Computation of Approximate Polynomial Gcds and an Extension.

Victor Y. Pan
Mathematics and Computer Science Department
Lehman College, City University of New York
Bronx, NY 10468
Internet: VPAN@LCVAX.LEHMAN.CUNY.EDU
(Supported by NSF Grant CCR 9625344
and PSC CUNY Award No. 667340)
February 6, 2002

Abstract

Computation of approximate polynomial gcds is important both theoretically and due to applications, in particular to linear control systems. We study two approaches to the solution so far omitted by the researchers, in spite of intensive recent work in this area. Correlation to numerical Padé approximation enabled us to improve computations for both problems (gcds and Padé). Reduction to approximating polynomial zeros enabled us to obtain new insight into the gcd problem and to devise effective solution algorithms. In particular, unlike the known algorithms, we compute an upper bound on the degree of approximate gcds at a low computational cost, and this enables us to certify the correctness of the solution. We also argue in favor of restating the problem in terms of the perturbation of the zeros (rather than coefficients) of the input polynomials, which leads us to the solution via the computation of maximum matchings or connected components in the associated bipartite graphs. Key words: polynomial gcds, approximate gcds, Padé approximation, Hankel matrices, polynomial zeros, root neighborhoods, bipartite graphs. 1991 Mathematics Subject Classification: 68Q40, 65D99, 65Y20.
1 Introduction.

Computation of polynomial greatest common divisors (gcds) is a fundamental problem of algebraic computing and has important widespread applications in network theory and control linear systems (see [?], [?], [?], and [??]), which require numerical solution of the problem, where the input is given approximately, within some fixed error bounds. On the other hand, computation of polynomial gcds is an excellent example of numerically ill-posed problems. For instance, let \( v(x) \) be a non-constant divisor of a polynomial \( u(x) \). Then \( \gcd(u(x), v(x)) = v(x) \), but \( \gcd(u(x) + \delta v(x)) = 1 \) for any constant \( \delta \neq 0 \). Thus, a small perturbation of \( u(x) \) may cause a dramatic decrease of the degree of the gcds. [?], [?], [?], [?], and [?] define approximate gcds so as to avoid the latter deficiency. Namely, for two polynomials \( u(x) \) and \( v(x) \), a fixed polynomial norm, and a positive \( \epsilon \), one may non-uniquely define an approximate gcd or, we say \( \epsilon \)-gcd \( d^*(x) \), as \( \gcd(u^*(x), v^*(x)) \) whose degree \( d^* \) is maximized over all polynomials \( u^*(x) \) and \( v^*(x) \) in an \( \epsilon \)-neighborhood of \( u(x) \) and \( v(x) \), satisfying

\[
\deg u^*(x) \leq m = \deg u(x), \quad \deg v^*(x) \leq n = \deg v(x),
\]

\[
\| u^*(x) - u(x) \| \leq \epsilon \| u(x) \|, \quad \parallel v^*(x) - v(x) \parallel \leq \epsilon(x) \parallel .
\]

