A fast and numerically stable Euclidean–like algorithm for detecting relatively prime numerical polynomials

Bernhard Beckermann
Laboratoire d'Analyse Numérique et d'Optimisation,
Université des Sciences et Technologies de Lille,
59655 Villeneuve d'Ascq Cedex, France
e–mail: bbecker@ano.univ-lille1.fr
and
George Labahn
Department of Computing Science
University of Waterloo, Waterloo, Ontario, Canada
e–mail: glabahn@daisy.uwaterloo.ca

Jan 8, 1998

Abstract

In this paper we provide a fast, numerically stable algorithm to determine when two given polynomials a and b are relatively prime and remain relatively prime even after small perturbations of their coefficients. Such a problem is important in many applications where input data is only available up to a certain precision.

Our method – an extension of the Cabay–Meleshko algorithm for Padé approximation – is typically an order of magnitude faster than previously known stable methods. As such it may be used as an inexpensive test which may be applied before attempting to compute a "numerical GCD", in general a much more difficult task. We prove that the algorithm is numerically stable and give experiments verifying the numerical behaviour. Finally, we discuss possible extensions of our approach that can be applied to the problem of actually computing a numerical GCD.

1 Introduction

Let $a, b \in \mathbb{C}[z]$ be (univariate) polynomials with real or complex coefficients

$$a(z) = a_0 + a_1 z + \ldots + a_m z^m, \quad b(z) = b_0 + b_1 z + \ldots + b_n z^n, \quad a_m, b_n \neq 0$$

of degree m and n, respectively. In applications, the coefficients of a and b will only be known up to a certain precision. Thus in order to decide whether a and b are "numerically coprime" we have to determine (lower bounds of) the quantity

$$\epsilon(a,b) := \inf\{||(a-a^*, b-b^*)|| : \ (a^*, b^*) \text{ have a common root, } \deg a^* \leq m, \deg b^* \leq n\} \qquad (1)$$

for some norm $||\cdot||$ acting on the space of (matrix) polynomials (our choice of norms will be quantified below). In other words, we are sure that any pair of polynomials resulting from (a, b) by perturbations of order less than $\epsilon(a, b)$ will be relatively prime.[1]

[1] Of course, $\epsilon(a,b) = 0$ is equivalent to saying that a, b have a common root. However, a small $\epsilon(a,b)$ does not necessarily imply that one of the roots of a is close to the set of roots of b (see, e.g., [3, Example 5.3]) — and thus a numerical GCD cannot be obtained by comparing the sets of (numerical) roots of a and b. In fact, from the famous example of Wilkinson (the polynomial $p(z) = (z-1)\cdot(z-2)\cdot\ldots\cdot(z-20)$) we know that small perturbations of the coefficients of a polynomial (in the monomial basis) do not necessarily lead to small perturbations of its roots (for a more detailed study of the condition number of the underlying non–linear map in case of real zeros see [2]).

In [3, Corollary 4.4], a lower bound for $\epsilon(a,b)$ in terms of solutions of two diophantine equations has been given, namely

$$\epsilon(a,b) \geq \frac{1}{\kappa}, \quad \kappa := \left\| \begin{bmatrix} v & \underline{v} \\ u & \underline{u} \end{bmatrix} \right\|, \tag{2}$$

where $u, v, \underline{u}, \underline{v}$ are polynomials solving the diophantine equations

$$a \cdot v + b \cdot u = 1, \quad \deg u < m, \quad \deg v < n, \tag{3}$$

$$a \cdot \underline{v} + b \cdot \underline{u} = z^{m+n-1}, \quad \deg \underline{u} < m, \quad \deg \underline{v} < n. \tag{4}$$

Equations (3) and (4) are equivalent to determining the first and last columns of the inverse of $S(a,b)$, the Sylvester matrix of a and b. It is shown in [3, Corollary 4.4] that this is a sharper bound for coprimeness than those obtained from the smallest singular value of $S(a,b)$, the current measure for coprimeness used in [11, 12]. However, it still remains to actually compute the polynomials $u, v, \underline{u}, \underline{v}$.

In the case of exact arithmetic, equations (3) and (4) are typically solved using a Euclidean-like PRS algorithm [13]. However, it is known that Euclid's algorithm cannot be applied directly in the case of numeric polynomials without encountering numerical stability (see Example 2.1 in Section 2). In order to ensure numerical stability one can set up a linear system and use Gaussian elimination or QR factorization to obtain a solution to our diophantine equations. However such techniques do not take advantage of the special structure of the Sylvester coefficient matrix for such linear systems.

In this paper we present an algorithm for computing the coprime parameter κ. The method, an extension of the Cabay–Meleshko algorithm for Padé approximation [7], can be viewed as a look-ahead Euclidean algorithm which "jumps" over remainders that are in some sense ill-conditioned. The algorithm determines when a remainder is ill-conditioned by estimating (in terms of easily produced quantities) the condition number of the corresponding linear problem.

Our algorithm typically has complexity $\mathcal{O}((m+n)^2)$, in comparison with the $\mathcal{O}((m+n)^3)$ complexity of Gaussian elimination or QR factorization. We prove that our algorithm is weakly stable and so produces correct answers for our numerical problem. The low complexity and numerical correctness of our algorithm means that our coprimeness test may always be applied before starting the (sometimes quite expensive) computation of a numerical GCD, providing a reliable lower bound for $\epsilon(a,b)$ even in finite precision arithmetic.

The problem of computing a numerical GCD has been considered by a number of authors – for a summary see for instance [11, Section 2.3] or [12, Section 5] and the references cited therein. Schönhage formulated the task of computing a Quasi–GCD [18]. His algorithm [18, Section 3] is fast and probably numerically quite stable since the technique of pivoting is applied. However, for the conclusions of [18] it is required that the coefficients of a, b are available to an arbitrarily high precision.

A correct mathematical definition for an approximate GCD with precision ϵ was given by Karmarkar and Lakshman [15], together with a discussion of optimization methods for solving this problem. Such an approach is certainly numerically stable but quite expensive (the authors establish polynomial complexity). Before that Corless et al. [11] emphasized the role of the singular values of the underlying Sylvester matrix for determining the degree of an approximate GCD. A further account of this question is given by Emiris et al. [12] who also considered singular values of submatrices of a Sylvester matrix. There are numerically stable methods for computing the SVD, each however having a complexity of $\mathcal{O}((m+n)^3)$ since they do not take into account the special structure of a Sylvester matrix.

For methods based on first determining the degree of an approximate GCD and then the GCD itself, there remains the problem of actually computing this quantity. Rather than expensive optimization techniques, many authors [11, 12, 17] propose Euclidean-like PRS algorithms. However these methods do not take into account the problem of numerical stability which, even for perturbations much larger than the machine precision, should be a serious concern (see Example 2.1 below). In Appendix B we describe cases where our algorithm even determines a numerical GCD in a numerical correct way, confirming partially an open conjecture of Cabay and Meleshko [16].

The remainder of the paper is organized as follows. In Section 2 we give an example showing that Euclid's algorithm has important drawbacks in finite precision arithmetic. We then introduce the concept of unimodular reductions, and show their use for solving (3) and (4). In Section 3 we describe the algorithm COPRIME for computing these quantities in a numerically stable manner. We give an interpretation of our look–ahead strategy in terms of the condition number of Trudi submatrices, showing again why Euclid's algorithm may fail in a numerical setting. A brief proof of stability of the algorithm is presented in Section 4, where we extend ideas of Cabay and others [1, 7, 9]. In Section 5 we report on numerical experiments with our algorithm while Section 6 gives a summary along with topics for future research. We also include, in Appendix A precise lower and upper bounds for the condition number of Trudi matrices in terms of quantities computed by COPRIME and discuss, in Appendix B, the computation of numerical GCD's by our numerically stable method.

Notation: Following [3], for the remainder of this paper we make use of the following notation for polynomials, vectors and their respective norms. For $c \in \mathbb{C}[z]$, $c(z) = c_0 + .. + c_n z^n$ we set $\vec{c} = (c_0, .., c_n)^T$ as the vector of coefficients. A norm for $\mathbb{C}[z]$ is given by

$$||c|| = ||\vec{c}||_1 = \sum_j |c_j|,$$

the classical Hölder vector norm. This definition canonically extends to Laurent polynomials (with possibly negative powers of z). Our norm for $\mathbb{C}[z]^{r\times s}$, the space of $r \times s$ matrices of polynomials will be

$$||(c_{j,k})|| = ||(||c_{j,k}||)||_1 = \max_k \sum_j ||c_{j,k}||.$$

Thus for example, $||(a,b)|| = \max\{||a||, ||b||\} = \max\{\sum |a_j|, \sum |b_j|\}$. This choice of norms is motivated by the property $||c \cdot d|| \leq ||c|| \cdot ||d||$ being valid for any scalar or matrix polynomials c, d of suitable size (which is important for establishing (2)).

2 Unimodular Reduction

The aim of this and the following section is to describe how to compute solutions of equations (3) and (4) (i.e., the first and the last column of the inverse of the Sylvester matrix) and thus the quantity κ of (2) in an efficient, numerically correct way. We suppose for convenience[2] that $\deg a = m > \deg b = n$. Furthermore, we may suppose without loss of generality that the input polynomials are scaled with $1/2 \leq ||(a,b)|| \leq 1$.

[2] In the case $\deg a < \deg b$ we may interchange a and b. The case $\deg a = \deg b$ is excluded in order to simplify later considerations. However, here one may subtract a scalar multiple of a from b in order to have a degree reduction, with the multiplier being of modulus less than or equal to 1. The corresponding cofactors of the original polynomials a, b are obtained from those of the new ones by a simple transformation, and the errors induced by these floating point operations can be easily bounded.

According to the particular structure of the Sylvester matrix, there exist a number of fast ($\mathcal{O}(m^2)$) and superfast ($\mathcal{O}(m \cdot \log^2 m)$) inversion algorithms. There are also corresponding algorithms that have generalizations for use on a vector processor (for a summary, see, e.g., [4]). However, there seem to be only two fast methods where numerical stability has been established, both of them actually being *weakly stable*. Weak stability means (using the classification of Bunch [6] modified by Bojanczyk et al. [5]) that we compute a numerical solution with small residual – a property which will be sufficient for our purposes. A first possibility may be to apply the fast QR decomposition algorithm of Bojanczyk, Brent and de Hoog [5]. This method was originally proposed for solving Toeplitz systems of equations, but should equally be applicable in the more general case of Sylvester matrices.

In the present paper, we prefer to take advantage of the fact that the cofactor equation (3) is mathematically equivalent to determining some Padé approximant v/u to the function $-b/a$ (at infinity). This allows us to use the Cabay-Meleshko algorithm [7], the first fast numerically stable algorithm for computing Padé approximants (at zero). In our context, we will need some modifications since from a numerical point of view it seems to be numerically sensitive to explicitly form the power series expansion of $-b/a$. Note that (3) may be understood as a Hermite–Padé approximation problem (at infinity) of the two functions a, b. Generalizing [7], a weakly stable method for Hermite–Padé approximation has been given in [9] (see also [8, 10, 16]). However, for the proof of weak stability given in [9] it is necessary to assume that the coefficients in the power series expansion of $1/a$ at infinity do not become very large, which for our setting is an undesirable strong restriction. Therefore we prefer to compute simultaneously so–called *associated vectors* which allows us to monitor the quantity $\rho_\ell(a, b)$ (used already in [3] and defined in (15) below) and on the other hand enables us to solve at the same time for the cofactors in (4). It will also be appropriate to replace the variable z by $1/z$ in the algorithm NPADE of Cabay and Meleshko. This minor modification, discussed already in [1], allows one to understand the main recurrence of NPADE as a transformation of the ideal generated by the polynomials a, b. In addition, it illustrates the connections to classical methods for computing GCD's.

