

GRÖBNER BASES APPLIED TO SYSTEMS OF LINEAR DIFFERENCE EQUATIONS

V. P. Gerdt

Joint Institute for Nuclear Research, Dubna

We consider systems of partial (multidimensional) linear difference equations. Specifically, such systems arise in scientific computing under discretization of linear partial differential equations and in computational high energy physics as recurrence relations for multiloop Feynman integrals. The most universal algorithmic tool for investigation of linear difference systems is based on their transformation into an equivalent Gröbner basis form. We present an algorithm for this transformation implemented in Maple. The algorithm and its implementation can be applied to automatic generation of difference schemes for linear partial differential equations and to reduction of Feynman integrals. Some illustrative examples are given.

В представленной статье рассматриваются системы линейных многомерных разностных уравнений. Такие системы возникают, в частности, в вычислительной математике как дискретные версии (разностные схемы для) линейных дифференциальных уравнений в частных производных и в вычислительной физике высоких энергий как рекуррентные соотношения для многопетлевых фейнмановских интегралов. Наиболее универсальным алгоритмическим методом исследования линейных разностных многомерных систем является их преобразование к канонической эквивалентной форме, называемой базисом Гребнера. Представлен алгоритм этого преобразования, реализованный в виде программы на языке Maple. Данный алгоритм и программа могут быть использованы для автоматической генерации разностных схем для линейных уравнений в частных производных и для редукции фейнмановских интегралов. В работе приведены иллюстративные примеры такого использования.

PACS: 02.70.Wz; 02.60.Lj

INTRODUCTION

Let $\mathbb{N}_{>0}$ and $\mathbb{N}_{\geq 0}$ be the sets of positive and nonnegative integers, \mathbb{Q} be the set of rational numbers, $Y := \{y^j(x_1, \dots, x_n) \mid j = 1, \dots, m, m, n \in \mathbb{N}_{>0}\}$ be the set of functions in n variables, and θ_i be the right-shift operator for the i th variable: $\theta_i \circ y(x_1, \dots, x_n) := y(x_1, \dots, x_i + 1, \dots, x_n)$. For the power products $\theta_1^{\mu_1} \dots \theta_n^{\mu_n}$ of the shift operators we shall use the multiindex notation θ^μ , where $\mu := \{\mu_1, \dots, \mu_n\}$ ($\mu \in \mathbb{N}_{\geq 0}^n$) with $|\mu| := \sum \mu_i$. The set of all such operator products will be denoted by Θ .

And then the most general form of a system of $K \in \mathbb{N}_{>0}$ partial ($n > 1$) and multivariate ($m > 1$) linear difference equations is given by

$$a_0 + \sum_{j=1}^m \sum_{\nu} a_{kj;\nu} \vartheta_k^\nu \circ y^j = 0, \quad k = 1, \dots, K, \quad \vartheta_k^\nu \in \Theta, \quad (1)$$

where all sums are finite and coefficients $a_0, a_{kj;\nu}$ may depend on the variables $X := \{x_1, \dots, x_n\}$ and on a finite set of parameters $C := \{c_1, \dots\}$. Hereafter we shall assume

that all coefficients in (1) are rational functions of the variables and parameters with integer coefficients:

$$a_0, a_{kj}; \nu \in \mathbb{Q}(X \cup C). \quad (2)$$

This restriction on the coefficients allows one to apply algorithmic technique of the next section.

It is well known that, except very simple cases, systems of form (1), (2) do not admit exact solutions and are rather poorly studied in the literature [1]. However, such systems play a fundamental role in a number of important applications, for instance in:

Scientific Computing. Numerical solving of linear partial differential systems [2] with rational function coefficients. Recently, it was observed [3] that one can automatically generate finite-difference schemes for such partial differential equations (PDEs) by eliminating partial derivatives from certain linear partial and multivariate difference systems. In so doing, for homogeneous PDEs whose coefficients may also be rational functions of parameters one deals with systems of form (1), (2) with $a_0 = 0$.

Computational High Energy Physics. Reduction of multiloop Feynman integrals [4]. These integrals, after a proper fixed right shift of the variables in X , satisfy the univariate system of partial difference equations (recurrence relations) [5] whose rational function coefficients depend on such physical parameters as the space-time dimension, masses and external momenta. The problem is to reduce the integrals to be evaluated to a minimal set of basic or master integrals, i.e., those integrals which are independent modulo the difference system, and then to express other integrals in terms of the basic ones.

Computational Economics. Characterization of economic behavior in macroeconomics [6]. Here for some macroeconomic problems one has to solve the system (1) with constant parametric coefficients.

