Gröbner Bases and Generation of Difference Schemes for Partial Differential Equations

Vladimir P. GERDT [†], *Yuri A. BLINKOV* [‡] *and Vladimir V. MOZZHILKIN* [‡]

[†] *Laboratory of Information Technologies, Joint Institute for Nuclear Research, 141980 Dubna, Russia*
E-mail: *gerdt@jinr.ru*
URL: http://compalg.jinr.ru/CAGroup/Gerdt/

[‡] *Department of Mathematics and Mechanics, Saratov University, 410071 Saratov, Russia*
E-mail: *BlinkovUA@info.sgu.ru*
URL: http://www.sgu.ru/faculties/mathematics/chairs/alg/blinkov.php

Received December 07, 2005, in final form April 24, 2006; Published online May 12, 2006
Original article is available at http://www.emis.de/journals/SIGMA/2006/Paper051/

Abstract. In this paper we present an algorithmic approach to the generation of fully conservative difference schemes for linear partial differential equations. The approach is based on enlargement of the equations in their integral conservation law form by extra integral relations between unknown functions and their derivatives, and on discretization of the obtained system. The structure of the discrete system depends on numerical approximation methods for the integrals occurring in the enlarged system. As a result of the discretization, a system of linear polynomial difference equations is derived for the unknown functions and their partial derivatives. A difference scheme is constructed by elimination of all the partial derivatives. The elimination can be achieved by selecting a proper elimination ranking and by computing a Gröbner basis of the linear difference ideal generated by the polynomials in the discrete system. For these purposes we use the difference form of Janet-like Gröbner bases and their implementation in Maple. As illustration of the described methods and algorithms, we construct a number of difference schemes for Burgers and Falkowich–Karman equations and discuss their numerical properties.

Key words: partial differential equations; conservative difference schemes; difference algebra; linear difference ideal; Gröbner basis; Janet-like basis; computer algebra; Burgers equation; Falkowich–Karman equation

2000 Mathematics Subject Classification: 68W30; 65M06; 13P10; 39A05; 65Q05

1 Introduction

It is well-known that finite differences along with finite elements and finite volumes are most important discretization schemes for numerical solving of partial differential equations (PDEs) (see, for example, [1, 2, 3, 4, 5, 6, 7, 8]).

Mathematical operations used in the construction of difference schemes for PDEs are substantially symbolic. Thereby, it is a challenge for computer algebra to provide an algorithmic tool for automatization of the difference schemes constructing as well as for the investigation of properties of the difference schemes. One of the most fundamental requirements for a difference scheme is its stability which can be analyzed with the use of computer algebra methods and software [9].

Furthermore, if PDEs admit a conservation law form or/and have some symmetries, it is worthwhile to preserve these features at the level of difference schemes too. In particular, a tool for automatic construction of difference schemes should produce conservative schemes whenever the original PDEs can be written in the integral conservation law form. One of such tools GRIDOP written in Reduce [10, 11] is based on symbolic operator methods and generates conservative finite-difference schemes on rectangular domains in an arbitrary number of independent variables. However, the generation is not entirely automatic. A user of GRIDOP has to specify function spaces together with associated scalar products and define grid operators as finite-difference schemes. Then the user may provide partial differential equations in terms of the defined grid operators or the adjoints of those operators. Under these conditions the package returns the finite-difference equations for the dependent variables.

Besides, a few other applications of computer algebra are known to construct finite-difference schemes [12, 13] which, being also not completely automatic, are applicable to PDEs of a certain form.

In this paper we describe a universal algorithmic approach to the automatic generation of conservative difference schemes for linear PDEs with two independent variables admitting the conservation law form. This approach generalizes and extends the observations of paper [14] where it was noticed that a conservative difference scheme can be derived as a compatibility condition for a system of difference equations. The system is composed of a discrete form of the original PDEs taken in the integral conservation law form and of a number of natural integral relations between functions and their partial derivatives. The finite-difference scheme is obtained by elimination of all the partial derivatives from the system. We also show, by the example of Burgers equation, that one can also apply the difference elimination approach to generate of difference schemes without use of conservation law form.

To perform the difference elimination we apply the Gröbner bases method invented 40 years ago by Buchberger [15] for polynomial ideals. This method has become the most universal algorithmic tool in commutative algebra and algebraic geometry and found also numerous fruitful applications for computations in certain noncommutative polynomial rings as well as in rings of linear differential operators and differential polynomials [16]. Nowadays, all modern general-purpose computer algebra systems, for example, Maple [17] and Mathematica [18], have special built-in modules implementing algorithms for computing Gröbner bases. However, the fastest implementation of these algorithms for commutative polynomial algebra is done in the special-purpose systems Singular [19] and Magma [20]. As to the difference algebra [21], in spite of known for long time (see [22] and references therein) extensions of Buchberger's algorithm [23] to difference polynomial rings, there are only a few implementations of the algorithm specialized to shift Ore algebra: in the Ore algebra library package of Maple [24], in the library *OreModules* [25] developed using the latter package and in Singular (Plural) [26]. These packages can be used for computing Gröbner bases of linear difference ideals and modules, and, in particular, for those linear systems which are considered below.

In the given paper we present, however, another algorithm for computing difference Gröbner bases. This algorithm is superior over our Janet division algorithm whose polynomial version [27] in most cases is computationally more efficient than Buchberger's algorithm [28]. In addition, unlike the above mentioned implementations of Buchberger's algorithm for the shift Ore algebra, the algorithm described below and its recent implementation in Maple [29] admit a natural extension to nonlinear difference systems exactly in the same way as differential involutive algorithms [30, 31, 32]. The algorithm improves our Janet-like division algorithm [33] adapted to linear difference ideals [34]. The improvement includes, in particular, the difference form of the involutive criteria [27] modified for Janet-like reductions. These criteria allow to avoid some useless reductions, and thereby accelerate the computation.

The structure of the paper is as follows. In Section 2 we describe the basic idea of our approach to the generation of finite-difference schemes for two-dimensional PDEs. Section 3 contains definitions and notions of difference algebra which are used in the sequel. Here we define Gröbner bases for linear difference ideals and their special form called Janet-like bases.

In Section 4 we present an improved version of the algorithm in paper [34] and briefly discuss some relevant computational aspects. Section 5 illustrates our approach to construction of difference schemes by simplest second-order equations – Laplace's equation, the wave equation, the heat equation, and by the first-order advection equation. In Section 6 we generate several difference schemes for Burgers equation. But for all that, to avoid problems arising in computing of nonlinear Gröbner bases, we denote the square of the dependent variable by an extra function. Besides, we characterize some of the constructed schemes by the modified equation method. In Section 7 we consider the two-dimensional quadratically nonlinear Falkowich–Karman equation describing transonic flow in gas dynamics. Here, we succeeded in computing of the nonlinear Gröbner basis by hand, and in that way generated the cubic nonlinear difference scheme which possesses some attractive properties. These properties as well as those of the schemes generated for Burgers equation are illustrated by some numerical experiments in Section 8. We conclude in Section 9.

2 Basic idea

It is well-known [4, 5, 6, 8] that a rather wide class of scalar PDE and some systems of PDEs can be written in the conservation law form

$$\frac{\partial \boldsymbol{v}}{\partial x} + \frac{\partial}{\partial y}\boldsymbol{F}(\boldsymbol{v}) = 0, \tag{1}$$

where $\boldsymbol{v}$ is a m-vector function in the unknown n-vector function $\boldsymbol{u}$ and its partial derivatives $\boldsymbol{u}_x, \boldsymbol{u}_y, \boldsymbol{u}_{xx}, \ldots$. The vector function $\boldsymbol{F}$ maps R^m into R^m.

By Green's theorem (curl theorem in the plane), vector PDE (1) is equivalent to the integral relation

$$\oint_\Gamma -\boldsymbol{F}(\boldsymbol{v})dx + \boldsymbol{v}dy = 0, \tag{2}$$

where Γ is an arbitrary closed contour. Approximation of (2) rather than of (1) on a difference grid (balance or integro-interpolation method) is a natural way to generate conservative finite-difference schemes for PDEs of order two and higher.

Throughout this paper we shall consider orthogonal and uniform grids with the grid mesh steps h_1 and h_2

$$x_{j+1} - x_j = h_1, \qquad y_{k+1} - y_k = h_2 \tag{3}$$

and denote the grid values of the vector function $\boldsymbol{u}(x, y)$ and all its partial derivatives occurring in (2) by

$$\begin{array}{ll} \boldsymbol{u}(x,y) \Longrightarrow \boldsymbol{u}(x_j,y_k) \equiv \boldsymbol{u}_{jk}, & \boldsymbol{u}_x(x,y) \Longrightarrow \boldsymbol{u}_x(x_j,y_k) \equiv (\boldsymbol{u}_x)_{jk}, \\ \boldsymbol{u}_y(x,y) \Longrightarrow \boldsymbol{u}_y(x_j,y_k) \equiv (\boldsymbol{u}_y)_{jk}, & \boldsymbol{u}_{xx}(x,y) \Longrightarrow \boldsymbol{u}_{xx}(x_j,y_k) \equiv (\boldsymbol{u}_{xx})_{jk}, \\ \ldots\ldots\ldots\ldots\ldots\ldots & \end{array} \tag{4}$$

Choose the integration contour as follows (see Fig. 1) and add all the related integral relations between the dependent vector variable and its partial derivatives:

$$\begin{array}{ll} \displaystyle\int_{x_j}^{x_{j+2}} \boldsymbol{u}_x dx = \boldsymbol{u}(x_{j+2}, y) - \boldsymbol{u}(x_j, y), & \displaystyle\int_{y_k}^{y_{k+2}} \boldsymbol{u}_y dy = \boldsymbol{u}(x, y_{k+2}) - \boldsymbol{u}(x, y_k), \\ \displaystyle\int_{x_j}^{x_{j+2}} \boldsymbol{u}_{xx} dx = \boldsymbol{u}_x(x_{j+2}, y) - \boldsymbol{u}_x(x_j, y), & \displaystyle\int_{y_k}^{y_{k+2}} \boldsymbol{u}_{xy} dy = \boldsymbol{u}_x(x, y_{k+2}) - \boldsymbol{u}_x(x, y_k), \\ \ldots\ldots\ldots\ldots\ldots\ldots & \end{array} \tag{5}$$

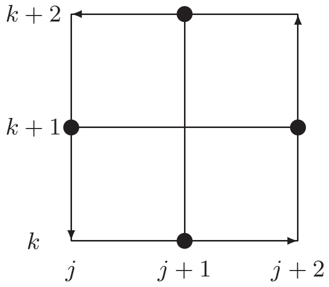

Figure 1. Integration contour on grid.

To obtain a source system of discrete equations for constructing of a difference scheme, we consider numerical approximations of the integral equations (2) for the contour of Fig. 1 and of the relations (5) in terms of the grid unknowns (4). Although generally one can use different numerical approximations for the integral equations in (2) and (5), we apply here for all these equations the simplest rectangle (midpoint) rule

$$\begin{array}{l} \boldsymbol{F}(\boldsymbol{v})_{j+1\,k+2} - \boldsymbol{F}(\boldsymbol{v})_{j+1\,k} + (\boldsymbol{v}_{j+2\,k+1} - \boldsymbol{v}_{j\,k+1}) = 0, \\ (\boldsymbol{u}_x)_{j+1\,k} \cdot 2h_1 = \boldsymbol{u}_{j+2\,k} - \boldsymbol{u}_{j\,k}, \qquad (\boldsymbol{u}_y)_{j\,k+1} \cdot 2h_2 = \boldsymbol{u}_{j\,k+2} - \boldsymbol{u}_{j\,k}, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \end{array} \tag{6}$$

Thereby, we obtained system (6) of difference equations in grid unknowns (4). In doing so, the number of scalar equations added to the (vector) integral equation (2) corresponds to the number of proper partial derivatives of order less than or equal to the order of partial derivatives involved in the integrand of (2).