Euclid's algorithm consists of determining a finite sequence of polynomials $r_{-1} = a, r_0 = b, r_1, \ldots, r_{\ell-1}, r_\ell \neq 0, r_{\ell+1} = 0$ referred to as *remainders*, with r_ℓ being the GCD of a, b. However, as already mentioned in [18, Section 3], it is not possible to create a correct numeric version of Euclid's algorithm by the naive method of doing polynomial divisions followed by converting all coefficients below a certain threshold into 0, even if we are willing to do our computations in higher precision.

Example 2.1 *Let*

$$a(z) = z^4 + z^3 + (1+\eta)\, z^2 + \eta z + 1 \text{ and } b(z) = z^3 - z^2 + 3\, z - 2.$$

Then the remainder after one step of Euclid's algorithm is $r_1(z) = \eta z^2 + (\eta - 4)\, z + 5$. *If* η *is just below a given threshold then one can eliminate terms of* r_1 *to obtain the correct numerical GCD (see the first part of Table 1). However, should* η *be just above the threshold then the next division step will introduce significant numerical errors making subsequent results meaningless.*

To make this statement more precise, let us adapt the following model (which is close to the one adapted in the procedure `quo/float` *of the computer algebra system MAPLE): before constructing the quotient or the remainder in an individual step, we check for each polynomial whether the modulus of a coefficient is smaller than a threshold parameter times the modulus of the largest coefficient of the polynomial. Such coefficients will be replaced by zero. We will also assume that the computation of the quotients and the remainders is done in higher precision, and therefore further errors due to floating point operations may be neglected.*

k	k^{th} quotient: q_k	k^{th} remainder: r_k
-1		$z^4+z^3+z^2+1$
0	$z+2$	$z^3-z^2+3\,z-2$
1	$-\frac{z^2}{4}-\frac{z}{16}-\frac{53}{64}$	$-4\,z+5$
2	$-\frac{256\,z}{137}+\frac{320}{137}$	$\frac{137}{64}$
3		0
-1		$z^4+z^3+(\eta+1)\,z^2+\eta z+1$
0	$z+2$	$z^3-z^2+3\,z-2$
1	$\frac{4}{\eta^2}+\frac{z-2}{\eta}$	$\eta z^2+(\eta-4)\,z+5$
2	$-\frac{\eta^2}{4}+\left(-\frac{1}{8}+\frac{z}{16}\right)\eta^3$	$\left(\frac{16}{\eta^2}-\frac{17}{\eta}\right)z-\frac{20}{\eta^2}+\frac{10}{\eta}$
3		$\frac{17\,\eta^2 z^2}{16}-\frac{11\,\eta^2 z}{4}+\frac{5\,\eta^2}{4}$
-1		$z^4+z^3+(\eta+1)\,z^2+\eta z+1$
0	$z+2$	$z^3-z^2+3\,z-2$
1	$\frac{4}{\eta^2}+\frac{z-2}{\eta}$	$\eta z^2+(\eta-4)\,z+5$
2	$-\frac{\eta^2}{4}+\left(-\frac{1}{8}+\frac{z}{16}\right)\eta^3+\left(-\frac{11}{1024}+\frac{17\,z}{256}\right)\eta^4$	$\left(5-\frac{17}{\eta}+\frac{16}{\eta^2}\right)z-\frac{20}{\eta^2}-2+\frac{10}{\eta}$
3		$\frac{209\,\eta^3 z^2}{256}-\frac{99\,\eta^3 z}{1024}+\frac{137\,\eta^2}{256}-\frac{73\,\eta^3}{512}$

Table 1: Euclid's algorithm may fail for numerical data, see Example 2.1.

In Table 1 we display the results obtained for three different threshold parameters, namely between η^{d-1} and η^d for $d=1,2,3$. In fact, as claimed above, we have no problem detecting the correct GCD for $d=1$. In the second part ($d=2$), we stopped the algorithm because the remainder r_3 does not have a degree less than that of r_2. If we suspect the leading coefficient to be zero, then also the other coefficients which are of the same magnitude would be zero. In other words, the correct GCD would be r_2 which seems to be of degree 1. Thus the answer furnished by Euclid's algorithm has no significance. Finally, in the case $d=3$ we meet a similar problem. Here one of the terms in r_3 has a different magnitude. In this case we would get the right answer if we are allowed to switch the threshold parameter in the algorithm, an idea that seems to be quite sensitive numerically.

We will see later in Remark 3.1 that this failure of Euclid's algorithm may be explained by the fact that a certain Trudi submatrix is severely ill–conditioned. □

Euclid's algorithm can be put into a matrix polynomial framework by observing that

$$\Big(r_{j-1},r_j\Big)\cdot\begin{bmatrix}0 & 1\\ 1 & -q_j\end{bmatrix}=\Big(r_j,r_{j+1}\Big),\quad \deg q_j=\deg r_{j-1}-\deg r_j>0.$$

Accumulating the matrix factors for $(\tilde a,\tilde b)=(r_j,r_{j+1})$ gives matrix equations of the form

$$\Big(a,b\Big)\cdot U=\Big(\tilde a,\tilde b\Big),\quad \deg a>\deg b\geq\deg\tilde a=:\deg a-k>\deg\tilde b, \tag{5}$$

with the elements of the 2×2 matrix polynomial U having degree bounds given by

$$\deg U\leq\begin{bmatrix}m-n+k-1 & m-n+k\\ k-1 & k\end{bmatrix}. \tag{6}$$

From a numerical standpoint it is better to consider the more general recursions defined by equations (5) and (6), rather than the less flexible Euclidean algorithm.

For the remainder of this paper we will refer to matrices U with $\det U \neq 0$ verifying (5) and (6) as *unimodular reductions (UR) of order k*, and say that $(\tilde{a}, \tilde{b})$ is obtained from (a, b) by an unimodular reduction of order k. The matrix U will be called *scaled UR* if, in addition, both columns have a norm between $1/2$ and 1. We will also need a so–called *associated vector of order k* verifying

$$(a(z), b(z)) \cdot \underline{U}(z) = z^{n+k-1} + \tilde{c}(z), \quad \deg \tilde{c} \leq m - k - 1, \quad \deg \underline{U} \leq \begin{bmatrix} n - m + k - 1 \\ k - 1 \end{bmatrix}.$$

The significance of these quantities becomes clear from the following lemma.

Lemma 2.2 **(a)** *Any UR is unimodular, that is, it has a polynomial inverse.*

(b) *Let $(\tilde{a}, \tilde{b})$ be obtained from (a, b) by some unimodular reduction. Then the ideals $< a, b >$ and $< \tilde{a}, \tilde{b} >$ are equal.*

(c) *The polynomials a, b are coprime if and only if there exists a unimodular reduction $U^{(m)}$ of order m if and only if there exists a unique associated vector $\underline{U}^{(m)}$ of order m.*

(d) *With the notation of part (c), solutions of (3) and (4) are obtained from the first column of $U^{(m)}$, and from $\underline{U}^{(m)}$, respectively. Furthermore,*

$$\epsilon(a, b) \geq \min\Big\{ \frac{|\tilde{a}(0)|}{||U^{(m)} \cdot (1, 0)^T||}, \frac{1}{||\underline{U}^{(m)}||} \Big\}. \tag{7}$$

(e) *With the notation of part (c), if in addition $U^{(m)}$ is scaled then $|\det U^{(m)}(0)| \cdot ||(v, u)^T|| \in [1/8, 2]$.*

Proof: For a proof of (a), let the entries of a UR U be denoted by

$$U = \begin{bmatrix} v_1 & v_2 \\ u_1 & u_2 \end{bmatrix} \tag{8}$$

and observe that

$$\underbrace{\begin{bmatrix} a & b \\ 0 & 1 \end{bmatrix}}_{\deg \det(\cdot) = m} \cdot U = \underbrace{\begin{bmatrix} \tilde{a} & \tilde{b} \\ u_1 & u_2 \end{bmatrix}}_{\deg \det(\cdot) \leq (m-k)+k}.$$

Therefore, $\deg \det U \leq 0$, and $\det U = \det U(0) = lc(u_2) \cdot lc(\tilde{a})/lc(a)$. Thus assertion (a) follows from Cramer's rule $U(z)^{-1} = (1/\det U(z)) \cdot adj U(z)$. In order to prove (b), notice that $< \tilde{a}, \tilde{b} >$ is contained in the ideal $< a, b >$ by (5). The inclusion in the other direction follows using a similar argument combined with (a).

Let now $U^{(m)}$ be a UR of order m. From (5) it follows that $\tilde{b} = 0$, and $\tilde{a}$ is a constant. Thus a, b are coprime according to (b). On the other hand, if we solve the diophantine equation (3) then a UR of order m is given by

$$U = \begin{bmatrix} v & -b \\ u & a \end{bmatrix}.$$

This shows the first equivalence of part (c), and the second one follows from the observation that associated vectors of order m are solutions of (4). In order to prove (d), notice that a UR $U^{(m)}$ of order m necessarily is obtained by multiplying the columns of the above matrix U with some scalars (in particular the first one with $\tilde{a}(0)$). Therefore estimate (7) is just a reformulating of (2). Finally, for a proof of part (e) one uses the scaling conditions for $U^{(m)}$ and (a, b). We omit the details. □

3 The algorithm COPRIME

We see from Lemma 2.2(d) that we obtain the solutions of (3) and (4) and thus a lower bound for $\epsilon(a,b)$ by determining an (exact) scaled UR of order m together with its (exact) associated vector. However, using finite precision arithmetic we instead obtain numerical counterparts of these quantities. We will show in Theorem 4.3(c) below that a statement similar to (7) holds as well.

Following Cabay and Meleshko [7], a (numerical) scaled UR of order m will be determined by recurrence in terms of (numerical) scaled UR's of lower order. From the considerations below it follows that a similar procedure may be applied for determining the (numerical) associated vectors. [3] Thus we successively construct (numerical) scaled UR's of order k of (a,b) together with (numerical) associated vectors

$$(a^{(k)}, b^{(k)}) \leftarrow (a,b) \cdot U^{(k)}, \quad z^{n+k-1} + c^{(k)} \leftarrow (a,b) \cdot \underline{U}^{(k)}, \tag{9}$$

for increasing k lying between 1 and m. Here $U^{(k+s)}$ and $\underline{U}^{(k+s)}$ are computed for some positive integer s (the *stepsize*) by the floating point operations

$$U^{(k+s)} \leftarrow U^{(k)} \cdot U^{(k,k+s)}, \quad \underline{U}^{(k+s)} \leftarrow z^s \cdot \underline{U}^{(k)} - U^{(k)} \cdot \underline{U}^{(k,k+s)}, \tag{10}$$

together with the initializations

$$U^{(0)} := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \underline{U}^{(0)} := \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{11}$$

The transition factor $U^{(k,k+s)}$ is a 2×2 matrix polynomial computed by constructing a (numerical) UR of order s of $(a^{(k)}, b^{(k)})$

$$(a^{(k+s)}, b^{(k+s)}) \leftarrow (a^{(k)}, b^{(k)}) \cdot U^{(k,k+s)}, \tag{12}$$

that is, by equating coefficients in (5). The updating vector $\underline{U}^{(k,k+s)}$ with polynomial components having degrees bounded by $s-1$ is chosen in order to satisfy the constraints for $\underline{U}^{(k+s)}$ to be an associated vector of order $k+s$. Namely we want that the right hand side in

$$c^{(k+s)} \leftarrow z^s \cdot c^{(k)} - (a^{(k)}, b^{(k)}) \cdot \underline{U}^{(k,k+s)}$$

has a degree bounded by $m-k-s-1$. To be more precise, we define the matrix

$$M_s^{(k)} := \begin{bmatrix} a_{m-k}^{(k)} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ a_{m-k-1}^{(k)} & a_{m-k}^{(k)} & \ddots & \vdots & b_{m-k-1}^{(k)} & \ddots & \vdots \\ \vdots & & \ddots & 0 & \vdots & \ddots & 0 \\ \vdots & & & a_{m-k}^{(k)} & \vdots & & b_{m-k-1}^{(k)} \\ \vdots & & & \vdots & \vdots & & \vdots \\ a_{m-k-2s+1}^{(k)} & \cdots & \cdots & a_{m-k-s}^{(k)} & b_{m-k-2s+1}^{(k)} & \cdots & b_{m-k-s}^{(k)} \end{bmatrix},$$

where by definition quantities with negative indices are equal to zero. We solve the three systems[4] with $2s$ unknowns and $2s$ equations, [5]

$$M_s^{(k)} \cdot x_1 = (0, \ldots, 0, 1), \quad M_s^{(k)} \cdot x_2 = -(b_{m-k-j}^{(k)})_{j=1,..,2s}, \quad M_s^{(k)} \cdot x_3 = (c_{m-k-j}^{(k)})_{j=1,..,2s}. \tag{13}$$

[3] This approach can be compared with a method given in [19] where, besides a NPADE–like recurrence, additional formulas are given in order to solve Toeplitz systems of equations.

