Computer Algebra in Problem Solving for Computational Fluid Dynamics: Term Rewriting and *All That*

Victor G. Ganzha[1], Dmytro Chibisov[1], and Evgenii V. Vorozhtsov[2]

[1] Institute of Informatics, Technical University of Munich, Munich 80290, Arcisstr. 21, Germany;
ganzha@informatik.tu-muenchen.de, chibisov@informatik.tu-muenchen.de
[2] Institute of Theoretical and Applied Mechanics, Russian Academy of Sciences, Novosibirsk 630090, Russia;
vorozh@itam.nsc.ru

Abstract. The present paper studies the application of symbolic facilities of Computer Algebra Systems (CAS), in particular Maple, to the development of numerical methods. As will be shown the complex algebraic relationships in the derivation of a particular numerical method can be captured by computer algebra. We use the formalism of term rewriting system in order to demonstrate the derivation of numerical schemes with the given properties, such as conservativity or non-linear stability property (TVD schemes) using CAS Maple. In order to demonstrate our approach we consider a two-dimensional inviscid gas flow involving shock waves.

1 Introduction

The rapid development of mathematical models and methods plays an important role in all fields of science and engineering. We are interested in developing general methods on the symbolic-numerical basis for solving modelling problems for inviscid fluid dynamics. The current methods of solving such problems range from complex analytic and numerical models to extensive numerical code. The Problem Solving Environments (PSE) are usually used to cope with this complexity. The existing PSEs, such as SciNapse [23] or CTADEL [11], have a lot of very useful features. For example, the PSE transforms high level description of initial- and boundary-value problems for PDE to efficient, documented and executable code, which is typically generated in C or Fortran.

But the solving of modern fluid dynamics tasks requires not simply an automatable process of transforming one description into another; it involves complex synthesis and analysis tasks in order to understand the multilevel relationships between different objects in the particular numerical method.

Among the current methods for the numerical solution of the conservation laws (13) widely used in fluid dynamics one can identify several groups of methods, which have gained a widespread acceptance:

– approximate Riemann solvers [24];
– Runge-Kutta finite volume schemes with artificial dissipators [6, 14, 17, 18];
– TVD methods [21, 24].

In [15, 16] we have shown how the graphical data modelling techniques can be used to obtain a numerical Runge–Kutta finite volume Euler solver automatically. In this approach the developer has to identify the objects involved in the Runge-Kutta method and to specify the relations among their attributes. Such objects are, for example, finite volumes $V^n_{j,k}$ ordered in space and time with values of pressure, density and Cartesian velocity components as attributes. Any numerical scheme can be expressed as relationship between the attributes of the appropriate volumes objects. In this way the specified associations between objects correspond, for example, to the numerical domain of dependence of a particular scheme. Furthermore we have presented the tool prototype called GROOME that provides graphical diagram editor to describe such objects and relations. It has been shown how the numerical code according to the Jameson scheme can be generated automatically from such diagrams.

The present paper deals with a wide variety of TVD methods. At the same time we present some important extensions of the general GROOME methodology concerning the usage of symbolic facilities of computer algebra for the derivation of numerical methods. As will be shown the traditional symbolic computation features of computer algebra systems, such as, for example, simplification, factorization or expansion of expressions do not support the operations needed for the derivation of many very useful numerical methods such as, for example, high order methods with TVD-property (Total Variation Diminishing) widely used as a non-linear stability property.

It was shown in [2] how the conservative but not necessarily stable high order numerical schemes can be derived from the integral form of physical laws algorithmically. The authors propose the usage of