To investigate or to solve difference systems (1) with rational coefficients (2), one can use the universal algorithmic Gröbner bases method invented about 40 years ago by Buchberger [7] for systems of multivariate commutative polynomials generating polynomial equations [8]. The main idea of this method is to rewrite the initial system of equations into a certain equivalent form called a Gröbner basis which makes easier investigation of the system and its solving. The underlying Buchberger algorithm [8] is built in all modern general-purpose computer algebra systems such as Maple, Mathematica and others.

On the basis of research made to date, the Gröbner bases theory was extended to some «weakly» noncommutative polynomials as well as to linear differential or difference polynomials and operators [9, 10]. Generally, however, the noncommutative and nonlinear differential or difference Gröbner bases may not exist (be infinite). For difference systems (1), (2) Gröbner bases are always finite and can be constructed by Buchberger's algorithm straightforwardly translated to difference algebra [10].

Recently [3, 11] we have presented the difference form of our polynomial algorithm devised in [12], improved in [13] and specialized to the so-called Janet and Janet-like monomial divisions [14] which go back to the constructive ideas of French mathematician Janet [15]. The algorithm constructs a Janet(-like) basis [14] which is also a Gröbner basis. Though generally Janet bases [12] and Janet-like bases are redundant as Gröbner ones, the algorithm in its improved version [3, 13] allows one also to output reduced Gröbner bases without any additional computational costs. The implementation [16] of the algorithm in Maple allows a user to compute linear difference Janet(-like Gröbner) bases.

In the present paper we describe briefly a simple version of the Janet division algorithm (Sec. 1) and consider its application to the above-listed problems from scientific computing (Sec. 2) and computational high energy physics (Sec. 3). Both problems are purely algebraic and can be completely solved with the use of Gröbner bases. We illustrate this fact by simple examples. Our presentation is addressed to non-algebraists. For this reason we slightly abuse algebraic terminology and refer to the references in bibliography for more precious definitions and notions.

1. TRANSFORMATION TO GRÖBNER BASIS

In this section we define the concept of a Gröbner basis form for the difference system (1), (2) and present an algorithm for its computation. The Gröbner basis form of the system (1) is defined by a ranking \succ (linear order on) of terms $\theta^\mu \circ y^j$ and such that for all i, j, k, μ, ν the following holds:

$$\theta_i \theta^\mu \circ y^j \succ \theta^\mu \circ y^j, \quad \theta^\mu \circ y^j \succ \theta^\nu \circ y^k \iff \theta_i \theta^\mu \circ y^j \succ \theta_i \theta^\nu \circ y^k.$$

If $|\mu| \succ |\nu| \implies \theta^\mu \circ y^j \succ \theta^\nu \circ y^k$, the ranking is called *orderly*. If $j > k \implies \theta^\mu \circ y^j \succ \theta^\nu \circ y^k$, the ranking is called *elimination*.

Denote by f_k the left-hand side of the k th equation in (1) and by $F := \{f_1, \dots, f_K\}$ the set of all the left-hand sides in the system. Fixing a ranking \succ provides every $f \in F$ with the *leading term* $\text{lt}(f) := a \vartheta \circ y^j$ ($\vartheta \in \Theta$, $a \neq 0$) and *leading coefficient* $\text{lc}(f) := a$. Furthermore, denote by $R \supset F$ the set of all right-hand sides $f \neq 0$ for linear difference equations $f = 0$ which are consequences of the system (1), (2). F is called *generating set or basis* of R (denotation: $R = \langle F \rangle$). In what follows we shall assume that, given a ranking \succ , all $f \in R$ are normalized, that is, divided by their leading coefficients. If $F \subseteq \mathbb{R}$, then $\text{lt}(F)$ will denote the set of the leading terms and $\text{lt}_j(F)$ will denote its subset for function y^j . Therefore,

$$\text{lt}(F) = \bigcup_{j=1}^m \text{lt}_j(F).$$

Now we are ready to define a *Gröbner basis* for given F and ranking \succ as a finite subset $G \subset R = \langle F \rangle$ such that $R = \langle G \rangle$ and

$$\forall f \in R, \exists g \in G, \theta \in \Theta : \text{lt}(f) = \theta \circ \text{lm}(g). \quad (3)$$

