Univariate Ore Polynomial Rings in Computer Algebra

S. A. Abramov[*]
Dorodnicyn Computing Centre
Russian Academy of Science
Moscow, Russia
`abramov@ccas.ru`

H. Q. Le[†], Z. Li[†]
Symbolic Computation Group
University of Waterloo
Waterloo, Canada
{`hqle,z6li`}`@scg.math.uwaterloo.ca`

Abstract

We present some algorithms related to rings of Ore polynomials (or Ore rings for short), and describe a computer algebra library for basic operations in an arbitrary Ore ring. The library can be used as a basis for various algorithms in Ore rings, in particular in differential, shift, and q-shift rings.

1 Introduction

The theory of Ore rings opens an opportunity to consider linear ordinary differential, difference, q-difference, and other operators from a uniform standpoint. This theory was proposed by Ore [24, 25, 26] who described, in particular, a uniform theory of the operator factorization, which generalizes the theory by Landau and Loewy for the differential case [17, 21, 22]. A way to interpret abstract Ore polynomials as linear operators on a vector space was proposed by Jacobson in [16].

---

[*] Partially supported by the Russian Foundation for Basic Research, project 01-01-0047.

[†] Partially supported by Natural Sciences and Engineering Research Council of Canada Grant No. CRD215442-98.

The study of rings of Ore polynomials is attractive because it not only allows statements concerning operators of various kind to be proved in one stroke, but also allows one to design general-purpose algorithms and corresponding programs adjustable to a specific form of operators and equations. It is worth mentioning that the idea to use Ore rings in computer algebra was first employed by Bronstein and Petkovšek [8] where a general-purpose algorithm for factorization in an arbitrary Ore ring is described.

In this paper we describe a few (but far from all) computer algebra algorithms related to Ore rings. Section 2 provides an overview of rings of univariate Ore polynomials. The material in Section 3 on adjoint operators is presented in a more general form than before [3], while the one in Section 4 on an efficient computation of greatest common divisors (gcd) and least common multiplies (lcm) is presented for the first time. Section 5 gives an overview of the `OreTools` package which provides facilities for working with univariate Ore polynomials in the Maple computer algebra system [23]. A comparison between this package and other related packages is done in Sections 4 and 6. Information on the availability of the package is provided in Section 7.

2 Rings of univariate Ore polynomials

In Sections 2.1, 2.4 we give an overview of univariate Ore polynomials and pseudo-linear operators. See [24, 16, 8] for a detailed discussion on them, and proofs of corresponding statements. In Section 2.2 we discuss the idea of Hilbert twist reduction following the description from [10, 8]. In Section 2.3 the definition and basic properties of adjoint polynomials are given (see [10, Chaps. 1 and 8 (Sect. 3)] for details).

2.1 Ore polynomials

Let k be a commutative field of characteristic 0, and $\sigma : k \to k$ be an automorphism of k.

Definition 2.1 *A derivation w.r.t.* σ *is any mapping* $\delta : k \to k$ *satisfying*

$$\delta\,(a+b) = \delta\,a + \delta\,b \text{ and } \delta\,(ab) = \sigma(a)\,\delta\,b + \delta a\,b, \quad \text{for any } a, b \in k. \tag{1}$$

Definition 2.2 *The set of constants (with respect to* σ *and* δ*) is*

$$\mathrm{Const}_{\sigma,\delta}(k) = \{a \in k \,:\, \sigma(a) = a,\, \delta a = 0\}.$$

It can be shown that $\mathrm{Const}_{\sigma,\delta}(k)$ is a subfield of k.

The following lemma describes the relation between σ and δ. When no confusion arises, we denote by 1 the identity mapping of k.

Lemma 2.1 *Let δ be a derivation of k w.r.t. σ.*

(i) *If $\sigma \neq 1$ then there is an element $\alpha \in k$ such that $\delta = \alpha(\sigma - 1)$.*

(ii) *If $\delta \neq 0$ then there is an element $\beta \in k$ such that $\sigma = \beta\,\delta + 1$.*

Example 2.1 Let $k = \mathbb{F}(t)$ in cases 1–4, and $k = \mathbb{F}(q,t)$ in cases 5–7 where $\mathbb{F}$ is any subfield of $\mathbb{C}$.

	Case	σ	δ
1	differential	1	$\frac{d}{dt}$
2	Eulerian differential	1	$t\frac{d}{dt}$
3	shift	E	0
4	difference	E	$E-1$
5	q-shift	Q	0
6	q-difference	Q	$Q-1$
7	q-differential	Q	$\frac{Q-1}{t(q-1)}$

Definition 2.3 *A (univariate) Ore ring over k, given by σ and δ and denoted by $k[x;\sigma,\delta]$, is the ring of polynomials in x over k with the usual polynomial addition, and the multiplication given by*

$$x\,a = \sigma(a)\,x + \delta a, \quad \textit{for any } a \in k. \tag{2}$$

Elements of the ring $k[x;\sigma,\delta]$ are called Ore polynomials. Note that it is possible to consider k as a ring (we will consider Ore polynomials over a ring in Sections 2.4, 3 and 4).

Let $p(x) \in k[x;\sigma,\delta]$, and $p(x) = p_m x^m + \cdots + p_1 x + p_0$, $p_m \neq 0$, then $m = \deg p(x)$, $p_m = \mathrm{lc}\, p(x)$. Set $\deg 0 = -\infty$, $\mathrm{lc}\, 0 = 0$. It can be shown that $k[x;\sigma,\delta]$ possesses the right and left division algorithms. Let $a, b \in k[x;\sigma,\delta] \setminus \{0\}$. By applying the right division algorithm, we get

$$a = q_1\, b + r_1, \;\; q_1, r_1 \in k[x;\sigma,\delta], \; \deg\, r_1 < \deg\, b.$$

r_1, q_1 are called the *right-remainder* and the *right-quotient* of a by b, respectively. Similarly, by applying the left division algorithm, we get

$$a = b\, q_2 + r_2, \;\; q_2, r_2 \in k[x;\sigma,\delta], \; \deg\, r_2 < \deg\, b.$$

r_2, q_2 are called the *left-remainder* and the *left-quotient* of a by b, respectively.

For given $a, b \in k[x; \sigma, \delta]$, one can find the *greatest common right divisor* (gcrd) by the right Euclidean algorithm, and the *least common left multiple* (lclm) by the *extended* right Euclidean algorithm. Computing the *greatest common left divisor* (gcld) and, resp., the *least common right multiple* (lcrm) can be reduced to computing the gcrd and, resp., the lclm using adjoint.

2.2 Hilbert twist reduction

The Hilbert twist reduction is a ring isomorphism which maps a general Ore ring to a ring with trivial derivation provided that σ is non-trivial.

Proposition 2.2 *If there exists* $\alpha \in k$ *such that* $\alpha \neq \sigma(\alpha)$*, then the bijection* $H_\alpha : k[x; \sigma, \delta] \to k[y; \sigma, 0]$*, given by*

$$H_\alpha\left(\sum_i a_i x^i\right) = \sum_i a_i \left(\frac{y + \delta\alpha}{\alpha - \sigma(\alpha)}\right)^i,$$

is a ring isomorphism.

