Math.comput.sci. 1 (2007), 427–437

1661-8270/020427-11, *published online* October 5, 2007
DOI 10.1007/s11786-007-0014-6

Mathematics in Computer Science

Structured Low Rank Approximation of a Bezout Matrix

Dongxia Sun and Lihong Zhi

Abstract. The task of determining the approximate greatest common divisor (GCD) of more than two univariate polynomials with inexact coefficients can be formulated as computing for a given Bezout matrix a new Bezout matrix of lower rank whose entries are near the corresponding entries of that input matrix. We present an algorithm based on a version of structured nonlinear total least squares (SNTLS) method for computing approximate GCD and demonstrate the practical performance of our algorithm on a diverse set of univariate polynomials.

Mathematics Subject Classification (2000). Primary 68W30; Secondary 65K10.

Keywords. Bezout matrix, approximate greatest common divisor, structured nonlinear total least squares, symbolic/numeric hybrid method.

1. Introduction

The computation of approximate GCDs of univariate polynomials has been extensively studied recently. The Euclidean algorithm has been considered early in [4, 11, 23, 24, 30–33] to compute approximate GCDs of polynomials with floating point coefficients. QR-decomposition or SVD-based total least squares methods were introduced in [7, 8, 18, 19, 29, 34, 35]. In [25] nearby roots are matched.

In [15], the approximate GCD computation was formulated as an optimization problem:

Problem 1.1. Given univariate polynomials $f_1, \dots, f_l \in \mathbb{R}[x] \setminus \{0\}$ with $\deg(f_1) = d_1, \dots, \deg(f_l) = d_l$, we assume $d_1 = \max(d_1, \dots, d_l)$. For a positive integer k, $k \le \min(d_1, \dots, d_l)$, we wish to compute $\triangle f_1, \dots, \triangle f_l \in \mathbb{R}[x]$ such that $\deg(\triangle f_1) \le d_1, \dots, \deg(\triangle f_l) \le d_l$, $\deg(\mathrm{GCD}(f_1 + \triangle f_1, \dots, f_l + \triangle f_l)) \ge k$ and $\|\triangle f_1\|_2^2 + \cdots + \|\triangle f_l\|_2^2$ is minimized.

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

In [15–17, 29], the authors transformed the above problem into computing for a generalized Sylvester matrix the nearest singular matrix with the generalized Sylvester structure. They presented iterative algorithms based on structured total least norm algorithms in [20, 21, 26, 27] to solve the optimization problem.

It is well known that Bezout matrix can also be used to compute GCDs of univariate polynomials [1–3, 5, 6, 12]. In [9, 10], the authors generalized the Bezout matrix for several univariate polynomials and apply SVD-based total least squares method to compute approximate GCDs. Compared with the generalized Sylvester matrix, the generalized Bezout matrix has smaller size. However, entries of the Bezout matrix are bilinear in coefficients of the polynomials. Hence, we propose to apply the structured nonlinear total least squares (SNTLS) algorithm [21, 28] to compute the nearest singular matrix with Bezout structure. We show how to solve Problem 1.1, at least for a local minimum, by applying SNTLS with L_2 norm to a submatrix of the generalized Bezout matrix.

We organize the paper as follows. In Section 2, we introduce some notations and discuss the equivalence between the GCD problem and the low rank approximation of a matrix with Bezout structure. In Section 3, we consider solving an overdetermined system with Bezout structure based on SNTLS. In Section 4, we describe our algorithm for two examples and compare our results with previous work in [15–17]. We conclude in Section 5 with remarks on the complexity and the rate of convergence of our algorithm.

2. Preliminaries

Suppose we are given two univariate polynomials $f_1, f_2 \in \mathbb{R}[x]\backslash\{0\}$ with $\deg(f_1) = m$ and $\deg(f_2) = n$, assume $m \geq n$,

$$\begin{aligned} f_1 &= u_m x^m + u_{m-1}x^{m-1} + \cdots + u_1 x + u_0\,, \quad u_m \neq 0\,, \\ f_2 &= v_n x^n + v_{n-1}x^{n-1} + \cdots + v_1 x + v_0\,, \quad v_n \neq 0\,. \end{aligned} \tag{2.1}$$

