Computer Science Journal of Moldova, vol.**20**, no.2(59), 2012

Computation of Difference Gröbner Bases

Vladimir P. Gerdt, Daniel Robertz

Abstract

This paper is an updated and extended version of our note [1] (cf. also [2]). To compute difference Gröbner bases of ideals generated by linear polynomials we adopt to difference polynomial rings the involutive algorithm based on Janet-like division. The algorithm has been implemented in Maple in the form of the package LDA (Linear Difference Algebra) and we describe the main features of the package. Its applications are illustrated by generation of finite difference approximations to linear partial differential equations and by reduction of Feynman integrals. We also present the algorithm for an ideal generated by a finite set of nonlinear difference polynomials. If the algorithm terminates, then it constructs a Gröbner basis of the ideal.

1 Introduction

Being invented 47 years ago by Buchberger [3] for algorithmic solving of the membership problem in the theory of polynomial ideals, the Gröbner basis method has become a powerful universal algorithmic tool for solving various mathematical problems arising in science and engineering.

Though the overwhelming majority of Gröbner basis applications is still found in commutative polynomial algebra, over the last two decades a substantial progress has also been achieved in applications of Gröbner bases to noncommutative polynomial algebra, to algebra of differential operators and to linear partial differential equations (cf., for example, the book [4]). As to the difference algebra, i.e. algebra of difference polynomials [5], in spite of its conceptual algorithmic similarity to differential algebra, only a few efforts have been made to extend the theory of Gröbner bases to difference algebra and to exploit their algorithmic power [5, 6, 7].

Recently, three promising applications of difference Gröbner bases were revealed:

- Generation of finite difference approximations to PDEs [8, 9].
- Consistency analysis of such approximations [10, 11].
- Reduction of multiloop Feynman integrals to the minimal set of basis integrals [12].

In this note we describe an algorithm (Section 4) for constructing Gröbner bases for linear difference systems that is an adaptation of the polynomial algorithm [13] to linear difference ideals. In so doing, we construct a Gröbner basis in its Janet-like form (Section 3), since this approach has shown its computational efficiency in the polynomial case [13, 14]. We briefly outline these efficiency issues in Section 5. The difference form of the algorithm exploits some basic notions and concepts of difference algebra (Section 2) as well as the definition of Janet-like Gröbner bases and Janet-like reductions together with the algorithmic characterization of Janet-like bases (Section 3). Extension of the notion of Gröbner basis to nonlinear difference polynomials, which has not been addressed in [1], [2], is briefly described in Section 6 where we also present the algorithm [11] for construction of such bases. In Section 7 we present our Maple package LDA for computing Gröbner bases of linear difference ideals, i.e. ideals generated by linear difference polynomials. The package is a modified version of our earlier package [17] oriented towards commutative and linear differential algebra and based on the involutive basis algorithm [14]. The modified version is specialized to linear difference ideals and uses both Janet and Janet-like divisions [13] adopted to linear difference polynomials [15]. In Sections 8 and 9 we illustrate LDA by simple examples of its application to the construction of finite difference approximations to linear systems of PDEs and to the reduction of Feynman integrals.

2 Elements of difference algebra

Let $\{y^1, \ldots, y^m\}$ be the set of *indeterminates*, e.g., m functions of n variables $x_1, \ldots, x_n$, and $\{\theta_1, \ldots, \theta_n\}$ be the set of mutually commuting *difference operators (differences)*, i.e.,

$$(\theta_i \circ y^j)(x_1, \ldots, x_n) = y^j(x_1, \ldots, x_i + 1, \ldots, x_n).$$

A *difference ring R with differences* $\theta_1, \ldots, \theta_n$ is a commutative ring R such that for all $f, g \in R,\ 1 \leq i, j \leq n$,

$$\begin{aligned} &\theta_i\theta_j = \theta_j\theta_i, \quad \theta_i \circ (f + g) = \theta_i \circ f + \theta_i \circ g, \\ &\theta_i \circ (f\,g) = (\theta_i \circ f)(\theta_i \circ g)\,. \end{aligned}$$

Similarly, one defines a *difference field.*

Let $\mathcal{K}$ be a difference field, and $\mathcal{R} := \mathcal{K}\{y^1, \ldots, y^m\}$ be the difference ring of polynomials over $\mathcal{K}$ in variables

$$\{\ \theta^\mu \circ y^k \ \mid \mu \in \mathbb{Z}^n_{\geq 0},\ k = 1, \ldots, m\ \}\,.$$

Hereafter, we denote by $\mathcal{R}_L$ the set of linear polynomials in $\mathcal{R}$ and use the notations:

$$\begin{aligned} &\Theta = \{\ \theta^\mu \ \mid\ \mu \in \mathbb{Z}^n_{\geq 0}\ \}, \quad \deg_i(\theta^\mu \circ y^k) = \mu_i, \\ &\deg(\theta^\mu \circ y^k) = |\mu| = \textstyle\sum_{i=1}^n \mu_i\,. \end{aligned}$$

A *difference ideal* is an ideal $\mathcal{I} \subseteq \mathcal{R}$ closed under the action of any operator from Θ. For $F \subset \mathcal{R}$, the smallest difference ideal containing F will be denoted by $\mathrm{Id}(F)$. If for an ideal $\mathcal{I}$ there is $F \subset \mathcal{R}_L$ such that $\mathcal{I} = \mathrm{Id}(F)$, then I is a *linear difference ideal.*

A total ordering $\succ$ on the set of $\theta^\mu \circ y^j$ is a *ranking* if for all i, j, k, μ, ν the following hold:

$$\begin{aligned} &\theta_i\theta^\mu \circ y^j \succ \theta^\mu \circ y^j\,, \\ &\theta^\mu \circ y^j \succ \theta^\nu \circ y^k \iff \theta_i\theta^\mu \circ y^j \succ \theta_i\theta^\nu \circ y^k\,. \end{aligned}$$

If $|\mu| > |\nu|$ implies $\theta^\mu \circ y^j \succ \theta^\nu \circ y^k$ for all j, k, then the ranking is *orderly.* If $j > k$ implies $\theta^\mu \circ y^j \succ \theta^\nu \circ y^k$ for all μ, ν, then the ranking is *elimination.*

Given a ranking $\succ$, a linear polynomial $f \in \mathcal{R}_L \setminus \{0\}$ has the *leading term* $a\,\vartheta \circ y^j$, $\vartheta \in \Theta$, $a \in \mathcal{K}$, where $\vartheta \circ y^j$ is maximal w.r.t. $\succ$ among all $\theta^\mu \circ y^k$ which appear with nonzero coefficient in f. $\mathrm{lc}(f) := a \in \mathcal{K} \setminus \{0\}$ is the *leading coefficient* and $\mathrm{lm}(f) := \vartheta \circ y^j$ is the *leading monomial*.