2.3 Adjoint polynomials

Definition 2.4 *Let* $k[x; \sigma, \delta]$ *be an Ore ring. The adjoint of* $k[x; \sigma, \delta]$ *is defined by the Ore ring* $k[x; \sigma^*, \delta^*]$ *where*

$$\sigma^* = \sigma^{-1}, \;\; \delta^* = -\delta\, \sigma^{-1}. \tag{3}$$

Let $a = a_n\, x^n + \cdots + a_1\, x + a_0 \in k[x; \sigma, \delta]$*. The adjoint polynomial* a^* *is defined by*

$$a^* = x^n\, a_n + \cdots + x\, a_1 + a_0 \in k[x; \sigma^*, \delta^*].$$

Note that the product $x^i\, a_i$ must be computed in the Ore ring $k[x; \sigma^*, \delta^*]$. It is easy to show that $\mathrm{Const}_{\sigma,\delta}(k) = \mathrm{Const}_{\sigma^*,\delta^*}(k)$ and $(\sigma^*)^* = \sigma, \; (\delta^*)^* = \delta$. One can also verify that the adjoint is a linear (over $\mathrm{Const}_{\sigma,\delta}$) bijective mapping, and $(a^*)^* = a, \;\; (ab)^* = b^* a^*$. Additionally,

$$\mathrm{gcld}(a, b) = (\mathrm{gcrd}(a^*, b^*))^*, \;\; \mathrm{lcrm}(a, b) = (\mathrm{lclm}(a^*, b^*))^*.$$

Example 2.2 It follows from Example 2.1 and Definition 2.4 that

	Case	σ^*	δ^*
1	differential	1	$-\frac{d}{dt}$
2	Eulerian differential	1	$-t\,\frac{d}{dt}$
3	shift	E^{-1}	0
4	difference	E^{-1}	$E^{-1}-1$
5	q-shift	Q^{-1}	0
6	q-difference	Q^{-1}	$Q^{-1}-1$
7	q-differential	Q^{-1}	$\frac{Q^{-1}-1}{t(q-1)}$

2.4 Ore polynomials as linear operators

Definition 2.5 *Let V be a vector space over k. A mapping $\theta : V \to V$ is pseudo-linear w.r.t. σ and δ if*

$$\theta(u+v) = \theta(u) + \theta(v), \quad \theta(a\,u) = \sigma(a)\,\theta(u) + \delta a\,u \tag{4}$$

for any $a \in k$, $u, v \in V$.

Assume K to be a σ, δ-compatible extension ring of k, i.e., σ and δ can be extended to an automorphism of the ring K, and a derivation of the ring w.r.t. σ, respectively. We also assume that $\mathrm{Const}_{\sigma,\delta}(K) = \mathrm{Const}_{\sigma,\delta}(k)$, and write C for this field for short. Note that K is a vector space over k, and hence can play the role of V. We will consider pseudo-linear mappings from K to K, assuming that relations (4) are valid for any $a, u, v \in K$.

Lemma 2.3 *For any $c \in K$, the mapping $\theta_c : K \to K$ given by*

$$\theta_c(a) = c\,\sigma(a) + \delta a \tag{5}$$

is K-pseudo-linear w.r.t. σ and δ, and $\theta_c(1) = c$. Conversely, for any K-pseudo-linear mapping θ, the element $c = \theta(1)$ is such that $\theta = \theta_c$ given in (5).

Consider the ring $k[\theta]$ of C-linear operators $L : K \to K$ of the form $L = p(\theta)$, $p(x) \in k[x; \sigma, \delta]$. The correspondence $p(x) \to p(\theta)$ provides us with an ring homomorphism $\Theta : k[x; \sigma, \delta] \to k[\theta]$ due to the pseudo-linearity of θ. We assume that

$$p(\theta) \text{ is the zero operator on } K \text{ iff } p(x) \text{ is the zero Ore polynomial,} \tag{6}$$

and as a consequence, the correspondence $p(x) \to p(\theta)$ gives a ring isomorphism. If $L = p(\theta)$, then we set $\operatorname{ord} L = \deg p$.

Sometimes it is convenient to consider the rings $K[x;\sigma,\delta]$ and $K[\theta]$ as well. We assume that (6) is valid for them.

It is easy to give an example which shows that (6) is not valid in the general case (say, $K=k=\mathbb{C}$, $\sigma(z)=\bar{z}$, $\delta=0$, $\theta=\sigma$). In Proposition 3.2 we will formulate a natural simple sufficient condition of (6).

As a consequence of Lemmas 2.1 and 2.3 we get

Proposition 2.4 *A K-pseudo-linear mapping θ w.r.t. σ and δ, is equal to $\delta+\theta(1)$ if $\sigma=1$ and to $(\theta(1)+\alpha)\sigma-\alpha$ if $\sigma\neq 1$, where α is in Lemma 2.1 (i).*

The assumption (6) is not valid in the case $\theta(1)+\alpha=0$ (otherwise, $\theta+\alpha=0$), so we derive that if $\sigma\neq 1$, then $(\theta(1)+\alpha)\sigma-\alpha$ and $\theta(1)+\alpha\neq 0$. We will additionally assume that $\theta(1)+\alpha$ is not a zero divisor in K (it is valid, e.g., if $\theta(1)\in k$ and as a consequence $\theta(1)+\alpha\in k$).

Example 2.3 The following table provides information on the pseudo-linear mapping θ and $c=\theta(1)$.

	Case	θ	c
1	differential	$\frac{d}{dt}$	0
2	Eulerian differential	$t\,\frac{d}{dt}$	0
3	shift	E	1
4	difference	$E-1$	0
5	q-shift	Q	1
6	q-difference	$Q-1$	0
7	q-differential	$\frac{Q-1}{t(q-1)}$	0

3 Adjoint operators

3.1 Operator ∇

Let θ be a pseudo–linear mapping from K to K w.r.t. σ,δ. Set $\nabla=\theta-\theta(1)$, $\nabla\in K[\theta]$. By Proposition 2.4 we have

$$\nabla=\begin{cases}\delta & \text{if } \sigma=1,\\ (\theta(1)+\alpha)(\sigma-1) & \text{if } \sigma\neq 1,\end{cases}$$

and by Lemma 2.1 and since $\theta(1)+\alpha$ is not a zero divisor, for any $f\in K$

$$\nabla(f)=0 \text{ iff } f\in C. \tag{7}$$

It is easy to derive from this that for $L \in K[\theta]$ we have $L(1) = 0$ iff there exists $M \in K[\theta]$ such that $L = M\nabla$. Taking additionally into account that $L(f) = (Lf)(1)$ and assumption (6), we get

Proposition 3.1 *Let $p \in K[x; \sigma, \delta]$, $L = p(\theta), f \in K$. Then $L(f) = 0$ iff there exists $M \in K[\theta]$ such that $Lf = M\nabla$, i.e., iff pf is right divisible by $x - \theta(1)$.*

Let $c = \theta(1)$ and $p \in K[x; \sigma, \delta] \setminus \{0\}$, $\deg p = d$, then there exists a non-negative integer n such that $p = (b_{d-n}(x-c)^{d-n} + \cdots + b_1(x-c) + b_0)(x-c)^n$, where $b_0, \ldots, b_{d-n} \in K$, $b_0 \neq 0$. This gives us

Proposition 3.2 *Suppose that for any non-negative integer n there exists $f \in K$ such that $\nabla^n(f) \in C \setminus \{0\}$. Then (6) is valid for any $p \in K[x; \sigma, \delta]$.*

Example 3.1 It follows from Example 2.3 that

	Case	∇
1	differential	$\frac{d}{dt}$
2	Eulerian differential	$t\,\frac{d}{dt}$
3	shift	$E - 1$
4	difference	$E - 1$
5	q-shift	$Q - 1$
6	q-difference	$Q - 1$
7	q-differential	$\frac{Q-1}{t(q-1)}$

3.2 Adjoint operators and integrating factors

From Lemma 2.3, $\theta = \theta_c = c\,\sigma + \delta$, where $c = \theta(1)$. Set $\theta^* = c\,\sigma^* + \delta^*$, where σ^*, δ^* are as in (3). Note that $\theta(1) = c = \theta^*(1)$.

