I, \Delta\mathbf{z}_R, \Delta\mathbf{z}_I} \left\| M \begin{bmatrix} \Delta\mathbf{z}_R & \Delta\mathbf{z}_I & \Delta\mathbf{x}_R & \Delta\mathbf{x}_I \end{bmatrix}^{Tr} + \begin{bmatrix} -w\mathbf{r}_R(\mathbf{z}, \mathbf{x}) & -w\mathbf{r}_I(\mathbf{z}, \mathbf{x}) & \mathbf{z}_R & \mathbf{z}_I \end{bmatrix}^{Tr} \right\| \quad (4.10)$$

这里

$$M = \begin{bmatrix} w(Y(\mathbf{x}_R) - P) & -wY(\mathbf{x}_I) & wA(\mathbf{c}_R + \mathbf{z}_R) & -wA(\mathbf{c}_I + \mathbf{z}_I) \\ wY(\mathbf{x}_I) & w(Y(\mathbf{x}_R) - P) & wA(\mathbf{c}_I + \mathbf{z}_I) & wA(\mathbf{c}_R + \mathbf{z}_R) \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (4.11)$$

[57, 方程 (2.12)].

§4.3.1 输入系数有线性限制的解法

在 (4.3)中的矩阵 $S^{sing}(f)$ 是一种特殊的广义 Sylvester 矩阵, 也就是 (4.2) 中的 $S(f_1, \dots, f_k)$, 输入的系数是有线性限制的。这里限制都是这种形式: $z_j = \lambda z_i$, 这里 λ 是整数。对于输入多项式的系数保持首一或者其它的性质比如稀疏性, 限制条件有如下形式: $z_j = \text{常数}$ 。

在文章[25, 小节 6.3]中我们发现算法[8; 39; 41] 可以允许加任意的线性限制到输入多项式的系数上。下面说明怎样来完成在有线性限制条件下用 STLN 方法得到最优解。

设 ζ 是结构问题的符号系数向量, 维数为 ν 。设 $\Gamma\zeta = \gamma$ 是一个作用在目标系数上的线性限制。这里我们不需要假设输入系数向量 $\mathbf{c}$ 满足: $\Gamma\mathbf{c} \approx \gamma$ 。利用高斯消去法构造一个线性系统 $\Gamma\zeta = \gamma - \Gamma\mathbf{c}$ 、 C , 一个向量 $\mathbf{d}$ 和一个自由参量的子向量

$[\zeta_{i_1}, \dots, \zeta_{i_\mu}]^{Tr}$ 使得

$$\zeta = C\zeta^- + \mathbf{d}, \quad \text{这里 } \zeta^- = \begin{bmatrix} \zeta_{i_1} \\ \vdots \\ \zeta_{i_\mu} \end{bmatrix}; \quad \text{而且 } \Gamma C = 0. \quad (4.12)$$

根据前面的初始值的方法,

$$\mathbf{z} = \mathbf{d} - C (H(\mathbf{v})C)^{Tr} (H(\mathbf{v})C (H(\mathbf{v})C)^{Tr})^{-1} S(\mathbf{c} + \mathbf{d})\mathbf{v},$$

这里 $\mathbf{v}$ 是矩阵 $S(\mathbf{c} + \mathbf{d})$ 的第一个奇异向量。同样, 有 $S(\mathbf{c} + \mathbf{z})\mathbf{v} = 0$ 。将 $\mathbf{v}$ 关于最大绝对值元素做正规化, 我们得到 $\mathbf{x}$ 的初始值(见(4.7))。迭代的更新公式 (4.9) 现在可以写成:

$$\min_{\Delta \mathbf{x}, \Delta \mathbf{z}^-} \left\| \begin{bmatrix} w(Y(\mathbf{x}) - P)C & wA(\mathbf{c} + \mathbf{z}) \\ C & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{z}^- \\ \Delta \mathbf{x} \end{bmatrix} + \begin{bmatrix} -w\mathbf{r}(\mathbf{z}, \mathbf{x}) \\ \mathbf{z} \end{bmatrix} \right\|. \quad (4.13)$$

新的系数值是 $\mathbf{c} + \mathbf{z} + \Delta \mathbf{z} = \mathbf{c} + \mathbf{z} + C\Delta \mathbf{z}^-$ 而且满足 $\Gamma(\mathbf{c} + \mathbf{z} + \Delta \mathbf{z}) = \gamma$, 这里假设 $\Gamma(\mathbf{c} + \mathbf{z}) = \gamma$ 。在整个迭代过程中, 初始值都能保证下面的式子成立: $\Gamma(\mathbf{c} + \mathbf{z}) = \Gamma(\mathbf{c} + \mathbf{d}) = \Gamma\mathbf{c} + (\gamma - \Gamma\mathbf{c}) = \gamma$ 。如果多项式系数是复数的情形, (4.13) 可以转化为实的优化问题:

$$\min_{\Delta \mathbf{x}_R, \Delta \mathbf{x}_I, \Delta \mathbf{z}_R^-, \Delta \mathbf{z}_I^-} \left\| M^- \begin{bmatrix} \Delta \mathbf{z}_R^- & \Delta \mathbf{z}_I^- & \Delta \mathbf{x}_R & \Delta \mathbf{x}_I \end{bmatrix}^{Tr} + \begin{bmatrix} -w\mathbf{r}_R(\mathbf{z}, \mathbf{x}) & -w\mathbf{r}_I(\mathbf{z}, \mathbf{x}) & \mathbf{z}_R & \mathbf{z}_I \end{bmatrix}^{Tr} \right\|$$

这里 M^- 是

$$\begin{bmatrix} w(Y(\mathbf{x}_R)C_R - Y(\mathbf{x}_I)C_I - PC_R) & -w(Y(\mathbf{x}_I)C_R + Y(\mathbf{x}_R)C_I - PC_I) & wA(\mathbf{c}_R + \mathbf{z}_R) & -wA(\mathbf{c}_I + \mathbf{z}_I) \\ w(Y(\mathbf{x}_I)C_R + Y(\mathbf{x}_R)C_I - PC_I) & w(Y(\mathbf{x}_R)C_R - Y(\mathbf{x}_I)C_I - PC_R) & wA(\mathbf{c}_I + \mathbf{z}_I) & wA(\mathbf{c}_R + \mathbf{z}_R) \\ C_R & -C_I & \mathbf{0} & \mathbf{0} \\ C_I & C_R & \mathbf{0} & \mathbf{0} \end{bmatrix},$$

