Contemporary Mathematics
Volume **556**, 2011

# The Numerical Greatest Common Divisor of Univariate Polynomials

Zhonggang Zeng

Abstract. This paper presents a regularization theory for numerical computation of polynomial greatest common divisors and a convergence analysis, along with a detailed description of a blackbox-type algorithm. The root of the ill-posedness in conventional GCD computation is identified by its geometry where polynomials form differentiable manifolds entangled in a stratification structure. With a proper regularization, the numerical GCD is proved to be strongly well-posed. Most importantly, the numerical GCD solves the problem of finding the GCD accurately using floating point arithmetic even if the data are perturbed. A sensitivity measurement, error bounds at each computing stage, and the overall convergence are established rigorously. The computing results of selected test examples show that the algorithm and software appear to be robust and accurate.

## 1. Introduction

As one of the fundamental algebraic problems with a long history, finding the greatest common divisor (GCD) of univariate polynomials is an indispensable component of many algebraic computations besides being an important problem in its own right. The classical Euclidean Algorithm has been known for centuries [**12**, p.58] and the problem is well studied in computer algebra, where algorithms are developed using exact arithmetic with exact data. These algorithms are not suitable for practical numerical computation because computing GCD is an ill-posed problem in the sense that it is infinitely sensitive to round-off error and data perturbations. A tiny error in coefficients generically degrades the GCD into a meaningless constant. The central problem of this paper is: How, and why, can we still recover the lost GCD accurately using the inexact data and floating point arithmetic?

For this purpose, we study the root of the ill-posedness by presenting the geometry of the polynomial GCD problem: The collection of polynomial pairs whose GCD's share a fixed positive degree forms a differential manifold of a positive codimension, and these manifolds entangle in a stratification structure in which a manifold is in the closure of manifolds of lower codimensions. The ill-posedness of the GCD problem lies in the dimension deficit of the GCD manifold from which a polynomial pair will pushed away by arbitrary perturbations.

1991 *Mathematics Subject Classification.* Primary 65F22, 68W30; Secondary 12D05, 13P05.
*Key words and phrases.* polynomial, greatest common divisor, ill-posed problem.
Research supported in part by NSF under Grant DMS-0715127.

Taking advantage of the geometric properties, we study the *numerical GCD* formulated by Corless, Gianni, Trager and Watt [**6**] as well as Karmarkar and Lakshman [**22**] by establishing a comprehensive regularization theory of numerical GCD. We prove that numerical GCD generalizes the traditional notion of GCD and, when the data are sufficiently accurate, the numerical GCD uniquely exists and is Lipschitz continuous, thereby making it strongly well-posed and computable using floating point arithmetic. Most importantly, the numerical GCD solves the central problem of this paper by approximating the exact GCD of the underlying polynomials hidden in data perturbation.

Building upon the thorough study on the theory of numerical GCD, we further establish a detailed analysis of the algorithm proposed by the author [**41**] for computing the numerical GCD, and prove the Numerical GCD Convergence Theorem. The algorithm is implemented using the code name UVGCD as part of the comprehensive package APATOOLS [**42**] for approximate polynomial algebra.

As mentioned above, GCD-finding is one of the basic operations in algebraic computation with a wide range of applications that include engineering problems such as graphics and modeling, robotics, computer vision, image restoration, control theory, system identification [**1, 11, 15, 21, 26, 29, 32, 34, 36**], as well as other branches of mathematics and computer science such as simplifying rational expressions, partial fraction expansions, canonical transformations, mechanical geometry theorem proving [**5, 13, 47**], hybrid rational function approximation [**18**], and decoder implementation for error-correction [**3**]. A robust GCD-finder is also crucial to root-finding studies when multiple roots are present [**9, 27, 41**]. In recent years, substantial effort has been spent on developing algorithms for computing the numerical GCD of inexact polynomials. These pioneering works include resultant-based algorithms [**6, 10, 30**], optimization strategies [**4, 22**], modifications of the Euclidean Algorithm [**2, 16, 27, 31**], root grouping [**28, 36**], QR factorization [**7, 39**], and low rank approximations [**19, 20, 37, 38**]. Several methods have been implemented as part of Maple SNAP package [**17**] that include QUASIGCD, EPSILONGCD and QRGCD. Particularly in [**6**], Corless, Gianni, Trager and Watt propose a novel, albeit unfinished, approach that includes the use of singular value decomposition to identify the GCD degree and several possible strategies for calculating the GCD factors including solving least squares problems.

In the context of polynomial root-finding, we developed a new special case algorithm for computing the GCD of a polynomial and its derivative. This algorithm is briefly described in [**41**] as an integral component of Algorithm MULTROOT [**40**] that calculates multiple roots of a polynomial with high accuracy without using multiprecision arithmetic even if the coefficients are perturbed. Due to the scope of the paper [**41**], that algorithm is narrowly featured without in-depth analysis of the problem regularity, algorithm convergence, error analysis, extensions, applications, or comprehensive testing/experiment. Our numerical GCD algorithm employs a successive Sylvester matrix updating process for identifying the maximum degree of the numerical GCD along with an initial approximation to the GCD factors. Then the Gauss-Newton iteration is applied to certify the GCD and to refine the polynomial factors via solving a regular quadratic least squares problem. Those new strategies apparently fill the main gaps in previous works and is mentioned in a recently published textbook:

> "This numerical common gcd algorithm ... appears to be the most efficient and reliable algorithm for that purpose; I have seen it too late to include it in the text." H. J. Stetter, *Numerical Polynomial Algebra* [**33**, p.223]

The software UVGCD has been tested rigorously and extensively. As sample test results shown in §11, UVGCD is substantially more robust and accurate than the existing packages. The complexity of our method is $O(n^3)$ for the combined degree $n$ of the given polynomials.

The main theorems in this paper appear to be new, including GCD Manifold Theorem, GCD Extension Theorem, Numerical GCD Regularity Theorem, Numerical GCD Approximation Theorem and Numerical GCD Convergence Theorem.

## 2. Difficulties of finding GCD in numerical computation

Computing polynomial GCD is a typical "ill-posed problem" whose numerical solutions are generally unattainable using conventional methods, even if the method is among the most celebrated in the history. The hypersensitivity of such problems can be illustrated in a simple example:

EXAMPLE 2.1. Consider a pair of polynomials

$$\begin{aligned} p(x) &= x^{10} + \tfrac{31}{3}x^9 + \tfrac{10}{3}x^8 + x + 10 \\ q(x) &= x^{10} + \tfrac{71}{7}x^9 + \tfrac{10}{7}x^8 - \tfrac{6}{7}x - \tfrac{60}{7} \end{aligned}$$

They can be factored as $(x+10)(x^9 + \frac{x^8}{3} + 1)$ and $(x+10)(x^9 + \frac{x^8}{7} - \frac{6}{7})$ respectively. There is no difficulty for a common computer algebra system like Maple to find the GCD using symbolic computation:

```
> gcd(x^10+(31/3)*x^9+(10/3)*x^8+x+10,x^10+71*x^9/7+10*x^8/7-6*x/7-60/7);
```

$$x + 10$$

However, the GCD quickly degrades from $x+10$ to a constant simply by replacing the fractional numbers in the coefficients with floating point values at the simulated hardware precision of 10 digits[1]:

```
> gcd(x^10+10.33333333*x^9+3.333333333*x^8+x+10.,
      x^10+10.14285714*x^9+1.428571429*x^8-.8571428571*x-8.571428571);
```

$$1.000000000$$

The constant 1 is, as a matter of fact, the correct GCD in exact sense from the given (perturbed) coefficients, and the nontrivial GCD $x+10$ is lost from a tiny perturbation in data. This is the ill-posed nature of GCD for being discontinuous with respect to its coefficients.

For hundreds of years, the classical Euclidean Algorithm has been *the* method for GCD finding. However, it can easily fail in numerical computation. The Euclidean Algorithm is a recursive process of polynomial division

$$f = q \cdot g + r \tag{2.1}$$

from a polynomial pair $(f, g)$ to obtain the quotient $q$ and the remainder $r$. Assume the degree of $g$ is no larger than that of $f$ without loss of generality, and initialize $f_0 = f$, $f_1 = g$. The Euclidean Algorithm

$$f_j = q_j \cdot f_{j+1} + f_{j+2},\ j = 0, 1, \ldots \tag{2.2}$$

[1]The test is carried out using Maple 12. Other versions of Maple may yield different results

generates a remainder sequence $f_2,\ f_3,\ \ldots$ that terminates at a scalar multiple of the GCD of $f$ and $g$. The Euclidean Algorithm and its modifications remain the method of choice for GCD computation in symbolic computation for exact polynomials. The following example illustrates why the Euclidean Algorithm behaves poorly in the presence of data error or round-off.

EXAMPLE 2.2. Consider polynomials appeared in Example 2.1;

$$\begin{array}{rcl} f(x) & = & (x+10)(x^9+\frac{x^8}{3}+1) \ = \ x^{10}+\frac{31}{3}x^9+\frac{10}{3}x^8+x+10, \\ g(x) & = & x+10. \end{array} \tag{2.3}$$

with their GCD equals to $g(x)$. The (exact) Euclidean Algorithm stops at one step since $f(x) = (x^9+\frac{x^8}{3}+1)\cdot g(x)+0$. However, if the coefficients of $f$ are inexact with a perturbation of a magnitude at the simulated hardware precision, say

$$\tilde{f}(x) = x^{10}+10.33333333x^9+3.333333333x^8+x+10.$$

the one-step Euclidean Algorithm involves a polynomial division

$$\begin{array}{rcl} \tilde{f}(x) & = & (x+10)(x^9+.33333333x^8+.000000033x^7-.00000033x^6+ \\ & & +.0000033x^5-.000033x^4+.00033x^3-.0033x^2+.033x+0.67)+3.3 \end{array}$$

Far from getting a zero, the remainder becomes 3.3, and the Euclidean Algorithm produces a constant GCD with a large difference from $x+10$ we are looking for, even though the data error is tiny. □

The numerical instability of the Euclidean Algorithm is inherent from polynomial division (2.1), which is equivalent to linear system for the coefficients of $q$ and $r$. Using the $f$ and $g$ in (2.3) as an example again, equation (2.1) can be written as

$$\begin{bmatrix} 1 & & & & \\ 10 & 1 & & & \\ & 10 & \ddots & & \\ & & \ddots & 1 & \\ & & & 10 & 1 \end{bmatrix} \begin{bmatrix} q_9 \\ q_8 \\ \vdots \\ q_0 \\ r_0 \end{bmatrix} = \begin{bmatrix} 1 \\ 10.33333333 \\ 3.333333333 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 10 \end{bmatrix} \tag{2.4}$$

Perturbations in coefficients of $f$ and/or $g$ are magnified by the large condition number $1.1\times 10^{10}$ of the $10\times 10$ matrix in (2.4), as shown by the nonzero remainder computed in Example 2.2.

The question is: Can we accurately compute the GCD, say $x+10$ of $p(x)$ and $q(x)$ in Example 2.1, using the inexact data and floating point arithmetic? The examples above indicate the futility of computing the *exact* GCD in this situation. Instead, we compute the *numerical greatest common divisor* to be formulated in §5. Such a numerical GCD will be proven insensitive with a finite GCD condition number. More importantly, the numerical GCD approximates the underlying exact GCD, say $x+10$, with an error in the order of data perturbation, as confirmed by our Maple software UVGCD result:

```
> u, v, w, res := uvGCD(x^10+10.33333333*x^9+3.333333333*x^8+x+10,
    x^10+10.14285714*x^9+1.428571429*x^8-.8571428571*x-8.571428571,x,1E-8):
> u/lcoeff(u,x)   # scale the numerical GCD to be monic
```

$$9.999999998 + 1.000000000\,x$$

The numerical GCD within $10^{-8}$ is $\tilde{u} = x+9.999999998$, an accurate approximation to the exact GCD $x+10$.

## 3. Preliminaries

The fields of complex numbers are denoted by $\mathbb{C}$. The $n$ dimensional complex vector space is denoted by $\mathbb{C}^n$, in which vectors are columns denoted by boldface lower case letters such as $\mathbf{a}$, $\mathbf{u}$, $\mathbf{v}_2$, etc, with $\mathbf{0}$ being a zero vector whose dimension can be understood from the context. Matrices are represented by upper case letters like $A$ and $J$. For every vector or matrix $(\cdot)$, the notation $(\cdot)^\top$ represents the transpose and $(\cdot)^{\mathsf{H}}$ the Hermitian adjoint (i.e. conjugate transpose) of $(\cdot)$. We find it convenient to use Matlab notation ";" to stack (column) vectors as

$$[3;\, -2;\, 4] \;\equiv\; \begin{bmatrix} 3 \\ -2 \\ 4 \end{bmatrix} \;\equiv\; [3,\, -2,\, 4]^\top, \quad [\mathbf{u};\mathbf{v}] \;\equiv\; \begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix} \;\equiv\; [\mathbf{u}^\top, \mathbf{v}^\top]^\top,$$

The norm $\|\mathbf{v}\|$ of a vector $\mathbf{v}$ is the Euclidean norm $\|\mathbf{v}\| = \sqrt{\mathbf{v}^{\mathsf{H}}\mathbf{v}}$ throughout this paper. The matrix norm $\|A\|$ of $A$ is induced from the vector norm $\|A\| = \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|$. We also use the Frobenius norm [**14**, Page 55] denoted by $\|\cdot\|_F$ in some occasions.

All vector spaces are in $\mathbb{C}$. A vector space spanned by vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ is denoted by $\mathit{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$. The notation $\mathit{dist}\,(\mathbb{S}, \mathbb{T})$ stands for the distance between two subspaces $\mathbb{S}$ and $\mathbb{T}$ in a larger vector space [**14**, p. 76]. The dimension of the kernel of matrix $A$ is $\mathit{nullity}\,(A)$, namely the nullity of $A$.

For any matrix $A$ of $m \times n$ with $m \geq n$, there are $n$ singular values [**14**, §2.5.3]

$$\sigma_1(A) \;\geq\; \sigma_2(A) \;\geq\; \ldots \;\geq\; \sigma_n(A) \;\geq\; 0.$$

of $A$ with $\sigma_1(A) = \|A\|$. We shall also denote the same singular values in reversed order

$$0 \;\leq\; \sigma_{-1}(A) \;\leq\; \sigma_{-2}(A) \;\leq\; \cdots \;\leq\; \sigma_{-n}(A) \;=\; \|A\|.$$

Singular value $\sigma_{-1}(A)$ is the smallest distance from $A$ to a matrix that is rank-deficient by one. Likewise, singular value $\sigma_{-2}(A)$ is the smallest distance from $A$ to a matrix that is rank-deficient by two, and so on. The matrix

$$A^+ = (A^{\mathsf{H}}A)^{-1}A^{\mathsf{H}}$$

exists uniquely as the Moore-Penrose inverse of $A$ when $\sigma_{-1}(A) > 0$. It is straightforward to verify that

$$\sigma_{-1}(A) \;=\; \left\|A^+\right\|^{-1} \;=\; \min_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|.$$

This minimum is attainable at the right singular vector $\mathbf{y}$ of $A$ corresponding to $\sigma_{-1}(A)$. Namely $\|A\mathbf{y}\| = \sigma_{-1}(A)$.

In this paper, polynomials are in $\mathbb{C}$ in a single variable $x$. The ring of such polynomials is denoted by $\mathbb{C}[x]$. A polynomial is denoted by lower a case letter, say $f$, $v$, or $p_1$, etc. A polynomial

$$p \;=\; \rho_0 + \rho_1 x + \rho_2 x^2 + \cdots + \rho_n x^n \tag{3.1}$$

is of degree $n$ if $\rho_n \neq 0$, or the degree is $-\infty$ if $f(x) \equiv 0$. We shall denote the degree of a polynomial $p$ by $\mathit{deg}(p)$.

For an integer $n$, the collection of polynomials with degrees less than or equal to $n$ form a vector space

$$P_n \;=\; \left\{p \in \mathbb{C}[x] \;\middle|\; \mathit{deg}(p) \leq n\right\}.$$

Thus the dimension of $P_n$ is

$$\mathit{dim}(P_n) \;=\; \begin{cases} 0 & \text{for } n < 0 \\ n+1 & \text{for } n \geq 0. \end{cases}$$

Throughout this paper, we use the monomial basis $\{1, x, x^2, \ldots, x^n\}$ for $P_n$, in which every polynomial $p$ can be written in the form of (3.1) and corresponds to a *coefficient vector*

$$\mathbf{p} \;=\; [\rho_0;\; \ldots;\; \rho_n] \;\in\; \mathbb{C}^{n+1}.$$

Notice that $\rho_n = 0$ is possible, and the same polynomial $p$ can be embedded in the space $P_m \supset P_n$ with a coefficient vector of higher dimension. Throughout this paper, if a letter (say $f$, $g$, $q_1$) represents a polynomial, the same letter in boldface, say $\mathbf{f}$, $\mathbf{g}$, $\mathbf{q}_1$, is its coefficient vector within a vector space $P_n$ that is clear from the context. The norm $\|a\|$ of polynomial $a \in P_n$ is defined as the Euclidean norm $\|\mathbf{a}\|$ of its coefficient vector $\mathbf{a}$.