Definition 3.1 *Let $k[x; \sigma, \delta]$ be an Ore ring and θ a pseudo-linear mapping w.r.t. σ, δ. The adjoint ring of $k[\theta]$ is defined to be the operator ring $k[\theta^*]$. If $p \in k[x; \sigma, \delta]$ and $L = p(\theta)$, then the adjoint operator for L is defined to be $L^* = p^*(\theta^*) \in k[\theta^*]$.*

We have $(LM)^* = M^*L^*$ for any $L, M \in k[\theta]$. If we suppose that (6) is valid for $K[x; \sigma^*, \delta^*]$ and $K[\theta^*]$, then we additionally have $(L^*)^* = L$ for any $L \in k[\theta]$.

Consider the operator $\nabla^* = \theta^* - \theta(1) = \theta^* - \theta^*(1)$. By Proposition 3.1 we have $L^*(f) = 0$ iff there exists $M \in K[\theta^*]$ such that $L^*f = M\nabla^*$, i.e., $fL = \nabla M^*$. This gives us

Proposition 3.3 *Let* $p \in K[x;\sigma,\delta]$, $L = p(\theta), f \in K$. *Then* $L^*(f) = 0$ *iff there exists* $N \in K[\theta]$ *such that* $fL = \nabla N$, *i.e., iff* fp *is left divisible by* $x - \theta(1)$.

Propositions 3.1 and 3.3 present an analog of the Bezout theorem for algebraic equations in one unknown.

Example 3.2 It follows from Examples 2.2 and 2.3 that

	Case	θ^*	∇^*
1	differential	$-\frac{d}{dt}$	$-\frac{d}{dt}$
2	Eulerian differential	$-t\,\frac{d}{dt}$	$-t\,\frac{d}{dt}$
3	shift	E^{-1}	$E^{-1}-1$
4	difference	$E^{-1}-1$	$E^{-1}-1$
5	q-shift	Q^{-1}	$Q^{-1}-1$
6	q-difference	$Q^{-1}-1$	$Q^{-1}-1$
7	q-differential	$\frac{Q^{-1}-1}{t(q-1)}$	$\frac{Q^{-1}-1}{t(q-1)}$

It is logical to call an *integrating factor* for L any $f \in K$ such that $fL = \nabla N$, $N \in K[\theta]$. Proposition 3.3 is an analog of the classical theorem from the theory of ordinary differential equations, but the statement of this proposition has a general "Ore form".

Example 3.3 Let $k = \mathbb{C}(n), \sigma = \theta = E, \delta = 0$, $\nabla = E - 1$, and K be the ring of sequences whose elements are in $\mathbb{C}$. Consider the operator

$$L = (n+4)\,E^2 + E - (n+1) \in k[\theta].$$

The corresponding adjoint equation $L^*(f) = 0$ is

$$L^*(f) = -(n+1)\,f(n) + f(n-1) + (n+2)\,f(n-2) = 0. \tag{8}$$

Therefore, an integrating factor f for L can be computed, if the factor is hypergeometric, by applying algorithm Hyper [28] to (8) which yields $f = (-1)^n$. As a consequence we have

$$(-1)^n L = (E-1)((-1)^{n-1}(n+3)E + (-1)^n(n+1)).$$

3.3 Accurate integration

An element $g \in K$ is a *primitive* of $f \in K$ if $\nabla(g) = f$. Suppose that $\theta(1) \in k$ (i.e., $\nabla \in k[\theta]$) and consider the following problem: Let $f \in K$ and the minimal annihilating operator $L \in k[\theta]$ for f be given. So $n = \operatorname{ord} L$ is minimal with the property that $L \in k[\theta]$ and $L(f) = 0$. Decide whether there exists a primitive g of f such that the minimal annihilating operator $\tilde{L}$ for g has order n. If so, then construct such a g together with its minimal annihilating operators.

This problem (the problem of *accurate integration*) was solved in [3]. The adjoint operators play a key role in the solution. We give below a short description of the algorithm. Note that in [3] a description is given for two (principal) cases: $\sigma = 1, \theta = \delta$ and $\theta = \sigma - 1, \delta = 0$. If the problem has a positive solution (the operator $\tilde{L}$ of order n does exist) then the algorithm constructs $r \in k[\theta]$, $\operatorname{ord} r = n - 1$, such that $g = r(f)$, together with $\tilde{L}$.

It was shown in [3] that $\tilde{L}$ such that $\operatorname{ord} \tilde{L} = n$ exists iff the equation $L^*(y) = 1$ has a solution l in k. In this case r is such that $1 - lL = \nabla r$ (so r can be found by the left division) and $\tilde{L} = 1 - r\nabla$. If such l does not exists then the integrating operator r does not exists either, while the minimal annihilator $\tilde{L}$ for g is $L\nabla$, $\operatorname{ord} \tilde{L} = n + 1$.

As mentioned in [3], this algorithm generalizes Gosper's algorithm for hypergeometric indefinite summation [14] in two ways: (a) it solves the analogous problem for a wider class of equations, and (b) it works for any order n, instead of only for $n = 1$.

Example 3.4 We show in this example the use of accurate integration in computing primitives of

$$p_1 = (27\,t^2 + 4)^{5/4}\, \mathcal{P}^{5/2}_{2/3\sqrt{7}-1/2}\left(-\frac{3}{2}\sqrt{3}i\,t\right),$$

$$p_2 = (27\,t^2 + 4)^{5/4}\, \mathcal{Q}^{5/2}_{2/3\sqrt{7}-1/2}\left(-\frac{3}{2}\sqrt{3}i\,t\right).$$

Both p_1 and p_2 are annihilated by the differential operator

$$L = \left(27\,t^2 + 4\right) D^2 - 81\,tD + 24.$$

The corresponding adjoint equation $L^*(y) = 1$ is

$$\left(27\,t^2 + 4\right) \frac{d^2}{dt^2} y\left(t\right) + 189\,t \frac{d}{dt} y\left(t\right) + 159\,y(t) = 1,$$

which admits $l = 1/159$ as a rational solution [1]. Therefore, the operator $r \in k[\theta]$ such that $\int p_1 dt = r(p_1)$ and $\int p_2 dt = r(p_2)$ is the left quotient of $1 - lL$ by ∇ which is

$$\left(-\frac{9}{53}t^2 - \frac{4}{159}\right)D + \frac{45}{53}t.$$

Note that both Maple 8 and Mathematica 4 are unable to compute these two indefinite integrals.

4 New modular techniques for gcd and lcm computations

The usual polynomial ring $k[x]$ is a special case of Ore polynomial rings. Various efficient techniques in the commutative $k[x]$ have been generalized to the non-commutative $k[x;\sigma,\delta]$ (see [15, 19, 20, 12]). In this section, we present new modular techniques for computing gcrd and lclm of Ore polynomials. There are some minor restrictions on the coefficient field in order to apply modular techniques. Let $\mathbb{D}$ be either the ring $\mathbb{Z}$ of the integers, or the ring of polynomials in several variables over $\mathbb{Z}$. Let t be a new indeterminate over $\mathbb{D}$, and $\mathbb{D}[t]$ the ring of usual commutative polynomials in t over $\mathbb{D}$. We shall work in the Ore ring $\mathbb{D}[t][x;\sigma,\delta]$ whose constant ring contains $\mathbb{D}$. Note that σ is an automorphism of $\mathbb{D}[t]$.

