Using Riemannian SVD for Problems in Approximate Algebra

Brad Botting, Mark Giesbrecht and John May

Abstract. Many fundamental problems for approximate polynomials can be reformulated as structured linear approximation problems. Problems such as finding polynomials with a GCD which are nearest to given relatively prime polynomials, finding the nearest Ore polynomials with a non-trivial GCRD, finding the nearest polynomial which functionally decomposes, and finding the nearest multivariate polynomial which factors or divides another polynomial, can all be cast as so-called Structured Total Least Squares (STLS) problems. More generally, given a basis of "structure" matrices for a vector space $\mathcal{V}$ of structured matrices, and a matrix A in $\mathcal{V}$, the STLS problem seeks to find the "nearest" B to A in $\mathcal{V}$ such that B is singular.

We present an implementation of one heuristic (the Riemannian SVD) to solve the STLS problem, and demonstrate its effectiveness on the problems discussed above. For the approximate GCD problem, we compare the solutions this approach finds to solutions found by other methods.

Introduction

In this paper we consider the application of the Riemannian SVD algorithm to the structured linear algebra problems associated with some fundamental problems for approximate polynomials.

We begin in Section 1 by presenting the "linearization" of these approximate polynomial problems. That is, we encode the input as a structured matrix, and show that a solution corresponds to finding a nearby singular matrix with the same structure. Nearness will be measured in the coefficient 2-norm of the input polynomials, and our resulting matrix formulations will optimize exactly this norm. We consider the problems of univariate and multivariate polynomial GCD, specific degree GCD and polynomial division, multivariate polynomial factorization, approximate functional decomposition, and Ore polynomial GCRD. Only the last of these linearizations is new, but the consistent formulation proves useful in what follows.

In Section 2 we introduce the Riemannian SVD algorithm, as developed in De Moor (1993, 1994). This is an effective and very general heuristic tool for finding nearby singular structured matrices.

Finally in Section 3, we present preliminary data suggesting the effectiveness of the Riemannian SVD algorithm on approximate polynomial problems. A more complete analysis will be provided in the full version of this paper.

1. Linearization of Polynomial Algebra Problems

In this section we formulate a number of fundamental problems with (approximate and exact) polynomials as structured linear algebra problems.

Univariate and Multivariate Polynomial GCD

The (exact) polynomial GCD problem can be stated as follows: given $f, g \in \mathbb{C}[x_1, \ldots, x_n]$, compute $d \in \mathbb{C}[x_1, \ldots, x_n]$ of maximal degree such that d divides both f and g. The conversion of the univariate GCD to a linear problem has been known since the 19th century. The approach, which works for the multivariate case as well, is to reformulate the problem as follows: given f and g, find u and v of so that

$$u\,f + v\,g = 0 \text{ with } \deg u < \deg g \text{ and } \deg v < \deg f. \tag{1}$$

For multivariate polynomials several notions of degree will work in (1); the most commonly used notion is probably total degree.

Since polynomial multiplication is a linear map, for a fixed polynomial basis, we can encode (1) as a matrix:

$$\mathrm{Syl}(f, g) = \left[C^{[\deg g - 1]}(f) \;\; C^{[\deg f - 1]}(g) \right]$$

where $C^{[d]}(f)$ is the matrix representing the multiplication of f by a polynomial of degree d. In the univariate case, the matrix $\mathrm{Syl}(f, g)$ is, of course, the Sylvester matrix of f and g. This reduces the GCD problem to computing a non-trivial null-vector of the matrix $\mathrm{Syl}(f, g)$.

The approximate version of univariate GCD has been considered in Karmarkar and Lakshman Y. N. (1998); Schönhage (1985); Corless et al. (1995); Emiris et al. (1997); Ruppert (1999); Corless et al. (2004); Zhi (2003); Zeng (2004); Li and Zeng (2005); Beckermann and Labahn (1998b,a) and the multivariate GCD in Ochi et al. (1991); Sasaki and Sasaki (1997); Zhi and Noda (2000); Zhi et al. (2001); Zeng and Dayton (2004). The univariate problem is, in some sense, solved by Karmarkar and Lakshman Y. N. (1998) (or at least shown to be computable in polynomial time), but there has since been much work done in making approximate GCD computation more practical. The problem of finding, in polynomial time, a guaranteed nearest pair of multivariate polynomials with a non-trivial GCD is still open, though some of the above listed algorithms appear effective and efficient in practice.