We denote the vector space of polynomial pairs $(p, q) \in P_m \times P_n$ as

$$P_{m,n} \;=\; \big\{(p, q) \in \mathbb{C}[x]^2 \;\big|\; \mathit{deg}(p) \leq m,\; \mathit{deg}(q) \leq n\big\},$$

and its subset formed by polynomial pairs of degrees equal to $m$ and $n$, respectively, as

$$\mathscr{P}_{m,n} \;=\; \big\{(p, q) \in P_{m,n} \;\big|\; \mathit{deg}(p) = m,\; \mathit{deg}(q) = n\big\}.$$

For every polynomial pair $(p, q)$, a *greatest common divisor* or GCD of $(p, q)$ is *any* polynomial $u$ of the highest degree that divides both $p$ and $q$. Notice that we do not require a GCD to be monic here to avoid scaling a polynomial by a tiny leading coefficient in computation. In this setting, GCD's are not unique and two GCD's of the same polynomial pair differ by a nonzero constant multiple. We define an equivalence relation $\sim$ between two polynomials in the sense that $f \sim g$ if $f = \alpha\, g$ for a constant $\alpha \neq 0$. Thus the collection of all GCD's of a polynomial pair $(p, q)$ forms a $\sim$-equivalence class, denoted by $\mathit{gcd}(p, q)$, which is unique in the quotient ring $\mathbb{C}[x]/\sim$.

The collection of polynomial pairs with a specified GCD degree $k$ is denoted by

$$\mathscr{P}^k_{m,n} \;=\; \big\{(p, q) \in \mathscr{P}_{m,n} \;\big|\; \mathit{deg}\big(\mathit{gcd}(p, q)\big) = k\big\} \tag{3.2}$$

If $u \in \mathit{gcd}(p, q)$, then polynomials $v = p/u$ and $w = q/u$ are called the *cofactors* of polynomial pair $(p, q)$. The distance between two polynomial pairs, or generally the distance between two polynomial arrays $(p_1, \ldots, p_l)$ and $(q_1, \ldots, q_l)$ is naturally derived from the polynomial norm

$$\big\|(p_1, \ldots, p_l) - (q_1, \ldots, q_l)\big\| \;=\; \sqrt{\|p_1 - q_1\|^2 + \ldots + \|p_l - q_l\|^2}. \tag{3.3}$$

Let $\Psi_n \;:\; P_n \longrightarrow \mathbb{C}^{n+1}$ denote the isomorphism that maps a polynomial $a$ in $P_n$ to its coefficient vector $\mathbf{a}$ in $\mathbb{C}^{n+1}$, namely $\Psi_n(a) = \mathbf{a}$. For a fixed $f \in P_n$ and any $g \in P_m$, the polynomial multiplication $f\, g$ is a linear transformation

$$\mathcal{F}_m \;:\; P_m \longrightarrow P_{m+n} \quad \text{with} \quad \mathcal{F}_m(g) = f \cdot g \text{ for every } g \in P_m.$$

Let $\mathbf{f} = [\phi_0;\ \phi_1;\ \ldots;\ \phi_n]$ be the coefficient vector of $f$. The matrix for the linear transformation $\mathcal{F}_m$ is called a convolution matrix

$$C_m(f) \ = \ \begin{bmatrix} \overbrace{\begin{matrix} \phi_0 & & \\ \vdots & \ddots & \\ \phi_n & & \phi_0 \end{matrix}}^{m+1} \\ \begin{matrix} & \ddots & \vdots \\ & & \phi_n \end{matrix} \end{bmatrix}. \tag{3.4}$$

For polynomials $u \in P_j$ and $v \in P_k$ with coefficient vectors $\mathbf{u}$ and $\mathbf{v}$ respectively,

$$\mathbf{w} \ = \ C_k(u)\,\mathbf{v} \ = \ C_j(v)\,\mathbf{u} \ = \ \Psi_{j+k}(u \cdot v)$$

is the coefficient vector $\mathbf{w} \in \mathbb{C}^{j+k+1}$ of polynomial product $w = u\,v \in P_{j+k}$.

The classical Sylvester matrices can be derived naturally. Let $(p, q)$ be a given pair of polynomials of degrees $m$ and $n$ respectively, if $u$ is a GCD of $(p, q)$ with cofactors $v$ and $w$. Then $p \cdot w - q \cdot v \ = \ uvw - uwv \ = \ 0$, namely

$$C_{n-j}(p)\,\mathbf{w} - C_{m-j}(q)\,\mathbf{v} \ \equiv \ \left[C_{n-j}(p) \mid C_{m-j}(q)\right] \begin{bmatrix} \mathbf{w} \\ -\mathbf{v} \end{bmatrix} \ = \ \mathbf{0}. \tag{3.5}$$

for any degree $j \le \mathit{deg}(u)$. In other words, matrix $\left[C_{n-j}(p) \mid C_{m-j}(q)\right]$ is rank-deficient if $j \le \mathit{deg}(u)$, and the GCD problem is equivalent to the rank/kernel problem of such matrices.

With $p(x) = p_0 + p_1 x + \ldots + p_m x^m \in P_m$, $q(x) = q_0 + q_1 x + \cdots + q_n x^n \in P_n$, and $j = 1, 2, \ldots, \min\{m, n\}$, the $j$-th *Sylvester matrix* of $(p, q)$ in $P_{m,n}$ is defined as

$$S_j(p,q) \ = \ \left[C_{n-j}(p) \mid C_{m-j}(q)\right] \ = \ \begin{bmatrix} \overbrace{\begin{matrix} p_0 & & \\ p_1 & \ddots & \\ \vdots & \ddots & p_0 \\ p_m & & p_1 \\ & \ddots & \vdots \\ & & p_m \end{matrix}}^{n-j+1} & \overbrace{\begin{matrix} q_0 & & \\ q_1 & \ddots & \\ \vdots & \ddots & q_0 \\ q_n & & q_1 \\ & \ddots & \vdots \\ & & q_n \end{matrix}}^{m-j+1} \end{bmatrix} \tag{3.6}$$

Notice that, for convenience of discussion, we extend the use of the Sylvester matrices to allow degrees $\mathit{deg}(p) < m$ and/or $\mathit{deg}(q) < n$ in (3.6). For the special case of $j = 1$ and $(p, q) \in \mathscr{P}_{m,n}$, the matrix $S_1(p, q)$ is the standard Sylvester matrix in the literature whose determinant being zero is frequently used as an equivalent statement for the existence nontrivial GCD. Moreover, the GCD degree can be identified from the nullity of Sylvester matrices in the following lemma.

LEMMA 3.1. *Let $(p, q)$ be a polynomial pair in $\mathscr{P}_{m,n}$ and $S_j(p, q)$ be the $j$-th Sylvester matrix of $(p, q)$ in $P_{m,n}$. Then the degree of $\mathit{gcd}(p, q)$ equals to $k$ if and only if*

$$\mathit{nullity}\,(\,S_j(p,q)\,) \ = \ \mathit{dim}\big(P_{k-j}\big) \ = \ k - j + 1 \tag{3.7}$$

*for $j = 1, \ldots, k$ and $\mathit{nullity}\,(\,S_j(p, q)\,) \ = \ 0$ for $j = k + 1, \ldots, \min\{m, n\}$. In particular,*

(i) $\mathit{nullity}\,(\,S_1(p, q)\,) = \mathit{deg}\big(\mathit{gcd}\,(p, q)\big)$; *and*

(ii) *nullity* $(S_k(p,q)) = 1$ *with the kernel of* $S_k(p,q)$ *being spanned by the vector* $[\mathbf{w}; -\mathbf{v}]$ *formed by the cofactors* $v$ *and* $w$ *of* $(p,q)$.

**Proof.** Let $v$ and $w$ be the GCD cofactors of $(p,q)$. The kernel of $S_j(p,q)$ can be identified from the identity $(x^i w)\cdot p - (x^i v)\cdot q = 0$ for $i = 0, 1, \ldots, k-j$ if $j \geq k$. □

The identity (3.7) in various forms are well known in the literature (see e.g. [**10, 24, 30**]), while the algorithm in [**45, 41**] takes advantage of the special case *nullity* $(S_k(p,q)) = 1$ so that the cofactors $v$ and $w$ can be solved from the homogeneous linear system

$$S_k(p,q)\begin{bmatrix} \mathbf{w} \\ -\mathbf{v} \end{bmatrix} = \mathbf{0}. \tag{3.8}$$

Then the GCD can be determined via solving the linear system

$$C_k(v)\,\mathbf{u} = \mathbf{p} \text{ and } C_k(w)\,\mathbf{u} = \mathbf{q} \tag{3.9}$$

for polynomial $u$.

## 4. Geometry of GCD and its ill-posedness

In this section, we study the geometry of the polynomial GCD problem, the root of its ill-posedness, and the reason why it is not hypersensitive in a restricted domain in which it becomes numerically computable. The regularization theory that follows later is also derived from the differentiable manifolds and the stratification structure formed by the collections of polynomials pairs with common GCD degrees.

Let $(p,q) \in \mathscr{P}_{m,n}$ be a polynomial pair with a particular GCD $u_* \in \mathscr{P}_k$ and cofactors $v_*$ and $w_*$. For any vector $\mathbf{h} \in \mathbb{C}^{k+1}$ with $\mathbf{h}^{\mathsf{H}}\mathbf{u}_* = \beta \neq 0$, this GCD triplet $(u_*, v_*, w_*)$ of $(p,q)$ is the unique solution to the equation

$$\mathbf{f}_{\mathbf{h}}(u,v,w) = [\beta; \mathbf{p}; \mathbf{q}] \tag{4.1}$$

for $u \in P_k$, $v \in P_{m-k+1}$, $w \in P_{n-k+1}$, where

$$\mathbf{f}_{\mathbf{h}}(u,v,w) = \begin{bmatrix} \mathbf{h}^{\mathsf{H}}\mathbf{u} \\ C_k(v)\mathbf{u} \\ C_k(w)\mathbf{u} \end{bmatrix} \tag{4.2}$$

with its Jacobian

$$J_{\mathbf{h}}(u,v,w) = \begin{bmatrix} \mathbf{h}^{\mathsf{H}} & & \\ C_k(v) & C_{m-k}(u) & \\ C_k(w) & & C_{n-k}(u) \end{bmatrix} \tag{4.3}$$

in which a matrix block such as $C_k(v)$ is the convolution matrix (3.4) corresponding to the linear transformation $\mathcal{L} : g \in P_k \longrightarrow v \cdot g \in P_n$.

LEMMA 4.1. *Let polynomials* $u \in \mathscr{P}_k$, $v \in \mathscr{P}_{m-k}$, $w \in \mathscr{P}_{n-k}$ *and the vector* $\mathbf{h} \in \mathbb{C}^{k+1}$ *with* $\mathbf{h}^{\mathsf{H}}\mathbf{u} \neq 0$. *Then the matrix* $J_{\mathbf{h}}(u,v,w)$ *defined in* (4.3) *is injective if and only if there exists no non-constant polynomial that divides* $u$, $v$ *and* $w$ *simultaneously.*

**Proof.** Let $a \in P_k$, $b \in P_{m-k}$ and $c \in P_{n-k}$ be arbitrary polynomials whose coefficient vectors $\mathbf{a} \in \mathbb{C}^{k+1}$, $\mathbf{b} \in \mathbb{C}^{m-k+1}$ and $\mathbf{c} \in \mathbb{C}^{n-k+1}$ satisfy

$$\begin{bmatrix} \mathbf{h}^{\mathsf{H}} & & \\ C_k(v) & C_{m-k}(u) & \\ C_k(w) & & C_{n-k}(u) \end{bmatrix} \begin{bmatrix} -\mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \end{bmatrix} = \mathbf{0}. \tag{4.4}$$

The matrix $J_{\mathbf{h}}(u,v,w)$ is injective if $a = b = c = 0$. Equation (4.4) is equivalent to

$$\mathbf{h}^{\mathsf{H}}\mathbf{a} = 0, \quad b\cdot u - a\cdot v = 0, \quad c\cdot u - a\cdot w = 0. \tag{4.5}$$

If $u$ and $v$ are co-prime, then $bu - av = 0$ in (4.5) implies $a = ru$ for a polynomial $r$. Then $\mathit{deg}(a) \leq k = \mathit{deg}(u)$ leads to $r$ being a constant. Therefore $r = 0$ due to $\mathbf{h}^{\mathsf{H}}\mathbf{a} = r(\mathbf{h}^{\mathsf{H}}\mathbf{u}) = 0$. Namely, $a = 0$, which results in $b = c = 0$ from (4.5).

Assume $u$ and $v$ are not co-prime, namely $d \in \mathit{gcd}(u, v)$ is not a constant but $1 \in \mathit{gcd}(d, w)$. Write $u = d \cdot u_0$ and $v = d \cdot v_0$. From $bu - av = 0$ in (4.5) we have $bu_0 - av_0 = 0$ and thus $u_0$ divides $a$ since $1 \in \mathit{gcd}(u_0, v_0)$, and $a = s \cdot u_0$ for certain polynomial $s$. Combining with $cu - aw = 0$ yields $c \cdot d - s \cdot w = 0$ and thus $s = t \cdot d$ for certain polynomial $t$. Consequently $a = s \cdot u_0 = t \cdot d \cdot u_0 = t \cdot u$. The polynomial $t$ must be a constant because $\mathit{deg}(a) \leq \mathit{deg}(u)$. Moreover $\mathbf{h}^{\mathsf{H}}\mathbf{a} = t(\mathbf{h}^{\mathsf{H}}\mathbf{u}) = 0$ implies $t = 0$ and thus $a = 0$, leading to $b = c = 0$ from (4.5). As a result, we have proved $J_{\mathbf{h}}(u, v, w)$ is injective whenever $u$, $v$ and $w$ have no common non-constant factors.

Assuming there is a non-constant common factor $e$ among $u$, $v$ and $w$, we now prove $J_{\mathbf{h}}(u, v, w)$ is rank-deficient. Write $u = eu_0$, $v = ev_0$ and $w = ew_0$. If $\mathbf{h}^{\mathsf{H}}\mathbf{u}_0 = 0$, then $J_{\mathbf{h}}(u, v, w)[-\mathbf{u}_0; \mathbf{v}_0; \mathbf{w}_0] = \mathbf{0}$ by a straightforward verification, and thus $J_{\mathbf{h}}(u, v, w)$ is rank-deficient. Next we assume $\mathbf{h}^{\mathsf{H}}\mathbf{u}_0 \neq 0$. Let $a = gu_0$, $b = gv_0$ and $c = gw_0$ for $g = e - \gamma$ where $\gamma = (\mathbf{h}^{\mathsf{H}}\mathbf{u})/(\mathbf{h}^{\mathsf{H}}\mathbf{u}_0)$. Then $a \in P_k$, $b \in P_{m-k}$, and $c \in P_{n-k}$ with $bu - av = cu - aw = 0$. The polynomial $g \neq 0$ since $e$ is non-constant, and $a = gu_0 = du_0 - \gamma u_0 = u - \gamma u_0$, leading to $\mathbf{h}^{\mathsf{H}}\mathbf{a} = \mathbf{h}^{\mathsf{H}}(\mathbf{u} - \gamma\mathbf{u}_0) = 0$ Therefore $a$, $b$ and $c$ satisfy (4.5), implying $J_{\mathbf{h}}(u, v, w)$ is rank-deficient. □

Lemma 4.1 directly leads to the following injectiveness corollary for the Jacobian (4.3) at a GCD and cofactors.

COROLLARY 4.2. *Let $(p, q) \in \mathscr{P}^k_{m,n}$. For almost all $\mathbf{h} \in \mathbb{C}^{k+1}$ and $\beta \in \mathbb{C}$, there exists a unique GCD $u_* \in \mathit{gcd}(p, q)$ with cofactor pair $(v_*, w_*)$ satisfying* (4.2)*, and the Jacobian $J_{\mathbf{h}}(u, v, w)$ in* (4.3) *is injective at $(u_*, v_*, w_*)$.*

The following GCD Manifold Theorem provides the essential geometric properties of the GCD problem. We adopt a non-abstract notion of a differentiable manifold from differential topology: A (complex) differential manifold of dimension $d$ is a subset that locally resembles the Euclidean space $\mathbb{C}^d$. More specifically, a subset $\Pi \subset P_{m,n}$ is called a differentiable manifold of dimension $d$ if, for every point $(p, q) \in \Pi$, there is an open neighborhood $\Delta$ of $(p, q)$ in $P_{m,n}$ and a continuously differentiable mapping $\mathbf{g}$ from $\Delta \cap \Pi$ to an open subset $\Lambda$ of $\mathbb{C}^d$ with a continuously differentiable inverse $\mathbf{g}^{-1} : \Lambda \longrightarrow \Delta \cap \Pi$. The differentiable mapping $\mathbf{g}$ is called a *local diffeomorphism* for the manifold $\Pi$, and the *codimension* of $\Pi$ is

$$\mathit{codim}(\Pi) \;=\; \mathit{dim}(P_{m,n}) - \mathit{dim}(\mathbb{C}^d) \;=\; m + n + 2 - d.$$

THEOREM 4.3 (GCD Manifold Theorem). *With respect to the metric topology induced from the norm $\|\cdot\|$ in $P_{m,n}$, the subset $\mathscr{P}^k_{m,n}$ of $P_{m,n}$ is a differentiable manifold of codimension $k$. Moreover, GCD manifolds $\mathscr{P}^j_{m,n} \subset \overline{\mathscr{P}^k_{m,n}}$ if and only if $j \geq k$, and $\mathscr{P}^0_{m,n}$ is open dense in $P_{m,n}$.*

**Proof.** Let $(p, q) \in \mathscr{P}^k_{m,n}$ Then there exist a GCD $\hat{u}$ and its cofactor pair $(\hat{v}, \hat{w})$ that form the unique solution of the equation (4.1) for certain $\mathbf{h} \in \mathbb{C}^{k+1}$ and $\beta \in \mathbb{C}$. Let the column dimension (and rank) of $J_{\mathbf{h}}(u, v, w)$ be denoted as

$$l = (k + 1) + (m - k + 1) + (n - k + 1) = m + n - k + 3.$$

By Corollary 4.2, there exist $l$ rows of $J_{\mathbf{h}}(u, v, w)$ that are linearly independent. By the Inverse Function Theorem [**35**, Theorem 3.7.3, p.52], the vector $[\mathbf{u}; \mathbf{v}; \mathbf{w}]$ is locally a continuously differentiable (vector) function of $l$ entries of the vector

$[\beta; \mathbf{p}; \mathbf{q}]$. These $l$ rows must include the first row since otherwise there would be a contradiction that $(p, q)$ has a unique GCD. Consequently, the vector $[\mathbf{p}; \mathbf{q}]$ is a continuously differentiable function $\mathbf{g}$ of its $l-1$ components in a proper open domain. This mapping $\mathbf{g}$ is a local diffeomorphism and the codimension of $\mathscr{P}^k_{m,n}$ is thus $(m+n+2)-(l-1)=k$.

