Foundations of Computational Mathematics (2020) 20:331–366
https://doi.org/10.1007/s10208-019-09422-2

FOUNDATIONS OF COMPUTATIONAL MATHEMATICS
The Journal of the Society for the Foundations of Computational Mathematics

Computing Approximate Greatest Common Right Divisors of Differential Polynomials

Mark Giesbrecht[1] · Joseph Haraldson[1] · Erich Kaltofen[2]

Received: 8 January 2017 / Revised: 4 May 2018 / Accepted: 21 December 2018 / Published online: 13 June 2019

Abstract
Differential (Ore) type polynomials with "approximate" polynomial coefficients are introduced. These provide an effective notion of approximate differential operators, with a strong algebraic structure. We introduce the approximate greatest common right divisor problem (GCRD) of differential polynomials, as a non-commutative generalization of the well-studied approximate GCD problem. Given two differential polynomials, we present an algorithm to find *nearby* differential polynomials with a non-trivial GCRD, where *nearby* is defined with respect to a suitable coefficient norm. Intuitively, given two linear differential polynomials as input, the (approximate) GCRD problem corresponds to finding the (approximate) differential polynomial whose solution space is the intersection of the solution spaces of the two inputs. The approximate GCRD problem is proven to be locally well posed. A method based on the singular value decomposition of a differential Sylvester matrix is developed to produce an initial approximation of the GCRD. With a sufficiently good initial approximation, Newton iteration is shown to converge quadratically to an optimal solution. Finally, sufficient conditions for existence of a solution to the global problem are presented along with examples demonstrating that no solution exists when these conditions are not satisfied.

Keywords Symbolic–numeric computation · Approximate polynomial computation · Approximate GCD · Differential polynomials · Linear differential operators

Mathematics Subject Classification 12Y05 · 13P05 · 13N10 · 49M15

Communicated by Peter Bürgisser.

This research was partly supported by the Natural Sciences and Engineering Research Council (NSERC) Canada (Giesbrecht and Haraldson) and by the National Science Foundation (NFS) under Grant CCF-1421128 (Kaltofen).

Extended author information available on the last page of the article

1 Introduction

The problem of computing the GCRD in a symbolic and exact setting dates back to Ore [23], who presents a Euclidean-like algorithm. See [5] for an elaboration of this approach. Li and Nemes [22] introduce a differential-resultant-based algorithm which makes computation of the GCRD very efficient using modular arithmetic. The technique of Li and Nemes [22] is an extension of ideas presented by Grigor'ev [12] for computing GCRDs of differential operators.

The analogous approximate GCD problem for usual (commutative) polynomials has been a key topic of research in symbolic–numeric computing since its inception. A full survey is not possible here, but we note the deep connection between our current work and that of [7]; see also [19], [26], and [31]. Also important to this current work is the use of so-called structured (numerical) matrix methods for approximate GCD, such as structured total least squares (STLS) and structured total least norm (STLN); see [3] and [15]. More directly employed later in this paper is the multiple polynomial approximate GCD method of Kaltofen et al. [16]. This latter paper also provides a nice survey of the current state of the art in approximate GCDs. Finally, we modify the proof of Kaltofen et al. [17], an optimization approach to computing the GCD of multiple, multivariate *commutative* polynomials, to prove the existence of a globally nearest GCRD.

The goal of this paper is to devise an efficient, numerically robust algorithm to compute the GCRD when the coefficients in $\mathbb{R}$ are given approximately. Given $f, g \in \mathbb{R}(t)[\partial;']$, we wish to find $\widetilde{f}, \widetilde{g} \in \mathbb{R}(t)[\partial;']$, where $\widetilde{f}$ is *near* f and $\widetilde{g}$ is *near* g, such that $\deg_\partial \operatorname{gcrd}(\widetilde{f}, \widetilde{g}) \geq 1$, where *near* is taken with respect to a distributed Euclidean norm. That is, $\widetilde{f}$ and $\widetilde{g}$ have an exact, non-trivial GCRD.

Linear differential polynomials and GCRD's are key tools in finding closed form symbolic solution of systems of linear differential equations in modern computer algebra systems like Maple and Mathematica (see, e.g., [1,25]). Equations with real (floating point) coefficients or parameters are regularly encountered, and it is important to understand the stability of this fundamental tool in this case. Moreover, floating arithmetic is potentially much faster than managing large rational coefficients. We regard this paper as a positive and important initial exploration of this topic.

We commence with necessary preliminaries and well-known results that we expand upon in the remainder of this introductory section. In Sect. 2, we describe a linear algebra formulation of the approximate GCRD problem and that can be used in conjunction with truncated SVD [7,9,13] to compute nearby polynomials with an exact GCRD. Section 3 reformulates the approximate GCRD problem as a continuous unconstrained optimization problem. Sufficient conditions for existence of a solution are provided with an example showing that when this sufficient condition is not satisfied there is no solution. These results are complemented by showing that the Jacobian of the residuals has full rank and under ideal circumstances Newton iteration will converge quadratically. We generalize some results of Zeng and Dayton [31] and Zeng [30] to a non-commutative Euclidean domain showing that the problem is locally well posed. In Sect. 4 we present our algorithms explicitly, discuss their complexity and evaluate the numerical robustness of our implementation on examples of interest.

A part of this work, presenting the SVD-based approach to approximate GCRD, but without the proof of existence a nearest solution or analysis of the corresponding optimization, is presented in the workshop paper [9]. This is described in Sect. 4.1 of this current work.

1.1 Preliminaries

We review some well-known results [6,23] on differential polynomials.

The ring of differential (Ore) polynomials $\mathbb{R}(t)[\partial;']$ over the real numbers $\mathbb{R}$ provides a (non-commutative) polynomial ring structure to the linear ordinary differential operators. Differential polynomials have found great utility in symbolic computation, as they allow us to apply algebraic tools to the simplification and solution of linear differential equations; see [5] for a nice introduction to the mathematical and computational aspects.

Let $\mathbb{R}(t)[\partial;']$ be the ring of differential polynomials over the function field $\mathbb{R}(t)$. $\mathbb{R}(t)[\partial;']$ is the ring of polynomials in ∂ with coefficients from the commutative field of rational functions, under the usual polynomial addition along with the non-commutative multiplication defined by

$$\partial y(t) = y(t)\partial + y'(t) \text{ for } y(t) \in \mathbb{R}(t).$$

Here $y'(t)$ is the usual derivative of $y(t)$ with respect to t.

There is a natural action of $\mathbb{R}(t)[\partial;']$ on the space $\mathcal{C}^\infty[\mathbb{R}]$ of infinitely differentiable functions $y(t) : \mathbb{R} \to \mathbb{R}$. In particular, for any $y(t) \in \mathcal{C}^\infty[\mathbb{R}]$,

$$f(\partial) = \sum_{0\le i\le M} f_i(t)\partial^i \text{ acts on } y(t) \text{ as } \sum_{0\le i\le M} f_i(t)\frac{\mathrm{d}^i}{\mathrm{d}t^i}y(t).$$

We maintain a right canonical form for all $f \in \mathbb{R}(t)[\partial;']$ by writing

$$f = \frac{1}{f_{-1}} \sum_{0\le i\le M} f_i\partial^i, \tag{1.1}$$

for polynomials $f_{-1}, f_0, \ldots, f_M \in \mathbb{R}[t]$. That is, with coefficients in $\mathbb{R}(t)$ always written to the left of powers of ∂. An analogous left canonical form exists as well.

A primary benefit of viewing differential operators in this way is that they have the structure of a left (and right) Euclidean domain. In particular, for any two polynomials $f, g \in \mathbb{R}(t)[\partial;']$, there is a unique polynomial $h \in \mathbb{R}(t)[\partial;']$ of maximal degree in ∂ such that $f = f^*h$ and $g = g^*h$ for $f^*, g^* \in \mathbb{R}(t)[\partial;']$ (i.e., h divides f and g exactly on the right). This polynomial h is called the greatest common right divisor (GCRD) of f and g, and it is unique up to multiplication by a unit (nonzero element) of $\mathbb{R}(t)$. (We could make this GCRD have leading coefficient 1, but this would introduce denominators from $\mathbb{R}[t]$, as well as potential numerical instability, as we shall see.) An important geometric interpretation of GCRDs is that the GCRD h

of differential polynomials f and g is a differential polynomial whose solution space is the intersection of the solution spaces of f and g.

Approximations require a norm, so we need a proper definition of the *norm* of a differential polynomial.

Definition 1.1 We define the *Euclidean norm* for polynomials and a *distributed coefficient norm* for differential polynomials as follows:

1. For $p = \sum_{0\le i\le d} p_i t^i \in \mathbb{R}[t]$, define

$$\|p\| = \|p\|_2 = \left(\sum_{0\le i\le d} p_i^2\right)^{1/2}.$$

2. For $f = \sum_{0\le i\le M} f_i \partial^i \in \mathbb{R}[t][\partial;']$, define

$$\|f\| = \|f\|_2 = \left(\sum_{0\le i\le M} \|f_i\|_2^2\right)^{1/2}.$$

We could extend the above definition of norm over $\mathbb{R}(t)$ and $\mathbb{R}(t)[\partial;']$. However, it turns out that this is unnecessary and somewhat complicating. In practice, we perform most of our computations over $\mathbb{R}[t]$. In the cases where we are unable to avoid working over $\mathbb{R}(t)$, we simply solve an associate problem. This is done by clearing denominators and performing intermediate computations over $\mathbb{R}[t]$, then converting back to the representation over $\mathbb{R}(t)$. Note that the algebraic problem is always computing GCRDs and cofactors in $\mathbb{R}(t)[\partial;']$, and not the more intricate algebraic domain $\mathbb{R}[t][\partial;']$; see the discussion below.

Definition 1.2 For any matrix $S \in \mathbb{R}[t]^{(M+N)\times(M+N)}$, we define the Frobenius norm $\|S\|_F$ by

$$\|S\|_F^2 = \sum_{ij} \|S_{ij}\|^2.$$

Main problem: approximate GCRD Given $f, g \in \mathbb{R}[t][\partial;']$ such that $\operatorname{gcrd}(f, g) = 1$ we wish to compute $\widetilde{f}, \widetilde{g} \in \mathbb{R}[t][\partial;']$ with the same coefficient degree structure[1] as f and g such that $h = \operatorname{gcrd}(\widetilde{f}, \widetilde{g})$ with $D = \deg_\partial h \ge 1$ and

(i) $\|f - \widetilde{f}\|_2^2 + \|g - \widetilde{g}\|_2^2 = \varepsilon$ is minimized, and
(ii) D is the largest possible for the computed distance ε.

The differential polynomial h is said to be an *approximate GCRD* of f and g if these conditions are satisfied. In general, it is not easy to minimize ε, so instead we take a local optimization approach and compute an upper bound on this quantity.

[1] The polynomial coefficients of ∂^i have the same degree, i.e., $\deg \widetilde{f}_i \le \deg f_i$ and $\deg \widetilde{g}_i \le g_i$.

These upper-bounds will agree with the global minimum if ε is sufficiently small. The algorithmic considerations will generally assume D is fixed without loss of generality, since we can vary D from 1 to $\min\{M, N\}$ to determine the (local) optimal value.

The approximate GCRD problem is a generalization of computing an ε-GCD [7,8,19,27] in the commutative case. The requirement that the GCRD has maximal degree is difficult to certify outside the exact setting; however, this usually is not a problem in our experiments. We prove that our formulation of the approximate GCRD problem has a solution with a minimal ε (opposed to an infimum). Furthermore, if D is fixed, then for a computed pair of nearby differential polynomials, we are able to certify that ε is reasonably close to the optimal value through a condition number.

In our approach to the approximate GCRD problem, we devise methods of performing division and computing an exact GCRD numerically. These tools are used in conjunction with our algorithm for computing a nearby pair of differential polynomials with an exact GCRD via the SVD. The nearby differential polynomials with an exact GCRD are used as an initial guess in a post-refinement Newton iteration.

It will also be necessary to define a partial ordering on differential polynomials. In later sections, we will need to make use of this partial ordering to preserve structure.

Definition 1.3 Let $\overrightarrow{\deg} : \mathbb{R}[t][\partial;'] \to \mathbb{Z}^{M+1}$ be the degree vector function defined as

$$\overrightarrow{\deg}(f) = (\deg_t f_0, \deg_t f_1, \ldots, \deg_t f_M), \text{ for } f_0, \ldots, f_M \in \mathbb{R}[t].$$

For $f, g \in \mathbb{R}[t][\partial;']$ with $\deg_\partial f = \deg_\partial g = M$, we write

$$\overrightarrow{\deg}(f) < \overrightarrow{\deg}(g) \text{ if } \deg_t f_i \leq \deg_t g_i \text{ for } 0 \leq i \leq M.$$

We define $\overrightarrow{\deg}(f) = \overrightarrow{\deg}(g)$, $\overrightarrow{\deg}(f) < \overrightarrow{\deg}(g)$, $\overrightarrow{\deg}(f) \geq \overrightarrow{\deg}(g)$ and $\overrightarrow{\deg}(f) > \overrightarrow{\deg}(g)$ analogously.

We note that differential polynomials are written in a canonical ordering with highest degree coefficients appearing to the left in our examples. The degree vector function and most linearizations will appear in reverse order as a result. For convenience, we will assume that $\deg 0 = -\infty$.

Definition 1.4 Let $f \in \mathbb{R}[t][\partial;']$ where $\deg_\partial f = M$ is in standard form. The *content* of f is given by $\operatorname{cont}(f) = \gcd(f_0, f_1, \ldots, f_M)$. If $\operatorname{cont}(f) = 1$, we say that the differential polynomial is *primitive*.

Proposition 1.1 *The ring $\mathbb{R}(t)[\partial;']$ is a non-commutative principal left (and right) ideal domain. For $f, g \in \mathbb{R}(t)[\partial;']$, with $\deg_\partial f = M$ and $\deg_\partial g = N$, we have the following properties* [23].

(i) $\deg_\partial(fg) = \deg_\partial f + \deg_\partial g$, $\deg_\partial(f+g) \leq \max\{\deg_\partial f, \deg_\partial g\}$.
(ii) There exist unique $q, r \in \mathbb{R}(t)[\partial;']$ with $\deg_\partial r < \deg_\partial g$ such that $f = qg + r$ (right division with remainder).