Specified Degree GCD and Polynomial Division

When considering the GCD problem approximately, one often wishes to find an approximate GCD of a certain prescribed degree. The corresponding exact problem is represented by (1) with modified bounds on u and v:

$$u\,f + v\,g = 0 \text{ with } \deg u \leq \deg g - k \text{ and } \deg v \leq \deg f - k. \tag{2}$$

Then (2) has a solution (u, v) exactly when f and g have a GCD of degree at least k. Notice that this encodes the problem of division when $k = \min\{\deg f, \deg g\}$. One can encode (2) as a matrix which we will denote $\mathrm{Syl}_k(f, g)$. Now again, computing a GCD of degree at least k of f and g reduces to computing a null-vector of $\mathrm{Syl}_k(f, g)$.

Unlike the standard GCD problem, there is no known polynomial-time algorithm to find the closest pair of polynomials with a GCD of a specified degree, even for univariate polynomials.

We should note that an alternative notion of approximate GCD requires an $\epsilon > 0$ as input and asks for the polynomials with GCD of largest degree that lie within a perturbation of less than ϵ. We make no direct statement about this problem, other than to note that a solution to the minization problem we will provide a solution to this problem, though the presribed degree GCD problem may well be more difficult.

Multivariate Polynomial Factorization

The exact absolute factorization problem is: given a polynomial $f \in \mathbb{C}[x_1, \ldots, x_n]$, $n > 1$, find non-constant polynomials $f_1, \ldots, f_r \in \mathbb{C}[x_1, \ldots, x_n]$ so that $f = \prod_{i=1}^{r} f_i$ with r as large as possible. For $n = 2$, Ruppert (1999) shows testing if $r > 1$ is a linear problem and Gao (2003) shows that computing f_i is a linear problem. Both results are generalized for all $n > 1$ in May (2005). To convert this to a linear problem, consider the partial differential equations

$$\frac{\partial}{\partial x_i}\frac{g}{f} = \frac{\partial}{\partial x_1}\frac{h_i}{f},\ i = 1, \ldots, n-1, \tag{3}$$

with various constraints on the g and h_i. For testing irreducibility the following constraints are used: if $\deg_{x_i} f = e_i$ for $i = 1 \ldots n$,

$$\begin{aligned} &\deg_{x_1}(g) \leq e_1 - 2,\ \deg_{x_i}(g) \leq e_i \qquad i = 2 \ldots n-1, \\ &\deg_{x_i}(h_j) \leq \begin{cases} e_{i+1} & \text{if } i \neq j \\ e_{i+1} - 1 & \text{if } i = j \end{cases} \qquad j = 1, \ldots, n-1. \end{aligned}$$

Since (3) is linear in the coefficients of f, for a fixed polynomial basis, we can write a matrix of the equations and denote it $\mathrm{Rup}(f)$. The polynomial f has non-trivial factors exactly if $\mathrm{Rup}(f)$ has a non-trivial nullspace. In fact, the factors can be recovered from the nullspace of a very similar matrix. The matrix $\mathrm{Rup}(f)$ can be improved for polynomials without dense support (Gao and Rodrigues, 2003; May, 2005).

The approximate factorization problem has been studied in Sasaki et al. (1991, 1992); Galligo and Watt (1997); Huang et al. (2000); Sasaki (2001); Galligo and Rupprecht (2001); Corless et al. (2001, 2002); Nagasaka (2002); Kaltofen and May (2003);

Sommese et al. (2004); Gao et al. (2004). To date, computing, in polynomial time, the nearest polynomial which factors is still an open problem.

Univariate Functional Decomposition

Given a univariate polynomial $f \in \mathbb{C}[x]$, we say f is decomposable if we can compute $g, h \in \mathbb{C}[x]$, $\deg h > 1$ so that $f = g \circ h$. For a fixed h, computing g is clearly a linear problem. Due to a theorem of Fried (1970), we have that, except for a small class of special cases, f decomposes exactly when $\Phi(f) = (f(x) - f(y))/(x - y)$ factors over $\mathbb{C}[x, y]$. Thus, f decomposes exactly when $\mathrm{Rup}(\Phi(f))$ has a non-trivial nullspace and the decompostion factors of f can be recovered from the factors of $\Phi(f)$ as in Barton and Zippel (1985).