The Bezout matrix $\hat{B}(f_1, f_2) = (\hat{b}_{ij})$ is defined by

$$\hat{b}_{ij} = |u_0 v_{i+j-1}| + |u_1 v_{i+j-2}| + \cdots + |u_k v_{i+j-k-1}|\,,$$

where $|u_r v_s| = u_s v_r - u_r v_s$, $k = \min(i-1, j-1)$ and $v_r = 0$ if $r > n$ [3, 13]. It satisfies that

$$\frac{f_1(x)f_2(y) - f_1(y)f_2(x)}{x-y} = [1, x, x^2, \ldots, x^{m-1}]\hat{B}(f_1, f_2)[1, y, y^2, \ldots, y^{m-1}]^T\,. \tag{2.2}$$

Notice that the Bezout matrix $B(f_1, f_2)$ defined in Maple is as follows:

$$B(f_1, f_2) = -J\hat{B}(f_1, f_2)J\,, \tag{2.3}$$

where J is an anti-diagonal matrix with 1 as its nonzero entries.

The Bezout matrix can be generalized for nonzero univariate polynomials $f_1, \ldots, f_l \in \mathbb{R}[x] \setminus \{0\}$ with $\deg(f_1) = d_1, \ldots, \deg(f_l) = d_l$. Suppose $d_1 = \max(d_1, \ldots, d_l)$, $B(f_1, \ldots, f_l) \in \mathbb{R}^{(l-1)d_1 \times d_1}$ is defined by

$$B(f_1, \ldots, f_l) = \begin{bmatrix} B(f_1, f_2) \\ B(f_1, f_3) \\ \vdots \\ B(f_1, f_l) \end{bmatrix}. \tag{2.4}$$

The following three theorems summarize the relationship between the greatest common divisor(GCD) of $f_1, \ldots, f_l$ and the Bezout matrix $B(f_1, \ldots, f_l)$.

Theorem 2.1 (Theorem 3.2 in [9]). *Given univariate polynomials $f_1, \ldots, f_l \in \mathbb{R}[x] \setminus \{0\}$ with $\deg(f_1) = d_1, \ldots, \deg(f_l) = d_l$, $d_1 = \max(d_1, \ldots, d_l)$, then we have $\dim(\mathrm{Ker}B(f_1, \ldots, f_l))$ being equal to the degree of the GCD of $f_1, \ldots, f_l$.*

Theorem 2.2 (Theorem 3.3 in [9]). *Given univariate polynomials $f_1, \ldots, f_l \in \mathbb{R}[x] \setminus \{0\}$ with $\deg(f_1) = d_1, \ldots, \deg(f_l) = d_l$, $d_1 = \max(d_1, \ldots, d_l)$, then the degree of the GCD of $f_1, \ldots, f_l$ is at least k for $k \le \min(d_1, \ldots, d_l)$ if and only if the first $d_1 - k + 1$ columns of $B(f_1, \ldots, f_l)$ are linearly dependent.*

Theorem 2.3. *Given univariate polynomials $f_1, \ldots, f_l \in \mathbb{R}[x] \setminus \{0\}$ with $\deg(f_1) = d_1, \ldots, \deg(f_l) = d_l$, $d_1 = \max(d_1, \ldots, d_l)$, let $c(x) = \mathrm{GCD}(f_1(x), \ldots, f_l(x))$ be a polynomial of degree k, then we have:*

- $\mathrm{rank}(B(f_1, \ldots, f_l)) = d_1 - k$;
- *Suppose $\mathbf{y} = (y_0, y_1, \ldots, y_{d_1-k-1})^T$ satisfies $C\mathbf{y} = \mathbf{b}$, where C consists of the first $d_1 - k$ columns of $B(f_1, \ldots, f_l)$, and $\mathbf{b}$ is a vector formed from the $d_1 - k + 1$-th column of $B(f_1, \ldots, f_l)$. Let*

  $$\mathbf{w} = [w_0, \ldots, w_{d_1-k}]^T = \left(JB(f_1, 1)\right)_{d_1-k+1} [y_0, \ldots, y_{d_1-k}]^T ,$$

  where $y_{d_1-k} = -1$, and $(JB(f_1, 1))_{d_1-k+1}$ is the leading principal $d_1 - k + 1$-th submatrix, then

  $$f_1(x) = c(x)w(x) , \quad \text{with} \quad w(x) = \sum_{i=0}^{d_1-k} w_i x^i .$$

Proof. See Proposition 9.4, Remark 9.3 and Algorithm 9.1 in [6] for the case $l = 2$. The proof of Theorem 3.4 in [9] also gives us an alternative method to compute a GCD for the polynomials $f_1, \ldots, f_l$ from the generalized Bezout matrix. ☐