It follows that eliminating from (6) all the proper grid partial derivatives gives equations containing only independent (vector) function $\boldsymbol{u}$, and, hence composing a finite-difference scheme. If system (6) is linear, then this difference elimination can always be algorithmically achieved by the Gröbner bases method considered in the next section.

It should be noted that generation of finite-difference schemes on grid (3) by the elimination can be also applied to PDEs irrelative to their conservation law properties. Again, one has to add to the initial differential equations, written in terms of grid variables (4), the corresponding number of integral relations (5) and approximate them by numerical quadrature formulas. Such an approach may give more flexibility in generation of distinct difference schemes, and we apply it in Section 4 to the first-order advection equation, and in Section 5 to Burgers equation. Generally, however, for the second-(and higher-) order PDEs admitting the integral conservation law form, the difference scheme obtained directly from the differential form may not be conservative. Besides, the difference elimination based on the integral form is usually more efficient than that based on the differential form. This is because the number of partial derivatives to be eliminated in the former case is smaller than in the latter case. Indeed, the integrand in (2) has the differential order smaller by one than that in (1) whereas computational complexity of the elimination is at least exponential in the number of eliminated variables [35].

3 Difference Gröbner bases

A *difference ring* R is a commutative ring with a unity together with a finite set of mutually commuting injective endomorphisms $\theta_1, \ldots, \theta_n$ of R. Similarly, one defines a *difference field.* Elements $\{y^1, \ldots, y^m\}$ in a difference ring containing R are said to be *difference indeterminates* over R if the set

$$\left\{\theta_1^{k_1} \cdots \theta_1^{k_1} y^j \mid \{k_1, \ldots, k_n\} \in \mathbb{Z}_{\geq 0}^n,\ 1 \leq j \leq m\right\}$$

is algebraically independent over R.

Hereafter we shall consider the ring of functions of n variables $x_1,\ldots,x_n$ with the basis endomorphisms $\theta_i \circ f(x_1,\ldots,x_n) = f(x_1,\ldots,x_i+1,\ldots,x_n)$. acting as shift operators.

The field $\mathbb{Q}(x_1,\ldots,x_n)$ of rational functions in $\{x_1,\ldots,x_n\}$ whose coefficients are rational numbers is an example of difference field, and we shall assume in the next sections that the coefficients of PDEs belong to this field.

Let $\mathbb{K}$ be a difference field, and $\mathbb{R} := \mathbb{K}\{y^1,\ldots,y^m\}$ be the difference ring of polynomials over $\mathbb{K}$ in variables $\{\theta^\mu \circ y^k \mid \mu \in \mathbb{Z}^n_{\geq 0},\ k=1,\ldots,m\}$. Hereafter, we denote by $\mathbb{R}_L$ the set of linear polynomials in $\mathbb{R}$ and use the notations:

$$
\begin{aligned}
&\Theta := \{\theta^\mu \mid \mu \in \mathbb{Z}^n_{\geq 0}\}, \qquad \deg_i(\theta^\mu \circ y^k) := \mu_i, \qquad \deg(\theta^\mu \circ y^k) := |\mu| := \sum_{i=1}^n \mu_i,\\
&\mathrm{lcm}(\mu,\nu) := \{\max\{\mu_1,\nu_1\},\ldots,\max\{\mu_n,\nu_n\}\}, \qquad \mathrm{lcm}(\theta^\mu \circ y^k, \theta^\nu \circ y^k) := \theta^{\mathrm{lcm}(\mu,\nu)} \circ y^k,\\
&\theta^\mu \circ y^k \sqsubset \theta^\nu \circ y^k \ \text{ when } \ \nu - \mu \in \mathbb{Z}^n_{\geq 0} \ \wedge \ |\nu - \mu| > 0. \qquad (7)
\end{aligned}
$$

A *difference ideal* is an ideal $I \subseteq \mathbb{R}$ closed under the action of any operator from Θ. If $F := \{f_1,\ldots,f_k\} \subset \mathbb{R}$ is a finite set, then the smallest difference ideal containing F will be denoted by $\mathrm{Id}(F)$. If for an ideal I there is $F \subset \mathbb{R}_L$ such that $I = \mathrm{Id}(F)$, then I is a *linear difference ideal.*

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

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

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

Given a ranking $\succ$, a linear polynomial $f \in \mathbb{R}_L \setminus \{0\}$ has the *leading term* of the form $a\,\vartheta \circ y^j$, $\vartheta \in \Theta$, 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. $\mathrm{lc}(f) := a \in \mathbb{K} \setminus \{0\}$ is the *leading coefficient* and $\mathrm{lm}(f) := \vartheta \circ y^j$ is the *leading (head) monomial.*

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

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

Given a nonzero linear difference ideal $I = \mathrm{Id}(G)$ and a ranking $\succ$, the ideal generating set $G = \{g_1,\ldots,g_s\} \subset \mathbb{R}_L$ is a *Gröbner basis* [22, 23] of I if

$$\forall\, f \in I \cap \mathbb{R}_L \setminus \{0\}, \quad \exists\, g \in G,\ \theta \in \Theta \ : \quad \mathrm{lm}(f) = \theta \circ \mathrm{lm}(g). \qquad (8)$$

It follows that the head monomial of $f \in I \setminus \{0\}$, as well as the polynomial f itself, is *reducible modulo* G and yields *the head reduction*:

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

If $f' \neq 0$, then its leading monomial is again reducible modulo G, and, by repeating the reduction finitely many times [16, 22, 23] we obtain $f \xrightarrow[G]{} 0$. Generally, if a polynomial $h \in \mathbb{R}_L$ contains a term with monomial u and coefficient $c \neq 0$ such that $u = \vartheta \circ \mathrm{lm}(f)$ for some $\vartheta \in \Theta$ and $f \in F \subset \mathbb{R}_L \setminus \{0\}$, then h can be reduced:

$$h \xrightarrow[g]{} h' := h - c\,\theta \circ (f/\,\mathrm{lc}(f)). \qquad (9)$$

By applying the reduction finitely many times, one obtains a polynomial $\bar{h}$ which is either zero or such that all its (nonzero) terms have monomials *irreducible modulo the set* $F \subset \mathbb{R}_L$. 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 *reduced* if

$$\forall\, g \in G \;:\quad g = NF(g, G \setminus \{g\}).$$

In our algorithmic construction of reduced Gröbner bases we shall use a restricted set of reductions called *Janet-like* (cf. [33, 34]) and defined as follows.

For a finite set $F \subseteq \mathbb{R}_L$ and a ranking $\succ$, we partition every $\mathrm{lm}_k(F)$ into groups 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 group $[d_0, \ldots, d_i]_k$ is defined as

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

Now we characterize a monomial $u \in \mathrm{lm}_k(F)$ by the nonnegative integer λ_i:

$$\lambda_i(u, \mathrm{lm}_k(F)) := \max\{\deg_i(v) \mid u, v \in [d_0, \ldots, d_{i-1}]_k\} - \deg_i(u).$$

If $\lambda_i(u, \mathrm{lm}_k(F)) > 0$, 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 $DP(f, F)$ denotes the set of all difference powers for $f \in F$. Now we define the partition of the set Θ into two disjoint subsets

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

which is similar to the partition of monomials into nonmultiplicative and multiplicative ones in the involutive approach [27].

A finite basis G of $I = \mathrm{Id}(G)$ is called Janet-like [33, 34] if

$$\forall\, f \in I \cap \mathbb{R}_L \setminus \{0\},\ \exists\, g \in G, \theta \in \mathcal{J}(g, G) \;:\quad \mathrm{lm}(f) = \theta \circ \mathrm{lm}(g). \tag{10}$$

In full analogy with (9) a *$\mathcal{J}$-reduction* is defined as

$$h \underset{g}{\to} h' := h - c\,\theta \circ (f/\,\mathrm{lc}(f)), \qquad \theta \in \mathcal{J}(f, F), \tag{11}$$

for polynomial $h \in \mathbb{R}_L$ containing monomial u with coefficient $c \neq 0$ satisfying $u = \vartheta \circ \mathrm{lm}(f)$ for some $f \in F \subset \mathbb{R}_L \setminus \{0\}$ and $\vartheta \in \mathcal{J}(f, F)$.

Apparently, any element in the ideal $I = \mathrm{Id}(G)$ is *$\mathcal{J}$-head reduced* to zero by the finite sequence of $\mathcal{J}$-head reductions by elements $g \in G$ in the Janet-like basis G:

$$f \underset{g}{\to} f' := f - \mathrm{lc}(f)\,\theta \circ (g/\,\mathrm{lc}(g)), \qquad \theta \in \mathcal{J}(g, G), \qquad f' \in I. \tag{12}$$

If the leading monomial of $p \in \mathbb{R} \setminus \{0\}$ is not $\mathcal{J}$-reducible modulo a finite subset $F \subset \mathbb{R} \setminus \{0\}$ we say that p is in the $\mathcal{J}$-head normal form modulo F and write $p = HNF_{\mathcal{J}}(p, F)$. If none of monomials in p is $\mathcal{J}$-reducible modulo F we say that p is in *the (full) normal form modulo F* and write $p = NF_{\mathcal{J}}(p, F)$.

Since $\mathcal{J}$-reducibility implies the Gröbner reducibility (9), a Janet-like basis satisfying (10) is a Gröbner basis. The converse is generally not true, that is, not every Gröbner basis is Janet-like.

The properties of a Janet-like basis are very similar to those of a Janet basis [36], but the former is generally more compact than the latter. For all that we consider hereafter only minimal bases.

Let GB be a reduced Gröbner basis, satisfying [23]:

$$\forall\, g \in GB \ : \quad g = NF(g, GB \setminus \{g\}). \tag{13}$$

Let now JB be a Janet basis, and JLB be a Janet-like basis of the same ideal and for the same ranking. Then for their cardinalities the inclusion $\mathrm{Card}(GB) \leq \mathrm{Card}(JLB) \leq \mathrm{Card}(JB)$ holds [27, 33]. Here 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 [15, 23], the algorithmic characterization of a Janet-like basis G has the following form (cf. [33]):

$$\forall\, g \in G,\ \vartheta \in DP(g,G) : \quad NF_{\mathcal{J}}(\vartheta \circ g, G) = 0. \tag{14}$$

These conditions are at the root of the algorithmic construction of Janet-like bases as described in the following section.

4 Algorithm

In this section we present an algorithm for constructing a reduced Gröbner basis (8) of the ideal generated by an input set of linear difference polynomials. The algorithm is an improved version of the algorithm in paper [34] and translates to the difference form of the polynomial involutive algorithm [27] modified for the Janet-like reductions.

To apply the difference form of criteria to avoid some unnecessary reductions we need the following definition.

An *ancestor* of a difference polynomial $f \in F \subset \mathbb{R}_L \setminus \{0\}$ is a polynomial $g \in F$ of the smallest $\deg(\mathrm{lm}(g))$ among those satisfying $f = \theta \circ g$ modulo $\mathrm{Id}(F \setminus \{f\})$ with $\theta \in \Theta$. If for all that

$$\deg(\mathrm{lm}(g)) < \deg(\mathrm{lm}(f)),$$

then the ancestor g of f is called *proper*.

If an intermediate polynomial h that arose in the course of the below algorithm has a proper ancestor g in the intermediate basis G, then h has been obtained from g via a sequence of shift operations of the form $\vartheta \circ g$ where $\vartheta \in DP(g,G)$ with $\mathrm{lm}(\vartheta \circ g)$ $\mathcal{J}$-irreducible modulo G. For the ancestor g itself the equality $\mathrm{lm}(\mathrm{anc}(g)) = \mathrm{lm}(g)$ holds.