(iii) *There exists* $h \in \mathbb{R}(t)[\partial,']$ *of maximal degree in* ∂ *with* $f = f^*h$ *and* $g = g^*h$. h *is called the GCRD (greatest common right divisor) of* f *and* g, *written* $\operatorname{gcrd}(f, g) = h$. f^* *and* g^* *are called the left cofactors of* f *and* g. *The GCRD is unique up to multiplication from a unit belonging to* $\mathbb{R}(t)$.

(iv) *There exist* $\sigma, \tau \in \mathbb{R}(t)[\partial;']$ *such that* $\sigma f = \tau g = \ell$ *for* ℓ *of minimal degree.* ℓ *is called the LCLM (least common left multiple) of* f *and* g, *written* $\operatorname{lclm}(f, g) = \ell$. *The LCLM is unique up to multiplication from a unit belonging to* $\mathbb{R}(t)$.

(v) $\deg_\partial \operatorname{lclm}(f, g) = \deg_\partial f + \deg_\partial g - \deg_\partial \operatorname{gcrd}(f, g)$.

In an algebraic context, we can clear denominators of our inputs and assume without loss of generality that our GCRD belongs to $\mathbb{R}[t][\partial;']$. We will also assume our inputs and output are primitive. Again, this is not algebraically necessary but will be important for the convergence of our subsequent optimization formulation (see Sect. 3.2). It is important to note that the cofactors of the GCRD need not belong to $\mathbb{R}[t][\partial;']$ even if we have $f, g, h \in \mathbb{R}[t][\partial;']$ such that $\operatorname{gcrd}(f, g) = h$. This is not unexpected, as a similar situation occurs when computing GCD's over $\mathbb{Z}[x]$, where cofactors in the GCD of primitive polynomials may well lie in $\mathbb{Q}[x] \setminus \mathbb{Z}[x]$. In essence, this is a computational technique to narrow the input domain, not a change to the problem being considered.

A related but considerably more difficult problem is computing ideal bases and factorizations completely within $\mathbb{R}[t][\partial;']$. This has been dealt with algebraically and in terms of exact computation by a number of authors, though not with respect to approximate coefficients; see for example [2,11,14].

Most of our results involve transforming a representation of $f \in \mathbb{R}(t)[\partial;']$ into a representation over $\mathbb{R}(t)^{1\times K}$ for $K \geq \deg_\partial f$. We make extensive use of the following map.

Definition 1.5 For $f \in \mathbb{R}(t)[\partial;']$ of degree M in ∂ as in (1.1), and $K > M$, we define

$$\Psi_K(f) = \frac{1}{f_{-1}}(f_0, f_1, \ldots, f_M, 0, \ldots, 0) \in \mathbb{R}(t)^{1\times K}.$$

That is, Ψ_K maps polynomials in $\mathbb{R}(t)[\partial;']$ of degree (in ∂) less than K into $\mathbb{R}(t)^{1\times K}$.

It will be useful to linearize (differential) polynomials, that is, express them as an element of Euclidean space. For $p \in \mathbb{R}[t]$ with $\deg_t p = d$, we write

$$\mathbf{p} = (p_0, p_1, \ldots, p_d) \in \mathbb{R}^{1\times(d+1)}$$

For $f = f_0 + f_1\partial + \cdots f_M\partial^M \in \mathbb{R}[t][\partial;']$ with $\deg_\partial f = M$ and $\deg_t f_i = d_i$, we write

$$\mathbf{f} = (\mathbf{f_0}, \ldots, \mathbf{f_M}) \in \mathbb{R}^{1\times L},$$

where $L = (d_0 + 1) + \cdots + (d_M + 1)$. If $d \geq \max\{d_i\}$ we will sometimes pad each $\mathbf{f_i}$ with zeros to have precisely $d + 1$ coefficients, and by a slight abuse of notation regard

$$\mathbf{f} \in \mathbb{R}^{1\times(M+1)(d+1)}.$$

We will not do this unless specifically stated.

2 Computing the GCRD via Linear Algebra

In this section, we demonstrate how to reduce the computation of the GCRD to that of linear algebra over $\mathbb{R}(t)$, and then over $\mathbb{R}$ itself. This approach has been used in the exact computation of GCRDs [22] and differential Hermite forms [10], and has the benefit of reducing differential, and more general Ore problems, to a system of equations over a commutative field. Here, we will show that it makes our approximate version of the GCRD problem amenable to numerical techniques. We note that for computing approximate GCRDs of differential polynomials, much as for computing approximate GCDs of standard commutative polynomials, the Euclidean algorithm is numerically unstable, and thus, we employ resultant-based techniques, as described below.

Since $\mathbb{R}(t)[\partial;']$ is a right (and left) Euclidean domain [23], a GCRD may be computed by solving a Diophantine equation corresponding to the Bézout coefficients. Using the subresultant techniques of Li [21], we are able to transform the non-commutative problem over $\mathbb{R}(t)[\partial;']$ into a commutative linear algebra problem over $\mathbb{R}(t)$. This is done through a Sylvester-like resultant matrix. By using resultant-like matrices, we are able to express the Bézout coefficients as a linear system over $\mathbb{R}(t)$ and compute a GCRD via nullspace basis computation.

Lemma 2.1 *Suppose* $f, g \in \mathbb{R}(t)[\partial;']$ *with* $\deg_\partial f = N$ *and* $\deg_\partial g = M$. *Then,* $\deg_\partial \operatorname{gcrd}(f, g) \geq 1$ *if and only if there exist* $u, v \in \mathbb{R}(t)[\partial;']$ *such that* $\deg_\partial u < M$, $\deg_\partial v < N$, *and* $uf + vg = 0$.

Proof This follows immediately from Proposition 1.1. □

Using Lemma 2.1, we can solve a Bézout-like system to compute a GCRD of two differential polynomials. This is characterized by the differential Sylvester matrix, based on the subresultant method of Li and Nemes [22].

Definition 2.1 Suppose $h \in \mathbb{R}[t][\partial;']$ has $\deg_\partial = D$. For any $K \in \mathbb{N}$, the matrix

$$\mathcal{C}_K^{\mathbf{R}}(h) = \begin{pmatrix} \Psi_{K+D+1}(h) \\ \Psi_{K+D+1}(\partial h) \\ \vdots \\ \Psi_{K+D+1}(\partial^K h) \end{pmatrix} \in \mathbb{R}[t]^{(K+1)\times(K+D+1)}$$

is the Kth *right differential convolution matrix* of h. We note that the entries of $\mathcal{C}_K^{\mathbf{R}}(h)$ are written in their *right canonical form*, where the ∂'s appear to the right (polynomials in $\mathbb{R}[t]$ appear to the left). We note that $\deg_t \partial^i h = \deg_t h$, so the degree in t of all entries of $\mathcal{C}_K^{\mathbf{R}}(h)$ is at most $\deg_t h$.

We analogously define the Kth *left differential convolution matrix* of h as $\mathcal{C}_K^{\mathbf{L}}(h)$ as

$$\mathcal{C}_K^{\mathbf{L}}(h) = \begin{pmatrix} \Psi_{K+D+1}(h) \\ \Psi_{K+D+1}(h\partial) \\ \vdots \\ \Psi_{K+D+1}(h\partial^K) \end{pmatrix} \in \mathbb{R}[t]^{(K+1)\times(K+D+1)},$$

where elements are written in their *left canonical form*, where the ∂'s appear to the left (polynomials in $\mathbb{R}[t]$ always appear to the right).

Both right and left differential convolution matrices can be used to perform multiplication. Suppose $f^* \in \mathbb{R}(t)[\partial;']$, $h \in \mathbb{R}(t)[\partial;']$ and $f = f^*h \in \mathbb{R}[t][\partial;']$, with

$$f = \sum_{0\le i\le M} f_i\partial^i,\ f^* = \sum_{0\le i\le M-D} f_i^*\partial^i \text{ and } h = \sum_{0\le i\le D} h_i\partial^i, \tag{2.1}$$

with $f_i, h_i \in \mathbb{R}[t]$ and $f_i^* \in \mathbb{R}(t)$. We can express the product of f^* and h as

$$(f_0, f_1, \ldots, f_M) = (f_0^*, \ldots, f_{M-D}^*)\mathcal{C}_{M-D}^{\mathbf{R}}(h).$$

Similarly, we may write

$$(f_0, f_1, \ldots, f_M)^T = \mathcal{C}_D^{\mathbf{L}}(f^*)(h_0, h_1, \ldots, h_D)^T.$$

In keeping with our canonical ordering, we express our results in terms of right differential convolution matrices. We carefully observe that both the right and left differential convolution matrices described correspond to *right multiplication*. Left multiplication can be formulated in a similar manner.

Let $f, g \in \mathbb{R}(t)[\partial;']$ with $\deg_\partial f = M$ and $\deg_\partial g = N$. Then by Lemma 2.1 we have that $\deg_\partial \operatorname{gcrd}(f, g) \ge 1$ if and only if there exist $u, v \in \mathbb{R}(t)[\partial;']$ such that $\deg_\partial u < N$, $\deg_\partial v < M$ and $uf + vg = 0$. We can encode the existence of u, v as an $(M + N) \times (M + N)$ matrix over $\mathbb{R}(t)$ in what we will call the differential Sylvester matrix.

Definition 2.2 The matrix

$$S = S(f, g) = \begin{pmatrix} \mathcal{C}_{N-1}^{\mathbf{R}}(f) \\ \mathcal{C}_{M-1}^{\mathbf{R}}(g) \end{pmatrix} \in \mathbb{R}(t)^{(M+N)\times(M+N)}$$

is the differential Sylvester matrix of f and g.

This matrix [22] is analogous to the Sylvester matrix of real polynomials; see [29, Chapter 6]. As expected, many useful properties of the Sylvester matrix over real polynomials still hold with the differential Sylvester matrix. These similarities become evident when we consider

$$w = (u_0, u_1, \ldots, u_{N-1}, v_0, v_1, \ldots, v_{M-1}) \in \mathbb{R}(t)^{1\times(M+N)}.$$

Then, $uf + vg = 0$ implies that $wS = 0$; hence, w is a non-trivial vector in the (left) nullspace of S. In particular, this solution is equivalent to saying that S is singular. Clearing denominators of f and g, we may assume that $u, v \in \mathbb{R}[t][\partial;']$, i.e., they have polynomial coefficients, which implies that $S \in \mathbb{R}[t]^{(M+N)\times(M+N)}$. Moreover, for $f, g \in \mathbb{R}[t][\partial;']$ with $\deg_t f \le d$ and $\deg_t g \le d$ then $\deg_t S_{ij} \le d$.

We summarize these results in the following lemma.

Lemma 2.2 *Suppose* $f, g \in \mathbb{R}[t][\partial;']$, *where* $\deg_\partial f = M$, $\deg_\partial g = N$, $\deg_t f \le d$ *and* $\deg_t g \le d$.

(i) $S = S(f, g)$ *is singular if and only* $\deg_\partial \operatorname{gcrd}(f, g) \ge 1$.
(ii) $\deg_\partial \operatorname{gcrd}(f, g) = \dim \operatorname{null}_\cdot(S)$, *where* $\operatorname{null}_\cdot(S)$ *is the left nullspace of* S.
(iii) *For any* $w = (u_0, \ldots, u_{N-1}, v_0, \ldots, v_{M-1}) \in \mathbb{R}(t)^{1\times(M+N)}$ *such that* $wS = 0$, *we have* $uf + vg = 0$, *where* $u = \sum_{0\le i<N} u_i \partial^i$ *and* $v = \sum_{0\le i<M} v_i \partial^i$.
(iv) *Suppose that* $\deg_\partial \operatorname{gcrd}(f, g) \ge 1$. *Then there exists* $w \in \mathbb{R}[t]^{1\times(M+N)}$ *such that* $wS = 0$ *and* $\deg_t w \le \mu = 2(M + N)d$.

Proof Parts (i)–(iii) follow from Lemma 2.1 and the discussion above. Part (iv) follows from an application of Cramer's rule and a bound on the degree of the determinants of a polynomial matrix. □

2.1 Linear Algebra over $\mathbb{R}$

Let $S \in \mathbb{R}[t]^{(M+N)\times(M+N)}$ be the differential Sylvester matrix of $f, g \in \mathbb{R}[t][\partial;']$ of degrees M and N, respectively, in ∂, and degrees at most d in t. From Lemma 2.2, we know that if a GCRD of f and g exists, then there is a $w \in \mathbb{R}[t]^{1\times(M+N)}$ such that $wS = 0$, with $\deg_t w \le \mu = 2(M + N)d$.

Definition 2.3 The kth convolution matrix of $b \in \mathbb{R}[t]$ with $\deg b = m$ is defined as

$$C_k(b) = \begin{pmatrix} b_0 & & \\ b_1 & \ddots & \\ \vdots & \ddots & b_0 \\ b_m & & b_1 \\ & \ddots & \vdots \\ & & b_m \end{pmatrix} \in \mathbb{R}^{(m+k+1)\times(k+1)}.$$

Let $a \in \mathbb{R}[t]$ with $\deg a = \mu$ and define the mapping $\Gamma : \mathbb{R}[t] \to \mathbb{R}^{(\mu+1)\times(\mu+d+1)}$ by $\Gamma(a) = C_d(a)^T$. $\Gamma(a)$ is the left multiplier matrix of a with respect to the basis $\langle 1, t, \ldots, t^{\mu+d}\rangle$.

A differential convolution matrix generalizes the convolution matrix in the role of linearizing multiplication between differential polynomials.

Definition 2.4 Given the $(M + N) \times (M + N)$ differential Sylvester matrix S, we apply Γ entry-wise to S to obtain $\widehat{S} \in \mathbb{R}^{(M+N)(\mu+1)\times(M+N)(\mu+d+1)}$; each entry of

S in $\mathbb{R}[t]$ is mapped to a block entry in $\mathbb{R}^{(\mu+1)\times(\mu+d+1)}$ of $\widehat{S}$. We refer to $\widehat{S}$ as the *inflated differential Sylvester matrix* of f and g.