3. SNTLS for overdetermined system with Bezout structure

The Bezout matrix $B(f_1, \ldots, f_l)$ can be parameterized by a vector ζ which contains the coefficients of $f_1, \ldots, f_l$. By applying Theorem 2.1, we can transfer the Problem 1.1 into solving the following minimization problem:

$$\min_{\Delta\mathbf{s} \in \mathbb{R}^{d+l}} \|\Delta\mathbf{s}\|_2 \quad \text{with} \quad \dim\left(\mathrm{Ker}B(\mathbf{s} + \Delta\mathbf{s})\right) \ge k , \tag{3.1}$$

in which

$$\mathbf{s} = [f_{10}, \ldots, f_{1d_1}, \ldots, f_{l0}, \ldots, f_{ld_l}]\,, \tag{3.2}$$

where f_{ij} stands for the coefficient of x^j in polynomial f_i, and $d = \sum_{i=1}^{l} d_i$.

Let $B_k(\zeta) = [D_1(\zeta), \mathbf{b}(\zeta), D_2(\zeta)]$ be the first $d_1 - k + 1$ columns of $B(\zeta)$ and let $A(\zeta) = [D_1(\zeta), D_2(\zeta)] \in \mathbb{R}^{d_1(l-1)\times(d_1-k)}$. According to Theorem 2.2, the minimization problem (3.1) can be transferred into the following structured nonlinear total least squares problem:

$$\min_{\Delta\mathbf{s}\in\mathbb{R}^{d+l}} \|\Delta\mathbf{s}\|_2 \quad \text{with} \quad A(\mathbf{s}+\Delta\mathbf{s})\,\mathbf{x} = \mathbf{b}(\mathbf{s}+\Delta\mathbf{s})\,, \quad \text{for some vector } \mathbf{x}\,. \tag{3.3}$$

The choice of which column of B_k moved to the right side depends on whether the nearest singular matrix contains that column in a linear column relation. Similar to [15–17], we choose that column as $\mathbf{b}(\zeta) \in \mathbb{R}^{d_1(l-1)\times 1}$ for which the corresponding component in the first right singular vector of B_k is maximum in absolute value. In the following, we illustrate how to find the minimum solution of (3.3) using the structured nonlinear total least squares (SNTLS) method.

We can initialize $\mathbf{x}$ as the unstructured least squares solution $A(\mathbf{s})\mathbf{x} = \mathbf{b}(\mathbf{s})$ for the input vector $\mathbf{s}$. The perturbation can be initialized as $\Delta\mathbf{s} = \mathbf{0}$. However, as pointed by [20, Section 4.5.3] and [16,17], another way is to initialize $\Delta\mathbf{s}$ and $\mathbf{x}$ such that they satisfy the nonlinear constraints approximately, $A(\mathbf{s}+\Delta\mathbf{s})\mathbf{x} \approx \mathbf{b}(\mathbf{s}+\Delta\mathbf{s})$. We compute $\Delta\mathbf{s}$ as follows:

$$\Delta\mathbf{s} = -Y^T(\mathbf{s},\mathbf{v})\big(Y(\mathbf{s},\mathbf{v})Y^T(\mathbf{s},\mathbf{v})\big)^{-1}B_k(\mathbf{s})\mathbf{v}\,, \tag{3.4}$$

where $\mathbf{v}$ is the right singular vector corresponding to the smallest singular value of $B_k(\mathbf{s})$ and the matrix Y is the Jacobian of $B_k(\zeta)\mathbf{v}$ with respect to ζ, we have

$$B_k(\mathbf{s}+\Delta\mathbf{s})\mathbf{v} = B_k(\mathbf{s})\mathbf{v} + Y(\mathbf{s},\mathbf{v})\Delta\mathbf{s} + O(\|\Delta\mathbf{s}\|_2^2) = O(\|\Delta\mathbf{s}\|_2^2)\,.$$

Suppose $\mathbf{b}(\mathbf{s})$ is the t-th column corresponding to the absolutely largest component in $\mathbf{v}$; We initialize the vector $\mathbf{x}$ by normalizing the vector $\mathbf{v}$ to make $\mathbf{v}[t] = -1$ and deleting the t-th term $\mathbf{v}[t]$, i.e.,

$$\mathbf{x} = \left[-\frac{\mathbf{v}[1]}{\mathbf{v}[t]}, \ldots, -\frac{\mathbf{v}[t-1]}{\mathbf{v}[t]}, -\frac{\mathbf{v}[t+1]}{\mathbf{v}[t]}, \ldots\right]^T. \tag{3.5}$$