In the main algorithm **GröbnerBasis** and its subalgorithms we endow every element $f \in G$ in the intermediate set of difference polynomials G (occuring in the set T) with a triple structure of the form:

$$p = \{f,\ g,\ dpow\}\,,$$

where

$\mathrm{pol}(p) := f$ is the polynomial f itself,

$\mathrm{anc}(p) := g$ is an ancestor of f in G,

$\mathrm{dp}(p) := dpow$ is a (possibly empty) subset of $DP(f,G)$.

The set *dpow* associated with the polynomial f accumulates all the difference powers for f which have been already applied to f in the course of the algorithm. Keeping this information serves to avoid useless repeated applications of the difference power operators. Knowledge of ancestors for elements in the intermediate basis helps to avoid some unnecessary reductions by applying Buchberger's chain criterion [23] adapted to Janet-like reductions.

Algorithm: GröbnerBasis$(F, \prec)$

Input: $F \in \mathbb{R}_L \setminus \{0\}$, a finite set; $\prec$, a ranking
Output: G, a reduced Gröbner basis of $\mathrm{Id}(F)$
1: **choose** $f \in F$ with the lowest $\mathrm{lm}(f)$ w.r.t. $\succ$
2: $T := \{f, f, \varnothing\}$
3: $Q := \{\, \{q, q, \varnothing\} \mid q \in F \setminus \{f\}\,\}$
4: $Q :=$**HeadReduce**$(Q, T, \succ)$
5: **while** $Q \neq \varnothing$ **do**
6:    **choose** $p \in Q$ with the lowest $\mathrm{lm}(\mathrm{pol}(p))$ w.r.t. $\succ$
7:    $Q := Q \setminus \{p\}$
8:    **if** $\mathrm{pol}(p) = \mathrm{anc}(p)$ **then**
9:       **for all** $\{\, q \in T \mid \mathrm{lm}(\mathrm{pol}(q)) = \theta^{\mu} \circ \mathrm{lm}(\mathrm{pol}(p)),\ |\mu| > 0 \,\}$ **do**
10:          $Q := Q \cup \{q\}; \quad T := T \setminus \{q\}$
11:       **od**
12:    **fi**
13:    $h :=$**TailNormalForm**$(p, T, \prec)$
14:    $T := T \cup \{h, \mathrm{anc}(p), \mathrm{dp}(p)\}$
15:    **for all** $q \in T$ **and** $\vartheta \in DP(q, T) \setminus \mathrm{dp}(q)$ **do**
16:       $Q := Q \cup \{\{\vartheta \circ \mathrm{pol}(q), \mathrm{anc}(q), \varnothing\}\}$
17:       $\mathrm{dp}(q) := \mathrm{dp}(q) \cap DP(q, T) \cup \{\vartheta\}$
18:    **od**
19:    $Q :=$**HeadReduce**$(Q, T, \prec)$
20: **od**
21: **return** $\{\mathrm{pol}(f) \mid f \in T\}$ or $\{\mathrm{pol}(f) \mid f \in T \mid f = \mathrm{anc}(f)\}$

In the above main algorithm **GröbnerBasis** and its subalgorithms presented below, where no confusion can arise, we simply refer to the triple set T as the second argument in DP, $NF_{\mathcal{J}}$, and $HNF_{\mathcal{J}}$ instead of the polynomial set $\{g = \mathrm{pol}(t) \mid t \in T\}$. Sometimes we also refer to the triple p instead of $\mathrm{pol}(p)$. Besides, when we speak of reduction of the triple set Q modulo triple set T we mean reduction of the polynomial set

$$\{f = \mathrm{pol}(q) \mid q \in Q\}$$

modulo

$$\{g = \mathrm{pol}(t) \mid t \in T\}.$$

Correctness and termination of algorithm **GröbnerBasis** can be shown exactly as in the polynomial case [27, 33]. Here we only elucidate some related features of the algorithm.

At steps 4 and 19 the $\mathcal{J}$-head reduction is performed for the difference polynomials in Q modulo those in T. Then the remaining tail reduction is done in line 13 to obtain the (full) $\mathcal{J}$-normal form. Thereby, the main **while-**loop 5–20 terminates when the conditions (14) hold for the difference polynomial set G composed from the first elements of triples in T

$$G := \{\mathrm{pol}(g) \mid g \in T\}, \tag{15}$$

and the set Q is empty. The upper **for**-loop 9–11 provides minimality of the output Janet-like basis contained in T [27]. Another **for**-loop 15–18 constructs new conditions (14) which have to be further examined because of the insertion of a new element in T at step 14. Besides, the set of difference powers is upgraded at step 17.

Furthermore, the main algorithm **GröbnerBasis** together with its subalgorithms presented below ensures that every element in the output Janet-like basis composed from the first elements in the triple set T has one and only one ancestor. This ancestor is apparently irreducible, in the Gröbner sense (9), by other elements in the basis. Thereby, those elements in the output basis that have no proper ancestors constitute the reduced Gröbner basis (13) that is returned by the main algorithm at the last step 21.

The algorithm **HeadReduce** invoked in lines 4 and 19 of the main algorithm returns the set Q which, if nonempty, contains part of the intermediate basis $\mathcal{J}$-head reduced modulo T. The reductions are performed by its subalgorithm **HeadNormalForm** that is invoked at step 6 of the algorithm.

If algorithm **HeadNormalForm** returns $h \neq 0$, then $\mathrm{lm}(h)$ does not belong to the initial ideal generated by $\{\mathrm{lm}(\mathrm{pol}(f)) \mid f \in Q \cup T\}$ [27, 36]. In this case the triple $\{h, h, \varnothing\}$ for h is inserted (line 9) into the output set Q. Otherwise, the output set Q retains the triple p as it is in the input.

In the case when $h = 0$ and $\mathrm{pol}(p)$ has no proper ancestors that is checked at step 14, all the descendant triples for p, if any, are deleted from the intermediate set S at step 16. Such descendants cannot occur in T owing to the choice conditions at steps 1, 6 and to the displacement condition of step 9 in the main algorithm **GröbnerBasis**. Steps 14–18 serve for the memory saving and can be ignored if the memory restrictions are not very critical for a given problem. In this case all those descendants will be casted away by the criteria checked in the below algorithm **HeadNormalForm**.

Algorithm **HeadNormalForm** performs verification (step 3) of $\mathcal{J}$-head reducibility of the input polynomial h modulo the polynomial set (15). This verification consists in searching a difference polynomial (reductor) in the set G defined in (15) such that G yields the reduction (11). If the search fails, that is, there is no $\mathcal{J}$-reductor, the algorithm returns at step 4 the input polynomial.

For the $\mathcal{J}$-head reducible input polynomial $\mathrm{pol}(p)$ that is checked at step 3 of algorithm **HeadNormalForm**, the following three criteria are verified at step 9

$$Criteria(p, g) = C_1(p, g) \vee C_2(p, g) \vee C_3(p, g)\,, \tag{16}$$

where

$$\begin{aligned}
&C_1(p, g) \text{ is true} \iff \mathrm{lcm}(\mathrm{lm}(\mathrm{anc}(p)), \mathrm{lm}(\mathrm{anc}(g))) \sqsubset \mathrm{lm}(\mathrm{pol}(p)),\\
&C_2(p, g) \text{ is true} \iff \exists\, t \in T \text{ such that}\\
&\qquad \mathrm{lcm}(\mathrm{lm}(\mathrm{pol}(t)), \mathrm{lm}(\mathrm{anc}(p))) \sqsubset \mathrm{lcm}(\mathrm{lm}(\mathrm{anc}(p)), \mathrm{lm}(\mathrm{anc}(g))) \wedge\\
&\qquad \mathrm{lcm}(\mathrm{lm}(\mathrm{pol}(t)), \mathrm{lm}(\mathrm{anc}(g))) \sqsubset \mathrm{lcm}(\mathrm{lm}(\mathrm{anc}(p)), \mathrm{lm}(\mathrm{anc}(g))),\\
&C_3(p, g) \text{ is true} \iff \exists\, t \in T \wedge y \in NM_{\mathcal{L}}(t, T) \text{ with } \mathrm{lm}(\mathrm{pol}(t)) \cdot y = \mathrm{lm}(\mathrm{pol}(p)),\\
&\qquad \mathrm{lcm}(\mathrm{lm}(\mathrm{anc}(p)), \mathrm{lm}(\mathrm{anc}(t))) \sqsubset \mathrm{lm}(\mathrm{pol}(p)) \wedge \mathrm{idx}(t, T) < \mathrm{idx}(f, T),\\
&\qquad \text{where } \mathrm{idx}(t, T) \text{ enumerates the position of triple } t \text{ in set } T.
\end{aligned}$$

In aggregate, criteria (16) translate (cf. [27, 37]) Buchberger's chain criterion [23] into the linear difference algebra.

In addition, if all difference polynomials in the input set F for the main algorithm **Gröbner-Basis** have constant coefficients, then the set of criteria (16) can be enlarged with one more criterion C_4:

$$C_4(p, g) \text{ is true for } \mathrm{lm}(\mathrm{pol}(p)) = \theta \circ y^k,\ \mathrm{lm}(\mathrm{pol}(g)) = \vartheta \circ y^k \iff \mathrm{lcm}(\theta, \vartheta) = \theta\, \vartheta\,.$$

Criterion C_4 is the difference form (cf. [27]) of Buchberger's co-prime criterion [23].

Algorithm: HeadReduce$(Q, T, \prec)$

Input: Q and T, sets of triples; $\prec$, a ranking
Output: $\mathcal{J}$-head reduced set Q modulo T

```
S := Q
Q := ∅
while S ≠ ∅ do
   choose p ∈ S
   S := S \ {p}
   h := HeadNormalForm(p, T)
   if h ≠ 0 then
      if lm(pol(p)) ≠ lm(h) then
         Q := Q ∪ {h, h, ∅}
      else
         Q := Q ∪ {p}
      fi
   else
      if lm(pol(p)) = lm(anc(p)) then
         for all {q ∈ S | anc(q) = pol(p)} do
            S := S \ {q}
         od
      fi
   fi
od
return Q
```

Algorithm: HeadNormalForm$(p, T, \prec)$

Input: T, a set of triples; p, a triple; $\prec$, a ranking
Output: $h = HNF_{\mathcal{J}}(p, T)$, the $\mathcal{J}$-head normal form of $\mathrm{pol}(p)$ modulo T

```
h := pol(p)
G := { pol(g) | g ∈ T }
if lm(h) is J-irreducible modulo G then
   return h
else
   take g ∈ T s.t. lm(h) is J-reducible modulo pol(g)
   if lm(h) ≠ lm(anc(p)) then
      if pol(p) = θ ∘ pol(f) with f ∈ T, θ ∈ DP(f, T) then
         if Criteria(p, g) then
            return 0
         fi
      fi
   else
      while h ≠ 0 and ∃g ∈ T s.t. lm(h) is J-reducible by q := pol(g) do
         h := h − lc(h) θ ∘ (q/ lc(q)) with θ ∈ J(q, T) and lm(h) = θ ∘ lm(g)
      od
   fi
fi  return h
```

If all the criteria are false, then the $\mathcal{J}$-head reduction of h is done by the **while-**loop 14–16 in accordance with the definition of the head reduction in (12).

The last algorithm **TailNormalForm** completes $\mathcal{J}$-reduction of the $\mathcal{J}$-head reduced polynomial in the input triple by performing its $\mathcal{J}$-tail reduction. This algorithm is invoked at step 13 of the main algorithm **GröbnerBasis**. The tail reduction is performed in the **while-**loop as a sequence of elementary reductions (11).

