Paola Boito

Structured Matrix Based Methods for Approximate Polynomial GCD

EDIZIONI
DELLA
NORMALE

15

TESI

THESES

tesi di perfezionamento in Matematica sostenuta il 5 ottobre 2007

COMMISSIONE GIUDICATRICE
Mariano Giaquinta, Presidente
Bernhard Beckermann
Dario Bini
Luca Gemignani
Patrizia Gianni
Stefano Marmi
Alessandro Profeti

Paola Boito
Université de Limoges / CNRS
123 avenue Albert Thomas
87060 Limoges Cedex, France

Structured Matrix Based Methods for Approximate Polynomial GCD

Paola Boito

Structured Matrix Based Methods for Approximate Polynomial GCD

EDIZIONI
DELLA
NORMALE

ISBN: 978-88-7642-380-2
e-ISBN: 978-88-7642-381-9

“He found him under a pine tree, sitting on the ground, arranging fallen pine cones in a regular design: an isosceles triangle. At that hour of dawn Agilulf always needed to apply himself to some precise exercise: counting objects, arranging them in geometric patterns, resolving problems of arithmetic. It was the hour in which objects lose the consistency of shadow that accompanies them during the night and gradually reacquire colors, but seem to cross meanwhile an uncertain limbo, faintly touched, just breathed on by light; the hour in which one is least certain of the world’s existence. He, Agilulf, always needed to feel himself facing things as if they were a massive wall against which he could pit the tension of his will, for only in this way did he manage to keep a sure consciousness of himself. But if the world around was instead melting into the vague and ambiguous, he would feel himself drowning in that morbid half light, incapable of allowing any clear thought or decision to flower in that void. In such moments he felt sick, faint; sometimes only at the cost of extreme effort did he feel himself able to avoid melting away completely. It was then he began to count: trees, leaves, stones, lances, pine cones, anything in front of him. Or he put them in rows and arranged them in squares and pyramids.”

Italo Calvino, *The Nonexistent Knight*

Contents

Introduction

The computation of polynomial GCD is a basic algebraic task, which has many applications in several fields such as in polynomial root-finding, control theory, image deblurring, CAGD.

The problem is usually stated as follows: given the coefficients of two polynomials $u(x)$ and $v(x)$, compute the coefficients of their greatest common divisor $g(x)$. We focus here on the univariate case.

The usual notion of polynomial GCD, however, is ill-suited to deal with many applications where input data are affected by errors (due for example to roundoff, or to the fact that the data come from physical experiments or previous numerical computations). Indeed, if the given polynomials $u(x)$ and $v(x)$ have a nontrivial GCD, then arbitrarily small perturbations in the coefficients of $u(x)$ and $v(x)$ may transform $u(x)$ and $v(x)$ into relatively prime polynomials. Therefore the problem of finding an exact GCD is ill-posed in an approximate setting.

This is why the notion of approximate GCD has been introduced. Several definitions of approximate GCD are found in the literature; here we will mostly use the so-called ϵ-GCD. Roughly speaking, a polynomial $g(x)$ is an ϵ-GCD of $u(x)$ and $v(x)$ if there exist polynomials $\hat{u}(x)$ and $\hat{v}(x)$ such that

(i) $\hat{u}(x)$ and $\hat{v}(x)$ are "close" to $u(x)$ and $v(x)$, that is, $d(u, \hat{u}) < \epsilon$ and $d(v, \hat{v}) < \epsilon$ for some fixed polynomial metric d and tolerance ϵ;
(ii) $g(x)$ is an exact GCD of $\hat{u}(x)$ and $\hat{v}(x)$, and
(iii) $g(x)$ has maximum degree among the exact GCDs of pairs of polynomials that satisfy (i).