It follows that the leading term of every $f \in R$ is *reducible modulo G* and yields *the head reduction*:

$$f \xrightarrow{g} f' := f - \theta \circ g, \quad f' \in R.$$

If $f' \neq 0$, then its leading term is again reducible modulo G . And then by repeating the reduction finitely many times [8–10] we obtain $f \xrightarrow{G} 0$. Generally, if a linear difference expression h (not necessarily from R) contains a term u with coefficient $c \neq 0$ such that $u = c \vartheta \circ \text{lt}(f)$ for some $\vartheta \in \Theta$ and $f \in F \subset \mathbb{R}$, then h can be reduced:

$$h \xrightarrow{g} h' := h - c \vartheta \circ f. \quad (4)$$

By applying the reduction finitely many times, one obtains a polynomial \bar{h} which is either zero or such that all its (nonzero) terms are *irreducible modulo set F*. In both cases \bar{h} is said to be in the *normal form modulo F* (denotation: $\bar{h} = NF(h, F)$). A Gröbner basis G is called *reduced* if $g = NF(g, G \setminus \{g\})$ for every $g \in G$.

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

For a finite set F and a ranking \succ , we partition every set $\text{lt}_k(F)$ into groups labeled by $d_0, \dots, d_i \in \mathbb{N}_{\geq 0}$ ($0 \leq i \leq n$). Here $[0]_k := \text{lt}_k(F)$ and for $i > 0$ the group $[d_0, \dots, d_i]_k$ is defined as

$$[d_0, \dots, d_i]_k := \{u \in \text{lt}_k(F) \mid d_0 = 0, d_j = \deg_j(u), 1 \leq j \leq i\},$$

where $\deg_i(\theta^\mu \circ y^k) := \mu_i$. Operator θ_i is called *J(anet)-multiplicative* for $f \in F$ if $\text{lt}(f) \in [d_0, \dots, d_{i-1}]$ and $\deg_i(u) = \max\{\deg_i(v) \mid v \in [d_0, \dots, d_{i-1}]\}$. Denote by $M_J(f, F)$ the set of *J(anet)-multiplicative* shift operators for $f \in F$, the complement set $\{\theta_1, \dots, \theta_n\} \setminus M_J(f, F)$ of *J(anet)-nonmultiplicative* shift operators by $NM_J(f, F)$ and the set of all possible power products of *J*-multiplicative operators (including identity operator) by $J(f, F)$. It is clear that $J(f, F) \subset \Theta$.

A finite set $G \in R = \langle F \rangle$ is called a *Janet basis* (cf. [13]) if

$$\forall f \in R, \exists g \in G, \theta \in J(g, G) : \text{lt}(f) = \theta \circ \text{lt}(g). \quad (5)$$

Similarly to (4), a *J-reduction* is defined as

$$h \xrightarrow[g]{} h' := h - c\vartheta \circ f, \quad \vartheta \in J(f, F), \quad (6)$$

for a polynomial $h \in R$ containing a term u with coefficient $c \neq 0$ satisfying $u = c\vartheta \circ \text{lt}(f)$ for some $f \in F$ and $\vartheta \in J(f, F)$.

Since *J*-reducibility (6) implies the Gröbner reducibility (4), a Janet basis satisfying (5) is also a Gröbner basis. The converse is generally not true; that is, not every Gröbner basis is a Janet one. The algorithmic characterization of a Janet basis G is the following condition (cf. [13]):

$$\forall g \in G, \theta \in NM(g, G) : NF_J(\theta \circ g, G) = 0, \quad (7)$$

which is a cornerstone of the below algorithm for construction of Janet bases (3).

This algorithm is a translation (with some minor modifications) of the polynomial algorithm in [17] into the difference case. Due to the normalization of h done at Step 15 before insertion of h into the intermediate basis G , the algorithm outputs the minimal and normalized Janet basis which is uniquely defined by an input difference system F and a ranking [12]. Correctness and termination of the difference algorithm immediately follow from those for its polynomial counterpart [12, 13]. Algorithm **JanetBasis** implemented in its improved form [3] as the Maple package LDA (abbreviates Linear Difference Algebra) [16], and in the next two sections computation with the package is illustrated by examples.

Algorithm: JanetBasis (F, \succ)

```

Input:  $F$ , a finite linear difference set;  $\succ$ , a ranking
Output:  $G$ , a Janet basis of  $\langle F \rangle$ 
1: choose  $f \in F$  with the lowest  $\text{lt}(f)$  w.r.t.  $\succ$ 
2:  $G := \{f\}$ 
3:  $Q := F \setminus G$ 
4: do
5:    $h := 0$ 
6:   while  $Q \neq \emptyset$  and  $h = 0$  do
7:     choose  $p \in Q$  with the lowest  $\text{lt}(p)$  w.r.t.  $\succ$ 
8:      $Q := Q \setminus \{p\}$ 
9:      $h := NF_J(p, G)$ 
10:  od
11:  if  $h \neq 0$  then
12:    for all  $g \in G$  such that  $\text{lt}(g) = \theta^\mu \circ \text{lt}(h)$ ,  $|\mu| > 0$  do
13:       $Q := Q \cup \{g\}$ ;  $G := G \setminus \{g\}$ 
14:    od
15:     $G := G \cup \{h/\text{lc}(h)\}$ 
16:     $Q := Q \cup \{\theta^\beta \circ g \mid g \in G, \theta^\beta \in NM_J(g, G)\}$ 
17:  fi
18: od while  $Q \neq \emptyset$ 
19: return  $G$ 