Algorithm: TailNormalForm$(p, T, \prec)$

Input: p, a triple such that $\mathrm{pol}(p) = HNF_{\mathcal{J}}(p, T)$; T, a set of triples; $\prec$, a ranking
Output: $h = NF_{\mathcal{J}}(p, T)$, the (full) $\mathcal{J}$-normal form of $\mathrm{pol}(p)$ modulo T

$G := \{\, \mathrm{pol}(g) \mid g \in T \,\}$
$h := \mathrm{pol}(p)$
while h has a term $t = a\vartheta \circ y^j$ which is $\mathcal{J}$-reducible modulo G **do**
  take $g \in G$ s.t. $\vartheta \circ y^j = \theta \circ lm(g)$
  $h := h - \mathrm{lc}(h)\, \vartheta \circ (g/\,\mathrm{lc}(g))$
od
return h

Because of the lack of an appropriate collection of benchmarks for linear finite-difference polynomial systems, the algorithmic efficiency of algorithm **GröbnerBasis** can be indirectly analyzed by running its polynomial (non-difference) counterpart [27, 33] for the extensive benchmarks collection in [38, 39]. Some timings for our polynomial implementation can be found on the Web page [28].

Recently, the algorithm in its difference version was implemented in Maple [29]. Just this implementation was used for generation of linear finite-difference schemes as described in the next sections. Though one needs special and intensive benchmarking for linear difference systems, our first experimenting with the Maple implementation and with that for commutative polynomials gives us a good reason to expect that the following merits revealed for the pure polynomial version [27] hold also for the difference one:

- automatic avoidance of some useless reductions;
- weakened role of the criteria: even without applying any criteria the algorithm is reasonably fast;
- smoothed growth of intermediate coefficients;
- fast search of a reductor which provides the elementary Janet-like reduction (11) of a given term. It should be noted that there can be at most one reductor [27];
- natural and effective parallelism.

5 Illustrative examples of PDEs

5.1 Laplace equation

In this section we illustrate the approach of Section 2 to the automatic generation of difference schemes by simplest elliptic, parabolic and hyperbolic equations. To compute Gröbner bases providing the elimination of the partial derivatives to construct difference schemes we used the Maple package [29] implementing the algorithms described in the previous section.

We start with the Laplace equation [3, 4, 5, 6, 7]

$$u_{xx} + u_{yy} = 0 \tag{17}$$

and rewrite it as the conservation law (1)

$$\oint_{\Gamma} -u_y dx + u_x dy = 0. \tag{18}$$

Now we add the relations (5) for the partial derivatives u_x and u_y

$$\int_{x_j}^{x_{j+2}} u_x dx = u(x_{j+2}, y) - u(x_j, y), \qquad \int_{y_k}^{y_{k+2}} u_y dy = u(x, y_{k+2}) - u(x, y_k). \tag{19}$$

Thus, we obtain the system of three integral relations (18), (19) for three functions

$$u(x, y), \quad u_x(x, y), \quad u_y(x, y).$$

To discretize this system we choose the rectangular contour of Fig. 1 on the orthogonal and uniform grid (3) with

$$h_1 = h_2 = h \tag{20}$$

and use the midpoint integration method for both (18) and (19). This yields the system:

$$\begin{aligned}
&-((u_y)_{j+1\,k} - (u_y)_{j+1\,k+2}) + ((u_x)_{j+2\,k+1} - (u_y)_{j\,k+1}) = 0,\\
&(u_x)_{j+1\,k} \cdot 2\,h = u_{j+2\,k} - u_{j\,k},\\
&(u_y)_{j\,k+1} \cdot 2\,h = u_{j\,k+2} - u_{j\,k}.
\end{aligned}$$

Rewritten in terms of difference polynomials in the ring $\mathbb{Q}\{u, u_x, u_y\}$ (see Section 2) it reads:

$$\begin{aligned}
&(\theta_x\theta_y^2 - \theta_x) \circ u_y + (\theta_x^2\theta_y - \theta_y) \circ u_x = 0,\\
&2\,h\,\theta_x \circ u_x - (\theta_x^2 - 1) \circ u = 0,\\
&2\,h\,\theta_y \circ u_y - (\theta_y^2 - 1) \circ u = 0.
\end{aligned}$$

Computation of the Gröbner basis for the elimination ranking (Section 2) with $u_x \succ u_y \succ u$ and $\theta_x \succ \theta_y$ gives:

$$\begin{aligned}
&\theta_x \circ u_x - \frac{1}{2\,h}(\theta_x^2 - 1) \circ u = 0,\\
&\theta_y \circ u_x + \theta_x \circ u_y - \frac{1}{2\,h}(\theta_x\theta_y((\theta_x^2 - 1) + (\theta_y^2 - 1))) \circ u = 0,\\
&\theta_x^2 \circ u_y - \frac{1}{2\,h}(\theta_x^2\theta_y((\theta_x^2 - 1) + (\theta_y^2 - 1)) - \theta_y(\theta_x^2 - 1)) \circ u = 0,\\
&\theta_y \circ u_y - \frac{1}{2\,h}(\theta_y^2 - 1) \circ u = 0,\\
&\frac{1}{2\,h}(\theta_x^4\theta_y^2 + \theta_x^2\theta_y^4 - 4\theta_x^2\theta_y^2 + \theta_x^2 + \theta_y^2) \circ u = 0.
\end{aligned}$$

The latter equation with eliminated u_x and u_y is the standard difference scheme with the central approximation of the second-order derivatives in (17) written in double nodes

$$\frac{u_{j+2\,k} - 2\,u_{j\,k} + u_{j-2\,k}}{4h^2} + \frac{u_{j\,k+2} - 2\,u_{j\,k} + u_{j\,k-2}}{4h^2} = 0.$$

Similarly, the trapezoidal integration rule for relations (19) generates the same difference scheme but written in ordinary nodes

$$\frac{u_{j+1\,k} - 2\,u_{j\,k} + u_{j-1\,k}}{h^2} + \frac{u_{j\,k+1} - 2\,u_{j\,k} + u_{j\,k-1}}{h^2} = 0.$$

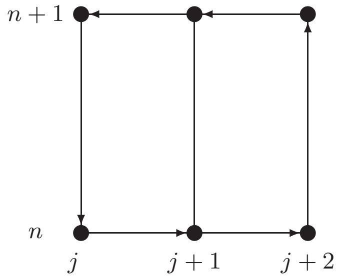

Figure 2. Integration contour for heat equation.

5.2 Heat equation

Consider now the heat equation [3, 4, 5, 6, 7, 8]

$$u_t + \alpha u_{xx} = 0$$

in its conservation law form

$$\oint_\Gamma -\alpha u_x dt + u dx = 0. \tag{21}$$

The integrand in (21) contains the only partial derivative u_x. Hence we add the single integral relation

$$\int_{x_j}^{x_{j+1}} u_x dx = u(x_{j+1}, t) - u(x_j, t). \tag{22}$$

Again we discretize $u(x,t)$ and $u_x(x,t)$ on the orthogonal and uniform grid with the spatial mesh step h and the temporal mesh step τ, and choose the contour shown in Fig. 2. Then, applying the midpoint rule for the contour integral and the trapezoidal rule for the relation integral we find two difference equations for two indeterminates u, u_x in the form

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

By elimination of u_x by means of the Gröbner basis with $u_x \succ u$ we obtain the famous Crank–Nicholson scheme [3, 4, 5, 6, 7, 8, 9]

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

The same scheme is obtained for the midpoint integration method applied to (22).

5.3 Wave equation

The wave equation [3, 4, 5, 6, 7, 8]

$$u_{tt} - u_{xx} = 0$$

in the conservation law form is given by

$$\oint_\Gamma u_x dt + u_t dx = 0.$$

Choosing the same grid with the mesh steps (20), the contour of Fig. 1 and integral relations (19) as are used in Section 5.2 for the Laplace equation (18) and applying the midpoint rule for the contour integral and the trapezoidal rule for the integral relations we obtain the operator equations

$$
\begin{aligned}
&(\theta_x - \theta_x\theta_t^2) \circ u_t + (\theta_x^2\theta_t - \theta_t) \circ u_x = 0,\\
&\frac{h}{2}\,(\theta_x + 1) \circ u_x - (\theta_x - 1) \circ u = 0,\\
&\frac{h}{2}\,(\theta_t + 1) \circ u_t - (\theta_t - 1) \circ u = 0.
\end{aligned}
$$

The Gröbner basis method yields the standard difference scheme

$$
u_j^{n+1} + u_j^{n-1} - u_{j+1}^n - u_{j-1}^n = 0.
$$

5.4 Advection equation

Consider now a simple form of the Advection (or convection or one-way wave) equation [3, 4, 5, 6, 7, 8, 9]

$$
u_t + \nu\, u_x = 0, \qquad \nu = \text{const.} \tag{23}
$$

Being of first order, the equation (23) has already the conservation law form (1). By this reason, to generate a difference scheme we shall not convert the equation into the integral form (2). In the latter case one has nothing to eliminate. Instead, we consider equation (23) together with the integral relations:

$$
\begin{aligned}
&u_t + \nu\, u_x = 0,\\
&\int_{t_1}^{t_2} u_t dt = u(t_2, x) - u(t_1, x), \qquad \int_{x_1}^{x_2} u_x dx = u(t, x_2) - u(t, x_1).
\end{aligned} \tag{24}
$$

Discretization of u, u_t and u_x on the orthogonal and uniform grid with the mesh steps h and τ and the explicit integration formula for the upper integral relation in (24) together with the midpoint integration rule for the lower relation give the difference system:

$$
u_t + \nu\, u_x = 0, \qquad \tau\, u_t = (\theta_t - 1) \circ u, \qquad 2\,h\,\theta_x \circ u_x = (\theta_x^2 - 1) \circ u. \tag{25}
$$

Let us apply the operator θ_x to both sides of the middle equation in (25) and then use the Lax method, that is, replace θ_x with $(\theta_x^2 + 1)/2$ in the second term of the right-hand side. This replacement yields

$$
\begin{aligned}
&u_t + \nu\, u_x = 0,\\
&\theta_x \circ u_t \cdot \tau - \left(\theta_t\theta_x - \frac{\theta_x^2 + 1}{2}\right) \circ u = 0,\\
&2\,\theta_x \circ u_x \cdot h - (\theta_x^2 - 1) \circ u = 0.
\end{aligned}
$$

The lexicographical Gröbner basis for the elimination ranking $u_t \succ u_x \succ u$ with $\theta_t \succ \theta_x$, is

$$
\begin{aligned}
&u_t + \nu\, u_x = 0,\\
&2h\,\theta_x \circ u_x - (\theta_x^2 - 1) \circ u = 0,\\
&(2h\,\theta_x\theta_t - h\,(\theta_x^2 + 1) + \tau(\theta_x^2 - 1) \cdot \nu) \circ u = 0.
\end{aligned}
$$

Its last element gives the scheme:

$$
u_{j+1}^{n+1} = \frac{u_{j+2}^n + u_j^n}{2} - \frac{\nu\tau(u_{j+2}^n - u_j^n)}{2\,h}.
$$

6 Burgers equation

6.1 Conservation law form

Consider Burgers equation [4, 5, 8, 9] in the form

$$u_t + f_x = \nu\, u_{xx}, \qquad \nu = \text{const}, \tag{26}$$

where we replaced u^2 by the flux function f in order to avoid computation of nonlinear difference Gröbner bases. ν is called the viscosity. This equation exhibits some difficult features from the point of view of simple finite difference schemes due to the term $f = u^2$. Let us, first, convert equation (26) into the conservation law form

$$\oint_\Gamma (\nu u_x - f)dt + u dx = 0.$$

Then choose the contour of Fig. 1 and add the integral relation

$$\int_{x_j}^{x_{j+2}} u_x dx = u(x_{j+2},t) - u(x_j,t).$$

Denoting as above the temporal and spatial mesh steps by τ and h, and applying the midpoint integration rule we obtain the system:

$$\begin{aligned}
&h\,(\theta_x\theta_t^2 - \theta_x)\circ u - \tau\,(\theta_x^2\theta_t - \theta_t)\circ(\nu u_x - f) = 0,\\
&2\,h\,\theta_x \circ u_x - (\theta_x^2 - 1)\circ u = 0.
\end{aligned}$$