A ranking acts in $\mathcal{R}_L$ as a *monomial order*. If $F \subseteq \mathcal{R}_L \setminus \{0\}$, $\mathrm{lm}(F)$ will denote the set of the leading monomials and $\mathrm{lm}_j(F)$ will denote its subset for the indeterminate y^j. Thus,

$$\mathrm{lm}(F) = \bigcup_{j=1}^{m} \mathrm{lm}_j(F)\,.$$

3 Janet-like Gröbner bases

Given a nonzero linear difference ideal $\mathcal{I} = \mathrm{Id}(G)$ and a ranking $\succ$, the ideal generating set $G = \{g_1, \ldots, g_s\} \subset \mathcal{R}_L$ is a *Gröbner basis* [4, 7] of $\mathcal{I}$ if for all $f \in \mathcal{I} \cap \mathcal{R}_L \setminus \{0\}$:

$$\exists\, g \in G,\ \theta \in \Theta \ :\ \mathrm{lm}(f) = \theta \circ \mathrm{lm}(g)\,. \tag{1}$$

It follows that $f \in \mathcal{I} \setminus \{0\}$ *is reducible modulo* G:

$$f \xrightarrow[g]{} f' := f - \mathrm{lc}(f)\,\theta \circ (g/\,\mathrm{lc}(g)), \quad f' \in \mathcal{I}\,.$$

If $f' \neq 0$, then it is again reducible modulo G, and, by repeating the reduction, in finitely many steps we obtain

$$f \xrightarrow[G]{} 0\,.$$

Similarly, a nonzero polynomial $h \in \mathcal{R}_L$, whose terms are reducible (if any) modulo a set $F \subset \mathcal{R}_L$, can be reduced to an irreducible polynomial $\bar{h}$, which is said to be in *normal form modulo* F (denotation: $\bar{h} = NF(h, F)$).

In our algorithmic construction of Gröbner bases we shall use a restricted set of reductions called *Janet-like* (cf. [13]) and defined as follows.

For a finite set $F \subseteq \mathcal{R}_L \setminus \{0\}$ and a ranking $\succ$, we partition every $\mathrm{lm}_k(F)$ into subsets labeled by $d_0, \ldots, d_i \in \mathbb{Z}_{\geq 0}$, $(0 \leq i \leq n)$. Here $[0]_k := \mathrm{lm}_k(F)$ and for $i > 0$ the subset $[d_0, \ldots, d_i]_k$ is defined as

$$\{u \in \mathrm{lm}_k(F) \mid d_0 = 0, d_j = \deg_j(u), 1 \leq j \leq i\}.$$

Denote by $h_i(u, \mathrm{lm}_k(F))$ the nonnegative integer

$$\max\{\deg_i(v) \mid u, v \in [d_0, \ldots, d_{i-1}]_k\} - \deg_i(u).$$

If $h_i(u, \mathrm{lm}_k(F)) > 0$, then $\theta_i^{s_i}$ such that

$$s_i := \min\{\deg_i(v) - \deg_i(u) \mid u, v \in [d_0, \ldots, d_{i-1}]_k,\ \deg_i(v) > \deg_i(u)\}$$

is called a *difference power* for $f \in F$ with $\mathrm{lm}(f) = u$.
Let $DP(f, F)$ be the set of difference powers for $f \in F$, and $\mathcal{J}(f, F) := \Theta \setminus \bar{\Theta}$ be the subset of Θ with

$$\bar{\Theta} := \{\theta^\mu \mid \exists\, \theta^\nu \in DP(f, F) \ :\ \mu - \nu \in \mathbb{Z}^n_{\geq 0}\}.$$

A Gröbner basis G of $I = \mathrm{Id}(G)$ is called *Janet-like* [13] if for all $f \in I \cap \mathcal{R}_L \setminus \{0\}$:

$$\exists\, g \in G, \vartheta \in \mathcal{J}(g, G) \ :\ \mathrm{lm}(f) = \vartheta \circ \mathrm{lm}(g)\,. \tag{2}$$

This implies $\mathcal{J}-$reductions and the $\mathcal{J}-$normal form $NF_{\mathcal{J}}(f, F)$. It is clear that condition (2) implies condition (1). Note, however, that the converse is generally not true. Therefore, not every Gröbner basis is Janet-like.

The properties of a Janet-like basis are very similar to those of a Janet basis [14], but the former is generally more compact than the latter. More precisely, let GB be a reduced Gröbner basis [4], JB be a minimal Janet basis, and JLB be a minimal Janet-like basis of the same ideal for the same ranking. Then we have

$$\mathrm{Card}(GB) \leq \mathrm{Card}(JLB) \leq \mathrm{Card}(JB), \tag{3}$$

where Card abbreviates *cardinality*, that is, the number of elements.

Whereas the algorithmic characterization of a Gröbner basis is zero redundancy of all its S-polynomials [3, 4], the algorithmic characterization of a Janet-like basis G is the following condition (cf. [13]):

$$\forall g \in G,\ \vartheta \in DP(g, G) :\ NF_{\mathcal{J}}(\vartheta \circ g, G) = 0\,. \tag{4}$$

This condition is at the root of the algorithmic construction of Janet-like bases as described in the next section.

4 Algorithm for Linear Difference Polynomials

The following algorithm is an adaptation of the polynomial version [13] to linear difference ideals.

