Fast Low Rank Approximation of a Sylvester Matrix by Structured Total Least Norm

Bingyu Li, Zhengfeng Yang, Lihong Zhi
Institute of Systems Science, AMSS,
Academia Sinica
Beijing 100080, China
email: {liby, zyang, lzhi}@mmrc.iss.ac.cn

Abstract

The problem of approximating the greatest common divisor(GCD) for polynomials with inexact coefficients can be formulated as a low rank approximation problem with a Sylvester matrix. In this paper, we present an algorithm based on fast Structured Total Least Norm(STLN) for constructing a Sylvester matrix of given lower rank and obtaining the nearest perturbed polynomials with exact GCD of given degree.

1 Introduction

Approximate GCD computation of univariate polynomials has been studied by many authors [24, 20, 8, 11, 4, 2, 15, 21, 25, 9, 28]. Particularly, in [8, 11, 25, 9, 28], Singular Value Decomposition(SVD) of the Sylvester matrix has been used to obtain a degree upper bound of approximate GCDs and furthermore to obtain an approximate GCD. However, SVD computation is not appropriate for computing the minimal distance to the structured low rank matrix.

In [10, 30], authors described an algorithm based on STLN [22] for constructing a Sylvester matrix of given lower rank and obtaining the nearest perturbed polynomials with exact GCD of given degree. For their algorithm, the overall computation time depends on solving a sequence least squares(LS) problems. In the present paper, based on the displacement structure of the coefficient matrices, we describe a fast algorithm using the generalized Schur algorithm [5, 6] for solving the LS problems and deriving a fast version of the approximate GCD algorithm.

*The work is Partially supported by a National Key Basic Research Project of China 2004CB318000 and Chinese National Science Foundation under Grant 10401035.

The organization of this paper is as follows. In Section 2, we briefly introduce equivalence between low rank approximation of a Sylvester matrix [10, 30] and solving a minimization problem with structured coefficient matrix. In Section 3, we propose a fast method based on the generalized Schur algorithm for solving the minimization problem efficiently. Finally, in Section 4, we present a fast version of the algorithm [10, 30] for computing approximate GCD of univariate polynomials.

2 Preliminaries

Given two polynomials a, $b \in \mathbb{C}[x]$ with $a = a_m x^m + \cdots + a_1 x + a_0$ and $b = b_n x^n + \cdots + b_1 x + b_0$, the Sylvester matrix for a and b is:

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

Denoting the perturbations of a and b by $\Delta a = \Delta a_m x^m + \cdots + \Delta a_1 x + \Delta a_0$, $\Delta b = \Delta b_n x^n + \cdots + \Delta b_1 x + \Delta b_0$ respectively, we consider the minimal perturbation problem: *minimize* $\|\Delta a\|_2^2 + \|\Delta b\|_2^2$ *preserving that* $a + \Delta a$ *and* $b + \Delta b$ *have an exact GCD of a given degree.*

Let us denote $S_k = [\mathbf{a}\ A_k]$ as the k-th Sylvester matrix,

$$S_k = \begin{pmatrix} a_m & 0 & \cdots & 0 & 0 & b_n & 0 & \cdots & 0 & 0 \\ a_{m-1} & a_m & \cdots & 0 & 0 & b_{n-1} & b_n & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & a_0 & a_1 & 0 & 0 & \cdots & b_0 & b_1 \\ 0 & 0 & \cdots & 0 & a_0 & 0 & 0 & \cdots & 0 & b_0 \end{pmatrix},$$
$$\underbrace{\qquad\qquad\qquad\qquad}_{n-k+1}\quad\underbrace{\qquad\qquad\qquad\qquad}_{m-k+1}$$

where $\mathbf{a}$ is the first column of S_k.