```

2. GENERATION OF DIFFERENCE SCHEMES

In paper [3] an algorithmic approach was developed to construct finite-difference schemes for linear PDEs in two independent variables and on uniform orthogonal grids with possibly distinct mesh steps for x and y . We outline here the main idea of the approach and refer to [3] for more details. In so doing, we restrict our consideration by scalar equations of order ≥ 2 which admit the conservation law form

$$\frac{\partial V}{\partial x} + \frac{\partial W}{\partial y} = 0, \quad (8)$$

where V and W are functions of independent variables x, y , dependent variable $u(x, y)$ and its partial derivatives u_x, u_y, u_{xx}, \dots . The differential equation (8) can be rewritten in the integral form

$$\oint_{\Gamma} -W dx + V dy = 0, \quad (9)$$

which is valid for arbitrary closed contour Γ . Discretization of (9) instead of (8) is natural for preserving the conservation law at the discrete level (conservative scheme).

Denote the grid values of function $u(x, y)$ and its derivatives by

$$\begin{aligned} u_{jk} &:= u(x_j, y_k), & (u_x)_{jk} &:= u_x(x_j, y_k), & (u_y)_{jk} &:= u_y(x_j, y_k), \\ (u_{xx})_{jk} &:= u_{xx}(x_j, y_k), & \dots & \end{aligned} \quad (10)$$

and fix some integration contour Γ in (9) on the grid. To be specific, let us choose a simple rectangular contour as shown in Fig. 1.

Now we add to the integral equation (9) for the rectangular contour of Fig. 1 all the related (exact) integral relations between $u(x, y)$ and its partial derivatives:

$$\left\{ \begin{array}{l} \int_{x_j}^{x_{j+2}} u_x dx = u(x_{j+2}, y) - u(x_j, y), \quad \int_{y_k}^{y_{k+2}} u_y dy = u(x, y_{k+2}) - u(x, y_k), \\ \int_{x_j}^{x_{j+2}} u_{xx} dx = u_x(x_{j+2}, y) - u_x(x_j, y), \quad \int_{y_k}^{y_{k+2}} u_{xy} dy = u_x(x, y_{k+2}) - u_x(x, y_k), \\ \dots \end{array} \right. \quad (11)$$

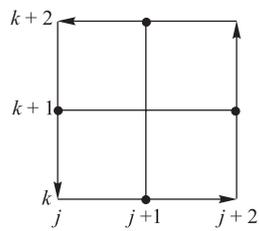


Fig. 1. Integration contour on grid

Our purpose is to obtain a difference scheme for $u_{j k}$ from a proper discretization of integral equations (9) and relations (11). To do that one should use as many relations in (11) as the number of all proper derivatives of u up to the maximal orders of their occurrence in the integrand of (9). Then the difference scheme can be obtained by an algebraic difference elimination of all discrete proper partial derivatives in list (10) from the combined system (9), (11). The algebraic elimination can be achieved by computing a Gröbner or Janet basis for the last system and a suitable elimination ranking (see Sec. 2) satisfying $u_{j k} \prec (u_x)_{j k} \prec (u_{xx})_{j k} \prec \dots$

Therefore, to construct an initial system of discrete equations for the following difference elimination, it suffices to approximate numerically the contour integral (9) for the chosen contour of Fig. 1 together with the integral relations (11) in terms of the grid unknowns (10). For this purpose one can choose various quadrature formulas for these integrals, and the difference scheme obtained may depend on the choice. For simplicity sake we apply here for all the integrals in (9) and (11) the simplest rectangle (midpoint) rule:

$$\left\{ \begin{array}{l} (W_{j+1 k+2} - W_{j+1 k})h_1 + (V_{j+2 k+1} - V_{j k+1})h_2 = 0, \\ (u_x)_{j+1 k} 2h_1 = u_{j+2 k} - u_{j k}, \\ (u_y)_{j k+1} 2h_2 = u_{j k+2} - u_{j k}, \\ \dots \end{array} \right. \quad (12)$$

where $h_1 := x_{j+1} - x_j$ and $h_2 := y_{k+1} - y_k$ are the grid mesh steps for our uniform orthogonal grid.