We have $A(\mathbf{s}+\Delta\mathbf{s})\mathbf{x} - \mathbf{b}(\mathbf{s}+\Delta\mathbf{s}) = O(\|\Delta\mathbf{s}\|_2^2)$. Since the initial values of $\Delta\mathbf{s}$ and $\mathbf{x}$ only satisfy the first order of the nonlinear constraints, for the second initialization method to be successful, we usually require that the initial perturbation (3.4) $\|\Delta\mathbf{s}\|_2 \ll 1$.

By introducing the Lagrangian multipliers, and neglecting the second-order terms in $\Delta\mathbf{s}$, the constrained minimization problem can be transformed into an unconstrained optimization problem [21,28]:

$$L(\Delta\mathbf{s},\mathbf{x},\lambda) = \frac{1}{2}\Delta\mathbf{s}^T\Delta\mathbf{s} - \lambda^T(\mathbf{b} - A\mathbf{x} - X\Delta\mathbf{s})\,, \tag{3.6}$$

where $X(\zeta, \mathbf{x})$ is the Jacobian of $\mathbf{r}(\zeta, \mathbf{x}) = A(\zeta)\mathbf{x} - \mathbf{b}(\zeta)$ with respect to ζ:

$$X(\zeta, \mathbf{x}) = \nabla_\zeta\big(A(\zeta)\mathbf{x}\big) - \nabla_\zeta\big(\mathbf{b}(\zeta)\big) = \sum_{j=1}^{d_1-k} x_j \nabla_\zeta a_j(\zeta) - \nabla_\zeta\big(\mathbf{b}(\zeta)\big)\,, \tag{3.7}$$

where $a_j(\zeta)$ represents the j-th column of $A(\zeta)$. Applying the Newton method on the Lagrangian L yields:

$$\begin{bmatrix} I_{d+l} & \mathbf{0}_{(d+l)\times(d_1-k)} & X(\mathbf{s}+\Delta\mathbf{s},\mathbf{x})^T \\ \mathbf{0}_{(d_1-k)\times(d+l)} & \mathbf{0}_{(d_1-k)\times(d_1-k)} & A(\mathbf{s}+\Delta\mathbf{s})^T \\ X(\mathbf{s}+\Delta\mathbf{s},\mathbf{x}) & A(\mathbf{s}+\Delta\mathbf{s}) & \mathbf{0}_{(l-1)d_1\times(l-1)d_1} \end{bmatrix} \begin{bmatrix} \Delta\tilde{\mathbf{s}} \\ \Delta\tilde{\mathbf{x}} \\ \Delta\tilde{\lambda} \end{bmatrix} = - \begin{bmatrix} \Delta\mathbf{s} + X(\mathbf{s}+\Delta\mathbf{s},\mathbf{x})^T\lambda \\ A(\mathbf{s}+\Delta\mathbf{s})^T\lambda \\ A(\mathbf{s}+\Delta\mathbf{s})\mathbf{x} - \mathbf{b}(\mathbf{s}+\Delta\mathbf{s}) \end{bmatrix}, \tag{3.8}$$

where $d = \sum_{i=1}^{l} d_i$. The iterative update $\mathbf{x} = \mathbf{x} + \Delta\tilde{\mathbf{x}}$, $\lambda = \lambda + \Delta\tilde{\lambda}$, $\mathbf{s} = \mathbf{s} + \Delta\mathbf{s} + \Delta\tilde{\mathbf{s}}$ is stopped when $\|\Delta\tilde{\mathbf{x}}\|_2$ and/or $\|\Delta\tilde{\mathbf{s}}\|_2$ and/or $\|\Delta\tilde{\lambda}\|_2$ becomes smaller than a given tolerance.

4. Experiments

Suppose we are given polynomials $f_1, \ldots, f_l \in \mathbb{R}[x] \setminus \{0\}$ with $\deg(f_1) = d_1, \ldots, \deg(f_l) = d_l$, and $d_1 = \max(d_1, \ldots, d_l)$, and a tolerance. We estimate the integer k from the singular values of the Bezoutian of $f_1, \ldots, f_l$. We compute the initial values of $\Delta\mathbf{s}, \mathbf{x}, \lambda$ by one of the two methods in Section 3, then solve the linear system (3.8) and update $\Delta\mathbf{s} = \Delta\mathbf{s} + \Delta\tilde{\mathbf{s}}$, $\mathbf{x} = \mathbf{x} + \tilde{\mathbf{x}}$, $\lambda = \lambda + \tilde{\lambda}$ until $\|\Delta\tilde{\mathbf{x}}\|_2$ and/or $\|\Delta\tilde{\mathbf{s}}\|_2$ and/or $\|\Delta\tilde{\lambda}\|_2$ are smaller than the given tolerance.