In spite of extensive work and substantial progress, there remain several open problems with the computation of \( \epsilon \)-gcds. The pioneering paper [?] studies only the asymptotic complexity of computing the \( \epsilon \)-gcds where \( \epsilon \to 0 \). The results of [?] rely on extremely tedious analysis but do not apply to the most realistic and practically important case, where the input errors can be relatively large, and the precision of computing is insufficiently high to ignore rounding errors of the computation. The latter assumptions imply the model of study with inexact input and infinite precision computations. In our paper, we will assume this model, following all papers cited above except for [?]. On the other hand, all these papers, except for [?], present heuristic solution algorithms, whereas the arithmetic complexity estimates for the solution presented in [?] are quite large (that is, unspecified polynomial in \( m + n \) and exponential in \( d^* \)), thus implying the need for further study. Furthermore, neither of these far presented algorithms allows its effective parallelization. Technically, there is another major omission. Several approaches have been studied or at least listed so far. They rely
on Euclidean algorithm [?], [?], [?], computations with subresultant matrices [?], [?], various techniques of least-squares computations, optimization, and quadratic programming [?], [?], and a version of Lazard's algorithm [L81], which is equivalent to the matrix pencil algorithm of [KM94] (described in [KM94] by using the terminology of automatic control). At least two important approaches are missing, however, from all these papers, that is, ones based on Padé approximation and approximating polynomial zeros. In our present study, we demonstrate several methodological and computational advantages of these two approaches over the cited ones. Padé approximation is an important and well developed subject with applications to algebraic computing, signal processing, and the study of analytic functions [?], [?], [?]. We show some computational benefits of reducing the approximate gcd problem to the computation of Padé approximations for an inexact input, versus the subresultant approach, intensively studied in [?], [?]. (In particular, the former approach involves better structured matrices of smaller size.) On the other hand, the reduction into the opposite direction, from Padé computations to the gcd computations, enables us to solve both problems by our second approach, which, as we show, is quite effective. Namely, we reduce the gcd problem to approximating the zeros of the input polynomials, where highly effective algorithms are available [?], [?], [?]. With such a reduction, we explain the numerically unstable behavior of approximate gcds and restate the problem in terms of the concept of the polynomial root neighborhoods of [?]. This enables us to choose a proper class of the input perturbations for the gcds and also to associate bipartite graphs with the gcds. Then the solution of the approximate gcd problem is reduced to computing maximum matchings in such graphs. In particular this enables us to compute easily a non-trivial upper bound on the degree of the approximate gcds and to certify correctness of the solution, which was the bottleneck of the known approaches. We made this progress assuming the customary formulation of the gcd problem, based on the perturbation of the input coefficients. In sections ?? and ?? we study the problem based on the perturbation of the zeros of the input polynomials and argue that this is a more appropriate basic assumption. We also show its computational advantages: it always enables us to compute an approximate gcd itself (rather than a candidate polynomial, which may have a lower or higher degree) and to simplify the stage of the computations in bipartite graphs, by replacing the matching stage by the computation
of the components. The proposed algorithms have lower computational complexity (which ranges between linear and quadratic in terms of arithmetic operations and comparisons involved, except for $O(n^{2.5})$ comparisons used for matching), and their parallelization enables NC (or RNC) and work efficient solution of the approximate gcd problem (cf. e.g. [?] or [?], ch. 4, on the definitions of NC, RNC and work efficiency). We organize our presentation as follows. After some preliminaries in section ??, we study the Padé approximation approach in sections ?? and ?? In sections ??--?? we study the $\epsilon$-gcds problem based on the concept of a polynomial root neighborhood. In sections ?? and ??, we study the $\delta$-gcd problem, where the perturbation is applied to the zeros of the input polynomials. In section ?? we comment on the extension of our approach to Padé approximation for an inexact input. Section ?? is left for a short discussion. In the appendix we briefly recall some major known methods for $\epsilon$-gcds and prove an auxiliary result for testing $\epsilon$-divisibility. Acknowledgements. I gratefully acknowledged receiving reprints of [CGTW95] from André Galligo and Erich Kaltofen, a preprint of [HS95] from the former, and a reprint of [HZ96] from Uri Zwick. My present work was substantially motivated by André Galligo’s comments on [CGTW95], delivered in [G95], and partly by Erich Kaltofen’s interest to the computation of numerical rank of a Toeplitz matrix. This paper was written and submitted for publication in 1995, turned into a research report[?] in 1996, and substantially revised in 1997, based on the valuable suggestions by the referees.

2 Polynomial and vector norms, $\epsilon$-divisibility and $\epsilon$-gcds (some definitions and preliminaries).

Hereafter, we will refer to arithmetic operations as to ops. A polynomial $p(x) = \sum_{i=0}^{k} p_i x^i$ can be identified with its coefficient vector $\vec{p} = [p_0, \ldots, p_k]^T$. (We write $w^T$ for the transpose of a vector or a matrix $w$.) The same norms will be used for both vectors and polynomials, in particular to measure the distances, $\text{dist}(s(x), t(x)) = \|s(x) - t(x)\| = \|\vec{s} - \vec{t}\|$. We recall
the customary norms
\[ \| \vec{p} \|_h = \left\| \sum_i p_i x^i \right\|_h = \left( \sum_i |p_i|^h \right)^{1/h}, \]
which for \( h = \infty \) turn into the maximum norm,
\[ \| \vec{p} \|_\infty = \| \sum_i p_i x^i \|_\infty = \max_i |p_i|. \]
In some cases, the weighted \( h \)-norms, \( \| \vec{p} \|_{h; \vec{w}} = \| \sum_i p_i x^i \|_{h; \vec{w}} = \left( \sum_i |p_i|^h w_i \right)^{1/h} \) for a fixed vector \( \vec{w} = (w_i) \), are also useful in the study of \( \epsilon \)-gcds \([7]\). When our study applies to any fixed norm, we will write \( \| \cdot \| \). The next two definitions re-introduce the \( \epsilon \)-gcds of \((??),(??)\), based on the concept of an \( \epsilon \)-divisor.