For linear difference system (12) Janet (Gröbner) basis exists for any ranking, and, hence, the elimination can be performed by applying the above algorithm **JanetBasis**. To illustrate this algorithmic procedure for the difference schemes generation, consider a simple example of the Heat equation in its conservation law form [3]:

$$u_t + \alpha u_{xx} = 0 \implies \oint_{\Gamma} -\alpha u_x dt + u dx = 0, \quad (13)$$

where α is a symbolic parameter. The integrand in (13) contains the only partial derivative u_x . Hence, we need to add the only integral relation

$$\int_{x_j}^{x_{j+1}} u_x dx = u(x_{j+1}, t) - u(x_j, t). \tag{14}$$

Now consider $u(x, t)$ and $u_x(x, t)$ on the uniform orthogonal grid with the spatial mesh step h and the temporal mesh step τ , and choose the simplest contour shown in Fig. 2. As this takes place, we can approximate the integral of $u_x(x, t)$ over x in (13), (14) on the grid points by the rectangular or trapezoidal rules.

Then, applying the midpoint rule for the contour integral and the trapezoidal rule for the relation integral, we find two difference equations for two dependent variables u, u_x :

$$\begin{cases} \alpha \frac{\tau}{2} (1 + \theta_t - \theta_x^2 - \theta_t \theta_x^2) \circ u_x - 2h (\theta_x \theta_t - \theta_x) \circ u = 0, \\ \frac{h}{2} (\theta_x + 1) \circ u_x - (\theta_x - 1) \circ u = 0. \end{cases} \tag{15}$$

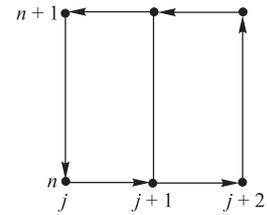


Fig. 2. Integration contour for the heat equation

Furthermore, we show how to generate a finite-difference scheme for the Heat equation (13) by using the Maple package LDA [16]:

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

First, we enter the independent and dependent variables for the problem.

```
> ivar := [j,k]; dvar := [ux,u]:
```

Second, we translate (15) into the input format of the main command JanetBasis in the package.

```
> L := [a*t/2*(ux(j,k)+ux(j+1,k)-ux(j,k+2)-ux(j+1,k+2))-
> 2*h*(u(j+1,k+1)-u(j,k+1)),
> h/2*(ux(j,k+1)+u(j,k)-u(j,k+1)+u(j,k))]:
```

Third, we compute the (minimal) Janet basis for L w.r.t. an elimination ranking with $u_x \succ u$ to eliminate the partial derivative u_x from the system (15). This ranking is chosen by using option 2 as below; in so doing we output only the element in Janet basis which does not contain u_x .

```
> JanetBasis(L, ivar, dvar, 2) [1] [1];
-2 a t u(j, k + 1) + h a t u(j, k) + 2 a t u(j, k) + 2 a t u(j, k + 3) - h a t u(j, k + 2)
-2 a t u(j, k + 2) + 2 a t u(j + 1, k + 3) - h a t u(j + 1, k + 2) - 2 a t u(j + 1, k + 2)
+4 h^2 u(j + 1, k + 2) - 4 h^2 u(j, k + 2) - 2 a t u(j + 1, k + 1) + h a t u(j + 1, k)
+2 a t u(j + 1, k)
```

Thereby, we obtain the classical Crank–Nicolson scheme

$$\frac{u_k^{j+1} - u_k^j}{\tau} + \alpha \frac{(u_{k+1}^{j+1} - 2u_k^{j+1} + u_{k-1}^{j+1}) + (u_{k+1}^j - 2u_k^j + u_{k-1}^j)}{2h^2} = 0,$$

if in the above Maple output one shifts the second index by -1 and uses the first index as a superscript. The same scheme is also obtained for the midpoint integration method applied to (14).

3. REDUCTION OF FEYNMAN INTEGRALS

Evaluation of Feynman integrals is the cornerstone step of perturbative computations in elementary particle physics [4]. Consider, for example, a typical scalar L -loop integral with E external legs:

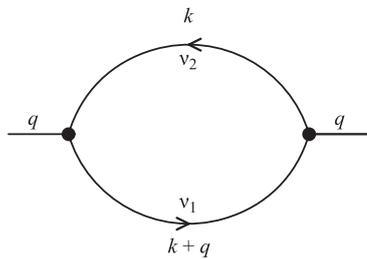
$$\mathcal{I}(\nu_1, \dots, \nu_n) = \int d^d k_1 \cdots d^d k_L \frac{1}{\prod_{j=1}^n D_j^{\nu_j}}, \tag{16}$$

which corresponds to n internal lines in the related Feynman diagram. Integration for every loop momentum k_i is done over the space-time of dimension $d = 4 - \epsilon$, where ϵ is the parameter of dimensional regularization [18]. The denominator D_j for the j th internal line with mass m_j is given by $D_j := p_j^2 - m_j^2$. Here the line momenta p_j are linearly expressed in terms of the loop momenta k_i ($i = 1, \dots, L$) and external momenta q_s ($s = 1, \dots, E$) as

$$p_j = \sum_{s=1}^L \alpha_{js} k_s + \sum_{t=1}^E \beta_{jt} q_t, \quad \alpha_{js}, \beta_{jt} \in \mathbb{Q}.$$

Consider the combined set of $L + E$ vectors

$$r_a := \begin{cases} k_a, & a = 1, \dots, L, \\ q_{a-L}, & a = L + 1, \dots, L + E. \end{cases}$$



Recurrence relations for integral (16) are derived by the integration-by-parts method [5], whose main idea is to use the integral identities (cf. [4, 19])

$$\int d^d k_1 \cdots d^d k_L \frac{\partial}{\partial k_i} \frac{r_j}{\prod_{k=1}^n D_k^{\nu_k}} = 0 \tag{17}$$

together with the d -vector identities

Fig. 3. One-loop propagator diagram $2p_i q_j = (p_i + q_j)^2 - (p_i^2 - m_i^2) - (q_j^2 + m_i^2). \tag{18}$

Integral identities (17) follow from an observation that any integral of $\partial/\partial k_i(\dots)$ vanishes since there are no surface terms in dimensional regularization (cf. [20]).

As a simple example consider one-loop propagator diagram of Fig. 3 with external momentum q and with one massive and another massless internal lines. This example was studied, for example, in [4, 21] and used already in [16]. The corresponding Feynman integral (16) is given by

$$\mathcal{I}(\nu_1, \nu_2) = \int \frac{d^d k}{D_1^{\nu_1} D_2^{\nu_2}}, \quad D_1 := (k + q)^2, \quad D_2 := k^2. \quad (19)$$

In accordance with (17), there are two independent integral identities. Their integrands are

$$\begin{aligned} \frac{\partial}{\partial k} \frac{k}{D_1^{\nu_1} D_2^{\nu_2}} &= \frac{1}{D_1^{\nu_1} D_2^{\nu_2}} \left[\frac{1}{\partial k} k - \frac{2\nu_1(k^2 + q \cdot k)}{D_1} - \frac{2\nu_2 k^2}{D_2} \right], \\ \frac{\partial}{\partial k} \frac{q}{D_1^{\nu_1} D_2^{\nu_2}} &= \frac{1}{D_1^{\nu_1} D_2^{\nu_2}} q \left[-\frac{2\nu_1(k + q)}{D_1} - \frac{2\nu_2 k}{D_2} \right]. \end{aligned}$$

Taking into account the identity $2k \cdot q = (k + q)^2 - (k^2 - m^2) - (q^2 + m^2)$ of the type (18) and equality $\partial/\partial k \cdot k = d$ we obtain the difference system

$$\begin{cases} [d - \nu_1 - 2\nu_2 - \nu_1 \theta_1 \theta_2^{(-1)} + \nu_1(q^2 - m^2)\theta_1 - 2m^2 \nu_2 \theta_2] \circ \mathcal{I}(\nu_1, \nu_2) = 0, \\ [\nu_1 - \nu_2 + \nu_1(q^2 - m^2)\theta_1 - \nu_1 \theta_1 \theta_2^{(-1)} + \nu_2 \theta_1^{(-1)} \theta_2 - \nu_2(q^2 + m^2)\theta_2] \circ \mathcal{I}(\nu_1, \nu_2) = 0, \end{cases} \quad (20)$$

where $\theta_i^{(-1)}$ and $\theta_2^{(-1)}$ denote the left-shift operators for indices ν_1 and ν_2 , respectively.

Now we construct the minimal set of master or basic integrals for the two-indexed family (19) of Feynman integrals by applying the Maple package LDA [16] with the input denotations $k := \nu_1, n := \nu_2$ and $f(k + 1, n + 1) := \mathcal{I}(\nu_1, \nu_2)$:

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

Then, we enter the recurrence relations (20).