Lemma 2.3 *Let $f, g \in \mathbb{R}[t][\partial;']$ have differential Sylvester matrix $S \in \mathbb{R}[t]^{(M+N)\times(M+N)}$ and inflated differential Sylvester matrix*

$$\widehat{S} \in \mathbb{R}^{(M+N)(\mu+1)\times(M+N)(\mu+d+1)}.$$

There exists a $w \in \mathbb{R}[t]^{1\times(M+N)}$ such that $wS = 0$, if and only if there exists a $\widehat{w} \in \mathbb{R}^{(\mu+d+1)\times(M+N)(\mu+1)}$ such that $\widehat{w}\widehat{S} = 0$. More generally,

$$\deg_\partial \operatorname{gcrd}(f, g) = \frac{\dim \operatorname{null}\cdot(\widehat{S})}{\mu + d + 1}.$$

Proof This follows directly from the definition of Γ and Lemma 2.2. □

We note that $\widehat{S}$ is no longer a square matrix. This will not pose too many problems as we will see in the following sections.

2.2 Division Without Remainder

While multiplication of differential polynomials with approximate numerical coefficients is straightforward, division is somewhat more difficult. We will generally require a division *without remainder*, for the computation of which we use a least squares approach. Given $f, h \in \mathbb{R}[t][\partial;']$ as in (2.1), we wish to find an $f^* \in \mathbb{R}(t)[\partial;']$ such that $\|f - f^*h\|$ is minimized. We will assume as usual that $\deg_\partial f = M$, $\deg_\partial h = D$ and $\deg_t f, \deg_t h \le d$.

Much as in the (approximate polynomial) commutative case, we do this by setting the problem up as a linear system and then finding a least squares solution. Let us assume for now that $f = f^*h$ is exact, so this can be expressed as a linear system over $\mathbb{R}(t)$ by writing

$$(f_0, f_1, \ldots, f_M) = (f_0^*, \ldots, f_{M-D}^*)\mathcal{C}^{\mathbf{R}}_{M-D}(h). \tag{2.2}$$

This system of equations is over-constrained (over $\mathbb{R}(t)$), but we note that the submatrix formed from the last $M - D + 1$ columns of $\mathcal{C}^{\mathbf{R}}_{M-D}(h)$ is lower triangular, with diagonal entry $h_D \in \mathbb{R}[t]$. Thus, any exact quotient $h \in \mathbb{R}[t][\partial;']$ such that $f = f^*h$, in lowest terms, must have denominators dividing h_D^{M-D+1}, and in particular have denominators of degree at most $(M - D + 1)\deg_t h_D \le (M - D + 1)d$. Equivalently, $h_D^{M-D+1} f^* \in \mathbb{R}[t]^{M-D+1}$. By applying Cramer's rule on the last $M - D + 1$ columns of $\mathcal{C}^{\mathbf{R}}_{M-D}(h)$, the degrees of the numerators in f^* must be at most $(M - D + 1)d$. Using this information, we can formulate an associated problem with coefficients from $\mathbb{R}[t]$ and avoid performing linear algebra over $\mathbb{R}(t)$.

Now let $v_{-1}, v_0, \ldots, v_{M-D}$ be generic polynomials in t, with indeterminate coefficients of degree at most $(M - D + 1)d$, i.e.,

$$v_i = \sum_{j=0}^{(M-D+1)d} v_{ij} t^j, \quad i = -1 \ldots M - D,$$

for indeterminates v_{ij} with $v_{-1} \neq 0$. Then, we are seeking to solve the linear system of equations

$$v_{-1} \cdot (f_0, \ldots, f_M) = (v_0, \ldots, v_{M-D})\, \mathcal{C}^{\mathbf{R}}_{M-D}(h)$$

for the v_{ij}. For each entry f_i, we have $(M-D+1)d+d+1$ equations; this is the degree $(v_0, \ldots, v_{M-D})\mathcal{C}^{\mathbf{R}}_{M-D}(h)$ plus one, and we get one equation per coefficient. Hence, there are $(M + 1)((M - D + 1)d + d + 1)$ equations in $(M - D + 2)(M - D + 1)d$ unknowns. We then use a standard linear least squares solution to find the v_i which minimizes the residual, and thus minimizes $\|f - f^*h\|$.

It may be desirable to find the lowest degree v_{-1} which meets this criteria, for which we can use a simple binary search for a lower degree with reasonable residual (or alternatively use an SVD-based identification procedure).

Finally, a more straightforward approach to solving (2.2) is to simply use the solution from the last $M - D + 1$ columns of $\mathcal{C}^{\mathbf{R}}_{M-D}(h)$. The last $M - D + 1$ columns of $\mathcal{C}^{\mathbf{R}}_{M-D}(h)$ are lower triangular, with diagonal entries consisting of $h_D \in \mathbb{R}[t]$. While this does not yield a solution to the least squares normal equations, it is usually sufficiently good in practice, and considerably easier to formulate.

3 Optimization-Based Formulation of Approximate GCRD

First, we standardize some notation and assumptions. We assume that $f, g, \widetilde{f}, \widetilde{g}, h \in \mathbb{R}[t][\partial;']$ and $f^*, g^* \in \mathbb{R}(t)[\partial;']$. Moreover, we assume that $\widetilde{f} = f^*h$ and $\widetilde{g} = g^*h$ and $h = \operatorname{gcrd}(\widetilde{f}, \widetilde{g})$. Intuitively, f, g are our "input polynomials" and we will be identifying "nearby" $\widetilde{f}, \widetilde{g}$ with a non-trivial GCRD h. Note that f^*, g^* have rational function coefficients. Later, we will find it useful to clear fractions and work with a primitive associate.

We also assume degree bounds as follows: $\deg_\partial f = \deg_\partial \widetilde{f} = M$, $\deg_t f, \deg_t \widetilde{f} \leq d$, $\deg_\partial g, \deg_\partial \widetilde{g} = N$, $\deg_t g, \deg_t \widetilde{g} \leq d$, $\deg_\partial h = D$, $\deg_\partial f^* = M - D$ and $\deg_\partial g^* = N - D$.

Using the method of Giesbrecht and Haraldson [9], essentially the generalization of the SVD-based method of Corless et al. [7] to differential polynomials, we will make an initial guess for $\widetilde{f}, \widetilde{g}$; details are described in Sect. 4.1 of this paper. We then use optimization techniques to hone in on polynomials with minimal distance. While the techniques in that paper are not particularly effective at providing a nearest solution, they do provide a suitable initial guess, which we employ here.

We next describe how to formulate an objective function Φ that, when minimized, corresponds to a solution to the approximate GCRD problem. Define the objective function $\Phi : \mathbb{R}[t][\partial;'] \times \mathbb{R}(t)[\partial;']^2 \to \mathbb{R}$ as

$$\Phi(h, f^*, g^*) = \|f - f^*h\|_2^2 + \|g - g^*h\|_2^2.$$

In keeping up with our notation from earlier, we observe that $\widetilde{f} = f^*h$ and $\widetilde{g} = g^*h$ in the context of the objective function Φ, as f and g will typically be relatively prime. To compute guesses for the cofactors given h, we will perform an approximate division without remainder using the method of Sect. 2.2. We only require an initial guess for f^* and g^* to minimize Φ, so this factorization doesn't need to be exact, in the event that $\gcrd(f, g) = h$.

We show that Φ has an attainable global minimum under appropriate assumptions. More precisely, there exist non-trivial $\widetilde{f}$ and $\widetilde{g}$ such that

$$\|f - \widetilde{f}\|_2^2 + \|g - \widetilde{g}\|_2^2 \tag{3.1}$$

is minimized. Furthermore, we will show that the approximate GCRD problem is locally well posed.

3.1 Existence of Solutions

Lemma 3.1 *Let $f, h \in \mathbb{R}[t][\partial;']$, with monic leading coefficients, be not necessarily primitive, such that $f = f^*h$ for $f^* \in \mathbb{R}[t][\partial;']$ with $\deg_\partial f = M$ and $\deg_\partial h = D$. Then, $\|f^*\|$ is bounded above.*

Proof It follows that f^* is bounded by the computing the Cramer solution to (2.2) using the last $M - D + 1$ columns of $\mathcal{C}^{\mathbf{R}}_{M-D}(h)$. □

As an observation, we relax the assumption that f is primitive (we work with an associate instead) in order to guarantee that $f^* \in \mathbb{R}[t][\partial;']$. This can be taken without loss of generality as the quantity $\|\cont(f)\|_2^2$ is bounded above and away from zero (as its leading coefficient is monic). Thus, we may divide by it without affecting the quality of the results, as $\|f - f^*h\|$ is still well defined.

We will make use of the following well-known fact from ([24], Theorem 4.16).

Fact 3.1 *Suppose that Φ is a continuous real function on a compact metric space X. Then, there exist points p and q in X such that*

$$\Phi(p) \leq \Phi(x) \leq \Phi(q),$$

for all $x \in X$. Precisely, Φ attains its minimum and maximum values at p and q, respectively.

We first state a general version of the theorem where a logical predicate $\Xi : \mathbb{R}^k \to \{\text{true}, \text{false}\}$ (for some k) can be chosen to impose additional constraints on the problem. For the rest of this section, let

$$\phi : \mathbb{R}[t][\partial;']^2 \to \mathbb{R}^{(M+N+2)(d+1)}$$

be the combined coefficient vector function, i.e., for arbitrary $f, g \in \mathbb{R}[t][\partial;']$ we write $\phi(f, g) = (\mathbf{f}, \mathbf{g})$, where $\mathbf{f}$ and $\mathbf{g}$ are padded with zeros to have the desired dimensions.

The following lemma and its proof are analogous to [17, Theorem 2], which in turn generalizes the univariate argument of [18, Theorem 1].

Theorem 3.1 (Existence of Global Minima) *Let $f, g \in \mathbb{R}[t][\partial;']\backslash\{0\}$, let $d = \max\{\deg_t f, \deg_t g\}$, $\deg_\partial f = M$, $\deg_\partial g = N$ and $D \leq \min\{M, N\}$. Furthermore, let $\Xi : \mathbb{R}^{(M+N+2)(d+1)} \to \{\text{true}, \text{false}\}$ be a predicate on $\phi(f, g)$. We assume that the preimage $\Xi^{-1}(\text{true})$ is a topologically closed set in $\mathbb{R}^{(M+N+2)(d+1)}$ with respect to the Euclidean norm. For a given $\Omega \in \mathbb{R}_{>0}$, we define the set of possible solutions by*

$$\mathcal{F}_\Omega = \left\{ (\widetilde{f}, \widetilde{g}) \in \mathbb{R}[t][\partial;']^2 \text{ such that } \begin{array}{c} \deg_\partial \widetilde{f} = M, \\ \deg_\partial \widetilde{g} = N, \\ \deg_\partial \widetilde{h} \geq D, \\ \widetilde{h} = \operatorname{gcrd}(\widetilde{f}, \widetilde{g}), \\ \|\widetilde{h}\| \leq \Omega, \\ \operatorname{lcoeff}_t(\operatorname{lcoeff}_\partial \widetilde{h}) = 1, \\ \text{and } \Xi(\phi(\widetilde{f}, \widetilde{g})) = \text{true} \end{array} \right\}.$$

Suppose that $\mathcal{F}_\Omega \neq \emptyset$. Then, the minimization problem

$$\min_{(\widetilde{f}, \widetilde{g}) \in \mathcal{F}_\Omega} \|f - \widetilde{f}\|_2^2 + \|g - \widetilde{g}\|_2^2 \tag{3.2}$$

has an attainable global minimum.

Proof Without loss of generality, we assume that $M \leq N$. Then, we iterate the minimization over all $\ell \in \mathbb{Z}_{\geq 0}$ such that $D \leq \ell \leq M$ and coefficients $\rho \subset \mathbb{R}[t]^\ell$. Let $\mathcal{H}_{\ell,\rho}$ denote the set of all differential polynomials over $\mathbb{R}[t][\partial;']$ of degree ℓ with coefficients from ρ.

We optimize over the continuous real objective function

$$\Phi(h, f^*, g^*) = \|f - f^* h\|_2^2 + \|g - g^* h\|_2^2,$$

for $h \in \mathcal{H}_{\ell,\rho}$, $\deg_\partial f^* \leq M - D$ and $\deg_\partial g^* \leq N - D$. We fix the leading coefficient of h with respect to ∂ to be monic, that is $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h = 1$.

Since the leading coefficient of h is monic, we can write $G = \operatorname{gcrd}(f^* h, g^* h)$ with $\deg_\partial G \geq D$. Since G is a multiple of h, we normalize G so that $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial G = 1$, i.e., the leading coefficient of G is also monic. The restriction on h that the leading coefficient of h is monic enforces that $\deg_\partial G \geq D$. Furthermore, we restrict the domain of our function Φ to those h, f^* and g^* for which $(f^* h, g^* h) \in \mathcal{F}_\Omega$.

If there is no such common factor h and cofactors f^* and g^*, then this pair of ℓ and ρ does not occur in minimization (3.2). By assumption we have that $\mathcal{F}_\Omega \neq \emptyset$, so there must be at least one possible case. We note that if $(0, 0) \in \mathcal{F}_\Omega$, then $f^* = g^* = 0$.

Now suppose that for the given ℓ and ρ, there are $\widetilde{h} \in \mathcal{H}_{\ell,\rho}$ and $\widetilde{f}^*, \widetilde{g}^*$ satisfying $\deg_\partial \widetilde{f}^* \leq M - \ell$ and $\deg_\partial \widetilde{g}^* \leq N - \ell$ such that $(\widetilde{f}^*\widetilde{h}, \widetilde{g}^*\widetilde{h}) \in \mathcal{F}_\Omega$. We shall prove that the function Φ has a value on a closed and bounded set (i.e., compact with respect to the Euclidean metric) that is smaller than elsewhere. Hence, Φ attains a global minimum by Fact 3.1.

Clearly, any solution $\widetilde{h} \in \mathcal{H}_{\ell,\rho}$ and $\widetilde{f}^*, \widetilde{g}^*$ with $(\widetilde{f}^*h, \widetilde{g}^*h) \in \mathcal{F}_\Omega$ but with $\Phi(\widetilde{h}, \widetilde{f}^*, \widetilde{g}^*) > \Phi(h, f^*, g^*)$ can be discarded. So the norm of the products $\|\widetilde{f}^*\widetilde{h}\|_2$ and $\|\widetilde{g}^*\widetilde{h}\|_2$ can be bounded from above. We have that $\|\widetilde{h}\|$ is bounded above by Lemma 3.1 because it is a right factor of $\widetilde{G} = \operatorname{gcrd}(\widetilde{f}^*\widetilde{h}, \widetilde{g}^*\widetilde{h})$ with $\|\widetilde{G}\| \leq \Omega$. We note that $\widetilde{h}$ has a monic leading coefficient, so $\|\widetilde{h}\| \geq 1$. We have that $\|\widetilde{f}^*\|$ and $\|\widetilde{g}^*\|$ (or the appropriate associate) are both bounded above by Lemma 3.1.