**Definition 2.1** A polynomial \( d(x) \) is an \( \epsilon \)-divisor of a polynomial \( p(x) \) (under a fixed norm \( \| \cdot \| \) ) if there exists a perturbation of \( p(x) \) by a polynomial \( \Delta(x) \) such that \( d(x) \) divides \( p(x) + \Delta(x) \), \( \deg \Delta(x) \leq N \), and \( \| \Delta(x) \| \leq \epsilon \| p(x) \| \).

**Definition 2.2** For two polynomials,
\[ u(x) = \sum_{i=0}^m u_i x^i, \quad v(x) = \sum_{i=0}^n v_i x^i, \quad u_m v_n \neq 0, \quad m \leq n, \]
every their monic \( \epsilon \)-divisor \( g(x) \) that has the maximum degree, \( d_\epsilon = d(u, v, \epsilon) \), is called their \( \epsilon \)-gcd.

Using the 2-norm, \( \| \cdot \|_2 \), often leads to some computational advantages.

**Proposition 2.1** \([7]\). Given two polynomials \( p(x) \) and \( d(x) \) of degrees \( k \) and \( l \), respectively, \( k > l \), and a positive \( \epsilon \), it suffices to use \( O(k \log(k - l) + \min\{(k - l) \log^2(k - l), kl\}) \) ops to decide if \( d(x) \) is an \( \epsilon \)-divisor of \( p(x) \) (under the 2-norm).

The algorithm of \([7]\) supporting this proposition actually computes two polynomials \( q(x) \) and \( \Delta(x) = p(x) - d(x)q(x) \) such that \( \| \Delta(x) \|_2 \) is minimum, and \( \deg \Delta(x) \leq \deg p(x) \). Having \( \Delta(x) \) available, we may check immediately if \( \| \Delta(x) \|_h \leq \epsilon \| p(x) \|_h \) for \( h = 1 \) or \( h = \infty \); the values \( \| \Delta(x) \|_h \) for \( h = 1 \) are within factors \( n \) or \( \sqrt{n} \) from the minimum \( \| p(x) - d(x)q(x) \|_h \), due to the following useful relations:
\[ \| \vec{v} \|_\infty \leq \| \vec{v} \|_2 \leq \| \vec{v} \|_1 \leq \sqrt{n} \| \vec{v} \|_2 \leq n \| \vec{v} \|_1, \]
which hold for any vector \( \vec{v} \) of dimension \( n \). Furthermore, we can prove the following extension of proposition ?? (see appendix ??).

**Proposition 2.2** The \( \epsilon \)-divisibility under an weighted 2-norm can be decided at the cost of performing \( O(N^2 \log N) \) ops.

**Remark 2.1** Our study can be easily extended to \((\epsilon_u, \epsilon_v)\)-gcds where \( g(x) \) is an \( \epsilon_u \)-divisor for \( u(x) \) and an \( \epsilon_v \)-divisor for \( v(x) \), for fixed \( \epsilon_u \) and \( \epsilon_v \), or more generally, where a distinct upper bound may be imposed on the allowed perturbation of each coefficient of \( u(x) \) and \( v(x) \).

3 Padé approximation approach to computing a polynomial gcd or \( \epsilon \)-gcd.

A pair of polynomials \( w(x) \) and \( z(x) \) of the smallest degree that satisfy the equation \( w(x)v(x) = z(x)u(x) \) immediately defines the gcd, \( g(x) = \gcd(v(x), u(x)) \), as the polynomial

\[
g(x) = \frac{u(x)}{w(x)} = \frac{v(x)}{z(x)}.
\]

The problem of the computation of \( w(x) \) and \( z(x) \) can be appropriately re-phrased by using the important classical concept of Padé approximation (cf. [?], [?], [?]).

**Definition 3.1** For any formal power series \( a(x) = \sum_{i=0}^{\infty} a_i x^i \) and for two non-negative integers \( k \) and \( \ell \), a pair of polynomials \( s(x) \) and \( t(x) \) is a \((k, \ell)\)-th Padé approximation of \( a(x) \) if \( \deg s(x) \leq k \), \( \deg t(x) \leq \ell \), and \( s(x) - a(x)t(x) = 0 \mod x^{N+1}, N = k + \ell \).

**Proposition 3.1** [?]. The pair of polynomials \( q(x) \) and \( t(x) \) of definition ?? is defined uniquely, up to its scaling by common factors or common divisors.