The first analysis of the approximate GCD problem dates back to 1985 ([118]); several approaches to the problem have been proposed since then. We seek here to give a comprehensive overview of the existing literature on the subject, with a focus on matrix-based methods. We next explore in detail the relationship between approximate GCD and resultant matrices (namely, Sylvester, Bézout and companion resultant matrices), their properties and factorizations. Three new algorithms for the computation of ϵ-GCD are presented:

- the algorithm TdBez uses Householder tridiagonalization of the Bézout matrix as its main tool;
- the algorithm PivQr is based on QR decomposition of the Sylvester matrix and subresultants, stabilized by column pivoting;
- the algorithm FastGcd exploits the Toeplitz-like structure of the Sylvester and Bézout matrices to compute an ϵ-GCD in a stable way and with a computational cost that is quadratic in the degrees of the input polynomials (whereas the complexity of known stable methods is at least cubic).

Chapters 1 to 5 present the definitions and formulations of the approximate GCD problem that can be found in the literature and outline the main ideas in the approaches proposed so far. Chapter 1 lists the definitions of quasi-GCD, ϵ-GCD, AGCD and δ-GCD and presents related topics that give useful insight into the approximate GCD problem, such as ϵ-root neighborhoods and a graph-theoretical description of approximate GCD.

Chapter 2 introduces the main tools that are necessary in a matrix-based approach to the polynomial GCD problem. Resultant matrices are defined and their relationship with polynomial GCD is examined. In particular, we prove some norm inequalities and a result on the QR decomposition of the Bézout matrix that will be useful later on. Resultant matrices have remarkable structure properties; therefore part of the chapter is devoted to a discussion of displacement structure and the fast method GKO for the factorization of structured matrices.

Chapter 3 analyses the use of variants of the Euclidean algorithm to compute an approximate GCD or estabilish approximate coprimeness. The basic version of the Euclidean algorithm has a low computational cost (*i.e.*, quadratic in the degree of the input polynomials), but it is numerically unstable. Several stabilized versions of the algorithm have been proposed; in some of them the crucial point is the choice of the termination criterion, whereas in other cases a look-ahead technique is employed to avoid ill-conditioned steps.

Chapter 4 is devoted to a description of known results and algorithms that rely on factorizations of the Sylvester matrix and subresultants. The singular value decomposition of the Sylvester matrix is often used to obtain estimates on the approximate rank of a resultant matrix, and therefore on the degree of an approximate GCD. The QR decomposition of the Sylvester matrix has been used in [41] to compute an approximate polynomial GCD, whereas the method outlined in [144] relies on the QR decomposition of Sylvester subresultants.

In most cases, algorithms for the computation of ϵ-GCD take a pair of polynomials and a tolerance ϵ as input, and output an ϵ-GCD. An alternative approach involves taking polynomials and the approximate GCD degree as input, and trying to minimize the norm of the perturbation that should be applied to the given polynomials so that they have an exact GCD of the prescribed degree. This is often called the optimization approach, and it is examined in Chapter 5.

Chapters 6 and 7 are devoted to the presentation of new methods for approximate GCD. The algorithms TdBez and PivQr are described in detail in Chapter 6, along with a study of the QR decomposition of resultant matrices and of the tridiagonalization of the Bezoutian. At the end of the chapter, three more algorithms for approximate GCD are briefly proposed. We feel that the algorithms described in this chapter, besides being quite effective, have the merit of highlighting some aspects of the interplay between resultant matrices and polynomial GCD that have been overlooked in the literature.

The main feature of the algorithm Fastgcd, presented in Chapter 7, is its low computational cost, combined with good stability properties. We show how a stabilized version of the GKO algorithm for the LU factorization of displacement structured matrices can be used to estimate the approximate GCD degree, compute the approximate GCD coefficients along with the associated cofactors, and perform iterative refinement.

The new algorithms presented here have been implemented in Matlab and applied to many test polynomials in order to evaluate the performance of these algorithms on typical “difficult” cases. Chapter 8 shows the most significant among these numerical experiments and compares the performance of our algorithms with the results given by other methods for which an implementation is available. A comparison between the notions of ϵ-GCD (based on perturbation of polynomial coefficients) and δ-GCD (based on perturbation of polynomial roots, see [103]) is also given in the last section.