EXAMPLE 2.1 *[30] Suppose $m = n = 3$, $k = 2$, then $S_2 = [\mathbf{a}\ A_2]$,*

$$\mathbf{a} = \begin{pmatrix} a_3 \\ a_2 \\ a_1 \\ a_0 \\ 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & b_3 & 0 \\ a_3 & b_2 & b_3 \\ a_2 & b_1 & b_2 \\ a_1 & b_0 & b_1 \\ a_0 & 0 & b_0 \end{pmatrix}.$$

For simplicity, we express the perturbations Δa and Δb by a $m+n+2$-dimensional vector $\mathbf{d}$,

$$\mathbf{d} = (d_1\,, d_2\,, \cdots\,, d_{m+n+1}\,, d_{m+n+2})^T,$$

and represent the k-th Sylvester structured perturbation of S_k as

$$[\Delta\mathbf{a}\, D_k] = \left(\begin{array}{ccccccccccc} d_1 & 0 & \cdots & 0 & 0 & d_{m+2} & 0 & \cdots & 0 & 0 \\ d_2 & d_1 & \cdots & 0 & 0 & d_{m+3} & d_{m+2} & \cdots & 0 & 0 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ 0 & 0 & \cdots & d_{m+1} & d_m & 0 & 0 & \cdots & d_{m+n+2} & d_{m+n+1} \\ 0 & 0 & \cdots & 0 & d_{m+1} & 0 & 0 & \cdots & 0 & d_{m+n+2} \end{array}\right),$$
$$\underbrace{\qquad\qquad\qquad}_{n-k+1}\qquad\underbrace{\qquad\qquad\qquad}_{m-k+1}$$

where $\Delta\mathbf{a}$ is the first column of the perturbation matrix.

The following two theorems are given in [10, 30].

Theorem 2.1 *Given $a(x)\,, b(x) \in \mathbb{C}[x]$ with $\deg(a) = m$ and $\deg(b) = n$. Let $S(a, b)$ be the Sylvester matrix of $a(x)$ and $b(x)$, S_k the k-th Sylvester matrix, $1 \le k \le \min(m, n)$. Then the following statements are equivalent:*

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

(2) $\deg(\gcd(a, b)) \ge k$.

(3) $\operatorname{rank}(S_k) \le m + n + 1 - 2k$.

(4) Rank deficiency of S_k is not less than one.

Theorem 2.2 *In the same assumption as in Theorem 2.1, let $S_k = [\mathbf{a}\ A_k]$, where $\mathbf{a}$ is the first column of S_k, then $\operatorname{rank}(S) \le m + n - k$ if and only if $A_k\mathbf{x} = \mathbf{a}$ has a solution.*

Based on the above two theorems, for a given degree k, it is always possible to find a k-th Sylvester structured perturbation matrix $[\Delta\mathbf{a}\, D_k]$ such that $\mathbf{a}+\Delta\mathbf{a} \in \mathrm{Range}(A_k+D_k)$. Then, the minimal perturbation problem can be formulated as the following equality-constrained least squares problem:

$$\min_{\mathbf{x}} \|\mathbf{d}\|_2, \quad \text{subject to } \mathbf{r} = 0, \tag{1}$$

where the structured residual $\mathbf{r}$ is given by

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

Applying the penalty method [22], we transform (1) into

$$\min_{\mathbf{d},\mathbf{x}} \left\| \begin{pmatrix} w\mathbf{r} \\ \mathbf{d} \end{pmatrix} \right\|_2, \tag{2}$$

where w is a large penal value.

It is an elementary calculus to show that, with matrices P_k and $X_k \in \mathbb{C}^{(m+n-k+1)\times(m+n+2)}$, vectors $\mathbf{\Delta a}$ and $D_k\mathbf{x}$ can be expressed as

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

Here

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

where I_{m+1} is an identity matrix of order $m+1$, and

$$X_k = \begin{pmatrix} 0 & & & x_{n+1-k} & & \\ x_1 & \ddots & & x_{n+2-k} & \ddots & \\ \vdots & \ddots & 0 & \vdots & \ddots & x_{n+1-k} \\ x_{n-k} & & x_1 & x_{m+n+1-2k} & & x_{n+2-k} \\ & \ddots & \vdots & & \ddots & \vdots \\ & & x_{n-k} & & & x_{m+n+1-2k} \end{pmatrix}.$$
$$\underbrace{\hspace{6em}}_{m+1} \quad \underbrace{\hspace{9em}}_{n+1}$$