而且线性限制被分成实部和虚部两部分: $\zeta_R + i\zeta_I = (C_R + iC_I)(\zeta_R^- + i\zeta_I^-) + \mathbf{d}_R + i\mathbf{d}_I$ 。

尽管 (4.3) 中的矩阵 $S_k^{sing}(f)$ 是 (4.2) 中的 $S_k(f_1, \dots, f_k)$ 的线性限制的形式, 实际上这两个优化问题并不相同。对于最近的有 k 重根的奇异多项式我们优化的是 $\|\Delta f\|$; 而在相对应的系数有线性限制的最大公因子问题中, 优化的是 $\sum \|d^i \Delta f / dy^i\|$, 这两个问题有不同的最小值。在文章[57]中已经提到最小问题 (4.5) 可以引入一个对角线的权矩阵 D , 也就是:

$$\min_{\Delta \mathbf{c} \in \mathbb{R}^\nu} \|D\Delta \mathbf{c}\| \text{ 或 } \min_{\Delta \mathbf{c} \in \mathbb{C}^\nu} \|D\Delta \mathbf{c}\| \quad \text{满足} \quad A(\mathbf{c} + \Delta \mathbf{c})\mathbf{x} = b(\mathbf{c} + \Delta \mathbf{c}), \text{ 对于某个向量 } \mathbf{x}.$$

那么, (4.13) 变为

$$\min_{\Delta \mathbf{x}, \Delta \mathbf{z}^-} \left\| \begin{bmatrix} w(Y(\mathbf{x}) - P)C & wA(\mathbf{c} + \mathbf{z}) \\ DC & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{z}^- \\ \Delta \mathbf{x} \end{bmatrix} + \begin{bmatrix} -w\mathbf{r}(\mathbf{z}, \mathbf{x}) \\ D\mathbf{z} \end{bmatrix} \right\|.$$

当矩阵的对角线中的元素是正实数时, 矩阵 D 可以取为对 GCD 问题中第一个多项式的扰动范数进行优化。这样求最近奇异值问题就转化为了系数有线性限制的 GCD 问题。对于一般复数上的 $D = D_R + \mathbf{i}D_I$, 上面给出的实数矩阵现在就变成:

$$M^- = \begin{bmatrix} M_{1,1}^- & M_{1,2}^- & wA(\mathbf{c}_R + \mathbf{z}_R) & -wA(\mathbf{c}_I + \mathbf{z}_I) \\ M_{2,1}^- & M_{2,2}^- & wA(\mathbf{c}_I + \mathbf{z}_I) & wA(\mathbf{c}_R + \mathbf{z}_R) \\ D_R C_R - D_I C_I & -D_R C_I - D_I C_R & \mathbf{0} & \mathbf{0} \\ D_R C_I + D_I C_R & D_R C_R - D_I C_I & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (4.14)$$

这里

$$\begin{aligned} M_{1,1}^- &= w(Y(\mathbf{x}_R)C_R - Y(\mathbf{x}_I)C_I - PC_R), & M_{1,2}^- &= -w(Y(\mathbf{x}_I)C_R + Y(\mathbf{x}_R)C_I - PC_I), \\ M_{2,1}^- &= w(Y(\mathbf{x}_I)C_R + Y(\mathbf{x}_R)C_I - PC_I), & M_{2,2}^- &= w(Y(\mathbf{x}_R)C_R - Y(\mathbf{x}_I)C_I - PC_R). \end{aligned}$$

对于上面的问题, 还有一个替代的方法: 直接给 (4.9) 加限制条件 $\Gamma(\mathbf{z} + \Delta\mathbf{z}) = \gamma$, 然后对下面的式子做迭代:

$$\min_{\Delta\mathbf{x}, \Delta\mathbf{z}} \left\| \begin{bmatrix} w(Y(\mathbf{x}) - P) & wA(\mathbf{c} + \mathbf{z}) \\ w\Gamma & \mathbf{0} \\ D & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta\mathbf{z} \\ \Delta\mathbf{x} \end{bmatrix} + \begin{bmatrix} -w\mathbf{r}(\mathbf{z}, \mathbf{x}) \\ w(\Gamma(\mathbf{c} + \mathbf{z}) - \gamma) \\ D\mathbf{z} \end{bmatrix} \right\|. \quad (4.15)$$

然而, 这个最小二乘问题的维数比 (4.13) 的维数大得多。在 [25, 小节 8.3] 中提到用线性规划的技巧可以用来解元素之间满足不等式线性限制的最小的 1-范数和 ∞ -范数, 这里不等式线性限制形如: $\Gamma(\mathbf{c} + \mathbf{z} + \Delta\mathbf{z}) \geq \gamma$ 。例如, 对于 ∞ -范数和实的输入和实的输出, 要满足线性限制的条件下计算最小问题 $\min_{\Delta\mathbf{z}} \|D(\mathbf{z} + \Delta\mathbf{z})\|_\infty$, 这里的线性限制是 $(Y(\mathbf{x}) - P)\Delta\mathbf{z} + A(\mathbf{c} + \mathbf{z})\Delta\mathbf{x} = \mathbf{r}(\mathbf{z}, \mathbf{x})$ 和 $\Gamma\Delta\mathbf{z} \geq \gamma - \Gamma(\mathbf{c} + \mathbf{z})$ 。关于 $\Delta\mathbf{z}, \Delta\mathbf{x}$ 和未知量 δ 的线性规划是:

$$\begin{aligned} &\text{最小: } \delta \\ &\text{满足: } \delta - D_i \Delta\mathbf{z} \geq D_i \mathbf{z}, \quad i = 1, 2, \dots \\ &\quad \delta + D_i \Delta\mathbf{z} \geq -D_i \mathbf{z}, \quad i = 1, 2, \dots \\ &\quad (Y(\mathbf{x}) - P)\Delta\mathbf{z} + A(\mathbf{c} + \mathbf{z})\Delta\mathbf{x} = \mathbf{r}(\mathbf{z}, \mathbf{x}) \\ &\quad \Gamma\Delta\mathbf{z} \geq \gamma - \Gamma(\mathbf{c} + \mathbf{z}) \end{aligned}$$