[4] In the case $m-n>1$ and $k=0$, we consider slightly different systems. Note that the matrices $M_s^{(0)}$ are singular for $s=1,2,\ldots,m-n-1$. Thus we start our iterations with $s=m-n$, and drop in $M_s^{(0)}$ the first $m-n-1$ rows and columns. The system (13) has to be adapted by omitting the first $m-n-1$ equations, and the first $m-n-1$ unknowns.

[5] For the first two systems of equations compare [7, p.747], where an additional scaling is considered.

Then it is easily verified that the unknown quantities $U^{(k,k+s)}$ and $\underline{U}^{(k,k+s)}$ may be computed by

$$\left(U^{(k,k+s)}(z), \underline{U}^{(k,k+s)}(z)\right) = \begin{bmatrix} z^{s-1} & \cdots & z^1 & z^0 & 0 & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & 0 & z^{s-1} & \cdots & z^1 & z^0 \end{bmatrix} \cdot \left(x_1, x_2, x_3\right) + \begin{bmatrix} 0 & 0 & 0 \\ 0 & z^s & 0 \end{bmatrix}.$$

The "small" systems (13) are solved by some (weakly) stable method such as Gaussian elimination with partial pivoting [7] or QR-decompositions combined with bordering techniques [1]. The latter has the advantage of being always weakly stable, and (for large s) requiring less arithmetic operations. Note that if $M_s^{(k)}$ is singular then we increase s by 1 and restart determining new candidates $U^{(k+s)}$, $\underline{U}^{(k+s)}$. Let us also mention in this context that we will obtain numerical quantities satisfying the "correct" degree constraints (on an element basis)

$$\deg U^{(k)} \leq \begin{bmatrix} n-m+k-1 & n-m+k \\ k-1 & k \end{bmatrix}, \quad \deg \underline{U}^{(k)} \leq \begin{bmatrix} n-m+k-1 \\ k-1 \end{bmatrix}. \tag{14}$$

For an error analysis it is important that the quantities involved in the "small" systems of equations (13), that is, some of the coefficients of $(a^{(k)}, b^{(k)}, c^{(k)})$, are computed with help of (9) only at the moment where they are required. This is in contrast to Euclid's algorithm, where one updates the remainders $(a^{(k)}, b^{(k)})$ by (12). Therefore we compute explicitly the transformation matrices $U^{(k)}$, as done also, for example, in the extended Euclidean algorithm.

From the above description it becomes clear that the choice of the stepsize s is important. In fact, a small s in each step leads to an overhead of $\mathcal{O}(m^2)$ arithmetic operations, whereas for numerical reasons it may be more appropriate to choose a larger s, e.g., in order to "jump" over singular or unstable subproblems. Following [7] and its generalizations [1, 9, 16, 19], a "good" stepsize is determined by the following "look–ahead" procedure: we compute for fixed k and for $s = 1, 2, \ldots$ successively $U^{(k,k+s)}$, and obtain a candidate $U^{(k+s)}$ by (10). This is then scaled by multiplication on the right with a diagonal matrix containing powers of two (that is, we rescale implicitly $U^{(k,k+s)}$). Afterwards, we check whether the quantity $|\det U^{(k+s)}(0)|$ of our candidate is larger than a certain given threshold parameter ϵ (this threshold being connected to the desired precision of the output, see Theorem 4.2 below). [6] Equally, we check whether $\underline{U}^{(k)}$ is sufficiently small. If this is the case, then our candidate is accepted as our new accumulated transformation matrix, and we may increase k by s. Otherwise we forget about our candidate, and the "look–ahead" process is continued by increasing s by 1. We also introduce a set $\mathcal{A} \subset \{0, 1, 2, \ldots, m\}$ of indices of scaled UR accepted by our criterion.

The order of computation is schematically described in Table 2. For further details and proofs we refer to [1, 7]. In addition, numerical experiments seem to indicate [7, 16, 19] that, for correctly scaled dense data, the case of stepsizes s larger than 3 is rather unlikely, leading in general to a total cost of $\mathcal{O}(m^2)$ arithmetic operations, and to $\mathcal{O}(m)$ storage requirements. Sparse input polynomials would typically result in larger stepsizes and require more specialized routines for solving the small subproblems to remain efficient. Finally, we also remark that there are pathological cases where the step sizes can be as high as m leading to $O(m^3)$ arithmetic operations (in the case where the algorithm uses QR decomposition to solve the subproblems.)

Remark 3.1 *The kth Trudi matrix $S_k(a, b)$ obtained by dropping $2(m-k)$ suitable columns and rows of the Sylvester matrix $S(a, b)$ (see Appendix A) is useful in exact arithmetic to determine the degree of the GCD of a and b, see, e.g., [13]. From the estimates of $||S_k(a, b)^{-1}||$ in terms of scaled UR's and*

[6] Note that $|\det U^{(k+s)}(0)| < 1$ because of scaling.

Method:	We iteratively construct scaled UR's $U^{(k)}$ of (a,b) of order $k \in \mathcal{A}$ together with associated vectors $\underline{U}^{(k)}$ (for increasing k between 1 and m).
Input:	Two polynomials a, b with $\deg a = m > \deg b$. A stability parameter ϵ of order of the cubic root of the machine precision.
Output:	If $m \in \mathcal{A}$: RETURN $\min\left\{ \lvert a^{(m)}(0)\rvert / \lVert U^{(m)} \cdot (1,0)^T \rVert, 1/\lVert \underline{U}^{(m)} \rVert \right\}$ as numerical lower bound of $\epsilon(a,b)$ (see (7) and Theorem 4.3(c)). If $m \notin \mathcal{A}$: message since this quantity does not exist or is "too small".
Initialization:	$k = 0$, $\mathcal{A} = \{\}$, and (11).
Single Step:	For $s = 1, 2, \ldots$: Compute UR $U^{(k,k+s)}$ of $(a^{(k)}, b^{(k)})$ of order s, and $\underline{U}^{(k,k+s)}$ Method: Solve (13) (if $\det M_s^{(k)} = 0$ then increase s and restart). New scaled UR candidate: $U^{(k+s)}$ scaled counterpart of $U^{(k)} \cdot U^{(k,k+s)}$.
Exit s–loop:	If $\lvert \det U^{(k+s)}(0)\rvert > \epsilon$ and $\lVert \underline{U}^{(k+s)} \rVert < 1/\epsilon$. In this case: $k \leftarrow k+s$, $\mathcal{A} \leftarrow \mathcal{A} \cup \{k\}$.
Exit ALGO:	If $k + s = m$.
Complexity:	In most cases: $\mathcal{O}(m^2)$.

Table 2: The algorithm COPRIME

associated vectors given in Theorem A.1 below it becomes clear that our "look–ahead" strategy allows us to only encounter subproblems of type (9) with a corresponding well–conditioned matrix of coefficients $S_k(a,b)$. In contrast, in the classical Euclidean algorithm there is no freedom of choosing a stepsize s, since we only encounter "small" triangular systems. In other words, we just take the first existing UR, though the corresponding quantity $\lvert \det U(0)\rvert$ might be very small. Thus it might happen that some of the unimodular reductions of the Euclidean algorithm are ill–conditioned problems. This is the fundamental problem of using the Euclidean algorithm in a numerical setting: solutions are sometimes built upon solutions of ill-conditioned subproblems making the final answers highly inaccurate.

Our observation may be nicely illustrated with help of the polynomials a, b of Example 2.1. Here

$$S_2(a,b) = \begin{bmatrix} 1+\eta & -1 & 3 \\ 1 & 1 & -1 \\ 1 & 0 & 1 \end{bmatrix}, \quad S_2(a,b)^{-1} = \begin{bmatrix} \eta^{-1} & \eta^{-1} & -2\eta^{-1} \\ -2\eta^{-1} & 1-2\eta^{-1} & 1+4\eta^{-1} \\ -\eta^{-1} & -\eta^{-1} & 1+2\eta^{-1} \end{bmatrix},$$

and thus $\lVert S_2(a,b)\rVert = 5$ and $\lVert S_2(a,b)^{-1}\rVert = 2 + 8\eta^{-1}$ for small $\eta > 0$, showing that this Trudi submatrix is ill–conditioned. In fact, we observed in Example 2.1 that the occurrence of a remainder of degree 2 in Euclid's algorithm makes the results meaningless. In the algorithm COPRIME, we would just not accept the candidate UR of order 2 since here $\lVert \underline{U}^{(2)}\rVert = 2 + 8\eta^{-1}$, and, more importantly, $1/\lvert \det U^{(2)}(0)\rvert \approx 64/\eta^2$, are too large. Such a flexibility of "jumping" over unstable subproblems is not available with Euclid's algorithm. □

4 A modified proof of weak stability

The aim of this section is to give bounds for the coefficients of the "undesired" powers (see Theorem 4.2) obtained if one forms the differences of the left and the right hand sides of (9). We will see in Theorem 4.3 that these bounds are sufficiently sharp to insure that Lemma 2.2 remains essentially valid. In particular, the output of COPRIME is correct (up to factor 2) even for finite precision arithmetic.

Our algorithm COPRIME for computing numerical scaled UR's is based on the Cabay–Meleshko algorithm [7] and its generalization [9]. However, as mentioned in the previous section, the error analysis given in [7, 9] requires an additional assumption which in our context will often not be verified (see, e.g., Example 5.1 below): the coefficients in the power series expansion at infinity of $a(z)^{-1}$ have to stay "small". As we will show below, instead of this restrictive assumptions it will be sufficient to know that (for $1 \leq \ell \leq m+n$) there exist polynomials g_a, g_b of "relatively small" norm verifying

$$\deg g_a < \ell, \quad \deg g_b < \ell, \quad z^n a(z) g_a(z) + z^m b(z) g_b(z) = z^{m+n+\ell-1} + \mathcal{O}(z^{m+n-1})_{z\to\infty}. \tag{15}$$

Allowing only $g_b = 0$ means that we recover the requirements on $a(z)^{-1}$ of [7, 9]. However, there may be much better choices, for example (numerical) associated vectors. Also, in the case $\ell \geq m$ we can take the solutions of the diophantine equation (4), namely $g_a(z) = z^{\ell-n} \cdot \underline{v}(z)$ and $g_b(z) = z^{\ell-m} \cdot \underline{u}(z)$. Following [3, Corollary 3.3], we will denote by $\rho_\ell(a,b)$ the minimum of the set of all products $||(a,b)|| \cdot ||(g_a, g_b)^T||$ where the pair (g_a, g_b) verifies (15).

Let μ be the machine precision. In order to simplify the presentation of our results, in what follows we will state our estimates in the form $|g| \leq C(j) \cdot |h|$ where $C(j)$ stands for some (explicit) polynomial in j, not necessarily the same at each occurrence (in our case, all these polynomials will have a degree not exceeding 3). The coefficients of such a polynomial C depend neither on the input data a, b nor the stability parameter ϵ. This notation is useful because changes in the choice of the norm will be absorbed by a change of the polynomial C. Furthermore we require the *cut operator* acting on $\mathbb{C}[z]$ by

$$\Pi_{i,j}(\sum_{\ell=0}^{s} a_\ell z^\ell) = \sum_{\ell=i}^{j} a_\ell z^\ell.$$

We start by introducing residual polynomials (α, β, γ) which vanish for exact arithmetic, and should be small in norm for our numerical unimodular reductions.