Thus, we can restrict the domain of Φ to values that lie within a sufficiently large closed ball B.

The function ζ that maps (h, f^*, g^*) to the combined coefficient vector $\phi(f^*h, g^*h)$ of f^*h and g^*h is continuous. We minimize over $\zeta^{-1}(\Xi^{-1}(\text{true}) \cap \zeta(B))$, which is a compact set. □

For the less general version of the theorem, given arbitrary $f, g \in \mathbb{R}[t][\partial;']$, we define

$$\mathcal{S} = \mathcal{S}(f,g) = \left\{ \phi(\widetilde{f}, \widetilde{g}) \mid \widetilde{f}, \widetilde{g} \in \mathbb{R}[t][\partial;'] \text{ such that } \begin{array}{c} \overrightarrow{\deg}(\widetilde{f}) \leq \overrightarrow{\deg}(f) \\ \overrightarrow{\deg}(\widetilde{g}) \leq \overrightarrow{\deg}(g) \end{array} \right\}.$$

We observe that $\mathcal{S}$ is a closed subset of $\mathbb{R}^{(M+N+2)(d+1)}$, where $\deg_\partial f = M$, $\deg_\partial g = N$ and $d = \max\{\deg_t f, \deg_t g\}$. The set $\mathcal{S}$ corresponds to the combined coefficient vectors of $\widetilde{f}$ and $\widetilde{g}$ that have the same degree structure as f and g.

Corollary 3.1 *Let $f, g \in \mathbb{R}[t][\partial;']\backslash\{0\}$, let $d = \max\{\deg_t f, \deg_t g\}$, $\deg_\partial f = M$, $\deg_\partial g = N$ and $D \leq \min\{M, N\}$. For a given $\Omega \in \mathbb{R}_{>0}$, we define the set of possible solutions by*

$$\mathcal{F}_\Omega = \left\{ (\widetilde{f}, \widetilde{g}) \in \mathbb{R}[t][\partial;'] \times \mathbb{R}[t][\partial;'] \text{ such that } \begin{array}{r} \deg_\partial \widetilde{f} = M, \\ \deg_\partial \widetilde{g} = N, \\ \phi(\widetilde{f}, \widetilde{g}) \in \mathcal{S}, \\ \deg_\partial \widetilde{h} \geq D, \\ \widetilde{h} = \operatorname{gcrd}(\widetilde{f}, \widetilde{g}), \\ \|\widetilde{h}\| \leq \Omega, \\ \text{and } \operatorname{lcoeff}_t(\operatorname{lcoeff}_\partial(\widetilde{h})) = 1 \end{array} \right\}.$$

Suppose that $\mathcal{F}_\Omega \neq \emptyset$. Then, the minimization problem

$$\min_{(\widetilde{f},\widetilde{g}) \in \mathcal{F}_\Omega} \|f - \widetilde{f}\|_2^2 + \|g - \widetilde{g}\|_2^2$$

has an attainable global minimum.

We note that Theorem 3.1 does not guarantee a unique minimum of Φ, merely that Φ has an attainable minimum (as opposed to an infimum). The choice of h, f^* and g^* that we optimize over is important. If $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h$ vanishes or $\|h_0\|, \ldots, \|h_{D-1}\|$ are quite large, then f^* and g^* can be ill-conditioned in the approximate GCRD problem. Furthermore, choosing overly large, small or poor degree structure in t for h can result in a Φ that cannot be minimized for the specified structure, but would otherwise have a minimum for a different choice of h.

Example 3.1 Consider $f = \partial^2 - 2\partial + 1$ and $g = \partial^2 + 2\partial + 2$ (see [17] for an example with complex perturbations). Then, f and g do not have a degree 1 approximate GCRD. That is, we show that there does not exist $\widetilde{f}, \widetilde{g} \in \mathbb{R}[t][\partial;']$ where $\deg_\partial \operatorname{gcrd}(\widetilde{f}, \widetilde{g}) = 1$ and $\|f - \widetilde{f}\|_2^2 + \|g - \widetilde{g}\|_2^2$ is minimized.

The real monic Karmakar–Lakshman distance [19,20] of

$$\|f - \widetilde{f}\|_2^2 + \|g - \widetilde{g}\|_2^2$$

occurs when the rational function

$$\frac{2h_0^4 + 14h_0^2 + 4h_0 + 5}{h_0^4 + h_0^2 + 1}$$

is minimized for $h_0 \in \mathbb{R}$. The minimum value (if it exists) of this function corresponds to the approximate GCRD $h = \partial - h_0$.

The infimum is 2, which is unattainable. There is no attainable global minimum.

The non-monic real Karmakar–Lakshman distance is 2, which is achieved if and only if the leading coefficient vanishes. The minimum occurs when the rational function

$$\frac{5h_1^4 - 4h_1^3 + 14h_1^2 + 2}{h_1^4 + h_1^2 + 1}$$

is minimized. The minimum value of this function corresponds to the approximate GCRD $h = h_1\partial + 1$.

In particular, if we consider $\widetilde{f} = (-2\partial + 1)(\varepsilon\partial + 1)$ and $\widetilde{g} = (2\partial + 2)(\varepsilon\partial + 1)$, then $\|f - \widetilde{f}\|_2^2 + \|g - \widetilde{g}\|_2^2$ becomes arbitrarily near 2 as $\varepsilon \to 0$.

There is no real degree 1 approximate GCRD, as

$$\min_{\{(\widetilde{f},\widetilde{g}) \in \mathbb{R}[t][\partial;']^2 \mid \deg_\partial \operatorname{gcrd}(\widetilde{f},\widetilde{g}) = 1\}} \|f - \widetilde{f}\|_2^2 + \|g - \widetilde{g}\|_2^2$$

is not defined in the monic case. In the non-monic case, if a minimum exists then it occurs when $\operatorname{lcoeff}_\partial h$ vanishes, so the minimum value is not defined either.

This example illustrates that not all $f, g \in \mathbb{R}[t][\partial;']$ have an approximate GCRD. Furthermore, we see that the requirement that $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h = 1$ and $\|h\|$ is bounded, from Theorem 3.1 are required, even if there are no additional constraints imposed.

Now it remains to show that it is possible to obtain a (locally) unique solution to Φ. One of many equivalent conditions for uniqueness of an exact GCRD is to require it to be primitive and have a monic leading coefficient. Numerically, to obtain a unique solution of the approximate GCRD problem, we impose the same constraints, making solutions locally unique.

3.2 Convergence of Newton Iteration and Conditioning

From Theorem 3.1 and Corollary 3.1, we know a solution to the approximate GCRD problem exists. We now show that a standard Newton iteration will converge quadratically when starting with an estimate sufficiently close to an approximate GCRD. We first describe the Jacobian of the residuals and show that the Jacobian has full rank. This leads to a first-order approximation of the Hessian matrix showing that it is locally positive definite around a global minimum when the residual is sufficiently small. The implication is that Newton's method will converge quadratically [4]. If we consider structured perturbations, then we are able to obtain results similar to that of Zeng and Dayton [31] to the overall conditioning of the system.

In this section, we assume without loss of generality that $f^*, g^* \in \mathbb{R}[t][\partial;']$ are primitive, and that f and g may no longer be primitive to simplify computations. We need to clear fractions of rational functions to apply our coefficient norms, and to linearize h, f^* and g^* as vectors of real numbers.

The residual of the approximate GCRD is

$$\begin{aligned} r &= r(h, f^*, g^*) \\ &= (\mathbf{f}^*\mathbf{h} - \mathbf{f}, \mathbf{g}^*\mathbf{h} - \mathbf{g})^T \in \mathbb{R}^{\eta\times 1}, \end{aligned}$$

where

$$\begin{aligned} \eta &= \sum_{0\le i\le M} \max\{\deg_t f_i, -1\} + \sum_{0\le i\le N} \max\{\deg_t g_i, -1\} + (M+1) + (N+1) \\ &\le (M+N+2)(d+1). \end{aligned}$$

Intuitively, η represents the number of components of $(\mathbf{f}, \mathbf{g}) \in \mathbb{R}^{1\times\eta}$. Let ν be the number of variables needed to represent the coefficients of h, f^* and g^*, i.e., $(\mathbf{h}, \mathbf{f}^*, \mathbf{g}^*) \in \mathbb{R}^{1\times\nu}$.

Recall that when $f = f^*h$, we can linearize this relationship with differential convolution matrices, by writing

$$f = (f_0^*, \ldots, f_{M-D}^*)\mathcal{C}_{M-D}^{\mathbf{R}}(h).$$

If f_i is a coefficient of f with $\deg_t f_i = d$, then we may write

$$f_i = \sum_{0\le j\le M-D} f_j^*(\mathcal{C}_{M-D}^{\mathbf{R}}(h)[j, i]).$$

This relationship may be linearized over $\mathbb{R}$ through the use of convolution matrices. Writing

$$\mathbf{f_i} = \sum_{0 \le j \le M-D} \mathbf{f_j^*} \cdot C_d \left(\mathcal{C}^{\mathbf{R}}_{M-D}(h)[j, i] \right)^T,$$

we now have a direct method of computing $\mathbf{f_i}$ in terms of the coefficients of f^* and h.

If we differentiate $\mathbf{f}^*\mathbf{h}$ with respect to an entry from $\mathbf{f}^*$, then we will obtain the corresponding (linearized) row of $\mathcal{C}^{\mathbf{R}}_{M-D}(h)$. Similarly, differentiating $\mathbf{f}^*\mathbf{h}$ with respect to an entry of $\mathbf{h}$ will give us a (linearized) column of $\mathcal{C}^{\mathbf{L}}_{D}(f^*)$. This relationship becomes clear when we observe that

$$(f_0^*, \ldots, f_{M-D}^*)\mathcal{C}^{\mathbf{R}}_{M-D}(h) = \left(\mathcal{C}^{\mathbf{L}}_{D}(f^*) \begin{pmatrix} h_0 \\ \vdots \\ h_D \end{pmatrix} \right)^{\mathrm{T}}.$$

Differentiating $\mathbf{g}^*\mathbf{h}$ with respect to variables from $\mathbf{g}^*$ and $\mathbf{h}$ will produce similar results.

The Jacobian of $r(h, f^*, g^*)$ for arbitrary h, f^* and g^* may be expressed (up to column permutation) in block matrix form as

$$J = \begin{pmatrix} \mathcal{C}^{\mathbf{R}}_{M-D}(h)^{\mathrm{T}} & 0 & \mathcal{C}^{\mathbf{L}}_{D}(f^*) \\ 0 & \mathcal{C}^{\mathbf{R}}_{N-D}(h)^{\mathrm{T}} & \mathcal{C}^{\mathbf{L}}_{D}(g^*) \end{pmatrix} \in \mathbb{R}^{\eta \times \nu},$$

where the block matrices are linearized accordingly. In our formulation of the approximate GCRD problem, we normalize $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h$ so that it is a predetermined constant, which results in essentially the same Jacobian as described above.

The only difference in the Jacobians is that the νth column would become the zero column if differentiated with respect to $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h$, since $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h$ is constant. When normalized for computational purposes, the Jacobian belongs to $\mathbb{R}^{\eta \times \nu - 1}$ instead (the last column is deleted). In the general case when $\operatorname{gcrd}(f^*, g^*) = 1$, J is rank deficient by 1 and the νth column is a linear combination of the other columns. The following lemma, similar to ([30], Lemma 4.1) formalizes this statement.

Lemma 3.2 *Let r be the residual described earlier with Jacobian J. Suppose that $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h$ is a fixed nonzero constant. If $\operatorname{gcrd}(f^*, g^*) = 1$, then all nonzero columns of J are linearly independent.*

Proof Let $\mathbf{e}_\nu \in \mathbb{R}^{1 \times \nu}$ be a unit vector whose last component is 1. We write

$$\mathbf{e}_\nu (0, \ldots, 0, \mathbf{h})^{\mathrm{T}} = \operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h \neq 0.$$

We shall prove the equivalent statement that the matrix

$$\begin{pmatrix} J \\ \mathbf{e}_\nu \end{pmatrix} = \begin{pmatrix} \mathcal{C}^{\mathbf{R}}_{M-D}(h)^{\mathrm{T}} & 0 & \mathcal{C}^{\mathbf{L}}_{D}(f^*) & \mathcal{C}^{\mathbf{R}}_{N-D}(h)^{\mathrm{T}} & \mathcal{C}^{\mathbf{L}}_{D}(g^*) \\ & & \mathbf{e}_\nu 0 & & \end{pmatrix} \in \mathbb{R}^{(\eta+1) \times \nu}$$

has full rank.

Suppose the converse holds, then there exists $q_1, q_2, p \in \mathbb{R}[t][\partial;']$ with $\deg_\partial q_1 \leq M - D$, $\deg_\partial q_2 \leq N - D$ and $\deg_\partial p \leq D$ such that their combined coefficient vector satisfies

$$\begin{pmatrix} J \\ \mathbf{e}_\nu \end{pmatrix} \begin{pmatrix} \mathbf{q_1}^{\mathrm{T}} \\ \mathbf{q_2}^{\mathrm{T}} \\ -\mathbf{p}^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

Expressing this as multiplication over $\mathbb{R}[t][\partial;']$, we have that

$$f^* p = q_1 h,$$
$$g^* p = q_2 h.$$

We conclude that $\operatorname{gcrd}(f^* p, g^* p) = p$, as $\operatorname{gcrd}(f^*, g^*) = 1$. If $p = 0$ or $q_1 = 0$ or $q_2 = 0$, then we are done (as $\mathbb{R}[t][\partial;']$ is a domain). Suppose that $p \neq 0$ and $q_1 \neq 0$ and $q_2 \neq 0$. Accordingly, we must also have that $\operatorname{gcrd}(q_1 h, q_2 h) = \operatorname{gcrd}(q_1, q_2)\alpha h = p$ for some $\alpha \neq 0$. Since $\deg_\partial p \leq \deg_\partial h$ it follows that $\operatorname{gcrd}(q_1, q_2) = 1$ so $p = \alpha h$.