这里 D_i 是矩阵 D 的第 i 行。

§4.3.2 实的输入计算复的最优

在前面一章中的 3.2.1 已经举了一个例子说明了对于实的输入可能有复的最优解。我们发现上一章中的 STLN 迭代对于实的输入只能得到实的局部最优解, 并

没有给出一个方法求解这类问题。这一小节要说明怎样计算得到一个复的局部最优解。我们应用改变初始化的方法用来避免整个计算过程都是实的情形。

首先假设矩阵 $S(\mathbf{c} + \mathbf{i} \Delta \mathbf{c}_{rand})$ 的第一个奇异向量是 $\mathbf{v}$, $\Delta \mathbf{c}_{rand}$ 是一个任意的小实扰动向量, 计算得到向量 $\mathbf{z}$ 为:

$$\mathbf{z} = -H(\mathbf{v})^{Tr}(H(\mathbf{v})H(\mathbf{v})^{Tr})^{-1}S(\mathbf{c} + \mathbf{i} \Delta \mathbf{c}_{rand})\mathbf{v}.$$

通过对向量 $\mathbf{v}$ 关于最大绝对值元素做正规化, 得到向量 $\mathbf{x}$, 并且满足

$$A(\mathbf{c} + \mathbf{i} \Delta \mathbf{c}_{rand} + \mathbf{z})\mathbf{x} = b(\mathbf{c} + \mathbf{i} \Delta \mathbf{c}_{rand} + \mathbf{z}). \quad (4.16)$$

然而, 这里并不将 $\mathbf{i} \Delta \mathbf{c}_{rand}$ 加到实的优化问题 (4.10) 上去。

对应于有线性限制的情形时, 我们首先计算向量 $\mathbf{z}$:

$$\mathbf{z} = \mathbf{d} - C(H(\mathbf{v})C)^{Tr}(H(\mathbf{v})C(H(\mathbf{v})C)^{Tr})^{-1}S(\mathbf{c} + \mathbf{d} + \mathbf{i} \Delta \mathbf{c}_{rand})\mathbf{v},$$

这里 $\mathbf{v}$ 是矩阵 $S(\mathbf{c} + \mathbf{d} + \mathbf{i} \Delta \mathbf{c}_{rand})$ 的第一个奇异向量。通过对向量 $\mathbf{v}$ 关于最大绝对值元素做正规化, 我们得到向量 $\mathbf{x}$ (4.7)。那么我们有

$$A(\mathbf{c} + \mathbf{i} C \Delta \mathbf{z}_{rand}^- + \mathbf{z})\mathbf{x} = b(\mathbf{c} + \mathbf{i} C \Delta \mathbf{z}_{rand}^- + \mathbf{z}), \quad (4.17)$$

这里 $\Delta \mathbf{z}_{rand}^-$ 是一个很小的自由参数值的任意实向量。在小节 §4.4 中提到, $\mathbf{x}, \mathbf{z}$ 离复的最优解较远, 因此, 算法可能需要很多次迭代才能找到最优解。

§4.4 算法和实验

我们已经实现了 STLN 算法并解决了如下几个问题:

- 计算多个多项式的近似最大公因子。
- 计算最近的有 k 重根的单变元多项式。
- 计算多个多项式系数有线性限制的多项式的近似最大公因子问题。

例 4.4.1. 考虑两个多项式

$$f = y(y - \mathbf{i})^2 + 0.01 \quad \text{和} \quad g = (y + \mathbf{i})(y - \mathbf{i})^2 - 0.01\mathbf{i}. \quad (4.18)$$

计算最近的一对复多项式 $\tilde{f}$ 和 $\tilde{g}$ 使得这两个多项式有相同的 2 重根 [58]。上面的问题可以转化为四个多项式 $f, g, df/dy, dg/dy$ 的有线性限制的近似最大公因子问

题而且距离的定义为 $\|f - \tilde{f}\|_2^2 + \|g - \tilde{g}\|_2^2$ 。写出多项式的系数为一个维数为 14 的向量 ζ ，并且得到参数化的线性限制的矩阵

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \zeta^- = \begin{bmatrix} \zeta_1 \\ \vdots \\ \zeta_8 \end{bmatrix} \quad \text{和} \quad \mathbf{d} = \mathbf{0},$$

(见小节 §4.3.1)。权矩阵是 $D = \text{diag}(I_8, \mathbf{0}_6)$ 。用系数有线性限制且罚值 $w = 10^8$ 的 STLN 的算法来计算，然后用实的式子 (4.14) 来进行迭代。经过 6 次迭代，当 $\Delta \mathbf{x} = 0.974 \cdot 10^{-6} < 10^{-6}$ 时，停止迭代。我们得到满足条件的多项式：

$$\begin{aligned} \tilde{f} &= 0.0039 - 0.0018i + 0.0013y^2 - 2.0006iy^2 + 0.0037iy - 1.0006y + 0.0011iy^3 + 0.9982y^3, \\ \tilde{g} &= 0.0018 - 1.0066i + 0.0006y^2 - 1.0022iy^2 - 0.0006iy + 1.0028y - 0.0018iy^3 + 0.9984y^3. \end{aligned}$$

从向量 $\mathbf{x}$ 得到多项式的余因子 (互素部分)：

$$\begin{aligned} \tilde{u}_1 &= 0.0018 + 0.0039i + 0.9982y^2 + 0.0011iy^2 - 0.0013y - 0.9994iy, \\ \tilde{u}_2 &= 1.0036 + 0.0003i + 0.9984y^2 - 0.0018iy^2 + 0.0008999y - 0.00081iy. \end{aligned}$$

用第二章中的近似多项式除法，得到多项式的二重因子： $\tilde{h} = 0.001504 - 1.003015i + 1.000000y$ 。最后，我们计算向后误差是

$$\|f - \tilde{h}^2 \tilde{u}_1\|_2^2 + \|g - \tilde{h}^2 \tilde{u}_2\|_2^2 = 0.947 \cdot 10^{-4}.$$

作为比较，四个多项式 f , g , df/dy , dg/dy 没有线性限制的近似最大公因子是：

$$\bar{h} = -6.504887 \cdot 10^{-8} - 1.000011i + 1.000000y$$

上面的 GCD 是用 STLN 算法经过两次迭代计算出来的。尽管 $\bar{h}$ 根据 (4.18) 的构造看上去更接近准确的结构，但是计算 f 和 $\bar{h}^2$ 的近似商以及 g 和 $\bar{h}^2$ 的近似商，得到的向后误差却是 0.000140。

