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, inparticular to linear control systems. We study two approaches to the solution so far omitted by theresearchers, in spite of intensive recent work in this area. Correlation to numerical Padé approximation enabled us toimprove computations for both problems (gcds and Padé). Reduction to approximating polynomial zeros enabled us to obtaina new insight into the gcd problem and to devise effectivesolution algorithms. In particular, unlike the known algorithms,we compute an upper bound on the degree of approximate gcds at a lowcomputational cost, and this enables us to certify the correctness ofthe solution. We also argue in favor of restating the problem interms of the perturbation of the zeros (rather than coefficients) of theinput polynomials, which leads us to the solution via the computation ofmaximum matchings or connected components in the associated bipartitegraphs. **Key words:** polynomial gcds,approximate

gcds, Padé approximation, Hankelmatrices, polynomial zeros, root neighborhoods, bipartite graphs. **1991 Mathematics Subject Classification:** 68Q40, 65D99, 65Y20.

1 Introduction.

Computation of polynomial *greatest common divisors (gcds)* is afundamental 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 givenapproximately, within some fixed error bounds. On the other hand,computation of polynomial gcds is an excellent example of numericallyill-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 gcd.[?], [?], [?],[?], [?], [?], and [?] define approximate gcds soas to avoid the latter deficiency. Namely, for two polynomials$u(x)$ and $v(x)$, a fixed polynomial norm, and a positive ϵ, one may non-uniquely define an approximate gcd or, we say *ϵ-gcd* $d^*(x)$, as gcd $(u^*(x), v^*(x))$ whose degree d^* is maximized over allpolynomials$u^*(x)$ and $v^*(x)$ in an ϵ-neighborhood of $u(x)$ and $v(x)$, satisfying

$$\deg u^*(x) \leq \ m = \ \deg u(x), \quad \deg v^*(x) \leq \ n = \ \deg v(x), \tag{1.1}$$

$$\| \ u^*(x) - u(x) \ \| \leq \epsilon \ \| \ u(x) \ \|, \ parallel v^*(x) - v(x) \ \| \leq \epsilon(x) \ \| \ . \tag{1.2}$$

In spite of extensive work and substantial progress, there remain several open problems with the computation of ϵ-gcds. The pioneering paper [?] studies only the asymptoticcomplexity of computing the ϵ-gcds where $\epsilon \to 0$. The results of [?] rely on extremely tedious analysis but donot apply to the most realistic and practically important case, where theinput errors can be relatively large, and the precision of computing issufficiently high to ignore rounding errors of the computation. The latterassumptions imply the model of study with inexact input and infinite precision computations. In our paper, we will assume thismodel, following all papers cited above except for [?]. On the other hand, all these papers, except for [?], presentheuristic solution algorithms, whereas the arithmetic complexityestimates 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 theso far presented algorithms allows its effective parallelization. Technically, there is another major omission. Several approaches havebeen studied or at least listed so far. They rely on Euclideanalgorithm [?], [?], [?], 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, wedemonstrate several methodological and computational advantages of thesetwo approaches over the cited ones. Padé approximation is an important and well developed subjectwith applications to algebraic computing, signal processing, and thestudy of analytic functions [?], [?], [?]. We show some computational benefits of reducing theapproximate gcd problem to the computation of Padé approximations for an inexact input, versus the subresultant approach, intensivelystudied in [?], [?]. (In particular, the formerapproach involves better structured matrices of smaller size.) On the other hand, the reduction into the opposit direction, from Padé computations to the gcd computations, enables us tosolve both problems by our second approach, which, as weshow, is quite effective. Namely, we reduce the gcd problem to approximating the zerosof the input polynomials, where highly effective algorithmsare available [?], [?], [?]. With such a reduction, we explain the numerically unstablebehavior of approximate gcds and restate the problem in terms of theconcept of the *polynomial root neighborhoods* of [?]. This enables us to choose a proper class of theinput perturbations for the gcds and also to associate bipartite graphs with the gcds. Then the solution of theapproximate gcd problem is reduced to computing *maximummatchings* in such graphs. In particular this enables us to computeeasily a non-trivial upper bound on the degree of the approximate gcdsand to certify correctness of the solution, which was the bottleneck ofthe known approaches. *We made this progress assuming the customary formulation of thegcd problem*, based on the *perturbation of the inputcoefficients.* In sections **??** and **??** we study theproblem based on the *perturbation of the zeros* of the inputpolynomials and argue that this is a more appropriate basicassumption. We also show its computational advantages: it alwaysenables us to compute an approximate gcd itself (rather than acandidate polynomial, which may have a lower or higher degree) andto simplify the stage of the computations in bipartite graphs, byreplacing the matching stage by the computation of the components. The proposed algorithms have lower computational complexity (which rangesbetween linear and quadratic in terms of arithmetic operations andcomparisons involved, except for$O(n^{2.5})$ comparisons used for matching), and their parallelization enables NC (or RNC) and work efficient solution of theapproximate gcd problem (cf. e.g. [**?**] or [**?**], ch. 4, onthe definitions of NC, RNC and work efficiency). We organize our presentation as follows. After some preliminatires insection **??**, we study the Padé approximation approach insections**??** and **??**. In sections **??**–**??** westudy the ϵ-gcds problem based on the concept ofa polynomial root neighborhood. In sections **??** and**??**, we study theδ-gcd problem, where the perturbation is applied to the zerosof the input polynomials. In section **??** we comment on theextension of our approach to Padé approximation for an inexactinput. Section**??** is left for a short discussion. In the appendix we brieflyrecall some major known methods for ϵ-gcds and prove anauxiliary result for testing ϵ-divisibility. **Acknowledgements.** I grate-