Since $p = \alpha h$, we must have that $\alpha f^* = q_1$ and $\alpha g^* = q_2$. Now,

$$\mathbf{e}_\nu (0, \ldots, 0, \mathbf{h})^{\mathrm{T}} = \operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h \neq 0.$$

On the other hand,

$$\mathbf{e}_\nu (0, \ldots, 0, \alpha\mathbf{h})^{\mathrm{T}} = 0.$$

This occurs if and only if $\alpha = 0$. But in this case $p = 0$ as well, so

$$\begin{pmatrix} \mathbf{q_1}^{\mathrm{T}} \\ \mathbf{q_2}^{\mathrm{T}} \\ -\mathbf{p}^{\mathrm{T}} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

It follows that the only vector in the null space is the zero vector; hence, $\begin{pmatrix} J \\ \mathbf{e}_\nu \end{pmatrix}$ has full rank. Since any subset of linearly independent vectors is also linearly independent, we have that when $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h$ is a fixed nonzero constant that J has rank $\nu - 1$. □

Note that from the proof we see that if $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h$ were not fixed, then the vector $(\mathbf{f}^*, \mathbf{g}^*, \mathbf{h})^{\mathrm{T}}$ forms a basis for the nullspace of J. Intuitively, if we did not fix $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial$ in advance, then there would be infinitely many tuples of (h, f^*, g^*) with the same degree structure over $\mathbb{R}[t]$ that minimized Φ, since for any $\alpha \neq 0$ we have

$$\|f - f^* h\|_2^2 + \|g - g^* h\|_2^2 = \|f - (\alpha f^*)(\alpha^{-1} h)\|_2^2 + \|g - (\alpha g^*)(\alpha^{-1} h)\|_2^2.$$

In other words, we need to normalize h in advance to obtain a unique solution.

Corollary 3.2 *Let r be the residual defined earlier in this section with* $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h$ *a nonzero constant. If $r = 0$, then the Hessian matrix $\nabla^2\Phi(h, f^*, g^*)$ is positive definite.*

Proof Let J be the Jacobian of r. J has full rank, so $J^{\mathrm{T}}J$ has full rank and is positive semidefinite. If $r = 0$, at the global minimum we have that $2J^{\mathrm{T}}J = \nabla^2\Phi$, and $\nabla^2\Phi(h, f^*, g^*)$ is positive definite. □

When there is no residual, the Hessian $\nabla^2\Phi(h, f^*, g^*)$ is positive definite. It follows that if f and g are perturbed by a sufficiently small amount, then $\nabla^2\Phi$ remains locally positive definite, and Newton iteration will converge to the (local) global minimum with an initial guess that is sufficiently close.

We are able to obtain a condition number for a structured perturbation through the Jacobian of the residuals. Since J has full rank, the smallest singular value $\sigma_{\nu-1}$ of $J(r(h, f^*, g^*))$ is strictly positive. If we consider structured perturbations, then we are able to show that the approximate GCRD problem is (locally) well posed.

In the next lemma, we make use of the fact that for any $f \in \mathbb{R}[t][\partial;']$, we have that $\|f\|_2 = \|\mathbf{f}\|_2$.

Lemma 3.3 *Let $f, g, h, f^*, g^* \in \mathbb{R}[t][\partial;']$ be such that $\Phi(h, f^*, g^*) < \varepsilon$ for some $\varepsilon > 0$, with* $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h$ *a fixed nonzero constant. Suppose $\widehat{f}, \widehat{g}, \widehat{h}, \widehat{f^*}, \widehat{g^*} \in \mathbb{R}[t][\partial;']$ possess the same degree structures as f, g, h, f^* and g^* and that*

$$\widehat{\Phi}(\widehat{h}, \widehat{f^*}, \widehat{g^*}) = \|\widehat{f} - \widehat{f^*}\widehat{h}\|_2^2 + \|\widehat{g} - \widehat{g^*}\widehat{h}\|_2^2 < \varepsilon.$$

Then,

$$\left\|(h - \widehat{h}, f^* - \widehat{f^*}, g^* - \widehat{g^*})\right\|_2^2 \leq \frac{1}{\sigma_{\nu-1}^2}\left(2\varepsilon + \left\|(f - \widehat{f}, g - \widehat{g})\right\|_2^2\right) + \textit{higher-order terms.}$$

Proof Let $J = J(r(h, f^*, g^*))$ be the Jacobian of the residuals from earlier in this section. We have that

$$(\mathbf{f}^*\mathbf{h} - \widehat{\mathbf{f}^*}\widehat{\mathbf{h}}, \mathbf{g}^*\mathbf{h} - \widehat{\mathbf{g}^*}\widehat{\mathbf{h}})^{\mathrm{T}} \approx J(\mathbf{h} - \widehat{\mathbf{h}}, \mathbf{f}^* - \widehat{\mathbf{f}^*}, \mathbf{g} - \widehat{\mathbf{g}^*})^{\mathrm{T}}.$$

Ignoring high-order terms and using the well-known fact that for a (left) pseudo-inverse J^+ of J, that $\|J^+\|_2 = \frac{1}{\sigma_{\nu-1}}$ gives us

$$\begin{aligned}\left\|(\mathbf{f}^*\mathbf{h} - \widehat{\mathbf{f}^*}\widehat{\mathbf{h}}, \mathbf{g}^*\mathbf{h} - \widehat{\mathbf{g}^*}\widehat{\mathbf{h}})^{\mathrm{T}}\right\|_2^2 &\approx \left\|J(\mathbf{h} - \widehat{\mathbf{h}}, \mathbf{f}^* - \widehat{\mathbf{f}^*}, \mathbf{g}^* - \widehat{\mathbf{g}^*})^{\mathrm{T}}\right\|_2^2 \\ &\geq \sigma_{\nu-1}^2 \left\|(\mathbf{h} - \widehat{\mathbf{h}}, \mathbf{f}^* - \widehat{\mathbf{f}^*}, \mathbf{g}^* - \widehat{\mathbf{g}^*})\right\|_2^2.\end{aligned}$$

A straightforward application of the triangle inequality gives

$$\begin{aligned}
&\left\| (h - \widehat{h},\, f^* - \widehat{f^*},\, g^* - \widehat{g^*}) \right\|_2^2 \\
&\quad \le \frac{1}{\sigma_{\nu-1}^2} \left\| (f^* h - \widehat{f^* h},\, g^* h - \widehat{g^* h}) \right\|_2^2 \\
&\quad \le \frac{1}{\sigma_{\nu-1}^2} \left(\Phi(h,\, f^*,\, g^*) + \widehat{\Phi}(\widehat{h},\, \widehat{f^*},\, \widehat{g^*}) + \left\| (f - \widehat{f},\, g - \widehat{g}) \right\|_2^2 \right) \\
&\quad \le \frac{1}{\sigma_{\nu-1}^2} \left(2\varepsilon + \left\| (f - \widehat{f},\, g - \widehat{g}) \right\|_2^2 \right) + \textit{higher-order terms.}
\end{aligned}$$

□

Corollary 3.3 *Suppose that* $h_{\text{opt}}, f^*_{\text{opt}}, g^*_{\text{opt}} \in \mathbb{R}[t][\partial;']$ *are a locally unique global minimum of* Φ *in some neighborhood around* h, f^* *and* g^*. *If*

$$\Phi(h,\, f^*,\, g^*) < \varepsilon \text{ and } \Phi(h_{\text{opt}},\, f^*_{\text{opt}},\, g^*_{\text{opt}}) < \varepsilon$$

for $\varepsilon > 0$, *then*

$$\left\| (h - h_{\text{opt}},\, f^* - f^*_{\text{opt}},\, g^* - g^*_{\text{opt}}) \right\|_2^2 \le \frac{2\varepsilon}{\sigma_{\nu-1}^2} + \textit{higher-order terms.}$$

If we compute different approximate GCRD pairs of f and g (using different optimization techniques or initial guesses), then we are able to bound the size of the perturbations of f^*, g^* and h based on how near they are. Furthermore, this corollary allows us to certify an upper bound on the distance between our computed approximate GCRD tuple and the actual global minimum.

4 Implementation of Approximate GCRD

This section discusses the particulars and implementation of the algorithms. The algorithms are described in a Maple-like pseudo code, with MATLAB style matrix indexing. All of the algorithms have been implemented in the Maple programming language. For convenience, the notation and assumptions introduced at the start of Sect. 3 will hold, unless otherwise stated. Additionally, we will assume that content from differential polynomials can be removed numerically, as computed quantities are typically not primitive due to round-off errors.

The matrices $S = S(f, g) \in \mathbb{R}[t]^{(M+N)\times(M+N)}$ will be the differential Sylvester matrix of f and g, and $\widehat{S} = \widehat{S}(f, g) \in \mathbb{R}^{(M+N)(\mu+1)\times(M+N)(\mu+d+1)}$ will be the inflated differential Sylvester matrix of f and g, where $\mu = 2(M + N)d$.

The presentation and theoretical analysis of the algorithms is presented in a bottom-up manner, reflecting their dependencies. Asymptotic upper-bounds on the number of floating point operations required are provided. Furthermore, we discuss whether the output of the algorithm can be certified in some manner, when applicable.

We demonstrate the robustness of our algorithms in practice. Specific examples are provided to thoroughly demonstrate the steps of the algorithms. We investigate

interesting families of input. In particular, we investigated exact inputs with an exact GCRD, and perturbed differential polynomials with varying errors and noise introduced. The test cases of differential polynomials of interest to us have

- low degree in t and high degree in ∂ (unbalanced in ∂),
- high degree in t and low degree in ∂ (unbalanced in t), and
- proportional degrees in t and ∂ (balanced degrees).

4.1 Algorithms for Approximate GCRD

We adapt techniques from the exact setting to a numerical setting to compute an exact GCRD numerically. These algorithms compute the rank of the differential Sylvester matrix and a least squares solution to a polynomial linear system, corresponding to the Bézout coefficients. We describe an algorithm for finding nearby differential polynomials introduced in [9], whose (inflated) differential Sylvester matrix is nearly singular. Using the least squares numeric GCRD algorithm, we can compute an approximate GCRD candidate from the nearly singular differential Sylvester matrix. From this candidate, we extract a guess for the cofactors numerically and proceed with post-refinement Newton iteration.

4.1.1 Numerical Computation of a GCRD

Before we can compute a GCRD numerically, the rank of the differential Sylvester matrix needs to be determined. Our numeric rank algorithm is an adaptation of the rank algorithm used by Corless et al. [7]. There are

$$(M+N)(\mu+d+1)-(M+N)(\mu+1)=(M+N)d=\mu/2$$

trivial singular values,[2] and $\mu/2 < \mu+d+1$, the column block size. These trivial singular values need to be accounted for when annihilating small singular values. In the full rank case, we should not underestimate the rank of S by inferring from $\widehat{S}$, as there are strictly fewer trivial singular values than the column block size.

Algorithm 1 computes a reasonable guess for the degree in ∂ of an approximate GCRD, although it is not generally certifiable. When $\gcrd(f,g)$ is non-trivial (no errors present in the input coefficients), we compute (generically) the degree of the GCRD of f and g. In the exact setting, we can now formulate a linear algebra problem over $\mathbb{R}[t]$ to compute a GCRD. We present two solutions to this problem. Algorithm 2 solves this problem using linear algebra over $\mathbb{R}(t)$. Algorithm 3 linearizes the problem over $\mathbb{R}$ and computes a least squares solution.

In the implementation of Algorithm 2, we take special care to ensure that $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h$ does not vanish when h is normalized. If $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h$ vanishes, then this could be an indication that the input is ill-conditioned or content removal

[2] The inflated differential Sylvester matrix has more columns than rows; however, the nullspace of the columns contains the information pertaining to the GCRD. The trivial singular values are the zero singular values occurring from there being more columns than rows.

Algorithm 1 : `DeflatedRank`

Input:
- An inflated differential Sylvester matrix

$$\widehat{S} \in \mathbb{R}^{(M+N)(\mu+1)\times(M+N)(\mu+d+1)};$$

- A user defined search radius $\varepsilon_{rank} > 0$ for comparing singular values.

Output:
- The (scaled) numeric rank ϱ of the (non-inflated) differential Sylvester matrix S.

1: Compute the singular values $\sigma_1, \sigma_2, \ldots, \sigma_{(M+N)(\mu+d+1)}$ of $\widehat{S}$ in descending order.
2: Find the maximum k such that $\sigma_k > \varepsilon_{rank} \frac{\sqrt{(M+N)(2\mu+d+2)}}{\mu+d+1}$ and $\sigma_{k+1} < \varepsilon_{rank}$.
3: if $\sigma_k > \varepsilon_{rank}$ for all k then $\widehat{S}$ has full rank.
4: If there is no significant change (there is no maximum k) between σ_k and σ_{k+1} for all k, as determined by step 2 then return failure.
5: Set $\varrho = \left\lceil \frac{k}{\mu+d+1} \right\rceil$, the scaled rank of S.

Algorithm 2 : `NumericGCRD`

Input:
- $f, g \in \mathbb{R}[t][\partial;']$ with $\|f\| = \|g\| = 1$;
- A search radius $\varepsilon_{rank} > 0$.

Output:
- $h = \gcrd(f, g) \in \mathbb{R}[t][\partial;']$ with $\deg_\partial h \geq 1$,
- or an indication that f and g are co-prime within search radius ε_{rank}.

1: $M \leftarrow \deg_\partial f$, $N \leftarrow \deg_\partial g$, $d \leftarrow \max\{\deg_t f, \deg_t g\}$, $\mu \leftarrow 2(M+N)d$.
2: $S \leftarrow S(f, g) \in \mathbb{R}[t]^{(M+N)\times(M+N)}$.
3: Form the inflated differential Sylvester matrix $\widehat{S} = \widehat{S}(f, g) \in \mathbb{R}^{(M+N)(\mu+1)\times(M+N)(\mu+d+1)}$ of S.
4: Compute the numerical rank ϱ of S using Algorithm 1 on $\widehat{S}$ with search radius ε_{rank}.
5: If $\varrho > 0$, then set $\deg_\partial h = D = M + N - \varrho$. Otherwise indicate that f and g are co-prime with respect to ε_{rank} and return.
6: Solve for $w \in \mathbb{R}[t][\partial;']^{1\times(M+N)}$ from

$$wS = (*_1, *_2, \ldots, *_{D+1}, 0, \ldots, 0),$$

ensuring that $\|\operatorname{lcoeff}_t(*_{D+1})\| \gg 0$.
7: Set $(h_0, h_1, \ldots, h_D, 0, \ldots, 0) = wS$.
8: **return** $\operatorname{cont}(h)^{-1} h$.

of h failed. In either case, it is possible that this instance of the approximate GCRD problem will not have an attainable global minimum in accordance with Theorem 3.1.