Two algorithms for computing approximate decompositions is given in Corless et al. (1999) and another approach is presented in Giesbrecht and May (2005). While both of these approaches seem to perform well, neither is guaranteed to find the closest polynomial which decomposes. No polynomial-time algorithm to compute the nearest polynomial which decomposes is known.

Ore Polynomial GCRD

Informally, an Ore polynomial is a polynomial in which the coefficients do not commute with the indeterminant in certain specific ways. Ore polynomials provide a natural generalization of both differential and difference operators, and are used extensively in computer algebra systems for the solution of such equations. We will consider univariate Ore polynomials with coefficients in $\mathbb{C}(x)$ and an indeterminate D. For simplicity, the indeterminant D can be considered to be either a differential operator $D(x^n) = n\,x^{n-1}$ or a shift operator $D(x) = x - 1$ which have commutation rules $D\,x = 1 + x\,D$ and $D\,x = (x + 1)\,D$ respectively. Since there is a Euclidean algorithm for left and right division over Ore polynomial rings, one can define the notions of greatest common right divisor and least common left multiple. As with the commutative case, f and g have a common right divisor d if they have a common left multiple, that is $u\,f = v\,g$ with $\deg_D u < \deg_D g$ and $\deg_D v < \deg_D f$ (Bronstein and Petkovsěk, 1996).

We will consider the simpler case of Ore polynomials with coefficients in $\mathbb{C}[x]$, and note that we can reduce to this case by multiplying to clear denominators. Then we can set up the equations

$$u\,f - v\,g = 0, \tag{4}$$

where

$$u = \sum_{i=0}^{\deg_D g - 1} d^i \sum_{j=0..N} u_{i,j}\,x^j \text{ and } v = \sum_{i=0}^{\deg_D f - 1} d^i \sum_{j=0..N} v_{i,j}\,x^j$$

have symbolic coefficients and $N = 2\max\{\deg_x f, \deg_x g\}(\deg_x f + \deg_x g)$ is an upper bound on the possible degree in x of u and v. As with standard polynomials, (right or left) multiplication by an Ore polynomial over $\mathbb{C}[x]$ is a linear operation and thus can be represented by a matrix with entries in $\mathbb{C}$ for a fixed bivariate basis (e.g. $x^i\,D^j$). Thus, as with the standard polynomial GCD, (4) leads to a matrix which is full rank exactly if f and g have no GCRD. Similar results can be found for the GCLD of Ore polynomials

as well as specified degree GCRDs and GCLDs and hence Ore polynomial left and right division.

While it is quite natural that differential and difference equations be considered with approximate or floating point coefficients, we know of no previous work addressing such questions, at least in a symbolic-numeric framework. A contribution of this work is that such problems can be linearized, and are hence amenable to the structured linear algebra methods, and specifically the Riemannian SVD, presented in Section 3 of this paper.

2. Structured Total Least Squares

The *total least squares problem* is a generalization of the typical linear least squares problems. Give an input matrix $A \in \mathbb{C}^{m \times n}$ of full rank, it finds the "smallest" $A_\triangle \in \mathbb{C}^{m \times n}$ such that $A + A_\triangle$ is of lower rank than A. The size of $A_\triangle$ is measured in terms of the matrix 2-norm. Viewed another way, given a $A \in \mathbb{C}^{m \times n}$ and $b \in \mathbb{C}^m$, the problem seeks $A_\triangle \in \mathbb{C}^{m \times n}$ and $b_\triangle \in \mathbb{C}^m$ such that there exists an $x \in \mathbb{C}^n$ with $(A + A_\triangle)x = b + b_\triangle$, and such that $\|[A_\triangle | b_\triangle]\|_2$ is minimal. The total least squares problem can be solved exactly and efficiently using the usual singular value decomposition algorithm; see Golub and Loan (1995).