4.1 Gcrd computation

Let p be a prime. A ring homomorphism ϕ_p from $\mathbb{D}[t]$ to $\mathbb{Z}_p[t]$ is said to be *modular w.r.t.* σ if $\phi_p(\mathbb{D}) = \mathbb{Z}_p$, $\phi_p(t) = t$, and $\deg_t(\sigma(t)) = \deg_t \phi_p(\sigma(t))$. Define the automorphism σ_p of $\mathbb{Z}_p[t]$ by sending t to $\phi_p(\sigma(t))$, and any element of $\mathbb{Z}_p$ to itself. Furthermore, define the additive mapping δ_p from $\mathbb{Z}_p[t]$ to itself by sending t^n to $\phi_p(\delta(t^n))$, for $n \in \mathbb{N}$. It is straightforward to verify that the diagrams

$$\begin{tikzcd}
\mathbb{D}[t] \arrow[r, "\sigma"] \arrow[d, "\phi_p"] & \mathbb{D}[t] \arrow[d, "\phi_p"] & \mathbb{D}[t] \arrow[r, "\delta"] \arrow[d, "\phi_p"] & \mathbb{D}[t] \arrow[d, "\phi_p"] \\
\mathbb{Z}_p[t] \arrow[r, "\sigma_p"] & \mathbb{Z}_p[t] & \mathbb{Z}_p[t] \arrow[r, "\delta_p"] & \mathbb{Z}_p[t]
\end{tikzcd}$$

are commutative, and that $\mathbb{Z}_p[t][x, \sigma_p, \delta_p]$ is an Ore ring. The modular homomorphism ϕ_p can be extended to a mapping from $\mathbb{D}[t][x, \sigma, \delta]$ to $\mathbb{Z}_p[t][x, \sigma_p, \delta_p]$ by sending $\sum_i a_i x^i$ to $\sum_i \phi_p(a_i) x^i$, where $a_i \in \mathbb{D}[t]$. This extended mapping will also be denoted by ϕ_p, which is a ring homomorphism by a direct verification.

Let e be an element of $\mathbb{Z}_p$. By an evaluation mapping ψ_e from $\mathbb{Z}_p[t]$ to $\mathbb{Z}_p$ we mean a mapping that sends $\sum_i m_i t^i$ to $\sum_i m_i e^i$, where $m_i \in \mathbb{Z}_p$. Such an evaluation mapping can be extended to a mapping from $\mathbb{Z}_p[t][x, \sigma_p, \delta_p]$ to $\mathbb{Z}_p[x]$ by sending $\sum_i a_i x^i$ to $\sum_i \psi_e(a_i) x^i$, where $a_i \in \mathbb{Z}_p[t]$. The extended mapping is again denoted by ψ_e.

Example 4.1 *Consider the differential ring $D = \mathbb{Z}_p[t][x; 1, \frac{\mathrm{d}}{\mathrm{d}t}]$ and an evaluation mapping ψ_e. If ψ_e is a ring homomorphism from D to $\mathbb{Z}_p[x]$, on which some multiplication is defined. We then have $\psi_e(xt) = \psi_e(tx + 1) = ex + 1$, and, on the other hand*

$$\psi_e(xt) = \psi_e(x)\psi_e(t) = xe = x(\underbrace{1 + \cdots + 1}_{e\,\mathrm{times}}) = ex.$$

This leads to a contradiction.

Thus, no matter how we define a multiplication on $\mathbb{Z}_p[x]$, ψ_e is usually not a ring homomorphism. It is merely a module homomorphism from the left module $\mathbb{Z}_p[t][x]$ over $\mathbb{Z}_p[t]$ to $\mathbb{Z}_p[t]$ over $\mathbb{Z}_p$.

A key problem in modular gcrd-methods is

Problem E. Given P_1, P_2 in $\mathbb{Z}_p[t][x, \delta_p, \sigma_p]$, and an evaluation mapping ψ_e, compute the image of $\mathrm{gcrd}(P_1, P_2)$ under ϕ_e

The algorithm GCRD_e described in [20] solves Problem E. Let $\deg P_i = n_i$, $i = 1, 2$, $n = \max(n_1, n_2)$, $n_t = \max(\deg_t P_1, \deg_t P_2)$ and $G = \mathrm{gcrd}(P_1, P_2)$ with degree g. The number of the ψ_e, for which GRCD_e produces incorrect images or failure, is no more than $(n_1 + n_2)n_t$. Hence, GCRD_e produces sufficiently many correct images for the combining process when the prime p is large enough. The cost of GCRD_e is dominated by $\left(n_t n^2 + n^3\right)$ in the differential case. The factor n^3 comes from a row-reduction process on the Sylvester matrix of P_1 and P_2, which has $(n_1 + n_2)$ rows and $(n_1 + n_2)$ columns. We will present an improved GCRD_e whose cost is dominated by $\left(n_t(n - g)^2 + (n - g)^3\right)$. This improvement allows our modular gcrd method to work efficiently even when g is quite large. Roughly speaking, the improved algorithm is a carefully-designed row reduction process on the matrix associated with $\mathrm{sres}_{g-1}(P_1, P_2)$ which has $(n_1 + n_2 - 2(g-1))$ rows and $(n_1 + n_2 - g + 2)$ columns.

To describe the improvement, we need some terminologies. The reader is referred to [19] for the definition of subresultants of P_1 and P_2 and related notation. Recall that the mth subresultant of P_1 and P_2 is denoted by S_m, for $m = n_2, n_2 - 1, \ldots, 0$. A pair of consecutive subresultants S_m and S_{m+1} is said to be a *gcrd pair* of P_1 and P_2 *with index* m if $\deg S_m = m$ and $S_{m+1} = 0$. It follows directly from Theorem 4.2 in [19] and the gap-structure of a subresultant chain that

Proposition 4.1 *Let P_1, P_2 in $\mathbb{Z}_p[t][x, \delta_p, \sigma_p]$ with degrees n_1 and n_2, respectively, where $n_1 \geq n_2 > 0$. Then P_1 and P_2 have a gcrd pair iff the* gcrd *of P_1 and P_2 is of positive degree. A gcrd pair is unique when existent.*

Given the sequence:

$$x^{n_2-1}P_1, \ldots, xP_1, P_1, \; x^{n_1-1}P_2, \ldots, xP_2, P_2, \tag{9}$$

an evaluation mapping ψ_e is said to be *proper* w.r.t. P_1 and P_2 if

$$\deg \psi_e(x^i P_1) = (n_1 + i), \quad \text{for } i = 0, \ldots, (n_2 - 1), \text{ and}$$

$$\deg \psi_e(x^j P_2) = (n_2 + j), \quad \text{for } j = 0, \ldots, (n_1 - 1).$$

A proper ψ_e w.r.t. P_1 and P_2 is said to be *unlucky* if $\deg \psi_e(S_m) < \deg S_m$ for some nonzero S_m. Note that this definition is less restrictive than that of unlucky evaluation mappings in [20]. A pair of images of consecutive subresultant S_m and S_{m+1} under ψ_e is said to be a *pseudo-gcrd pair with index* m if $\deg \psi_e(S_m) = m$ and $\psi_e(S_{m+1}) = 0$.