4.1.2 Nearby Differential Polynomials with GCRD Algorithm

The matrix $\widehat{S}$ is highly structured, as it is composed of block Toeplitz matrices. When we consider the matrix $\widehat{S} + \Delta\widehat{S}$, the nearest (unstructured) matrix of prescribed rank deficiency, we have considerable flexibility in how we recover the coefficients of $\widetilde{f}$ and $\widetilde{g}$, nearby differential polynomials with an exact, non-trivial GCRD as in (3.1). However, the matrix $\widehat{S} + \Delta\widehat{S}$ is not generally an inflated differential Sylvester matrix,

Algorithm 3 : `NumericGCRDviaLS`

Input:
- $f, g \in \mathbb{R}[t][\partial;']$ with $\|f\| = \|g\| = 1$;
- $\varepsilon_{rank} > 0$ used to compute the degree of the GCRD.

Output:
- $h \in \mathbb{R}[t][\partial;']$ that is numerically primitive with a fixed leading coefficient such that $\|wS(f,g) - h\|_2^2$ is minimized.

1: $M \leftarrow \deg_\partial f$, $N \leftarrow \deg_\partial g$, $d \leftarrow \max\{\deg_t f, \deg_t g\}$, $\mu \leftarrow 2(M+N)d$.
2: Compute D using Algorithm 1 with ε_{rank}.
3: $\overrightarrow{\deg} \leftarrow (\underbrace{\mu + d, \ldots, \mu + d}_{D+1}, 0, \ldots 0)$ (or another valid initial guess).
4: Compute a least squares solution of h from $\|wS(f,g) - h\|_2$ with $\overrightarrow{\deg}(h) = \overrightarrow{\deg}$ and $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h = 1$.
5: $\overrightarrow{\deg} \leftarrow \overrightarrow{\deg}(\operatorname{cont}(h)^{-1}h)$.
6: Compute a new least squares solution of h from $\|wS(f,g) - h\|_2$ with $\overrightarrow{\deg}(h) = \overrightarrow{\deg}$ and $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h = 1$.
7: **return** h.

but it is probably reasonably close to one (see Giesbrecht and Haraldson [9]; Haraldson [13]). We recall that the mapping $\Gamma : \mathbb{R}[t] \to \mathbb{R}^{(\mu+1)\times(\mu+d+1)}$ generates the (rectangular) Toeplitz blocks of $\widehat{S}$. To recover the coefficients of $\widetilde{f}$ and $\widetilde{g}$ one must make a suitable definition for the mapping $\Gamma^{-1} : \mathbb{R}^{(\mu+1)\times(\mu+d+1)} \to \mathbb{R}[t]$. We use Γ^{-1} to find $\widetilde{f}, \widetilde{g} \in \mathbb{R}[t][\partial;']$ such that $\widehat{S}(\widetilde{f}, \widetilde{g}) \approx \widehat{S}(f,g) + \Delta\widehat{S}(f,g)$.

Algorithm 4 : `DeflatedPerturbation`

Input:
- $f, g \in \mathbb{R}[t][\partial;']$ with $\|f\| = \|g\| = 1$;
- Perturbed inflated differential Sylvester matrix

$$\widehat{S} + \Delta\widehat{S} \in \mathbb{R}^{(M+N)(\mu+1)\times(M+N)(\mu+d+1)};$$

- $\Gamma^{-1} : \mathbb{R}^{(\mu+1)\times(\mu+d+1)} \to \mathbb{R}[t]$.

Output:
- $\widetilde{f}, \widetilde{g} \in \mathbb{R}[t][\partial;']$ where $\overrightarrow{\deg}(\widetilde{f}) \le \overrightarrow{\deg}(f)$ and $\overrightarrow{\deg}(\widetilde{g}) \le \overrightarrow{\deg}(g)$.

1: $M \leftarrow \deg_\partial f$, $N \leftarrow \deg_\partial g$, $d \leftarrow \max\{\deg_t f, \deg_t g\}$, $\mu \leftarrow 2(M+N)d$ and $N \leftarrow \mu + d + 1$.
2: **for** $0 \le i \le \deg_\partial f$ **do**
3: $\quad [I, J] \leftarrow [1 : \mu + 1][(i+1) + (i-1)N(\mu+d+1) : (i+1)N(\mu+d+1)]$
4: $\quad \widetilde{f}_i \leftarrow \Gamma^{-1}\left((\widehat{S} + \Delta\widehat{S})[I, J]\right)$
5: **end for**
6: **for** $0 \le i \le \deg_\partial g$ **do**
7: $\quad [I, J] \leftarrow [N(\mu+1)+1 : (N+1)(\mu+1)][(i+1) + (i-1)M(\mu+d+1) : (i+1)M(\mu+d+1)]$
8: $\quad \widetilde{g}_i \leftarrow \Gamma^{-1}\left((\widehat{S} + \Delta\widehat{S})[I, J]\right)$
9: **end for**
10: **return** $\widetilde{f}$ and $\widetilde{g}$.

Regardless of our choice of Γ^{-1}, this method of recovering $\widetilde{f}$ and $\widetilde{g}$ can lead to a differential Sylvester matrix that does not have the desired numeric rank, as determined by Algorithm 1. The perturbation $\Delta\widehat{S}$ is unstructured while $\Gamma(\widetilde{f}_i)$ and

Algorithm 5 : `NearbyWithGCRD`

Input:
- $f, g \in \mathbb{R}[t][\partial;']$ with $\|f\| = \|g\| = 1$;
- A search radius $\varepsilon_{rank} > 0$, used to validate the degree of h.

Output:
- $\widetilde{f}, \widetilde{g} \in \mathbb{R}[t][\partial;']$ where $\overrightarrow{\deg}(\widetilde{f}) \leq \overrightarrow{\deg}(f)$, $(\overrightarrow{\deg}(\widetilde{g}) \leq \overrightarrow{\deg}(g)$ and $h \approx \operatorname{gcrd}(\widetilde{f}, \widetilde{f}) \in \mathbb{R}[t][\partial;']$ with $\deg_\partial h \geq 1$, or ;
- An indication that f and g are co-prime within search radius ε_{rank}.

1: $M \leftarrow \deg_\partial f$, $N \leftarrow \deg_\partial g$, $d \leftarrow \max\{\deg_t f, \deg_t g\}$ and $\mu \leftarrow 2(M+N)d$.
2: $S \leftarrow S(f,g) \in \mathbb{R}[t]^{(M+N)\times(M+N)}$.
3: $\widehat{S} \leftarrow \widehat{S}(f,g) \in \mathbb{R}^{(M+N)(\mu+1)\times(M+N)(\mu+d+1)}$.
4: Compute the SVD of $\widehat{S}$, where $\widehat{S} = P\Sigma Q$.
5: Compute the numerical rank ϱ of S using Algorithm 1 on $\widehat{S}$ with search radius ε_{rank}.
6: If $\varrho > 0$ set the last $\varrho(\mu+d+1)$ singular values to 0 and compute $\overline{\Sigma}$.
   Otherwise indicate that f and g are co-prime with respect to ε_{rank}.
7: Compute $\widehat{S} + \Delta\widehat{S} = P\overline{\Sigma}Q$.
8: Compute $\widetilde{f}$ and $\widetilde{g}$ from $\widehat{S} + \Delta\widehat{S}$ using Algorithm 4.
9: Compute $h = \texttt{NumericGCRD}(\widetilde{f}, \widetilde{g})$ using Algorithm 3, with ε_{rank} used to validate the degree of h using Algorithm 1.
10: **return** $\widetilde{f}, \widetilde{g}$ and h.

$\Gamma(\widetilde{g}_j)$ are (highly structured) Toeplitz matrices. Consequently, some nonzero terms of $\Delta\widehat{S}$ are ignored.

4.1.3 Numeric Right Division

Numeric right division without remainder between two differential polynomials is a rational function linear algebra problem. The (approximate) quotient is a solution to a linear system, in a least squares sense. We present a naive algorithm that works well in practice and a more rigorous linear least squares variant.

The solution to this system may not be in (approximate) lowest terms. In our implementation, we use approximate GCD and real linear least squares to resolve this. We note that total least squares can also be employed to prevent the need of an approximate GCD computation to put the rational function coefficients in lowest terms.

4.1.4 Improved GCRD via Optimization: Newton's Method

Using Algorithm 5, we can compute an initial guess for an approximate GCRD, h_{init}. We can perform right division without remainder numerically to compute initial guesses for the cofactors, f^*_{init} and g^*_{init}. We now have enough information to set up a post-refinement Newton iteration, to hopefully compute an approximate GCRD. When the cofactors have polynomial coefficients, the products f^*h and g^*h are always polynomial. This makes Newton iteration a very straightforward procedure, as the objective function

$$\Phi(h, f^*, g^*) = \|f - f^*h\|_2^2 + \|g - g^*h\|_2^2$$

Algorithm 6 : `NaiveNumericRightDivision`

Input:
- $f, h \in \mathbb{R}[t][\partial;']$ with $\|f\| = \|h\| = 1$.

Output:
- $f^* \in \mathbb{R}(t)[\partial;']$ satisfying $f = f^*h$.

1: $M \leftarrow \deg_\partial f$, $D \leftarrow \deg_\partial h$.
2: Form the matrix $\mathcal{M}(h)$ from the last $M - D + 1$ columns of $\mathcal{C}^{\mathbf{R}}_{M-D}(h)$.
3: Solve

$$(f_D, f_{D+1}, \ldots, f_M) = (f_0^*, f_1^*, \ldots, f_{M-D}^*)\mathcal{M}(h)$$

by backwards substitution for the coefficients of f^*.
4: **for** $0 \le i \le M - D$ **do**
5: $\quad f_i^* \leftarrow$ Approximate f_i^* in rational function least terms
6: **end for**
7: **return** f^*.

Algorithm 7 : `NumericRightDivisionViaLS`

Input:
- $f, h \in \mathbb{R}[t][\partial;']$ with $\|f\| = \|h\| = 1$.

Output:
- $f^* \in \mathbb{R}(t)[\partial;']$ in lowest terms satisfying $f = f^*h$.

1: $M \leftarrow \deg_\partial f$, $D \leftarrow \deg_\partial h$.
2: Solve

$$v_{-1}(f_0, f_1, \ldots, f_M) = (v_0, v_1, \ldots, v_{M-D})\mathcal{C}^{\mathbf{R}}_{M-D}(h)$$

by linear least squares for the coefficients of $v_{-1}, v_0, \ldots, v_{M-D}$.
3: **for** $0 \le i \le M - D$ **do**
4: $\quad f_i^* \leftarrow$ Approximate $\dfrac{v_i}{v_{-1}}$ in rational function least terms
5: **end for**
6: **return** f^*.

is easily computed. However, when the cofactors have rational function coefficients, the quantities f^*h and g^*h usually have rational function coefficients due to round-off error. We can clear fractions and compute the least squares solution of an equivalent associate problem.

The normalization we impose, that $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h$ is fixed, ensures the solution is (locally) unique, by Corollary 3.2. We note that this normalization can be changed. However, one must ensure that the normalization vector is not orthogonal to $(\mathbf{f}^*, \mathbf{g}^*, \mathbf{h})$. We now generalize the Newton iteration for the instance when the cofactors have rational function coefficients.

4.2 Analysis of Algorithms

In this section, we assess the computational cost in terms of the number of floating point operations or *flops*. Where applicable, we discuss the numerical stability of the algorithms and whether or not their output can be certified. The algorithms are analyzed in the order they were presented. The assumption that content can be removed numeri-

Algorithm 8 : `NewtonIteration`

Input:
- $f, g, h_{init} \in \mathbb{R}[t][\partial;']$ with $\|f\| = \|g\| = \|h_{init}\| = 1$;
- $k \in \mathbb{N}$, the number of iterations.

Output:
- $f^*, g^* \in \mathbb{R}[t][\partial;']$ and $h \in \mathbb{R}[t][\partial;']$ such that $\Phi(f^*h, g^*h)$ is locally minimized and
- $\overrightarrow{\deg}(f^*h) \leq \overrightarrow{\deg}(f)$ and $\overrightarrow{\deg}(g^*h) \leq \overrightarrow{\deg}(g)$.

$M \leftarrow \deg_\partial f$, $N \leftarrow \deg_\partial g$ and $D \leftarrow \deg_\partial h_{init}$.
Compute initial guesses of f^* and g^* using Algorithm 7.
$\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h \leftarrow \operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h_{init}$.
$x^0 \leftarrow (\mathbf{f^*_{init}}, \mathbf{g^*_{init}}, \mathbf{h_{init}})^T$.
for $1 \leq i \leq k$ **do**
    Solve $\nabla^2\Phi(x^i) \cdot x^{i+1} = \nabla^2\Phi(x^i) \cdot x^i - \nabla\Phi(x^i)$ for x^{i+1}.
end for
return f^*, g^*, and h computed from x^k.

Algorithm 9 : `ModifiedNewtonIteration`

Input:
- $f, g, h_{init} \in \mathbb{R}[t][\partial;']$ with $\|f\| = \|g\| = \|h_{init}\| = 1$;
- $k \in \mathbb{N}$, the number of iterations.

Output:
- $f^*, g^* \in \mathbb{R}(t)[\partial;']$ and $h \in \mathbb{R}[t][\partial;']$ such that $\Phi(f^*h, g^*h)$ is locally minimized and
- $\overrightarrow{\deg}(f^*h) \leq \overrightarrow{\deg}(f)$ and $\overrightarrow{\deg}(g^*h) \leq \overrightarrow{\deg}(g)$.