The *Structured Total Least Squares* (STLS) problem solves a similar problem to the total least squares problem except that the input matrix A is structured, and we insist that its perturbation $A + A_\triangle$ has the same structure as A. More formally, we define structure by means of a basis $B_1, \ldots, B_\ell \in \mathbb{C}^{m \times n}$ for a vector space $\mathcal{V}$ of structured matrices. Given $A \in \mathcal{V}$, the structured total least squares problems seeks a $b = (b_1, \ldots, b_\ell) \in \mathbb{R}^\ell$ with $\|b\|_2$ minimal, and $y \in \mathbb{R}^n$ with $\|y\|_2 = 1$, such that $(A + \sum b_i B_i)y = 0$.

There are many algorithms for the STLS problem: the constrained total least squares method of Abatzoglou and Mendel (1987) and Abatzoglou and Hackman (1991), the Riemannian SVD of De Moor (1993, 1994), and the structured total least norm (STLN) of Rosen et al. (1996); Van Huffel et al. (1996), are the three primary (and to some extent equivalent) algorithms for this problem. See the thesis of Lemmerling (1999) for an extensive survey and detailed analysis. It should be noted that in general this is a non-convex optimization problem for which none of these algorithms is guaranteed to find a solution.

De Moor (1993), Lemmerling et al. (2002), and Lemmerling (1999) discuss the Structured Total Least Squares, and present heuristic algorithms. While they do not prove their algorithms always gets a minimal solution (indeed, they do not), experimental evaluation seems suggests considerable robustness.

An interesting point of comparison for the Riemannian SVD approach presented in this current paper should be the use of the STLN to approximate low-rank Sylvester matrices by Kaltofen et al. (2005).

Riemannian SVD

In this paper we will consider the application of the Riemannian SVD to the STLS problem. While not particularly efficient, at least not in the general form we employ it, it will have the advantage that it attempts to optimize a norm which corresponds directly to the coefficient 2-norm of the polynomials involved in our problems.

De Moor (1993), Lemmerling (1999) and Lemmerling et al. (2002) show STLS to be equivalent to a non-linear Singular Value Decomposition: Let $A = \sum_i c_i B_i$ be the input matrix. The STLS problem is reduced to finding the triplet (u, τ, v) corresponding to the smallest τ such that

$$\begin{aligned} Av &= D_v u\tau \qquad & u^T D_v u &= 1, \\ A^T u &= D_u v\tau \qquad & v^t D_u v &= 1, \\ v^T v &= 1. \end{aligned}$$

Here D_u and D_v have a quadratic dependence on u, v respectively. Ultimately, $y = v$ and $\hat{c_i} = c_i - u^T T_i v\tau$ form a solution to the STLS problem. In order to determine (u, τ, v), we hold D_u, D_v fixed, and perform an inverse iteration to determine updated vectors u, v. Each iteration is solved by putting the constraints of the RiSVD into a large linear system, and solving for u, v. D_u, D_v are then updated, and another iteration is performed. The linear system is derived from the full QR decomp. of A,

$$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R \\ 0 \end{bmatrix}.$$

We construct the system to be lower-triangular:

$$\begin{bmatrix} R^T & 0 & 0 \\ Q_2^T D_v Q_1 & Q_2^T D_v Q_2 & 0 \\ Q_1^T D_v Q_1 \tau & Q_1^T D_v Q_2 \tau & -R \end{bmatrix} \begin{bmatrix} z \\ w \\ v \end{bmatrix} = \begin{bmatrix} D_u v\tau \\ 0 \\ 0 \end{bmatrix}$$

where $u = Q_1 z + Q_2 w$.

As we have noted, this is not a particularly efficient algorithm as stated. It uses the basis for the structured matrices, and does not take advantage of its form or do any normalization. This does have the advantage that it makes for a widely applicable tool. De Moor (1994) notes that its application may be most effective when the algorithm is specialized to specific matrix structures.

3. Experiments: GCD

An unoptimized implementation of RiSVD has been written in Maple 9 and discussed in detail in Botting (2004). In order to use it to compute approximate GCDs we need to first compute a structure basis B. For a pair of polynomials of fixed degree, it is trivial to compute B in Maple by first creating a symbolic Sylvester matrix:

```
M:= LinearAlgebra:-Transpose(LinearAlgebra:-SylvesterMatrix(
         add(a[i]*x^i ,i=0..degree(f,x)),add(b[i]*x^i,
               i=0..degree(g,x)), x));

 B:=[seq(subs([a[i]=1, seq(a[j]=0,j=0..degree(f,x)),
          seq(b[j]=0,j=0..degree(g,x)) ],M), i=0..degree(f,x)),
          seq(subs([b[i]=1, seq(a[j]=0,j=0..degree(f,x)),
          seq(b[j]=0,j=0..degree(g,x)) ],M), i=0..degree(g,x))]:
```