The manifold $\mathscr{P}^0_{m,n}$ is of codimension zero and thus open in $P_{m,n}$. Consequently, the manifold $\mathscr{P}^k_{m,n}$ is of positive codimension for $k>0$ and $P_{m,n} \setminus \mathscr{P}^k_{m,n}$ is open dense. The manifold $\mathscr{P}^0_{m,n} = \bigcap_{k>0} \left(P_{m,n} \setminus \mathscr{P}^k_{m,n}\right)$ is thus open dense as an intersection of finitely many open dense subsets.

Let $(p, q) \in \mathscr{P}^k_{m,n}$, $\mathscr{P}^{k'}_{m',n'} \subset P_{m,n}$ and $\inf_{(r,s)\in\mathscr{P}^{k'}_{m',n'}} \big\|(p,q)-(r,s)\big\| = 0$. Then there is a sequence $(p_i, q_i) = (u_i \cdot v_i,\ u_i \cdot w_k) \in \mathscr{P}^{k'}_{m',n'}$, $i = 1, 2, \ldots$ converges to $(p, q)$, where $u_i \in \mathscr{P}_{k'}$ is a GCD of $(p_i, q_i)$. It is clear that $m' = m$ and $n' = n$ since $(p_i, q_i)$ can not have lower degrees when $i$ is sufficiently large. Because $\big\{(p_i, q_i)\big\}_{i=1}^{\infty}$ is bounded, the sequences $\big\{u_i\big\}_{i=1}^{\infty}$, $\big\{v_i\big\}_{i=1}^{\infty}$ and $\big\{w_i\big\}_{i=1}^{\infty}$ can be chosen to be bounded and thus can be assumed as convergent sequences to polynomials $u \in P_{k'}$, $v \in P_{m-k'}$ and $w \in P_{n-k'}$, respectively. Consequently, we have $(p, q) = (u \cdot v,\ u \cdot w)$ and thus $u \in \mathscr{P}_{k'}$ since otherwise one would have a contradiction in $\mathit{deg}(p) < m$ and $\mathit{deg}(q) < n$. Therefore $u \in \mathscr{P}_{k'}$ divides $\mathit{gcd}(p, q)$. □

The ill-posedness of exact GCD can now be clearly explained: When a polynomial pair $(f, g) \in \mathscr{P}^k_{m,n}$ for $k>0$ is perturbed, generically the resulting polynomial pair $(\tilde{f}, \tilde{g})$ belongs to $\mathscr{P}^0_{m,n}$ since $\mathscr{P}^k_{m,n}$ is dimension deficient and $\mathscr{P}^0_{m,n}$ is open dense in $P_{m,n}$. Consequently the GCD degree drops from $k$ to 0 discontinuously, degrading the exact GCD to a constant. On the other hand, $\mathscr{P}^k_{m,n}$ is a differentiable manifold and the diffeomorphism (4.2) has a smooth inverse, indicating that the GCD is *not* discontinuous if the perturbation is structure-preserving so that $(\tilde{f}, \tilde{g})$ remains in $\mathscr{P}^k_{m,n}$.

Theorem 4.3 also provides an important geometric property: A small neighborhood of a polynomial pair $(p, q)$ intersect all the GCD manifolds $\mathscr{P}^j_{m,n}$ for $j \le k = \mathit{deg}\big(\mathit{gcd}(p, q)\big)$. Furthermore, the residing manifold $\mathscr{P}^k_{m,n}$ has a distinct identity to be given in Lemma 5.3.

## 5. The notion of numerical GCD

We study the numerical GCD for two simultaneous objectives: To eliminate the ill-posedness of the exact GCD and to solve a specific problem of approximating the GCD that is lost due to data perturbations and round-off errors. The precise problem statement is as follows.

PROBLEM 5.1 (The numerical GCD Problem). *Let $(p, q)$ be a given polynomial pair that constitutes the available data containing a possible perturbation of small magnitude from an underlying pair $(\hat{p}, \hat{q})$. Find the numerical GCD of $(p, q)$, namely a polynomial $u$ of degree identical to $\mathit{deg}\big(\mathit{gcd}(\hat{p}, \hat{q})\big)$ with an accuracy*

$$\inf_{\hat{u}\in\mathit{gcd}(\hat{p},\hat{q})} \big\|u-\hat{u}\big\| \;=\; O\big(\|(p,q)-(\hat{p},\hat{q})\|\big).$$

We have regularized ill-posed problems by formulating a "numerical solution" using a "three-strikes" principle [**43, 44**] that consists of backward nearness, maximum codimension and minimum distance. Namely, the numerical solution of the problem is the exact solution of a nearby problem (backward nearness) that resides

in the manifold of the highest codimension (maximum codimension) and has the minimum distance to the given data (minimum distance).

We shall introduce the *numerical greatest common divisor* as a well-posed problem to make numerical computation feasible. As common in numerical computation, the first and foremost requirement for computing numerical GCD is its backward accuracy: The numerical GCD of a given polynomial pair $(p, q)$ must be the exact GCD of a "nearby" pair $(\tilde{p}, \tilde{q})$ with $\big\|(p, q) - (\tilde{p}, \tilde{q})\big\| < \varepsilon$ for a specified threshold $\varepsilon > 0$. However, a major distinction here is that $(\hat{p}, \hat{q})$ can not be required as the "nearest" pair to $(p, q)$, as shown in Example 5.2 below.

EXAMPLE 5.2. Consider the univariate polynomial pair $(p, q)$:

$$\left\{ \begin{array}{rcl} p(x) & = & (x^2 - 3x + 2)\,(x + 1.0) + 0.01 \\ q(x) & = & (x^2 - 3x + 2)\,(x + 1.2) - 0.01 \end{array} \right. \tag{5.1}$$

which is small perturbation of magnitude $\sqrt{0.0002} \approx 0.01414$ from a polynomial pair with $\gcd(p, q) = x^2 - 3x + 2$ of degree 2. the nearest polynomial pair with a nontrivial GCD is $(\hat{p}_1, \hat{q}_1)$ where

$$\begin{array}{rcl} \hat{p}_1 & \approx & (x - 2.00002)(0.9990\,x^2 - 0.00255\,x - 1.0054) \\ \hat{q}_1 & \approx & (x - 2.00002)(1.0011\,x^2 + 0.2026\,x - 1.1946) \end{array}$$

with distance 0.00168. The GCD of $(\hat{p}_1, \hat{q}_1)$ is of degree 1, not a meaningful approximation to the GCD of degree 2.

In fact, the nearest polynomial pairs with an GCD degree 2 to be approximately

$$\big(\ (x^2 - 3.0001\,x + 1.9998)(1.0017\,x + 1.0026),\ (x^2 - 3.0001\,x + 1.9998)(0.9982\,x + 1.1973)\ \big)$$

with larger distance 0.0111. If one searches the nearest polynomial pair without a proper constraint, the actual GCD degree can be misidentified. □

It is easy to see from Example 5.2 that, if $(p, q)$ is a polynomial pair with a nontrivial GCD, then the pair $(p, q)$ is *closer* to polynomial pairs with GCD of *lower* degrees. This phenomenon is first reported in [**6**] where it is suggested to seek the *highest degree* for the numerical GCD. Other than certain non-generic exceptions as we shall see later, this degree requirement is consistent with a general geometric constraint for regularizing ill-posed problems: The numerical GCD must be an exact GCD of a nearby polynomial pair in the GCD manifold of the *highest codimension*.

We shall call $\mathscr{P}^k_{m,n}$ the *GCD manifold* of degree $k$. A given polynomial pair $(p, q)$ has a distance to each of the GCD manifolds defined as

$$\theta_k(p, q) \;=\; \inf\Big\{ \big\|(p, q) - (r, s)\big\| \;\Big|\; (r, s) \in \mathscr{P}^k_{m,n} \Big\}, \tag{5.2}$$

By Theorem 4.3, those GCD manifolds form a *stratification* assuming $m \geq n$:

$$\emptyset \;=\; \overline{\mathscr{P}^{n+1}_{m,n}} \;\subsetneq\; \overline{\mathscr{P}^{n}_{m,n}} \;\subsetneq\; \cdots \;\subsetneq\; \overline{\mathscr{P}^{1}_{m,n}} \;\subsetneq\; \overline{\mathscr{P}^{0}_{m,n}} \;\equiv\; P_{m,n}, \tag{5.3}$$

where $\overline{S}$ denotes the closure of any set $S$. Consequently, for every $(p, q) \in \mathscr{P}_{m,n}$,

$$0 \;=\; \theta_0(p, q) \;\leq\; \theta_1(p, q) \;\leq\; \cdots \;\leq\; \theta_n(p, q).$$

Particularly, for $(p, q) \in \mathscr{P}^k_{m,n}$,

$$0 \;=\; \theta_0(p, q) \;=\; \cdots \;=\; \theta_k(p, q) \;<\; \theta_{k+1}(p, q) \;\leq\; \cdots \;\leq\; \theta_n(p, q) \tag{5.4}$$

since $k = \mathit{deg}(\mathit{gcd}(p,q))$. The strict inequality in (5.4) holds because, by Lemma 3.1, the singular value $\sigma_{-1}(S_{k+1}(p,q))$ is strictly positive, while $\sigma_{-1}(S_{k+1}(r,s)) = 0$ for all polynomial pair $(r,s) \in \mathscr{P}^{k+1}_{m,n}$.

LEMMA 5.3. *Let* $(\hat{p},\hat{q}) \in \mathscr{P}^k_{m,n}$ *and* $\mathcal{J} = \{\, j | \theta_j(\hat{p},\hat{q}) = 0\}$. *Then* $\mathcal{J} = \{0, 1, \ldots k\}$, *namely* $k = \max \mathcal{J}$. *Furthermore, there exists a* $\theta > 0$ *such that, from any* $(p,q) \in \mathscr{P}_{m,n}$ *with* $\eta = \big\|(p,q) - (\hat{p},\hat{q})\big\| < \theta$, *the GCD degree* $k$ *of* $(\hat{p},\hat{q})$ *is identifiable as*

$$k = \max\{j \mid \theta_j(p,q) < \varepsilon\} \tag{5.5}$$

*for any* $\varepsilon$ *in the interval* $(\eta,\theta)$.

**Proof.** A straightforward verification from the GCD Manifold Theorem. □

When the given polynomial pair $(p,q)$ is a small perturbation from $(\hat{p},\hat{q}) \in \mathscr{P}^k_{m,n}$, it can land in any of the GCD manifold $\mathscr{P}^j_{m,n}$ of lower or equal codimension $j \le k$. However, the underlying GCD degree $k$ distinguishes itself as the *maximum codimension*

$$k \equiv \mathit{codim}\big(\mathscr{P}^k_{m,n}\big) = \max\big\{\, \mathit{codim}\big(\mathscr{P}^j_{m,n}\big) \mid \theta_j(p,q) < \varepsilon\big\} \tag{5.6}$$

of all GCD manifolds $\mathscr{P}^j_{m,n}$ with distance $\theta_j(p,q) < \varepsilon$ if the threshold $\varepsilon$ satisfies

$$\theta_k(p,q) < \varepsilon < \theta_{k+1}(p,q), \tag{5.7}$$

or the more stringent inequalities $\big\|(p,q) - (\hat{p},\hat{q})\big\| < \varepsilon < \frac{1}{2}\theta_{k+1}(\hat{p},\hat{q})$.

Revisiting Example 5.2, the polynomial pair $(p,q)$ in (5.1) is perturbed from $\mathscr{P}^2_{4,4}$, which is in the closure of $\mathscr{P}^1_{m,n}$. By our calculations,

$$\theta_1(p,q) \approx 0.00168,\ \theta_2(p,q) \approx 0.0111 \text{ and } \theta_3(p,q) \approx 0.45.$$

The desired GCD manifold $\mathscr{P}^2_{4,4}$ is the one that possesses the highest codimension 2 and passes through the $\varepsilon$-neighborhood of $(p,q)$ for any $\varepsilon \in (0.0111, 0.225)$.

In this paper, we assume the given polynomial pair $(p,q)$ is a small perturbation from the underlying pair $(\hat{p},\hat{q}) \in \mathscr{P}^k_{m,n}$, such that $\big\|(p,q) - (\hat{p},\hat{q})\big\| \ll \theta_{k+1}(\hat{p},\hat{q})$, and a threshold $\varepsilon$ can be chosen in between and thus (5.7) holds. If the numerical GCD of $(p,q)$ is of the degree $k$ satisfying (5.6), we can recover the underlying GCD degree. Furthermore, the minimum distance from $(p,q)$ to the GCD manifold $\mathscr{P}^k_{m,n}$ can be reached at a pair $(\tilde{p},\tilde{q}) \in \mathscr{P}^k_{m,n}$. We can naturally designate the exact GCD of $(\tilde{p},\tilde{q})$ as the numerical GCD of $(p,q)$.

The essential requirements of following numerical GCD definition are first discovered by Corless, Gianni, Trager and Watt [**6**] in 1995, and formally proposed by Karmarkar and Lakshman in 1996 [**22**].

DEFINITION 5.4. *Let* $(p,q) \in \mathscr{P}_{m,n}$ *and a threshold* $\varepsilon > 0$. *A* numerical greatest common divisor *of* $(p,q)$ *within* $\varepsilon$ *is an exact GCD of* $(\tilde{p},\tilde{q}) \in \mathscr{P}^k_{m,n}$ *where* $k$ *satisfies* (5.5) *and* $\big\|(\tilde{p},\tilde{q}) - (p,q)\big\| = \theta_k(p,q)$. *The* $\sim$*–equivalence class of all numerical GCD's of* $(p,q)$ *is denoted by* $\mathit{gcd}_\varepsilon(p,q)$. *Namely* $\mathit{gcd}_\varepsilon(p,q) = \mathit{gcd}(\tilde{p},\tilde{q})$.

The formulation of numerical GCD is consistent with the "three-strikes principle" which have been successfully applied to other ill-posed problems [**41, 46**].

*Backward nearness:* The numerical GCD of a given polynomial pair $(p,q)$ is the exact GCD of a nearby polynomial pair $(\tilde{p},\tilde{q})$ within a specified distance $\varepsilon$.

*Maximum codimension* of the solution manifold: The nearby pair $(\tilde{p}, \tilde{q})$ resides in the highest codimension manifold $\mathscr{P}^k_{m,n}$ among all the GCD manifolds intersecting the "nearness" $\varepsilon$-neighborhood of the given pair $(p, q)$.

*Minimum distance* to the solution manifold: The pair $(\tilde{p}, \tilde{q})$ is the nearest point on the manifold $\mathscr{P}^k_{m,n}$ to the given pair $(p, q)$.

The numerical GCD defined in Definition 5.4 extends the notion of GCD in the sense that the exact GCD becomes a special case of the numerical GCD. When a pair $(p, q)$ possesses a nontrivial GCD, the numerical GCD $\mathit{gcd}_\varepsilon(p, q)$ and the exact GCD $\mathit{gcd}(p, q)$ are identical for all $\varepsilon$ satisfying $0 < \varepsilon < \theta_{k+1}(p, q)$.