Algorithm: *Janet-like Gröbner Basis*($F, \succ$)

Input: $F \subseteq \mathcal{R}_L \setminus \{0\}$, a finite set; $\succ$, a ranking
Output: G, a Janet-like basis of $\mathrm{Id}(F)$
choose $f \in F$ with the lowest $\mathrm{lm}(f)$ w.r.t. $\succ$
$G := \{f\}$
$Q := F \setminus G$
while $Q \neq \emptyset$ **do**
    $h := 0$
    while $Q \neq \emptyset$ **and** $h = 0$ **do**
        choose $p \in Q$ with the lowest $\mathrm{lm}(p)$ w.r.t. $\succ$
        $Q := Q \setminus \{p\}$
        $h :=$ **Normal Form**$(p, G, \succ)$
    od
    if $h \neq 0$ **then**
        for all $g \in G$ such that $\mathrm{lm}(g) = \theta^{\mu} \circ \mathrm{lm}(h),\ |\mu| > 0$ **do**
            $Q := Q \cup \{g\};\ \ G := G \setminus \{g\}$
        od
        $G := G \cup \{h\}$
        $Q := Q \cup \{\, \theta^{\beta} \circ g \mid g \in G,\ \theta^{\beta} \in DP(g, G)\, \}$
    fi
od
return G

It outputs a minimal Janet-like Gröbner basis which (if monic, that is, normalized by division of each polynomial by its leading coefficient) is uniquely defined by the input set F and ranking $\succ$. Correctness and termination of the algorithm follow from the proof given in [13]; in so doing the displacement of some elements of the intermediate sets G into Q at step 13 provides minimality of the output basis. The algorithm terminates when the set Q becomes empty in accordance with (4).

The subalgorithm ***Normal Form***$(p, G, \succ)$ performs the Janet-like reductions (Section 3) of the input difference polynomial p modulo the set G and outputs the Janet-like normal form of p. As long as the intermediate difference polynomial h has a term Janet-like reducible modulo G, the elementary reduction of this term is done at step 4. As usually in the Gröbner bases techniques [4], the reduction terminates after finitely many steps due to the properties of the ranking (Section 2).

An improved version of the above algorithm can easily be derived from the one for the involutive algorithm [14] if one replaces the input involutive division by a Janet-like monomial division [13] and then translates the algorithm into linear difference algebra. In particular, the improved version includes Buchberger's criteria adjusted to Janet-like division and avoids the repeated prolongations $\theta^\beta \circ g$ at step 16 of the algorithm.

Algorithm: *Normal Form*$(p, G, \succ)$

Input: $p \in \mathcal{R}_L \setminus \{0\}$, a polynomial; $G \subset \mathcal{R}_L \setminus \{0\}$, a finite set; $\succ$, a ranking
Output: $h = NF_{\mathcal{J}}(p, G)$, the $\mathcal{J}-$normal form of p modulo G
1: $h := p$
2: **while** $h \neq 0$ **and** h has a monomial u with nonzero coefficient $b \in \mathcal{K}$ such that u is $\mathcal{J}-$reducible modulo G **do**
3:    **take** $g \in G$ such that $u = \theta^\gamma \circ \mathrm{lm}(g)$ with $\theta^\gamma \in \mathcal{J}(g, G)$
4:    $h := h/b - \theta^\gamma \circ (g/\,\mathrm{lc}(g))$
5: **od**
6: **return** h

5 Computational aspects

The polynomial version of algorithm ***Janet-like Gröbner Basis*** is implemented in its improved form in C++ [13] as a part of the specialized computer algebra system GINV [16]. It has disclosed its high computational efficiency for the standard set of benchmarks[1]. If one compares this algorithm with the involutive one [14] specialized to Janet division, then all the computational merits of the latter algorithm are retained, namely:

- Automatic avoidance of some useless reductions.
- Weakened role of the criteria: even without applying any criteria the algorithm is reasonably fast. By contrast, Buchberger's algorithm without applying the criteria becomes unpractical even for rather small problems.
- Smooth growth of intermediate coefficients.
- Fast search of a polynomial reductor which provides an elementary Janet-like reduction of the given term. It should be noted that as well as in the involutive algorithm such a reductor, if it exists, is unique. The fast search is based on the special data structures called Janet trees [14].
- Natural and effective parallelism.

Though one needs intensive benchmarking for linear difference systems, we have solid grounds to believe that the above listed computational merits hold also for the difference case.

As this takes place, computation of a Janet-like basis is more efficient than computation of a Janet basis by the involutive algorithm [14]. The inequality (3) for monic bases is a consequence of the inclusion [13]:

$$GB \subseteq JLB \subseteq JB\,. \tag{5}$$

[1] Cf. the web page `http://invo.jinr.ru`.

There are many systems for which the cardinality of a Janet-like basis is much closer to that of the reduced Gröbner basis than the cardinality of a Janet basis. Certain binomial ideals called toric form an important class of such problems. Toric ideals arise in a number of problems of algebraic geometry and closely related to integer programming. For this class of ideals the cardinality of Janet bases is typically much larger than that of reduced Gröbner bases [13]. For illustrative purposes consider a difference analogue of the simple toric ideal [13, 18] generated in the ring of difference operators by the following set:

$$\{ \ \theta_x^7 - \theta_y^2\theta_z, \theta_x^4\theta_w - \theta_y^3, \theta_x^3\theta_y - \theta_z\theta_w \ \} .$$

The reduced Gröbner basis for the degree-reverse-lexicographic ranking with $\theta_x \succ \theta_y \succ \theta_z \succ \theta_w$ is given by

$$\{ \ \theta_x^7 - \theta_y^2\theta_z, \theta_x^4\theta_w - \theta_y^3, \theta_x^3\theta_y - \theta_z\theta_w, \theta_y^4 - \theta_x\theta_z\theta_w^2 \ \} .$$

The Janet-like basis computed by the above algorithm contains one more element $\theta_x^4\theta_w - \theta_y^3$ whereas the Janet basis adds another six elements to the Janet-like basis [13].

The presence of extra elements in a Janet basis in comparison with a Janet-like basis is obtained because of certain additional algebraic operations. That is why the computation of a Janet-like basis is more efficient than the computation of a Janet basis. Both bases, however, contain the reduced Gröbner basis as the internally fixed [14] subset of the output basis[2]. Hence, having any of the bases computed, the reduced Gröbner basis is easily extracted without any extra computational costs.