fully acknowledgereceiving reprints of [CGTW95]from André Galligo and Erich Kaltofen, a preprint of [HS95] fromthe 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 thecomputaton of numerical rank of a Toeplitz matrix. This paper was written and submitted for publication in 1995, turned intoa research report[**?**] in 1996, and substantially revised in 1997, based on thevaluable suggestions by the referees.

2 Polynomial and vector norms,ϵ-divisibility and ϵ-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

$$\|\vec{p}\|_h = \|\sum_i p_i x^i\|_h = (\sum_i |p_i|^h)^{1/h}, \tag{2.1}$$

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, $\|\vec{p}\|_{h,\vec{w}} = \|\sum_i p_i x^i\|_{h,\vec{w}} = (\sum_i |p_i|^h w_i)^{1/h}$ for a fixed vector $\vec{w} = (w_i)$, are also useful in the study of ϵ-gcds [?]. When our study applies to any fixed norm, we will write$\|.\|$. The next two defintions re-introducethe ϵ-gcds of (**??**),(**??**), based on the concept of an ϵ-divisor.

Definition 2.1 *A polynomiald(x) is an ϵ-divisor of a polynomial $p(x)$ (under afixed norm $\|.\|$) 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,*

$$u(x) = \sum_{i=0}^{m} u_i x^i, \quad v(x) = \sum_{i=0}^{n} v_i x^i, \quad u_m v_n \neq 0, \quad m \leq n, \tag{2.2}$$

every their monic ϵ-divisor $g(x)$ that has the maximum degree, $d_\epsilon = d(u, v, \epsilon)$, is called their ϵ-gcd.

Using the *2-norm*, $\|.\|_2$, often leads to some computational advantages.

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

The algorithm of [?] supporting this proposition actuallycomputes two polynomials $q(x)$ and $\Delta(x) = p(x) - d(x)q(x)$ suchthat $\|\Delta(x)\|_2$ 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 $h = 1$ arewithin 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 ofproposition **??** (see appendix **??**).

Proposition 2.2 *The ϵ-divisibility under aweighted 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 $(\epsilon_{\vec{u}}, \epsilon_{\vec{v}})$-gcds where $g(x)$ is an $\epsilon_{\vec{u}}$-divisor for $u(x)$ and an$\epsilon_{\vec{v}}$-divisor for $v(x)$, for fixed $\epsilon_{\vec{u}}$and $\epsilon_{\vec{v}}$, or more generally, where a distinctupper bound may be imposed on the allowed perturbation of eachcoefficient of $u(x)$ and $v(x)$.*

3 Padé approximation approach to computing a polynomial gcd or ϵ-gcd.

A pair of polynomials $w(x)$ and $z(x)$ of the smallestdegree that satisfy the equation $w(x)v(x) = z(x)u(x)$ immediately defines the gcd, $g(x) = \text{gcd }(v(x), u(x))$, 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-phrazed 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 ℓ, a pair of polynomials $s(x)$ and $t(x)$ is a (k, l)-th Padé approximation of $a(x)$ if $\deg s(x) \leq k$, $\deg t(x) \leq l$, and $s(x) - a(x)t(x) = 0 \bmod x^{N+1}$, $N = k + l$.*

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 mayassume that $v(0) \neq 0$ and define the formal power series

$$h(x) = \sum_{i=0}^{\infty} h_i x^i = \frac{u(x)}{v(x)}. \tag{3.1}$$

(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, andadding the factor $x^{\min\{i,j\}}$ to their gcd. Alternatively, one maywork with the reverse polynomials $U(x) = x^m u(1/x)$, $V(x) = x^n v(1/x)$,