THEOREM 5.5 (GCD Extension Theorem). *There exists a constant $\theta > 0$ associated with every polynomial pair $(\hat{p}, \hat{q}) \in \mathscr{P}_{m,n}$ possessing an exact GCD of degree $k$ such that, for every $(p, q) \in \mathscr{P}_{m,n}$ that is sufficiently close to $(\hat{p}, \hat{q})$, there exists a numerical GCD of $(p, q)$ within every $\varepsilon \in (0, \theta)$. This numerical GCD is unique and is of the same degree $k$. Moreover,*

$$\lim_{(p,q)\to(\hat{p},\hat{q})} \mathit{gcd}_\varepsilon(p, q) \;=\; \mathit{gcd}(\hat{p}, \hat{q}).$$

*When $(p, q) = (\hat{p}, \hat{q})$ in particular, the numerical GCD of $(p, q)$ within $\varepsilon \in (0, \theta)$ is identical to the exact GCD of $(\hat{p}, \hat{q})$.*

**Proof.** There is a minimum distance $\tau$ from all GCD manifolds having a positive distance to $(\hat{p}, \hat{q})$. Let $\xi$ be the minimum magnitude of nonzero coefficients of $(\hat{p}, \hat{q})$ and let $\theta = \frac{1}{2}\min\{\tau, \xi\}$. For any $(p, q)$ with $\|(p, q) - (\hat{p}, \hat{q})\| < \varepsilon$, the distance $\theta_k(p, q) < \varepsilon$ and (5.5) holds, implying $\mathit{deg}\big(\mathit{gcd}_\varepsilon(p, q)\big) = \mathit{deg}\big(\mathit{gcd}(\hat{p}, \hat{q})\big)$. The set

$$\mathcal{S} \;=\; \big\{(f, g) \in P_{m,n} \;\big|\; \|(f, g) - (p, q)\| \le \|(p, q) - (\hat{p}, \hat{q})|\big\} \bigcap \mathscr{P}^k_{m,n}$$

is bounded. Therefore there exists a convergent sequence $(p_i, q_i) \in \mathcal{S}$ converging to $(p_*, q_*) \in \overline{\mathscr{P}^k_{m,n}}$ such that $\lim_{i\to\infty} \|(p_i, q_i) - (p, q)\| = \theta_k(p, q)$. Since

$$\|(p_*, q_*) - (\hat{p}, \hat{q})\| \;\le\; \|(p_*, q_*) - (p, q)\| + \|(p, q) - (\hat{p}, \hat{q})\| \;\le\; 2\varepsilon \;\le\; \xi,$$

hence $(p_*, q_*) \in \mathscr{P}_{m,n}$. If $(p_*, q_*) \notin \mathscr{P}^k_{m,n}$, then $\theta_k(p_*, q_*) = 0$ which lead to $\mathit{deg}\big(\mathit{gcd}(p_*, q_*)\big) > k$ by the GCD Manifold Theorem (Theorem 4.3), contradicting the choice of $\theta \ge 2\tau$. Consequently, the distance $\theta_k(p, q)$ is attainable as $\|(p, q) - (p_*, q_*)\|$, and a $\mathit{gcd}_\varepsilon(p, q)$ exists.

By Definition 5.4, the equivalence class $\mathit{gcd}_\varepsilon(p, q) = \mathit{gcd}(\tilde{p}, \tilde{q})$ where the pair $(\tilde{p}, \tilde{q}) \in \mathscr{P}^k_{m,n}$, and

$$\big\|(\tilde{p}, \tilde{q}) - (\hat{p}, \hat{q})\big\| \le \big\|(\tilde{p}, \tilde{q}) - (p, q)\big\| + \big\|(p, q) - (\hat{p}, \hat{q})\big\| \le 2\,\big\|(p, q) - (\hat{p}, \hat{q})\big\|.$$

Since $\mathscr{P}^k_{m,n}$ is a differentiable manifold and there is a local diffeomorphism that maps $(u, v, w)$ to $(p, q) \in \mathscr{P}^k_{m,n}$ with $u \in \mathit{gcd}(\tilde{p}, \tilde{q})$, we have

$$\lim_{(p,q)\to(\hat{p},\hat{q})} \mathit{gcd}_\varepsilon(p, q) \;=\; \lim_{(\tilde{p},\tilde{q})\to(\hat{p},\hat{q})} \mathit{gcd}(\tilde{p}, \tilde{q}) \;=\; \mathit{gcd}(\hat{p}, \hat{q})$$

and the theorem follows. □

## 6. Strong Hadamard well-posedness of numerical GCD

As introduced by Hadamard, a problem is well-posed (or regular) if its solution satisfies existence, uniqueness, and certain continuity with respect to data. For solving a computational problem accurately using floating point arithmetic with fixed hardware precision, the continuity must be Lipschitz so that the Lipschitz constant

serves as the finite sensitivity measure, or otherwise the problem is still incompatible with numerical computation. For instance, polynomial roots are continuous with respect to coefficients regardless of multiplicities. However, multiple roots are not Lipschitz continuous and thus infinitely sensitive to coefficient perturbations, rendering the root-finding problem extremely difficult until proper regularization is applied [**40, 41, 44**]. Consequently, the well-posed problem is often defined in recent literature as having a finite condition number [**8**]. To emphasize the requirement of finite sensitivity, we call the problem as *strongly* well-posed if the continuity is Lipschitz.

We shall establish the strong Hadamard well-posedness of numerical GCD as formulated in Definition 5.4. Particularly, we shall prove a strong well-posedness in Lipschitz continuity. To this end, we need the following lemma to prove the regularity of the numerical GCD.

LEMMA 6.1. *For an open subset $\Omega$ in $\mathbb{C}^n$, let $\mathbf{f} : \Omega \longrightarrow \mathbb{C}^m$ be analytic with an injective Jacobian $J(\mathbf{z})$ at every $\mathbf{z} \in \Omega$. Assume $\mathbf{f}(\Omega)$ is a differentiable manifold in $\mathbb{C}^n$. Then there is an open neighborhood $\Delta$ of every $\mathbf{z}$ in $\Omega$ and an open subet $\Sigma$ of $\mathbf{f}(\mathbf{z})$ in $\mathbb{C}^m$ such that, for every $\mathbf{b} \in \Sigma$, there is a unique solution $\mathbf{z}_* \in \Delta$ to the least squares problem*

$$\|\mathbf{f}(\mathbf{z}_*) - \mathbf{b}\|^2 = \min_{\mathbf{y}\in\Omega} \|\mathbf{f}(\mathbf{y}) - \mathbf{b}\|^2.$$

*There are also $\sigma$, $\gamma > 0$ such that the Gauss-Newton iteration*

$$\mathbf{z}_{k+1} = \mathbf{z}_k - J(\mathbf{z}_k)^+ [\mathbf{f}(\mathbf{z}_k) - \mathbf{b}], \ k = 0, 1, \cdots \tag{6.1}$$

*converges to $\mathbf{z}_*$ from every initial iterate $\mathbf{z}_0 \in \Delta$ with*

$$\left\|\mathbf{z}_{k+1} - \mathbf{z}_*\right\| \leq \sigma \left\|\mathbf{z}_k - \mathbf{z}_*\right\| + \gamma \left\|\mathbf{z}_k - \mathbf{z}_*\right\|^2 \leq \mu \left\|\mathbf{z}_k - \mathbf{z}_*\right\| \tag{6.2}$$

*for $k = 0, 1, 2, \ldots$ with $\mu = \sigma + \gamma \|\mathbf{z}_0 - \mathbf{z}_*\| < 1$. Moreover, if $\|\mathbf{f}(\mathbf{z}_*) - \mathbf{b}\| = 0$, the convergence rate is quadratic with $\sigma = 0$.*

**Proof.** This is basically a combination of Lemma 2 and Lemma 3 in [**44**] with a minor variation form the statements of [**44**, Lemma 3] and the proof accordingly. □

We now state and prove the following regularity theorem of the numerical GCD.

THEOREM 6.2 (Numerical GCD Regularity Theorem). *The numerical GCD problem is strongly well-posed. More specifically, for every polynomial pair $(\hat{p}, \hat{q}) \in \mathscr{P}_{m,n}$, there is a neighborhood $\mathcal{D}$ of $(\hat{p}, \hat{q})$ in $P_{m,n}$ and a constant $\theta > 0$ such that, for every $(p, q) \in \mathcal{D}$ and $\varepsilon$ in the interval $(\|(p, q) - (\hat{p}, \hat{q})\|, \theta)$, the following assertions hold:*

(i) (Existence) *The numerical GCD $\mathit{gcd}_\varepsilon(p, q)$ exists.*
(ii) (Uniqueness) *$\mathit{gcd}_\varepsilon(p, q)$ is unique in $P_{m,n}/\sim$.*
(iii) (Lipschitz continuity) *There is a constant $\alpha > 0$ such that, for all $(p_1, q_1)$, $(p_2, q_2) \in \mathcal{D}$, we have*

$$\|(u_1, v_1, w_1) - (u_2, v_2, w_2)\| < \alpha \left\|(p_1, q_1) - (p_2, q_2)\right\|.$$

*for certain $u_1 \in \mathit{gcd}_\varepsilon(p_1, q_1)$ and $u_2 \in \mathit{gcd}_\varepsilon(p_2, q_2)$ with cofactor pairs $(v_1, w_1)$ and $(v_2, w_2)$ respectively.*

**Proof.** The existence of $\mathit{gcd}_\varepsilon(p, q) = \mathit{gcd}(\tilde{p}, \tilde{q})$ for $(p, q)$ near $(\hat{p}, \hat{q})$ with $(\tilde{p}, \tilde{q}) \in \mathscr{P}^k_{m,n}$ is part of the GCD Extension Theorem. To prove the uniqueness and the Lipschitz continuity, let $\mathbf{f}_\mathbf{h}(\cdot, \cdot, \cdot)$ be as in (4.2) along with the Jacobian $J_\mathbf{h}(\cdot, \cdot, \cdot)$

as in (4.3) with a proper choice of the scaling vector $\mathbf{h}$. Then there is a unique $\hat{u} \in \mathit{gcd}(\hat{p}, \hat{q})$ along with cofactors $\hat{v}$ and $\hat{w}$ such that $\mathbf{f_h}(\hat{u}, \hat{v}, \hat{w}) = [\hat{\beta};\, \hat{\mathbf{p}};\, \hat{\mathbf{q}}]$ for every scalar $\hat{\beta} > 0$. Applying Lemma 6.1 to $\mathbf{f_h}$, there is a neighborhoods $\Sigma$ of $[\hat{\beta};\, \hat{\mathbf{p}};\, \hat{\mathbf{q}}]$ and $\Delta$ of $(\hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\mathbf{w}})$ respectively such that for every $[\beta;\, \mathbf{p};\, \mathbf{q}] \in \Sigma$, there is a unique $(\tilde{u}, \tilde{v}, \tilde{w}) \in \Delta$ that solves the least squares problem

$$\|\mathbf{f_h}(\tilde{u}, \tilde{v}, \tilde{w}) - [\beta; \mathbf{p}; \mathbf{q}]\| \;=\; \min_{(u,v,w) \in P_k \times P_{m-k} \times \in P_{n-k}} \|\mathbf{f_h}(u, v, w) - [\beta; \mathbf{p}; \mathbf{q}]\|.$$

Let $(\tilde{p}, \tilde{q}) = (\tilde{u}\tilde{v},\, \tilde{u}\tilde{w})$. Then $(\tilde{p}, \tilde{q}) \in \mathscr{P}_{m,n}$ since we can assume that $\mathcal{D}$ is small so that $\mathcal{D} \subset \mathscr{P}_{m,n}$. Thus $(u, v, w) \in \mathscr{P}_k \times \mathscr{P}_{m-k} \times \mathscr{P}_{n-k}$ and $\mathit{deg}\big(\mathit{gcd}(\tilde{p}, \tilde{q})\big) \geq \mathit{deg}(u) = k$. Since $\mathscr{P}^k_{m,n}$ is the GCD manifold of the highest GCD degree near $(p, q)$ within $\varepsilon$, we have $\mathit{deg}\big(\mathit{gcd}(\tilde{p}, \tilde{q})\big) = k$. Consequently, the uniqueness assertion holds.

Let $[\beta;\, \check{\mathbf{p}};\, \check{\mathbf{q}}] \in \Sigma$ and let $(\check{u}, \check{v}, \check{w})$ be the least squares solution to $\mathbf{f_h}(\cdot, \cdot, \cdot) = [\beta;\, \check{\mathbf{p}};\, \check{\mathbf{q}}]$. Apply one step of the Gauss-Newton iteration on $\mathbf{f_h}(u, v, w) = [\beta;\, \check{\mathbf{p}};\, \check{\mathbf{q}}]$ from $(\tilde{u}, \tilde{v}, \tilde{w})$ and denote

$$[\mathbf{u}_1;\, \mathbf{v}_1;\, \mathbf{w}_1] = [\tilde{\mathbf{u}};\, \tilde{\mathbf{v}};\, \tilde{\mathbf{w}}] - J_{\mathbf{h}}(\tilde{u}, \tilde{v}, \tilde{w})^{+}\big(\mathbf{f_h}(\tilde{u}, \tilde{v}, \tilde{w}) - [\beta;\, \check{\mathbf{p}};\, \check{\mathbf{q}}]\big). \tag{6.3}$$

Combining (6.3) with $[\tilde{\mathbf{u}};\, \tilde{\mathbf{v}};\, \tilde{\mathbf{w}}] = [\tilde{\mathbf{u}};\, \tilde{\mathbf{v}};\, \tilde{\mathbf{w}}] - J_{\mathbf{h}}(\tilde{u}, \tilde{v}, \tilde{w})^{+}\big(\mathbf{f_h}(\tilde{u}, \tilde{v}, \tilde{w}) - [\beta;\, \mathbf{p};\, \mathbf{q}]\big)$ yields $\big\|[\mathbf{u}_1;\, \mathbf{v}_1;\, \mathbf{w}_1] - [\tilde{\mathbf{u}};\, \tilde{\mathbf{v}};\, \tilde{\mathbf{w}}]\big\| \leq \big\|J_{\mathbf{h}}(\tilde{u}, \tilde{v}, \tilde{w})^{+}\big\|\big\|[\check{\mathbf{p}};\, \check{\mathbf{q}}] - [\mathbf{p};\, \mathbf{q}]\big\|$. By (6.2),

$$\begin{aligned}
&\big\|[\check{\mathbf{u}};\, \check{\mathbf{v}};\, \check{\mathbf{w}}] - [\tilde{\mathbf{u}};\, \tilde{\mathbf{v}};\, \tilde{\mathbf{w}}]\big\| \\
&\quad\leq\quad \big\|[\check{\mathbf{u}};\, \check{\mathbf{v}};\, \check{\mathbf{w}}] - [\mathbf{u}_1;\, \mathbf{v}_1;\, \mathbf{w}_1]\big\| + \big\|[\mathbf{u}_1;\, \mathbf{v}_1;\, \mathbf{w}_1] - [\tilde{\mathbf{u}};\, \tilde{\mathbf{v}};\, \tilde{\mathbf{w}}]\big\| \\
&\quad\leq\quad \mu\big\|[\check{\mathbf{u}};\, \check{\mathbf{v}};\, \check{\mathbf{w}}] - [\tilde{\mathbf{u}};\, \tilde{\mathbf{v}};\, \tilde{\mathbf{w}}]\big\| + \big\|J_{\mathbf{h}}(\tilde{u}, \tilde{v}, \tilde{w})^{+}\big\|\big\|[\mathbf{p};\, \mathbf{q}] - [\check{\mathbf{p}};\, \check{\mathbf{q}}]\big\|
\end{aligned}$$

Namely

$$\big\|[\check{\mathbf{u}};\, \check{\mathbf{v}};\, \check{\mathbf{w}}] - [\tilde{\mathbf{u}};\, \tilde{\mathbf{v}};\, \tilde{\mathbf{w}}]\big\| \leq \frac{\big\|J_{\mathbf{h}}(\tilde{u}, \tilde{v}, \tilde{w})^{+}\big\|}{1 - \mu}\,\big\|[\mathbf{p};\, \mathbf{q}] - [\check{\mathbf{p}};\, \check{\mathbf{q}}]\big\|$$

where $0 < 1 - \mu < 1$ for a sufficiently small $\Sigma$, leading to the Lipschitz continuity. □

Finding numerical GCD not only is a well-posed problem by the Numerical GCD Regularity Theorem but also solves the problem of computing the GCD accurately from perturbed data, as specified in Problem 5.1, by the following Numerical GCD Approximation Theorem.

Corollary 6.3 (Numerical GCD Approximation Theorem). *The numerical GCD formulated in Definition 5.4 solves Problem 5.1. More specifically, under the assumptions of Theorem 6.2, the numerical GCD $\mathit{gcd}_{\varepsilon}(p, q)$ satisfies the following addition properties.*

(iv) (Identical degrees) $\mathit{deg}\big(\mathit{gcd}_{\varepsilon}(p, q)\big) \;=\; \mathit{deg}\big(\mathit{gcd}(\hat{p}, \hat{q})\big) \;=\; k.$

(v) (Convergence) $\lim\limits_{(p,q) \to (\hat{p}, \hat{q})} \mathit{gcd}_{\varepsilon}(p, q) \;=\; \mathit{gcd}(\hat{p}, \hat{q}).$

(vi) (Bounded sensitivity)

$$\begin{aligned}
&\limsup_{(p,q) \to (\hat{p}, \hat{q})} \frac{\inf\limits_{u \,\in\, \mathit{gcd}_{\varepsilon}(p,\, q),\ \hat{u} \,\in\, \mathit{gcd}(\hat{p},\, \hat{q})} \big\|(u, v, w) - (\hat{u}, \hat{v}, \hat{w})\big\|}{\|(p, q) - (\hat{p}, \hat{q})\|} \\
&\qquad\leq \inf_{\substack{\mathbf{h} \,\in\, \mathbb{C}^{k+1},\ \hat{u} \,\in\, \mathit{gcd}(\hat{p},\, \hat{q}) \\ (\hat{u}\hat{v},\, \hat{u}\hat{w}) \,=\, (\hat{p},\, \hat{q})}} \big\|J_{\mathbf{h}}(\hat{u}, \hat{v}, \hat{w})^{+}\big\| \;<\; \infty
\end{aligned} \tag{6.4}$$

*where $(v, w)$ and $(\hat{v}, \hat{w})$ are cofactor pairs of $(p, q)$ and $(\hat{p}, \hat{q})$ respectively.*

The above theorem for numerical GCD substantially improves the similar result in [**45**, Proposition 2] and justifies the definition

$$\kappa_\varepsilon(p,q) = \inf_{\substack{\mathbf{h}\in\mathbb{C}^{k+1},\ u\in \gcd_\varepsilon(p,q)\\ \|(uv,uw)-(p,q)\|=\theta_k(p,q)}} \left\|J_{\mathbf{h}}(u,v,w)^+\right\| \tag{6.5}$$

of the *numerical GCD condition number* [**45**, Definition 2] of $(p,q)$ within $\varepsilon$. We believe the sensitivity measure (6.4) is optimal.