Finally, Chapter 9 gives an overview of the many possible generalizations of the approximate GCD problem, as well as indications on further work.

Acknowledgements

First and foremost, I would like to thank my advisor, Dario Bini. I am grateful for his constant help and inspiring ideas, and for all the time and effort he devoted to the supervision of this thesis. Luca Gemignani also gave a remarkable contribution to this work, both through useful suggestions and his thorough knowledge of the existing literature.

Part of the research work that led to this thesis was supported by the PRIN04 project “Structured matrix analysis: numerical methods and applications”.

I would like to thank Victor Pan for suggesting the comparison between ϵ-GCD and δ-GCD in Section 8.10 and Zhonggang Zeng for sending me his software for approximate polynomial GCD.

I greatly appreciated the help that Filippo Callegaro, Giuseppe Della Sala, Antongiulio Fornasiero and Laura Luzzi gave me with technical issues. I also wish to mention Ivan Markowski, with whom I had stimulating discussions on the topic of polynomial GCD. My heartfelt thanks go to Roberto Grena, both for technical help and for his invaluable patience and support. Finally, I would like to thank my parents and all the people who have been close to me and have helped me throughout these years.

Pisa, 22 February 2007

Having defended my thesis, I would like to thank all the members of the committee and the referees, and in particular Bernhard Beckermann, who carefully examined this work and offered many insightful remarks. Most of the improvements in this revised version are due to his suggestions.

I would also like to thank Patrizia Gianni and Rob Corless for their helpful comments. I wish to mention Joab Winkler and John Allan: the summer school they organized in Oxford in September 2007 was for me a remarkable opportunity to learn about the latest developments on the subject of polynomial GCD and to meet people with similar interests.

Seeing my thesis published by Le Edizioni della Normale is a privilege I am very grateful for. My thanks go to the invaluable Luisa Ferrini, to Giuseppe Tomassini, to the referees and to all the editorial team.

Limoges, 22 April 2011

Notation

Fields of numbers are denoted here, as usual, by $\mathbb{R}$ (real) and $\mathbb{C}$ (complex). The imaginary unit is denoted by $\hat{\imath}$ (so as not to be confused with the letter i used as an index).

The vector space of real vectors of length n is denoted by $\mathbb{R}^n$ and the vector space of real $m \times n$ matrices is denoted by $\mathbb{R}^{m\times n}$. Analogously, $\mathbb{C}^n$ and $\mathbb{C}^{m\times n}$ are the vector spaces of complex n-vectors and $m \times n$ matrices, respectively.

We use capital letters (*e.g.* A, B, C) for matrices and boldface lower case letters for (column) vectors (*e.g.* $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$); the identity matrix of order n is denoted by I_n, whereas I is used when the size of the identity matrix is not explicitly specified. For diagonal matrices, the notation $A = \operatorname{diag}(a_1, \dots, a_n)$ stands for

$$A = \begin{pmatrix} a_1 & & \text{O} \\ & \ddots & \\ \text{O} & & a_n \end{pmatrix}.$$

Transpose matrices and vectors are denoted by A^T, $\mathbf{u}^T$, whereas the notation A^*, $\mathbf{u}^*$ is used to indicate transpose in the real case and Hermitian adjoint (*i.e.*, conjugate transpose) in the complex case.

Unless otherwise specified, a Matlab[1]-like notation is used for submatrices and matrix entries. Namely:

- $A(i, j)$ is the entry of A that belongs to the i-th row and to the j-th column;
- $A(i, :)$ is the i-th row of A and $A(:, j)$ is the j-th column of A;

[1] Matlab is a registered trademark of The MathWorks, Inc.

- $A(m{:}n,\ p{:}q)$ is the submatrix of A formed by the intersection of rows m to n and columns p to q.