6 Nonlinear Difference Polynomials

In this section we follow the paper [11] and define *difference standard bases* which generalize the concept of Gröbner bases to arbitrary ideals in the ring $\mathcal{R} = \mathcal{K}\{y^1, \ldots, y^m\}$ of difference polynomials.

[2]In the improved versions of the algorithms.

A total ordering $\succ$ on the set $\mathcal{M}$ of *difference monomials*

$$\mathcal{M} := \{\, (\theta_1 \circ y^1)^{i_1} \cdots (\theta_m \circ y^m)^{i_m} \mid \theta_j \in \Theta,\ i_j \in \mathbb{Z}_{\geq 0},\ 1 \leq j \leq m \,\}$$

is *admissible* if it extends a ranking and satisfies

$$(\forall\, t \in \mathcal{M} \setminus \{1\})\ [t \succ 1]\ \wedge\ (\forall\, \theta \in \Theta)\ (\forall\, t, v, w \in \mathcal{M}\,)$$
$$[\, v \succ w \Longleftrightarrow t \cdot \theta \circ v \succ t \cdot \theta \circ w\,].$$

As an example of admissible monomial ordering we indicate the *lexicographical monomial ordering* compatible with a ranking.

Given an admissible ordering $\succ$, every nonzero difference polynomial f has the *leading monomial* $\mathrm{lm}(f) \in \mathcal{M}$ with the *leading coefficient* $\mathrm{lc}(f)$. In what follows, every nonzero difference polynomial is to be *normalized (i.e., monic)* by division of the polynomial by its leading coefficient.

If for $v, w \in \mathcal{M}$ the equality $w = t \cdot \theta \circ v$ holds with $\theta \in \Theta$ and $t \in \mathcal{M}$ we shall say that v *divides* w and write $v \mid w$. It is easy to see that this divisibility relation yields a *partial order*.

Given a difference ideal $\mathcal{I}$ and an admissible monomial ordering $\succ$, a subset $G \subset \mathcal{I}$ is its *(difference) standard basis* if $\mathrm{Id}(G) = \mathcal{I}$ and

$$(\,\forall\, f \in \mathcal{I}\,)(\,\exists\, g \in G\,)\ \ [\,\mathrm{lm}(g) \mid \mathrm{lm}(f)\,]\,. \tag{6}$$

As in differential algebra [19], if a standard basis is finite it is called Gröbner basis.

A polynomial $p \in \mathcal{R} \setminus \{0\}$ is said to be *head reducible modulo* $q \in \mathcal{R} \setminus \{0\}$ *to* r if $r = p - m \cdot \theta \circ q$ and $m \in \mathcal{M}$, $\theta \in \Theta$ are such that $\mathrm{lm}(p) = m \cdot \theta \circ \mathrm{lm}(q)$. In this case the transformation from p to r is an *elementary reduction* and denoted by $p \underset{q}{\rightarrow} r$. Given a set $F \subset \mathcal{R} \setminus \{0\}$, p *is head reducible modulo* F (denotation: $p \underset{F}{\longrightarrow}$) if there is $f \in F$ such that p is head reducible modulo f. A polynomial p *is head reducible to* r *modulo* F if there is a chain of elementary reductions

$$p \underset{F}{\longrightarrow} p_1 \underset{F}{\longrightarrow} p_2 \underset{F}{\longrightarrow} \cdots \underset{F}{\longrightarrow} r\,. \tag{7}$$

Similarly, one can define *tail reduction.* If r in (7) and each of its monomials is neither head nor tail reducible modulo F, then we shall say that r *is in normal form modulo* F and write $r = \mathrm{NF}(p, F)$. A polynomial set F with more then one element is *interreduced* if

$$(\,\forall f \in F\,)\ [\, f = \mathrm{NF}(f, F \setminus \{f\})\,]\,. \tag{8}$$

Admissibility of $\succ$, as in commutative algebra, provides termination of chain (7) for any p and F. In doing so, $\mathrm{NF}(p, F)$ can be computed by the difference version of a multivariate polynomial division algorithm [20, 21]. If G is a standard basis of $\mathrm{Id}(G)$, then from the above definitions it follows

$$f \in \mathrm{Id}(G) \Longleftrightarrow \mathrm{NF}(f, G) = 0\,.$$

Thus, if an ideal has a finite standard (Gröbner) basis, then its construction solves the ideal membership problem as well as in commutative [20, 21] and differential [19, 22] algebra. The algorithmic characterization of standard bases, and their construction in difference polynomial rings is done in terms of difference S-polynomials.

Given an admissible ordering, and monic difference polynomials p and q, the polynomial

$$S(p, q) := m_1 \cdot \theta_1 \circ p - m_2 \cdot \theta_2 \circ q$$

is called *S-polynomial* associated to p and q (for $p = q$ we shall say that the *S-polynomial* is associated with p) if

$$m_1 \cdot \theta_1 \circ \mathrm{lm}(p) = m_2 \cdot \theta_2 \circ \mathrm{lm}(q)$$

with coprime $m_1 \cdot \theta_1$ and $m_2 \cdot \theta_2$.

Algorihmic characterization of standard bases: Given a difference ideal $\mathcal{I} \subset \mathcal{R}$ and an admissible ordering $\succ$, a set of polynomials $G \subset \mathcal{I}$ is a standard basis of $\mathcal{I}$ if and only if $\mathrm{NF}(S(p, q), G) = 0$ for all S-polynomials, associated with polynomials in G. This result follows from the above definitions in line with the standard proof of the analogous characterization of Gröbner bases in commutative algebra [20, 21] and with the proof of similar characterization of standard bases in differential algebra [19].

Let $\mathcal{I} = \mathrm{Id}(F)$ be a difference ideal generated by a finite set $F \subset \mathcal{R}$ of difference polynomials. Then for a fixed admissible monomial ordering the following algorithm ***StandardBasis***, if it terminates, returns a standard basis G of $\mathcal{I}$. The subalgorithm ***Interreduce*** invoked in step 11 performs mutual interreduction of the elements in $\tilde{H}$ and returns a set satisfying (8).