The condition number $\kappa_\varepsilon(p,q)$ can be estimated as a by-product of numerical GCD computation. Upon exit of the Gauss-Newton iteration (6.1), the last Jacobian $J_{\mathbf{h}}(u_i,v_i,w_i)$ is available along with its QR decomposition. Applying one step of the null vector finder in [**43**, p.130] will yield an approximation of the smallest singular value $\sigma_{min}$ of $J_{\mathbf{h}}(u_i,v_i,w_i)$, while $\|J_{\mathbf{h}}(u_i,v_i,w_i)^+\| = 1/\sigma_{min}$ can substitute for $\kappa_\varepsilon(p,q)$ as a good estimate.

By Definition 5.4, the GCD Extension Theorem, the Numerical GCD Regularity Theorem and the Numerical GCD Approximation Theorem, we have now established the strong Hadamard well-posedness, and validated the so-defined numerical GCD for its intended objective of solving the numerical GCD Problem as stated in Problem 5.1.

Lemma 4.1 provides an insight into the sensitivity of the numerical GCD by specifying the necessary and sufficient condition for $J_{\mathbf{h}}(u,v,w)$ to be rank-deficient. Computing the numerical GCD of $(p,q)$ within $\varepsilon$ is ill-conditioned if and only if $J_{\mathbf{h}}(u,v,w)$ is "nearly" rank-deficient, namely $u$, $v$ and $w$ can be "nearly" divisible by a nonconstant polynomial. Consequently, computing the numerical GCD of $(p,q)$ is not ill-conditioned even if it is also near a other GCD manifold as long as the numerical GCD triplet $u$ and cofactors $v$, $w$ do not share an approximate common divisor.

A typical ill-conditioned example can be constructed in the following example.

EXAMPLE 6.4. Consider the following polynomial pair

$$\begin{cases} p_\delta(x) &= (x^2-1)\big[(x-1+\delta)(x^4+1)\big] \\ q_\delta(x) &= (x^2-1)\big[(x-1-\delta)(x^3+2)\big] \end{cases} \tag{6.6}$$

The GCD triplet consists of

$$u_\delta(x) = x^2-1,\ v_\delta(x) = (x-1+\delta)(x^4+1),\ w_\delta(x) = (x-1-\delta)(x^3+2).$$

For $\delta = 0$, there is a common factor $x-1$ among $u_0$, $v_0$ and $w_0$. Or, $x-1$ "nearly" divides all $u_\delta$ $v_\delta$ and $w_\delta$. Consequently, the pair $(p_\delta, q_\delta)$ is ill-conditioned for $\delta \ll 1$. Our experiment with UVGCD indicates that the condition number $\kappa_\varepsilon(p_\delta,q_\delta) \approx \frac{1.14}{\delta}$. □

**Remark on formulations of numerical GCD.** In 1985, Schönhage [**31**] first proposed the *quasi-GCD* for univariate polynomials that needs to satisfy only the backward nearness. Schönhage also assumes the given polynomial pair can be arbitrarily precise even though it is inexact. In 1995, Corless, Gianni, Trager and Watt [**6**] proposed a "highest degree" requirement of GCD in addition to Schönhage's notion. The same paper also suggests minimizing the distance between the given polynomial pair to the set of pairs with certain GCD degree. In 1996/1998 Karmarkar and Lakshman [**22, 23**] formally defined "highest degree approximate common divisor problem" and explicitly included the requirements of backward nearness, highest degree, and minimum distance. It should be noticed that the understanding of numerical GCD can be significantly different in other works. Notably there is

another notion of numerical GCD as the nearest GCD within a certain given degree [**19, 20**].

## 7. The initial numerical GCD approximation

The GCD degree can be identified by the nullity of the Sylvester matrices (cf. Lemma 3.1). Likewise, the GCD manifold of maximum codimension specified in the definition of numerical GCD can be revealed by the *numerical* nullity of the Sylvester matrices. The following lemma provides a necessary condition for such a GCD manifold to be nearby.

LEMMA 7.1. *Let $(p,q)$ be a polynomial pair in $\mathscr{P}_{m,n}$ and $\varepsilon > 0$. If the distance $\theta_k(p,q)$ between $(p,q)$ and a GCD manifold $\mathscr{P}^k_{m,n}$ is less than $\varepsilon$, then*

$$\sigma_{-i}\big(S_j(p,q)\big) \;<\; \varepsilon \cdot \sqrt{\max\{m,n\} - j + 1} \tag{7.1}$$

*for $i = 1, 2, \ldots, k-j+1$ and $j \le k$, where $\sigma_{-i}\big(S_j(p,q)\big)$ is the $i$-th smallest singular value of the $j$-th Sylvester matrix for $(p,q)$ in $P_{m,n}$.*

**Proof.** Since $\theta_k(p,q) < \varepsilon$, there exists $(r,s) \in \mathscr{P}^k_{m,n}$ such that $\big\|(p,q)-(r,s)\big\| < \varepsilon$. By Lemma 3.1, singular values $\sigma_i\big(S_j(r,s)\big) = 0$ for $i = 1, 2, \ldots, k-j+1$. From the linearity $S_j(p,q) = S_j(r,s) + S_j(p-r, q-s)$ of the Sylvester matrices (3.6) and [**14**, Corollary 8.6.2]

$$\begin{aligned}
\sigma_i\big(S_j(p,q)\big) &\le \sigma_i\big(S_j(r,s)\big) + \big\|S_j(p-r,q-s)\big\| \;\le\; \big\|S_j(p-r,q-s)\big\|_F \\
&= \sqrt{(n-j+1)\|p-r\|^2 + (m-j+1)\|q-s\|^2} \\
&< \varepsilon \cdot \sqrt{\max\{m,n\} - j + 1}.
\end{aligned}$$

□

However, inequality (7.1) does not guarantee the nearness $\theta_k(p,q) < \varepsilon$, as shown in an example in [**10**]. The actual distance $\theta_k(p,q)$ can nonetheless be calculated during the subsequent computation to ensure finding the numerical GCD accurately.

LEMMA 7.2. *For a given $(p,q) \in \mathscr{P}_{m,n}$ and $\varepsilon > 0$, let $(\tilde{p},\tilde{q}) \in \mathscr{P}^k_{m,n}$ be the polynomial pair that defines $\mathit{gcd}_\varepsilon(p,q) = \mathit{gcd}(\tilde{p},\tilde{q})$ containing $\tilde{u}$ with cofactors $\tilde{v}$ and $\tilde{w}$. If $[\mathbf{w};-\mathbf{v}] \in \mathbb{C}^{n-k+1} \times \mathbb{C}^{m-k+1}$ is the singular vector of $S_k(p,q)$ with $\big\|S_k(p,q)[\mathbf{w};-\mathbf{v}]\big\| = \sigma_{-1}\big(S_k(p,q)\big)$, then $\sigma_{-2}\big(S_k(\tilde{p},\tilde{q})\big) \ne 0$ and the distance*

$$\mathit{dist}\left(\mathit{span}\left\{\begin{bmatrix}\mathbf{w}\\ -\mathbf{v}\end{bmatrix}\right\}, \mathit{span}\left\{\begin{bmatrix}\tilde{\mathbf{w}}\\ -\tilde{\mathbf{v}}\end{bmatrix}\right\}\right) < \frac{2\,\varepsilon\,\sqrt{\max\{m,n\} - k + 1}}{\sigma_{-2}\big(S_k(\tilde{p},\tilde{q})\big)}. \tag{7.2}$$

**Proof.** From Lemma 3.1, we have $\sigma_{-2}\big(S_k(\tilde{p},\tilde{q})\big) \ne 0$. Consider the singular value decomposition $S_k(\tilde{p},\tilde{q}) = \big[\tilde{U},\tilde{\mathbf{z}}\big]\begin{bmatrix}\tilde{\Sigma} & \\ & 0\end{bmatrix}\big[\tilde{V},\tilde{\mathbf{y}}\big]^*$ and let $\mathbf{y} = [\mathbf{w};-\mathbf{v}]$. We have

$$\begin{aligned}
\big\|S_k(\tilde{p},\tilde{q})\mathbf{y}\big\| &= \big\|\tilde{\Sigma}\tilde{V}^{\mathsf{H}}\mathbf{y}\big\| \;\ge\; \sigma_{-2}\big(S_k(\tilde{p},\tilde{q})\big)\big\|\tilde{V}^{\mathsf{H}}\mathbf{y}\big\| \text{ and} \\
\big\|S_k(\tilde{p},\tilde{q})\mathbf{y}\big\| &\le \big\|S_k(p,q)\mathbf{y}\big\| + \big\|S_k(\tilde{p},\tilde{q}) - S_k(p,q)\big\|\,\|\mathbf{y}\| \\
&< 2\varepsilon\sqrt{\max\{m,n\} - k + 1}.
\end{aligned}$$

Therefore, the inequality (7.2) follows from the identity [**14**, Theorem 2.6.1]

$$\mathit{dist}\left(\mathit{span}\left\{\begin{bmatrix}\mathbf{w}\\ -\mathbf{v}\end{bmatrix}\right\}, \mathit{span}\left\{\begin{bmatrix}\tilde{\mathbf{w}}\\ -\tilde{\mathbf{v}}\end{bmatrix}\right\}\right) \;=\; \big\|\tilde{V}^{\mathsf{H}}\mathbf{y}\big\|.$$

□

Lemma 7.1 provides mechanisms for identifying the numerical GCD degree and numerical cofactor pair. When inequality (7.1) holds then it is *possible* to have an numerical GCD degree $k$, and $(v, w)$ can be extracted from the right singular vector. The smallest singular value and the corresponding right singular vector can be computed accurately and efficiently using a numerical rank-revealing iteration [**24, 41**] in the following lemma.

LEMMA 7.3. *Under the assumptions of Lemma* 7.1, *assume the inequality* $\sigma_{-2}\big(S_k(p,q)\big) > 2\varepsilon\sqrt{\max\{m,n\}-k+1}$ *holds and* $Q \cdot R$ *is the QR decomposition* [**14**, §5.2] *of* $S_k(p,q)$. *Then, for almost all initial vector* $\mathbf{z}_0$ *of proper dimension, the following iteration*

$$\left\{\begin{array}{l} \text{Solve } R^{\mathsf{H}}\mathbf{y}_j = \mathbf{z}_{j-1} \text{ by forward substitution} \\ \text{Solve } R\,\mathbf{z}_j = \mathbf{y}_j \text{ by backward substitution} \\ \text{Normalize } \mathbf{z}_j, \text{ for } j = 1, 2, \ldots \end{array}\right. \tag{7.3}$$

*generates a sequence of unit vectors* $\mathbf{z}_j$, $j = 1, 2, \ldots$ *converging to* $\mathbf{z}_*$ *and*

$$\|S_k(p,q)\,\mathbf{z}_*\| \;=\; \|R\,\mathbf{z}_*\| \;=\; \sigma_{-1}\big(S_k(p,q)\big) \tag{7.4}$$

*at convergence rate*

$$\|\mathbf{z}_j - \mathbf{z}_*\| \;\le\; \big[\sigma_{-1}\big(S_k(p,q)\big)/\sigma_{-2}\big(S_k(p,q)\big)\big]^{2j}\|\mathbf{z}_0 - \mathbf{z}_*\| \tag{7.5}$$

**Proof.** From $\sigma_{-2}\big(S_k(p,q)\big) > 2\varepsilon\sqrt{\max\{m,n\}-k+1}$ and Lemma 7.1, we have

$$\begin{aligned} \sigma_{-1}\big(S_k(p,q)\big) &\le \varepsilon\sqrt{\max\{m,n\}-k+1} \text{ and} \\ \sigma_{-2}\big(S_k(p,q)\big) &\ge \sigma_{-2}\big(S_k(p,q)\big) - \varepsilon\sqrt{\max\{m,n\}-k+1} \\ &> \varepsilon\sqrt{\max\{m,n\}-k+1} \end{aligned}$$

and thus $\sigma_{-1}\big(S_k(p,q)\big)/\sigma_{-2}\big(S_k(p,q)\big) < 1$. The assertions of the lemma then follows from [**41**, Lemma 2.6]. □

Equations in (7.4) implies $\mathbf{z}_*$ is the vector $[\mathbf{w}; -\mathbf{v}]$ in Lemma 7.2 containing the coefficients of the numerical cofactors $v$ and $w$. The next lemma provides an error estimate for the initial approximation $u$ of the numerical GCD from solving the least squares solution to system (3.9).

LEMMA 7.4. *Under the assumptions of Lemma* 7.1 *and* 7.2 *with the same notations along with* $\mu\,\varepsilon$ *denoting the right hand side of* (7.2), *let*

$$\xi = \left\| \begin{bmatrix} C_k(\tilde{v}) \\ C_k(\tilde{w}) \end{bmatrix} \right\|, \;\; \tau \;=\; \xi \left\| \begin{bmatrix} C_k(\tilde{v}) \\ C_k(\tilde{w}) \end{bmatrix}^{+} \right\|,$$

*and* $\mathbf{z} = \mathbf{u}$ *be the least squares solution to*

$$\begin{bmatrix} C_k(v) \\ C_k(w) \end{bmatrix} \mathbf{z} \;=\; \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix}. \tag{7.6}$$

*If* $\eta = \mu\tau\sqrt{k+1}\,\varepsilon < 1$, *then there is an* $\alpha \in \mathbb{C} \setminus \{0\}$ *such that*

$$\Big\|\tilde{u} - \alpha\,u\Big\| \;\equiv\; \Big\|\tilde{\mathbf{u}} - \alpha\,\mathbf{u}\Big\| \;\le\; \tfrac{\tau}{1-\eta}\left[\sqrt{k+1}\,\|\tilde{u}\|\mu + \tfrac{1}{\xi}\right]\varepsilon \tag{7.7}$$

**Proof.** Let $A = \begin{bmatrix} C_k(\tilde{v}) \\ C_k(\tilde{w}) \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} \tilde{\mathbf{p}} \\ \tilde{\mathbf{q}} \end{bmatrix}$. The overdetermined linear system $A\mathbf{z} = \mathbf{b}$ has a conventional solution $\mathbf{z} = \tilde{\mathbf{u}}$. Due to (7.2), there is a $\gamma \in \mathbb{C} \setminus \{0\}$ such that $\|\gamma(v,w) - (\tilde{v}, \tilde{w})\| \le \mu\,\varepsilon\|(\tilde{v}, \tilde{w})\|$. Rewrite the linear system (7.6) as

$$\begin{bmatrix} C_k(\gamma v) \\ C_k(\gamma w) \end{bmatrix}(\mathbf{z}/\gamma) = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix}$$

that can be considered as the perturbed system $(A+\delta A)(\mathbf{z}+\delta\mathbf{z}) = \mathbf{b}+\delta\mathbf{b}$ where

$$\begin{aligned}
\|\delta A\| &= \left\| \begin{bmatrix} C_k(\gamma v - \tilde{v}) \\ C_k(\gamma w - \tilde{w}) \end{bmatrix} \right\| \le \left\| \begin{bmatrix} C_k(\gamma v - \tilde{v}) \\ C_k(\gamma w - \tilde{w}) \end{bmatrix} \right\|_F \\
&\le \sqrt{k+1} \cdot \big\| \gamma(v,w) - (\tilde{v},\tilde{w}) \big\| = \sqrt{k+1}\,\mu\,\varepsilon \|(\tilde{v},\tilde{w})\|, \\
\|A\| &\ge \frac{1}{\sqrt{k+1}} \|A\|_F = \frac{1}{\sqrt{k+1}} \sqrt{k+1}\,\|(\tilde{v},\tilde{w})\| = \|(\tilde{v},\tilde{w})\|, \\
\|\mathbf{b}\| &= \|(\tilde{p},\tilde{q})\|,\ \|\delta\mathbf{b}\| = \left\| \begin{bmatrix} \mathbf{p}-\tilde{\mathbf{p}} \\ \mathbf{q}-\tilde{\mathbf{q}} \end{bmatrix} \right\| \le \varepsilon,\ \tau = \|A\| \left\|A^{+}\right\|,
\end{aligned}$$

Then inequality (7.7) follows from Theorem 1.4.6 and Remark 1.4.1 in [**25**, pp. 30-31], residual $\|A\mathbf{u}-\mathbf{b}\| = 0$ and $\alpha = 1/\gamma$. □

Lemma 7.1 and Lemma 7.4 lead to the following lemma that ensures the initial approximation of the numerical GCD and cofactors to be sufficiently accurate if the perturbation to the polynomial pair $(p,q)$ is small, satisfying the local convergence condition of the Gauss-Newton iteration given in Lemma 9.1.