例 4.4.2. 考虑多项式

$$f = 1000y^{10} + y^3 - 1 \quad \text{和} \quad g = y^2 - \frac{1}{100}.$$

计算最近的一对多项式 $\tilde{f}$ 和 $\tilde{g}$ 使得它们有非平凡的公因子。首先给初始值加任意的小的复的扰动 (小节 §4.3.2)。平均下来用十次迭代, *STLN* 算法对于不同的初始值计算出下面这些局部最优解:

$$0.0421579, 0.0463113, 0.0474087, 0.0493292, \dots$$

在这些解中, 多项式

$$\begin{aligned} \tilde{f} &= 1000.0y^{10} + 0.0000147908y^9 + 0.0000297998y^8 + 0.0000604355y^7 + 0.000122287y^6 \\ &\quad + 0.000247491y^5 + 0.000500837y^4 + 1.00101y^3 + 0.00205103y^2 + 0.00415059y - 0.991601, \\ \tilde{g} &= 0.956139y^2 - 0.0887590y - 0.189618, \end{aligned}$$

有一个相同的因子 $y - 0.4941547$, 向后误差是

$$\|f - \tilde{f}\|_2^2 + \|g - \tilde{g}\|_2^2 = 0.0421579.$$

这个解与用全局方法[41; 25]求非首一的全局最小解相同。因此, *STLN* 算法也计算出全局的最优解。多项式 f 和 g 也可以说明一个事实: 一个有很大的非结构条件数的 *Sylvester* 矩阵可能有很小的结构条件数。设 S 是 f 和 g 的 *Sylvester* 矩阵, $\hat{S}$ 是离 S 在矩阵的 *Frobenius* 范数意义下的最近的奇异 *Sylvester* 矩阵, 这里 *Frobenius* 范数用 $\|\cdot\|_F$ 表示。首先, 假设 $\hat{S}$ 是对应于复多项式 $\hat{f}$ 和 $\hat{g}$ 的 *Sylvester* 矩阵, 多项式的次数分别为 10 和 2。因为 $\hat{S}$ 是奇异的, $\hat{f}$ 和 $\hat{g}$ 有一个相同的根。因此,

$$\begin{aligned} \|S - \hat{S}\|_F^2 &= 2\|f - \hat{f}\|_2^2 + 10\|g - \hat{g}\|_2^2 \\ &\geq 2\|f - \hat{f}\|_2^2 + 2\|g - \hat{g}\|_2^2 \\ &\geq 2\|f - \tilde{f}\|_2^2 + 2\|g - \tilde{g}\|_2^2 \geq 0.084315. \end{aligned}$$

上面的最小界可以用经典的不等式[71, 小节 4.2, 例13]推广到其它矩阵范数。最近的非结构的奇异多项式 $\hat{A}$ 有: $\|S - \hat{A}\|_F^2$ 等于 S 最小奇异值的平方 [12; 71, 定理 6.7], 也就是 0.000000098975。对于其它矩阵范数, 类似的值可以计算[29, p. 775]。离奇异矩阵的结构的和非结构的距离的这种差异是不可能出现在 *Toeplitz* 矩阵上[62]。□

在这三个表中, 所有的实验是在 Pentium 4 频率为 2.0 Ghz 的机器上计算的。我们在 Maple 10 中假设 *Digits* = 14 应用 *STLN* 算法计算近似最大公因子和最近的奇异多项式。

<i>Ex.</i>	<i>m</i>	<i>k</i>	<i>it.</i> (<i>STLN</i>)	<i>error</i> (<i>ZNKW</i>)	<i>error</i> (<i>STLN</i>)
1	4	2	12	.1763296120	.1763296118
		3	34	.6261127476	.6261127498
2	4	2	4	.1552760123e-12	.1552723415e-12
		3	11	.8834609009e-9	.9814886696e-9
		4	4	.2021848972e-4	.1958553174e-4
3	4	2	4	.1645037985e-10	.16450617515e-10
		3	4	.4144531274e-6	.4144531274e-6
		4	12	.1049993144	.1049993152
4	5	2	1	.2460987981e-8	.2461467456e-8
	5	3	20	.3681785214	.3681784856
5	6	2	2	.3231668276e-5	.3231668277e-5
6	6	2	3	.3009788845e-11	.3009789157e-11
		3	3	.7453849284e-6	.7453849284e-6
		4	24	.4449023547	.4449023547
7	5	2	8	.8565349347	.8565327605
8	21	2	2	.190477e-8	.1893347157e-8
		3	6	.963776e-4	.9637591989e-4

表 4.1 单变元奇异多项式的实验结果

在第 56 页的表 4.1 中, 用 STLN 的算法计算了最近的首一的奇异多项式。这里 m 表示多项式次数; k 表示根的重数; 而 $it. (STLN)$ 表示用 STLN 算法所需的迭代次数; $error (STLN)$ 和 $error (ZNKW)$ 分别表示用 STLN 算法和文章[80] 中的算法计算的最小扰动 $\|\tilde{f} - f\|_2^2$, 使得多项式 $\tilde{f}$ 有 k 重根。

例 1 和例 2 是文章[81]中的例子。例 1 是一个实多项式。对于 $k = 2, 3$ 时, 根据多项式的零点性质, 多项式有四个离得最近的有 k 重根的奇异多项式。例 2 是系数为复的多项式。例 3 到例 8 是文章[80] 中的例子。有一点需要注意: 文章[81]中表 2 和 5 中的计算的最小值和文章[80] 中的例 5 的最小值都是有错误的。这里用文章[81; 80] 的原始 Maple 程序更正了这些最小值。除了例 7, 多项式的系数都是实的。实际上, 我们用 STLN 算法 (基于第 47 页的(4.3)得到的矩阵 S_k^{sing} 和基于线性限制的近似最大公因子) 能得到与 ZNKW 算法一样的向后误差。

在第 57 页中的表 4.2, 给出了用 STLN 算法计算多变元的近似最大公因子的

<i>Ex.</i>	m_i	k	e	<i>it.</i> (<i>STLN</i>)	<i>error</i> (<i>Zeng</i>)	<i>error</i> (<i>AMVGCD</i>)	<i>error</i> (<i>STLN</i>)
1	7,7	4	3	2	2.44360e-4	2.59476e-4	6.50358e-5
2	7,7	4	5	1	2.44404e-8	2.59194e-8	6.50357e-9
3	7,7	4	7	1	2.44405e-12	2.59191e-12	6.50357e-13
4	7,7	4	9	1	2.44396e-16	2.59187e-16	6.50361e-17
5	6,6	3	2	3	2.26617	1.49524	4.80154e-1
6	10,10	5	4	2		2.74672e-3	1.84914e-3
7	8,8	4	5	2	7.09371e-5	2.38059e-5	2.01393e-5
8	40,40	30	5	2	1.39858e-3	4.83931e-4	4.39489e-4
9	10,9,8	5	3	2			6.21772e-2
10	8,7,8,6	4	5	2			4.04458e-6