Suppose $\tilde{B}_k$ is the nearest singular matrix with Bezout structure computed successfully by SNTLS algorithm. Let the perturbed polynomials be $\tilde{f}_1, \ldots, \tilde{f}_l$. Suppose $k = \deg(\mathrm{GCD}(\tilde{f}_1, \ldots, \tilde{f}_l))$, the polynomial $c(x) = \mathrm{GCD}(\tilde{f}_1, \ldots, \tilde{f}_l)$ can be computed according to Theorem 2.3. However, we can also use the vector $\mathbf{x}$ returned from the iterations (3.8) to compute the GCD directly. Let $\mathbf{y} = [x_1, \ldots, x_{t-1}, -1, x_t, \ldots, x_{d_1-k}]^T$, compute the vector $\mathbf{w} = [w_0, \ldots, w_{d_1-k}]^T = (JB(\tilde{f}_1, 1))_{d_1-k+1}\mathbf{y}^T$, where $(JB(\tilde{f}_1, 1))_{d_1-k+1}$ is the leading principal d_1-k+1-th submatrix. The polynomial $c(x)$ is computed by a division of the polynomial $\tilde{f}_1(x)$ by the polynomial $w(x) = \sum_{i=0}^{d_1-k} w_i x^i$, and it is returned as the approximate GCD of $f_1, \ldots, f_l$.

Remark 4.1. When k is smaller than $\deg(\mathrm{GCD}(\tilde{f}_1, \ldots, \tilde{f}_l))$, as suggested in [17], we may increase k by $k+1$ and run our SNTLS algorithm on $\tilde{f}_1, \ldots, \tilde{f}_l$ again until we find the correct k.

We have implemented the algorithm in Maple 10 for computing the approximate GCDs of several univariate polynomials with real coefficients by structured low rank approximation of a Bezout matrix. The following two examples are computed by our algorithm in Maple 10 with *Digits* = 14. The results are listed with five digits.

Example 1 ([10]). Consider the polynomials

$$\begin{aligned} f_1 &= (x^5-1)(x^4-x+1)\,,\\ f_2 &= (x^5-0.9999)(x+4.0001)\,,\\ f_3 &= (x^5-0.9999)(x^4-3.0003x-2.9999)\,,\\ f_4 &= (x^5-1.0001)(x^4-3.0001x-0.9999)\,. \end{aligned}$$

The matrix $B_5(f_1,f_2,f_3,f_4)$ is of size 27×5, whereas the generalized Sylvester matrix used in [16] is of size 39×17. By our algorithms, for $k=5$, after two iterations, we stop the algorithm at $\Delta\mathbf{x} = .28218\times 10^{-5}$. The deformed polynomials are:

$$\begin{aligned} \tilde f_1 &= x^9 + .16753\times 10^{-5}x^8 + .26750\times 10^{-5}x^7 - x^6 + x^5 - .99997x^4\\ &\quad + .19010\times 10^{-5}x^3 + .23249\times 10^{-5}x^2 + .99998x - .99998\,,\\ \tilde f_2 &= .99998x^6 + 4x^5 + .16014\times 10^{-4}x^4 + .21290\times 10^{-4}x^3 + .17999\times 10^{-4}x^2\\ &\quad - .99992x - 3.9998\,,\\ \tilde f_3 &= .99996x^9 - .10714\times 10^{-5}x^8 - .80617\times 10^{-5}x^7 - 3.0002x^6 - 2.9998x^5\\ &\quad - .99994x^4 - .10723\times 10^{-5}x^3 - .80515\times 10^{-5}x^2 + 3.0001x + 2.9997\,,\\ \tilde f_4 &= 1.0001x^9 - .63203\times 10^{-5}x^8 - .43391\times 10^{-5}x^7 - 3.0003x^6 - .99998x^5\\ &\quad - x^4 - .63321\times 10^{-5}x^3 - .43314\times 10^{-5}x^2 + 3.0002x + 0.99992\,. \end{aligned}$$