Definition 4.1 *Define (as in (9))*

$$a^{(k)} := \Pi_{0,m-k}(a,b) \cdot U^{(k)} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \alpha^{(k)} := \Pi_{m-k+1,n+k-1}(a,b) \cdot U^{(k)} \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$
$$b^{(k)} := \Pi_{0,m-k-1}(a,b) \cdot U^{(k)} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad \beta^{(k)} := \Pi_{m-k,n+k}(a,b) \cdot U^{(k)} \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$
$$c^{(k)} := \Pi_{0,m-k-1}(a,b) \cdot \underline{U}^{(k)}, \quad \gamma^{(k)} := \Pi_{m-k,n+k-1}\Big((a,b) \cdot \underline{U}^{(k)} - z^{n+k-1}\Big),$$

and therefore

$$(a,b)(U^{(k)}, \underline{U}^{(k)}) = (a^{(k)}, b^{(k)}, z^{n+k-1} + c^{(k)}) + (\alpha^{(k)}, \beta^{(k)}, \gamma^{(k)}).$$

The scaled UR $U^{(k)}$ will be referred to as well–behaved if $k \in \mathcal{A}$, and

$$8 \cdot \rho_{n-m+2k}(a,b) \cdot ||(\alpha^{(k)}, \beta^{(k)})|| \leq |\det U^{(k)}(0)|.$$

Similarly, the associated vector $\underline{U}^{(k)}$ *will be referred to as well–behaved if* $k \in \mathcal{A}$, *and* $2 \cdot ||\gamma^{(k)}|| \leq 1$. □

Cabay and Meleshko showed in [7, Theorem 6.5, Theorem 6.9] that the global error $(\alpha^{(k)}, \beta^{(k)})$ is obtained by an additive superposition of (small) local errors, and thus is controllable. [7] In order to restate their findings in our setting, and to state a similar result for $\gamma^{(k)}$, we use the abbreviation $\mathcal{A}_k := \{j \in \mathcal{A} : j < k\}$, with corresponding stepsizes given by $s_k := k - \max \mathcal{A}_k$ (or by $s_k = k$ if $\mathcal{A}_k$ is empty).

Theorem 4.2 *Let* $U^{(k)}$ *be a candidate in algorithm COPRIME, with corresponding (numerical) associated vector* $\underline{U}^{(k)}$. *Furthermore, suppose that all* $U^{(j)}$ *and* $\underline{U}^{(j)}$ *for* $j \in \mathcal{A}_k$ *are well–behaved. Then*[8]

$$||(\alpha^{(k)}, \beta^{(k)})|| \leq \mu \cdot \Big(\sum_{j \in \mathcal{A}_k} \frac{C(j - s_j, s_j)}{|\det U^{(j)}(0)| \cdot |\det U^{(j-s_j)}(0)|} + \frac{C(k - s_k, s_k)}{|\det U^{(k-s_k)}(0)|} \Big),$$

and

$$||\gamma^{(k)}|| \leq \mu \cdot \sum_{j \in \mathcal{A}_k \cup \{k\}} C(j - s_j, s_j) \cdot \frac{||\underline{U}^{(j-s_j)}|| + ||\underline{U}^{(j)}||}{|\det U^{(j-s_j)}(0)|}.$$

By taking into account Theorem 4.3(a) below, we see that our look–ahead criterion is just designed to insure that scaled UR's (and the associated vectors) accepted by this criterion will also be well–behaved, at least for sufficiently large ϵ. We will not quantify exactly such a choice of ϵ, since in general the estimate of Theorem 4.2 leads to a large overestimation of $||(\alpha^{(k)}, \beta^{(k)})||$ and $||\gamma^{(k)}||$. For improved look–ahead criteria based also on numerical experiments we refer to [1, 7, 9, 10, 16, 19].

We still require a lower bound of $\epsilon(a, b)$ in terms of the numerical $U^{(m)}$ and $\underline{U}^{(m)}$ determined by the algorithm COPRIME, that is, we look for a floating point analogue of (7). This and some further properties are summarized in

Theorem 4.3 **(a)** *If* $\underline{U}^{(k)}$ *is well–behaved, then* $\gamma_{n-m+2k}(a, b) \leq 2 \cdot ||\underline{U}^{(k)}||$.

(b) *If* $U^{(k)}$ *is well–behaved, then it is also close to a unimodular matrix. More precisely, there exists a* $V^{(k)} \in \mathbb{C}^{2\times 2}[z]$ *with*

$$||V^{(k)}|| \leq 4/|\det U^{(k)}(0)|, \quad U^{(k)}(z) \cdot V^{(k)}(z) = V^{(k)}(z) \cdot U^{(k)}(z) = I_2 + \mathcal{O}(z^{m+n+1})_{z \to 0}.$$

(c) *Suppose that both* $U^{(m)}$ *and* $\underline{U}^{(m)}$ *are well–behaved. Then the output of algorithm COPRIME is a lower bound at least for* $2 \cdot \epsilon(a, b)$.

From Theorem 4.3(c) we obtain *reliable* lower bounds for $\epsilon(a, b)$ provided that $m \in \mathcal{A}$. Also, from Lemma 2.2(e) we see that we may expect $m \in \mathcal{A}$ if the lower bound in (2) is still larger than 8ϵ. Thus, if our algorithm COPRIME fails to solve the diophantine equations (i.e., $m \notin \mathcal{A}$) then this indicates that the cofactors are too large in norm, and thus the lower bound proposed in (2) is not useful.

[7] A multiplicative superposition might lead to an exponential and therefore uncontrollable growth of the size of the error.

[8] For establishing the assertions of the theorem we have to assume in addition that $17 \cdot m \cdot \mu \leq \epsilon$, and that $m \cdot \mu/\epsilon^2$ is bounded by some modest constant. These conditions on ϵ may be dropped for a first order error analysis.

Proof of Theorem 4.3: Let us start with the following simple observation: if g is a polynomial verifying $2 \cdot ||g - g(0)|| \leq |g(0)|$ then with h denoting a partial sum of order ℓ of the expansion of $1/g$ around zero we have

$$||h|| \leq \frac{1}{|g(0)|} \sum_{j=0}^{\ell} ||\frac{g - g(0)}{g(0)}||^j \leq \frac{1}{|g(0)| - ||g - g(0)||} \leq \frac{2}{|g(0)|}. \tag{16}$$

Proof of part (a): By definition of $\gamma^{(k)}$ there holds with $\ell := n - m + 2k$

$$(a, b) \cdot \underline{U}^{(k)} \cdot \frac{1}{z^{n+k-1} + \gamma^{(k)}} = 1 + \frac{c^{(k)}}{z^{n+k-1} + \gamma^{(k)}} = 1 + \mathcal{O}(z^{-\ell})_{z \to \infty}.$$

Let $(g_a, g_b)^T$ be the partial sum of order $\ell - 1$ of the power series expansion at infinity of $diag\,(z^{m+\ell-1}, z^{n+\ell-1}) \cdot \underline{U}^{(k)}/(z^{n+k-1} + \gamma^{(k)})$. One verifies without difficulties using (14) that g_a, g_b are polynomials verifying (15). It remains to discuss their norm. Let $g(z) := 1 + z^{n+k-1} \cdot \gamma^{(k)}(1/z) \in \mathbb{C}[z]$. By assumption there holds $2 \cdot ||\gamma^{(k)}|| = 2 \cdot ||g - 1|| \leq 1$ and thus with $\gamma := \gamma^{(k)}_{n+k-1}$ we obtain

$$|g(0)| = |1 + \gamma| \geq 1 - 2 \cdot |\gamma| \geq 2 \cdot ||\gamma^{(k)}|| - 2 \cdot |\gamma| = 2 \cdot ||g - g(0)||.$$

It follows from (16) that the partial sum h of order $\ell - 1$ of the power series expansion of $1/g$ at zero has a norm bounded by $2/[2 \cdot |1 + \gamma| - (1 - 2 \cdot |\gamma|)] \leq 2$, and thus

$$\rho_\ell(a, b) \leq ||(a, b)|| \cdot ||(g_a, g_b)^T|| \leq ||\underline{U}^{(k)}|| \cdot ||h|| \leq 2 \cdot ||\underline{U}^{(k)}||.$$

Proof of part (b): Let $g(z) := \det U^{(k)}(z)$, and denote by h the partial sum of order $m+n$ of the power series expansion of $1/g$ at zero. Then with $V^{(k)} := h \cdot adj\, U^{(k)}$ we get

$$U^{(k)} \cdot V^{(k)} = V^{(k)} \cdot U^{(k)} = I_2 \cdot g \cdot h = I_2 + \mathcal{O}(z^{m+n+1})_{z \to 0},$$

moreover, $||V^{(k)}|| \leq ||h|| \cdot ||adj\, U^{(k)}|| \leq 2 \cdot ||h||$. In view of (16), for establishing (b) it will therefore be sufficient to show the relation $2 \cdot ||g - g(0)|| \leq |g(0)|$. Let $\ell := n - m + 2k$, then we find polynomials g_a, g_b as in (15) verifying $||(g_a, g_b)^T|| = \rho_\ell(a, b)/||(a, b)|| \leq 2\rho_\ell(a, b)$. We have

$$(a, b) \cdot g = (a, b) \cdot U^{(k)} \cdot adj\, U^{(k)} = (a^{(k)}, b^{(k)}) \cdot adj\, U^{(k)} + (\alpha^{(k)}, \beta^{(k)}) \cdot adj\, U^{(k)},$$

and $\deg g \leq \ell - 1$ by (14). Consequently, g is completely determined via

$$\begin{aligned} &z^{\ell-1} \cdot g + \mathcal{O}(z^{\ell-2})_{z \to \infty} = g \cdot (z^{\ell-1} + \mathcal{O}(z^{-1})_{z \to \infty}) = (a, b) \cdot g \cdot (z^{-m} \cdot g_a, z^{-n} \cdot g_b)^T \\ &= (a^{(k)}, b^{(k)}) \cdot adj\, U^{(k)} \cdot (z^{-m} \cdot g_a, z^{-n} \cdot g_b)^T + (\alpha^{(k)}, \beta^{(k)}) \cdot adj\, U^{(k)} \cdot (z^{-m} \cdot g_a, z^{-n} \cdot g_b)^T. \end{aligned}$$

With the aid of (14) one verifies that $(a^{(k)}, b^{(k)}) \cdot adj\, U^{(k)} \cdot (z^{-m} \cdot g_a, z^{-n} \cdot g_b)^T = \mathcal{O}(z^{\ell-1})_{z \to \infty}$. Therefore,

$$z^{\ell-1} \cdot (g - g(0)) = \Pi_{\ell, 2\cdot\ell-1}(\alpha^{(k)}, \beta^{(k)}) \cdot adj\, U^{(k)} \cdot (z^{-m} \cdot g_a, z^{-n} \cdot g_b)^T$$

which allows us to deduce the estimate

$$||g - g(0)|| \leq ||(\alpha^{(k)}, \beta^{(k)})|| \cdot ||adj\, U^{(k)}|| \cdot ||(g_a, g_b)^T|| \leq 4 \cdot ||(\alpha^{(k)}, \beta^{(k)})|| \cdot \rho_{n-m+2k}(a, b). \tag{17}$$

Using the fact that $U^{(k)}$ is well–behaved, we obtain $||g - g(0)|| \leq |g(0)|/2$, as required for a proof of assertion (b).