Given the polynomials \( u(x) \) and \( v(x) \) of (??), we may assume that \( v(0) \neq 0 \) and define the formal power series

\[
h(x) = \sum_{i=0}^{\infty} \frac{h_i x^i}{v(x)}.
\]

(3.1)

(The restriction \( v(0) \neq 0 \) can be removed by removing the maximum degree factors \( x^i \) and \( x^j \) from \( u(x) \) and \( v(x) \), respectively, and adding the factor \( x^{\min\{i,j\}} \) to their gcd. Alternatively, one may work with the reverse polynomials \( U(x) = x^m u(1/x), V(x) = x^n v(1/x) \),
\[ W(x) = x^{m-d} w(1/x), \quad Z(x) = x^{n-d} z(1/x), \]

The computation of \( h(x) \mod x^{N+1} \) amounts to the computation modulo \( x^N \) of a polynomial reciprocal and a polynomial product and costs \( O(n \log n) \) ops, \( [?] \), p. 22. The \((k,l)\)-th Padéapproximation can be computed by means of the extended Euclidean algorithm at the cost \( O(N \log^2 N) \) ops for \( N = k + l \), \( [?] \), \( [?] \), pp. 38–39. These considerations give us a \text{gcd} algorithm, and we may extend it to computing an \( \epsilon \)-\text{gcd} as follows.

\begin{enumerate}
  \item For each \( d, d = 0, 1, \ldots, m \), successively or concurrently compute \( w_d(x), z_d(x) \), the \((m-d, n-d)\)-th Padéapproximation to \( h(x) \); then apply proposition \( [?] \) to test if \( w_d(x) \) is an \( \epsilon \)-divisor of \( u(x) \). If ”not”, discard such a value \( d \). Otherwise compute a polynomial \( g_{d,e}(x) \) (an \( \epsilon \)-quotient) such that \( \|w_d(x)g_{d,e}(x) - u(x)\| \leq \epsilon \). Then apply proposition \( [?] \) again, to test if \( g_{d,e}(x) \) is an \( \epsilon \)-divisor of \( v(x) \) too. If ”not”, discard this \( d \). Otherwise, store such a \( d \) and \( g_{d,e}(x) \).

  \item If the set of \( d \)'s is empty, output \text{FAILURE}; otherwise select the largest remaining \( d = d^* \) and output \( d^* \) and \( g^*(x) = g_{d^*,e}(x) \).
\end{enumerate}

By construction, \( g_{d^*,e}(x) \) is a common \( \epsilon \)-divisor of \( u(x) \) and \( v(x) \), though some \( \epsilon \)-perturbations of \( u(x) \) and \( v(x) \) may have a common \( \epsilon \)-divisor of a larger degree. The algorithm is performed at quite a low computational cost of \( O(mn \log^2 n) \) ops. Furthermore, the stage of computing the \((m-d, n-d)\) Padé approximations of \( h(x) \) for all \( d \) requires only \( O(n \log^2 n) \) ops, except for some cases of degeneration\( [?] \), sect. 5. On the other hand, this approach, like the Euclidean and subresultant approaches, gives us no effective general means for detecting degeneration or verifying that the output polynomial is an \( \epsilon \)-\text{gcd}.

4 The Hankel/Bezout techniques for Padéapproximation.

In the algorithms of the previous sections, one may compute Padéapproximations by relying on computations with Hankel matrices, rather than on the extended Euclidean algorithm. This approach may require a little more ops but enables better numerical control of the computations and provides an additional insight into the behavior of \( \epsilon \)-\text{gcds}. The basic idea is to associate the formal power series \( h(x) \) of \( ([?] \) with the infinite Hankel matrix

\[ H = H(u, v) = (h_{i,j}), \quad h_{i,j} = h_{i+j+m-n+1}, \quad i, j = 0, 1, \ldots \]

For readers' convenience, we recall
that a matrix \((h_{i,j})\) is called a \textit{Hankel matrix} if all its entries are invariant in their shift into antidiagonal direction, that is, if \(h_{i,j} = h_{i+1,j-1}\) for all pairs \(i, j\) for which \(h_{i,j}\) and \(h_{i+1,j-1}\) are defined. A general \(n \times n\) matrix may have \(n^2\) distinct entries, but an \(n \times n\) Hankel matrix is symmetric \((h_{i,j} = h_{j,i})\), has at most \(2n - 1\) distinct entries, and is completely defined by the pair of its first and last rows or columns. Computations with Hankel matrices are also dramatically simplified versus the case of general matrices. In particular we have the following results [?].

\textbf{Proposition 4.1} The multiplication of an \(N \times N\) Hankel matrix by a vector can be reduced to multiplication of two polynomials of degree \(O(N)\) and performed in \(O(N \log N)\) ops.

\textbf{Proposition 4.2} \(O(N \log^2 N)\) ops suffice to solve a non-singular linear system of \(N\) equations with a Hankel coefficient matrix.