$M \leftarrow \deg_\partial f$, $N \leftarrow \deg_\partial g$ and $D \leftarrow \deg_\partial h_{init}$.
Compute initial guesses of f^* and g^* using Algorithm 7.
$\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h \leftarrow \operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h_{init}$.
$f \leftarrow f^*_{-1} f$, $g \leftarrow g^*_{-1} g$, $f^* \leftarrow f^*_{-1} f^*$ and $g^* \leftarrow g^*_{-1} g^*$.
$x^0 \leftarrow (\mathbf{f^*_{init}}, \mathbf{g^*_{init}}, \mathbf{h_{init}})^T$.
for $1 \leq i \leq k$ **do**
    Solve $\nabla^2\Phi(x^{i+1}) \cdot x^i = \nabla^2\Phi(x^i) \cdot x^i - \nabla\Phi(x^i)$ for x^{i+1}.
end for
$f^* \leftarrow \frac{1}{f^*_{-1}} f^*$ and $g^* \leftarrow \frac{1}{g^*_{-1}} g^*$.
return f^*, g^* and h computed from x^k.

cally is not without loss of generality; content removal can be unstable if implemented poorly.

In our implementation, we remove content by (re)formulating our solutions as a solution to a (total) least squares problem. This can be done by performing the SVD on a generalized Sylvester matrix of several univariate polynomials [16] to infer the degree of the content. Computing the degree of the content this way generalizes the method of Corless et al. [7] to several polynomials. In our implementation, the only important information is the degree of an approximate GCD, so we assume that the run-time of approximate GCD is cubic in the number of variables. One could compute an approximate GCD of several polynomials and perform a least squares division; however, post-refinement would likely be needed. We generally assume that unstructured linear algebra techniques are used on the problems; however, structured methods could lead to a modest asymptotic improvement.

4.2.1 Analysis of Algorithm 1—DeflatedRank

The number of flops Algorithm 1 requires is dominated by the cost of performing the SVD on $\widehat{S}$. The SVD requires $O((M+N)^3(\mu+d+1)^3) = O((M+N)^6 d^3)$ flops, using standard arithmetic. As mentioned earlier, this algorithm is generally not certified to produce the degree of an approximate GCRD.

4.2.2 Analysis of Algorithm 2—NumericGCRD

The number of flops Algorithm 2 requires is ultimately bounded by the cost of computing the rank of S using Algorithm 1. The cost of Algorithm 1 is $O((M+N)^6 d^3)$ flops. The cost of computing a GCRD given the degree in ∂ is $O((M+N)^3)$ operations over $\mathbb{R}(t)$ which corresponds to $O((M+N)^3 d^2)$ flops. The cost of the approximate GCD and division to remove content depends on the specific method used, but is usually negligible when compared to the rank computation.

This algorithm is not numerically stable for large degree inputs in t and ∂. Performing linear algebra over $\mathbb{R}(t)$ leads to considerable degree growth in t, and removing (approximate) content with a division further perturbs the coefficients of the GCRD. The output of this algorithm is not certified to be correct in most instances.

4.2.3 Analysis of Algorithm 3—NumericGCRDviaLS

There are $(M+N)(\mu+d+1)$ equations and $(M+N)(\mu+1)+(D+1)(\mu+d+1) = O((M+N)^2 d)$ unknowns. The cost of computing the least squares solution is $O((M+N)^6 d^3)$ flops. The cost of inferring the content by looking at singular values of the (generalized) Sylvester matrix is bounded by $O((D)^3(\mu+d+1)^3) = O((M+N)^6 d^3)$ flops. The total number of flops required for this algorithm is $O((M+N)^6 d^3)$.

This algorithm relies on solving a real linear least squares problem. As such, this algorithm is numerically stable, provided that the underlying least squares problem is reasonably conditioned, and solved in a reasonable way. One such method of solving the least squares problem is the SVD and arising pseudo-inverse. We are able to certify the correctness of the answer obtained via least squares, provided that the underlying approximate GCD algorithm computes the degree of the content correctly.

4.2.4 Analysis of Algorithm 4—DeflatedPerturbation

The number of flops Algorithm 4 requires is $O((M+N)^2 d^2)$, assuming that Γ^{-1} uses the weighted block average. We use this in our implementation. This algorithm is not certified to provide meaningful output.

4.2.5 Analysis of Algorithm 5—NearbyWithGCRD

The number of flops Algorithm 5 requires is dominated by the cost of computing the singular values of $\widehat{S}$, which is $O((M+N)^6 d^3)$ flops.

This algorithm is exactly the same as Algorithm 2 when $\widehat{S}$ has the desired rank deficiency. In the event that the input is approximate, the quality of our answer depends

on the largest singular value of $\widehat{S}$ that we annihilate. This algorithm is not certified to provide meaningful output, but if used in conjunction with Algorithm 3, the output can be certified as a least squares approximation to the solution of the Bézout coefficients.

4.2.6 Analysis of Algorithm 6—`NaiveNumericRightDivision`

The number of flops Algorithm 6 requires depends on the method used to solve the linear system. The particular system is highly structured so we can solve it by backwards substitution directly, which costs $O((M-D)^2)$ operations over $\mathbb{R}(t)$. This corresponds to $O((M-D)^2 d^2)$ flops. An upper bound on the degree required for approximate GCD computations is $(M-D+1)d$. The total cost of each approximate GCD computation is at most $O(((M-D)d)^3)$ flops. There are at most $M-D+1$ approximate GCD computations performed, so the total cost of the algorithm is $O((M-D)^4 d^3)$ flops.

The output of this answer is generally only certifiable if the residual of a least squares division is zero, i.e., the coefficients are exact. If we assume that $\operatorname{lcoeff}_t \operatorname{lcoeff}_\partial h = 1$ and $\|h\|$ is not arbitrarily large, then the backwards substitution is well conditioned. The approximate GCD computations and following divisions can perturb the coefficients, so the algorithm can be unstable for poorly conditioned inputs. This is especially problematic when $\operatorname{lcoeff}_\partial h$ is poorly conditioned.

4.2.7 Analysis of Algorithm 7—`NumericRightDivisionViaLS`

If f^* has polynomial coefficients, then $\deg_t f^* \leq \deg_t f \leq d$ as f and h have polynomial coefficients as well. If f^* has rational function coefficients, we recall from Sect. 2.2 that there are $O(M(M-D)d)$ equations and $O((M-D)^2 d)$ unknowns. The cost of solving this linear least squares problem is $O((M(M-D)d)^3) \subseteq O(M^6 d^3)$ flops.

The output of this algorithm is certified as a linear least squares solution. Like Algorithm 6, the conditioning of this algorithm is strongly related to the conditioning of h_D.

4.2.8 Analysis of Algorithms 8–9—`NewtonIteration` and `ModifiedNewtonIteration`

We transform the problem of computing a GCRD to that of optimizing $\Phi : \mathbb{R}[t][\partial;']\times \mathbb{R}(t)[\partial;']^2 \to \mathbb{R}$. We can assume without loss of generality that f^* and g^* have polynomial coefficients, as we can solve an equivalent associate problem instead. The dominating cost of the Newton iteration is solving a linear system to get the next value which requires $O(\nu^3)$ operations, where ν is the number of variables needed to represent the coefficients of h, f^* and g^*.

Newton iteration can fail for many reasons, (it is, afterall, a locally convergent method); however, our Newton iteration usually fails because:

1. $\nabla^2 \Phi$ is positive semidefinite at a point in the iteration, the stationary point is a saddle point;

2. The initial guess is poorly chosen and $\nabla^2 \Phi$ is indefinite at a point.

In the event that Newton iteration fails, we can perform a Gauss–Newton iteration instead. Despite Gauss–Newton iteration having at least linear convergence, $J^T J$ is positive definite, so saddle points are no longer a problem if the optimal residual is sufficiently small. According to Corollary 3.2, if the residual is sufficiently small then Newton iteration will converge to a global minimum.

4.3 Examples and Experimental Results

This section contains some examples of our implementation.[3] The (inflated) differential Sylvester matrix is ill-conditioned for large degree inputs in t and ∂. This ill-conditioning occurs because the columns (rows) become unbalanced due to the falling factorials, where some columns have a Frobenius norm factorially larger than others. We restrict ourselves to modest examples with minimal coefficient growth. Computations are done using the default precision in `Maple`, which is approximately 10 decimal points of accuracy.

Example 4.1 (No Noise, many factors)

$$\begin{aligned}
f &= .00769\partial^5 + (.00035t^2 + .05386t - .05386)\partial^4 \\
&\quad + (.00140t^3 + .06820t^2 - .16928t + .17313)\partial^3 \\
&\quad + (-.09513t^3 + .22559t^2 + .16928t - .33472)\partial^2 \\
&\quad + (.18607t^3 - .65720t^2 - .04617t + .32702)\partial \\
&\quad + (-.09234t^3 + .36305t^2 - .00769t - .11927). \\
g &= (.01001t - .01001)\partial^5 + (.04019t^2 - .07007t + .03003)\partial^4 \\
&\quad + (.00063t^3 - .01048t^2 + .15014t - .11010)\partial^3 \\
&\quad + (.27901t^3 - .32921t^2 - .09008t + .17016)\partial^2 \\
&\quad + (-.55990t^3 + .52909t^2 - .04004t - .08007)\partial \\
&\quad + (.28026t^3 - .22959t^2 + .04004t).
\end{aligned}$$

We compute initial guesses (removing content numerically where appropriate):

$$\begin{aligned}
h_{\text{guess}} &= .09285\partial^3 + (.37139t - .27854)\partial^2 + (-.74278t + .27854)\partial + (.37139t - .09285), \\
f^*_{\text{guess}} &= .08287\partial^2 + (.00377t^2 + .24862t - .33150)\partial \\
&\quad + (-2.05844 \times 10^{-10}t^3 - .24862t^2 + .91162t - .04144), \\
g^*_{\text{guess}} &= (.10780t - .10780)\partial^2 + (.00168t^2 + 8.67540 \times 10^{-9}t - 2.71283 \times 10^{-9})\partial \\
&\quad + (2.35115 \times 10^{-8}t^3 + .75463t^2 - .43122t + 6.78976 \times 10^{-8}).
\end{aligned}$$

[3] A proof-of-concept implementation of the algorithms is available at https://www.scg.uwaterloo.ca/software/ApproxOreFoCM-2019.tgz.

The quality of this initial guess is

$$\|f - f^*_{\text{guess}} h_{\text{guess}}\|_2^2 + \|g - g^*_{\text{guess}} h_{\text{guess}}\|_2^2 = 4.04506 \times 10^{-14}.$$

The condition number for the Hessian matrix evaluated at our initial guess is 18354.38336 and our smallest eigenvalue is .00314. Since $\nabla^2 \Phi$ is locally positive definite, we know that we will converge to a unique (local) minimum. The minimum we converge to is 2.33030×10^{-20}.

The exact GCRD in this example is $h = (\partial + 4t - 1)(\partial - 1)(\partial - 1)$.

Example 4.2 (Noise) In this example, we introduced normalized noise of size 10^{-5} to f and g.

$$\begin{aligned}
f &= .00583\partial^5 \\
&\quad + (-9.45614 \times 10^{-7} t^3 + .00027t^2 + .03498t - .03498)\partial^4 \\
&\quad + (-8.26797 \times 10^{-7} t^5 + .04743t^3 + .01113t^2 - .05247t + .07287)\partial^3 \\
&\quad + (-9.08565 \times 10^{-8} t^5 + .13885t^4 - .21623t^3 \\
&\quad + .30950t^2 - .17781t - .05247)\partial^2 \\
&\quad + (-.18655t^5 - .02226t^4 - .20166t^3 - .41974t^2 + .33812t - .10202)\partial \\
&\quad + (.18655t^5 - .30315t^4 + .43935t^3 - .22868t^2 - .13117t + .15740). \\
g &= (.00780t - .00779)\partial^5 \\
&\quad + (9.10928 \times 10^{-7} t^5 + 6.83196 \times 10^{-7} t^3 \\
&\quad + .02351t^2 - .07018t + .02729)\partial^4 \\
&\quad + (5.94796 \times 10^{-8} t^4 \\
&\quad + .02376t^3 - .07822t^2 + .12086t - .06238)\partial^3 \\
&\quad + .16326t^4 + .04654t^3 - .27267t^2 + .12476t + .03898)\partial^2 \\
&\quad + (-.21833t^5 - .10868t^4 - .05617t^3 + .63939t^2 - .38597t + .14036)\partial \\
&\quad + (.21833t^5 - .27291t^4 - .01462t^3 - .09418t^2 + .24952t - .12086).
\end{aligned}$$

We compute initial guesses (removing content numerically where appropriate):

$$\begin{aligned}
h_{\text{guess}} &= .11192\partial^3 + (.33514t - .22357)\partial^2 \\
&\quad + (-.44667t^2 - .22358t - .11327)\partial + .44754t^2 - .55869t + .22453, \\
f^*_{\text{guess}} &= (5.97992 \times 10^{-8} t^5 - .00001t^4 \\
&\quad + .05212 - 8.67362 \times 10^{-18} t^2 + 5.20417 \times 10^{-18} t)\partial^2 \\
&\quad + (-.0001t^5 - .00002t^4 - .00001t^3 + .00238t^2 + .15646t - .20842)\partial \\
&\quad + (.00003t^5 - +.41663t^3 - .15629t^2 + .57193t - .02463),
\end{aligned}$$

$$\begin{aligned} g^*_{\text{guess}} = (&-2.10091 \times 10^{-8} t^5 + -6.93889 \times 10^{-18} t^3 - 1.73472 \\ &\times 10^{-17} t^2 + .06967t - .06963)\partial^2 \\ &+ (.00002t^4 + .00001t^3 + .00146t^2 - .27937t + .10474)\partial \\ &+ (-.00004t^5 + .00002t^4 + .48596t^3 + .00189t^2 - .27763t - .00158). \end{aligned}$$

The quality of this initial guess is

$$\|f - f^*_{\text{guess}} h_{\text{guess}}\|_2^2 + \|g - g^*_{\text{guess}} h_{\text{guess}}\|_2^2 = .00003.$$

The condition number for the Hessian matrix evaluated at our initial guess is 21971.20356 and our smallest eigenvalue is .00818. Since $\nabla^2 \Phi$ is locally positive definite, we know that we will converge to a unique (local) minimum. The minimum we converge to is 1.06759×10^{-10}, which is roughly the amount of noise we added.