For the following table, we generated pairs of polynomials with machine size floating point coefficients between -100 and 100, of degree 5 or 4 with an exact GCD of degree 1 or 2 and then added a random "noise" polynomial of equal degree and norm $\approx 10^{-4}$ to each. On 100 of such pairs, we ran the RiSVD algorithm, the QRGCD (Corless et al., 2004) algorithm as implemented in maple and an approximate GCD algorithm using SVD to approximate a solution to the STLS problem (similar to algorithms discussed in Corless et al. (1995) and Zeng (2004)). Using the approximate division algorithm in Maple's SNAP as necessary, we computed a d, u and v so that $d\,u - g \approx 0$ and $d\,v - f \approx 0$. Instead of measuring the quality of the triplet (d, u, v), we performed an iterative refinement as in Zeng (2004) to find a triplet so that $err = \|d\,u - g\|^2 + \|d\,v - f\|^2$ is locally minimal. The results of these experiments are presented in the following table:

Method	Average CPU time	Avg. err	# of Failures
QRGCD	0.1s	0.28980×10^{-6}	3
SVD GCD	0.05s	0.28980×10^{-6}	1
RiSVD GCD	1.9s	0.17115×10^{-6}	16

In the last column, a failure is any example in which the algorithm failed to produce an err of less than 10^{-4}. Only examples where all three algorithms succeeded are counted towards the averages in the second and third columns.

As we can see, the SVD and QRGCD algorithms are quite a bit faster, but clearly does not produce optimal results; the RiSVD method offers a slight improvement. The large number of RiSVD failures are due to some of the approximate GCDs of degree two having complex roots. The RiSVD implementation always looks for the GCD of degree one, and wince, the implementation is rational, we cannot possibly converge to the optimal complex GCD.

In the following table, we used examples as in the previous table, but this time allowed only GCDs of degree one.

Method	Average CPU time	Avg. err	# of Failures
QRGCD	0.09s	$0.178754226 \times 10^{-6}$	4
SVD GCD	0.04s	$0.178751501 \times 10^{-6}$	1
RiSVD GCD	2.02s	$0.178748409 \times 10^{-6}$	0

In none of the above examples where all three algorithms succeeded did SVD or QRGCD find a GCD with smaller err than the one found by RiSVD, though we have found that RiSVD is occasionally beaten by the other two algorithms especially in higher degree examples.

In practice, RiSVD seems to produce very good numerically singular structured matrices (Botting, 2004). However, it is still necessary to develop robust methods to recover the answers we desire if they cannot be read directly from the entries of the matrix or its null-vectors. Given the improvement over QRGCD, we are optimistic that RiSVD will prove useful for other approximate polynomial problems as well.

For the final version of this paper we will provide an extensive empirical evaluation of the Riemannian SVD on all the above listed problems, in comparison to other approaches (which are typically specific to the problem under consideration).

References

T. Abatzoglou and R. Hackman. Constrained total least squares based extraction of acoustic resonances of elastic targets. In *Proc. IEEE Int. Conf Acoustics, Speech and Signal Processing*, pages 3385–3388, 1991.

T. Abatzoglou and J. Mendel. Constrained total least sqares. In *Proc. IEEE Int. Conf Acoustics, Speech and Signal Processing*, pages 1485–1488, 1987.

D. Barton and R. Zippel. Polynomial decomposition algorithms. *J. Symbolic Comput.*, 1: 159–168, 1985.

B. Beckermann and G. Labahn. A fast and numerically stable Euclidean-like algorithm for detecting relative prime numerical polynomials. *J. Symbolic Comput.*, 26(6):691–714, 1998a. Special issue on Symbolic Numeric Algebra for Polynomials S. M. Watt and H. J. Stetter, editors.

B. Beckermann and G. Labahn. When are two numerical polynomials relatively prime? *J. Symbolic Comput.*, 26(6):691–714, 1998b. Special issue on Symbolic Numeric Algebra for Polynomials S. M. Watt and H. J. Stetter, editors.