Proposition 4.2 *Let P_1, P_2 in $\mathbb{Z}_p[t][x, \delta_p, \sigma_p]$ with degrees n_1 and n_2, respectively, where $n_1 \geq n_2 > 0$. Let G be the monic gcrd of P_1 and P_2 with degree g. Let ψ_e be a proper evaluation mapping w.r.t. P_1 and P_2. Then*

1. *If ψ_e is not unlucky and g is positive, then $(\psi_e(S_g), \psi_e(S_{g-1}))$ is the only pseudo-gcrd pair of P_1 and P_2 under ψ_e , and $\psi_e(G)$ is the monic associate of $\psi_e(S_g)$*

2. *If ψ_e is not unlucky and g is zero, then P_1 and P_2 do not have any pseudo-gcrd pairs, and $\phi_e(S_0)$ is nonzero.*

3. *If ψ_e is unlucky and P_1, P_2 have a pseudo-gcrd pair $(\psi_e(S_m), \psi_e(S_{m+1}))$ then $m \geq g$. In the case $m = g$, $\psi_e(G)$ is still the monic associate of $\psi_e(S_g)$.*

Proof. The first and second assertions follow from Proposition 4.1 and the fact that ψ_e maps the subresultant chain of P_1 and P_2 in a degree-preserving manner. The last follows from the fact that $\text{sres}_{g-1}(P_1, P_2)$, $\text{sres}_{g-2}(P_1, P_2)$, $\ldots$, and $\text{sres}_0(P_1, P_2)$ are all equal to 0. $\square$

For given a sequence (9) and a proper evaluation mapping ψ_e w.r.t. P_1 and P_2, we search for a pseudo-gcrd pair in the sequence $\psi_e(P_2)$, $\psi_e(S_{n_2-1})$, $\psi_e(S_{n_2-2})$, $\psi_e(S_0)$. We begin to evaluate the matrix M_{n_2-1} associated with S_{n_2-1}. Compute $H_{n_2-1} = \psi_e(S_{n_2-1})$ by Gaussian elimination on the rows of $\psi_e(M_{n_2-1})$. If H_{n_2-1} is zero, we get a pseudo-gcrd pair $(\psi_e(P_2), H_{n_2-1})$ and return the monic associate of $\psi_e(P_2)$. Otherwise, let $d = \deg H_{n_2-1}$.

We need to compute only $H_d = \psi_e(S_d)$ by Theorem 4.2 in [19]. If the degree of H_d is less than d, then ψ_e is unlucky, report failure. Otherwise, we compute $H_{d-1} = \psi_e(S_{d-1})$ by Gaussian elimination on the rows of the matrix associated with $\psi_e(S_{d-1})$. If $H_{d-1} = 0$, we get a pseudo-gcrd pair (H_d, H_{d-1}) and return the monic associate of H_d. Otherwise update d to be $\deg H_{d-1}$, and repeat the process. If no pseudo-gcrd pair is found, we will eventually compute $H_0 = \psi_e(S_0)$. If $H_0 \neq 0$, then return 1 (in this case P_1 and P_2 have the trivial gcrd). Otherwise, report failure (in this case e must be unlucky).

The above-described process may output: either a monic polynomial H in $\mathbb{Z}_p[x]$ with positive degree, or 1, or failure. In the first case H is either the image of G under ψ_e or $\deg H > g$, which implies that ψ_e is unlucky by Proposition 4.2. In the second case, G is trivial. In the last case, ψ_e is unlucky. There are at most $n_2^2(n_1 + n_2)n_t$ unlucky evaluation mappings. Since the matrix M_i associated with S_i is a submatrix of the matrix M_j associated with S_j when $i > j$, the results obtained by Gaussian elimination on M_i can be recycled for the Gaussian elimination on M_j. Thus, the cost for computing $\psi_e(S_{n_2-1})$, $\psi_e(S_{n_2-2})$, $\ldots$, $\psi_e(S_{g-1})$ is the same as the cost for computing $\psi_e(S_{g-1})$ by Gaussian elimination. The latter cost is dominated by $(n_t(n-g)^2 + (n-g)^3)$, in which $n_t(n-g)^2$ is the cost for computing $\psi_e(M_{g-1})$ and $(n-g)^3$ is the cost for Gaussian elimination on M_{g-1} in the differential case. After replacing GCRD_e by the above process, we see an overall improvement of the modular gcrd method when g is close to n_2.

Experiment 1 For computing gcrd of two given Ore polynomials p_1 and p_2, three different methods are implemented: Euclidean, fraction-free, and modular. A heuristic is carried out to choose one of these three methods. It is based on a guess for the degree of the $\text{gcrd}(p_1, p_2)$.

Table 1 shows the timing comparison of our experiment [1]. A set of 10 pairs of polynomials in the differential ring is randomly generated. For each pair of polynomials p_1 and p_2, the following constraints are imposed:

$$\deg p_1, \deg p_2 \leq 17, \quad \deg \operatorname{gcrd}(p_1, p_2) \geq 2.$$

Note that we also include the time taken by the function `DEtools[GCRD]`.

Table 1: gcrd computation: timing (in seconds) for different methods.

	Euclidean	Fraction-free	Modular	Heuristic	DEtools
1	65.72	27.09	16.57	16.94	33.30
2	184.96	56.11	28.64	29.44	49.85
3	168.88	103.03	31.87	32.10	55.60
4	221.47	166.94	43.09	43.89	70.11
5	25.06	22.58	21.43	22.14	14.94
6	65.61	53.16	33.70	31.92	30.27
7	123.79	79.32	40.87	41.96	37.97
8	148.57	68.68	33.89	35.05	52.83
9	28.71	14.76	15.42	15.63	15.79
10	120.57	85.44	27.54	28.65	59.24

If $\operatorname{gcrd}(p_1, p_2)$ is trivial, then the modular method is considerably faster than any non-modular method. This is because the modular method can detect if p_1 and p_2 are relatively prime by a lucky modular homomorphism and a lucky evaluation mapping. If $\operatorname{gcrd}(p_1, p_2)$ is nontrivial, experimental results show that the efficiency of the modular method depends on the following factors:

- how many divisions it takes to compute $\operatorname{gcrd}(p_1, p_2)$ in the right Euclidean algorithm;
- how "simple" $\operatorname{gcrd}(p_1, p_2)$ is.

By "simple" we mean that coefficients are of low degree and with short integral coefficients. The more divisions it takes, the more work the non-modular methods will do. The simpler the $\operatorname{gcrd}(p_1, p_2)$ is, the fewer images

---

[1] All the reported timings were obtained on a 400Mhz SUN SPARC SOLARIS with 1Gb RAM.

are needed to recover the true gcrd in a modular method. The modular method appears to be very stable with respect to different types of data, because it does not cause any intermediate expression swell.

4.2 Lclm computation

Next, we apply modular techniques to the lclm computation. Let $P_1, \ldots, P_m$ be in $\mathbb{D}[t][x;\sigma,\delta]$ with respective positive degrees $d_1, \ldots, d_m$. Let L be the lclm of $P_1, \ldots, P_m$. To compute L, we may first compute the lclm L_{12} of P_1 and P_2, and then compute the lclm of $L_{12}, P_3, \ldots, P_m$, recursively (on m). This "nested" algorithm does not work very well in practice, partly because the coefficients of the intermediate lclm's are usually far more complicated than those of the P_i's.