Example 4.3 (GCRD via LS) In this example, we added a noise factor of 10^{-4} to f and g. Performing linear algebra over $\mathbb{R}(t)$ produced completely unacceptable answers, so we used a least squares algorithm to compute an approximate GCRD.

$$\begin{aligned} f = &(.11329t^6 + .23414t^5 + .12840t^4 + .00755t^3 + .00005)\partial^3 \\ &+ (.00001t^6 + .23414t^5 + .59667t^4 \\ &+ .02269t^3 - .04528t^2 - .02266t + 3.67436 \times 10^{-7})\partial^2 \\ &+ (-.11329t^6 + .33231t^5 - .43054t^4 - .00754t^3 - .00003t^2 - .06798t + .00003)\partial \\ &(-.00001t^6 - .23414t^5 + .34741t^4 + .01510t^3 - .06799t^2 + .09064t + .00004). \\ g = &(.01938t^4 - .03876t^3 - .07752t^2 + .03876t + .05819)\partial^3 \\ &+ (.13567t^4 + .23252t^3 - .07750t^2 - .34879t + .29066)\partial^2 \\ &+ (-.01938t^4 + .13563t^3 + .03873t^2 + .25195t - .23257)\partial \\ &+ (-.13562t^4 + .44570t^3 - .56198t^2 - .03874t + .17439). \end{aligned}$$

Using a least squares variant of our numeric GCRD algorithm, we are able to compute (without removing content):

$$\begin{aligned} h_{\text{guess}} = &(t^2 + 1.94162t + .93768)\partial^2 + 2.87182\partial \\ &+ (-.94502t^2 + 2.84696t - 3.82712), \\ f^*_{\text{guess}} = &(.00712t^6 - .01655t^5 + .71917t^4 \\ &+ .05630t^3 - .00048t^2 - .00199t + .00155)\partial \\ &+ (-.00061t^6 - .01248t^5 + .02319t^4 \\ &+ .04309t^3 - .02430t^2 - .12387t - .00820), \\ g^*_{\text{guess}} = &(.00041t^4 - .00715t^3 + .13885t^2 - .50330t + .36942)\partial \\ &+ (.00381t^4 - .01139t^3 + .86465t^2 - .48856t + .00231). \end{aligned}$$

The quality of this initial guess is

$$\|f - f^*_{\text{guess}} h_{\text{guess}}\|_2^2 + \|g - g^*_{\text{guess}} h_{\text{guess}}\|_2^2 = .00328.$$

The condition number for the Hessian matrix evaluated at our initial guess is 148.62547 and our smallest eigenvalue is .04615. Since $\nabla^2 \Phi$ is locally positive definite, we know that we will converge to a unique (local) minimum. The minimum we converge to is 9.53931×10^{-9}.

4.4 General Examples

We provide results that demonstrate the robustness of our algorithms. We consider differential polynomials whose degrees in t and ∂ are balanced and unbalanced. The coefficients of the inputs were generated using the Maple routine `randpoly()`. The inputs f and g were normalized so that $\|f\| = \|g\| = 1$. We introduced normalized noise to the coefficients of f and g, so that the relative error is size of the perturbation. Precisely, if $f + \Delta f$ and $g + \Delta g$ are perturbed from f and g by the quantities Δf and Δg, then the relative error in the coefficients of f and g is given by $\|\Delta f\|_2 = \|\Delta g\|_2$.

We recall that the Newton iteration optimizes $\|f - \widetilde{f}\|_2^2 + \|g - \widetilde{g}\|_2^2$, which is the *sum of the squares* of the errors. The initial error and error from post-refinement are expressed as the sum of square errors accordingly. In our experiments, the *Initial Error* is the quantity $\|f - f_{init}\|_2^2 + \|g - g_{init}\|_2^2$ and the error after post-refinement, *Newton Error* is the quantity $\|f - f_{\text{opt}}\|_2^2 + \|g - g_{\text{opt}}\|_2^2$. In all of the examples when there were no perturbations in the coefficients of f and g, our numeric GCRD algorithm and post-refinement procedures were able to compute an exact GCRD to machine precision. When perturbations imposing a relative error of 10^{-8} in the coefficients of f and g were introduced, we were able to compute a solution to the approximate GCRD problem in every example.

Introducing perturbations imposing a relative error of order 10^{-4} and 10^{-2} into the coefficients of f and g prevented computation of an approximate GCRD in some examples. Instead, we provide examples of the largest perturbation in the coefficients of f and g that we were able to compute an approximate GCRD. Instances that are denoted as "FAIL" occur when the post-refinement did not converge. The implementation of Newton's method is not globalized to converge to a stationary point; hence, the iterates may diverge. In our examples, iterates diverge because the Hessian matrix is indefinite at an initial guess.

4.4.1 Balanced Degrees in t and ∂

The following results of experiments were conducted on differential polynomials whose degrees in t and ∂ were proportional, or balanced.

Example	Input (∂, t)	GCRD (∂, t)	noise	Initial error	Newton Error
1	(2,2)	(1,1)	1e−2	2.63579e−3	9.37365e−5
2	(2,2)	(1,1)	1e−2	6.98136e−4	8.96068e−5
3	(3,2)	(2,1)	1e−2	1.69968e−2	1.26257e−4
4	(3,4)	(2,2)	1e−2	3.8269e−3	1.04271e−4
5	(4,4)	(3,2)	1e−2	3.15314e−1	FAIL
5	(4,4)	(3,2)	1e−4	9.29336e−7	8.97294e−9

4.4.2 Unbalanced Degrees in ∂

The following results of experiments were conducted on differential polynomials whose degrees in ∂ were relatively larger than their degree in t.

Example	Input (∂, t)	GCRD (∂, t)	noise	Initial error	Newton Error
1	(2,2)	(1,1)	1e−2	1.13109e−3	2.90713e−5
2	(3,2)	(2,1)	1e−2	6.72179e−4	1.13998e−4
3	(4,2)	(3,1)	1e−2	3.00365e−4	1.04038e−4
4	(5,2)	(4,1)	1e−2	9.01982e−4	1.23557e−4
5	(6,2)	(5,1)	1e−2	6.61552e−3	FAIL
5	(6,2)	(5,1)	1e−4	2.74084e−4	1.12566e−8

4.4.3 Unbalanced Degrees in t

The following results of experiments were conducted on differential polynomials whose degrees in t were relatively larger than their degree in ∂.

Example	Input (∂, t)	GCRD (∂, t)	noise	Initial error	Newton Error
1	(2,3)	(1,2)	1e−2	1.27092e−2	1.43153e−4
2	(2,6)	(1,4)	1e−2	5.04286e−1	FAIL
2	(2,6)	(1,4)	1e−4	7.78993e−4	1.31180e−8
3	(2,8)	(1,6)	1e−4	6.9361e−2	FAIL
3	(2,8)	(1,6)	1e−8	3.92268e−10	1.15653e−16
4	(2,11)	(1,8)	1e−8	6.20749e−10	1.26549e−16
5	(2,13)	(1,10)	1e−8	2.23588e−10	1.03136e−16

5 Conclusion

In this paper, we have formally defined an approximate GCRD problem for differential polynomials, and given an approach to a robust numerical solution. We have seen that, under reasonable assumptions the approximate GCRD problem is well posed. In particular, we show that Newton iteration will converge to an optimal solution if

the residual is sufficiently small. We employ the earlier results in [9], analogous to SVD-based approximate GCD methods like Corless et al. [7], to compute a reasonable initial estimate for the Newton iteration. The results were presented for real differential polynomials; however, the results generalize in a very straight forward way to the instance of complex differential polynomials.

We believe that some aspects of our problems could also be approached from a structured low-rank approximation viewpoint Kaltofen et al. [15]; Schost and Spaenlehauer [28]. In particular, the work of Schost and Spaenlehauer [28] can be used to obtain an initial low-rank differential Sylvester matrix in which cofactors and a GCRD can be extracted for post-refinement. This holds more generally than differential polynomials, and a particular example to consider is the shift operator, commonly associated with linear difference equations.

Another area of future work is in the certification of the degree of an approximate GCRD. We can obtain a reasonable guess by enumerating over the degrees of all possible approximate GCRDs, similar to the structured total least norm approach adopted for multivariate polynomial approximate GCD Kaltofen et al. [16]. A possible direction would be to look at the differential subresultant sequence and the singular values of their inflated block matrices Emiris et al. [8].

The differential polynomials defined in this paper are special case of more general Ore polynomials, which have broader application in the solution of differential and difference equations. In particular, we could potentially apply our methods in the context of q-differentiation (Jackson differentiation) or derivations on exponential polynomials. Ultimately, any Ore structure will have a well-defined Sylvester-like matrix (see, e.g., Giesbrecht and Kim [10]). However, the numerical properties of different derivations may well be quite difficult or even problematic and may well introduce poles or other significant sources of numerical instability.

We also hope, the results of this paper are a foundation for extending the approximate polynomial toolbox to other problems with differential polynomials and more general linear differential operators. Much like approximate GCD, the approximate GCRD is both a stepping stone and a key tool toward operations like approximate factorization and (functional) solution of differential polynomials. More immediately, computation of an approximate GCRD enables computation of a corresponding approximate LCLM, and multiple GCRD's, and to multiple differential variables (i.e., iterated Ore polynomials), which provide an effective method for dealing with linear PDEs.

Acknowledgements The authors would like to thank George Labahn for his comments. The authors would also like to thank the two anonymous referees for their careful reading and comments.

References

1. Abramov S, Le H, Li Z (2005) Univariate Ore polynomial rings in computer algebra. J Math Sci 131(5):5885–5903
2. Bell J, Heinle A, Levandovskyy V (2017) On noncommutative finite factorization domains. Trans AMS 369:2675–2695
3. Botting B, Giesbrecht M, May J (2005) Using the Riemannian SVD for problems in approximate algebra. In: Proceedings workshop on symbolic-numeric computation (SNC'05), pp 209–219

4. Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge University Press, New York
5. Bronstein M, Petkovšek M (1994) On Ore rings, linear operators and factorisation. Programmirovanie 20:27–45
6. Bronstein M, Petkovšek M (1996) An introduction to pseudo-linear algebra. Theor Comput Sci 157(1):3-33
7. Corless RM, Gianni PM, Trager BM, Watt SM (1995) The singular value decomposition for polynomial systems. In: Proceedings international symposium on symbolic and algebraic computation (ISSAC'95), pp 189–205
8. Emiris IZ, Galligo A, Lombardi H (1997) Certified approximate univariate GCDs. J Pure Appl Algebra 117–118:229–251. https://doi.org/10.1016/S0022-4049(97)00013-3
9. Giesbrecht M, Haraldson J (2014) Computing GCRDs of approximate differential polynomials. In: Proceedings symposium on symbolic-numeric computation (SNC '14), pp 78–87
10. Giesbrecht M, Kim M (2013) Computing the hermite form of a matrix of Ore polynomials. J Algebra 376:341–362
11. Giesbrecht M, Heinle A, Levandovskyy V (2016) Factoring linear partial differential operators in n variables. J Symb Comput 75:127–148
12. Grigor'ev D (1990) Complexity of factoring and calculating the GCD of linear ordinary differential operators. J Symb Comput 10(1):7–37
13. Haraldson J (2015) Computing approximate GCRDs of differential polynomials. Master's thesis, University of Waterloo
14. Heinle A, Levandovskyy V (2016) A factorization algorithm for g-algebras and applications. In: Proceedings international symposium on symbolic and algebraic computation (ISSAC 16), ACM Press, pp 263–270
15. Kaltofen E, Yang Z, Zhi L (2005) Structured low rank approximation of a Sylvester matrix. In: Proceedings workshop on symbolic-numeric computation (SNC'05), pp 69–83
16. Kaltofen E, Yang Z, Zhi L (2006) Approximate greatest common divisors of several polynomials with linearly constrained coefficients and singular polynomials. In: Proceedings international symposium on symbolic and algebraic computation (ISSAC'06), pp 169–176
17. Kaltofen E, Yang Z, Zhi L (2007a) Approximate greatest common divisors of several polynomials with linearly constrained coefficients and singular polynomials, unpublished manuscript
18. Kaltofen E, Yang Z, Zhi L (2007b) Structured low rank approximation of a sylvester matrix. In: Symbolic-numeric computation, Birkhäuser Verlag, Basel, Switzerland, Trends in Mathematics, pp 69–83
19. Karmarkar N, Lakshman YN (1996) Approximate polynomial greatest common divisors and nearest singular polynomials. In: Proceedings international symposium on symbolic and algebraic computation (ISSAC'96), pp 35–39
20. Karmarkar N, Lakshman YN (1998) On approximate GCDs of univariate polynomials. J Symb Comput 26(6):653–666
21. Li Z (1998) A subresultant theory for Ore polynomials with applications. In: Proceedings international symposium on symbolic and algebraic computation (ISSAC'98), ACM, pp 132–139
22. Li Z, Nemes I (1997) A modular algorithm for computing greatest common right divisors of Ore polynomials. In: Proceedings international symposium on symbolic and algebraic computation (ISSAC'97), pp 282–289
23. Ore O (1933) Theory of non-commutative polynomials. Ann Math Second Ser 34:480–508
24. Rudin W (1976) Principles of mathematical analysis. . New York
25. Salvy B, Zimmermann P (1994) Gfun: a Maple package for the manipulation of generating and holonomic functions in one variable. ACM Trans Math Softw 20(2):163–177
26. Sasaki T, Sasaki M (1997) Polynomial remainder sequence and approximate GCD. ACM SIGSAM Bull 31:4–10
27. Schönhage A (1985) . J Complex 1:118–137
28. Schost É, Spaenlehauer P (2016) A quadratically convergent algorithm for structured low-rank approximation. Found Comput Math 16(2):457–492
29. von zur Gathen J, Gerhard J (2013) Modern Computer Algebra, 3rd edn. Cambridge University Press, New York
30. Zeng Z (2011) The numerical greatest common divisor of univariate polynomials. In: Randomization, relaxation, and complexity in polynomial equation solving, contemporary mathematics, vol 556. ACM Press, pp 187–217

31. Zeng Z, Dayton BH (2004) The approximate GCD of inexact polynomials. In: Proceedings international symposium on symbolic and algebraic computation (ISSAC'04), pp 320–327

Affiliations

Mark Giesbrecht[1] · Joseph Haraldson[1] · Erich Kaltofen[2]

✉ Mark Giesbrecht
mwg@uwaterloo.ca

Joseph Haraldson
jharalds@uwaterloo.ca

Erich Kaltofen
kaltofen@math.ncsu.edu

1 Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada

2 Department of Mathematics, North Carolina State University, Raleigh, USA