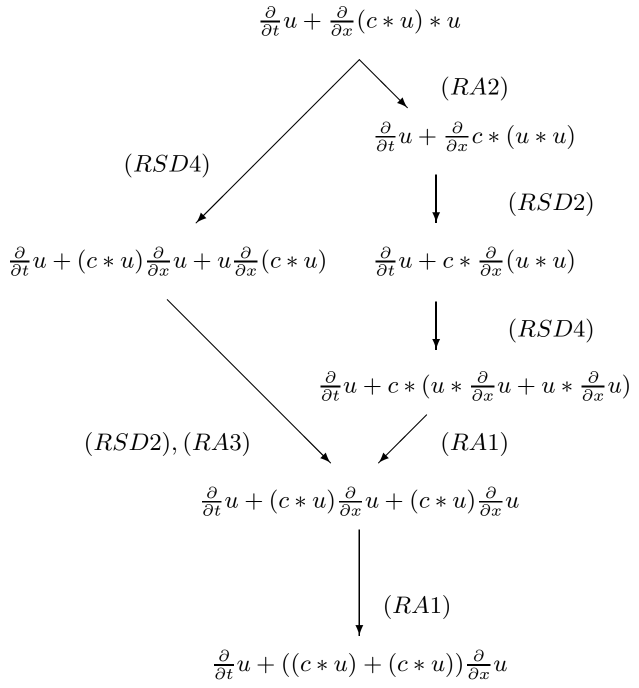

Fig. 1. Possible reduction of the Burgers equation

Gröbner Bases to obtain conservative high order schemes as a compatibility condition for the system of equations consisting of equations describing the conservation of some quantities and equations describing the integral approximation rules.

In this paper we investigate the usage of term rewriting systems to support the developer by derivation of numerical methods. The connection between Buchberger's Gröbner Bases algorithm and critical-pairs/completion of term rewriting systems was first observed in [22, 5], and more closely analysed in [3, 4].

We will show how the symbolic term rewriting can be used to derive the numerical schemes with given properties, such as conservativity or TVD property. Our approach is motivated by the book of Franz Baader and Tobias Nipkow "Term Rewriting and *All That*" [1].

First of all let us give one example that illustrates some of the key issues arising in connection with term rewriting systems generally and in particular with those that arise in connection with symbolic-numerical methods (for a more precise introduction to term rewriting systems see [1]).

2 Motivating Example

As a simple example of a Term Rewriting System (**TRS**) consider the symbolic differentiation of arithmetic expressions that are built with the operations `+`, `*` and /, indeterminates `u` (any function), `c` (any constant), `x`, `t` and numbers `0`, `1`.

For the partial derivative with respect to one of variables `x` and `t` we introduce the additional function symbols `Dx` resp. `Dt`. The following rules are some of the well-known rules for computing the derivative:

$$(RSD1) \qquad Dx(c) \to 0,$$

$$(RSD2) \qquad Dx(c * u) \to c * Dx(u),$$

$$(RSD3) \qquad Dx(a + b) \to Dx(a) + Dx(b),$$

$$(RSD4) \qquad Dx(a * b) \to a * Dx(b) + b * Dx(a).$$

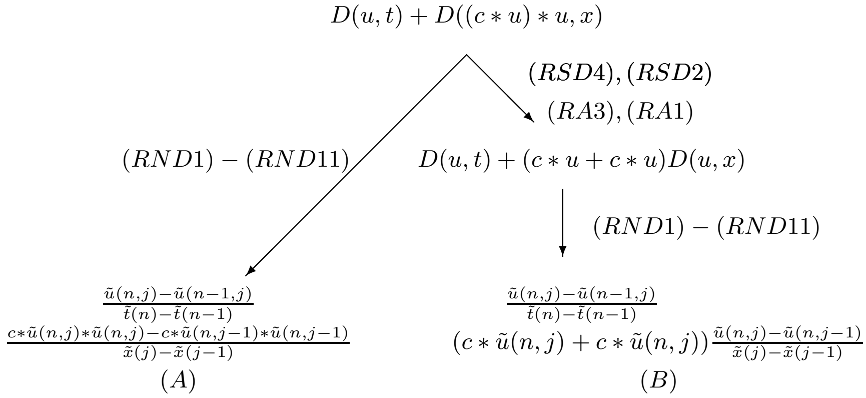

Fig. 2. Non-confluence of (RSD2), (RSD4), (RA1), (RA4), (RND1)–(RND11) leads to the possible derivation of the non-conservative scheme (B)

$$(RSD5) \qquad Dx(a(b)) \to Dx(a) * Dx(b)$$

The symbols a and b are variables that can be replaced by arbitrary expression. These variables should not be confused with the indeterminates, for example, c and u, which are constant symbols. The rules for the computation of time derivative `Dt` can be formulated in a similar way.

In order to demonstrate our example we need additionally some well-known algebraic rules.