Now let \(H_k\) denote the \(k \times k\) leading principal submatrix of \(H\), that is, one where \(i\) and \(j\) range from 0 to \(k - 1\). (Clearly, \(H_k\) is a Hankel matrix.) We have the following results ([?], [?], [?], pp. 140).

\textbf{Proposition 4.3} Let \(r = \text{rank } H_n\), \(d = \gcd g(x)\). Then the matrix \(H_r\) is non-singular, and \(d + r = n\).

\textbf{Proposition 4.4} The computation of the \((m,n)\) Padé approximation of the formal power series \(h(x)\) of (??) can be reduced to the solution of a consistent linear system of equations with the coefficient matrix \(H_n\), multiplication of an \(m \times n\) triangular Hankel matrix by a vector, and asubtraction of a pair of \(m\)-dimensional vectors from each other.

The harder part of the latter computation is the solution of the linear system. It can be performed in three steps:

a) compute the rank \(r\) of the matrix \(H_n\),

b) solve a non-singular linear system of equations with the coefficient matrix \(H_r\) (non-singularity is by proposition ??, the solution cost is \(O(r \log^2 r)\) by proposition ??).

c) recover the solution of the original consistent linear system with the matrix \(H_n\) (in \(O(n \log n)\) ops by proposition ??).
Remark 4.1 At stage b), one may substantially improve numerical stability of the computations by means of a simple transition from the Hankel to Bezout linear systems (?, p.162).

In the context of the computation of $\varepsilon$-gcds, the matrix $H_n$ should be allowed to vary with the input polynomials $u(x)$ and $v(x)$. The $\varepsilon$-perturbations of $u(x)$ by $\delta_u(x)$ and $v(x)$ by $\delta_v(x)$, $\|\delta_u(x)\| \leq \varepsilon, \|\delta_v(x)\| \leq \varepsilon$, cause the perturbations of the polynomial $(u(x)/v(x)) \mod x^{m+n+1}$ within $\delta_{u,\varepsilon} + \delta_{v,\varepsilon}$ where

$$\delta_{u,\varepsilon} = \varepsilon \|(1/v(x)) \mod x^{m+n+1}\|,$$  \hspace{1cm} (4.1)

$$\delta_{v,\varepsilon} = \varepsilon \|(u(x) + \delta_u(x))/(v(x)(v(x) + \delta_v(x))) \mod x^{m+n+1}\|$$ \hspace{1cm} (4.2)

(for any fixed norm). To estimate the resulting perturbation of the matrix $H_n$, we recall the definition and some properties of the operator matrix norms $\|\cdot\|_h$, which are also called subordinate to and consistent with the vector norms $\|\cdot\|_h$ of (??) (cf. [?], pp.55–57). For a matrix $A = (a_{i,j})$, we have

$$\|A\|_h = \sup_{\vec{v} \neq \vec{0}} \frac{\|A\vec{v}\|_h}{\|\vec{v}\|_h}, \; h = 1, 2, \infty,$$

$$\|A\|_1 = \|A^T\|_\infty = \max_j \sum_i |a_{i,j}|,$$ \hspace{1cm} (4.3)

$$\|A\|_2^2 \leq \|A\|_1 \|A\|_\infty,$$

so that

$$\|A\|_2 \leq \|A\|_1 = \|A\|_\infty$$ \hspace{1cm} (4.4)

for a symmetric matrix $A$, in particular for $A = H_n$. (??) implies that the perturbation $\Delta_\varepsilon$ of $A$ caused by the $\varepsilon$-perturbations of $u(x)$ and $v(x)$ satisfies

$$\|\Delta_\varepsilon\|_1 = \|\Delta_\varepsilon\|_\infty \leq \delta_{u,\varepsilon} + \delta_{v,\varepsilon}$$

for $\delta_{u,\varepsilon}$ and $\delta_{v,\varepsilon}$ of (??), (??). (The equation in the above follows from (??) since $\Delta_\varepsilon$ is a Hankel and therefore symmetric matrix.) Due to (??), we also obtain that

$$\|\Delta_\varepsilon\|_2 \leq \delta_{u,\varepsilon} + \delta_{v,\varepsilon}.$$ \hspace{1cm} (4.5)

To control the resulting impact of the $\varepsilon$-perturbations of $u(x)$ and $v(x)$ on the $\varepsilon$-gcds and in particular on their degree $d_\varepsilon$, one may apply some known results on the eigendecomposition of a real symmetric matrix (see [?], Theorems 8.1.1, 8.1.13).