Algorithm ***StandardBasis*** is a difference analogue of the simplest version of Buchberger's algorithm (cf. [19, 20, 21]). Its correctness is provided by the above formulated algorithmic characterization of standard bases. The algorithm always terminates when the input polynomials are linear. If this is not the case, the algorithm may not terminate. This means that the **do while**-loop (steps 2–10) may be infinite as in the differential case [19, 22]. One can improve the algorithm by taking into account Buchberger's criteria to avoid some useless zero reductions in step 5. The difference criteria are similar to the differential ones [19].

Algorithm: *StandardBasis* $(F, \succ)$

Input: $F \subset \mathcal{R} \setminus \{0\}$, a finite set of nonzero polynomials;
  $\succ$, an admissible monomial ordering
Output: G, an interreduced standard basis of $\mathrm{Id}(F)$
1: $G := F$
2: **do**
3:   $\tilde{H} := G$
4:   **for all** S-polynomials s associated with elements in $\tilde{H}$ **do**
5:     $g := \mathrm{NF}(s, \tilde{H})$
6:     **if** $g \neq 0$ **then**
7:       $G := G \cup \{g\}$
8:     **fi**
9:   **od**
10: **od while** $G \neq \tilde{H}$
11: $G :=$***Interreduce***(G)
12: **return** G

7 The Maple Package LDA

The package LDA (abbreviation for **L**inear **D**ifference **A**lgebra)[3] implements the involutive basis algorithm [14] for linear systems of difference equations using Janet division. In addition, the package implements a modification of the algorithm oriented towards Janet-like division [13] and, thus, computes Janet-like Gröbner bases of linear difference ideals.

Table 1 collects the most important commands of LDA. Its main procedure `JanetBasis` converts a given set of difference polynomials into its Janet basis or Janet-like Gröbner basis form. More precisely, let $\mathcal{R}$ be the difference ring (cf. Section 2) of polynomials in the variables $\theta^\mu \circ y^k$, $\mu \in \mathbb{Z}^n_{\geq 0}$, $k = 1, \ldots, m$, with coefficients in a difference field $\mathcal{K}$ containing $\mathbb{Q}$ for which the field operations can be carried out constructively in Maple. We denote again by $\mathcal{R}_L$ the set of linear polynomials in $\mathcal{R}$. Given a finite generating set $F \subset \mathcal{R}_L$ for a linear difference ideal $\mathcal{I}$ in $\mathcal{R}$, `JanetBasis` computes the minimal Janet(-like Gröbner) basis J of $\mathcal{I}$ w.r.t. a certain monomial order (ranking). The input for `JanetBasis` consists of the left hand sides of a linear system of difference equations in the dependent variables $y^1, \ldots, y^m$, e.g., functions of $x_1, \ldots, x_n$. The difference ring $\mathcal{R}$ is specified by the lists of independent variables $x_1, \ldots, x_n$ and dependent variables given to `JanetBasis`. The output is a list containing the Janet(-like Gröbner) basis J and the lists of independent and dependent variables.

After J is computed, the involutive/$\mathcal{J}-$normal form of any element of $\mathcal{R}_L$ modulo J can be computed using `InvReduce`. Given $p \in \mathcal{R}_L$ representing a residue class $\overline{p}$ of the difference residue class ring $\mathcal{R}/\mathcal{I}$, `InvReduce` returns the unique representative $q \in \mathcal{R}_L$ of $\overline{p}$ which is not involutively/$\mathcal{J}-$reducible modulo J. A $\mathcal{K}$-basis of the vector space $\mathcal{R}_L/(I \cap \mathcal{R}_L)$ is returned by `ResidueClassBasis` as a list if it is finite or is enumerated by a formal power series [25] in case it is infinite. For examples of how to apply these two commands, cf. Section 9.

Given an affine (i.e. inhomogeneous) linear system of difference equations, a call of `CompCond` after the application of `JanetBasis` re-

[3]The package LDA is downloadable from the web page `http://wwwb.math.rwth-aachen.de/Janet`

turns a generating set of compatibility conditions for the affine part of the system, i.e. necessary conditions for the right hand sides of the inhomogeneous system for solvability.

Moreover, combinatorial devices to compute the Hilbert series and polynomial and function etc. [17] are included in LDA.

For the application of LDA to the reduction of Feynman integrals, a couple of special commands were implemented to impose further relations on the master integrals: By means of `AddRelation` an infinite sequence of master integrals parametrized by indeterminates which are not contained in the list of independent variables is set to zero. Subsequent calls of `InvReduce` and `ResidueClassBasis` take these additional relations into account (cf. Section 9).

LDA provides several tools for dealing with difference operators. Difference operators represented by polynomials can be applied to (lists of) expressions containing $y^1, \ldots, y^m$ as functions of the independent variables. Conversely, the difference operators can be extracted from systems of difference equations. Leading terms of difference equations can be selected.

We consider difference rings containing shift operators which act in one direction only. If a linear system of difference equations is given containing functions shifted in both directions, then the system needs to be shifted by the maximal negative shift in order to obtain a difference system with shifts in one direction only. However, LDA allows to change the shift direction globally.

Unnecessary computations of involutive reductions to zero are avoided using the four involutive criteria described in [14, 23, 24]. Fine-tuning is possible by selecting the criteria individually.

The implemented monomial orders/rankings are the (block) degree-reverse-lexicographical and the lexicographical one. In the case of more than one dependent variable, priority of comparison can be either given to the difference operators ("term over position") or to the dependent variables ("position over term"/elimination ranking).

The ranking is controlled via options given to each command separately. The other options described above can be set for the entire LDA session using the command `LDAOptions` which also allows to select

Table 1. Main commands of LDA

`JanetBasis`	Compute Janet(-like Gröbner) basis
`InvReduce`	Involutive / $\mathcal{J}$−reduction modulo Janet(-like Gröbner) basis
`CompCond`	Return compatibility conditions for inhomogeneous system
`HilbertSeries` etc.	Combinatorial devices
`Pol2Shift`, `Shift2Pol`	Conversion between shift operators and equations
Some interpretations of commands for the reduction of Feynman integrals:	
`ResidueClassBasis`	Enumeration of the master integrals
`AddRelation`	Definition of additional relations for master integrals
`ResidueClassRelations`	Return the relations defined for master integrals

Janet or Janet-like division.