Then (2) becomes the following least squares problem:

$$\min_{\Delta\mathbf{x}\,\Delta\mathbf{d}} \left\| \begin{pmatrix} w(X_k - P_k) & w(A_k + D_k) \\ I_{m+n+2} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{d} \\ \Delta\mathbf{x} \end{pmatrix} + \begin{pmatrix} -w\mathbf{r} \\ \mathbf{d} \end{pmatrix} \right\|_2, \tag{3}$$

where I_{m+n+2} is an identity matrix of order $m+n+2$.

EXAMPLE 2.2 *Continued from Example 2.1,*

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

$$X_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & x_2 & 0 & 0 & 0 \\ x_1 & 0 & 0 & 0 & x_3 & x_2 & 0 & 0 \\ 0 & x_1 & 0 & 0 & 0 & x_3 & x_2 & 0 \\ 0 & 0 & x_1 & 0 & 0 & 0 & x_3 & x_2 \\ 0 & 0 & 0 & x_1 & 0 & 0 & 0 & x_3 \end{pmatrix}.$$

3 A fast algorithm for solving the least squares problem (3)

Now we present an efficient method, based on the displacement structure of the involving coefficient matrix, to solve the least squares problem (3).

The displacement structure of an $i \times i$ Hermitian matrix H was originally defined by Kailath, Kung and Morf[16] as

$$\nabla H = H - Z_i H Z_i^T, \tag{4}$$

where i is a positive integer. Throughout this paper, Z_i denotes the $i \times i$ lower shift matrix with ones on the first subdiagonal and zeros elsewhere: For example, when $i = 3$,

$$Z_3 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

If ∇H has a lower rank r ($<< i$) independent of i, then r is referred to as the displacement rank of H. It follows that ∇H can be factored as $\nabla H = GJG^T$, where G is an $i \times r$ matrix and J is a signature matrix. The pair (G, J) is said to be a generator pair of H. Triangular factorization of H can be efficiently carried out by a generalized Schur algorithm [17], which operates on the generator pair (G, J) of H directly.

Let us denote the coefficient matrix of the system in (3) by M,

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

Theorem 3.1 *M is a structured matrix with displacement rank at most* 4.

Proof of Theorem 3.1: We construct two block-shift matrices

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

then the rank of the matrix $M - F_1 M F_2^T$ is at most 4. In fact,

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

where $\mathbf{e}_i$ denotes the i-th column of an identity matrix $I_{2m+2n-k+3}$, and

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

EXAMPLE 3.1 *Continued from Example 2.1,* $\mathbf{u_1}, \cdots, \mathbf{u_4}$ *can be written down directly as:*

$$\begin{aligned} \mathbf{u_1} &= [-w,\ wx_1,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]^T, \\ \mathbf{u}_2 &= [wx_2,\ wx_3,\ 0,\ 0, 0, 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0]^T, \\ \mathbf{u}_3 &= [0,\ w(a_3+d_1), w(a_2+d_2), w(a_1+d_3), w(a_0+d_4),\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]^T, \\ \mathbf{u}_4 &= [w(b_3+d_5), w(b_2+d_6), w(b_1+d_7), w(b_0+d_8),\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0]^T. \end{aligned}$$

Let $\mathbf{y} = \begin{pmatrix} \Delta\mathbf{d} \\ \Delta\mathbf{x} \end{pmatrix}$ and $\mathbf{z} = \begin{pmatrix} w\mathbf{r} \\ -\mathbf{d} \end{pmatrix}$, the least squares problem (3) can be rewritten as

$$\min_{\mathbf{y}} \|M\mathbf{y} - \mathbf{z}\|_2. \qquad (5)$$

As in [5, 6], we consider the augmented matrix

$$T = \begin{pmatrix} M^T M & M^T \\ M & 0 \end{pmatrix}.$$

Theorem 3.2 *The Hermitian matrix T is a block-shift structured matrix, and its displacement rank is at most 8.*