表 4.2 多变元的近似最大公因子实验结果

	$j = 2$	$j = 3$	$j = 4$	$j = 5$	$j = 6$	$j = 7$	$j = 8$
<i>it.</i> (<i>STLN</i>)	1	1	1	1	3	4	5
<i>error</i> (<i>STLN</i>)	3.12e-14	2.48e-12	3.33e-10	3.39e-8	3.76e-6	2.53e-4	2.93e-2

表 4.3 四个变元的多项式的线性限制的实验结果[76]

结果。这里 m_i 表示多项式的全次数； k 表示近似最大公因子的全次数；而 *error* (*Zeng*), *error* (*AMVGCD*) 和 *error* (*STLN*) 分别表示用[76], 算法 2.2.1和 *STLN* 算法计算的最小扰动 $\sum_i \|\tilde{f}_i - f_i\|_2^2$ 。

表 4.2 中的例 1 到例 4 是[76] 中的例 4 加不同的扰动得到的例子。

例 5 和例 7-10是这样构造的：取多项式系数为 $-25 \leq c \leq 25$ 内的整数，而且公因子次数是给定的整数 k ，然后任意次数与给定多项式的次数相同的系数在 $[-10^e, 10^e]$ 范围内的扰动多项式；给扰动多项式乘上一个比例常数使它们和给定多项式的相对误差是 10^{-e} 。

例 6 是复系数的多项式，它的实部和虚部的系数的范围为 $-25 \leq c \leq 25$ 。扰动跟上面说的一样分别对多项式的实部和虚部加以扰动。

在第 57页的表 4.3中，给出了用 *STLN* 算法计算近似有 2 重线性因子的一对多项式 f_1, f_2 ， f_1, f_2 是 4 个变元和次数为 4 的多项式（见[76]的例 5）。对于这种情形，可以计算 $f_1, \partial f_1, f_2, \partial f_2$ 的近似公因子而且要加上线性限制条件，这些限制条

件是分别由 $f_1, \partial f_1$ 和 $f_2, \partial f_2$ 的系数关系得到的。这里 ∂f_i 是多项式 f_i 关于其中一个变元的偏导数。加的扰动阶数是 10^{j-10} 。

§4.5 结论

在这一章里，我们用结构最小二乘范数方法解决了多项式系数有线性限制的近似最大公因子问题。当输入的多项式的相对误差不大于 10^{-2} 时，实验说明基于 STLN 的算法能很快得到最优解，而且迭代次数一般不超过 10 次。然而，对于近似最大公因子和最近的奇异多项式问题，有两种比较特殊的情形：第一，实的输入要得到复的最优解；第二，对于有线性限制条件时，输入的多项式离这些线性限制的条件有一定距离。我们知道这些情形对于所有的迭代算法都是比较难以处理的。然而，基于 STLN 的算法能计算最优解。而且，我们给出了一个新的方法来取比较好的初始点，但是在一些比较困难的情形，这种算法还是需要很多次迭代才能找到最优解。

这一章用的算法是基于罚函数的 STLN 方法。我们也可用其他的结构最小二乘方法 CTLS[48] 和 Riemannian SVD[43] 解决我们的问题，我们已经把 CTLS 方法应用到近似最大公因子问题。初步实验已经说明了 CTLS 方法同样可以计算全局最优的解。

第五章 结论与展望

我们给出了有效的算法和 Maple 程序计算多变元多项式的因式分解。跟别人的算法相比,我们的算法能处理一些非常困难的情况,例如近似的多项式离有准确因式分解的多项式的距离特别远的情形。

我们用结构最小二乘方法 (STLN) 对 (广义) Sylvester 矩阵低秩逼近,从而设计了新的计算多项式 (单变元和多变元) 的近似最大公因子算法。跟基于 SVD 的最大公因子算法相比,这种算法更稳定而且能得到更小的向后误差。

本文还解决了怎样求系数有线性限制的多项式近似最大公因子问题,这里包括单变元多项式和多变元多项式的情形。这类问题有重要的应用:对于单变元多项式情形,我们把基于 STLN 算法的最大公因子算法应用到计算最近的有 k 重根的奇异多项式上。跟以前的算法相比,STLN 算法也能得到全局最优解;而且运算量要小很多。在多变元多项式情形,可以用 STLN 算法计算最近的有 k 重因子的多元多项式。这在计算近似因式分解算法中有非常重要的用途。

今后我们的工作可以从以下几个方面来考虑。

1. Ruppert 矩阵也是结构矩阵。因此,可以将系数有线性限制的结构最小二乘方法应用到近似因式分解和相关的问题上。在进行多项式近似因式分解过程中,我们已经实现了用 CTLS 和 STLN 方法对 Ruppert 矩阵进行结构低秩逼近,但是结果并不理想。希望在以后的工作能更好解决这一问题。
2. 本文计算近似最大公因子问题用的方法是基于罚函数的 STLN 算法。同样近似最大公因子问题还可以用基于 Lagrange 乘子的最小二乘 (CTLS) 方法去解决。初步的实验说明了 CTLS 方法同样能计算最优的全局解。
3. 在本文中,我们用的都是线性代数经典的算法计算最小二乘问题。由于算法中所涉及的矩阵都有结构性,所以把快速算法和 STLN 算法结合会大大减少计算复杂度。对于单变元多项式的近似最大公因子问题,根据 Sylvester 的位移结构,我们已经应用了快速算法。这样不但减少了整个算法的复杂度,而且没有丢失算法的准确性。由于广义 Sylvester 矩阵和 Ruppert 矩阵非常复杂,现在还没有找到有效的矩阵位移算子。希望以后能把快速算法应用于多变元多项式的近似最大公因子和因式分解问题上。