Its Gröbner basis form for the elimination order with $u_x \succ u \succ f$ and $\theta_t \succ \theta_x$ is given by

$$\begin{aligned}
&2\,\nu\tau h\,\theta_t \circ u_x + 2\,h^2\theta_x(\theta_t^2 - 1)\circ u + 2\,\tau h\theta_t(\theta_x^2 - 1)\circ f - \nu\tau\theta_t\theta_x(\theta_x^2 - 1)\circ u = 0,\\
&2\,h\,\theta_x \circ u_x - (\theta_x^2 - 1)\circ u = 0,\\
&2\,h^2\theta_x^2(\theta_t^2 - 1)\circ u - \nu\tau\theta_t(\theta_x^4 - 2\,\theta_x^2 + 1)\circ u + 2\,\tau h\,\theta_t\theta_x(\theta_x^2 - 1)\circ f = 0.
\end{aligned}$$

The obtained difference scheme

$$\frac{u_{j+2}^{n+2} - u_{j+2}^n}{\tau} - \nu\frac{u_{j+4}^{n+1} - 2u_{j+2}^{n+1} + u_j^{n+1}}{2h^2} + \frac{f_{j+3}^{n+1} - f_{j+1}^{n+1}}{h} = 0. \tag{27}$$

is the standard explicit scheme with forward time and forward space differencing. It is well-known that schemes of this type are unstable [5, 8, 9]. Furthermore, by using implicit schemes one can provide the von Neumann stability[1]. However, all such schemes are usually not very satisfactory when one considers non-smooth or discontinuous solutions (shock waves) of Burgers equation.

6.2 Lax method

To exploit more flexibility and freedom in our difference elimination approach to generation of finite-difference schemes, we go back to the original differential equation (26) and consider it together with the integral relations providing the elimination. For discretization of the relations we combine the midpoint rule for integration over x with the explicit integration over t and apply the Lax method to the last integration:

$$u_t + f_x = \nu u_{xx}, \qquad\qquad (u_t)_j^n + (f_x)_j^n = \nu(u_{xx})_j^n,$$

[1] For example, if one uses the central differencing in the last term of (27), and the Crank–Nicolson approach to the second (diffusion) term.

$$
\begin{aligned}
&\int u_t dt = u, && u_t\tau = u_j^{n+1} - \frac{u_{j+2}^n + u_j^n}{2},\\
&\int f_x dx = f, \qquad \Longrightarrow && 2\,h(f_x)_{j+1}^n = f_{j+2}^n - f_j^n,\\
&\int u_x dx = u, && 2\,h(u_x)_{j+1}^n = u_{j+2}^n - u_j^n,\\
&\int u_{xx} dx = u_x, && 2\,h(u_{xx})_{j+1}^n = (u_x)_{j+2}^n - (u_x)_j^n.
\end{aligned}
\tag{28}
$$

The Gröbner basis based elimination with $u_{xx} \succ u_t \succ u_x \succ f_x \succ u \succ f$ yields the scheme

$$
\frac{2\,u_{j+2}^{n+1} - (u_{j+3}^n + u_{j+1}^n)}{2\tau} + \frac{f_{j+3}^n - f_{j+1}^n}{2h} - \nu\frac{u_{j+4}^n - 2\,u_{j+2}^n + u_j^n}{4h^2} = 0. \tag{29}
$$

One can also use the trapezoidal rule for the spatial integrations. This derives other schemes. Since there are three spatial integrals in (28), by selecting either the midpoint or the trapezoidal rule for these integrals, we obtain eight possible variants of the difference schemes. Our computation with the Gröbner bases reveals seven different schemes. Apart from (29) there are

$$
\frac{2(u_{j+2}^{n+1} + u_{j+1}^{n+1}) - (u_{j+3}^n + u_{j+2}^n + u_{j+1}^n + u_j^n)}{4\tau} + \frac{(f_{j+3}^n + f_{j+2}^n) - (f_{j+1}^n + f_j^n)}{4h}
= \nu\frac{(u_{j+3}^n - u_{j+2}^n) - (u_{j+1}^n - u_j^n)}{2h^2}, \tag{30}
$$

$$
\frac{2u_{j+1}^{n+1} - (u_{j+2}^n + u_j^n)}{2\tau} + \frac{f_{j+2}^n - f_j^n}{2h} = \nu\frac{u_{j+2}^n - 2u_{j+1}^n + u_j^n}{h^2}, \tag{31}
$$

$$
\frac{2(u_{j+3}^{n+1} + 2u_{j+2}^{n+1} + u_{j+1}^{n+1}) - (u_{j+4}^n + 2u_{j+3}^n + 2u_{j+2}^n + 2u_{j+1}^n + u_j^n)}{8\tau}
+ \frac{(f_{j+4}^n + 2f_{j+1}^n) - (2f_{j+1}^n + f_j^n)}{8h} = \nu\frac{u_{j+3}^n - 2u_{j+2}^n + u_{j+1}^n}{h^2}, \tag{32}
$$

$$
\frac{2(u_{j+3}^{n+1} + u_{j+2}^{n+1}) - (u_{j+4}^n + u_{j+3}^n + u_{j+2}^n + u_{j+1}^n)}{4\tau} + \frac{f_{j+3}^n - f_{j+2}^n}{h}
= \nu\frac{((u_{j+5}^n + u_{j+4}^n) - 2u_{j+3}^n) - (2u_{j+2}^n - (u_{j+1}^n + u_j^n))}{8h^2}, \tag{33}
$$

$$
\frac{2(u_{j+2}^{n+1} + u_{j+1}^{n+1}) - (u_{j+3}^n + u_{j+2}^n + u_{j+1}^n + u_j^n)}{4\tau} + \frac{f_{j+2}^n - f_{j+1}^n}{h}
= \nu\frac{(u_{j+3}^n - u_{j+2}^n) - (u_{j+1}^n - u_j^n)}{2h^2}, \tag{34}
$$

$$
\frac{2(u_{j+3}^{n+1} + 2u_{j+2}^{n+1} + u_{j+1}^{n+1}) - (u_{j+4}^n + 2u_{j+3}^n + 2u_{j+2}^n + 2u_{j+1}^n + u_j^n)}{8\tau} + \frac{f_{j+3}^n - f_{j+1}^n}{2h}
= \nu\frac{u_{j+3}^n - 2u_{j+2}^n + u_{j+1}^n}{h^2}. \tag{35}
$$

Just the scheme (34) is obtained twice in the course of generating eight schemes.

6.3 Two-step Lax–Wendroff method

Our Gröbner basis based technique can also be applied to generate other types of difference schemes. For example, one can generate two-step Lax–Wendroff schemes [41]. Let $\overline{u}$ and $\overline{f}$ denote the values of u and f on the intermediate time levels. Then, applying again the midpoint rule for the spatial integrals, gives the following difference system:

$$
{u_t}_j^n + {f_x}_j^n = \nu\; {u_{xx}}_j^n,
$$

$$
\begin{aligned}
&{u_t}_j^n\,\tau = \overline{u}_j^{\,n+1} - \frac{u_{j+2}^n + u_j^n}{2},\\
&2{f_x}_{j+1}^n\,h = f_{j+2}^n - f_j^n,\\
&2{u_x}_{j+1}^n\,h = u_{j+2}^n - u_j^n,\\
&2{u_{xx}}_{j+1}^n\,h = {u_x}_{j+2}^n - {u_x}_j^n,\\
&{\overline{u}_t}_j^n + {\overline{f}_x}_j^n = \nu\,{\overline{u}_{xx}}_j^n,\\
&{\overline{u}_t}_j^n\,\tau = u_j^{n+1} - u_j^n,\\
&2{\overline{f}_x}_{j+1}^n\,h = \overline{f}_{j+2}^n - \overline{f}_j^n,\\
&2{\overline{u}_x}_{j+1}^n\,h = \overline{u}_{j+2}^n - \overline{u}_j^n,\\
&2{\overline{u}_xx}_{j+1}^n\,h = {\overline{u}_x}_{j+2}^n - {\overline{u}_x}_j^n.
\end{aligned}
$$

For the elimination ranking with

$$
u_{xx} \succ \overline{u}_{xx} \succ u_x \succ \overline{u}_x \succ u_t \succ \overline{u}_t \succ f_x \succ \overline{f}_x \succ f \succ u \succ \overline{f} \succ \overline{u}
$$

the Gröbner basis contains the Lax–Wendroff scheme

$$
\begin{aligned}
&\frac{\overline{u}_{j+2}^{n+1} - (u_{j+3}^n + u_{j+1}^n)}{2\,\tau} + \frac{f_{j+3}^n - f_{j+1}^n}{2\,h} = \nu\frac{u_{j+4}^n - 2u_{j+2}^n + u_j^n}{4h^2},\\
&\frac{u_{j+3}^{n+1} - \overline{u}_{j+2}^n}{2\,\tau} + \frac{\overline{f}_{j+3}^n - \overline{f}_{j+1}^n}{2\,h} = \nu\frac{\overline{u}_{j+4}^n - \overline{u}_{j+2}^n + \overline{u}_j^n}{4\,h^2}.
\end{aligned} \tag{36}
$$

With all possible combinations of the trapezoidal and midpoint rules one obtains 49 different Lax–Wendroff schemes.

6.4 Differential approximation

To analyze properties of a difference scheme it can be useful to compute its differential approximation [40] that is often called the modified equation(s) of the difference scheme. There are whole classes of different schemes for which their stability properties can be obtained with the aid of the differential approximation [2]. For all that, in many cases, the computation can be easily done with modern computer algebra software. In our research we use Maple [17]. Consider, for example, the schemes (29)–(35).

Their differential approximation for $f = u^2$ and with collection of the coefficients at τ, h^2, h^2/τ is given by:

$$
\begin{aligned}
u_t + u_x u - \nu\,u_{xx} = {}& \left(-\frac{1}{2}\,\nu^2 u_{xxxx} + (u_{xxx}u + 2\,u_{xx}u_x)\nu - u_x^2 u - \frac{1}{2}\,u^2 u_{xx}\right)\tau\\
&+ (*)h^2 + \frac{1}{2}\,u_{xx}\frac{h^2}{\tau} + \cdots .
\end{aligned}
$$

The schemes (29)–(35) differ in the coefficient $(*)$ at h^2 only. These coefficients are as follows

$$
\begin{aligned}
&(32) \qquad -\frac{1}{6}\,u_{xxxx}\nu + \frac{1}{3}\,u_{xxx}u + \frac{1}{2}\,u_{xx}u_x,\\
&(33) \qquad -\frac{1}{6}\,u_{xxxx}\nu + \frac{1}{12}\,u_{xxx}u - \frac{1}{4}\,u_{xx}u_x,\\
&(34) \qquad -\frac{5}{12}\,u_{xxxx}\nu + \frac{1}{3}\,u_{xxx}u + \frac{1}{2}\,u_{xx}u_x,\\
&(35) \qquad -\frac{1}{6}\,u_{xxxx}\nu - \frac{1}{6}\,u_{xxx}u - u_{xx}u_x,
\end{aligned}
$$

(36) $\frac{1}{12}\,u_{xxxx}\nu + \frac{1}{3}\,u_{xxx}u + \frac{1}{2}\,u_{xx}u_x,$

(37) $-\frac{1}{6}\,u_{xxxx}\nu + \frac{1}{3}\,u_{xxx}u + \frac{1}{2}\,u_{xx}u_x,$

(38) $-\frac{1}{6}\,u_{xxxx}\nu - \frac{1}{4}\,u_{xx}u_x + \frac{1}{12}\,u_{xxx}u.$

Thereby, comparison of differential approximations for schemes (29)–(35) shows that

- all the schemes provide the same order of approximation in τ, h;
- they have identical linear numerical dissipation (viscosity) [4, 9] determined by $u_{xx}h^2/(2\,\tau)$;
- the schemes possess similar dispersion properties with distinction in the rational coefficients of the differential polynomial in u multiplied by h^2.