The distributive law of arithmetics can be expressed with the aid of the bidirectional rewriting rule as follows:

$$(RA1) \qquad a * b + a * d \leftrightarrow a * (b + d)$$

Note that this rule applied in "$\leftarrow$"-direction corresponds to expansion implemented in almost all computer algebra systems (in Maple the command `expand(a*(b+d))`). In "$\to$"-direction this rule corresponds to Maple command `collect(a,a*b+a*d)`.

The associativity and commutativitiy of multiplication:

$$(RA2) \qquad a * (b * d) \leftrightarrow (a * b) * d$$

$$(RA3) \qquad a * b \leftrightarrow b * a$$

And the following rules that we will need to derive the TVD methods:

$$(RA4) \qquad \frac{a}{b} * \frac{b}{d} \to \frac{a}{d}$$

$$(RA5) \qquad \frac{a}{b} + \frac{d}{b} \to \frac{1}{b} * (a + d)$$

Starting with the left-hand side of the Burgers equation described by the term $t = \frac{\partial}{\partial t}u + \frac{\partial}{\partial x}cu^2$ the above defined rules lead to the possible reduction depicted in Fig. 1.

Obviously there are different ways of applying rules to a given term t leading to different derived terms t_1 and t_2. As shown in Fig. 1 such terms in our example can be joined. But can we always find a common term s that can be reached both from t_1 and t_2 by the rule application? If this is the case the TRS is called **confluent**.

If we add the following numerical differentiation rules to discretize the above Burgers equation in space by left one-sided differences through replacing of the derivative operator Dx by the difference operator $\tilde{Dx}$, the function u by its value at the appropriate grid point $\tilde{u}(n,j)$ (n and j are indeterminates) , the continuous variables x, t by their discrete form $\tilde{x}(n)$, $\tilde{t}(j)$ and introducing the shift operator with respect to a particular discretization variable, for example, $Tx(\tilde{u}(n,j)) = \tilde{u}(n,j-1)$

$$(RND1) \qquad Dx(u) \to \tilde{Dx}(u)$$

$(RND2)$ $$u \to \tilde{u}(n, j)$$

$(RND3)$ $$\tilde{Dx}(a) \to \frac{a - Tx(a)}{d - Tx(d)}$$

$(RND4)$ $$Tx(a * b) \to Tx(a) * Tx(b)$$

$(RND5)$ $$Tx(c) \to c$$

$(RND6)$ $$Tx(\tilde{u}(n, j)) \to \tilde{u}(n, j - 1)$$

$(RND8)$ $$x \to \tilde{x}(j)$$

$(RND9)$ $$t \to \tilde{t}(n)$$

$(RND10)$ $$Tx(\tilde{x}(j)) \to \tilde{x}(j - 1)$$

in (RSD1)–(RSD5), we lose the confluence. As shown in Fig. 2 we obtain two different discrete equations that can not be joined. It is well known that one of them is conservative, another one is not.

In the present paper we consider the discretization of conservation laws widely used in fluid dynamics as term rewriting strategies. Our aim is the development of such strategies in the way that would enable us to obtain the numerical schemes with given properties, in particular, the non-linear stability or TVD property.

3 Total Variation Diminishing Methods

A well established approach for constructing high-order TVD schemes is the flux limiter approach [21]. This requires a high-order flux F^H associated with a scheme of accuracy greater than or equal to two and a low-order flux F^L associated with a monotone first-order scheme. At first we present the approach in terms of model conservation law

$$u_t + f(u)_x = 0 \tag{1}$$

and then show a term rewriting system based on this approach that enables one to obtain the TVD schemes for a particular non-linear equation dependent on the user specified fluxes.

Equation (1) is approximated by the following difference scheme:

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}(F_{i+1/2} - F_{i-1/2}) \tag{2}$$

One then defines a high-order TVD flux as

$$F_{i+1/2}^{TVD} = F_{i+1/2}^L + \phi_{i+1/2}(F_{i+1/2}^H - F_{i+1/2}^L) \tag{3}$$

$$F_{i-1/2}^{TVD} = F_{i-1/2}^L + \phi_{i-1/2}(F_{i-1/2}^H - F_{i-1/2}^L), \tag{4}$$

where $\phi_{i\pm1/2}$ is a flux limiter function yet to be determined.