$W(x) = x^{m-d}w(1/x)$, $Z(x) = x^{n-d}z(1/x)$.) The computation of $h(x) \bmod x^{N+1}$ amounts to the computationmodulo 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 Euclideanalgorithm at the cost $O(N \log^2 N)$ ops for $N = k + l$, [?], [?], pp. 38–39. These considerations give us a gcd algorithm, and we may extend it to computing an ϵ-gcd as follows.

a) For each d, $d = 0, 1, \ldots, m$, successively orconcurrently 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 ϵ-divisor of $u(x)$. If "not", discard sucha value d. Otherwise compute a polynomial $g_{d,\epsilon}(x)$ (an ϵ-quotient) such that $\|w_d(x)g_{d,\epsilon}(x) - u(x)\| \leq \epsilon$. Then applyproposition **??** again, to test if $g_{d,\epsilon}(x)$ is anϵ-divisor of$v(x)$ too. If "not", discard this d. Otherwise, store such a d and $g_{d,\epsilon}(x)$.

b) If the set of d's is empty, output FAILURE; otherwiseselect the largest remaining $d = d^*$ and output d^* and $g^*(x) = g_{d^*,\epsilon}(x)$.

By construction, $g_{d^*,\epsilon}(x)$ is a common ϵ-divisor of$u(x)$ and $v(x)$, though some ϵ-perturbations of $u(x)$ and $v(x)$ may have a common ϵ-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 verifyingthat the output polynomial is an ϵ-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 ϵ-gcds. The basic idea is to associate the formal power series $h(x)$ of (**??**) with the infinite Hankel matrix $H = H(u,v) = (h_{i,j})$, $h_{i,j} = h_{i+j+m-n+1}$, $i, j = 0, 1, \ldots$. For readers' convenience, we recall that a matrix $(h_{i,j})$ is called a *Hankel matrix* if all its entries are invariant intheir 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 orcolumns. Computations with Hankel matrices are also dramatically simplifiedversus the case of general matrices. In particular we have the following results [**?**].

Proposition 4.1 *The multiplication of an $N \times N$ Hankel matrix by a vectorcan be reduced to multiplication of two polynomials of degree $O(N)$and performed in $O(N \log N)$ ops.*

Proposition 4.2 *$O(N \log^2 N)$ ops suffice to solve a non-singular linearsystem 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).

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

Proposition 4.4 *The computation of the (m, n) Padé approximationof the formal power series $h(x)$ of (**??**)can be reduced to the solution of a consistent linearsystem of equations with the coefficient matrixH_n, multiplication of an $m \times m$triangular Hankel matrix by a vector, and asubstraction 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 thecoefficient matrix H_r (non-singularity is byproposition **??**, the solution cost is $O(r \log^2 r)$ byproposition **??**).

c) recover the solution of the original consistent linear systemwith 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 transitionfrom the Hankel to Bezout linear systems ([?], p.162).*

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

$$\delta_{u,\epsilon} = \epsilon\|(1/v(x)) \bmod x^{m+n+1}\|, \tag{4.1}$$

$$\delta_{v,\epsilon} = \epsilon\|(u(x) + \delta_u(x))/(v(x)(v(x) + \delta_v(x))) \bmod x^{m+n+1}\| \tag{4.2}$$

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

$$\begin{gathered} \|A\|_h = \sup_{\vec{v}\neq\vec{0}} \|A\vec{v}\|_h/\|\vec{v}\|, \ \ h = 1, 2, \infty, \\ \|A\|_1 = \|A^T\|_\infty = \max_j \sum_i |a_{i,j}|, \\ \|A\|_2^2 \leq \|A\|_1\|A\|_\infty, \end{gathered} \tag{4.3}$$

so that

$$\|A\|_2 \leq \|A\|_1 = \|A\|_\infty \tag{4.4}$$

for a symmetric matrix A, in particular for $A = H_n$. (**??**) implies that the perturbation Δ_ϵ of Acaused by the ϵ-perturbations of $u(x)$ and $v(x)$ satisfies

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

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

$$\|\Delta_\epsilon\|_2 \leq \delta_{u,\epsilon} + \delta_{v,\epsilon}. \tag{4.5}$$

To control the resulting impact of theϵ-perturbations of$u(x)$ and$v(x)$ on the ϵ-gcds and in particular on their degree d_ϵ, onemay apply some known results on the eigendecomposition of a real symmetric matrix (see [**?**], Theorems 8.1.1, 8.1.13).