The code *LCLM* in the Maple package `DEtools` written by van Hoeij provides a direct method for computing the lclm of several Ore polynomials. The method works as follows. Let

$$d = d_1 + \cdots + d_m \quad \text{and} \quad Q_d = q_d x^d + \cdots + q_0$$

where $q_0, \ldots, q_d$ are unspecified coefficients. For $i = 1, \ldots, m$, compute the right remainder R_i of Q_d and P_i. Clearly, Q_d is a common left multiple of $P_1, \ldots, P_m$ with degree no more than d iff $R_1 = \cdots = R_m = 0$. This gives rise to a linear homogeneous algebraic system

$$(q_0, \ldots, q_d) M_d = 0 \tag{10}$$

where M is a $(d+1) \times d$ matrix over k. For convenience, we say that

$$\tilde{Q}_d = \tilde{q}_d x^d + \cdots + \tilde{q}_0$$

of $\mathbb{D}[t][x;\sigma,\delta]$ is a solution of (10) if $(\tilde{q}_0, \ldots, \tilde{q}_d)$ solves (10). With this convention, we see that L is a nonzero solution of (10) with smallest degree.

To find L, we need to solve (10) and to find a solution with smallest degree, because $\deg L$ may be smaller than d. Can we solve only one linear system in $(1 + \deg L)$ unknowns to get L ? The following lemma provides an answer.

Proposition 4.3 $\deg L$ *is equal to the rank of* M_d *given in (10).*

Proof. Let L be of degree l. Since $l \leq d$, $L, xL, \ldots, x^{d-l}L$ are solutions of (10). Thus, the solution space of (10) is of dimension no less than $(d+1-l)$.

On the other hand, any nonzero solution $\tilde{Q}_d$ of (10) is a common left multiple of $P_1, \ldots, P_m$ with degree no more than d, so the right remainder of $\tilde{Q}_d$ and L is zero, that is, $\tilde{Q}$ is a k-linear combination of $L, xL, \ldots, x^{d-l}L$. Hence, the solution space of (10) is of dimension $(d+1-l)$. Consequently, the rank of M_d is equal to d. □

We compute L as follows. First, construct matrix M_d given by (10). Second, apply a modular and an evaluation mapping to entries of M_d to obtain a matrix M'_d over $\mathbb{Z}_p$. Third, compute the rank r of M'_d. Fourth, set

$$Q_r = q_r x^r + \cdots + q_0$$

where $q_0, \ldots, q_r$ are unspecified coefficients. For $i = 1, \ldots, m$, compute the right remainder R_i of Q_r and P_i. The condition $R_1 = \cdots = R_m = 0$ gives rise to a linear homogeneous algebraic system

consists of 10 triplets of polynomials in the di erential ring. For each triplet

$$(q_0, \ldots, q_r) M_r = 0. \tag{11}$$

Any nontrivial solution of (11) corresponds to the lclm L of $P_1, \ldots, P$ $_m$ because $r \leq \deg L$ by Proposition 4.3. If (11) has only the trivial so

then update r to be $(r\ ^3 +\ \ 1)$, and repeat the fourth step. Since r is almost always equal to the rank of M_d, we hardly repeat the fourth step in practice.

We shall refer this method as the "unnested" method.

Experiment 2 This experiment is on lclm computation. The set of tests

of polynomials p_1, $_5$, and p_3 we impose the following constraints:

$$\deg p_1 = \deg p_2 = 5,\ \mathrm{gcrd}(p_1, p_2) = 2,\ \deg p = 3.$$

Table 2 shows the timing comparison between the nested and unnested methods. We also include the timing fdr the function DEtools[LCLM]. Note that if deg lclm(p_1 $_2$ $_1$ + deg p $_3$, then the timings for the unnested method and `DEtools[LCLM]` would be approximately the same.

$, p$

We conclude this section with an application of lclm computation in the direct algorithm for computing the minimal telescoper for a rational function [18]. Consider the rational function $R(n,k) = R$ $_1$ $_2$ $_3$

$$R_1 = \frac{}{(2n+5k+3)} + \frac{n}{(2n+5k+5)^2}$$

R $_2$ $_2$

$n + 4$

$9n + 4k + 4$

$$R_3 = \frac{(n\ \ 3)^1}{n - 7k + 5} + \frac{3n + 4k\ \ 3}{n - 7k + 6} \, .$$

$+R$ 1

$+R$

$, p$

Tahle 2: lclm computation: timing (in seconds) for different methods.

	Nested	Unnested	DEtools
7	96.84	18.60	61.73
1	114.58	25.79	85.48
2	147.36	24.28	107.60
4	111.95	33.36	105.33
1	124.15	33.41	+ 184.43
4	120.65	30.76	102.63
6	147.56	29.35	103.03
8	115.04	28.36	92.64
9	140.59	21.18	122.81
10	127.97	16.13	66.41

The computed minimal telescopers L $_9$, L_2 for R_2 and L_3 $_3$ are

$$_1 = \text{OrePoly}\,((n+5)(n+4)(n+3)(n+2)(2n+7),\ 5(n+5)(n+4)(n+3),$$

$$-5(n+5)(n+4)(n+5),^2 5\,(n+5)(^8n+2)(n+1),$$

$$-5\,(n+7)(n+2)(n+1), -(2n+5)(n+9)(n+3)(n+2)(n+1))\,,$$

$$800\ 10n-15,\ 0,\ -12,\ 0,\ n \quad (n+1)$$

$$_2 = \text{OrePoly}\left(-n^{2} \qquad L \qquad\right) \qquad +6 \quad 1$$

$$= \text{OrePoly}\left(n^{14} + 14\,n \quad + 63\,n^{12} + 28\,n \quad - 553\,n^{10} - 1218\,n^{9} - 7296\,n^{8} + \quad + 929\,n\right.$$

$$4984\,n^{7} + 1848\,n \quad 6496\,n^{5} - 4392\,n^{4} + 2681\,n^{3} + 2304\,n$$

$$-7 \quad (2\,n+1)\,(n\ \ 1)^{2}(n+3)(n+2) \quad (n^{2}+1)^{1}\,n^{2},\ 7\,(2\,n+1)\,(n+3)^{3}$$

$$(n+2)^{2}\,(n+1)^{2}\ \ ^{2},\ -7\,(2\,n+1)\,(n+3)\ (n+2) \qquad ^{2}$$

$$^{2}\,(n{+}2)^{2} \qquad ,\ 14\,y+7, \qquad ^{2}$$

$$-n^{14} + 28\,n^{92} - 294\,n^{10} + 1444\,n \quad 3409\,n_{11} + 3528\,n^{6} \quad , \quad ^{2} - 1\Big)\,.$$

eref re t e minimal tele c per L for the rational function ^2R is lclm(L $_2$).

If one uses the *unnested* method, it would take 6.28 seconds to compute L, as opposed to 273.90 seconds using the *nested* method.

5 The `OreTools` Package

The `OreTools` package is implemented in the Maple computer algebra sys-

, L

tem. Its main goal is to provide basic operations in a given Ore ring, and to

$$,-7(2n+1)(n+3)$$

facilitate further development of various Ore ring based applications. The package is integrated into Maple. particular, it is used (a) as the main
engine in the `LinearOperators` package which includes functions for com-
minimal Z-pairs of rational functions [18], for computing indefinite sums us-
puting minimal completely factorable annihilators [5], and for computing d'Alembertian solutions of inhomogeneous linear functional equations [3]; (b) in the `SumTools` package [2] for computing directly and efficiently the

ing the method of accurate integration; and (c) in the `Slode` package [29] for finding formal solutions with d'Alembertian series coefficients of homogeneous linear differential equations.