Proof of Theorem 3.2: Write a block-shift matrix

$$F = \mathrm{diag}(Z_{m+1}, Z_{n+1}, Z_{m-k}, Z_{n-k+1}, Z_{m+n-k+1}, Z_{m+n+2}),$$

we can construct a generator pair (G, J) for T so that $T - FTF^T = GJG^T$. Here $J = \mathrm{diag}(I_4, -I_4)$, where I_4 is the identity matrix of order 4, and G consists of the following eight columns:

$$
\begin{aligned}
g_1 &= \text{Column}(T,1) \text{ except that} \\
&\quad g_1[1] = (T[1,1]+1)/2, g_1[m+2] = T[1,m+2]/2, \\
g_2 &= \text{Column}(T,m+2) \text{ except that} \\
&\quad g_2[1] = T[1,m+2]/2, g_2[m+2] = (T[m+2,m+2]+1)/2, \\
g_3 &= \text{Column}(T,m+n+3),\ \text{except that } g_3[1]=0,\ g_3[m+2]=0, \\
&\quad g_3[m+n+3] = (T[m+n+3,m+n+3]+1)/2, \\
&\quad g_3[2m+n-k+3] = T[m+n+3,2m+n-k+3]/2, \\
g_4 &= \text{Column}(T,2m+n-k+3),\ \text{except that } g_4[1]=0,\ g_4[m+2]=0, \\
&\quad g_4[m+n+3] = T[m+n+3,2m+n-k+3]/2, \\
&\quad g_4[2m+n-k+3] = (T[2m+n-k+3,2m+n-k+3]+1)/2, \\
g_5 &= \text{Column}(T,1),\ \text{except that} \\
&\quad g_5[1] = (T[1,1]-1)/2,\ g_5[m+2] = T[1,m+2]/2, \\
g_6 &= \text{Column}(T,m+2),\ \text{except that} \\
&\quad g_6[1] = T[1,m+2]/2,\ g_6[m+2] = (T[m+2,m+2]-1)/2. \\
g_7 &= \text{Column}(T,m+n+3),\ \text{except that } g_7[1]=0,\ g_7[m+2]=0, \\
&\quad g_7[m+n+3] = (T[m+n+3,m+n+3]-1)/2, \\
&\quad g_7[2m+n-k+3] = T[m+n+3,2m+n-k+3]/2, \\
g_8 &= \text{Column}(T,2m+n-k+3),\ \text{except that } g_8[1]=0,\ g_8[m+2]=0, \\
&\quad g_8[m+n+3] = T[m+n+3,2m+n-k+3]/2, \\
&\quad g_8[2m+n-k+3] = (T[2m+n-k+3,2m+n-k+3]-1)/2.
\end{aligned}
$$

■

Remark We don't explicitly form the large matrix T. For example,

$$\text{Column}(T,m+2) = \begin{pmatrix} M^T\text{Column}(M,m+2) \\ \text{Column}(M,m+2) \end{pmatrix},$$

Since

$$M^T - F_2 M^T F_1^T = [\mathbf{e}_1, \mathbf{e}_{m+2}, \mathbf{e}_{m+n+3}, \mathbf{e}_{2m+n-k+3}][\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4]^T, \tag{6}$$

the matrix-vector product can be efficiently implemented by the convolution computations. Let us assume that $m \geq n$, M^T can be represented by columns of generator as

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

where

$$L_j = [\mathbf{v}_j, F_2\mathbf{v}_j, \cdots, F_2^m \mathbf{v}_j], \;\; N_j = [\mathbf{u}_j, F_1\mathbf{u}_j, \cdots, F_1^m \mathbf{u}_j],$$

and $\mathbf{v}_j (j = 1, \cdots, 4)$ denotes the four unit vectors in (6). Hence, the involving matrix-vector product can be efficiently implemented by a few number of vectors products.