To preserve some generality we assume that $F_{i+1/2}^L$ and $F_{i-1/2}^H$ are respectively of the form

$$\begin{aligned} F_{i+1/2}^L &= -a_1\alpha_1 u_j^n + a_1\alpha_1 u_{j+1}^n \\ F_{i+1/2}^H &= -a_1\beta_1 u_j^n + a_1\beta_1 u_{j+1}^n \\ F_{i-1/2}^L &= a_1\alpha_0 u_j^n - a_1\alpha_0 u_{j-1}^n \\ F_{i-1/2}^H &= a_1\beta_0 u_j^n - a_1\beta_0 u_{j-1}^n \end{aligned} \tag{5}$$

for some a_1, α_0, β_0, α_1, β_1.

The following theorem of Harten can be used to give the constraints on the $\phi_{i\pm1/2}$:

Theorem (Harten) 1 . *In order for the method of the form*

$$u_j^{n+1} = u_j^n - C_{j-1}(u_j - u_{j-1}) + D_j(u_{j+1} - u_j) \tag{6}$$

to be TVD, the following conditions on the coefficients are sufficient:

$$\begin{aligned} C_{j-1} &\geq 0 && \forall j \\ D_j &\geq 0 && \forall j \\ D_j + C_j &\leq 1 && \forall j \end{aligned} \tag{7}$$

Proof. *The half-page proof of this Theorem can be found in [21].*

Note that the coefficients C_{j-1} and D_j are in general assumed to be dependent on the data $u_{j-k}^n \dots u_{j+k}^n$ for some k.

The substitution of (3) and (4) in (2) yields:

$$u_i^{n+1} = u_i^n - \frac{\Delta t}{\Delta x}[(F_{i+1/2}^L - F_{i-1/2}^L) + \phi_{i+1/2}(F_{i+1/2}^H - F_{i+1/2}^L) - \phi_{i-1/2}(F_{i-1/2}^H - F_{i-1/2}^L)]\,. \tag{8}$$

The problem to find $\phi_{i\pm 1/2}$ in equation (8) in accordance with the TVD condition given by Harten theorem can now be solved in two steps:

- determine the coefficients C_{j-1} and D_j in (6) with regard for (8)
- apply any user defined flux limiter function ϕ to C_{j-1} and D_j that would guarantee (7) (a well established flux limiter function is *minmod*, see, for example, [21]).

Let us add the function symbol $F(u)$ (flux function) to our TRS. Then application of (RSD5) to our model problem reduces it to the so-called wave form:

$$D(u,t) + D(F(u),x) \rightarrow D(u,t) + D(F(u),u) * D(u,x)\,.$$

If we add the rule

$(RND12)$ $$T(F(a),d) \rightarrow F(T(a,x))$$

both terms $D(u,t)+D(F(u),x)$ and $D(u,t)+D(F(u),u)*D(u,x)$ can be discretized to the same unique irreducible form

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + \frac{F(u_j^n) - F(u_{j-1}^n)}{\Delta x} \tag{9}$$

by using (RND1)–(RND12) and (RA4).

Furthermore we add the following rule with two new function symbols $F_{i\pm 1/2}$:

$(RND13)$ $$F \rightarrow F_{i+1/2} - F_{i-1/2}$$

According to (3) and (4) let us add the TVD flux rules:

$(RTVD1)$ $$F_{i+1/2} \rightarrow F_{i+1/2}^L + \phi_{i+1/2}(F_{i+1/2}^H - F_{i+1/2}^L)$$

$(RTVD2)$ $$F_{i-1/2} \rightarrow F_{i-1/2}^L + \phi_{i-1/2}(F_{i-1/2}^H - F_{i-1/2}^L)$$

But with regard for (5) the following rules must hold too:

$(RTVD3)$ $$F_{i+1/2}^L \rightarrow -a_1\alpha_1 u_j^n + a_1\alpha_1 u_{j+1}^n$$

$(RTVD4)$ $$F_{i+1/2}^H \rightarrow -a_1\beta_1 u_j^n + a_1\beta_1 u_{j+1}^n$$

$$u_{n,j} - v\left(\underline{\mathbf{F_{i+1/2}}} - F_{i-1/2}\right)$$
$$(RTVD1)-(RTVD6)\ \downarrow$$
$$a_1\alpha_1\left(\underline{-\mathbf{u_{n,j}}} + \underline{\mathbf{u_{n,j+1}}}\right) + \phi_{i+1/2}\left(a_1\beta_1\left(\underline{-\mathbf{u_{n,j}}} + \underline{\mathbf{u_{n,j+1}}}\right) - a_1\alpha_1\left(\underline{-\mathbf{u_{n,j}}} + \underline{\mathbf{u_{n,j+1}}}\right)\right) - F_{i-1/2}$$
$$(RA1)\ \downarrow$$
$$-\left(a_1\alpha_1 + \phi_{i+1/2}a_1\beta_1 - \phi_{i+1/2}a_1\alpha_1\right)u_{n,j} + \left(a_1\alpha_1 + \phi_{i+1/2}a_1\beta_1 - \phi_{i+1/2}a_1\alpha_1\right)u_{n,j+1} - \underline{\mathbf{F_{i-1/2}}}$$
$$(RTVD1)-(RTVD8)\ \downarrow$$
$$\left(a_1\alpha_1 + \phi_{i+1/2}a_1\beta_1 - \phi_{i+1/2}a_1\alpha_1\right)u_{n,j} + \left(a_1\alpha_1 + \phi_{i+1/2}a_1\beta_1 - \phi_{i+1/2}a_1\alpha_1\right)u_{n,j+1} -$$
$$\left(a_1\alpha_0\left(\underline{\mathbf{u_{n,j}}} - \underline{\mathbf{u_{n,j-1}}}\right) + \phi_{i-1/2}\left(a_1\beta_0\left(\underline{\mathbf{u_{n,j}}} - \underline{\mathbf{u_{n,j-1}}}\right) - a_1\alpha_0\left(\underline{\mathbf{u_{n,j}}} - \underline{\mathbf{u_{n,j-1}}}\right)\right)\right)$$
$$(RA11)\ \downarrow$$
$$\underline{\underline{-\left(\mathbf{a_1}\alpha_1 + \phi_{\mathbf{i+1/2}}\mathbf{a_1}\beta_1 - \phi_{\mathbf{i+1/2}}\mathbf{a_1}\alpha_1\right)\mathbf{u_{n,j}} + \left(\mathbf{a_1}\alpha_1 + \phi_{\mathbf{i+1/2}}\mathbf{a_1}\beta_1 - \phi_{\mathbf{i+1/2}}\mathbf{a_1}\alpha_1\right)\mathbf{u_{n,j+1}} -}}$$
$$\underline{\left(\left(\mathbf{a_1}\alpha_\mathbf{0} + \phi_{\mathbf{i-1/2}}\left(\mathbf{a_1}\beta_\mathbf{0} - \mathbf{a_1}\alpha_\mathbf{0}\right)\right)\mathbf{u_{n,j}} - \left(\mathbf{a_1}\alpha_\mathbf{0} + \phi_{\mathbf{i-1/2}}\left(\mathbf{a_1}\beta_\mathbf{0} - \mathbf{a_1}\alpha_\mathbf{0}\right)\right)\mathbf{u_{n,j-1}}\right)}$$
$$(RA11)\ \downarrow$$
$$\left(a_1\alpha_1 + \phi_{i+1/2}a_1\beta_1 - \phi_{i+1/2}a_1\alpha_1\right)(u_{n,j+1} - u_{n,j}) -$$
$$\left(\left(a_1\alpha_0 + \phi_{i-1/2}\left(a_1\beta_0 - a_1\alpha_0\right)\right)(u_{n,j} - u_{n,j-1})\right)$$

Fig. 3. Application of rules (RTVD1)-(RTVD6) and (RA1) to flux limiter scheme 2 in order to obtain the algebraic form corresponding to Harten theorem. The underlined symbols denote the terms to which the appropriate rules are applied

$$(RTVD5) \qquad F^L_{i-1/2} \to a_1\alpha_0 u^n_j - a_1\alpha_0 u^n_{j-1}$$

$$(RTVD6) \qquad F^H_{i-1/2} \to a_1\beta_0 u^n_j - a_1\beta_0 u^n_{j-1}$$

Starting from equation (9) obtained with (RND1)–(RND12) and (RA5) the reduction depicted in Fig. 3 leads to the following equations in order to satisfy the Harten theorem:

$$D_j = a_1\alpha_1 + \phi_{i+1/2}a_1\beta_1 - \phi_{i+1/2}a_1\alpha_1 \geq 0 \tag{10}$$

$$C_{j-1} = a_1\alpha_0 + \phi_{i-1/2}\left(a_1\beta_0 - a_1\alpha_0\right) \geq 0 \tag{11}$$

$$\begin{aligned} 1 - C_{j-1} - D_j = 1 &- \left(a_1\alpha_1 + \phi_{i+1/2}a_1\beta_1 - \phi_{i+1/2}a_1\alpha_1\right) - \\ &\left(a_1\alpha_0 + \phi_{i-1/2}\left(a_1\beta_0 - a_1\alpha_0\right)\right) \geq 0 \end{aligned} \tag{12}$$

In Appendix A we present the Maple implementation of the used TRS to the derivation of these conditions. In particular, one can use the *minmod* function to fulfill the conditions (10), (11), and (12) (see [20]).

Let us demonstrate some results obtained by the application of this technique to a more complex fluid dynamics task involving the shock waves in a two-dimensional gas flow.

4 Equations to Solve

Consider the Euler equation in the following conservation form:

$$\frac{\partial w}{\partial t} + \frac{\partial f(w)}{\partial x} + \frac{\partial f(w)}{\partial y} = 0, \tag{13}$$

where x and y are Cartesian coordinates and

$$w = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \; f(w) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{pmatrix}, \; g(w) = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vH \end{pmatrix}. \tag{14}$$

Here p, ρ, u, v, E and H denote the pressure, density, Cartesian velocity components, total energy, and total enthalpy. For a perfect gas

$$E = \frac{p}{(\gamma - 1)\rho} + \frac{1}{2}(u^2 + v^2), \quad H = E + \frac{p}{\rho}, \tag{15}$$

where γ is the ratio of specific heats.

As a flow problem we have chosen a simple problem of inviscid flow developed by an oblique stationary shock wave reflecting from a rigid surface (Fig. 4). This test problem is often used for the validation of new numerical methods in computational fluid dynamics. The advantage of this test is that it is possible to obtain the exact solution for it by using the theory of stationary oblique shocks. This solution represents a piecewise constant function. We have used the value $\varphi = \pi/6$ for the angle between the incident shock wave front and the x axis (see Fig. 4). In the case of perfect gas (air) with $\gamma = 1.4$, the constants of the exact solution in subregions 1, 2, and 3 indicated in Fig. 4 are as follows:

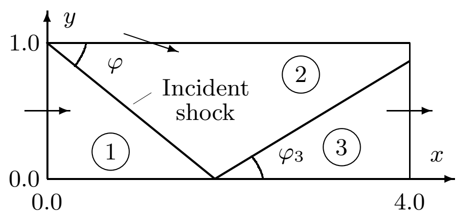

Fig. 4. Spatial region in the shock reflection problem

Subregion 1	Subregion 2	Subregion 3
$u_1 = 1.0$	$u_2 = 0.890755053$	$u_3 = 0.806645743$
$v_1 = 0.0$	$v_2 = -0.189217798$	$v_3 = 0.0$
$p_1 = 0.084932903$	$p_2 = 0.194177850$	$p_3 = 0.390838939$
$\rho_1 = 1.0$	$\rho_2 = 1.776135164$	$\rho_3 = 2.898621574$
$M_1 = 2.90$	$M_2 = 2.327642861$	$M_3 = 1.856588584$

Here M_1, M_2, and M_3 are the values of the Mach number $M = \sqrt{u^2 + v^2}/c$ in subregions 1, 2, and 3, respectively; c is the sound velocity, $c = \sqrt{\gamma p/\rho}$. The reflected shock wave front makes the angle $\varphi_3 = 0.418279545$ with the positive direction of the x-axis.

Initial conditions. Initially, the entire flow field is set equal to the free stream supersonic inflow values given above for subregion 1, that is the initial gas flow is parallel with the x-axis.

Analytical boundary conditions. The spatial region has the size 4 along the x-axis and the size 1 along the y-axis. The boundary conditions are given as follows:
(a) supersonic inflow at $x = 0$, $0 \le y \le 1$, which allows the values of u, v, p, ρ to be fixed at free stream conditions given above as subregion 1;
(b) prescribed fixed values from subregion 2 at $y = 1$, $0 \le x \le 4$, which produce the desired shock strength and shock angle;
(c) supersonic outflow at $x = 4$, $0 \le y \le 1$;
(d) a rigid flat surface at $y = 0$, $0 \le x \le 4$, which is properly represented by the condition $v = 0$ with the additional condition $\partial p/\partial y = 0$ at $y = 0$ from the normal y-momentum equation.