As to scheme (27), the right-hand side of its differential approximation reads

$$\left(-\frac{1}{2}\nu^2 u_{xxxx} + (u_{xxx}u + 2u_{xx}u_x)\nu - u_x^2 u - \frac{1}{2}u^2 u_{xx}\right)\tau \\ + \left(\frac{1}{3}u_{xxxx}\nu - \frac{1}{6}u_{xxx}u - \frac{1}{2}u_{xx}u_x\right)h^2 + \cdots .$$

This explicitly shows instability of the scheme which does not yield linear numerical viscosity.

We obtained also analogous results on stability and on close properties for the different Lax–Wendroff schemes of type (36) and its variations due to the choice of different numerical integration rules for the spatial integrals.

6.5 Godunov method

It is especially difficult to simulate numerically nonsmooth and discontinuous solutions which are among most interesting problems in computational fluid and gas dynamics [1, 4, 5, 8]. Most of the known difference schemes fail to handle these singularities. The most appropriate numerical approach to such problems was developed by Godunov [1, 42] and based on solving a local Riemann problem [4, 6] as a cornerstone of the Godunov's scheme generation. There are special numerical Riemann solvers, for example [43], designed for these purposes and for application to computational fluid dynamics.

Instead of the use of numerical Riemann solvers, we apply the Gröbner bases technique to generate the Godunov-type scheme for inviscid Burgers equation when $\nu = 0$ in (26). For this purpose we discretize the corresponding system in (28) in the following way

$$\begin{aligned}
&{u_t}_j^n + {f_x}_j^n = 0,\\
&{u_t}_j^n\,\tau = u_j^{n+1} - u_j^n,\\
&({f_x}_j^n\,h - (f_{j+1}^n - f_j^n))({f_x}_{j+1}^n\,h - (f_{j+1}^n - f_j^n)) = 0,\\
&2\,{u_x}_{j+1}^n\,h = u_{j+2}^n - u_j^n,\\
&2\,{u_{xx}}_{j+1}^n\,h = {u_x}_{j+2}^n - {u_x}_j^n.
\end{aligned} \qquad (37)$$

Here, the third equation contains in its left-hand side the product of two different solutions for the flux function f of the local Riemann problem [43]. Therefore, we add to the system composed of the original differential equation and discrete forms of the integral relations for partial derivatives u_t, u_x, u_{xx} the nonlinear difference equation on f and f_x containing solutions of the local Riemann problem.

Since the Riemann condition on the flux is now a constituent of the difference system, an elimination of all the partial derivatives of u and f gives the difference scheme consistent with that condition. To do the elimination from the nonlinear system (37) we apply the Gröbner factoring approach [44]: if a Gröbner basis contains a polynomial which factors, then the computation is split into the computation of two or more Gröbner bases corresponding to the factors. In doing so, we choose the elimination ranking

$$u_{xx} \succ u_x \succ u_t \succ f_x \succ f \succ u$$

and compute two Gröbner bases, for every factor in (37). Then we compose the product of two obtained difference polynomials in u and f that gives us the Godunov-type difference scheme:

$$\left(\frac{u_{j+2}^{n+1}-u_{j+2}^{n}}{\tau}+\frac{f_{j+2}^{n}-f_{j+1}^{n}}{h}\right)\cdot\left(\frac{u_{j+2}^{n+1}-u_{j+2}^{n}}{\tau}+\frac{f_{j+3}^{n}-f_{j+2}^{n}}{h}\right)=0. \tag{38}$$

Below (Section 7) we compare schemes (29), (36) and (38) by numerical simulation of a discontinuous solution.

7 Falkowich–Karman equation

Consider now the nonlinear two-dimensional Falkowich–Karman equation [45]

$$\varphi_{xx}(K-(\gamma+1)\varphi_x)+\varphi_{yy}=0 \tag{39}$$

describing transonic flow in gas dynamics in its non-stationary form

$$\varphi_{xx}(K-(\gamma+1)\varphi_x)+\varphi_{yy}-2\varphi_{xt}-\varphi_{tt}=0. \tag{40}$$

This form can be used to find a stationary solution by the steady-state method. We rewrite equation (40) into the conservation law form

$$\int_{t_n}^{t_{n+1}}\left(\oint_{\Gamma}-\varphi_y dx+\varphi_x\left(K-\frac{(\gamma+1)}{2}\varphi_x\right)dy\right)dt-\int_{x_j}^{x_{j+2}}\int_{y_k}^{y_{k+2}}(2\varphi_x+\varphi_t)\Big|_{t_n}^{t_{n+1}}dxdy=0.$$

Here we use a grid decomposition of the three-dimensional domain in (x,y,t) into elementary volumes. Fig. 3 shows an elementary volume.

Again we add the integral relations for partial derivatives with the use of the trapezoidal integration rule for φ_x, φ_y and the midpoint rule for φ_t.

Then we obtain the nonlinear operator equations:

$$\begin{aligned}
&\left(-(\theta_x-\theta_x\theta_y^2)\circ\varphi_y+(\theta_x^2\theta_y-\theta_y)\circ\left(\varphi_x\left(K-\frac{(\gamma+1)}{2}\,\varphi_x\right)\right)\right)\cdot 2\,h\,\tau\\
&\qquad-(\theta_t-1)(\theta_x^2\theta_y-\theta_y)\circ 2\,\varphi\cdot 2\,h-\theta_x\theta_y\circ\varphi_t\cdot 4\,h^2=0,\\
&(\theta_x+1)\circ\varphi_x\cdot\frac{h}{2}=(\theta_x-1)\circ\varphi,\\
&(\theta_y+1)\circ\varphi_y\cdot\frac{h}{2}=(\theta_y-1)\circ\varphi,\\
&\theta_t\circ\varphi_t\cdot 2\,\tau=(\theta_t^2-1)\circ\varphi.
\end{aligned} \tag{41}$$

Because of nonlinearity in the initial differential equation (40), the difference system obtained is also nonlinear. By this reason the Maple package [29] implementing algorithm **GröbnerBasis** (Section 4) is inapplicable to (41). Since there is no software available for computing difference Gröbner bases for nonlinear systems, we had to perform hand computations in accordance with

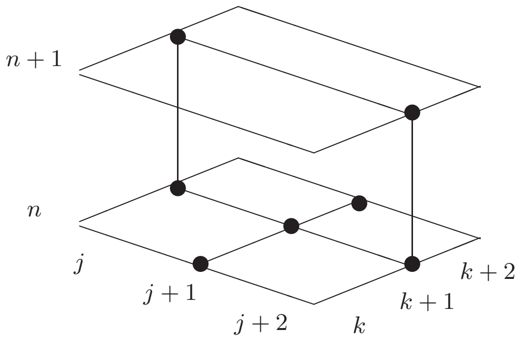

Figure 3. Cell for Falkowich–Karman equation.

the above described algorithm and with an assistance of Maple to check some intermediate results.

In these calculations we used the lexicographical ranking such that $\varphi_x \succ \varphi_y \succ \varphi_t \succ \varphi$ and $\theta_x \succ \theta_y \succ \theta_t$. The resulting Gröbner basis has the form:

$$
\begin{aligned}
&(\theta_x - 1)^2\theta_y \circ \varphi \cdot (\gamma+1)\varphi_x \frac{\tau}{h} + \theta_x\theta_y \circ \varphi_t \cdot h \\
&\quad = ((\theta_x - 1)^2\theta_y \circ \varphi \cdot (K - (\theta_x - 1)^2\theta_y \circ (\gamma+1)\varphi) + \theta_x(\theta_y - 1)^2 \circ \varphi)\frac{\tau}{h} \\
&\quad - (\theta_x^2\theta_y - \theta_y)(\theta_t - 1) \circ \varphi, \\
&(\theta_y + 1) \circ \varphi_y = (\theta_y - 1) \circ \varphi \cdot \frac{2}{h}, \\
&\theta_t \circ \varphi_t = (\theta_t^2 - 1) \circ \varphi \cdot \frac{1}{2\tau}, \\
&0 = \theta_x(\theta_x - 1)^2\theta_y\theta_t \circ \varphi \cdot \Big[\Big((\theta_x - 1)^2\theta_y\theta_t \circ \varphi \cdot \Big(K - (\theta_x^3 - \theta_x^2 + \theta_x - 1)\theta_y\theta_t \circ \frac{(\gamma+1)}{2}\varphi\Big) \\
&\quad + \theta_x(\theta_y - 1)^2\theta_t \circ \varphi\Big)\frac{\tau}{h} - (\theta_x^2 - 1)\theta_y(\theta_t^2 - \theta_t) \circ \varphi - \theta_x\theta_y(\theta_t - 1)^2 \circ \varphi \cdot \frac{h}{2\tau}\Big] \\
&\quad + (\theta_x - 1)^2\theta_y\theta_t \circ \varphi \cdot \Big[\Big(\theta_x(\theta_x - 1)^2\theta_y\theta_t \circ \varphi \cdot \Big(K - (\theta_x^3 - \theta_x^2 + \theta_x - 1)\theta_y\theta_t \circ \frac{(\gamma+1)}{2}\varphi\Big) \\
&\quad + \theta_x^2(\theta_y - 1)^2\theta_t \circ \varphi\Big)\frac{\tau}{h} - (\theta_x^3 - \theta_x)\theta_y(\theta_t^2 - \theta_t) \circ \varphi - \theta_x^2\theta_y(\theta_t - 1)^2 \circ \varphi \cdot \frac{h}{2\tau}\Big].
\end{aligned}
$$

The last element is the finite-difference scheme for equation (40):

$$
\begin{aligned}
&(\varphi^n_{j+1\,k} - 2\varphi^n_{j\,k} + \varphi^n_{j-1\,k}) \cdot \Big[\Big((\varphi^n_{j\,k} - 2\varphi^n_{j-1\,k} + \varphi^n_{j-2\,k})\Big(K - \frac{(\gamma+1)}{2h}(\varphi^n_{j+1\,k} - \varphi^n_{j\,k} \\
&\quad + \varphi^n_{j-1\,k} - \varphi^n_{j-2\,k}) + (\varphi^n_{j-1\,k+1} - 2\varphi^n_{j-1\,k} + \varphi^n_{j-1\,k-1})\Big)\frac{\tau}{h} \\
&\quad - (\varphi^{n+1}_{j\,k} - \varphi^{n+1}_{j-2\,k} - \varphi^n_{j\,k} + \varphi^n_{j-2\,k}) - (\varphi^{n+1}_{j-1\,k} - 2\varphi^n_{j-1\,k} + \varphi^{n-1}_{j-1\,k})\frac{h}{2\tau}\Big] \\
&\quad + (\varphi^n_{j\,k} - 2\varphi^n_{j-1\,k} + \varphi^n_{j-2\,k}) \cdot \Big[\Big((\varphi^n_{j+1\,k} - 2\varphi^n_{j\,k} + \varphi^n_{j-1\,k})\Big(K - \frac{(\gamma+1)}{2h}(\varphi^n_{j+1\,k} - \varphi^n_{j\,k} \\
&\quad + \varphi^n_{j-1\,k} - \varphi^n_{j-2\,k}) + (\varphi^n_{j\,k+1} - 2\varphi^n_{j\,k} + \varphi^n_{j\,k-1})\Big)\frac{\tau}{h} \\
&\quad - (\varphi^{n+1}_{j+1\,k} - \varphi^{n+1}_{j-1\,k} - \varphi^n_{j+1\,k} + \varphi^n_{j-1\,k}) - (\varphi^{n+1}_{j\,k} - 2\varphi^n_{j\,k} + \varphi^{n-1}_{j\,k})\frac{h}{2\tau}\Big] = 0.
\end{aligned}
\tag{42}
$$

By construction, this scheme is fully conservative and does not involve switches that is typical for computing transonic flows [47].