Proof of part (c): Following the arguments of the proof of [3, Corollary 4.4], for establishing (c) it is sufficient to show that

$$\begin{aligned} &|z^{1-m-n} \cdot (a(z), b(z)) \cdot \underline{U}^{(m)}(z)| \geq 1/2 \quad \text{for all } |z| \geq 1, \\ &|(a(z), b(z)) \cdot U^{(m)}(z) \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix}| \geq |a^{(m)}(0)|/2 \quad \text{for all } |z| \leq 1. \end{aligned}$$

For a proof of the first relation, notice that for all $|z| \geq 1$ there holds by assumption on $\underline{U}^{(m)}$

$$|z^{1-m-n} \cdot (a(z), b(z)) \cdot \underline{U}^{(m)}(z) - 1| \leq |z^{1-m-n} \cdot \gamma^{(m)}(z)| \leq ||\gamma^{(m)}|| \leq 1/2.$$

We now turn to the second relation, denote the entries of $U^{(m)}$ as in (8), and choose $|z| \leq 1$. Then

$$|(a(z), b(z)) \cdot U^{(m)}(z) \cdot \binom{1}{0}| = |a(z) \cdot v_1(z) + b(z) \cdot u_1(z)| = |a^{(m)} + \alpha^{(m)}(z)| \geq |a^{(m)}| - ||\alpha^{(m)}||,$$

and it just remains to show that $|a^{(m)}| = |a^{(m)}(0)| \geq 2 \cdot ||\alpha^{(m)}||$. In fact, with $g(z) = \det U^{(m)}(z)$ we have by assumption on $U^{(m)}$

$$|a^{(m)}| \geq ||(a^{(m)}, 0) \cdot adj\, U^{(m)}|| \geq ||g \cdot (a, b)|| - ||(\alpha^{(m)}, \beta^{(m)}) \cdot adj\, U^{(m)}|| \geq \frac{||g|| \cdot ||(a, b)||}{\rho_{m+n}(a, b)}$$

$$-2 \cdot ||(\alpha^{(m)}, \beta^{(m)})|| \geq \frac{|g(0)| \cdot ||(a, b)||}{\rho_{m+n}(a, b)} - \frac{|g(0)|}{4 \cdot \rho_{m+n}(a, b)} \geq \frac{|g(0)|}{4 \cdot \rho_{m+n}(a, b)} \geq 2 \cdot ||(\alpha^{(m)}, \beta^{(m)})||,$$

as required for proving part (c). □

Proof of Theorem 4.2: As in [7, Theorem 6.5], we define for $i \in \mathcal{A} \cup \{0\}$, $j \geq i$, the propagation factors

$$U^{(i,i)} = I_2, \quad U^{(i,j)} = \Big(\prod_{\ell \in \mathcal{A}_j, \ell > i} U^{(\ell - s_\ell, \ell)} \Big) \cdot U^{(j - s_j, j)}.$$

Then, in view of $(\alpha^{(0)}, \beta^{(0)}, \gamma^{(0)}) = 0$, we obtain

$$(\alpha^{(k)}, \beta^{(k)}) = \sum_{j \in A_k \cup \{k\}} [(\alpha^{(j)}, \beta^{(j)}) - (\alpha^{(j-s_j)}, \beta^{(j-s_j)}) \cdot U^{(j-s_\ell, j)}] \cdot U^{(j,k)}, \tag{18}$$

$$\gamma^{(k)} = \sum_{j \in A_k \cup \{k\}} [\gamma^{(j)} - \gamma^{(j-s_j)} \cdot z^{s_j}] \cdot z^{k-j}. \tag{19}$$

Consider first the term in the square brackets for fixed j. Let $j' := j - s_j$. The local errors $(\alpha^{(j)}, \beta^{(j)}) - (\alpha^{(j')}, \beta^{(j')}) \cdot U^{(j',j)}$ and $\gamma^{(j)} - \gamma^{(j')} \cdot z^{s_j}$ consist of three parts, namely (i) the error in the floating point operations (9) required for building up the systems (13), (ii) the residual error of the "small" linear systems (13), and (iii) the error in the floating point operations (10) required for building up the quantities $U^{(j)}$ and $\underline{U}^{(j)}$. With regard to (iii) one shows that[9]

$$||U^{(j)} - U^{(j')} \cdot U^{(j',j)}|| \leq 1.01 \cdot \mu \cdot (2s_j + 1) \cdot ||U^{(j',j)}||, \tag{20}$$

$$||\underline{U}^{(j)} - z^{s_j} \cdot \underline{U}^{(j')} - U^{(j')} \cdot \underline{U}^{(j,j')}|| \leq 1.01 \cdot (2s_j + 2) \cdot \mu \cdot (||\underline{U}^{(j')}|| + ||\underline{U}^{(j',j)}||). \tag{21}$$

Also, bounds for the other local errors are well–known, we may summarize these findings in the form[10]

$$||(\alpha^{(j)}, \beta^{(j)}) - (\alpha^{(j')}, \beta^{(j')}) \cdot U^{(j',j)}|| \leq C(j', s_j) \cdot \mu \cdot ||U^{(j',j)}||. \tag{22}$$

Similarly, for the residual $\gamma^{(j)}$, one establishes an estimate of the form

$$||\gamma^{(j)} - \gamma^{(j')} \cdot z^{s_j}|| \leq C(j', s_j) \cdot \mu \cdot (||\underline{U}^{(j')}|| + ||\underline{U}^{(j',j)}||). \tag{23}$$

[9] For estimating the error in floating point operations between matrix polynomials we use well–known results on the error in computing a floating point scalar product (cf. e.g., [14, Eqn.(2.4.1)], or [7, Lemma 2.7 and Lemma 2.8]). Here we assume tacitly $2\mu \cdot m \leq 0.01$.

[10] Up to a term $\mathcal{O}(\mu^2)$, this estimate was established in [7, Lemmas 6.1–6.3]. Later it was shown in [1, Section 4] that the term $\mathcal{O}(\mu^2)$ may be dropped provided that some power of m does not exceed the reciprocal of the machine precision. Also, the constant in (22) contains the Gaussian growth factor of the matrix $M_{s_j}^{(j')}$ which at least theoretically can be as large as $2^{2s_j - 1}$. Following [1, Lemma 4], this growth factor may be dropped if QR–decomposition techniques are used for solving the linear systems (13).

We want to replace the term $||\underline{U}^{(j',j)}||$ on the right hand side of (23). Consider the case $j' > 0$. Then $U^{(j')}$ is well–behaved. Consequently, the matrix $V^{(j')}$ introduced in Theorem 4.3(b) satisfies $1.01 \cdot (2s_j + 2) \cdot \mu \cdot ||V^{(j')}|| < 1/2$ by assumption on ϵ. Multiplying the expression in the norm on the left hand side of (21) by $V^{(j')}$ leads to the estimate

$$||\underline{U}^{(j',j)}|| \leq \frac{C(1)}{|\det U^{(j')}(0)|} \cdot (||\underline{U}^{(j')}|| + ||\underline{U}^{(j)}||), \tag{24}$$

which is also trivially true in the case $j' = 0$. A combination of (19), (23), and (24) yields the claimed bound for $||\gamma^{(k)}||$.

In order to establish a bound for $||U^{(i,j)}||$, $i \in \mathcal{A}_k \cup \{0\}$ and $i \leq j \leq k$, one first shows similar to [1, Lemma 5] that

$$||U^{(j)} - U^{(i)} \cdot U^{(i,j)}|| \leq \frac{\epsilon}{8} \cdot ||U^{(j',j)}|| + 4\mu \cdot \sum_{\ell \in \mathcal{A}_j, \ell > i} s_\ell \cdot ||U^{(\ell - s_\ell, \ell)}|| \cdot ||U^{(\ell,j)}||.$$

(here again the conditions on ϵ are essential). Then we multiply the term in the norm on the left by $V^{(i)}$, and show by recurrence on $j - i$ that

$$||U^{(j',j)}|| \leq 8/|\det U^{(j')}(0)|, \quad ||U^{(i,j)}|| \leq C(1)/|\det U^{(i)}(0)|. \tag{25}$$

As a consequence, the second part of the Theorem follows by combining (18), (22), and (25). □

Remark 4.4 *We may still give another interpretation of the look–ahead strategy of algorithm COPRIME. Suppose that $U^{(k)}$ is well–behaved. Using Theorem 4.3(b) it is not difficult to show the inequalities*

$$\begin{aligned} &\min_{|z| \leq 1} ||(a^{(k)}(z), b^{(k)}(z))|| - ||(\alpha^{(k)}, \beta^{(k)})|| \leq \min_{|z| \leq 1} ||(a(z), b(z))|| \\ &\leq [\min_{|z| \leq 1} ||(a^{(k)}(z), b^{(k)}(z))|| + ||(\alpha^{(k)}, \beta^{(k)})||] \cdot \frac{4}{|\det U^{(k)}(0)|}. \end{aligned}$$

Provided that the closest common root of both pairs (a, b) and $(a^{(k)}, b^{(k)})$ lie in the unit disk, we may conclude from [3, Theorem 4.1] and the above inequalities that the ratio of $\epsilon(a, b)$ and $\epsilon(a^{(k)}, b^{(k)})$ lies approximately between 1 and $4/|\det U^{(k)}(0)|$. In other words, the ideal $< a^{(k)}, b^{(k)} >$ provides as much information with regard to coprimeness as the original one if $|\det U^{(k)}(0)|$ is not too small. □

5 Numerical Experiments

The algorithm COPRIME of Table 2 was implemented in Matlab and experiments were run in order to verify the predicted behavior. In this section we report on the results of some of these experiments. In all cases the experiments provide support for our theoretical results.

In our examples, as input data we took polynomials a and b with $m = \deg a > n = \deg b$, and with randomly chosen coefficients. Also, each time we scaled (a, b) by a power of two in order to get $1/2 \leq ||(a, b)|| \leq 1$. The small systems of equations (13) encountered in algorithm COPRIME were solved using the LINPACK subroutines ZGEDI and ZGEFA. Also, the data of the examples below have been chosen such that algorithm COPRIME terminates successfully with $m \in \mathcal{A}$, i.e., the final basis $U^{(m)}$ together with its associated vector $\underline{U}^{(m)}$ have been accepted.