8 Generation of finite difference schemes for PDEs

We consider the Laplace equation $u_{xx} + u_{yy} = 0$ and rewrite it as the conservation law

$$\oint_{\Gamma} -u_y dx + u_x dy = 0.$$

Adding the integral relations

$$\begin{aligned} \int_{x_j}^{x_{j+2}} u_x dx &= u(x_{j+2}, y) - u(x_j, y), \\ \int_{y_k}^{y_{k+2}} u_y dx &= u(x, y_{k+2}) - u(x, y_k) \end{aligned}$$

and using the midpoint integration method we obtain the following discrete system:

$$\left\{ \begin{array}{l} -(\theta_x - \theta_x\theta_y^2)\circ u_y + (\theta_x^2\theta_y - \theta_y)\circ u_x = 0, \\ 2\triangle h\,\theta_x\circ u_x - (\theta_x^2 - 1)\circ u = 0, \\ 2\triangle h\,\theta_y\circ u_y - (\theta_y^2 - 1)\circ u = 0, \end{array} \right. \tag{9}$$

where θ_x and θ_y represent the right-shift operators w.r.t. x and y, e.g., $(\theta_x \circ u_y)(x,y) = u_y(x+1,y)$.

We show how to use LDA to find a finite difference scheme for the Laplace equation:

```
>  with(LDA):
```

We enter the independent and the dependent variables for the problem ($ux > uy > u$):

```
>  ivar:=[x,y]:  dvar:=[ux,uy,u]:
```

Next, we translate (9) into the input format of `JanetBasis`. Note that one can in general use `AppShiftOp` to apply a difference operator given as a polynomial similar to the ones in (9) to a difference polynomial.

```
>  L:=[2*h*ux(x+1,y)-u(x+2,y)+u(x,y),
>  2*h*uy(x,y+1)-u(x,y+2)+u(x,y),
>  2*h*(ux(x+2,y+1)-ux(x,y+1))+2*h*(uy(x+1,y+2)-
>  uy(x+1,y))]:
```

Then we compute the minimal Janet basis of the linear difference ideal generated by L w.r.t. a ranking which compares the dependent variables prior to the corresponding difference monomials ("position over term" order; this ranking is chosen when using the option 2 as below). The least element of this Janet basis is by construction a difference polynomial which does not contain any monomial in ux and uy because $ux > uy > u$.

```
>  JanetBasis(L,ivar,dvar,2)[1][1];
```

$$\mathrm{u}(x+4,\, y+2) - 4\,\mathrm{u}(x+2,\, y+2) + \mathrm{u}(x,\, y+2) + \\ \mathrm{u}(x+2,\, y+4) + \mathrm{u}(x+2,\, y)$$

The computation takes less than one second of time on a Pentium III (1 GHz).

Dividing this difference polynomial by $4h^2$ we obtain the following finite difference scheme:

$$D_j^2(u_{jk}) + D_k^2(u_{jk}) = 0,$$

where

$$D_j^2(u_{jk}) = \frac{u_{j+2\,k} - 2u_{j\,k} + u_{j-2\,k}}{4h^2}$$

and

$$D_k^2(u_{jk}) = \frac{u_{j\,k+2} - 2u_{j\,k} + u_{j\,k-2}}{4h^2}$$

are discrete approximations of the second order partial derivatives occurring in Laplace's equation.

9 Reduction of Feynman integrals

In order to demonstrate how to use LDA for the reduction of Feynman integrals, we consider a simple one-loop propagator type scalar integral with one massive and another massless particle:

$$f(k,n) := I_{k,n} = \frac{1}{i\pi^{d/2}} \int \frac{d^d s}{P^k_{s-q,m} P^n_{s,0}}.$$

(Here k, n are the exponents of the propagators.)

The basis integrals for this example and the corresponding reduction formulae were found and studied by several authors (see, e.g., [26, 27]). Here we apply the Gröbner basis method, as implemented in LDA, directly to the recurrence relations which have the form:

$$\left\{\begin{array}{l} [d - 2k - n - 2m^2k\mathbf{1}^+ - \\ n\mathbf{2}^+(\mathbf{1}^- - q^2 + m^2)]\, f(k+1, n+1) = 0, \\ [n - k - k\mathbf{1}^+(q^2 + m^2 - \mathbf{2}^-) - \\ n\mathbf{2}^+(\mathbf{1}^- - q^2 + m^2)]\, f(k+1, n+1) = 0, \end{array}\right. \tag{10}$$

where

$$\mathbf{1}^\pm f(k,n) = f(k \pm 1, n),\ \ \mathbf{2}^\pm f(k,n) = f(k, n \pm 1).$$

In addition, it is known that

$$f(k+i, n+j) = 0 \quad \forall\, i \leq 0 \quad \forall\, j \tag{11}$$

which we will take into account later.

```
>  ivar:=[k,n]:  dvar:=[f]:
```

We enter the recurrence relations (10):

```
>  L:=[(d-2*k-n)*f(k+1,n+1)-2*m^2*k*f(k+2,n+1)-n*f(k,n+2)-
>  n*(m^2-q^2)*f(k+1,n+2),
>  (n-k)*f(k+1,n+1)-k*(q^2+m^2)*f(k+2,n+1)+
>  k*f(k+2,n)-n*f(k,n+2)-n*(m^2-q^2)*f(k+1,n+2)]:
>  JanetBasis(L,ivar,dvar):
```

Again, the computation time is less than one second. Now, the master integrals are given by:

```
>  ResidueClassBasis(ivar,dvar);
```

$$[\mathrm{f}(k,\, n),\, \mathrm{f}(k,\, n+1),\, \mathrm{f}(k+1,\, n),\, \mathrm{f}(k,\, n+2),\, \mathrm{f}(k+1,\, n+1),\, \mathrm{f}(k+2,\, n)]$$

(11) implies additional relations on the master integrals. (Here, j is recognized as not being contained in `ivar` and thus serves as a parameter to define the additional relations.)

```
>  AddRelation(f(k,n+j)=0,ivar,dvar):
```