Based on the results in [5, 6], after applying $2m + 2n - 2k + 3$ positive steps and $2m + 2n - k + 3$ negative steps of the generalized Schur algorithm to the generator pair (G, J) of T, we can obtain a backward stable triangular factorization:

$$\hat{T} = \begin{pmatrix} \hat{R}^T & 0 \\ \hat{Q} & \hat{D} \end{pmatrix} \begin{pmatrix} \hat{R} & \hat{Q}^T \\ 0 & -\hat{D}^T \end{pmatrix}. \; and \; \|T - \hat{T}\|_2 < \varepsilon. \tag{7}$$

When the matrix M is well-conditioned, the lower triangular matrix $\hat{D}$ is also well-conditioned and $\hat{D}^{-1}\hat{Q}$ is numerically orthogonal [5]. From (7), we also have

$$\|M - \hat{Q}\hat{R}\|_2 \leq \varepsilon.$$

Then by solving the nearby system

$$\hat{D}(\hat{D}^{-1}\hat{Q})\hat{R}\mathbf{y} = \mathbf{z},$$

we obtain the solution of (5) as

$$\mathbf{y} = \hat{R}^{-1}(\hat{Q}^T \hat{D}^{-T})\hat{D}^{-1}\mathbf{z}. \tag{8}$$

If M is ill-conditioned, in order to obtain a backward stable triangular factorization, we consider the perturbed matrix:

$$\bar{T} = \begin{pmatrix} M^T M + \alpha \mathbf{I} & M^T \\ M & -\beta \mathbf{I} \end{pmatrix},$$

$\alpha \mathbf{I}$ and $\beta \mathbf{I}$ are all small multiples of an identity matrix. The displacement rank of $\bar{T}$ increases by two:

$$\bar{T} - F\bar{T}F^T = \bar{G}\bar{J}\bar{G}^T,$$

where

$$\bar{J} = \mathrm{diag}(I_4, -I_6),$$

and the first 8 columns of $\bar{G}$ have similar forms as G, we only need to update the four diagonal elements as

$$\begin{array}{rcl} T[1,1] & \longrightarrow & T[1,1]+\alpha, \\ T[m+2,m+2] & \longrightarrow & T[m+2,m+2]+\alpha, \\ T[m+n+3,m+n+3] & \longrightarrow & T[m+n+3,m+n+3]+\alpha, \\ T[2m+n-k+3,2m+n-k+3] & \longrightarrow & T[2m+n-k+3,2m+n-k+3]+\alpha. \end{array}$$

The last two columns of $\bar{G}$ are

$$\begin{array}{rcl} \bar{g}_9 & = & [\ \underbrace{0,\ \cdots,\ 0}_{2m+2n-2k+3},\sqrt{\beta},\ 0,\ \cdots,\ 0,\ 0\ ]^T, \\ \bar{g}_{10} & = & [\ \underbrace{0,\ \cdots,\ 0,\ 0}_{3m+3n-3k+4},\ \sqrt{\beta},\ 0,\ \cdots,\ 0\ ]^T. \end{array}$$

Applying the generalized Schur algorithm to the generator pair $(\bar{G}, \bar{J})$ of the perturbed matrix $\bar{T}$, we can obtain a backward stable solution, which has a representation identical to the formula (8) [5].

In practice, if we start from the degree upper bound of GCDs (obtained from the computation of the singular value decomposition of the Sylvester matrix), we can always avoid ill-conditioned cases.

4 Fast version of Approximate GCD Algorithm

The results of the previous sections provide us a fast version of the approximate GCD algorithm [10, 30].

Algorithm AppFSylv-k

Input - A Sylvester matrix S generated by two polynomials $a(x)$ and $b(x)$ of degrees m and n respectively, $m \geq n$, $\|a\|_2 = \|b\|_2 = 1$; an integer k, $1 \leq k \leq n$; and a tolerance tol.