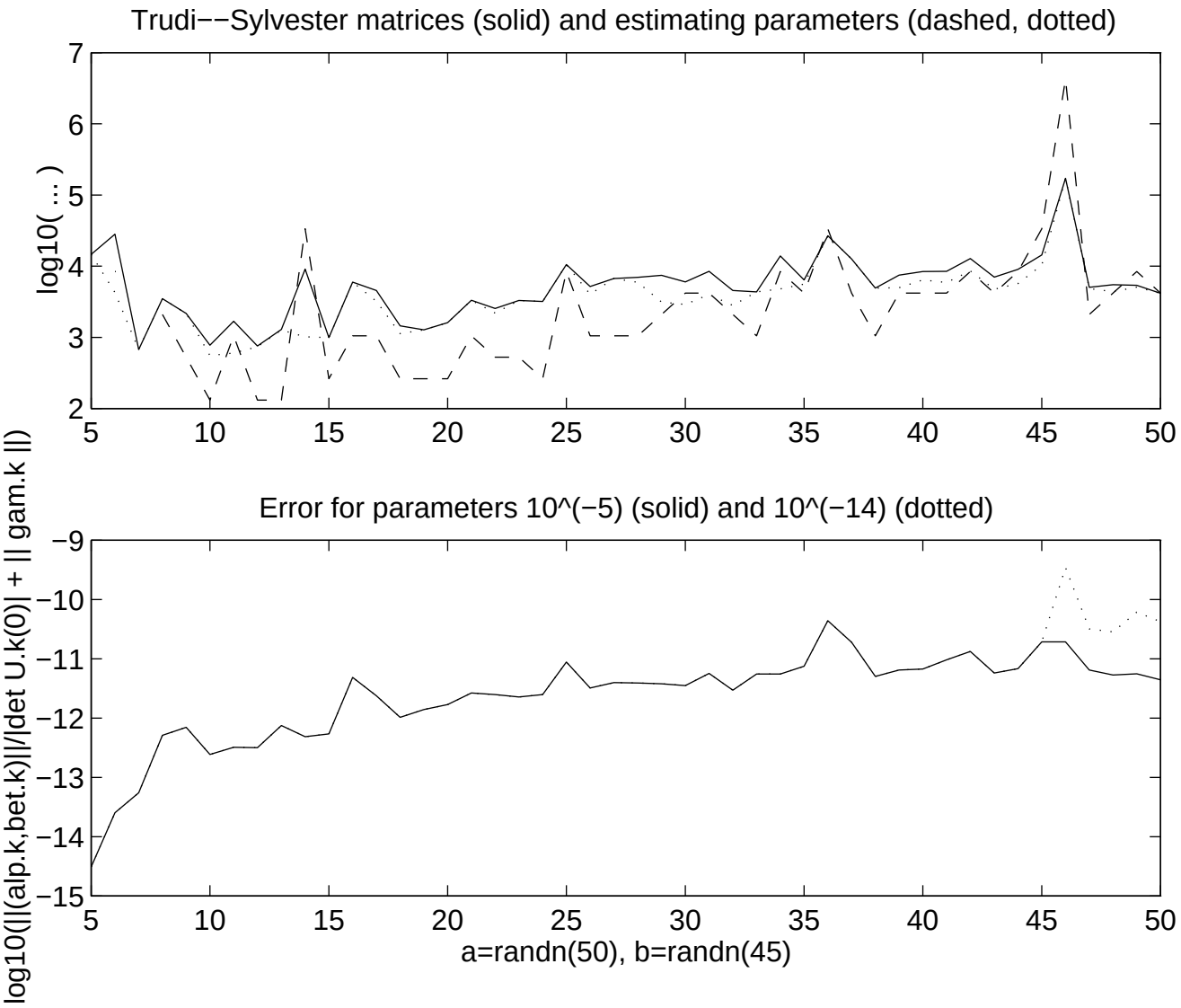

Figure 1: The plot for Example 7.1.

In order to check the effectiveness of our look–ahead criterion, we plotted in a first box the 1–norm of the inverse of the Trudi submatrix $S_k(a,b)$, introduced in Appendix A, as a function of the iteration index $k = m-n, m-n+1, \ldots, m$ (solid line, these quantities have been computed by Matlab using the LINPACK subroutines ZGEDI and ZGEFA). In all examples one clearly sees the link between this quantity and its estimators $||\underline{U}^{(k)}||$ (dotted line) and $1/|\det U^{(k)}(0)|$ (dashed line) computed by our algorithm COPRIME. In fact, in the examples the quantity $||\underline{U}^{(k)}||$ is always smaller but (up to some exceptional steps) quite close to $||S_k(a,b)^{-1}||_1$. In addition, the other estimator $1/|\det U^{(k)}(0)|$ follows quite closely the graph of $||S_k(a,b)^{-1}||_1$, with the distance being connected to the quantity[11] $\rho_{m-n+2k}(a,b) \leq \rho_{m+n}(a,b)$, confirming the results of Remark 3.1 and Theorem A.1.

In a second box, we plotted the residual errors

$$\delta_k(\epsilon) := ||(\alpha^{(k)}, \beta^{(k)})||/|\det U^{(k)}(0)| + ||\gamma^{(k)}||$$

[11] Instead of computing the quantity $\rho_\ell(a,b)$ by some optimization method, we give for each example the related quantity $\widetilde{\rho}_\ell(a,b)$ obtained by taking the 2–norm instead of the 1–norm of coefficient vectors. Here we only need to solve a least square problem.

as a function of the iteration index $k = m-n, m-n+1, \ldots, m$, $k \in \mathcal{A}$, for two different threshold parameters ϵ (in the case $k \notin \mathcal{A}$ we plotted the value of the last accepted basis, and so a horizontal section in this plot corresponds to a jump over unstable subproblems). The smaller threshold parameter (solid line) equals 10^{-5} or 10^{-6} which has to be compared with the machine precision $2^{-52} \approx 2.2 \cdot 10^{-16}$. In fact here we never encountered subproblems with "large" $||S_k(a,b)^{-1}||_1$. We have chosen the larger threshold parameter (dotted line) such that there are no look–ahead steps (as in the usual Euclidean algorithm). Notice that the jumps for a given threshold parameter may approximately be predicted by drawing a horizontal line through the corresponding threshold value in box 1.

Example 5.1 *In the first experiment we tested for whether the jump over one slightly ill–conditioned subproblem may lead to a significant increase in precision. Here we have chosen $m = 50, n = 45$, and the Trudi submatrices have a 1–condition number less than $5 \cdot 10^4$, up to the step $k = 46$ where we have approximately the value $2 \cdot 10^5$. Algorithm COPRIME computed the quantities*

$$|a^{(m)}(0)|/||U^{(m)} \cdot (1,0)^T|| \approx 3.2 \cdot 10^{-4}, \quad 1/||\underline{U}^{(m)}|| \approx 2.4 \cdot 10^{-4}.$$

For the threshold parameter $\epsilon = 10^{-5}$ we just obtain the jump over the unstable subproblem $k = 46$, as predicted by our theory (see Figure 1). In addition, the two choices of parameters allowed us to gain approximately one digit since $\delta_m(10^{-5})/\delta_m(10^{-14}) \approx 1/10$.

Notice that the distance between the solid and the dashed polygon of the first box in Figure 1 is of the same order of magnitude as the quantity $\widetilde{\rho}_{m+n}(a,b) \approx 29$, as predicted by Theorem A.1. However, only both estimators together yield a reliable estimate of $||S_k(a,b)^{-1}||_1$ (see iteration $k = 14$).

Also, for this example, the additional assumption for the error analysis given in [7, 9] is not verified, namely, for the coefficients in the power series expansion at infinity of $a(z)^{-1}$ we have $\rho_{m+n}(a,0) \approx 9.1 \cdot 10^6$, which has to be compared to $\widetilde{\rho}_{m+n}(a,b) \approx 29$. If one reverses the order of coefficients, things become even worse, since $\widetilde{\rho}_{m+n}(\underline{a},\underline{b}) \approx 28$, and $\rho_{m+n}(\underline{a},0) \approx 5.3 \cdot 10^{31}$. □

Example 5.2 *In order to refine the choice of the threshold parameter, we have run a "bigger" example with $m = 150$, $n = 149$. Here by COPRIME we obtain*

$$|a^{(m)}(0)|/||U^{(m)} \cdot (1,0)^T|| \approx 2.0 \cdot 10^{-5}, \quad 1/||\underline{U}^{(m)}|| \approx 1.7 \cdot 10^{-5}.$$

The condition numbers of the Trudi submatrices vary between 10^4 and $5 \cdot 10^6$, and $\widetilde{\rho}_{m+n}(a,b) \approx 250$. In fact, as seen from Figure 2, we gain again about two digits for the residual δ_m. There are no jumps for the threshold 10^{-14}, whereas for $\epsilon = 10^{-6}$ we do not accept the bases with indices 35, 49, 63, 66, 82, 93, 96-97, 100, 108, 110-111, and 117. As displayed in Figure 2, these jumps occur partly because $|\det U^{(k)}(0)|$ was too small, and partly because $||\underline{U}^{(k)}||$ was too large.

The largest jump for $\epsilon = 10^{-6}$ is of size 2. However, for the threshold 10^{-5} we would have one jump of size 18 (namely the indices 104-121) and three other large jumps (namely 87-93, 96-102, 130-135), though we only gain for the corresponding δ_m a factor $1/2$. □

Example 5.3 *Finally we have chosen polynomials a of degree $m = 50$ with 11 non–zero coefficients, and b of degree $n = 49$ with 10 non–zero coefficients. Here COPRIME gives*

$$|a^{(m)}(0)|/||U^{(m)} \cdot (1,0)^T|| \approx 9.1 \cdot 10^{-4}, \quad 1/||\underline{U}^{(m)}|| \approx 6.9 \cdot 10^{-4}.$$

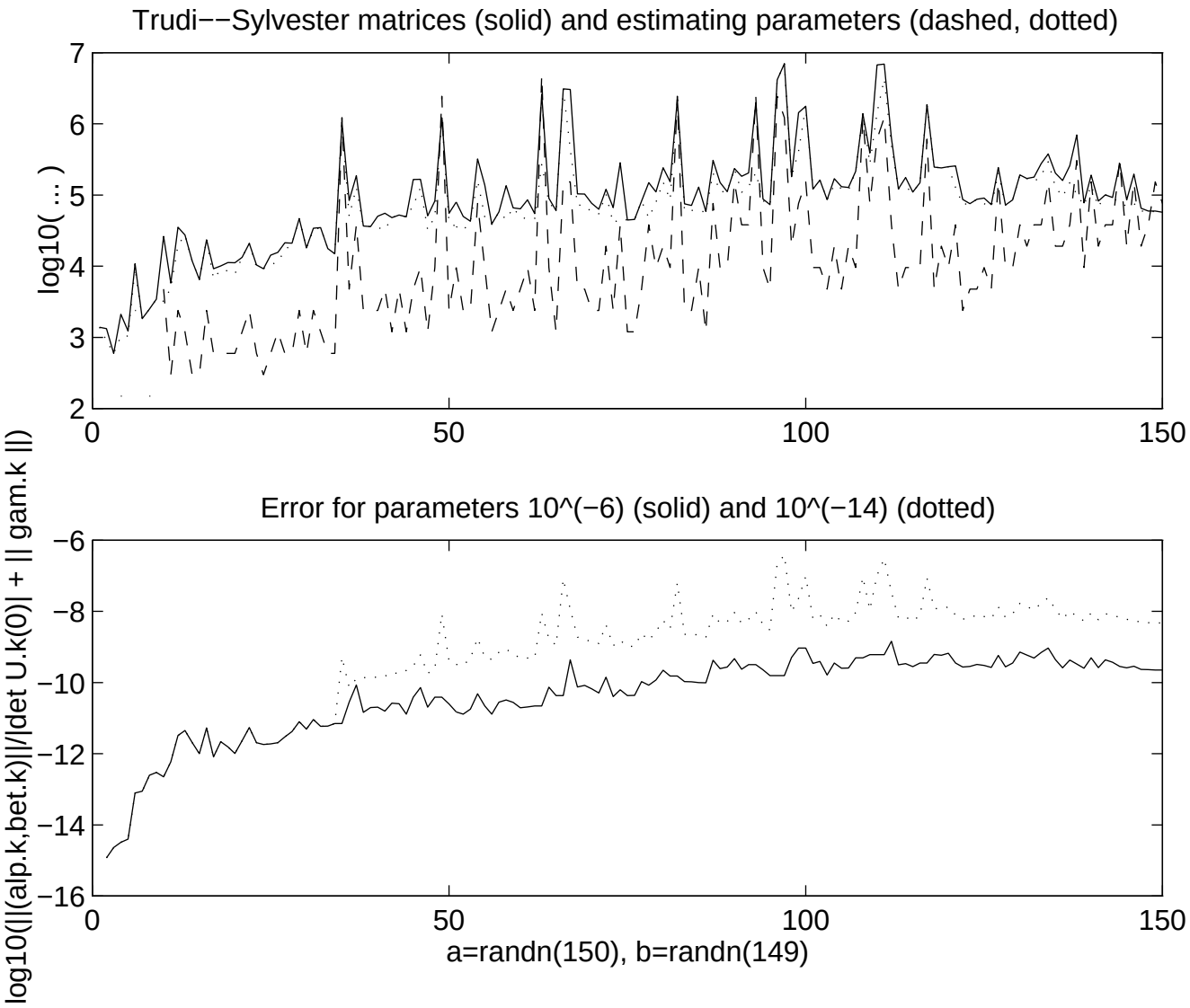

Figure 2: The plot for Example 7.2.