The backward error

$$\mathcal{N} = \sqrt{\|\tilde f_1 - f_1\|_2 + \|\tilde f_2 - f_2\|_2^2 + \|\tilde f_3 - f_3\|_2^2 + \|\tilde f_4 - f_4\|_2^2} = .41295\times 10^{-3}\,.$$

The backward error computed by STLS algorithm in [16] is $.41292\times 10^{-3}$. However, our algorithm only takes 0.641 seconds while the STLS algorithm takes 7.031 seconds. The backward error given in [10] is larger than $.47610\times 10^{-3}$. The approximate GCD computed by our algorithm is

$$\begin{aligned} c(x) &= x^5 + .29055\times 10^{-5}x^4 + .43923\times 10^{-5}x^3\\ &\qquad + .37214\times 10^{-5}x^2 + .31134\times 10^{-5}x - .99995\,. \end{aligned}$$

Example 2 ([16]). Consider the polynomials

$$\begin{aligned} f_1 &= 1000x^{10} + x^3 - 1\,,\\ f_2 &= x^2 - 0.01\,. \end{aligned}$$

Case 1. If we initialize $\Delta\mathbf{s} = 0$ and $\mathbf{x}$ being the unstructured least squares solution of $A(\mathbf{s})\mathbf{x} = \mathbf{b}(\mathbf{s})$. After 10 iterations, we obtain the deformed polynomials

$$\tilde{f}_1 = 1000x^{10} - 0.00011x^9 - .00014x^8 - .00009x^7 + .00008x^6 - .00026x^5 + .00049x^4 + .99901x^3 + .00195x^2 - .00386x - .99238\,,$$

$$\tilde{f}_2 = .95204x^2 + .09462x - .19666\,,$$

which have a common divisor $x + 0.50690$, and the backward error is

$$\mathcal{N} = \|\tilde{f}_1 - f_1\|_2^2 + \|\tilde{f}_2 - f_2\|_2^2 = 0.04617\,.$$

As discussed in [16], this is only one of the local minimum.

Case 2. We initialize $\Delta\mathbf{s}$ by formula (3.4) and choose $\mathbf{v}$ being the right singular vector corresponding to the smallest singular value of $B_k(\mathbf{s})$ and normalized with respect to the largest entry. After 8 iterations, the algorithm returns

$$\tilde{f}_1 = 1000x^{10} + .00012x^9 - .00013x^8 + .00006x^6 + .00023x^5 + .00049x^4 + 1.0010x^3 + .00205x^2 + .00415x - .99156\,,$$

$$\tilde{f}_2 = .95614x^2 - .08876x - .18962\,,$$

which have a common divisor $x - 0.49415$, the backward error is

$$\mathcal{N} = \|\tilde{f}_1 - f_1\|_2^2 + \|\tilde{f}_2 - f_2\|_2^2 = .04216\,.$$

It is the global minimum similar to the one derived in [16].

In Table 1, we show the performance of our algorithm for computing approximate GCD of univariate polynomials on Pentium 4 at 2.0 Ghz for *Digits* = 14 in Maple 10 under Windows. For every example, we use 50 random cases for each $(d_1, \ldots, d_l)$, and report the average over all results. For each example, the prime parts and GCD of polynomials are constructed by choosing polynomials with random integer coefficients in the range $-10 \leq c \leq 10$, and then adding a perturbation. For noise we choose a relative tolerance 10^{-e}, then randomly choose a polynomial that has the same degree as the product, and coefficients in $[-10^e, 10^e]$. Finally, we scale the perturbation so that the relative error is 10^{-e}. Here d_i denotes the degree of the polynomial f_i; k is the degree of the approximate GCD of $f_1, \ldots, f_l$; *it.* $(STLS)$ is the number of the iterations needed by method in [17]; whereas *it.* $(SNTLS)$ denotes the number of iterations by our algorithm; *error* $(STLS)$ denotes the perturbation $\|\tilde{f}_1 - f_1\|_2^2 + \cdots + \|\tilde{f}_l - f_l\|_2^2$ computed by algorithm in [17]; whereas *error* $(SNTLS)$ is the minimal perturbation computed by our algorithm; the last two columns denote the time in seconds costed by two algorithms respectively.

5. Concluding remarks

In this paper we present a new way based on SNTLS to compute the approximate GCD of several univariate polynomials. The overall computational complexity of

Table 1. Algorithm performance on benchmarks (univariate case).