Output- Two polynomials $\tilde{a}$ and $\tilde{b}$ with $\text{rank}(S(\tilde{a}, \tilde{b})) \leq m+n-k$, and the Euclidean distance $\|\tilde{a}-a\|_2^2 + \|\tilde{b}-b\|_2^2$ is reduced to a minimum.

1. Form the k-th Sylvester matrix S_k, let $\mathbf{a}$ be the first column of S_k and A_k be the last $m+n-2k+1$ columns of S_k. Let $D_k = \mathbf{0}$, $\Delta\mathbf{a} = \mathbf{0}$.

2. Compute $\mathbf{x}$ from $\min \|A_k\mathbf{x} - \mathbf{a}\|_2$ and $\mathbf{r} = \mathbf{a} - A_k\mathbf{x}$. Form P_k and X_k as shown in Section 2.

3. Repeat

   (a) Apply fast algorithm described in Section 3 to solve

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

   (b) Set $\mathbf{x} = \mathbf{x} + \Delta\mathbf{x}$, $\mathbf{d} = \mathbf{d} + \Delta\mathbf{d}$.

   (c) Construct matrices $[\Delta\mathbf{a}\, D_k]$ from $\mathbf{d}$ and X_k from $\mathbf{x}$. Set $A_k = A_k + D_k$, $\mathbf{a} = \mathbf{a} + \Delta\mathbf{a}$, $\mathbf{r} = \mathbf{a} - A_k\mathbf{x}$.
   until $(\|\Delta\mathbf{x}\|_2 \le tol$ and $\|\Delta\mathbf{d}\|_2 \le tol)$

4. Output the polynomials $\tilde{a}$ and $\tilde{b}$ formed from $[\mathbf{a}\, A_k]$.

Given a tolerance ϵ, the fast algorithm AppFSylv-k can be used to compute the highest degree ϵ-GCD of polynomials a and b with degrees m and n respectively. Denote by $\bar{r}$ the degree upper bound of ϵ-GCD, we start from $k = \bar{r}$ and perform AppFSylv-k to compute the minimum $\mathcal{N} = \|\tilde{a} - a\|_2^2 + \|\tilde{b} - b\|_2^2$ with $\operatorname{rank}(S(\tilde{a}, \tilde{b})) \le m + n - k$. If $\mathcal{N} < \epsilon$, then we can compute the ϵ-GCD from the matrix $S_k(\tilde{a}, \tilde{b})$ taking a method as in [12]. Otherwise, we reduce k by one and repeat the AppFSylv-k algorithm.

5 Concluding Remarks

In this paper, we have presented a fast algorithm for constructing low rank approximation of a Sylvester matrix. The overall computation time of the algorithm AppFSylv-k depends on the third step which costs $O(s^2 + st + t^2)$, where s, t denote the row dimension $2m + 2n - k + 3$ and column dimension $2m + 2n - 2k + 3$ of the involving coefficient matrix respectively. The complexity needed in solving the least squares problem by algorithm presented in [10, 30] is $O(st^2)$. This shows that our algorithm is one order faster than the previous algorithm.

References

[1] Anda, A.A. and Park, H. Fast plane with dynamic scaling, *SIAM J. Matrix Anal. Appl.* vol.15, pp. 162–174, 1994.

[2] N. Karmarkar and Lakshman Y. N., Approximate polynomial greatest common divisors and nearest singular polynomials, in *International Symposium on Symbolic and Algebraic Computation*, Zürich, Switzerland, 1996, pp. 35–42, ACM.

[3] N. K. Karmarkar and Lakshman Y. N., On approximate GCDs of univariate polynomials, *Journal of Symbolic Computation*, vol.26, no. 6, pp. 653–666, 1998, Special issue of the JSC on Symbolic Numeric Algebra for Polynomials S. M. Watt and H. J. Stetter, editors.