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

As the next step we compute a Janet basis for an orderly ranking (Sec. 1) induced by $\theta_1 \succ \theta_2$.

```
> JB := JanetBasis(L, ivar, dvar):
```

In order to compute the set of master integrals we have to take into account that $\mathcal{I}(\nu_1, \nu_2) = 0$ for $\nu_2 \leq 0$ [4, 20]. This extra *boundary* information is input as

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

Master integrals are those $f(k, n)$ which are independent modulo all the consequences R (see Sec. 1) of (20). Thereby, the master integrals are easily determined via the leading terms of the Janet basis. Namely, one has to determine those $f(k, n)$ that are not expressible as the action of a power product $\theta_1^{\mu_1} \theta_2^{\mu_2}$ ($\mu_1, \mu_2 \in \mathbb{N}_{\geq 0}$) on a leading term in the Janet basis (cf. definition (3)).

The set of master integrals is computed by invoking the command:

```
> ResidueClassBasis(ivar,dvar);
      [f(k, n + 1), f(k, n + 2), f(k + 1, n + 1)].
```

Now any integral $f(k+i, n+j)$ can be explicitly expressed as a linear combination of the master integrals whose coefficients are rational functions in parameters d, q^2 and m^2 . The explicit expression is obtained algorithmically by applying the Gröbner or Janet reductions described in Sec. 1. In LDA the Janet reductions are performed. To show the output of such an expression for $f(k+3, n+2)$ and make the output more compact, we let $m=0$ and show the underlying piece of the Maple code:

```
> m:=0: J:=JanetBasis(L,ivar,dvar):
> AddRelation(f(k,n+j)=0,ivar,dvar):
> ResidueClassBasis(ivar,dvar);
      [f(k + 1, n + 1)]
> InvReduce(f(k+3,n+2),J,"F");
-((d - 2 - 2k - 2n)(d - 4 - 2k - 2n)(d - 2 - k - n)(d - 3 - k - n)(-n - 2k + d) ×
× f(k + 1, n + 1))/(q6(k + 1)(-2k + d - 4)k(-2k - 2 + d)n)-1.
```

In the massless case ($m=0$) a new extra relation $f(k, n+j)$ equivalent to the *boundary condition* $\mathcal{I}(\nu_1, \nu_2) = 0$ for $\nu_1 \leq 0$ is added that yields the only master integral $f(k+1, n+1)$. The last command invokes the LDA procedure that computes the J -normal form of $f(k+3, n+2)$ modulo the Janet basis. This normal form just represents $f(k+3, n+2)$ in terms of the master integral. Option «F» provides factorization of the numerator and denominator in the output rational function coefficient. It should be noted that, since integral $\mathcal{I}(\nu_1, \nu_2)$ is non-vanishing only when both its indices are positive, the master integral can be identified with $\mathcal{I}(1, 1)$.

CONCLUSION

We have shown above that the Gröbner bases technique can be applied to generate difference schemes for linear PDEs and to reduce multiloop Feynman integrals. Each of our simple illustrative examples of Secs. 2 and 3 needs less than 1 s of computing time on a 1.7 MHz personal computer with 512 Mb RAM. Larger examples, however, can require much more computer resources since complexity of a Gröbner basis computation is at least singly exponential, and may be even doubly exponential, in a number of variables [9, 22]. Besides, blowing-up of intermediate coefficients, especially in the presence of parameters, as in the case of recurrence relations for Feynman integrals, is a serious obstacle in the practice. That is why to apply Gröbner bases to multivariate and multiparametric problems, one has not only to optimize and improve the underlying algorithms and data structures, but also to implement them in lower level languages than Maple or Mathematica. Our Janet division algorithms [13] have already been implemented in C and C++ (see the Web page [23]) for commutative polynomials, and extension of these codes to differential and difference equations is planned for the coming years. As argued in [21, 24], differential Gröbner bases can also

be applied to reduction of Feynman integrals. A practical specialization of the Gröbner bases ideas to reduction of Feynman integrals was suggested recently in [25,26] where the whole index space for integrals (16) is partitioned into the so-called *sectors* in accordance with the extra *boundary conditions*. Then in every sector a certain kind of a Gröbner-like basis is constructed. As to an extended discussion of generating difference schemes by means of Gröbner bases we refer to our recent paper [3].

Acknowledgements. The research presented in the paper was partially supported by grants 04-01-00784 and 05-02-17645 from the Russian Foundation for Basic Research and by grant 5362.2006.2 from the Ministry of Education and Science of the Russian Federation.