Where explicitly stated, $\mathbf{u}$ denotes the column vector of length $n+1$ associated with a univariate polynomial $u(x) = \sum_{i=0}^{n} u_i x^i$, *i.e.*, $\mathbf{u} = [u_0, u_1, \ldots, u_n]^T$. See Section B.1 for notation in the multivariate case.

First-order approximations are sometimes denoted by $\doteq$ and $\dot{<}$, so that $a \doteq b$ and $c \dot{<} d$ mean $a = b + \mathcal{O}(\epsilon^2)$ and $c \leq d + \mathcal{O}(\epsilon^2)$ respectively.

Chapter 1
Approximate polynomial GCD

Finding the greatest common divisor (GCD) of two given polynomials is a basic problem in algebraic computing. The problem is usually stated as follows: given the (real or complex) coefficients of two polynomials, compute the coefficients of their greatest common divisor.

The range of applications is very wide; we mention here some examples.

- *Polynomial root-finding.* Computing the roots of a polynomial $p(x)$ which has multiple roots is an ill-conditioned problem. If a robust GCD finder is available, computing $g(x) = \mathrm{GCD}(p, p')$ may help to solve this difficulty, since the roots of the polynomial $p(x)/g(x)$ will turn out to be better conditioned.
- *Simplifying rational functions.* Representing or performing computations with a rational function $R(x) = a(x)/b(x)$ might require $a(x)$ and $b(x)$ to be coprime. If $g(x) = \mathrm{GCD}(a, b)$ is computed, then $R(x)$ can be replaced by $\tilde{R}(x) = \tilde{a}(x)/\tilde{b}(x)$, where $\tilde{a}(x) = a(x)/g(x)$ and $\tilde{b}(x) = b(x)/g(x)$. An application is degree reduction of rational curves, such as Bézier curves (see *e.g.* [120] and [18]).
- *Control theory.* Polynomial coprimeness is related to the controllability of linear control systems (see [7]).
- *Image restoration.* Polynomial GCD computations can be used for blind image deblurring (see [108]).

We will be mainly concerned here with the problem of evaluating the GCD of univariate polynomials $u(x)$ and $v(x)$.

The problem is well-understood in the exact case, that is, under the assumption that the coefficients of $u(x)$ and $v(x)$ are error-free. However, in many applications, input data are represented as floating point numbers or derive from the results of physical experiments or previous computation, so that they are generally affected by errors. The application of ordinary polynomial computations to such *empirical polynomials* is a field of study comprising elements from computer algebra and numerical analysis, to which a considerable amount of work has lately been devoted; see [121] for a review and further bibliography.

In our case, if $u(x)$ and $v(x)$ have a nontrivial GCD, it turns out that arbitrarily small perturbations in the coefficients of $u(x)$ and $v(x)$ may transform $u(x)$ and $v(x)$ into relatively prime polynomials. Therefore, it is clear that the concept of GCD is not well suited to deal with applications where data are approximatively known. This is why the notion of *approximate GCD* has been introduced.

1.1. Coefficient-based definitions

Starting from Schönhage ([118]), several different definitions of approximate polynomial GCD are found in the literature. The common underlying idea, however, is to look for a pair of polynomials $\hat{u}(x)$ and $\hat{v}(x)$ which are “close” to $u(x)$ and $v(x)$ and have a nontrivial exact GCD of maximum degree. The precise meaning of “close” depends both on the technical details of the definition and on the choice of a tolerance $\epsilon > 0$, which is related to the magnitude of the errors that may affect the coefficients of $u(x)$ and $v(x)$.

Throughout this work, the expression *approximate GCD* will be used to denote any of the different polynomials (quasi-GCD, ϵ-GCD, AGCD, δ-GCD) defined in the following sections, whereas the specific denomination will be used when appropriate. Notice that, while the acronym AGCD used by Zeng in [144] (see Section 1.1.3) actually stands for “approximate GCD”, the abbreviated form will be reserved for Zeng’s definition, so that the expression *approximate GCD* keeps its generic meaning.

1.1.1. Quasi-GCD

The first formalization of the notion of approximate GCD (*quasi-GCD*) is due to Schönhage ([118]) and dates back to 1985.