Ex.	d_i	k	*it.* (*STLS*)	*it.* (*SNTLS*)	*error* (*STLS*)	*error* (*SNTLS*)	*time*(s) (*STLS*)	*time*(s) (*SNTLS*)
1	$2, 2$	1	2.18	1.90	$6.96e{-6}$	$6.96e{-6}$	.25	.12
2	$3, 3$	2	2.17	1.93	$1.05e{-5}$	$1.07e{-5}$	.31	.14
3	$5, 4$	3	2.06	1.91	$1.56e{-5}$	$1.56e{-5}$	.44	.18
4	$5, 5$	3	2.27	2.00	$2.04e{-5}$	$2.75e{-5}$	.53	.20
5	$6, 6$	4	2.10	2.00	$2.18e{-5}$	$2.18e{-5}$	.58	.21
6	$8, 7$	4	2.10	1.90	$1.70e{-5}$	$1.70e{-5}$	.95	.31
7	$10, 10$	5	2.60	2.10	$3.43e{-4}$	$3.44e{-5}$	1.40	.43
8	$14, 13$	7	2.60	1.90	$5.73e{-5}$	$6.47e{-5}$	2.31	.80
9	$28, 28$	14	2.00	2.00	$2.60e{-5}$	$2.60e{-5}$	10.65	11.97
10	$10, 9, 8$	5	4.00	3.002	$7.96e{-5}$	$9.86e{-5}$	4.17	1.99
11	$8, 7, 8, 6$	4	4.40	3.20	$3.24e{-5}$	$3.28e{-5}$	5.83	1.41

the algorithm depends on the number of iterations needed for the first order update. If the starting values are good, then the iteration will converge quickly. This can be seen from the above table. Since the matrices involved in the minimization problems are all structured matrix, they have low displacement rank [14]. It would be possible to apply the fast algorithm to solve these minimization problems as in [22]. This would reduce the complexity of our algorithm to be only quadratic with respect to the degrees of the given polynomials.

Our methods can be generalized to several polynomials with arbitrary linear or nonlinear equational constraints imposed on the coefficients of the input and perturbed polynomials. However, at present, our algorithm can't deal with the polynomials with complex coefficients or the global minimal perturbations being complex. Notice that our algorithm also can not deal with the case $k = d_1$, because in that case the Bezout matrix B_k is not defined.

Acknowledgements

We thank the referees of an earlier version of this paper for their helpful remarks. We also thank Erich Kaltofen and Zhengfeng Yang for their valuable comments. This work was initiated by the discussions with Robert M. Corless during the Special Semester on Groebner Bases, February 20–24, 2006, organized by RICAM, Austrian Academy of Sciences, and RISC, Johannes Kepler University, Linz, Austria.

References

[1] S. Barnett, *Greatest common divisor of two polynomials*, Linear Algebra Appl., 3 (1970), pp. 7–9.

[2] S. Barnett, *Greatest common divisor of several polynomials*, Proc. Camb. Phil. Soc, 70 (1971), pp. 263–268.

[3] S. Barnett, *A note on the Bezoutian matrix*, SIAM J. Appl. Math., 22 (1972), pp. 84–86.

[4] B. Beckermann and G. Labahn, *A fast and numerically stable Euclidean-like algorithm for detecting relative prime numerical polynomials*, J. Symbolic Comput., 26 (1998), pp. 691–714.

[5] D. Bini and L. Gemignani, *Fast parallel computation of the polynomial remainder sequence via Bezout and Hankel matrices*, SIAM J. Comput., 22 (1993), pp. 63–77.

[6] D. Bini and V. Y. Pan, *Polynomial and matrix computations*, Vol. 1 of Fundamental Algorithms, Birkhäuser, 1994.

[7] R. M. Corless, P. M. Gianni, B. M. Trager, and S. M. Watt, *The singular value decomposition for polynomial systems*, in Proc. 1995 Internat. Symp. Symbolic Algebraic Comput. ISSAC'95, A. H. M. Levelt, ed., New York, 1995, ACM Press, pp. 96–103.

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

[9] G. Diaz-Toca and L. Gonzalez-Vega, *Barnett's theorems about the greatest common divisor of several univariate polynomials through Bezout-like matrices*, J. Symbolic Comput., 34 (2002).

[10] G. Diaz-Toca and L. Gonzalez-Vega, *Computing greatest common divisors and squarefree decompositions through matrix methods: The parametric and approximate cases*, Linear Algebra Appl., 412 (2006).