The list of master integrals now becomes:

```
>  ResidueClassBasis(ivar,dvar);
```

$$[\mathrm{f}(k+1,\, n),\, \mathrm{f}(k+1,\, n+1),\, \mathrm{f}(k+2,\, n)]$$

Next, we recompute the Janet basis for $m = 0$:

```
>  m:=0:  J:=JanetBasis(L,ivar,dvar):
```

For the special case where $m = 0$, we impose the relation $f(k+i, n) = 0$ for all i:

```
>  AddRelation(f(k+i,n)=0,ivar,dvar):
```

Now, we are left with one master integral:

```
> ResidueClassBasis(ivar,dvar);
```

$$[\mathrm{f}(k+1,\, n+1)]$$

We reduce $f(k+2, n+3)$ modulo J taking also the additionally imposed relations on the master integrals into account. (Here, the option "F" lets `InvReduce` return the result in factorized form.)

```
> InvReduce(f(k+2,n+3),J,"F");
```

$$-((2\,n+4-d+2\,k)\,(2\,n+2-d+2\,k)\,(2\,k+n-d)\,(n+3-d+k)$$
$$(n+2-d+k)\,\mathrm{f}(k+1,\,n+1))/((n+1)\,(2\,n-d+4)\,n\,q^6\,k\,(d-2\,k-2))$$

Using `ResidueClassRelations` one can display the relations imposed on the master integrals:

```
> ResidueClassRelations(ivar,dvar,[i,j]);
```

$$[\mathrm{f}(k,\, n+j),\, \mathrm{f}(k+i,\, n)]$$

The difference operators occurring in the last result can be extracted as polynomials in δ_k, δ_n:

```
> Shift2Pol(%,ivar,dvar,[delta[k],delta[n]]);
```

$$[{\delta_n}^j,\, {\delta_k}^i]$$

10 Conclusion

The above presented algorithm ***Janet-like Gröbner Basis*** is implemented, in its improved form, in the Maple package LDA, and can be applied for generation of finite difference approximations to linear systems of PDEs, to the consistency analysis of such approximations [10], and to reduction of some loop Feynman integrals.

Alternatively, the Gröbner package in Maple in connection with the Ore algebra package [6] can be used to get the same results.

Two of these three applications were illustrated by rather simple examples. The first difference system (discrete Laplace equation and integral relations) contains two independent variables (x, y) and three dependent variables (u, u_x, u_y). The second system (recurrence relations for one-loop Feynman integral) also contains two independent variables/indices (k, n), but the only dependent variable f. The second system, however, is computationally slightly harder than the first one because of explicit dependence of the recurrence relations on the indices and three parameters (d, m^2, q^2) involved in the dependence on indices.

Dependence on index variables and parameters is an attribute of recurrence relations for Feynman integrals. Similar dependence may occur in the generation of difference schemes for PDEs with variable coefficients containing parameters. Theoretically established exponential and superexponential (depends on the ideal and ordering) complexity of constructing polynomial Gröbner bases implies that construction of difference Gröbner bases is at least exponentially hard in the number of independent variables (indices). Besides, in the presence of parameters the volume of computation grows very rapidly as the number of parameters increases.

The reduction of loop Feynman integrals for more than 3 internal lines with masses is computationally hard for the current version of the package. One reason for this is that the Maple implementation does not support Janet trees since Maple does not provide efficient data structures for trees.

Another reason is that in the improved version of the algorithm there is still some freedom in the selection strategy for elements in Q to be reduced modulo G. Though our algorithms are much less sensitive to the selection strategy than Buchberger's algorithm, the running time still depends substantially on the selection strategy: mainly because of dependence of the intermediate coefficients growth on the selection strategy. To find a heuristically good selection strategy one needs to do intensive benchmarking with difference systems. In turn, this requires an extensive data base of various benchmarks that, unlike polynomial benchmarks, up to now is missing for difference systems.

For the problem of reduction of multiloop Feynman integrals recently some new reduction algorithms have been designed (cf. [28] and references therein) that exploit special structure of these integrals and by this reason are computationally much more efficient then the universal Gröbner bases method.

The comparison of implementations of polynomial involutive algorithms for Janet bases in Maple and in C++ [17] shows that the C++ code is of two or three order faster than its Maple counterpart. Together with efficient parallelization of the algorithm this gives a real hope for its practical applicability to problems of current interest in reduction of loop integrals.

Thus, for successful application of the Gröbner basis technique to multiloop Feynman integrals with masses and to multidimensional PDEs with multiparametric variable coefficients one has not only to improve our Maple code but also to implement the algorithms for computing Janet and/or Janet-like difference bases in C++ as a special module of the GINV software [16] available on the web page `http://invo.jinr.ru`.

As to the algorithm ***StandardBasis***, it has not been yet implemented. Another algorithmic development also aimed at computation of Gröbner bases for systems of nonlinear difference polynomials is described in recent paper [29].

11 Achnowledgements

The first author was supported in part by grants 01-01-00200 and 12-07-00294 from the Russian Foundation for Basic Research and by grant 3802.2012.2 from the Ministry of Education and Science of the Russian Federation.

References

[1] V. Gerdt, D. Robertz. *Computation of Gröbner Bases for Systems of Linear Difference Equations.* Computeralgebra-Rundbrief Nr. 37, GI_DMV_GAMM, 2005, 8–13.

[2] V. P. Gerdt, D. Robertz. *A Maple Package for Computing Gröbner Bases for Linear Recurrence Relations.* Nucl. Instrum. Methods 559(1), 2006, pp. 215–219. `arXiv:cs.SC/0509070`

[3] B. Buchberger. *An Algorithm for Finding a Basis for the Residue Class Ring of a Zero-Dimensional Polynomial Ideal.* PhD Thesis, University of Innsbruck, 1965 (in German).

[4] B. Buchberger, F. Winkler (Eds.) *Gröbner Bases and Applications*, Cambridge University Press, 1998.

[5] A. Levin. *Difference Algebra.* Algebra and Applications, vol. 8. Springer, 2008.