Numerical boundary conditions. The supersonic outflow values $w_{J,k}$, $k = 1, \ldots, K$, are obtained by zeroth-order extrapolation, i.e., $w_{J,k} = w_{J-1,k}$, $k = 1, \ldots, K$.

For the computation of the TVD corrections in the Chakravarthy–Osher method, the numerical solution values are needed in two rows of image cells adhering to the spatial region boundaries. We have specified the values of the components of the solution vector w in image cells in accordance with the symmetry technique presented in [10]:

Rigid wall	Outflow boundary $x = 4$
$p_{j,0} = p_{j,1};\ p_{j,-1} = p_{j,2}$	$w_{J,k} = w_{J-1,k}$
$\rho_{j,0} = \rho_{j,1};\ \rho_{j,-1} = \rho_{j,2}$	$w_{J+1,k} = w_{J-2,k}$
$u_{j,0} = u_{j,1};\ u_{j,-1} = u_{j,2}$	
$v_{j,0} = -v_{j,1};\ v_{j,-1} = -v_{j,2}$	

The numerical solution values in two rows of cells adhering to the inflow boundaries $x = 0$ and $y = 1$ were specified similarly to the case of the outflow boundary $x = 4$.

As a criterion for the convergence of pseudo-unsteady difference solution w^n to the stationary limit we checked the inequality $\text{Res}(n) < \varepsilon$, where ε is a user-specified small positive number;

$$\text{Res}(n) = \max_{j,k} \left\{ \max\left(|R^n_{1,j,k}|, |R^n_{2,j,k}|, |R^n_{3,j,k}|, |R^n_{4,j,k}| \right) \right\},$$

where

$$\left\{ R^n_{1,j,k}, R^n_{2,j,k}, R^n_{3,j,k}, R^n_{4,j,k} \right\}^T = \frac{\tilde{f}^n_{j+1/2,k} - \tilde{f}^n_{j-1/2,k}}{h_1} + \frac{\tilde{g}^n_{j,k+1/2} - \tilde{g}^n_{j,k-1/2}}{h_2}$$

in the case of the Chakravarthy–Osher scheme (16); the superscript T denotes the transposition operation.

4.1 Chakravarthy-Osher Scheme

The following scheme proposed by S.R. Chakravarthy and S. Osher [7] belongs to a wide class of the TVD schemes (see also [21, 24]) and can be derived by flux limiter method using the combination of Lax-Wendroff and Beam-Warming methods. As we have shown in Sec. 3 our TRS can be used to perform this task.

Applied to the Euler equations (13), (14) this scheme has the following form:

$$\frac{w^{n+1}_{j,k} - w^n_{j,k}}{\tau} + \frac{\tilde{f}^n_{j+1/2,k} - \tilde{f}^n_{j-1/2,k}}{h_1} + \frac{\tilde{g}^n_{j,k+1/2} - \tilde{g}^n_{j,k-1/2}}{h_2} = 0, \tag{16}$$

where h_1 and h_2 are the steps of a uniform rectangular grid along the x- and y- axes, respectively. The fluxes $\tilde{f}^n_{j+1/2,k}$ and $\tilde{g}^n_{j,k+1/2}$ are computed as follows:

$$\begin{aligned} \tilde{f}^n_{j+1/2,k} &= \frac{1}{2}\left[f(w^n_{j+1,k}) + f(w^n_{j,k}) - \psi(A)_{j+1/2,k} \Delta_{j+1/2,k} w^n_{j,k} \right] - \tilde{W}^n_{j+1/2,k}, \\ \tilde{g}^n_{j,k+1/2} &= \frac{1}{2}\left[g(w^n_{j,k+1}) + g(w^n_{j,k}) - \psi(B)_{j,k+1/2} \Delta_{j,k+1/2} w^n_{j,k} \right] - \tilde{W}^n_{j,k+1/2}, \end{aligned} \tag{17}$$