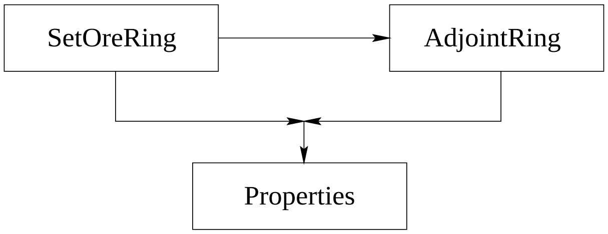
In this section we give an overview of the package. See [4] for a detailed discussion on the proposed functionalities, and the implementation details. Various Maple worksheets illustrating the use of the package are also made available (see Section 7).

Note that an early version of the package `OreTools` was reported in [5, Sect. 6]. The code of that version was designed by Zima.

5.1 Define an Ore ring, its adjoint, and access its properties

Figure 1 shows the set of functions which help define an Ore ring, the adjoint of a given Ore ring (which is an Ore ring itself), and those for accessing properties of an Ore ring.

Figure 1: Define an Ore ring, and access its properties



A univariate Ore ring is defined via the function *SetOreRing*. The *differential*, *shift*, and *qshift* rings are pre-defined. To define other rings, one needs to provide procedures to compute σ, δ, $\theta(1)$, and σ^{-1}.

The adjoint of a given Ore ring is defined via the function *AdjointRing*. The input is an Ore ring, and the output its adjoint.

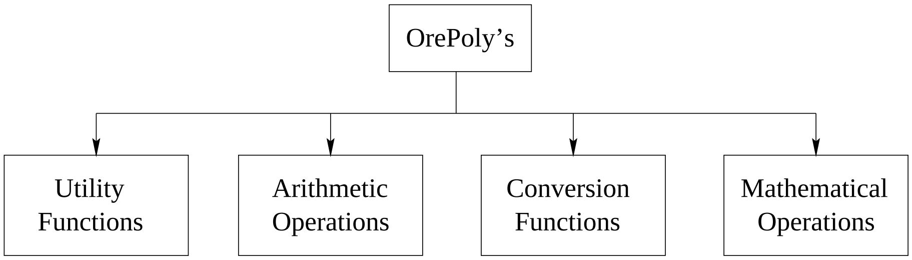
Properties of a given Ore ring, e.g., $\sigma, \sigma^{-1}, \theta(1), \delta$, can be accessed via the sub-module `Properties`.

5.2 Operations and manipulations on Ore polynomials

An Ore polynomial is represented by an OrePoly structure. It consists of the keyword OrePoly with a sequence of coefficients starting with the one of degree zero. For example, in the differential case with the differential operator D, $\mathrm{OrePoly}(2/t, t, t+1, 1)$ represents the operator $2/t + t\,D + (t+1)\,D^2 + D^3$.

Figure 2 shows basic operations and manipulations on Ore polynomials. They can be classified into four groups: utility functions, arithmetic operations, conversion functions, and mathematical operations.

Figure 2: Operations and manipulations on Ore polynomials



Utility functions include those for manipulating Ore polynomials, e.g., the leading and trailing coefficients, or the degree of a given Ore polynomial.

The basic arithmetic operations on Ore polynomials include

1. linear operations: addition, subtraction, scalar multiplication;
2. operations for normalization: computation of the content part, the primitive part, left and right monic associates.
3. multiplication, divisions (left and right remainders and quotients);
4. left and right gcd, lcm, extended gcd; and gcrd depending on a parameter [13].

Conversion functions act as an interface between the package `OreTools` and the Maple system. It includes functions for converting back and forth between a given Ore polynomial and the corresponding linear functional equation.

The package provides support for some mathematical operations. They includes functions which perform accurate integration (Section 3.3), and compute an integrating factor (Section 3.2).

The submodule `Modular` provides users with basic operations on Ore polynomials whose coefficients are rational functions over $\mathbb{Z}_p$; and the submodule `FractionFree` provides users with fraction-free operations on Ore polynomials whose coefficients are polynomials over $\mathbb{Z}$.

5.3 Examples

In the shift ring A:

```
> A := SetOreRing(n,'shift');
```

$$A := \mathrm{UnivariateOreRing}(n, \mathit{shift})$$

Consider two Ore polynomials p_1 and p_2:

```
> p_1 := OrePoly((n-3)*n^2,n^4+n^3-4*n^2-n-2,
  n^4+3*n^3+2*n^2+n-4,n^3+6*n^2+10*n+2,n^2+6*n+6):
> p_2 := OrePoly((n-3)*n^3,n^5+n^4-6*n^3+4*n^2-3*n-2,
  n^5+n^4-n^3+7*n^2-2*n-3,n^4+5*n^3+7*n^2+5*n+1,(n^2+6*n+6)*n):
```

Compute gcrd of p_1 and p_2:

```
> GCD['right'](p_1,p_2,A);
```

$$\mathrm{OrePoly}\left(\frac{n-3}{n^2-3}, 1\right)$$

Compute gcld of p_1 and p_2:

```
> GCD['left'](p_1,p_2,A);
```

$$\mathrm{OrePoly}\left(n^2, n+1, 1\right)$$

For the two Ore polynomials p_3 and p_4:

```
> p_3 := OrePoly(1,1,0,(a+2)*n):
> p_4 := OrePoly(0,(a+2)*(a+1)*n):
```

Suppose a priori that the value of the parameter a satisfies the equation $(a+1)(a-1)a = 0$. We now compute the gcrd of p_3 and p_4 depending on the parameter a:

```
> ParametricGCRD(p_3, p_4, (a+1)*(a+2)*a, a, A);
```

$$\begin{cases} \mathrm{OrePoly}\,(-1) & a = 0 \\ \mathrm{OrePoly}\,(1, 1, 0, n) & a + 1 = 0 \\ \mathrm{OrePoly}\,(1, 1) & a + 2 = 0 \end{cases}$$

6 A Comparison

There are other Maple packages which provide suitable environments for working with general Ore rings, or with a particular Ore ring. They include the `Ore_algebra` package [9] for multivariate Ore rings, the `DEtools` package for differential case, the `LREtools` package for the shift case, and the `QDifferenceEquations` package for the q-shift case. While the main focus of `LREtools` and `QDifferenceEquations` is for finding solutions of specific types (e.g., polynomial, rational) of linear shift/q-shift equations with polynomial coefficients, the `Ore_algebra` and `DEtools` packages do provide, although in a lesser extent in the comparison with the `OreTools` package, support for basic operations in Ore rings.

A comparison between `DEtools` and `OreTools` is done via two experiments in Section 4. For the remainder of this section, we show a comparison between `Ore_algebra` and `OreTools`.

The only functionality which allows a direct comparison between the two packages is the one which performs extended right Euclidean algorithm: *skew_gcdex* in `Ore_algebra`, and *ExtendedGCD* in `OreTools`. Using *skew_gcdex* is the only way to compute gcrd in `Ore_algebra` package. This involves the construction of two co-sequences which are redundant.

In this experiment, we generated two sets of tests. Each set consists of 10 pairs of polynomials p_1 and p_2. Those in the first set were generated in the *shift* ring, and those in the second set in the *differential* ring.

For each pair p_1, p_2, the following constraints are imposed:

$$7 \leq \deg p_1, \deg p_2 \leq 10; \quad \deg \mathrm{gcrd}(p_1, p_2) \geq 2;$$

Each coefficient of p_1 and p_2 is a polynomial of degree at most 5, and consists of at most 2 monomials.