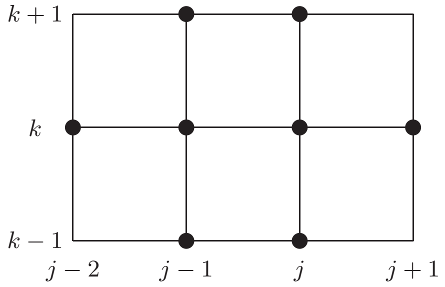

Figure 4. Stencil for stationary Falkowich–Karman equation.

In its stationary form scheme (42)

$$D_{xx}(\varphi^n_{j\,k}) \cdot \left[D_{xx}(\varphi^n_{j-1\,k}) \left(K - \frac{(\gamma+1)}{2} (D_x(\varphi^n_{j+1\,k}) + D_x(\varphi^n_{j-1\,k})) \right) + D_{yy}(\varphi^n_{j-1\,k}) \right]$$
$$+ D_{xx}(\varphi^n_{j-1\,k}) \cdot \left[D_{xx}(\varphi^n_{j\,k}) \left(K - \frac{(\gamma+1)}{2} (D_x(\varphi^n_{j+1\,k}) + D_x(\varphi^n_{j-1\,k})) \right) \right.$$
$$\left. + D_{yy}(\varphi^n_{j\,k}) \right] = 0. \tag{43}$$

is related to equation (39). Here symbols D_x and D_y are the forward differencing operators and D_{xx} and D_{yy} are the central second-order differencing operators with respect to x and y. The stencil for scheme (43) is shown in Fig. 4.

It should be noted that, unlike the original differential equation (40) which is quadratically nonlinear, both schemes (42) and (43) have the the cubic nonlinearity in the grid function. This is the rigorous algebraic consequence of the difference system (41). In accordance to the well-known fact [46], that algebraic elimination of variables from a nonlinear system leads generally to increase of its degree of nonlinearity.

As an application of this scheme, in the next section we consider an example of one-dimensional transonic flow with shock-wave taken from [47].

8 Numerical experiments

8.1 Burgers equation

We used schemes (29), (36) and (38) for numerical simulation in $0 < x < 1$ of the following Riemann problem for the inviscid Burgers equation (26) with $\nu = 0$

$$u_t + u_x u = 0, \tag{44}$$

and discontinuous initial condition $u(x, 0)$:

$$u(x,0) = \begin{cases} u_l, & 0 < x < \frac{1}{2}, \\ u_r, & \frac{1}{2} < x < 1. \end{cases} \tag{45}$$

In (45) the initial data at $t = 0$ is a piecewise-constant function with the state u_l on the left of the discontinuity $x = 0$ and the state u_r on the right of the discontinuity. We consider $\nu = 0$, since in this case the problem (44), (45) admits the exact solution:

$$u(x,t) = u_l H \left(\frac{1}{2} + \frac{u_l + u_r}{2} t - x \right) + u_r H \left(x - \frac{1}{2} - \frac{u_l + u_r}{2} t \right). \tag{46}$$

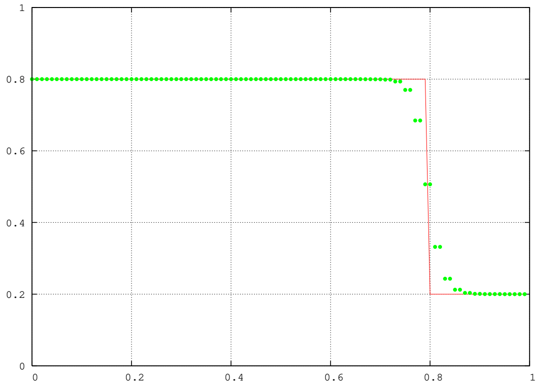

Figure 5. Lax scheme (29) with Courant number 0.9.

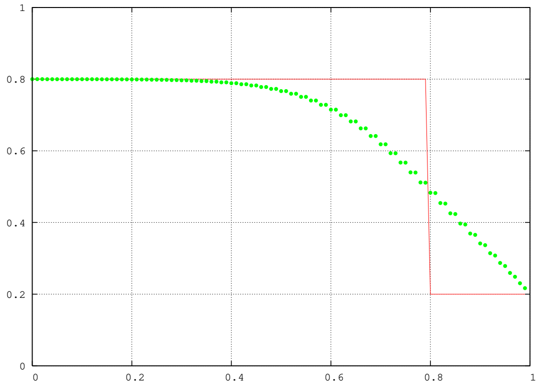

Figure 6. Lax scheme (29) with Courant number 0.1.

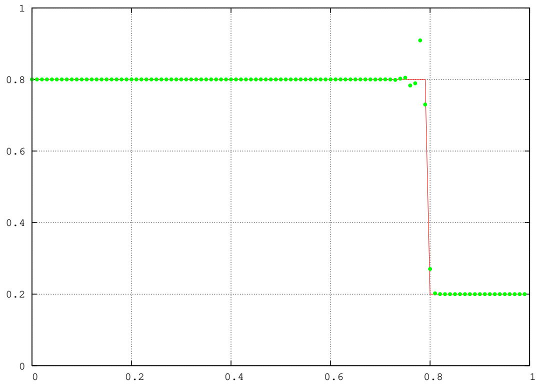

Figure 7. Lax–Wendroff scheme (36) with Courant number 0.9.

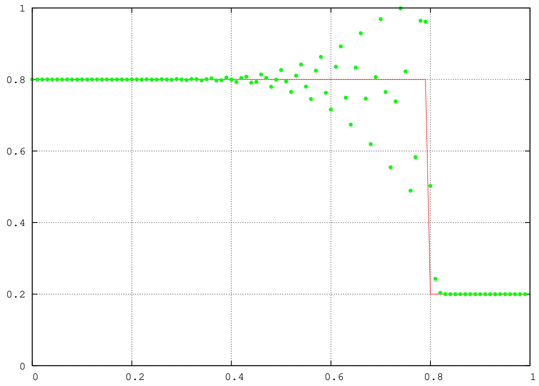

Figure 8. Lax–Wendroff scheme (36) with Courant number 0.1.

Here $H(y)$ is the Heaviside step function [48] whose derivative is the Dirac delta function:

$$H(y) = \begin{cases} 0 & y < 0, \\ \frac{1}{2} & y = 0, \\ 1 & y > 0, \end{cases} \qquad \frac{d}{dy}H(y) = \delta(y).$$

Physically, the solution (46) defined by the initial condition (45) represents a shock wave which moves with constant speed $(u_l + u_r)/2$ without changing its shape.

In our numerical simulation the values of u_l and u_l were chosen as 0.8 and 0.2. The pictures below demonstrate the numerical solution of the Riemann problem (44), (45) at time $t = 2/3$. Solid line shows the exact solution (46), and the numerical results are depicted by green dots. For the ratio τ/h of mesh steps which is called Courant (or Courant–Friedrichs–Levy) number [1] we have chosen the two values 0.9 and 0.1.

All schemes are numerically stable. For schemes (29) and (36) their stability is analytically showed by the differential approximation (Section 6.4). Because of the nonlinearity in f_x in the third equation of Godunov scheme (38) we did not compute the differential approximation for this scheme.

As can be expected (cf. [3]), the dispersion effects in schemes (29)–(36) become stronger for the smaller value of the Courant number (Figs. 6 and 8). Qualitatively [49, 50], the behavior of Lax scheme (29) in Figs. 5, 6 is typical for the classical first-order schemes when they are applied to problem (44)–(45) whereas the Lax–Wendroff scheme (36) behaves as the second-oder method. The Godunov scheme (38), as a shock capturing one (cf. [49, 50]), is much better for numerical description of solution (46) than the schemes (29)–(36), and does not reveal its sensitivity to the value of the Courant number.

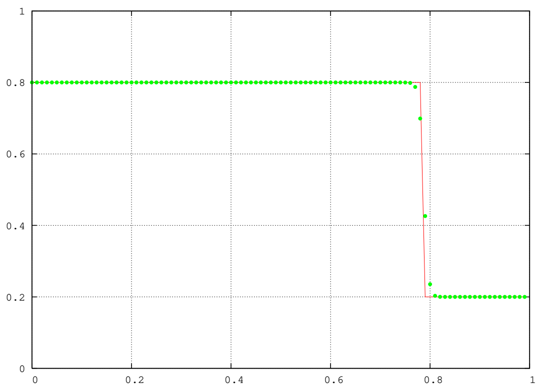

Figure 9. Godunov scheme (38) with Courant number 0.9.

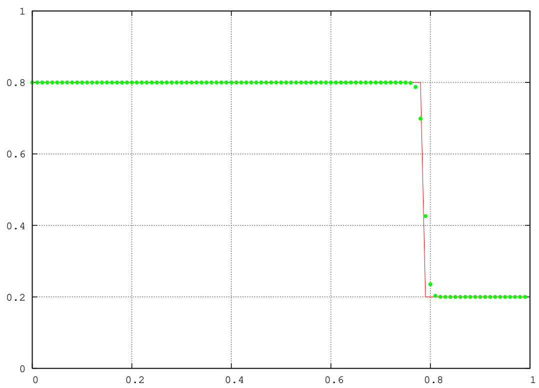

Figure 10. Godunov scheme (38) with Courant number 0.1.

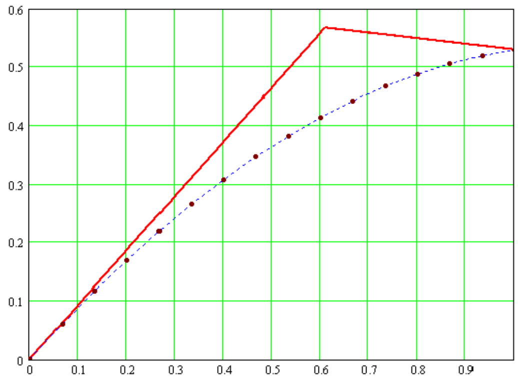

Figure 11. Initial numerical approximation for equation (39).

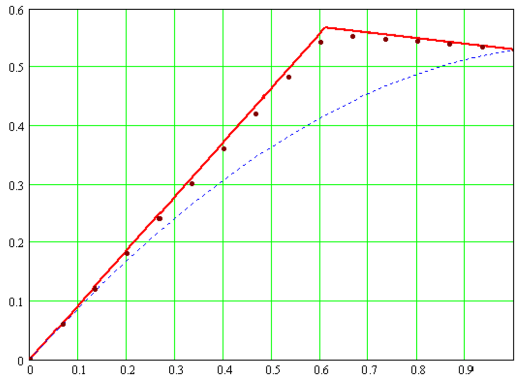

Figure 12. Numerical solution of equation (39).

8.2 Falkowich–Karman equation

Now we consider the application of difference scheme (43) to the one-dimensional stationary transonic flow in a channel with a straight density jump [47]. The exact shock-wave solution of equation (39) at $0 \le x \le 1$ is shown in Figs. 11 and 12 by solid red line. Circles depict the numerical data obtained from difference scheme (43). As an initial approximation, the parabola was chosen satisfying the following boundary conditions: at the left, both the function and its derivative are fixed by the values from the exact solution; at the right, the only function is bound to the exact solution.

As one can see from Fig. 12, scheme (43) possesses a stable and uniform convergence to the exact shock-wave solution. Because, by its construction, the scheme is fully conservative, it does not reveal non-uniqueness of solutions that is typical for the traditional difference schemes [47].