REFERENCES

1. Kelley W. G., Petersen A. C. *Difference Equations: An Introduction with Applications*. Acad. Press, 2000.
2. Hildebrand F. B. *Finite-Difference Equations and Simulations*. Prentice-Hall, 1968.
3. Gerdt V. P., Blinkov Yu. A., Mozzhilkin V. V. Gröbner Bases and Generation of Difference Schemes for Partial Differential Equations // SIGMA. 2006. 2. 051; math.RA/0605334.
4. Smirnov V. A. *Evaluating Feynman Integrals*. Springer-Verlag, 2004.
5. Tkachov F. V. A Theorem on Analytical Calculability of 4-Loop Renormalization Group Functions // Phys. Lett. B. 1981. V. 100. P. 65–68;
Chetyrkin K. G., Tkachov F. V. Integration by Parts: The Algorithm to Calculate β Functions in 4 Loops // Nucl. Phys. B. 1981. V. 192. P. 159–204.
6. Wang P. F., Wen Y. Solving Linear Difference Systems with Lagged Expectations by a Method of Undetermined Coefficients. Working Paper 2006-003C. Federal Reserve Bank of St. Louis, 2006; <http://research.stlouisfed.org/wp/more/2006-003/>
7. Buchberger B. An Algorithm for Finding a Basis for the Residue Class Ring of a Zero-Dimensional Polynomial Ideal. PhD Thesis. Univ. of Innsbruck, 1965 (in German).
8. Buchberger B. Gröbner Bases: an Algorithmic Method in Polynomial Ideal Theory // Recent Trends in Multidimensional System Theory. Reidel, 1985. P. 184–232.
9. Buchberger B., Winkler F. *Gröbner Bases and Applications*. Cambridge Univ. Press, 1998.
10. Kondratieva M. V. et al. *Differential and Difference Dimension Polynomials*. Kluwer, 1999.
11. Gerdt V. P. On Computation of Gröbner Bases for Linear Difference Systems // Nucl. Instr. Meth. A. 2006. V. 559(1). P. 211–214; math-ph/0509050.
12. Gerdt V. P., Blinkov Yu. A. Involutive Bases of Polynomial Ideals // MATCOM. 1998. V. 45. P. 519–542; math.AC/9912027; Minimal Involutive Bases // Ibid. P. 543–560; math.AC/9912029.
13. Gerdt V. P. Involutive Algorithms for Computing Gröbner Bases // Computational Commutative and Non-Commutative Algebraic Geometry. IOS Press, 2005. P. 199–225; math.AC/0501111.
14. Gerdt V. P., Blinkov Yu. A. Janet-Like Monomial Division. Janet-Like Gröbner Bases // Comp. Algebra in Sci. Comp. Springer, 2005. LNCS 3781. P. 174–195.

15. *Janet M.* Leçons sur les Systèmes d'Equations aux Dérivées Partielles // Cahiers Scientifiques. IV. Gauthier-Villars, 1929.
16. *Gerdt V.P., Robertz D.* A Maple Package for Computing Gröbner Bases for Linear Recurrence Relations // Nucl. Instr. Meth. A. 2006. V. 559(1). P. 215–219; cs.SC/0509070.
17. *Gerdt V.P., Blinkov Yu.A.* On Computing Janet Bases for Degree Compatible Orderings // Proc. of the 10th Rhine Workshop on Computer Algebra. Univ. of Basel, 2006. P. 107–117; math.AC/0603161.
18. *'t Hooft G., Veltman M.* Regularization and Renormalization of Gauge Fields // Nucl. Phys. B. 1972. V. 44. P. 189–213.
19. *Smirnov V.A., Steinhauser M.* Solving Recurrence Relations for Multi-Loop Feynman Integrals // Nucl. Phys. B. 2003. V. 672. P. 199–201; hep-ph/0307088.
20. *Grozin A.G.* Lectures on Multiloop Calculations // Intern. J. Mod. Phys. A. 2004. V. 19, No. 4. P. 473–520; hep-ph/0307297.
21. *Tarasov O.V.* Reduction of Feynman Graph Amplitudes to a Minimal Set of Basic Integrals // Acta Phys. Polon. B. 1998. V. 29. P. 2655–2666; hep-ph/9812250.
22. *von zur Gathen J., Gerhard J.* Modern Computer Algebra. 2nd ed. Cambridge Univ. Press, 2003.
23. <http://invo.jinr.ru>
24. *Tarasov O.V.* Computation of Gröbner Bases for Two-Loop Propagator Type Diagrams // Nucl. Instr. Meth. A. 2006. V. 534(1–2). P. 293–298; hep-ph/0403253.
25. *Smirnov V.A., Smirnov A.V.* Applying Groebner Bases to Solve Reduction Problems for Feynman Integrals // JHEP. 2006. 0601. 001; hep-lat/0509187.
26. *Smirnov A.V.* An Algorithm to Construct Gröbner Bases for Solving Integration by Parts Relations // Ibid. 0604. 026; hep-ph/0602078.