The figures in Tables 3 and 4 provide a comparison in both time (in seconds) and memory (in kilobytes) requirements between *ExtendedGCD* and *skew_gcdex*.

It is worth noting that all Ore rings declared in the `Ore_algebra` package are a priori with integer coefficients, and any other type of coefficients has

Table 3: OreTools and Ore_algebra: *shift* case.

shift	ExtendedGCD		skew_gcdex	
	time	memory	time	memory
1	123	703,329	1,691	4,669,006
2	55	276,828	487	1,555,668
3	183	830,378	1,269	3,420,360
4	44	230,488	648	1,977,186
5	145	654,363	364	1,219,685
6	113	511,026	268	979,230
7	47	236,447	470	1,549,453
8	179	780,795	656	1,984,256
9	49	241,977	128	490,365
10	89	417,157	177	635,439

to be explicitly declared. Hence, performing basic operations might require a non-trivial effort and knowledge from users.

7 Availability

Information on the availability of the library archive for the package `OreTools`, sampled Maple worksheets, and also on installation of the package can be found at the URL

`http://www.scg.math.uwaterloo.ca/~hqle/code/OreTools/OreTools.html`

References

[1] S. A. Abramov. Rational solutions of linear difference and differential equations with polynomial coefficients, *USSR Comput. Maths. Math. Phys.* **29**, 7 – 12. Transl. from *Zl. Vychislit. matem. mat. fiz.* **29** (1989) 1611 – 1620.

[2] S.A. Abramov, J.C. Carette, K.O. Geddes, H.Q. Le. *Symbolic Summation in Maple.* Technical Report CS-2002-32, School of Computer Science, University of Waterloo, Ontario, Canada, 2002.

Table 4: OreTools and Ore_algebra: *differential* case.

differential	ExtendedGCD		skew_gcdex	
	time	memory	time	memory
1	24	245,039	765	2,828,435
2	20	169,934	189	976,940
3	38	340,290	437	1,968,124
4	20	167,486	300	1,324,531
5	11	81,216	151	861,778
6	23	206,490	53	360,019
7	17	159,388	216	1,030,755
8	23	201,707	333	1,370,342
9	13	113,148	47	319,017
10	13	117,665	61	418,924

[3] S.A. Abramov, M. van Hoeij. Integration of solutions of linear functional equations. *Integral Transformations and Special Functions* **8** (1999) no. 1-2, 3–12.

[4] S.A. Abramov, H.Q. Le, Ziming Li. `OreTools`: a computer algebra library for univariate Ore polynomial rings. *Technical Report CS-2003-12*, School of Computer Science, University of Waterloo, Ontario, Canada.

[5] S.A. Abramov, E.V. Zima. Minimal completely factorable annihilators. In: W. Küchlin (ed.), *Proceedings of the 1997 International Symposium on Symbolic and Algebraic Computation*, ACM Press, 290–297.

[6] S.A. Abramov, E.V. Zima. D'Alembertian solutions of inhomogeneous linear equations (differential, difference, and some other). In: Y.N. Lakshman (ed.), *Proceedings of the 1996 International Symposium on Symbolic and Algebraic Computation*, ACM Press, 232–240.

[7] S.A. Abramov, E.V. Zima. A universal program to uncouple linear systems. In: *Proceedings of International Conference on Computational Modeling and Computing in Physics*, Dubna, Russia, Sept. 16-21, 1996 (1997) 16–26.

[8] M. Bronstein, M. Petkovšek. An introduction to pseudo–linear algebra, *Theoretical Computer Science* **157** (1996) 3–33.

[9] F. Chyzak, B. Salvy. Non-commutative elimination in Ore algebras proves multivariate identities. *Journal of Symbolic Computation* **26** (1998) no. 2, 187–227.

[10] P.M. Cohn. *Free Rings and Their Relations.* Academic Press, 1971.

[11] P.M. Cohn. *Skew Fields, Theory of General Division Rings.* Encyclopedia of Mathematics and its applications **57** (1995) Cambridge University press.

[12] M. Giesbrecht, Y. Zhang. Factoring and decomposing Ore polynomials over $\mathbb{F}_p(t)$. To appear in the *Proceedings of the 2003 International Symposium on Symbolic and Algebraic Computation.*

[13] P.E. Glotov. An algorithm of searching the greatest common divisor for Ore polynomials with polynomial coefficients depending on a parameter. *Programming and Computer Software* **24** (1998) no. 6, 275–283. Transl. from *Programmirovanie* (1998) no. 6, 14 – 21.

[14] R.W. Gosper. Decision procedure for indefinite hypergeometric summation. *Proc. Natl. Acad. Sci. USA* **75** (1978) 40–42.

[15] J. van der Hoeven. FFT-like multiplication of linear differential operators. *Journal of Symbolic Computation* **33** (2002) no. 1, 123 - 127.

[16] N. Jacobson. Pseudo-linear transformations. *Annals of Mathematics* **38** (1937) no. 2, 484–507.

[17] E. Landau. Über irreduzible Differentialgleichungen. *J. für die reine und angewandte Mathematik* **124** (1902) 115–120.

[18] H.Q. Le. A direct algorithm to construct the minimal Z-pairs for rational functions. *Advances in Applied Mathematics* **30** (2003) 137–159.

[19] Z. Li. A subresultant theory for ore polynomials with applications. In: O. Gloor (ed), *Proceedings of the 1998 International Symposium on Symbolic and Algebraic Computation*, ACM Press, 132–139.

[20] Z. Li, I. Nemes. A modular algorithm for computing greatest common right divisors of Ore polynomials. In: W. Küchlin (ed.), *Proceedings of*

the 1997 International Symposium on Symbolic and Algebraic Computation, ACM Press, 282–289.

[21] A. Loewy. Über reduzible lineare homogene Differentialgleichungen. *Math. Annalen* **56** (1903) 549–584.

[22] A. Loewy. Über vollstandig reduzible lineare homogene Differentialgleichungen. *Math. Annalen* **62** (1906) 89–117.

[23] M.B. Monagan, K.O. Geddes, K.M. Heal, G. Labahn, S.M. Vorkoetter, J. McCarron, P. DeMarco. *Maple 8 Introductory Programming Guide.* Waterloo Maple Inc., Waterloo, Ontario, Canada, 2002.

[24] O. Ore. Theory of non-commutative polynomials. *Annals of Mathematics* **34** (1933) 480–508.

[25] O. Ore. Formale Theorie der linearen Differentialgleichungen (Erster Teil). *J. für die reine und angewandte Mathematik* **167** (1932) 221–234.

[26] O. Ore. Formale Theorie der linearen Differentialgleichungen (Zweiter Teil). *J. für die reine und angewandte Mathematik* **1** (1932), 233–252.

[27] E. G. C. Poole. *Introduction to the theory of linear ordinary differential equations.* Dover Publications Inc., New York, 1936.

[28] M. Petkovšek. Hypergeometric solutions of linear recurrences with polynomial coefficients. *Journal of Symbolic Computation* **14** (1992) 243–264.

[29] A. Ryabenko. A Maple package for the symbolic construction of power series solutions to linear ordinary differential equations. *Programming and Computer Software* **25** (1999) no. 5, 296–305.

[30] J.H.M. Wedderburn. Non-commutative domains of integrity. *J. Reine Angew. Math.* **167** (1932) 129–141.