The condition number of the Trudi submatrices are in general smaller than 10^4. *However, for* $k = 8$ *there is a peak of approximately* 10^8, *and there are also some other peaks of order* 10^5 *to* 10^6, *as seen in the first box of Figure 3. Also,* $\widetilde{\rho}_{m+n}(a,b) \approx 28$, *whereas* $\rho_{m+n}(a,0) \approx 3.3 \cdot 10^6$.

Notice that $\delta_m(10^{-13}) \approx 3 \cdot 10^{-4}$, *of order of* $\epsilon(a,b)$, *and thus the output of COPRIME for this choice of threshold parameter is of no significance. In terms of the notation introduced in Section 4, the quantities* $U^{(m)}$, $\underline{U}^{(m)}$, *are accepted, but not well–behaved for the threshold parameter* $\epsilon = 10^{-13}$.

We have chosen several threshold parameters in order to show the dependence between accepted bases and size of the residual, the results are displayed in Table 3. Clearly, each time we accept more bases we get results with poorer residual. □

Implementations of COPRIME in Matlab and MAPLE, along with the example polynomials can be obtained via email from the authors.

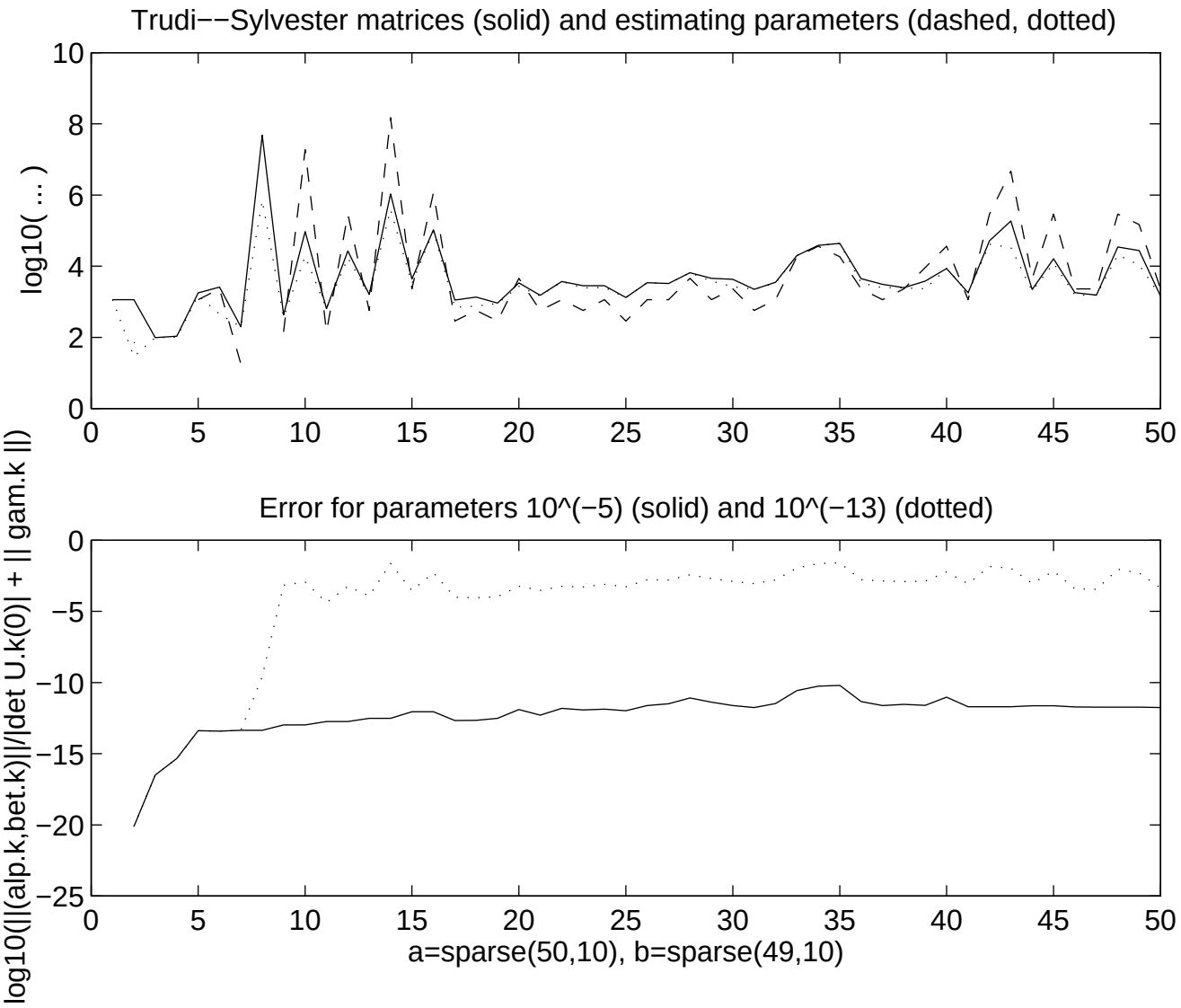

Figure 3: The plot for Example 7.3.

6 Conclusion and future research

We have considered the problem of determining when two univariate polynomials are numerically relatively prime, that is, they remain coprime even after a suitable perturbation of their coefficients. In [3], a parameter has been given that improves previously existing measures based on singular values of Sylvester matrices. We propose in the present paper to compute this coprimeness parameter in a efficient and numerically stable way using the algorithm COPRIME, an extension of the Cabay-Meleshko algorithm for Padé approximation.

Our method can be considered as a stabilized version of the Euclidean algorithm where using look–ahead techniques one steps over unstable remainders. We have illustrated with help of examples, theoretical results and numerical experiments that ill–conditioned subproblems are detected in a reliable manner by COPRIME, whereas classical implementations of Euclid's algorithm may suffer in finite precision arithmetic from numerical instabilities.

$\log_{10}(\epsilon)$	indices of skipped bases	$\log_{10}(\delta_m(\epsilon))$
-5	$8, 10, 12, 14, 16, 42-43, 45, 48-49$	-11.75
-6	$8, 10, 14, 16, 43$	-10.52
-7	$8, 10, 14$	-9.62
-8	$8, 14$	-8.33
$-9, -10, -11, -12$	8	-7.27
≤ -13	*no*	-3.34

Table 3: Indices of skipped bases in Example 5.3 for different threshold parameters.

If the algorithm COPRIME fails to give a lower bound for $\epsilon(a, b)$, then the cofactors in the diophantine equations are too large in norm to give useful information. Further research is desirable to give an improved lower bound for $\epsilon(a, b)$ which can also be computed in a fast, numerically stable way. Let us however mention that in general the degrees of the remainders of the final accepted basis will be relatively small in comparison to the degrees of the input polynomials. Thus here it may be appropriate to employ optimization techniques [11, 15] for these remainders in order to obtain further information on $\epsilon(a, b)$ by using inequalities as stated for example in Remark 4.4.

In case of "small" $\epsilon(a, b)$ our algorithm will not accept the final unimodular reduction of order m – here we are faced with the problem of computing a non–trivial numerical GCD. Let us recall the following conjecture of Cabay and Meleshko [16] (slightly reformulated): The numerical defect of the Sylvester matrix (and hence the degree of the numerical GCD) is determined by m minus the order $\widetilde{m}$ of the last accepted scaled UR of algorithm COPRIME. In addition, the numerical GCD is determined by the last successful unimodular reduction.

Here the key observation seems to be that quite often the quantity $b^{(\widetilde{m})}$ is small in norm, and thus $a^{(\widetilde{m})}$ is a "good" candidate for a numerical GCD. In fact, in appendix B we specify upper bounds for $||b^{(\widetilde{m})}||$ for insuring that $a^{(\widetilde{m})}$ is the Quasi–GCD of Schönhage [18, Task 1.4] or an η–GCD in the sense of [11, 12, 15]. However, an exact analysis for the computation of numerical GCD's by the algorithm COPRIME remains a subject for further research.

References

[1] B. Beckermann, The stable computation of formal orthogonal polynomials, *Numerical Algorithms* **11** (1996) 1-23.

[2] B. Beckermann, The Condition Number of real Vandermonde, Krylov and positive definite Hankel matrices, Publication ANO 380, Université de Lille (1997).

[3] B. Beckermann & G. Labahn, When are two numerical polynomials relatively prime? *Journal of Symbolic Computation* (this issue)

[4] D. Bini & V. Pan, *Polynomial and matrix computations* (Birkhäuser, 1994)

[5] A.W. Bojanczyk, R.P. Brent & F.R. de Hoog, Stability analysis of a general Toeplitz systems solver, *Numerical Algorithms* **10** (1995) 225-244.

[6] J. Bunch, The weak and strong stability of algorithms in numerical linear algebra, *Lin. Alg. Appl.* **88/89** (1987) 49-66.

[7] S. Cabay and R. Meleshko, A weakly stable Algorithm for Padé Approximants and the Inversion of Hankel matrices, *SIAM J. Matrix Analysis and Applications* **14** (1993) 735-765.

[8] S. Cabay, A. R. Jones and G. Labahn, Computation of Numerical Padé-Hermite and Simultaneous Padé Systems I: Near inversion of generalized Sylvester matrices, *SIAM J. Matrix Analysis and Applications* **17** (1996) 237-267.

[9] S. Cabay, A. R. Jones and G. Labahn, Computation of Numerical Padé-Hermite and Simultaneous Padé Systems II: A Weakly Stable Algorithm, *SIAM J. Matrix Analysis and Applications* **17** (1996) 268-297.

[10] S. Cabay, A. R. Jones and G. Labahn, Experiments with a Weakly Stable Algorithm for Computing Padé-Hermite and Simultaneous Padé Approximants, *ACM Trans. of Mathematical Software (TOMS)* **23**(1) (1997) 91-110.

[11] R.M. Corless, P.M. Gianni, B.M. Trager & S.M. Watt, The Singular Value Decomposition for Polynomial Systems, Proceedings ISSAC '95, ACM Press (1995) 195-207.

[12] I.Z. Emiris, A. Galligo & H. Lombardi, Certified approximate univariate GCDs, *J. Pure and Applied Algebra* **117 & 118** (1997) 229-251.

[13] K.O. Geddes, S.R. Czapor and G. Labahn, *Algorithms for Computer Algebra* (Kluwer, Boston, MA, 1992)

[14] G.H. Golub, C.F. Van Loan, Matrix Computations, Second Edition, Johns Hopkins University Press, Baltimore, London (1993).

[15] N. Karmarkar & Y.N. Lakshman, Approximate polynomial greatest common divisors and nearest singular polynomials, Proceedings ISSAC '96, ACM Press (1996) 35–39.

[16] R. Meleshko and S. Cabay, "Experience with a Fast Stable Sylvester Solver", Fourth SIAM conference in Applied Linear Algebra, Minneapolis, Minnesota, Sept 11-14, 1991.

[17] M.-T. Noda & T. Sasaki, Approximate GCD and its applications to ill–conditioned algebraic equations, *J.CAM* **38** (1991) 335-351.

[18] A. Schönhage, Quasi–GCD Computations, *J. Complexity* **1**(1985) 118-137.

[19] M. Van Barel and A. Bultheel, A look–ahead algorithm for the solution of block Toeplitz systems, Report TW 224, Katholieke Universiteit Leuven, (1995). To appear in *Linear Algebra and its Applications.*

A Condition number estimates for Trudi matrices

A unimodular reduction of order k, $m-n < k \leq m$, together with its associated vector may be obtained by solving systems of equations where the (common) matrix of coefficients is the kth *Trudi matrix* with stripes of size $n-m+k$ and k

$$S_k(a,b) := \begin{bmatrix} a_{m-k} & a_{m-k-1} & \cdots & a_{2m-n-2k+1} & b_{m-k} & b_{m-k-1} & \cdots & b_{m-2k+1} \\ \vdots & & & \vdots & \vdots & & & \vdots \\ a_m & a_{m-1} & \cdots & a_{m-n-k+1} & b_n & b_{n-1} & \cdots & b_{n-k+1} \\ 0 & a_m & & a_{m-n-k+2} & 0 & b_n & & b_{n-k+2} \\ \vdots & \ddots & \ddots & \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & a_m & 0 & \cdots & 0 & b_n \end{bmatrix}.$$

In other words, we drop from the Sylvester matrix $S(a,b)$ the first $(m-k)$ columns of each column block, and the first $2m-2k$ rows, in particular $S_m(a,b) = S(a,b)$. These matrices are useful for determining the degree of a non–trivial GCD of a and b (see, e.g., [13]). The condition number of the kth Trudi matrix is thus closely related to the numerical condition of the problem of determining a UR $U^{(m)}$ of order k together with its associated vector $\underline{U}^{(k)}$. Conversely, it is possible to estimate the condition number of Trudi matrices in terms of these quantities.