B. Botting. Structured total least squares for approximate polynomial operations. Master's thesis, School of Computer Science, University of Waterloo, 2004.

M. Bronstein and M. Petkovsěk. An introduction to pseudo-linear algebra. In *Selected papers of the conference on Algorithmic complexity of algebraic and geometric models*, pages 3–33, Amsterdam, The Netherlands, 1996. Elsevier Science Publishers B. V.

R. Corless, A. Galligo, I. Kotsireas, and S. Watt. A geometric-numeric algorithm for absolute factorization of multivariate polynomials. In *Proc. 2002 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'02)*, pages 37–45, 2002.

R. Corless, P. Gianni, B. Trager, and S. Watt. The singular value decomposition for polynomial systems. In *Proc. 1995 Internat. Symp. Symbolic Algebraic Comput. ISSAC'95*, pages 96–103, 1995.

R. Corless, M. Giesbrecht, M. van Hoeij, I. Kotsireas, and S. Watt. Towards factoring bivariate approximate polynomials. In *ISSAC 2001 Proc. 2001 Internat. Symp. Symbolic Algebraic Comput.*, pages 85–92, 2001.

R. Corless, M. Giesbrecht, D. Jeffrey, and S. Watt. Approximate polynomial decomposition. In Samuel S. Dooley, editor, *Internat. Symp. Symbolic Algebraic Comput.*, pages 213–220, Vancouver, Canada, 1999. ACM Press.

R. Corless, S. Watt, and L. Zhi. Qr factoring to compute the gcd of univariate approximate polynomials. *IEEE Transactions on Signal Processing*, 52(12):3394–3402, December 2004.

B. De Moor. Structured total least squares and ℓ_2 approximation problems. *Linear algebra and its applications*, 188–189:163–207, 1993.

B. De Moor. Total least squares for affinely structured matrices and the noisy realization problem. *IEEE Transactions on Signal Processing*, 42:3004–3113, 1994.

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

M. Fried. On a conjecture of Schur. *Mich. Math. Journal*, 17:41–55, 1970.

A. Galligo and D. Rupprecht. Semi-numerical determination of irreducible branches of a reduced space curve. In *ISSAC 2001 Proc. 2001 Internat. Symp. Symbolic Algebraic Comput.*, pages 137–142, 2001.

A. Galligo and S. Watt. A numerical absolute primality test for bivariate polynomials. In *ISSAC 97 Proc. 1997 Internat. Symp. Symbolic Algebraic Comput.*, pages 217–224, 1997.

S. Gao. Factoring multivariate polynomials via partial differential equations. *Math. Comput.*, 72(242):801–822, 2003.

S. Gao, E. Kaltofen, J. May, Z. Yang, and L. Zhi. Approximate factorization of multivariate polynomials via differential equations. In *ISSAC 2004 Proc. 2004 Internat. Symp. Symbolic Algebraic Comput.*, pages 167–174, 2004.

S. Gao and V. Rodrigues. Irreducibility of polynomials modulo p via Newton polytopes. *J. Number Theory*, 101:32–47, 2003.

M. Giesbrecht and J. May. New algorithms for exact and approximate polynomial decomposition. In *Proc. International Workshop on Symbolic-Numeric Computation (SNC'05)*, 2005. To appear.

G. Golub and C. Van Loan. *Matrix Computations*. Johns Hopkins, 3rd edition, 1995.

Y. Huang, W. Wu, H. Stetter, and L. Zhi. Pseudofactors of multivariate polynomials. In *Internat. Symp. Symbolic Algebraic Comput. ISSAC 2000 Proc. 2000 Internat. Symp. Symbolic Algebraic Comput.*, pages 161–168, 2000.

E. Kaltofen and J. May. On approximate irreducibility of polynomials in several variables. In *ISSAC 2003 Proc. 2003 Internat. Symp. Symbolic Algebraic Comput.*, pages 161–168, 2003.

E. Kaltofen, Z. Yang, and L. Zhi. Structured low rank approximation of a sylvester matrix. In *Proc. International Workshop on Symbolic-Numeric Computation (SNC'05)*, 2005. To appear.