[4] Bernhard Beckermann and George Labahn, When are two polynomials relatively prime?, *Journal of Symbolic Computation*, vol. 26, no. 6, pp. 677–689, 1998, Special issue of the JSC on Symbolic Numeric Algebra for Polynomials S. M. Watt and H. J. Stetter, editors.

[5] S.Chandrasekaran and A.H.Sayed, A fast stable solver for nonsymmetric Toeplitz and quasi-Toeplitz systems of linear equations, *SIMAX,* Vol.19, No.1, 1998, pp. 107-139.

[6] S.Chandrasekaran and A.H.Sayed, Stabilizing the generalized Schur algorithm, *SIMAX*, Vol.17, No.4, 1996, pp.950-983.

[7] P.Chin, R.M.Corless, C.F.Corless, Optimization strategies for the approximate GCD problems, In *Proc. 1998 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'98)* pp. 228–235.

[8] R.M.Corless, P.M.Gianni, B.M.Trager, and S.M.Watt, The singular value decomposition for polynomial systems, *Proc. 1995 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'95)* pp. 195–207.

[9] Robert M. Corless, Stephen M. Watt, and Lihong Zhi QR factoring to compute the GCD of univariate approximate polynomials, submitted, 2002.

[10] E. Kaltofen, Z. Yang and L. Zhi Structured low rank approximation of a Sylvester matrix. Submitted to ISSAC05.

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

[12] S.Gao, E.Kaltofen, J.May, Z.Yang and L.Zhi, Approximate factorization of multivariate polynomials via differential equations. In *Proc. 2004 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'04)* pp. 167–174.

[13] Gene H. Golub and Charles F. Van Loan, " Matrix computations," Johns Hopkins, 3nd edition, 1996.

[14] M.Gu, Stable and efficient algorithms for structured systems of linear equations," In *SIAM J. Matrix Anal. Appl.*, vol.19, pp. 279-306, 1997.

[15] V.Hribernig and H.J.Stetter, Detection and validation of clusters of polynomials zeros, *J. Symb. Comput.* vol.24: pp 667–681, 1997.

[16] Thomas Kailath, S.Y.Kung, and M.Morf, Displacement ranks of a matrix, *Bull. Amer. Math. Socl.,* Vol.1, 1979, pp.769-773.

[17] Thomas Kailath and Ali H. Sayed, Displacement structure: theory and applications, *SIAM Review,* Vol.37, No.3, 1995, pp.297-386.

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

[19] Li, B.Y., Liu, Z.J. and Zhi, L.H. A structured rank-revealing method for Sylvester matrix. Submitted.

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

[21] V.Y.Pan Numerical computation of a polynomial GCD and extensions. *Information and computation*, 167:71-85, 2001.

[22] Park, H., Zhang, L. and Rosen, J.B. Low rank approximation of a Hankel matrix by structured total least norm, *BIT Numerical Mathematics* vol.35-4: pp. 757–779,1999.

[23] Rosen, J.B., Park, H. and Glick, J. Total least norm formulation and solution for structured problems. *SIAM Journal on Matrix Anal. Appl.* vol.17-1: pp. 110–128, 1996.

[24] A. Schönhage, "Quasi-gcd computations," *Journal of Complexity*, vol.1, pp. 118–137, 1985.

[25] Zeng, Z. A method computing multiple roots of inexact polynomials. In *Proc. 2003 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'03)*, pp. 266–272.

[26] Zeng, Z. The approximate GCD of inexact polynomials. Part I: a univariate algorithm . Manuscript, 2004.

[27] Zhi,L. and Noda, M.T. Approximate GCD of multivariate polynomials *Proc.ASCM 2000, World Scientific Press.* pp. 9–18, 2000.

[28] 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.

[29] C.-P.Jeannerod and L.Zhi Fast generation of Sylvester matrix's embeddings and applications. Manuscript, July, 2004.

[30] Lihong Zhi and Zhengfeng Yang Computing approximate GCD of univariate polynomials by structure total least norm. In*MM Research Preprints, 11-19, No.23, December 2004.*