Theorem A.1 *Let $U^{(k)}$ be a scaled UR of order k and $\underline{U}^{(k)}$ be an associated vector of order k. Furthermore, define $\rho_\ell(a,b)$ as in (15) (which at least in the examples of Section 5 is not much larger than one) and $\sigma := ||S_k(a,b)|| \cdot ||S_k(a,b)^{-1}||$. Then*

$$||S_k(a,b)^{-1}|| \leq 2\rho_{n-m+2k}(a,b)/|\det U^{(k)}(0)|, \tag{26}$$

$$1 \leq 1/|\det U^{(k)}(0)| \leq 4\sigma + 4\sigma^2, \tag{27}$$

$$1 \leq 2||(a,b)||/||(lc(a),lc(b))^T|| \leq \rho_{n-m+2k}(a,b) \leq ||\underline{U}^{(k)}|| \leq ||S_k(a,b)^{-1}||. \tag{28}$$

Proof: Estimate (26) with $\rho_{n-m+2k}(a,b)$ being replaced by $\rho_{n-m+2k}(a,0)$ has been established[12] in [8, Theorem 4 and Corollary 6]. For our refinement (26) one just has to modify slightly the proof of [8, Theorem 4] along the lines of the proofs of [3, Corollaries 3.2 and 3.3]. Part (27) follows from [10, Theorem in Appendix A]. Finally, for showing (28) notice that $2||(a,b)||/||(lc(a),lc(b))^T|| = \rho_1(a,b)$, and that from $\underline{U}^{(k)}$ we may obtain a candidate for the minimum in $\rho_{m-n+2k}(a,b)$. The remaining last inequality follows from the observation that the coefficient vector of $\underline{U}^{(k)}$ is the last column of $S_k(a,b)^{-1}$ (and, similarly, the first column of $U^{(k)}/lc(a^{(k)})$ is related to the first column of $S_k(a,b)^{-1}$). □

Following [9, 10], it is possible to restate a slightly weaker form of Theorem A.1 for well–behaved numerical scaled UR's and associated vectors. From Theorem A.1 we may conclude that, roughly, $||S_k(a,b)^{-1}||$ lies between $1/|\det U^{(k)}(0)|$ and its square root. However, in numerical experiments one observes that the quantities $1/|\det U^{(k)}(0)|$ and $||\underline{U}^{(k)}||$ are usually quite good estimators[13] for $||S_k(a,b)^{-1}||$ even for $k \notin \mathcal{A}$, see Section 5 and [10].

From the second inequality of (28) we see that — though unlikely — it may happen that the quantity $\rho_\ell(a,b)$ is large, e.g., if the zeros of both a and b are far from the unit disk. Our algorithm will detect such cases since the norm of the associated vectors will be large.

B Numerical GCD Computations with the algorithm COPRIME

The aim of this appendix is to show that in some cases our algorithm COPRIME even furnishes a numerical GCD in a numerically stable and efficient manner. We start by recalling different concepts of numerical GCD's introduced in [11, 12, 15, 18]: Let $\eta > 0$ be some parameter (usually larger than ϵ). Following Schönhage [18, Task 1.4], a polynomial d is called a *Quasi–GCD* with precision η if there exist polynomials u_1, v_1, u_2, v_2 satisfying

$$||(a,b) - d \cdot (u_3, v_3)|| < \eta, \quad ||av_1 + bu_1 - d|| < \eta \cdot ||d||, \quad \deg u_1 < m, \quad \deg v_1 < n. \tag{29}$$

Recently, another approach was discussed by several authors, see [11, Section 2.4], [15, Introduction], [12, Definition 1]: one first defines the degree of an η–GCD as being the largest integer j such that there exist polynomials $\tilde{a}, \tilde{b}$ of degree $\leq m$ and $\leq n$, respectively, with GCD having degree j, and verifying $||(a,b) - (\tilde{a},\tilde{b})|| \leq \eta$. Then the GCD of such a pair $(\tilde{a},\tilde{b})$ will be called an η–GCD.

[12] Recall that $\rho_{n-m+2k}(a,0)$ is obtained by determining the first coefficients of the expansion of $1/a$ around infinity. It is reported in [10, Observation 2] that this possibly quite large constant in numerical experiments does not seem to appear in upper bounds for $||S_k(a,b)^{-1}||$.

[13] From the example in Remark 3.1 we see that the use of only the estimator $1/|\det U(0)|$ may lead to a significant overestimation of $||S_k(a,b)^{-1}||$.

The exact relationships between these two and other well–known concepts of a numerical GCD seem to be still unclear (cf., e.g., the discussions in [11, Section 2.3] and [12, Section 5] for drawbacks of the individual approaches). Notice also that there are no non–trivial η–GCD's for $\eta < \epsilon(a,b)$. We propose the following

Theorem B.1 *Let $U^{(k)}$ be a well–behaved numerical scaled UR computed by the algorithm COPRIME, with corresponding remainders $(a^{(k)}, b^{(k)})$, and residuals $(\alpha^{(k)}, \beta^{(k)})$.*

(a) *If the norm of the remainder $b^{(k)}$ is so small that*

$$||b^{(k)}|| + ||(\alpha^{(k)}, \beta^{(k)})|| < \eta \cdot |\det U^{(k)}(0)|/12$$

with $\eta \in (0, 1/6)$ then $a^{(k)}$ is a Quasi–GCD with precision η in the sense of Schönhage.

(b) *If the norm of the remainder $b^{(k)}$ is so small that*

$$2 \cdot ||b^{(k)}|| + (2 + 4 \cdot \rho_{n-m+2k}(a,b)) \cdot ||(\alpha^{(k)}, \beta^{(k)})|| \leq \eta \cdot |\det U^{(k)}(0)|$$

then the degree of an η–GCD is at least equal to $m-k$.

(c) *If $|\det U^{(k)}(0)| > 4 \cdot \rho_{n-m+2k}(a,b) \cdot \eta$ then the degree of an η–GCD will be not larger than $m-k$. If in addition the inequality of part (b) holds then $a^{(k)}$ is also an η–GCD.*

Proof: **(a):** We first recall from Definition 4.1 that $(a,b) \cdot U^{(k)} = (a^{(k)}, b^{(k)}) + (\alpha^{(k)}, \beta^{(k)})$. In the proof of Theorem 4.3(b) (see (17)) we have shown that for $g(z) := \det U^{(k)}(z)$ there holds $||g(z) - g(0)|| \leq |g(0)|/2$. With help of (16) we may find a polynomial h with $||h \cdot g - 1|| \leq 2\eta/3$, and $||h|| \leq 2/|g(0)|$. Defining $V := h \cdot adj\, U^{(k)}$ we obtain

$$||V|| \leq ||h|| \cdot ||adj\, U^{(k)}|| \leq \frac{4}{|g(0)|}, \quad ||U^{(k)} \cdot V - I|| = ||h \cdot g - 1|| \leq \frac{2\eta}{3}.$$

It follows that

$$||(a,b) - (a^{(k)}, 0) \cdot V|| \leq ||(a,b)|| \cdot ||U^{(k)} \cdot V - I|| + (||b^{(k)}|| + ||(\alpha^{(k)}, \beta^{(k)})||) \cdot ||V|| < \eta.$$

Consequently, the first row (u_3, v_3) of V verifies (29) with $d = a^{(k)}$. Furthermore,

$$||(a,b) \cdot U^{(k)}|| \cdot ||V|| \geq ||(a,b)|| - \frac{2\eta}{3} \cdot ||(a,b)|| \geq \frac{1}{2} - \frac{2\eta}{3},$$

and thus

$$||a^{(k)}|| \geq ||(a,b) \cdot U^{(k)}|| - (||b^{(k)}|| + ||(\alpha^{(k)}, \beta^{(k)})||) > \frac{|g(0)|}{8} \cdot (1 - 2\eta) \geq \frac{|g(0)|}{12} \geq \frac{||\alpha^{(k)}||}{\eta}.$$

Denoting the first column of $U^{(k)}$ by $(v_1, u_1)^T$, the degree requirements of (29) are valid according to (14), and $||a \cdot v + b \cdot u - a^{(k)}|| \leq ||\alpha^{(k)}|| < \eta \cdot ||a^{(k)}||$, as required for the Quasi–GCD of Schönhage.
(b): Notice that the polynomials $(\tilde{a}, \tilde{b}) = a^{(k)} \cdot (u_3, v_3)$ from part (a) will in general have a too high degree to be taken into consideration for an η–GCD. By slightly modifying the approach of part (a), let $\tilde{V} := (1/g(0)) \cdot adj\, U^{(k)}$, where as before $g = \det U^{(k)}$. Then using (17) we get

$$\begin{aligned} &||(a,b) - (a^{(k)}, 0) \cdot V|| \leq ||(a,b)|| \cdot ||U^{(k)} \cdot \tilde{V} - I|| + (||b^{(k)}|| + ||(\alpha^{(k)}, \beta^{(k)})||) \cdot ||\tilde{V}|| \\ &\leq ||(g - g(0)/g(0)|| + 2 \cdot (||b^{(k)}|| + ||(\alpha^{(k)}, \beta^{(k)})||)/|g(0)| < \eta, \end{aligned}$$

showing part (b).
(c): We will use a similar argument as in the proof of [3, Lemma 2.1]: Let $(\tilde{a}, \tilde{b})$ have a GCD of degree larger than $m-k$. Then the Trudi matrix $S_k(\tilde{a}, \tilde{b})$ is known to be singular, and thus

$$||(a-\tilde{a}, b-\tilde{b})|| \geq ||S_k(a-\tilde{a}, b-\tilde{b})|| = ||S_k(a,b) - S_k(\tilde{a}, \tilde{b})|| \geq \frac{1}{||S_k(a,b)||}.$$

It remains to be shown that the latter quantity is larger than η by assumption on $\det U^{(k)}(0)$. This inequality follows from the estimates for the norm of Trudi matrices in terms of (exact) scaled UR's as stated in (28) of Appendix A. The extension to numeric UR's is possible but quite technical – we omit the details. □

Recall that the look–ahead criterion of algorithm COPRIME is just designed to keep the quantity $||(\alpha^{(k)}, \beta^{(k)})||/|\det U^{(k)}(0)|$ small. Thus, after having found an accepted scaled UR $U^{(k)}$ by algorithm COPRIME, we may check whether the quantity $||b^{(k)}||/|\det U^{(k)}(0)|$ is smaller than a given threshold parameter η, which indicates that $a^{(k)}$ is the corresponding Quasi–GCD with precision η (and possibly also an η–GCD) of the given polynomials a and b.

Theorem B.1(b) has to be compared with [12, Algorithm 2 and Proposition 13] where also a lower bound for the degree of an η–GCD is derived using Euclid's algorithm with a particular stop criterion. However, their algorithm does take into account the problem of finite precision arithmetic.

Upper bounds for the degree of an η–GCD have been obtained in [11] and [12, Algorithm 1] using SVD decompositions of the Sylvester matrix and of suitable submatrices. The bounds obtained from algorithm COPRIME via Theorem B.1(c) will probably be weaker, however, they are obtained in a more efficient and as well numerically stable manner.