参考文献

- [1] ANDA, A. A., AND PARK, H. Fast plane with dynamic scaling. *SIAM J. Matrix Anal. Applic.* 15 (1994), 162–174.
- [2] ANDA, A. A., AND PARK, H. Self-scaling fast rotations for stiff and equality-constrained linear least squares problems. *Linear Algebra and Applications* 234 (1996), 137–161.
- [3] BECKERMANN, B., AND LABAHN, G. When are two polynomials relatively prime? *J. Symbolic Comput.* 26 (1998), 677–689.
- [4] CHÈZE, G., AND GALLIGO, A. From an approximate to an exact absolute polynomial factorization. Paper submitted, 22 pages, 2003.
- [5] CHU, M. T., FUNDERLIC, R. E., AND PLEMMONS, R. J. Structured low rank approximation. *Linear Algebra and Applications* 366 (2003), 157–172.
- [6] CORLESS, R. M., GALLIGO, A., KOTSIREAS, I. S., AND WATT, S. M. A geometric-numeric algorithm for absolute factorization of multivariate polynomials. In Mora [49], pp. 37–45.
- [7] CORLESS, R. M., GIANNI, P. M., TRAGER, B. M., AND WATT, S. M. The singular value decomposition for polynomial systems. In Levelt [45], pp. 96–103.
- [8] CORLESS, R. M., GIANNI, P. M., TRAGER, B. M., AND WATT, S. M. The singular value decomposition for polynomial systems. In Levelt [45], pp. 96–103.
- [9] CORLESS, R. M., GIESBRECHT, M. W., VAN HOEIJ, M., KOTSIREAS, I. S., AND WATT, S. M. Towards factoring bivariate approximate polynomials. In Mourrain [50], pp. 85–92.
- [10] CORLESS, R. M., WATT, S. M., AND ZHI, L. QR factoring to compute the GCD of univariate approximate polynomials. *IEEE Transactions on Signal Processing* 52 (Dec. 2004), 3394–3402.
- [11] DEMMEL, J. W. On condition numbers and the distance to the nearest ill-posed problem. *Numer. Math.* 51 (1987), 251–289.
- [12] ECKART, C., AND YOUNG, G. The approximation of one matrix by another of lower rank. *Psychometrika* 1, 3 (Sept. 1936), 211–218.

- [13] EMIRIS, I. Z., GALLIGO, A., AND LOMBARDI, H. Certified approximate univariate GCDs. *J. Pure Applied Algebra*. 117 & 118 (May 1996), 229–251. Special Issue on Algorithms for Algebra.
- [14] GALLIGO, A., AND RUPPRECHT, D. Semi-numerical determination of irreducible branches of a reduced space curve. In Mourrain [50], pp. 137–142.
- [15] GALLIGO, A., AND RUPPRECHT, D. Irreducible decomposition of curves. *J. Symbolic Comput.* 33, 5 (2002), 661–677.
- [16] GAO, S. Factoring multivariate polynomials via partial differential equations. *Math. Comput.* 72, 242 (2003), 801–822.
- [17] GAO, S., KALTOFEN, E., MAY, J. P., YANG, Z., AND ZHI, L. Approximate factorization of multivariate polynomials via differential equations. In Gutierrez [24], pp. 167–174. ACM SIGSAM’s ISSAC 2004 Distinguished Student Author Award (May and Yang).
- [18] GAO, S., KALTOFEN, E., MAY, J. P., YANG, Z., AND ZHI, L. Approximate factorization of multivariate polynomials via differential equations. In Gutierrez [24], pp. 167–174.
- [19] VON ZUR GATHEN, J., AND GERHARD, J. *Modern Computer Algebra*. Cambridge University Press, Cambridge, New York, Melbourne, 1999. Second edition 2003.
- [20] GELFOND, A. O. Transcendental and algebraic numbers. *Translated from the 1st Russian ed. by Leo F. Boron*. Dover, New York, 1960.
- [21] GIESBRECHT, M., LABAHN, G., AND LEE, W.-S. Symbolic-numeric sparse interpolation of multivariate polynomials (extended abstract). In *Proc. Ninth Rhine Workshop on Computer Algebra (RWCA’04), University of Nijmegen, the Netherlands* (2004), pp. 127–139. Full paper under preparation.
- [22] GOLUB, G. H., AND VAN LOAN, C. F. *Matrix Computations*, third ed. Johns Hopkins University Press, Baltimore, Maryland, 1996.
- [23] GRABMEIER, J., KALTOFEN, E., AND WEISPFENNING, V. *Computer Algebra Handbook*. Springer Press, 2000.
- [24] GUTIERREZ, J., Ed. *ISSAC 2004 Proc. 2004 Internat. Symp. Symbolic Algebraic Comput.* (New York, N. Y., 2004), ACM Press.

- [25] HITZ, M. A., AND KALTOFEN, E. Efficient algorithms for computing the nearest polynomial with constrained roots. In *Proc. 1998 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'98)* (New York, N. Y., 1998), O. Gloor, Ed., ACM Press, pp. 236–243.
- [26] HITZ, M. A., KALTOFEN, E., AND LAKSHMAN Y. N. Efficient algorithms for computing the nearest polynomial with a real root and related problems. In *Proc. 1999 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'99)* (New York, N. Y., 1999), S. Dooley, Ed., ACM Press, pp. 205–212.
- [27] HRIBERNIG, V., AND STETTER, H. J. Detection and validation of clusters of polynomials zeros. *J. Symbolic Comput.* 24 (1997), 667–681.
- [28] HUANG, Y., STETTER, H. J., WU, W., AND ZHI, L. Pseudofactors of multivariate polynomials. In *Internat. Symp. Symbolic Algebraic Comput. ISSAC 2000 Proc. 2000 Internat. Symp. Symbolic Algebraic Comput.* (New York, N. Y., 2000), C. Traverso, Ed., ACM Press, pp. 161–168.
- [29] KAHAN, W. Numerical linear algebra. *Canadian Math. Bull.* 9 (1966), 757–801.
- [30] KAILATH, T., AND SAYED, A. H. Displacement structure: theory and applications. *SIAM Review* 37, 3 (1995), 297–386.
- [31] KALTOFEN, E. Fast parallel absolute irreducibility testing. *J. Symbolic Comput.* 1, 1 (1985), 57–67. Misprint corrections: *J. Symbolic Comput.* vol. 9, p. 320 (1989).
- [32] KALTOFEN, E. Polynomial factorization 1987-1991. In I.SIMON(editor),*Proc. LATIN'92*, volumn 583 of *Lect. Notes Comput. Sci.*, Pages 294-313. Springer Verlag, Heidelberg, Germany.
- [33] KALTOFEN, E. Challenges of symbolic computation my favorite open problems. *J. Symbolic Comput.* 29, 6 (2000), 891–919. With an additional open problem by R. M. Corless and D. J. Jeffrey.
- [34] KALTOFEN, E., AND MAY, J. On approximate irreducibility of polynomials in several variables. In Sendra [68], pp. 161–168.
- [35] KALTOFEN, E., MAY, J., YANG, Z., AND ZHI, L. Approximate factorization of multivariate polynomials using singular value decomposition. Manuscript, 22 pages, Dec. 2005.