LEMMA 7.5. *Let $(\tilde{p},\tilde{q}) \in \mathscr{P}^k_{m,n}$ and $\tilde{u} \in \gcd(\tilde{p},\tilde{q})$ with cofactor pair $(\tilde{v},\tilde{w})$. Then for any $\delta > 0$, there is an $\eta > 0$ such that for all $(p,q) \in \mathscr{P}_{m,n}$ with distance $\big\|(\tilde{p},\tilde{q}) - (p,q)\big\| < \eta$, the inequality $\big\|(\tilde{u},\tilde{v},\tilde{w}) - (\frac{1}{\gamma}u, \gamma v, \gamma w)\big\| < \delta$ holds for certain $\gamma \in \mathbb{C} \setminus \{0\}$ where $(u,v,w)$ is defined in Lemma 7.2 and Lemma 7.4 corresponding to $(p,q)$.*

**Proof.** A straightforward verification using Lemmas 7.1, 7.2 and 7.4. □

In summary, to calculate the numerical GCD of a given polynomial pair $(p,q)$ within a prescribed threshold $\varepsilon$, we first identify the numerical GCD degree $k$. Lemma 7.1 suggests that we can calculate the smallest singular value $\sigma_{-1}\big(S_j(p,q)\big)$ for $j$ decreasing from $\min\{m,n\} = n$ until $\sigma_{-1}\big(S_j(p,q)\big) < \varepsilon\sqrt{m-k+1}$ and set $k = j$. After $k = \deg\big(\gcd_\varepsilon(p,q)\big)$ is determined, the corresponding singular vector of $S_k(p,q)$ provides an approximation $(v,w)$ to $(\tilde{v},\tilde{w})$ with error bound (7.2). An approximation $u$ to $\tilde{u} \in \gcd_\varepsilon(\tilde{p},\tilde{q})$ is obtained by solving the overdetermined linear system (7.6) for the least squares solution $\mathbf{z} = \mathbf{u}$ with error bound (7.7). The triplet $(u,v,w)$ will be taken as an initial iterate for the Gauss-Newton iteration (9.2) for verification and refinement.

## 8. Sensitivity of numerical GCD computation via Sylvester matrices

The sensitivity of the triplet $(u,v,w)$ in Lemma 7.2 and Lemma 7.4 can be measured by the reciprocal of $\sigma_{-2}\big(S_k(\tilde{p},\tilde{q})\big)$, as indicated by inequalities in (7.2) and (7.7). In other words, computing the triplet $(u,v,w)$ by iteration (7.3) in combination with solving the linear system (7.6) is ill-conditioned whenever $\sigma_{-2}\big(S_k(\tilde{p},\tilde{q})\big)$ is tiny. Such ill-condition is certain to occur when the pair $(p,q) \in \mathscr{P}^k_{m,n}$ is also near another GCD manifold $\mathscr{P}^j_{m,n}$ of *higher* GCD degree $j$. We can actually estimate the magnitude of $\sigma_{-2}\big(S_k(\tilde{p},\tilde{q})\big)$ as follows.

Let $(\hat{p},\hat{q}) \in \mathscr{P}^j_{m,n}$ with degree $j > k$ and the distance $\big\|(\tilde{p},\tilde{q}) - (\hat{p},\hat{q})\big\| = \delta$ being small. By Lemma 3.1, $\mathit{nullity}\,(\,S_k(\hat{p},\hat{q})\,) = j-k+1 > 2$. Similar to the proof of Lemma 7.1,

$$\begin{aligned}
\sigma_{-2}\big(S_{\mathbf{k}}(\tilde{p},\tilde{q})\big) &\le \sigma_{-2}\big(S_{\mathbf{k}}(\hat{p},\hat{q})\big) + \delta \cdot \sqrt{\max\{m,n\} - k + 1} \\
&= \delta \cdot \sqrt{\max\{m,n\} - k + 1}.
\end{aligned}$$

Roughly speaking, the error of the numerical GCD triplet $(u, v, w)$ computed as in Lemma 7.2 and Lemma 7.4 is inversely proportional to the distance between the polynomial pair $(\tilde{p}, \tilde{q})$ and the nearest GCD manifold of higher codimension. The following is a typical example in which the polynomial pair is sensitive for computing the initial numerical GCD approximation but well-conditioned if it is measured by the GCD condition number.

EXAMPLE 8.1. Consider the polynomial pair $p_\mu = u \cdot v_\mu$ and $q = u \cdot w$ where

$$u(x) \;=\; x^2 + 1,\; v_\mu(x) = (x - 1 + \mu)(x^4 + 1),\; w(x) = (x - 1)(x^3 - 2)$$

Clearly, $(p_\mu, q) \in \mathscr{P}^2_{7,6}$ with $\mathit{gcd}(p_\mu, q) = u$ for all $\mu \neq 0$, but $(p_\mu, q)$ is near $\mathscr{P}^3_{7,6}$ when $\mu$ is small. In fact, the distance $\theta_3(p_\mu, q)$ between $(p_\mu, q)$ and $\mathscr{P}^3_{7,6}$ is bounded by $\|r_\mu\| = 2\mu$ for $r_\mu = \mu(x^2 + 1)(x^4 + 1)$. While $\sigma_{-2}\big(S_2(p_\mu, q)\big) > 0$ by Lemma 3.7 but the nullity of $S_2(p_\mu - r_\mu, q)$ is at least 2 since

$$\begin{aligned} (p_\mu - r_\mu) \cdot (x^3 - 1) - q \cdot (x^4 + 1) &= 0, \text{ and} \\ (p_\mu - r_\mu) \cdot (x - 1)(x^3 - 1) - q \cdot (x - 1)(x^4 + 1) &= 0. \end{aligned}$$

Hence $\sigma_{-2}\big(S_2(p_\mu - r_\mu, q)\big) = 0$. As a result,

$$\sigma_{-2}\big(S_2(p_\mu, q)\big) \leq \|C_{6-2}(r_\mu)\|_F = 2\sqrt{5}\mu,$$

and it is sensitive to compute the numerical GCD solely relying on Lemma 7.2 and Lemma 7.4. For instance, let $\mu = 10^{-12}$. A straightforward computation of $(\tilde{u}, \tilde{v}, \tilde{w})$ in Matlab by Lemma 7.2 and Lemma 7.4 results only three to four digits accuracy:

$$\begin{aligned} \tilde{u}(x) &\approx x^2 + 0.99992 \\ \tilde{v}(x) &\approx x^5 - 0.9995x^4 + x - 0.9995 \\ \tilde{w}(x) &\approx x^4 - 0.9995x^3 - 2x + 1.9991 \end{aligned}$$

It may seem to be a surprise that computing $\mathit{gcd}_\varepsilon(p_\mu, q)$ is *not* ill-conditioned even if $\mu \ll 1$. The numerical GCD condition number is nearly a constant of moderate magnitude ($\approx 3.55$) for varying $\mu$. Even for $\mu = 10^{-12}$, our software UVGCD still calculates the numerical GCD with an accuracy around machine precision ($\approx 2.2 \times 10^{-16}$). The reason for such a healthy numerical condition is revealed in Lemma 4.1: Even though polynomials $v_\mu$ and $w_\mu$ are close to having a nontrivial common factor $(x - 1)$, the GCD triplet members $u, v_\mu$, and $w$ as a whole are not. □

The sensitivity analysis the example above show that, to ensure accuracy, it is essential to refine the numerical GCD after obtaining an initial approximation to the numerical GCD and cofactors. Such refinement can be carried out by the Gauss-Newton iteration that is to be discussed in the next section.

## 9. Minimizing the distance to a GCD manifold

For a given polynomial pair $(p, q) \in \mathscr{P}_{m,n}$ with the degree $k = \mathit{deg}\big(\mathit{gcd}_\varepsilon(p, q)\big)$ of the numerical GCD being calculated from Lemma 7.1, finding its numerical GCD and cofactors becomes the problem of minimizing the distance from $(p, q)$ to the GCD manifold $\mathscr{P}^k_{m,n}$:

$$\big\|(p, q) - (u \cdot v,\; u \cdot w)\big\| \;=\; \min_{(r,s) \in \mathscr{P}^k_{m,n}} \Big\|(p, q) - (r, s)\Big\|,$$

where $\mathit{deg}(u) = k$, $\mathit{deg}(v) = m-k$ and $\mathit{deg}(w) = n-k$. Naturally, this minimization leads to the least squares problem for the quadratic system consists of

$$C_k(v)\,\mathbf{u} \;=\; \mathbf{p},\; C_k(w)\,\mathbf{u} \;=\; \mathbf{q} \tag{9.1}$$

which are the vector form of $u\cdot v = p$ and $u\cdot w = q$ respectively. However, the system (9.1) is not regular since the least squares solutions are not isolated. Any solution $(u, v, w)$ can be arbitrarily scaled to $(\alpha u,\; v/\alpha,\; w/\alpha)$. A simple auxiliary equation $\mathbf{h}^{\mathsf{H}}\mathbf{u} = \beta$ takes away this dimension of the solution and ensures the Jacobian to be injective.

We minimize the distance from a point $(uv, uw)$ in the GCD manifold to the give polynomial pair $(p, q)$ by solving the system $\mathbf{f_h}(u, v, w) = [\beta;\, \mathbf{p};\, \mathbf{q}]$ as in (4.1) for its least squares solution, where the function $\mathbf{f_h}(u, v, w)$ is defined in (4.2). The Gauss-Newton iteration (6.1) for finding $u_* \in \mathit{gcd}_{\varepsilon}(p, q)$ and cofactors becomes

$$\begin{bmatrix} \mathbf{u}_{j+1} \\ \mathbf{v}_{j+1} \\ \mathbf{w}_{j+1} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_{j} \\ \mathbf{v}_{j} \\ \mathbf{w}_{j} \end{bmatrix} - J_{\mathbf{h}}(u_j, v_j, w_j)^{+} \left( \mathbf{f_h}(u_j, v_j, w_j) - \begin{bmatrix} \beta \\ \mathbf{p} \\ \mathbf{q} \end{bmatrix} \right) \tag{9.2}$$

for $j = 0, 1, \ldots$ where $J_{\mathbf{h}}(\cdot,\cdot,\cdot)$ is the Jacobian of $\mathbf{f_h}(\cdot,\cdot,\cdot)$ given in (4.3). Lemma 4.1 ensures this iteration to be locally convergent for finding the least squares solution $(\hat{u}, \hat{v}, \hat{w})$ to the system $\mathbf{f_h}(u, v, w) = [\beta;\, \mathbf{p};\, \mathbf{q}]$.

LEMMA 9.1. *For $(p, q) \in \mathscr{P}_{m,n}$ with numerical GCD $u_* \in \mathit{gcd}_{\varepsilon}(p, q)$ and cofactor pair $(v_*, w_*)$, let $\mathbf{h} \in \mathbb{C}^{k+1}$ and $\beta = \mathbf{h}^{\mathsf{H}}\mathbf{u}_* \neq 0$. Define $\mathbf{f_h}(\cdot,\cdot,\cdot)$ as in (4.2). There is a $\rho > 0$ such that, if $\big\|(u_*v_*, u_*w_*) - (p, q)\big\| < \rho$, there exists a $\mu > 0$ and the Gauss-Newton iteration (9.2) converges to $(u_*, v_*, w_*)$ from any initial iterate $(u_0, v_0, w_0)$ satisfying $\mathbf{h}^{\mathsf{H}}\mathbf{u}_0 = \beta$ and $\big\|(u_0, v_0, w_0) - (u_*, v_*, w_*)\big\| < \mu$.*

**Proof.** The proof is a straightforward verification using Lemma 6.1. □.

## 10. The two-staged univariate numerical GCD algorithm

Based on the general analysis in previous sections, we present the algorithm originally proposed in [**41**] for computing the numerical GCD triplet $(u, v, w)$ of a given polynomial pair $(p, q) \in \mathscr{P}_{m,n}$ within a given tolerance $\varepsilon$ of backward error $\big\|(p, q) - (uv, uw)\big\|$. The algorithm consists of two stages. At opening stage, we calculate the degree $k$ of the numerical GCD and an initial approximation $(u_0, v_0, w_0)$ to $(u, v, w)$. Then the Gauss-Newton iteration is applied to generate a sequence $(u_j, v_j, w_j)$ such that $(p_j, q_j) = (u_jv_j, u_jw_j) \in \mathscr{P}^k_{m,n}$ converges to $(\tilde{p}, \tilde{q})$ that is the nearest point on the manifold $\mathscr{P}^k_{m,n}$ to the given pair $(p, q)$.

For simplicity, we assume polynomials $p$ and $q$ are arranged such that $(p, q) \in \mathscr{P}_{m,n}$ with $m \geq n$ in this section.

**10.1. The numerical GCD degree and the initial GCD approximation.** Let polynomials $p$ and $q$ be given along with backward error tolerance $\varepsilon$. From Lemma 7.1, there are no numerical GCD's of degree $j$ within $\varepsilon$ when the smallest singular value

$$\sigma_{-1}\big(S_j(p, q)\big) \;>\; \varepsilon\sqrt{\max\{m, n\} - j + 1} \;=\; \varepsilon\sqrt{m - j + 1}.$$

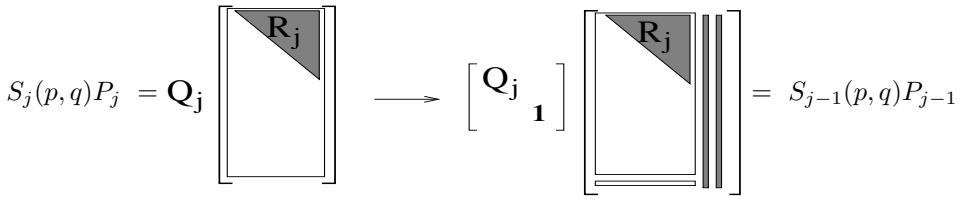
The first stage of numerical GCD computation is to calculated $\sigma_{-1}\big(S_j(p,q)\big)$ for $j$ decreasing from $\min\{m,n\} = n$ through $n-1, n-2, \ldots$ to exclude the possibility of numerical GCD of those degrees. The process *tentatively* stops when $\sigma_{-1}\big(S_k(p,q)\big) \le \varepsilon\sqrt{m-k+1}$, pending certification at the refinement stage.

The full singular value decompositions of $S_j(p,q)$'s are unnecessary. Only the smallest singular value and the associated right singular vector are needed for each $S_j(p,q)$. The iteration (7.3) is specifically designed for our purpose here. It requires the QR decomposition $Q_jR_j = S_j(p,q)$. The straightforward computation of each $R_j$ requires $O(j^2n)$ floating point operations (flops), and the whole process may require $O(n^4)$ which is unnecessarily expansive. A successive QR updating strategy as follows substantially reduces the total flops to $O(n^3)$.

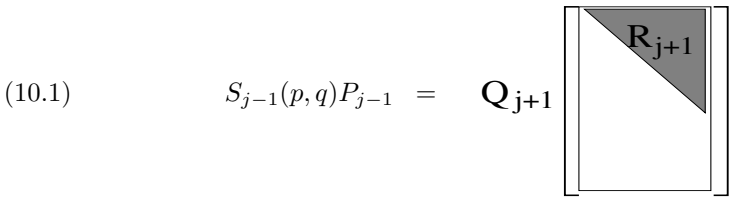
We first calculate the QR decomposition of $S_n(p,q) = \big[\, C_0(p) \big| C_{m-n}(q) \,\big]$:

$$S_n(p,q) = Q_nR_n, \qquad R_n = \begin{bmatrix} * & \cdots & * \\ 0 & \ddots & \vdots \\ 0 & \ddots & * \\ \vdots & & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{bmatrix}_{m\times(m-n+2)}$$

where each "$*$" represents an entry that is potentially nonzero. When $S_j(p,q)$ is formed for $j = n, n-1, \ldots$, the Sylvester matrix $S_{j-1}(p,q)$ is constructed by appending a zero row at the bottom of $S_j(p,q)$ followed by inserting two columns $[\mathbf{0};\, \mathbf{p}]$ and $[\mathbf{0};\, \mathbf{q}]$. By a proper column permutation $P_{j-1}$, these two columns are shifted to the right side as the last two columns of $S_{j-1}(p,q)P_{j-1}$. Let initial column permutation $P_n = I$. Then the QR decomposition $S_n(p,q)P_n = Q_nR_n$ is available. If the QR decomposition $S_j(p,q)P_j = Q_jR_j$ is available for $j = n, n-1, \ldots$, the expansion from $S_j(p,q)P_j$ to $S_{j-1}(p,q)P_{j-1}$ can be illustrated as




Accordingly, we update $Q_jR_j$ to $Q_{j-1}R_{j-1}$ by eliminating the lower triangular entries using the Householder transformation and obtain




The total flops for decomposing all $S_j(p,q)P_j = Q_jR_j$'s is $O(n^3)$.

With the $R = R_j$ at each QR update step, we apply the iteration (7.3) from a randomly generated initial vector $\mathbf{z}_0$. Theoretically, there is a zero probability that such $\mathbf{z}_0$ is perpendicular to the singular subspace $\mathit{span}\{\mathbf{y}\}$ and the condition for convergence may not be satisfied. In practical floating point computation, however,

the round-off error quickly destroys this orthogonality and the iteration *always* converges. Moreover, when $\sigma_{-1}\big(S_j(p,q)\big)$ is near zero, the convergence rate (7.5) in Lemma 7.3 is quite fast. If $\sigma_{-1}\big(S_j(p,q)\big)$ is, say, less than $10^{-3}$ of the second smallest singular value every iteration step in (7.3) will produce 6 correct digits. It rarely takes more than 3 to 5 iterations to reach a near zero $\sigma_{-1}\big(S_j(p,q)\big)$.