The definition is given at first for homogeneous polynomials, in order to account for the fact that a system of polynomial equations may have solutions close to (or at) infinity, which corresponds to nearly vanishing leading coefficients. Let

$$A(z_0, z_1) = \sum_{i=0}^{n} \alpha_i z_0^{n-i} z_1^i,$$

$$B(z_0, z_1) = \sum_{j=0}^{m} \beta_j z_0^{m-j} z_1^j,$$

with degrees $1 \leq n \leq m$. Define a polynomial norm as

$$|A| = \sum_{i=0}^{n} |\alpha_i|,$$

$$|B| = \sum_{i=0}^{m} |\beta_j|,$$

i.e. $|\cdot|$ is induced by the 1-norm applied to the vector of coefficients. It is also convenient to assume some kind of normalization on A and B, such as

$$|A|, |B| \in \left[\frac{1}{2}, 1\right].$$

Definition 1.1.1. Given $\epsilon > 0$, a homogeneous polynomial $H(z_0, z_1)$ of degree k is called a quasi-GCD of A and B within error ϵ if:

- there exist homogeneous polynomials A_1 of degree $n - k$ and B_1 of degree $m - k$ such that $|HA_1 - A| < \epsilon$ and $|HB_1 - B| < \epsilon$;
- for any exact common divisor D of A and B there exists a homogeneous polynomial Q of degree $k - \deg D$ such that $|DQ - H| < \epsilon|H|$.

It is convenient, however, to reduce Definition 1.1.1 to the case of ordinary univariate polynomials. Let $f(z)$ and $g(z)$ be polynomials of degree n and m, respectively, with $m < n$, and let $0 < \epsilon \leq 1/2$. Let $\rho(f)$ be the root radius of $f(z)$, that is,

$$\rho(f) = \max_{i=1,\dots,n} \{|z_i| \quad \text{such} \quad \text{that} \quad f(z_i) = 0\}.$$

As a normalization condition, assume that $|f|, |g| \in [\frac{1}{2}, 1]$ and f has bounded root radius, *e.g.* $\rho(f) \leq 1/4$.

Definition 1.1.2. A polynomial $h(x)$ is a quasi-GCD for f and g with tolerance ϵ if there are polynomials $u(z)$ and $v(z)$ of degree $m - 1$ and $n - 1$ respectively, such that:

- $|hf_1 - f| < \epsilon$, $|hg_1 - g| < \epsilon$ for suitable f_1, g_1;
- $|uf + vg - h| < \epsilon|h|$.

The problem of quasi-GCD computation can therefore be stated as follows: given the coefficients of polynomials $f(z)$ and $g(z)$ and given ϵ as above, compute the coefficients of polynomials $h(z)$, $u(z)$ and $v(z)$ that satisfy Definition 1.1.2.

Schönhage proposes and discusses an algorithm for quasi-GCD computation, based on a modification of the Euclidean algorithm with pivoting. This part of his pioneering work, while theoretically interesting, is of little use for practical purposes, because input numbers are assumed to be available at any desired precision. In other words, if a number $\alpha \in \mathbb{R}$ belongs to the input set, then an *oracle* called with an arbitrary parameter s will deliver a rational number a such that $|\alpha - a| < 2^{-s}$. This is hardly the case in most practical applications, where the input polynomials are known only to a limited accuracy, once and for all.

1.1.2. ϵ−GCD

We will present in this section the definition of approximate GCD that is most widely used in the literature (see *e.g.* [40, 49, 103, 63]). This is also the definition that is used in most cases throughout this work.