- [36] KALTOFEN, E., YANG, Z., AND ZHI, L. Structured low rank approximation of a Sylvester matrix. Manuscript, 15 pages, Oct. 2005. Preliminary version in SNC 2005 Proceedings, Dongming Wang and Lihong Zhi eds., pp. 188–201, distributed at the International Workshop on Symbolic-Numeric Computation in Xi'an, China, July 19–21, 2005.
- [37] KALTOFEN, E., YANG, Z., AND ZHI, L. Structured low rank approximation of a generalized Sylvester matrix. Manuscript, 2005. 14 pages. Abstract at the Asian Symposium on Computer Mathematics ASCM 2005, 4 pages.
- [38] KARMARKAR, N., AND LAKSHMAN Y. N. Approximate polynomial greatest common divisors and nearest singular polynomials. In Lakshman Y. N. [42], pp. 35–42.
- [39] KARMARKAR, N., AND LAKSHMAN Y. N. Approximate polynomial greatest common divisors and nearest singular polynomials. In Lakshman Y. N. [42], pp. 35–42.
- [40] KARMARKAR, N. K., AND LAKSHMAN Y. N. On approximate GCDs of univariate polynomials. *J. Symbolic Comput.* 26 (1998), 653–666. Special issue of the JSC on Symbolic Numeric Algebra for Polynomials S. M. Watt and H. J. Stetter, editors.
- [41] KARMARKAR, N. K., AND LAKSHMAN Y. N. On approximate GCDs of univariate polynomials. *J. Symbolic Comput.* 26, 6 (1998), 653–666. Special issue on Symbolic Numeric Algebra for Polynomials S. M. Watt and H. J. Stetter, editors.
- [42] LAKSHMAN Y. N., Ed. *ISSAC 96 Proc. 1996 Internat. Symp. Symbolic Algebraic Comput.* (New York, N. Y., 1996), ACM Press.
- [43] LEMMERLING, P. Structured total least squares: analysis, algorithms and applications. Dissertation, Katholieke Universiteit Leuven, Belgium, 1999.
- [44] LEMMERLING, P., MASTRONARDI, N., AND VAN HUFFEL, S. Fast algorithm for solving the Hankel/Toeplitz structured total least squares problem. *Numerical Algorithms* 23 (2000), 371–392.
- [45] LEVELT, A. H. M., Ed. *Proc. 1995 Internat. Symp. Symbolic Algebraic Comput. ISSAC'95* (New York, N. Y., 1995), ACM Press.
- [46] LI, B., YANG, Z., AND ZHI, L. Fast low rank approximation of a Sylvester matrix by structured total least norm. *J. JSSAC (Japan Society for Symbolic and Algebraic Computation)* 11, 3,4 (2005), 165–174.

- [47] LI, T. Y., AND ZENG, Z. A rank-revealing method and its application. Manuscript available at <http://www.neiu.edu/~zzeng/papers.htm>, 2003.
- [48] MASTRONARDI, N., LEMMERLING, P., AND VAN HUFFEL, S. Fast structured total least squares algorithm for solving the basic deconvolution problem. *SIAM J. Matrix Anal. Applic.* 22, 2 (2000), 533–553.
- [49] MORA, T., Ed. *ISSAC 2002 Proc. 2002 Internat. Symp. Symbolic Algebraic Comput.* (New York, N. Y., 2002), ACM Press.
- [50] MOURRAIN, B., Ed. *ISSAC 2001 Proc. 2001 Internat. Symp. Symbolic Algebraic Comput.* (New York, N. Y., 2001), ACM Press.
- [51] May, John Paul. Approximate factorization of polynomials in many variables and other problems in approximate algebra via singular value decomposition methods. PhD thesis, North Carolina State Univ., Raleigh, North Carolina, August 2005.
- [52] NAGASAKA, K. Towards certified irreducibility testing of bivariate approximate polynomials. In Mora [49], pp. 192–199.
- [53] NAGASAKA, K. Towards more accurate separation bounds of empirical polynomials II. In: Ganzha, V. G., Mayr, E. W., Vorozhtsov, E. V. (Eds.), *Computer Algebra in Scientific Computing, 8th International Workshop, CASC 2005*, Kalamata, Greece, September 12-16, 2005, Proceedings. Vol. 3718 of *Lecture Notes Comput. Sci.* Springer Verlag, Heidelberg, Germany, pp. 318–329.
- [54] NODA, M. T., AND SASAKI, T. Approximate GCD and its application to ill-conditioned algebraic equations. *J. Comput. Appl. Math.* 38 (1991), 335–351.
- [55] OCHI, M., NODA, M. A., AND SASAKI, T. Approximate greatest common divisor of multivariate polynomials and its application to ill-conditioned system of algebraic equations. *J. Inf. Process* 12 (1991), 292–300.
- [56] PAN, V. Y. Numerical computation of a polynomial GCD and extensions. *Information and computation* 167 (2001), 71–85.
- [57] PARK, H., ZHANG, L., AND ROSEN, J. B. Low rank approximation of a Hankel matrix by structured total least norm. *BIT* 39, 4 (1999), 757–779.

- [58] POPE, S., AND SZANTO, A. Nearest multivariate system with given root multiplicities. Manuscript available at <http://www.math.ncsu.edu/~aszanto/papers.html>, 2005.
- [59] ROBERT, E., CHU, M. T., AND ROBERT.PLEMMONS, J. Structured low rank approximation. Manuscript, 1998.
- [60] ROSEN, J. B., PARK, H., AND GLICK, J. Total least norm formulation and solution for structured problems. *SIAM J. Matrix Anal. Applic.* 17, 1 (1996), 110–128.
- [61] ROSEN, J. B., PARK, H., AND GLICK, J. Structured total least norm for nonlinear problems. *SIAM J. Matrix Anal. Applic.* 20, 1 (1999), 14–30.
- [62] RUMP, S. M. Structured perturbations part I: Normwise distances. *SIAM J. Matrix Anal. Applic.* 25, 1 (2003), 1–30.
- [63] RUPPERT, W. M. Reducibility of polynomials $f(x,y)$ modulo p . *J. Number Theory* 77 (1999), 62–70.
- [64] RUPPRECHT, D. Semi-numerical absolute factorization of polynomials with integer coefficients. *J. Symbolic Comput.* 37, 5 (2004), 557–574.
- [65] SASAKI, M., AND SASAKI, T. Polynomial remainder sequence and approximate GCD. *ACM SIGSAM Bulletin* 31 (2001), 4–10.
- [66] SASAKI, T. Approximate multivariate polynomial factorization based on zero-sum relations. In Mourrain [50], pp. 284–291.
- [67] SCHÖNHAGE, A. Quasi-gcd computations. *Journal of Complexity* 1 (1985), 118–137.
- [68] SENDRA, J. R., Ed. *ISSAC 2003 Proc. 2003 Internat. Symp. Symbolic Algebraic Comput.* (New York, N. Y., 2003), ACM Press.
- [69] SOMMESE, A. J., VERSCHELDE, J., AND WAMPLER, C. W. Numerical factorization of multivariate complex polynomials. *Theoretical Comput. Sci.* 315, 2–3 (2004), 651–669. Special issue on Algebraic and Numerical Algorithms.
- [70] STEWART, G. W. Error and perturbation bounds for subspaces associated with certain eigenvalue problems. *SIAM Review* 15, 4 (Oct. 1973), 727–764.
- [71] STEWART, G. W. *Introduction to Matrix Computations*. Academic Press, Inc., New York, 1973.