According to (7.1) in Lemma 7.1, when $\sigma_{-1}\big(S_j(p,q)\big) < \varepsilon\sqrt{m-j+1}$ is reached along with the singular vector $\mathbf{y}$, the entries of $\mathbf{y}$ form $\mathbf{v}_0$ and $\mathbf{w}_0$ that approximate the coefficients of cofactors $v$ and $w$ respectively with error bound (7.2). By Lemma 7.4, an approximation $u_0$ to the numerical GCD can be obtained from solving the linear system

$$\begin{bmatrix} C_j(v_0) \\ C_j(w_0) \end{bmatrix} \mathbf{u}_0 = \begin{bmatrix} \mathbf{p} \\ \mathbf{q} \end{bmatrix} \tag{10.2}$$

with error bound (7.7).

The iteration (7.3) is applied at $j = n, n-1, \ldots, k$, if the numerical GCD degree is $k$. Each step in (7.3) requires $O((n-j)^2)$ flops. The total cost in calculating the numerical GCD degree is no higher than $O(n^3)$. The system (10.2) costs $O(n^3)$ to solve and it is to be solved only when a possible numerical GCD is detected.

Notice that, inequality $\sigma_{-1}\big(S_j(p,q)\big) \le \varepsilon\sqrt{m-j+1}$ is not a sufficient condition for the given polynomial pair $(p,q)$ to have a numerical GCD of degree $k=j$ within tolerance $\varepsilon$. Satisfying this inequality alone does not guarantee the existence of numerical GCD within $\varepsilon$. Even if the numerical GCD degree is found by this inequality, the numerical GCD triplet $(\hat{u},\hat{v},\hat{w})$ may not be accurate enough. An iterative refinement below verifies the numerical GCD degree and refines the numerical GCD and cofactors.

**10.2. Iterative refinement.** After obtaining a possible numerical GCD degree $k$, a GCD manifold $\mathscr{P}^k_{m,n}$ of codimension $k$ is tentatively targeted for seeking minimum distance to the given polynomial pair $(p,q)$. Using the degree $k$, we set the numerical GCD system $\mathbf{f_h}(u,v,w) = [\beta;\, \mathbf{p};\, \mathbf{q}]$ where $\mathbf{f_h}$ is defined in (4.2) with an objective in finding the least squares solution $(u,v,w)$ with $\mathit{deg}(u) = k$, $(uv, uw) \in \mathscr{P}^k_{m,n}$, and $\big\|(uv,uw) - (p,q)\big\| = \theta_k(p,q)$.

There are several choices for the scaling vector $\mathbf{h}$. If the numerical GCD is required to be monic, then $\mathbf{h} = [0; 0; \ldots; 1]$. There is a drawback in this choice: when leading coefficients of $p$ and $q$ are small, forcing $u$ to be monic may cause its remaining coefficients to be large in magnitude and creating unbalanced system (4.2). A random vector as $\mathbf{h}$ can be a good choice, although there is a zero probability that such $\mathbf{h}$ would make (4.2) singular, or a small probability that the system is ill-conditioned. The most preferred choice appears to be a scalar multiple of the initial approximation $\mathbf{u}_0$ of the numerical GCD determined by (10.2). This is because $u_0$ is close to the numerical GCD $u$ as ensured by Lemma 7.4 when $(p,q)$ is near manifold $\mathscr{P}^k_{m,n}$, and coefficient vector $\mathbf{u}_0$ cannot be perpendicular to $\mathbf{u}$.

With the choice of scaling vector $\mathbf{h} = \beta\mathbf{u}_0$ in (4.2) and the initial approximation $(u_0, v_0, w_0)$ described in §10.1, the Gauss-Newton iteration (9.2) is applied and iteration stops when the distance $\delta_j \equiv \| \mathbf{f_h}(u_j, v_j, w_j) - [\beta;\, \mathbf{p};\, \mathbf{q}] \|$ stops decreasing.

This refinement stage outputs the nearness $\rho = \delta_j$ and the refined numerical GCD triplet $(u,v,w)$. If $\rho < \varepsilon$, then the $(u,v,w)$ is certified as numerical GCD triplet for $p$ and $q$. On the other hand, if the distance $\rho \ge \varepsilon$, then the numerical GCD degree $k$ that is tentatively determined in §10.1 is incorrect and needs to be adjusted downward by one.

At each iteration step in (9.2), it is neither desirable nor necessary to construct the Moore-Penrose inverse $J_{\mathbf{h}}(u_j, v_j, w_j)^+$ in explicit form. The new iterate $(u_{j+1}, v_{j+1}, w_{j+1})$ is obtained via solving a linear least squares problem

$$\begin{cases} \text{Solve} & J_{\mathbf{h}}(u_j, v_j, w_j)(\Delta \mathbf{z}) \;=\; \mathbf{f}_{\mathbf{h}}(u_j, v_j, w_j) - [\beta;\, \mathbf{p};\, \mathbf{q}] \;\text{ for }\; \Delta \mathbf{z} \\ \text{Set} & [\mathbf{u}_{j+1}; \mathbf{v}_{j+1}; \mathbf{w}_{j+1}] \;=\; [\mathbf{u}_j; \mathbf{v}_j; \mathbf{w}_j] - \Delta \mathbf{z}. \end{cases}$$

While solving $J_{\mathbf{h}}(u_j, v_j, w_j)(\Delta \mathbf{z}) = \mathbf{f}_{\mathbf{h}}(u_j, v_j, w_j) - [\beta; \mathbf{p}; \mathbf{q}]$ for its least squares solution, the QR decomposition of $J_{\mathbf{h}}(u_j, v_j, w_j) = Q\,R$ is obtained. Upon exiting the iteration (9.2), the final upper triangular matrix $R$ can replace the $R_j$ in (7.3) to calculate the smallest singular value of $J_{\mathbf{h}}(u_j, v_j, w_j)$. The reciprocal of this singular value is the GCD condition number of the polynomial pair $(p, q)$ and the computed numerical GCD triplet $(u, v, w)$. Calculating the condition number requires negligible flops.

The Gauss-Newton iteration here is crucial for its two-fold purpose:

*Verifying the numerical GCD.* By minimizing the residual $\rho = \| \mathbf{f}_{\mathbf{h}}(u, v, w) - [\beta; \mathbf{p}; \mathbf{q}] \|$, the Gauss-Newton iteration either certifies the numerical GCD triplet $(u, v, w)$ from verifying $\rho < \varepsilon$, or disqualify $k$ as the numerical GCD degree when $\rho \geq \varepsilon$. In the latter case the process of computing $\sigma_{-1}\big(S_j(p, q)\big)$ needs to be continued for decreased $j$ by one.

*Refining the numerical GCD triplet.* The Gauss-Newton iteration filters out the error bounded by (7.2) and (7.7), obtaining the numerical GCD to the optimal accuracy bounded by (6.4).

**10.3. The main algorithm and its convergence theorem.** In summary, the overall algorithm for finding a numerical GCD of a polynomial pair within a tolerance $\varepsilon$ is described in the following pseudo-code, which contains two exit points.

**Algorithm** UVGCD

- **Input**: Pair $(p, q) \in \mathscr{P}_{m,n}$ with $m \geq n$ ,backward nearness tolerance $\varepsilon$.
- Initialize permutation $P_n = I$ and QR decomposition $S_n(p, q)P_n = Q_n R_n$,
- **For** $j = n, n-1, \ldots, 1$ **do**
  - Apply iteration (7.3) on $R = R_j$ and obtain the smallest singular value $\sigma_{-1}\big(S_j(p, q)\big) \equiv \sigma_{-1}\big(S_j(p, q)P_j\big)$ and corresponding singular vector $\mathbf{y} = P_j[\mathbf{w}_0;\, -\mathbf{v}_0]$ of $S_j(p, q)P_j$.
  - **If** $\sigma_{-1}\big(S_j(p, q)\big) < \varepsilon\sqrt{m - j + 1}$ then
    - Set GCD degree as $k = j$, extract $v_0$ and $w_0$ from $\mathbf{y} = P_j[\mathbf{w}_0;\, -\mathbf{v}_0]$, and compute an initial approximation $u_0$ to the numerical GCD by solving (10.2).
    - Set up $\mathbf{f}_{\mathbf{h}}(\cdot, \cdot, \cdot)$, $J_{\mathbf{h}}(\cdot, \cdot, \cdot)$ as in (4.2) and (4.3) with $k = j$ and the scaling vector $\mathbf{h} = \beta \mathbf{u}_0$ for $\beta = 1$. Apply the Gauss-Newton iteration (9.2) with initial iterate $(u_0, v_0, w_0)$ and terminate the iteration at the triplet $(u, v, w) = (u_l, v_l, w_l)$ when the residual $\delta_l = \|\mathbf{f}_{\mathbf{h}}(u_l, v_l, w_l) - [\beta; \mathbf{p}; \mathbf{q}]\|$ stops decreasing; set $\rho = \delta_l$.
    - **If** $\rho < \varepsilon$, then **break** the do-loop, **end if**

    **end if**
  - Update $S_{j-1}(p, q)P_{j-1} = Q_{j-1}R_{j-1}$ as in (10.1).

  **end do**
- **Output** GCD triplet $(u, v, w)$ if $\rho < \varepsilon$, or trivial GCD triplet $(1, p, q)$ if $\rho \geq \varepsilon$.

The following is the Numerical GCD Convergence Theorem for the numerical GCD algorithm. The theorem asserts that Algorithm UVGCD converges to a numerical GCD and cofactors that can be arbitrarily accurate if the given polynomial pair is within a sufficiently small perturbation.

THEOREM 10.1 (Numerical GCD Convergence Theorem). *Let $(\hat{p},\hat{q})$ be any polynomial pair in $\mathscr{P}^k_{m,n}$. Then for every $\delta > 0$, there is an $\eta > 0$ such that, if input items $(p,q) \in \mathscr{P}_{m,n}$ and $\varepsilon > 0$ satisfy*

$$\|(p,q)-(\hat{p},\hat{q})\| \;<\; \eta \;<\; \varepsilon \;<\; \theta_{k+1}(\hat{p},\hat{q}) - \eta,$$

*there is a unique numerical GCD $\mathit{gcd}_\varepsilon(p,q) = \mathit{gcd}(\tilde{p},\tilde{q})$ with $(\tilde{p},\tilde{q}) \in \mathscr{P}^k_{m,n}$ satisfying $\|(\tilde{p},\tilde{q})-(\hat{p},\hat{q})\| < \delta$. Moreover, Algorithm UVGCD generates a sequence of polynomial triplets $(u_j,v_j,w_j)$ satisfying*

$$\lim_{j\to\infty} u_j \;=\; u \in \mathit{gcd}_\varepsilon(p,q) \quad \textit{and} \quad \lim_{j\to\infty} \big\|(u_jv_j,u_jw_j)-(\tilde{p},\tilde{q})\big\| \;=\; 0$$

**Proof.** Assume $m > n$ without loss of generality. From $\mathit{deg}\big(\mathit{gcd}(\hat{p},\hat{q})\big) = k$, we have $\theta_{k+1}(\hat{p},\hat{q}) > 0$ and we can choose a $\eta_1$ with $0 < \eta_1 < \theta_{k+1}(\hat{p},\hat{q})/2$. If $\|(p,q)-(\hat{p},\hat{q})\| < \eta_1$ and $\eta_1 < \varepsilon < \theta_{k+1}(\hat{p},\hat{q}) - \eta_1$, then

$$\sigma_{-1}\big(S_k(p,q)\big) \;<\; \eta_1\sqrt{m-k+1} \;<\; \varepsilon\sqrt{m-k+1}$$

by Lemma 7.1 and the Gauss-Newton iteration (9.2) will be initiated at certain $j \ge k$. For any $j > k$, the distance $\theta_j(p,q) \ge \theta_{k+1}(p,q) \ge \theta_{k+1}(\hat{p},\hat{q}) - \eta_1 > \varepsilon$. Consequently the Gauss-Newton iteration either diverges or converges to a point with residual larger than $\varepsilon$. As a result, Algorithm UVGCD will not be terminated at $j > k$.

From $\|(p,q)-(\hat{p},\hat{q})\| < \eta_1$ and $\eta_1 < \varepsilon < \theta_{k+1}(\hat{p},\hat{q}) - \eta_1$, $\mathscr{P}^k_{m,n}$ is the GCD manifold of highest codimension within $\varepsilon$ of $(p,q)$, namely

$$k = \mathit{codim}\big(\mathscr{P}^k_{m,n}\big) = \max_{0\le j\le n}\big\{\,\mathit{codim}\big(\mathscr{P}^j_{m,n}\big)\,\big|\,\theta_j(p,q) < \varepsilon\,\big\}.$$

Clearly, $\theta_k(p,q) \le \eta_1 < \varepsilon$ is attainable at certain $(\tilde{p},\tilde{q}) \in \overline{\mathscr{P}^k_{m,n}}$. Consequently, the unique numerical GCD $\mathit{gcd}_\varepsilon(p,q)$ exists and is identical to the exact GCD of $(\tilde{p},\tilde{q})$.

Let $\hat{u} \in \mathit{gcd}(\hat{p},\hat{q})$. For any fixed $\mathbf{h} \in \mathbb{C}^{k+1}$ with $\mathbf{h}^{\mathsf{H}}\hat{\mathbf{u}} \ne 0$, let $(\hat{u},\hat{v},\hat{w})$ be the unique solution to the equation $\mathbf{f}_{\mathbf{h}}(u,v,w) = [1;\hat{\mathbf{p}};\hat{\mathbf{q}}]$. By Lemma 6.1, there is a neighborhood $\Delta$ of $(\hat{u},\hat{v},\hat{w})$ and a neighborhood $\Sigma$ of $(\hat{p},\hat{q})$ such that for all $(p,q) \in \Sigma$, the Gauss-Newton iteration on the system $\mathbf{f}_{\mathbf{h}}(u,v,w) = [1,p,q]$ converge to the least squares solution $(u_*,v_*,w_*)$ from any initial iterate $(u_0,v_0,w_0) \in \Delta$. By Lemma 7.5, there is an $\eta_2 > 0$ such that $(u_0,v_0,w_0) \in \Delta$ and $(p,q) \in \Sigma$ whenever $\|(p,q)-(\hat{p},\hat{q})\| < \eta_2$. Set $\eta = \min\{\eta_1,\eta_2\}$, the conclusion of the theorem follows. □

## 11. Computing experiment and benchmark

Our method is implemented as a package UVGCD in Maple and Matlab. In addition to a symbolic GCD-finder GCD, there are three numerical GCD finders in the SNAP package [**17**] in Maple: QUASIGCD [**2**], EPSILONGCD [**2**], and QRGCD [**7**]. Among them QRGCD is clearly superior to the other two by a wide margin. We thereby compare UVGCD with QRGCD and GCD only. Actually, QUASIGCD and EPSILONGCD output failure messages for all the test examples in this section.

All test results are obtained on a desktop PC with an Intel Pentium 4 CPU of 1.8 MHz and 512 Mb memory. Unless mentioned specifically (Example 4), both

UVGCD and QRGCD are tested in Maple 9 with precision set to 16 digits to simulate hardware precision.

We believe that numerical GCD finders should be tested and compared based on results from the following aspects.

(1) Performance on polynomials with increasing numerical GCD sensitivity.
(2) Performance on polynomial having different numerical GCD's within different tolerance.
(3) Performance on numerical GCD's of large degrees.
(4) Performance on polynomials with large variation in coefficient magnitudes.
(5) Performance in finding the numerical GCD of $(p, p')$ when $p$ has roots of high multiplicities.

We have established a test suite that includes polynomials satisfying the above requirements along with those collected from the literature. We demonstrate the robustness and accuracy of UVGCD with sample results below.

**Test 1: A high sensitivity case.** For an even number $n$ and $k = n/2$, let $p_n = u_n\, v_n$ and $q_n = u_n\, w_n$, where

$$u_n = \prod_{j=1}^{k} \left[(x - r_1\alpha_j)^2 + r_1^2\beta_j^2\right],\ v_n = \prod_{j=1}^{k} \left[(x - r_2\alpha_j)^2 + r_2^2\beta_j^2\right],$$

$$w_n = \prod_{j=k+1}^{n} \left[(x - r_1\alpha_j)^2 + r_1^2\beta_j^2\right], \quad \alpha_j = \cos\frac{j\pi}{n}, \beta_j = \sin\frac{j\pi}{n}$$

for $r_1 = 0.5$, $r_2 = 1.5$. The roots of $p_n$ and $q_n$ spread on the circles of radius 0.5 and 1.5. When $n$ increases, the GCD condition number grows quickly. Table 1 shows that error on the computed numerical GCD.

| $n$ | condition number | QRGCD error | uvGCD error |
|---|---|---|---|
| $n = 6$ | 566.13 | $0.55 \times 10^{-14}$ | $0.15 \times 10^{-14}$ |
| $n = 10$ | 742560.0 | $0.18 \times 10^{-11}$ | $0.47 \times 10^{-12}$ |
| $n = 16$ | $0.33 \times 10^{11}$ | $0.18 \times 10^{-4}$ | $0.65 \times 10^{-9}$ |
| $n = 18$ | $0.17 \times 10^{13}$ | FAIL | $0.53 \times 10^{-5}$ |
| $n = 20$ | $0.71 \times 10^{14}$ | FAIL | $0.99 \times 10^{-6}$ |

TABLE 1. Comparison in Test 1

**Test 2: Multiple numerical GCD's**. Let

$$\begin{aligned} p(x) &= \textstyle\prod_{j=1}^{10}(x - x_j), \quad \text{with} \quad x_j = (-1)^j \left(\frac{j}{2}\right) \\ q(x) &= \textstyle\prod_{j=1}^{10}\left[x - x_j + 10^{-j}\right] \end{aligned}$$

The roots of $q$ have decreasing distances $0.1, 0.01, \ldots$ with those of $p$. Therefore there are different numerical GCD's for different tolerances. As shown in Table 5.2, UVGCD accurately separates the numerical GCD factors according to the given tolerance on the listed cases.