Moreover, the size of the shock transition zone is just one spatial mesh step that is a consequence of preserving at the discrete level of all algebraic properties of the initial PDE (40). This is a result of algebraic difference elimination provided by the Gröbner bases method. Another merit of scheme (43) is that it does not involve switches that is typical for computing transonic flow as we already pointed out in Section 7.

This example shows a principal possibility of constructing difference schemes for transonic flow without switches and with the same stencil for both subsonic and supersonic flow.

9 Conclusion

In the present paper we have shown that the Gröbner bases method, being a universal algorithmic tool for linear difference algebra, can be effectively applied to the construction of differences schemes for linear PDEs with two independent variables and with rational function coefficients. Owing to the Gröbner bases, this construction is an algorithmic procedure. It consists in elimination of partial derivatives from the system of difference equations composed from a discrete version of the original PDEs (on an orthogonal uniform grid) and numerically approximated integral relations between the unknown functions and their partial derivatives. As this takes place, the difference scheme obtained may depend on the choice of the integration contour and numerical approximations for integral relations.

The method is especially efficient when a PDE or a system of PDEs admits the conservation law form. In this case the difference schemes obtained are fully conservative. The structure of a scheme generated may depend on the choice of integration contour and numerical integration rules. In so doing, it is not clear a priori which integration rule leads to a better scheme.

We also described an efficient algorithm for the construction of Gröbner bases for linear difference ideals. The algorithm is based on the concept of Janet-like reductions. Its first implementation in Maple is already available, and we used this implementation in the generation of all linear difference schemes presented in the paper.

For classical linear PDEs such as the Laplace equation, the Heat equation, the Wave equation and the Advection equation our algorithmic technique leads to the well-known finite difference schemes. For Burgers equation we generated several schemes based on the Lax and Lax–Wendroff methods and computed their numerical dissipation and dispersion by the differential approximation (modified equation) method. By example of Burgers equation we also demonstrate that it is possible to combine the Godunov method with Gröbner bases to derive a shock capturing scheme.

The non-traditional cubic nonlinear difference scheme generated by our difference elimination method for the Falkowich–Karman equation describing transonic flow in gas dynamics possesses a number of attractive properties in comparison with traditional schemes. Among them there are a stable convergence in time to the exact solution with a one-dimensional shock wave and absence of switches. It should be noted, however, that due to its cubic nonlinearity, scheme (43) has a slower convergence in comparison with the traditional schemes specially optimized for numerical simulation of transonic flows in gas dynamics. By this reason one needs additional research for optimizing nonlinear schemes obtained by the difference elimination.

As we already mentioned in the introduction, algorithm **GröbnerBasis** admits a generalization to polynomial-nonlinear systems of difference equations exactly in the same way as the differential involutive algorithm of paper [32]. In doing so, if every equation in the initial system is linear with respect to the highest ranking difference term and this property of the system is not violated during its completion to involution, then algorithm **GröbnerBasis** will work correctly and provide the desirable output. Such is indeed the case for system (41). In the most general case of a difference system with polynomial nonlinearity, it can be split into a finite number of subsystems such that every subsystem can be converted into the Gröbner basis form by applying our algorithm. The underlying splitting algorithm is a difference analogue of that described in [31]. The latter algorithm is similar to the splitting algorithm implemented in the library package *diffalg* in Maple.

The above described approach can be also generalized to PDEs with three and more independent variables. Thus, if PDEs admit the conservation law form, then one can use multidimensional analogues of equations (1) and (2) together with their elementary volume discretization.

Acknowledgements

We would like to thank the referees for their important remarks that allowed us to correct the manuscript. We are also grateful to Daniel Robertz and Viktor Levandovskyy for useful discussions and comments. The contribution of two authors (V.P.G. and Yu.A.B.) was partially supported by grants 04-01-00784 and 05-02-17645 from the Russian Foundation for Basic Research and by grant 2339.2003.2 from the Ministry of Education and Science of the Russian Federation.

[1] Godunov S.K., Ryaben'kii V.S., Difference schemes. An introduction to the underlying theory, New York, Elsevier, 1987.

[2] Strikwerda J.C., Finite difference schemes and partial differential equations, 2nd ed., Philadelphia, SIAM, 2004.

[3] Ganzha V.G., Vorozhtsov E.V., Numerical solutions for partial differential equations: problem solving using Mathematica, Boca Raton, CRC Press, 1996.

[4] Quarteroni A.,Valli A., Numerical approximation of partial differential equations, 2nd ed., Berlin, Springer-Verlag, 1997.

[5] Thomas J.W., Numerical partial differential equations: finite difference methods, 2nd ed., New York, Springer-Verlag, 1998.

[6] Thomas J.W., Numerical partial differential equations: conservation laws and elliptic equations, New York, Springer-Verlag, 1999.

[7] Samarskii A.A., The theory of difference schemes, New York, Marcel Dekker, 2001.

[8] Morton K.W., Mayers D.F., Numerical solution of partial differential equations, 2nd ed., Cambridge, Cambridge University Press, 2005.

[9] Ganzha V.G., Vorozhtsov E.V., Computer-aided analysis of difference schemes for partial differential equations, New York, Wiley-Interscience, 1996.

[10] Liska R., Shashkov M.Yu., Algorithms for difference schemes construction on non-orthogonal logically rectangular meshes, in Proceedings of ISSAC'91, New York, ACM Press, Addison Wesley, 1991, 419–426.

[11] Liska R., Shashkov M.Y., Solovjov A.V., Support-operators method for PDE discretization: symbolic algorithms and realization, *Mathematics and Computers in Simulation*, 1994, V.35, 173–183.

[12] Fournié M., Symbolic derivation of different class of high-order schemes for partial differential equations, in Computer Algebra in Scientific Computing / CASC'99, Springer-Verlag, 1999, 93–100.

[13] Köller R., Mohl K.D., Schramm H., Zeitz M., Kienle A., Mangold M., Stein E., Gilles E., Method of lines within the simulation environment DIVA for chemical processes, in Adaptive Method of Lines, Chapmann and Hall, 2001, 371–406.

[14] Mozzhilkin V.V., Blinkov Yu.A., Methods of constructing difference schemes in gas dynamics, *Transactions of Saratov University*, 2001, V.1, N 2, 145–156 (in Russian).

[15] Buchberger B., An algorithm for finding a basis for the residue class ring of a zero-dimensional polynomial ideal, PhD Thesis, University of Innsbruck, Institute for Mathematics, 1965 (in German).

[16] Buchberger B.,Winkler F. (Editors), Gröbner bases and applications, Cambridge University Press, 1998.

[17] `http://www.maplesoft.com/`

[18] `http://www.wolfram.com/`

[19] Greuel G.-M., Pfister G., Schönemann H., Singular 3.0. A computer algebra system for polynomial computations, Centre for Computer Algebra, University of Kaiserslautern, 2005, `http://www.singular.uni-kl.de/`

[20] `http://magma.maths.usyd.edu.au/magma/`

[21] Cohn R.M., Difference algebra, *Tracts in Mathematics*, Vol. 17, Interscience Publishers, 1965.

[22] Kondratieva M.V., Levin A.B., Mikhalev A.V., Pankratiev E.V., Differential and difference dimension polynomials. Mathematics and its applications, Dordrecht, Kluwer, 1999.

[23] Buchberger B., Gröbner bases: an algorithmic method in polynomial ideal theory, in Recent Trends in Multidimensional System Theory, Dordrecht, Reidel, 1985, 184–232.

[24] Chyzak F., Salvy B., Non-commutative elimination in Ore algebras proves multivariate identities, *J. Symbolic Computation*, 1998, V.26, 187–227.

[25] Chyzak F., Quadrat A., Robertz D., OreModules: a symbolic package for the study of multidimensional linear systems, in Applications of Time-Delay Systems, Editors J. Chiasson and J.-J. Loiseau, Springer, to appear (see `http://wwwb.math.rwth-aachen.de/OreModules`).

[26] Greuel G.-M., Levandovskyy V., Schönemann H., Plural. A subsystem of the computer algebra system Singular 3.0 for computations with non-commutative polynomial algebras, Centre for Computer Algebra, University of Kaiserslautern, 2005, `http://www.singular.uni-kl.de`

[27] Gerdt V.P., Involutive algorithms for computing Gröbner bases, in Computational Commutative and Non-Commutative Algebraic Geometry, Amsterdam, IOS Press, 2005, 199–225, math.AC/0501111.

[28] `http://invo.jinr.ru`

[29] Gerdt V.P., Robertz D., A Maple package for computing Gröbner bases for linear recurrence relations, cs.SC/0509070.

[30] Janet M., Leçons sur les Systèmes d'Equations aux Dérivées Partielles, Cahiers Scientifiques, IV, Paris, Gauthier-Villars, 1929.

[31] Thomas J., Differential systems, New York, American Mathematical Society, 1937.

[32] Gerdt V.P., Completion of linear differential systems to involution, in Computer Algebra in Scientific Computing CASC'99, Editors V.G. Ganzha, E.W. Mayr and E.V. Vorozhtsov, Berlin, Springer, 1999, 115–137, math.AP/9909114.

[33] Gerdt V.P., Blinkov Yu.A., Janet-like monomial division. Janet-like Gröbner bases, in Computer Algebra in Scientific Computing / CASC 2005, LNCS, Springer, 2005, 174–195.

[34] Gerdt V.P., On computation of Gröbner bases for linear difference systems, math-ph/0509050.

[35] von zur Gathen J., Gerhard J., Modern computer algebra, 2nd ed., Cambridge University Press, 2003.

[36] Gerdt V.P., Blinkov Yu.A., Involutive bases of polynomial ideals. Minimal involutive bases, *Mathematics and Computers in Simulation*, 1998, V.45, 519–560, math.AC/9912027, math.AC/9912029.

[37] Apel J., Hemmecke R., Detecting unnecessary reductions in an involutive basis computation, RISC Linz Report Series 02-22, 2002.

[38] `http://www-sop.inria.fr/saga/POL`

[39] `http://www.math.uic.edu/~jan/demo.html`

[40] Shokin Yu.I., Yanenko N.N., Method of differential approximation. Application to gas dynamics, Nauka, Siberian Division, 1985 (in Russian).

[41] Richtmyer R.D., Morton K.W., Difference methods for initial-value problems, 2nd ed., New York, John Wiley & Sons, 1967 (reprinted Krieger Publishing Company, New York, 1994).

[42] Godunov S.K., A finite difference method for numerical computation and discontinuous solutions of the equations of fluid dynamics, *Math. Sb.*, 1959, V.47, 271–306 (in Russian).

[43] Toro E.F., Riemann solvers and numerical methods for fluid dynamics, 2nd ed., Berlin, Springer-Verlag, 1997.

[44] Czapor S.R., Solving algebraic equations: combining Buchberger's algorithm with multivariate factorization, *J. Symbolic Computation*, 1989, V.7, 49–53.

[45] von Karman T., Collected works, von Karman Institute for Fluid Dynamics, 1975 (see also Butterworth Scientific Publ., London, 1956).

[46] Bykov V., Kutmanov A., Lazman M., Elimination methods in polynomial computer algebra, Kluwer Academic Publishers, 1997.

[47] Jameson A., Transonic flow calculations, von Karman Institute for Fluid Dynamics Lecture Series, Vol. 87, 1976;
Jameson A., Numerical methods in fluid dynamics, Hemisphere, 1978, 1–87.

[48] Spanier J.,Oldham K.B., The unit-step $u(x-a)$ and related functions, Ch. 8, in An Atlas of Functions, Washington, DC, Hemisphere, 1987.

[49] Yee H.C., A class of high-resolution explicit and implicit shock-capturing methods, Technical Report Lecture Series 1989-04, von Karman Institute for Fluid Dynamics, 1989.

[50] Manizini M., Numerical methods for 1D compressible flows, an interactive book, see here.