- [72] STEWART, G. W. On the asymptotic behavior of scaled singular value and QR decompositions. *Mathematics of Computations* 43 (1983), 488–489.
- [73] ZENG, Z. A method computing multiple roots of inexact polynomials. In Sendra [68], pp. 266–272.
- [74] ZENG, Z. The approximate GCD of inexact polynomials. part I: a univariate algorithm. Manuscript, 2004.
- [75] ZENG, Z. Computing multiple roots of inexact polynomials. *Math. Comput.* 74 (2005), 869–903.
- [76] ZENG, Z., AND DAYTON, B. H. The approximate GCD of inexact polynomials part II: a multivariate algorithm. In Gutierrez [24], pp. 320–327.
- [77] ZHI, L. Displacement structure in computing approximate GCD of univariate polynomials. In *Proc. Sixth Asian Symposium on Computer Mathematics (ASCM 2003)* (Singapore, 2003), Z. Li and W. Sit, Eds., vol. 10 of *Lecture Notes Series on Computing*, World Scientific, pp. 288–298.
- [78] ZHI, L. Displacement structure in computing approximate GCD of univariate polynomials. In *Proc. Sixth Asian Symposium on Computer Mathematics (ASCM 2003)* (Singapore, 2003), Z. Li and W. Sit, Eds., vol. 10 of *Lecture Notes Series on Computing*, World Scientific, pp. 288–298.
- [79] ZHI, L., AND NODA, M. T. Approximate GCD of multivariate polynomials. In *Proc. Fourth Asian Symposium on Computer Mathematics (ASCM 2000)* (Singapore, 2000), X. Gao and D. Wang, Eds., vol. 8 of *Lecture Notes Series on Computing*, World Scientific, pp. 9–18.
- [80] ZHI, L., NODA, M.-T., KAI, H., AND WU, W. Hybrid method for computing the nearest singular polynomials. *Japan J. Industrial and Applied Math.* 21, 2 (June 2004), 149–162.
- [81] ZHI, L., AND WU, W. Nearest singular polynomial. *J. Symbolic Comput.* 26, 6 (1998), 667–675. Special issue on Symbolic Numeric Algebra for Polynomials S. M. Watt and H. J. Stetter, editors.

发表文章目录

- [1] Shuhong Gao, Erich Kaltofen, John P. May, Zhengfeng Yang and Lihong Zhi, Approximate factorization of multivariate polynomials via differential equations. In ISSAC'04, 167–174, ACM Press, New York, 2004. (EI)
- [2] Bingyu Li, Zhengfeng Yang and Lihong Zhi, Fast low rank approximation of a Sylvester matrix by structured total least norm, J. JSSAC (Japan Society for Symbolic and Algebraic Computation), Vol. 11, No. 3-4, 165-174, 2005.
- [3] Erich Kaltofen, John P. May, Zhengfeng Yang and Lihong Zhi, Approximate factorization of multivariate polynomials using singular value decomposition. 22pages, submitted to Journal of Symbolic Computation.
- [4] Erich Kaltofen, Zhengfeng Yang and Lihong Zhi, Structured low rank approximation of a Sylvester matrix. accepted by the book "Symbolic-Numeric Computation", to be published by Birkhäuser, Basel Boston, 2006.
- [5] Erich Kaltofen, Zhengfeng Yang and Lihong Zhi, Structured low rank approximation of a generalized Sylvester matrix. Manuscript, 14 pages. Abstract at the Asian Symposium on Computer Mathematics ASCM 2005, 4 pages.
- [6] Erich Kaltofen, Zhengfeng Yang and Lihong Zhi, Approximate greatest common divisors of several polynomials with linearly constrained coefficients and singular polynomials, accepted by ISSAC'06. (EI)

致 谢

在论文的最后，我想衷心感谢所有的老师和同学五年来的鼓励和帮助！

首先要感谢我的导师吴文俊院士与支丽红副研究员。吴文俊院士八十多的高龄依然精力充沛，思维敏捷。他所倡导的构造性方法是本论文写作的指导思想和基础。回首论文的研究期间，最要感谢的人是支丽红老师。本论文从选题到最后的写作始终是在支老师的悉心指导下完成的。五年来，支老师不断的鼓励和教诲无论是在学习上还是在生活上都使我受益良多。她活跃的思维、崇高的敬业精神和严谨的治学态度深深的影响着我。在她的辛勤指导和无私的帮助下，我才能顺利完成本文。值此论文完成之际，再次感谢支老师对我的大力支持和鼓励。

感谢 Erich Kaltofen 教授邀请我访问 North Carolina State 大学并在我访问期间提供了很多的帮助；在和他合作过程中，我从他那里学到了很多新的知识。感谢数学机械化研究中心的高小山研究员、李洪波研究员、刘卓军研究员、石赫研究员、李子明研究员、杜宏副研究员、王定康副研究员、马玉杰助理研究员、闫振亚助理研究员、冯如勇助理研究员，从他们那里我获得了很多启发和热心的帮助。感谢周代珍老师，王莎莎女士平时对我的诸多帮助和照顾。

感谢数学机械化研究中心的其他各位同学以及同届的好友，大家在一起度过的很多愉快的时光给我留下了许多美好的回忆。

感谢数学与系统科学研究院的各位老师，他们的辛勤工作为我们创造了良好的学习与生活的环境。

最后感谢我的家人，感谢父母对我的不断的教诲和默默的奉献。他们的淳朴和善良深深的影响了我。感谢我的妻子许敏，感谢她对我的学业给予了极大的鼓励与支持，并一直陪伴我度过这段充实而又快乐的时光。

于数学与系统科学研究院
杨争峰