**Test 3: numerical GCD of large degrees**. For fixed cofactors $v(x) = \sum_{j=0}^{3} x^j$ and $w(x) = \sum_{j=0}^{4}(-x)^j$, let $p_n = u_n\, v$ and $q_n = u_n\, w$ with $u_n$ being a polynomial of degree $n$ of random integer coefficients in $[-5, 5]$. For the sequence of polynomial pairs $(p_n, q_n)$, the GCD is known to be $u_n$ and we can calculate the actual accuracy. As shown in Table 11, UVGCD maintains its robustness and high accuracy even when for $n$ reaches 2000, while QRGCD works for $n < 100$.

| tolerance | degree (& nearness) of numerical GCD found by | |
|---|---|---|
| $\varepsilon$ | QRGCD | UVGCD |
| $10^{-2}$ | 7 (0.0174) | 9 (0.56E-02) |
| $10^{-3}$ | Fail | 8 (0.26E-03) |
| $10^{-4}$ | Fail | 7 (0.14E-04) |
| $10^{-5}$ | Fail | 6 (0.11E-05) |
| $10^{-6}$ | Fail | 5 (0.41E-07) |
| $10^{-8}$ | Fail | 4 (0.42E-08) |
| $10^{-9}$ | Fail | 3 (0.14E-09) |
| $10^{-10}$ | Fail | 2 (0.24E-10) |

TABLE 2. The calculated degrees (and nearness in parentheses) of numerical GCD within various tolerance on Test 2.

| GCD | coefficient-wise error on computed numerical GCD | |
|---|---|---|
| degree | QRGCD | UVGCD |
| $n = 50$ | 0.168E-12 | 0.500E-15 |
| $n = 80$ | 0.927E-12 | 0.805E-15 |
| $n = 100$ | Fail | 0.341E-15 |
| $n = 200$ | Fail | 0.100E-14 |
| $n = 500$ | Fail (*) | 0.133E-14 |
| $n = 1000$ | Fail (*) | 0.178E-14 |
| $n = 2000$ | Fail (*) | 0.178E-14 |

TABLE 3. Coefficient errors on random numerical GCD's of degree $n$. (*): Presumably failed after running hours without results.

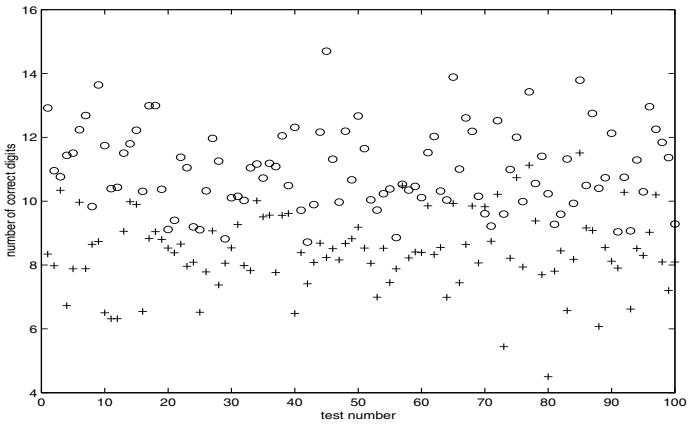
**Test 4: A case where computing numerical GCD by UVGCD is faster than calculating GCD by Maple.** For polynomials with integer coefficients, Maple's symbolic GCD finder is often faster than UVGCD. However, UVGCD can be substantially more efficient in other cases. Here is an example. For fixed cofactors $v$ and $w$ as in Test 3, let $u_n$ be the polynomial of degree $n$ with random rational coefficients and

$$p_n = u_n\, v, , \quad q_n = u_n\, w. \tag{11.1}$$

The GCD is a multiple of $u_n$. We compare the Maple GCD on exact coefficients with our Matlab UVGCD on approximate coefficients. Table 4 shows the running time on increasing $n$. In this polynomial series, not only UVGCD is faster, the speed ratio of UVGCD over GCD increases from 2 to 11 when $n$ increases from 50 to 2000. Of course, this result should be taken with caution because Maple GCD always has zero error.

| | Maple GCD | | UVGCD | |
|---|---|---|---|---|
| | time | error | time | error |
| $n = 50$ | 0.25 | 0 | 0.125 | 3.53e-15 |
| $n = 200$ | 7.47 | 0 | 2.437 | 8.69e-14 |
| $n = 1000$ | 574.90 | 0 | 82.270 | 1.64e-13 |
| $n = 2000$ | 10910.60 | 0 | 969.625 | 1.82e-12 |

TABLE 4. Comparison between Maple's symbolic GCD and UVGCD on polynomial pairs $(p_n, q_n)$ in (11.1)




○: UVGCD, +: QRGCD

FIGURE 1. Accuracy comparison on 100 polynomial pairs in Test 4. The height of each point is the number of correct digits approximating the numerical GCD coefficients by UVGCD or QRGCD at each polynomial pair.

**Test 5: Numerical GCD with large variation in coefficient magnitudes**. For fixed $v$ and $w$ as in Test 3, let

$$u(x) = \sum_{j=0}^{15} c_j \, 10^{e_j} x^j$$

where for every $j$, $c_j$ and $e_j$ are random integers in $[-5, 5]$ and $[0, 6]$ respectively. The polynomial pair $p = u\,v$ and $q = u\,w$ are then constructed while QRGCD and UVGCD are called to find the numerical GCD of $(p, q)$. Notice that $u$ is the known GCD whose coefficient jumps between 0 and $5 \times 10^6$ in magnitude. After applying the numerical GCD finders on each pair $(p, q)$, we calculated the coefficient-wise relative errors $\theta$ and $\vartheta$ of QRGCD and UVGCD respectively. Roughly speaking, $-\log_{10}\theta$ and $-\log_{10}\vartheta$ are the minimum number of correct digits obtained for approximating coefficients of $u$ by QRGCD and UVGCD respectively. This test is repeated 100 times. Figure 1 shows that on average QRGCD gets about 8 digits correct on each coefficient, while UVGCD attains about 11.

Figure 2 shows the *difference* in the number of correct digits obtained on coefficients from each test. On those 100 tests, UVGCD obtains up to 6.5 more correct digits than QRGCD on 99 test, while slightly less accurate than QRGCD on only one polynomial pairs (i.e. the test 70).

**Test 6: GCD of $p$ and $p'$.** Let $p$ be

$$p(x) = (x-1)^{m_1}(x-2)^{m_2}(x-3)^{m_3}(x-4)^{m_4}$$

for different sets of $m_1, m_2, m_3, m_4$. Finding the numerical GCD of $p$ and $p'$ may be difficult for some numerical GCD finders, as shown in Table 5 for QRGCD and [**30**]. This numerical GCD computation has an important application in polynomial root-finding. On the other hand, UVGCD is originally built for this purpose and shows its tremendous robustness.

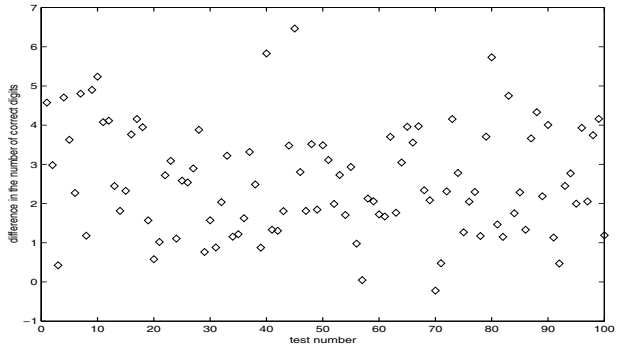



FIGURE 2. Accuracy comparison on 100 polynomial pairs in Test 4. The vertical axis is the difference in the number of correct digits approximating the numerical GCD coefficients by UVGCD or QRGCD at each test.

| $[m_1, m_2, m_3, m_4]$ | coefficient-wise relative error | | |
|---|---|---|---|
| | Maple QRGCD | Maple GCD | UVGCD |
| $[2, 1, 1, 0]$ | 1.0E-13 | 1.0e-16 | 6.7E-16 |
| $[3, 2, 1, 0]$ | 1.5E-12 | 1.0e-16 | 1.8E-14 |
| $[4, 3, 2, 1]$ | 1.6E-07 | 1.0e-16 | 4.5E-14 |
| $[5, 3, 2, 1]$ | Fail | 3.5e-16 | 4.6E-13 |
| $[9, 6, 4, 2]$ | Fail | Fail(*) | 3.5E-12 |
| $[20, 14, 10, 5]$ | Fail | Fail(*) | 1.7E-12 |
| $[80, 60, 40, 20]$ | Fail | Fail(*) | 3.5E-11 |
| $[100, 60, 40, 20]$ | Fail | Fail(*) | 2.6E-11 |

TABLE 5. Comparison on $gcd(p, p')$ for $p$ in Test 5.
(*): Symbolic GCD fails because $p$ is no longer exact.

## References

[1] S. Barnett. *Polynomials and Linear Control Systems*. Monographs and textbooks in pure and applied mathematics, Marcel Dekker, Inc, New York, 1983.

[2] B. Beckermann and G. Labahn. A fast and numerically stable Euclidean-like algorithm for detecting relatively prime numerical polynomials. *J. Symb. Comp.*, 26:691–714, 1998.

[3] R. P. Brent and H. T. Kung. Systolic VLSI arrays for polynomial GCD computation. *IEEE Trans. on Computers*, C-33:731–736, 1984.

[4] P. Chin, R. M. Corless, and G. F. Corless. Optimization strategies for the approximate GCD problem. Proc. ISSAC '98, ACM Press, pp 228-235, 1998.

[5] S.-C. Chou. *Mechanical Geometry Theorem Proving*. D. Reidel Publishing Co., Dordrecht Holand, 1988.

[6] R. M. Corless, P. M. Gianni, B. M. Trager, and S. M. Watt. The singular value decomposition for polynomial systems. Proc. ISSAC '95, ACM Press, pp 195-207, 1995.

[7] R. M. Corless, S. M. Watt, and L. Zhi. QR factoring to compute the GCD of univariate approximate polynomials. *IEEE Trans. Signal Processing*, 52:3394–3402, 2003.

[8] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, Philadelphia, 1997.

[9] D. K. Dunaway. Calculation of zeros of a real polynomial through factorization using Euclid algorithm. *SIAM J. Numer. Anal.*, 11:1087–1104, 1974.

[10] I. Z. Emiris, A. Galligo, and H. Lombardi. Certified approximate univariate GCDs. *J. Pure Appl. Algebra*, 117/118:229–251, 1997.

[11] O. D. Faugeras. *Three-Dimentional Computer Vision: A Geometric Viewpoint*. MIT Press, Cambridge, Mass., 1993.

[12] J. V. Gathern and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, second edition, 2003.
[13] K. O. Geddes, S. R. Czapor, and G. Labahn. *Algorithms for Computer Algebra*. Kluwer Academic Publishers, Boston, 1992.
[14] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore and London, 3rd edition, 1996.
[15] D. Henrion and M. Sebek. Reliable numerical methods for polynomial matrix triangulation. *IEEE Trans. on Automatic Control*, 44:497–501, 1997.
[16] V. Hribernig and H. J. Stetter. Detection and validation of clusters of polynomial zeros. *J. Symb. Comput.*, 24:667–681, 1997.
[17] C.-P. Jeannerod and G. Labahn. The SNAP package for arithmetic with numeric polynomials. In International Congress of Mathematical Software, World Scientific, pages 61-71, 2002.
[18] H. Kai and M.-T. Noda. Hybrid rational approximation and its applications. *Reliable Computing*, 6:429–438, 2000.
[19] E. Kaltofen, J. May, Z. Yang, and L. Zhi. Structured low rank approximation of Sylvester matrix. *in Symbolic-Numeric Computation, Trends in Mathematics, D. Wang and L. Zhi, editors, Birkhäuser Verlag, Basel, Switzerland*, pages 69–83, 2007.
[20] E. Kaltofen, Z. Yang, and L. Zhi. Approximate greatest common divisor of several polynomials with linearly constrained coefficients and singular polynomials. Proc. ISSAC'06, ACM Press, pp 169–176, 2006.
[21] N. Karcanias and M. Mitrouli. Numerical computation of the Least Common Multiple of a set of polynomials. *Reliable Computing*, 6:439–457, 2000.
[22] N. K. Karmarkar and Y. N. Lakshman. Approximate polynomial greatest common divisors and nearest singular polynomials. Proc. ISSAC'96, pp 35-42, ACM Press, 1996.
[23] N. K. Karmarkar and Y. N. Lakshman. On approximate polynomial greatest common divisors. *J. Symb. Comput.*, 26:653–666, 1998.
[24] T.-Y. Li and Z. Zeng. A rank-revealing method with updating, downdating and applications. *SIAM J. Matrix Anal. Appl.*, 26:918–946, 2005.
[25] Åke Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996.
[26] J. P. Merlet. *Les Robots Parallèles*. Traités de Nouvellles Technologiques, Hermeès, 1990.
[27] M.-T. Noda and T. Sasaki. Approximate GCD and its application to ill-conditioned algebraic equations. *J. Comput. Appl. Math.*, 38:335–351, 1991.
[28] V. Y. Pan. Numerical computation of a polynomial GCD and extensions. *Information and Computation*, 167:71–85, 2001.
[29] S. Pillai and B. Liang. Blind image deconvolution using GCD approach. *IEEE Trans. Image Processing*, 8:202–219, 1999.
[30] D. Rupprecht. An algorithm for computing certified approximate GCD of n univariate polynomials. *J. Pure and Appl. Alg.*, 139:255–284, 1999.
[31] A. Schönhage. Quasi-GCD computations. *J. Complexity*, 1:118–137, 1985.
[32] T. W. Sederberg and G. Z. Chang. Best linear common divisors for approximate degree reduction. *Comp.-Aided Design*, 25:163–168, 1993.
[33] H. J. Stetter. *Numerical Polynomial Algebra*. SIAM, 2004.
[34] P. Stoica and T. Söderström. Common factor detection and estimation. *Automatica*, 33:985–989, 1997.
[35] J. Taylor. *Several Complex Variables with Connections to Algebraic Geometry and Lie Groups*. American Mathematical Society, Providence, Rhode Island, 2000.
[36] J. Verschelde and Y. Wang. Computing dynamic output feedback laws. *IEEE Trans. Automatic Control*, pages 1552–1571, 2004.
[37] J. R. Winkler and J. D. Allan. Structured low rank approximations of the Sylvester resultant matrix for approximate GCDs of Bernstein basis polynomial. *Electronic Transactions on Numerical Analysis*, pages 141–155, 2008.
[38] J. R. Winkler and J. D. Allan. Structured total least norm and approximate GCDs of inexact polynomial. *J. of Computational and Applied Math.*, pages 1–13, 2008.
[39] C. J. Zarowski, X. Ma, and F. W. Fairman. A QR-factorization method for computing the greatest common divisor of polynomials with real-valued coefficients. *IEEE Trans. Signal Processing*, 48:3042–3051, 2000.

[40] Z. Zeng. Algorithm 835: MultRoot – A Matlab package for computing polynomial roots and multiplicities. *ACM Trans. Math. Software*, 30:218–235, 2004.
[41] Z. Zeng. Computing multiple roots of inexact polynomials. *Math. Comp.*, 74:869–903, 2005.
[42] Z. Zeng. ApaTools: A Maple and Matlab toolbox for approximate polynomial algebra. In M. Stillman, N. Takayama, and J. Verschelde, editors, *Software for Algebraic Geometry, IMA Volume 148*, pages 149–167. Springer, 2008.
[43] Z. Zeng. Regularization and matrix computation in numerical polynomial algebra. In *Approximate Commutative Algebra*, SpringerWienNewYork, L. Robbinano and J. Abbott eds., pp. 125–162, 2009.
[44] Z. Zeng. The approximate irreducible factorization of a univariate polynomial. Revisited. Proceedings of ISSAC '09, ACM Press, pp. 367–374, 2009.
[45] Z. Zeng and B. Dayton. The approximate GCD of inexact polynomials. II: A multivariate algorithm. Proceedings of ISSAC'04, ACM Press, pp 320-327, (2004).
[46] Z. Zeng and T.-Y. Li. A numerical method for computing the Jordan Canonical Form. Preprint, 2007.
[47] R. Zippel. *Effective Polynomial Computation*. Kluwer Academic Publishers, Boston, 1993.

Department of Mathematics, Northeastern Illinois University, Chicago, IL 60625
*E-mail address*: zzeng@neiu.edu