[6] F. Chyzak. *Gröbner Bases, Symbolic Summation and Symbolic Integration.* In book [4], pp. 32–60.

[7] A. V. Mikhalev, A. B. Levin, E. V. Pankratiev, M. V. Kondratieva. *Differential and Difference Dimension Polynomials. Mathematics and Its Applications*, Kluwer, Dordrecht, 1999.

[8] V. P. Gerdt, Yu. A. Blinkov, V. V. Mozzhilkin. *Gröbner Bases and Generation of Difference Schemes for Partial Differential Equations.* SIGMA 2, 051, 2006. `arXiv:math.RA/0605334`

[9] B. Martin, V. Levandovskyy. *Symbolic Approach to Generation and Analysis of Finite Difference Schemes of Partial Differential Equations.* In: U. Langer, P. Paule (Eds.) Numerical and Symbolic Scientific Computing: Progress and Prospects. Springer, Wien, 2012, pp. 123–156.

[10] V. P. Gerdt, D. Robertz. *Consistency of Finite Difference Approximations for Linear PDE Systems and its Algorithmic Verification.* In: S. M. Watt (Ed.) Proceedings of ISSAC 2010, Association for Computing Machinery, 2010, pp. 53–59.

[11] V. P. Gerdt. *Consistency Analysis of Finite Difference Approximations to PDE Systems.* Proceedings of MMCP 2011 (July 3-8, 2011, Stará Lesná, High Tatra Mountains, Slovakia), G. Adam, J. Buša, M. Hnatič (Eds.), Lect. Notes Comput. Sci. 7175, Springer, Heidelberg, 2012, pp. 28–42. `arXiv:math.AP/1107.4269`

[12] V. P. Gerdt. *Gröbner Bases in Perturbative Calculations.* Nucl. Phys. B (Proc. Suppl.) 135, 2004, pp. 232–237. `arXiv:/hep-ph/0501053`

[13] V. P. Gerdt, Yu. A. Blinkov. *Janet-like Monomial Division*. Computer Algebra in Scientific Computing / CASC 2005, V. G. Ganzha, E. W. Mayr, E. V. Vorozhtsov (Eds.), Springer-Verlag, Berlin, 2005, pp. 174–183. *Janet-like Gröbner Bases*, ibid., pp. 184–195.

[14] V. P. Gerdt. *Involutive Algorithms for Computing Gröbner Bases*. Computational commutative and non-commutative algebraic geometry. S. Cojocaru, G. Pfister, V. Ufnarovski (Eds.), IOS Press, Amsterdam, 2005, pp. 199–225. `arXiv:math.AC/0501111`

[15] V. P. Gerdt. *On Computation of Gröbner Bases for Linear Difference Systems*. Nucl. Instrum. Methods 559(1), 2006, pp. 211–214. `arXiv:math-ph/0509050`

[16] V. P. Gerdt, Yu. A. Blinkov. *Specialized Computer Algebra System GINV*. Program. Comput. Soft. 34(2), 2008, 112–123.

[17] Yu. A. Blinkov, V. P. Gerdt, C. F. Cid, W. Plesken, D. Robertz. *The Maple Package "Janet": I. Polynomial Systems*. Computer Algebra in Scientific Computing / CASC 2003, V. G. Ganzha, E. W. Mayr, and E. V. Vorozhtsov (Eds.), Institute of Informatics, Technical University of Munich, Garching, 2003, pp. 31–40. Cf. also `http://wwwb.math.rwth-aachen.de/Janet`

[18] A. M. Bigatti, R. La Scala, L. Robbiano. *Computing Toric Ideals*. J. Symb. Comput. 27, 1999, pp. 351–365.

[19] F. Ollivier. *Standard Bases of Differential Ideals*. In: S. Sakata (Ed.), Proceedings of AAECC-8, Lect. Notes Comput. Sci. 508, Springer, New York, 1990, pp.304–321.

[20] T. Becker, V. Weispfenning. *Gröbner Bases: A Computational Approach to Commutative Algebra. Graduate Texts in Mathematics*, vol. 141. Springer, New York, 1993.

[21] D. Cox, J. Little, D. O'Shea. *Ideals, Varieties and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra*. 3rd edition. Springer, New York, 2007.

[22] A. Zobnin. *Admissible Orderings and Finiteness Criteria for Differential Standard Bases*. In: M. Kauers (ed.) Proceedings of ISSAC'05. Association for Computing Machinery, 2010, pp. 365–372.

[23] V. P. Gerdt, D. A. Yanovich. *Experimental Analysis of Involutive Criteria*. Algorithmic Algebra and Logic 2005, A. Dolzmann, A. Seidl, T. Sturm (Eds.), BOD Norderstedt, Germany, 2005, pp. 105–109.

[24] J. Apel, R. Hemmecke. *Detecting Unnecessary Reductions in an Involutive Basis Computation*. J. Symb. Comput. 40(4-5), 2005, pp. 1131–1149.

[25] W. Plesken, D. Robertz. *Janet's approach to presentations and resolutions for polynomials and linear pdes*. Archiv der Mathematik 84 (1), 2005, pp. 22–37.

[26] O. V. Tarasov. *Reduction of Feynman graph amplitudes to a minimal set of basic integrals*. Acta Physica Polonica B29, 1998, pp. 2655–2666. `arXiv:/hep-ph/9812250`

[27] V. A. Smirnov. *Evaluating Feynman Integrals*, STMP 211, Springer, Berlin, 2004.

[28] A. V. Smirnov. *Algorithm FIRE Feynman Integral REduction*. JHEP10, 2008, 107. `arXiv:/hep-ph/0807.3243`

[29] R. La Scala. Gröbner bases and gradings for partial difference ideals. `arXiv:math.RA/1112.2065`

Vladimir P. Gerdt, Daniel Robertz,
Received June 15, 2012

Vladimir P. Gerdt
Laboratory of Information Technologies
Joint Institute for Nuclear Research
141980 Dubna, Russia
E–mail: *gerdt@jinr.ru*

Daniel Robertz
Lehrstuhl B für Mathematik
RWTH Aachen
52062 Aachen, Germany
E–mail: *daniel@momo.math.rwth-aachen.de*