[11] D. K. Dunaway, *Calculation of zeros of a real polynomial through factorization using Euclid's algorithm*, SIAM J. Numer. Anal., 11 (1974), pp. 1087–1104.

[12] U. Helmke and P. Fuhrmann, *Bezoutians*, Linear Algebra Appl., 124 (1989), pp. 1039–1097.

[13] A. Householder, *Householder, Bezoutians, elimination and localization*, SIAM Review, 12 (1970), pp. 73–78.

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

[15] E. Kaltofen, Z. Yang, and L. Zhi, *Structured low rank approximation of a Sylvester matrix*. Manuscript, 15 pages, Oct. 2005. Preliminary version in SNC 2005 Proceedings, D. Wang and L. Zhi, eds., pp. 188–201, distributed at the International Workshop on Symbolic-Numeric Computation in Xi'an, China, July 19–21, 2005.

[16] E. Kaltofen, Z. Yang, and L. Zhi, *Approximate greatest common divisors of several polynomials with linearly constrained coefficients and singular polynomials*, in ISSAC'06 Proc. 2006 Internat. Symp. Symbolic Algebraic Comput., J. G. Dumas, ed., New York,2006, ACM Press, pp. 169–176.

[17] E. Kaltofen, Z. Yang, and L. Zhi, *Approximate greatest common divisors of several polynomials with linearly constrained coefficients and singular polynomials*. Manuscript, 20 pages, Dec 2006.

[18] N. Karmarkar and Y. N. Lakshman, *Approximate polynomial greatest common divisors and nearest singular polynomials*, in ISSAC'96 Proc. 1996 Internat. Symp. Symbolic Algebraic Comput., Y. N. Lakshman , ed., New York, 1996, ACM Press, pp. 35–42.

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

[20] P. Lemmerling, *Structured total least squares: Analysis, algorithms and applications*, dissertation, Katholieke Universiteit Leuven, Belgium, 1999.

[21] P. Lemmerling, N. Mastronardi, and S. V. Huffel, *Fast algorithm for solving the Hankel/Teoplitz strucured total least squares problem*, Numerical Algorithms, 23 (2000), pp. 371–392.

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

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

[24] M. Ochi, M. A. Noda, and T. Sasaki, *Approximate greatest common divisor of multivariate polynomialsand its application to ill-conditioned system of algebraic equations*, J. Inf. Process, 12 (1991), pp. 292–300.

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

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

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

[28] J. B. Rosen, H. Park, and J. Glick, *Structure total least norm for nonlinear problems*, SIAM J. Matrix Anal. Appl., 20 (1998), pp. 14–30.

[29] D. Rupprecht, *An algorithm for computing certified approximate gcd of n univariate polynomials*, J. Pure Appl. Algebra, 139 (1999), pp. 255–284.

[30] M. Sanuki, *Computing approximate GCD of multivariate polynomials* (*exented abstract*), in Proc. 2005 International Workshop on Symbolic-Numeric, D. Wang and L. Zhi, eds., July 2005, pp. 188–201. Distributed at the Workshop in Xi'an, China.

[31] M. Sasaki and T. Sasaki, *Polynomial remaider sequence and approximate GCD*, ACM SIGSAM Bulletin, 31 (2001), pp. 4–10.

[32] T. Sasasaki and M. T. Noda, *Approximate square-free decomposition and root-finding of ill-conditioned algebraic equations*, J. Inf. Process., 12 (1989), pp. 159–168.

[33] A. Schönhage, *Quasi-gcd computations*, Journal of Complexity, 1 (1985), pp. 118–137.

[34] Z. Zeng, *A method computing multiple roots of inexact polynomials*, in Proc. 2003 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'03), J. R. Sendra, ed., New York, 2003, ACM Press, pp. 266–272.

[35] L. Zhi, *Displacement structure in computing approximate GCD ofunivariate polynomials*, in Proc. Sixth Asian Symposium on Computer Mathematics (ASCM 2003), Z. Li and W. Sit, eds., Vol. 10 of Lecture Notes Series on Computing, Singapore, 2003, World Scientific, pp. 288–298.

Dongxia Sun and Lihong Zhi
Key Lab of Mathematics Mechanization
AMSS, Beijing 100080
China
e-mail: dsun@mmrc.iss.ac.cn
lzhi@mmrc.iss.ac.cn

Received: December 22, 2006.
Accepted: April 12, 2007.