N. Karmarkar and Lakshman Y. N. On approximate GCDs of univariate polynomials. *J. Symbolic Comput.*, 26(6):653–666, 1998. Special issue on Symbolic Numeric Algebra for Polynomials S. M. Watt and H. J. Stetter, editors.

P. Lemmerling. *Structured total least squares: analysis, algorithms and applications*. PhD thesis, K.U. Leuven (Leuven, Belgium), 1999.

P. Lemmerling, S Van Huffel, and B. De Moor. The structured total least squares approach for nonlinearly structured matrices. *Numerical Linear Algebra with Applications*, 9(1–4):321–332, 2002.

T.Y. Li and Z. Zeng. A rank revealing method and its applications. *SIAM J. Matrix Anal. Appl.*, 26(4):918–946, 2005.

J. May. *Approximate Factorization of Polynomials in Many Variables and Other Problems in Approximate Algebra via Singular Value Decomposition Methods*. PhD thesis, North

Carolina State University, 2005.

K. Nagasaka. Towards certified irreducibility testing of bivariate approximate polynomials. In *Proc. 2002 Internat. Symp. Symbolic Algebraic Comput. (ISSAC'02)*, pages 192–199, 2002.

M.-A. Ochi, M. Noda, and T. Sasaki. Approximate greatest common divisors of multivariate polynomials and its application to ill-conditioned systems of algebraic equations. *J. Inf. Proces*, 12:292–300, 1991.

J. Rosen, H. Park, and J. Glick. Total least norm formulation and solution for structured problems. *SIAM Journal on Matrix Anal. Appl.*, 1996.

W. Ruppert. Reducibility of polynomials *f(x,y)* modulo *p*. *J. Number Theory*, 77:62–70, 1999.

T. Sasaki. Approximate multivariate polynomial factorization based on zero-sum relations. In *ISSAC 2001 Proc. 2001 Internat. Symp. Symbolic Algebraic Comput.*, pages 284–291, 2001.

T. Sasaki, T. Saito, and T. Hilano. Analysis of approximate factorization algorithm I. *Japan J. of Industrial and Applied Mathem.*, 9(3):351–368, October 1992.

T. Sasaki and M. Sasaki. Polynomial remainder sequence and approximate GCD. *ACM SIGSAM Bulletin*, 31:4–10, 1997.

T. Sasaki, M. Suzuki, M. Kolář, and M. Sasaki. Approximate factorization of multivariate polynomials and absolute irreducibility testing. *Japan J. of Industrial and Applied Mathem.*, 8(3):357–375, October 1991.

A. Schönhage. Quasi-GCD computations. *J. Complexity*, 1:118–137, 1985.

A. Sommese, J. Verschelde, and C. Wampler. Numerical factorization of multivariate complex polynomials. *Theoretical Comput. Sci.*, 315(2–3):651–669, 2004. Special issue on Algebraic and Numerical Algorithms.

S. Van Huffel, H. Park, and J. Rosen. Formulation and solution of structured total least norm problems for parameter estimation. *IEEE Trans on Signal Processing*, 44(10): 2464–2474, 1996.

Z. Zeng. The approximate GCD of inexact polynomials part I: a univariate algorithm. Manuscript, 8 pages, 2004.

Z. Zeng and B. Dayton. The approximate GCD of inexact polynomials part II: a multivariate algorithm. In *ISSAC 2004 Proc. 2004 Internat. Symp. Symbolic Algebraic Comput.*, 2004.

L. Zhi. Displacement structure in computing approximate GCD of univariate polynomials. In Ziming Li and William Sit, editors, *Proc. Sixth Asian Symposium on Computer Mathematics (ASCM 2003)*, volume 10 of *Lecture Notes Series on Computing*, pages 288–298, Singapore, 2003. World Scientific.

L. Zhi, K. Li, and M.-T. Noda. Approximate GCD of multivariate polynomials using Hensel lifting. MMRC Preprint, 7 pages, 2001.

L. Zhi and M.-T. Noda. Approximate GCD of multivariate polynomials. In *Proc. Fourth Asian Symposium on Computer Mathematics (ASCM 2000)*, pages 9–18, 2000.

Brad Botting
School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
e-mail: bbotting@cs.uwaterloo.ca

Mark Giesbrecht
School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
e-mail: mwg@uwaterloo.ca

John May
School of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
e-mail: jpmay@uwaterloo.ca