Definition 1.1.3. Let $u(x)$ and $v(x)$ be univariate (real or complex) polynomials, with $n = \deg u(x)$ and $m = \deg v(x)$. Choose $\|\cdot\|$ a polynomial norm (see Section A.3) and ϵ a positive real number. Then a polynomial $g(x)$ is called

- an *ϵ-divisor* of $u(x)$ and $v(x)$ if there exist perturbed polynomials $\hat{u}(x)$ and $\hat{v}(x)$ such that

$$\begin{aligned} &\deg \hat{u}(x) \leq n, \\ &\deg \hat{v}(x) \leq m, \\ &\|\hat{u}(x) - u(x)\| \leq \epsilon, \qquad (1.1.1) \\ &\|\hat{v}(x) - v(x)\| \leq \epsilon \qquad (1.1.2) \end{aligned}$$

  and $g(x)$ is an exact divisor of $\hat{u}(x)$ and $\hat{v}(x)$;
- an *ϵ-GCD* of $u(x)$ and $v(x)$ if it is an ϵ-divisor of maximum degree.

A few comments about this definition are needed.

First of all, notice that the definition requires to choose a polynomial norm. A common choice is the 2-norm of the vector of coefficients, or another vector-induced norm; however, for some purposes one might want to use a different norm, or even a polynomial distance not necessarily induced by a norm. See Section A.3 for a brief discussion of this topic.

It should also be observed that several authors (*e.g.* [103]) prefer to use a *normalized version* of Definition 1.1.3, replacing (1.1.1) and (1.1.2) with

$$\begin{aligned} \|\hat{u}(x) - u(x)\| &\leq \epsilon \|u(x)\|, \\ \|\hat{v}(x) - v(x)\| &\leq \epsilon \|v(x)\|. \end{aligned}$$

Lastly, it is important to point out that the ϵ-GCD, as defined here, is not unique. More precisely, its degree is uniquely defined, but its coefficients are not. This does not only happen because of the lack of normalization requirements on $g(x)$; there might be – and usually are – several polynomials that satisfy Definition 1.1.3, even without being scalar multiples of each other.

1.1.3. AGCD

In [144], Zhonggang Zeng points out that an approximate polynomial GCD (AGCD) for a set of polynomials should exhibit the following characteristics:

1. *nearness*: an AGCD is the exact GCD of a set of polynomials which are close to the given ones;
2. *maximum degree*: an AGCD has maximum degree among the polynomials that satisfy (1);
3. *minimum distance*: an AGCD minimizes the distance between the given set of polynomials and the set of polynomials of which it is the exact GCD.

Nearness and maximum degree are the key ideas shared by all the definitions of approximate GCD. Minimum distance is not always addressed in the literature, but it can certainly be desirable, though maybe difficult to achieve or check with certainty.

In order to achieve nearness, maximum degree and minimum distance, Zeng describes the AGCD problem as follows. Let $p_1(x), \ldots, p_l(x)$ be polynomials of degrees $m_1, \ldots, m_l$ respectively.

Saying that a polynomial $u(x)$ is an exact common divisor of fixed degree k for the p_i's means that there exist polynomials $v_1(x), \ldots, v_l(x)$ such that $p_i(x) = u(x)v_i(x)$ for all $i = 1, \ldots, l$. But these equations characterize $u(x)$ only up to multiplication by a constant; so one might want to add some normalization condition on $u(x)$, which can be expressed as $\mathbf{r}^*\mathbf{u} = 1$ for some given vector $\mathbf{r}$. For example, if $u(x)$ is expected to be monic, then $\mathbf{r}$ will be chosen as $[1 \quad 0 \quad \ldots \quad 0]^*$. So one obtains the following system:

$$F(\mathbf{z}) = b, \tag{1.1.3}$$

where

$$F(\mathbf{z}) = \begin{bmatrix} \mathbf{r}^H\mathbf{u} - 1 \\ C_k(v_1)\mathbf{u} \\ \vdots \\ C_k(v_l)\mathbf{u} \end{bmatrix}, \qquad \mathbf{z} = \begin{bmatrix} \mathbf{u} \\ \mathbf{v_1} \\ \vdots \\ \mathbf{v_l} \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 0 \\ \mathbf{p_1} \\ \vdots \\ \mathbf{p_l} \end{bmatrix}$$