where

$$\Delta_{j+1/2,k} w^n_{j,k} = w^n_{j+1,k} - w^n_{j,k}, \quad \Delta_{j,k+1/2} w^n_{j,k} = w^n_{j,k+1} - w^n_{j,k},$$

$$\begin{aligned} \psi(A)_{j+1/2,k} &= R^{(1)}_{j+1/2,k} \text{Diag}\left(\psi(\lambda^{(1)}_1, \psi(\lambda^{(1)}_2, \psi(\lambda^{(1)}_3, \psi(\lambda^{(1)}_4)) \right)_{j+1/2,k} \left(R^{(1)}_{j+1/2,k} \right)^{-1}, \\ \psi(B)_{j,k+1/2} &= R^{(2)}_{j,k+1/2} \text{Diag}\left(\psi(\lambda^{(2)}_1), \psi(\lambda^{(2)}_2), \psi(\lambda^{(2)}_3), \psi(\lambda^{(2)}_4) \right)_{j,k+1/2} \left(R^{(2)}_{j,k+1/2} \right)^{-1}. \end{aligned} \tag{18}$$

In each of the expressions (17), the expression $\frac{1}{2}[\ldots]$ ensures the first order of accuracy of the scheme. The terms $\tilde{W}^n_{j+1/2,k}$, $\tilde{W}^n_{j,k+1/2}$ involve the monotonizing corrections ensuring the second or third order of approximation in space.

The matrices A and B entering (17) and (18) are the Jacobi matrices corresponding to the flux vectors $f(w)$ and $g(w)$ (14), that is

$$A(w) = \frac{\partial f(w)}{\partial w}, \qquad B(w) = \frac{\partial g(w)}{\partial w}.$$

The matrices $R^{(m)}$ and $(R^{(m)})^{-1}$, $m = 1, 2$, which enter formulas (18), are taken from the decompositions

$$A = R^{(1)} D_1 \left(R^{(1)} \right)^{-1}, \quad B = R^{(2)} D_2 \left(R^{(2)} \right)^{-1},$$

where

$$
\begin{gathered}
D_m = \text{Diag}\left(\lambda_1^{(m)}, \lambda_2^{(m)}, \lambda_3^{(m)}, \lambda_4^{(m)}\right), \quad m = 1,2, \\
\lambda_1^{(1)} = u - c,\ \lambda_2^{(1)} = u,\ \lambda_3^{(1)} = u + c,\ \lambda_4^{(1)} = u; \\
\lambda_1^{(2)} = v - c,\ \lambda_2^{(2)} = v,\ \lambda_3^{(2)} = v + c,\ \lambda_4^{(2)} = v.
\end{gathered} \tag{19}
$$

The explicit expressions for the matrices $R^{(m)}$ and $(R^{(m)})^{-1}$, $m = 1, 2$, for the case of the ordering of eigenvalues $\lambda_k^{(m)}$ in accordance with (19) may be found in [25], therefore, we do not present these expressions here for the purpose of brevity.

The function $\psi(z)$ entering (18) is defined as [25, 21]

$$
\psi(z) = \begin{cases} |z|, & |z| \geq \delta \\ \dfrac{z^2 + \delta^2}{2\delta}, & |z| < \delta, \end{cases} \tag{20}
$$

where δ is a positive user-specified constant taken in the interval $[0.01, 0.25]$.

We now present the expressions for the corrections $\tilde{W}^n_{j+1/2,k}$. The formulas for $\tilde{W}^n_{j,k+1/2}$ are similar, therefore, we do not present them for the purpose of brevity.

$$
\tilde{W}^n_{j+1/2,k} = \frac{1-\phi}{4}\left[\Delta\tilde{f}^-_{j+3/2,k} - \Delta\bar{f}^+_{j-1/2,k}\right] + \frac{(1+\phi)}{4}\left[\Delta\bar{f}^-_{j+1/2,k} - \Delta\tilde{f}^+_{j+1/2}\right], \tag{21}
$$

where

$$
\begin{aligned}
\Delta\tilde{f}^-_{j+3/2,k} &= R^{(1)}_{j+3/2,k}\text{minmod}\left[\sigma^-_{j+3/2,k}, \beta\sigma^-_{j+1/2,k}\right], \\
\Delta\bar{f}^+_{j-1/2,k} &= R^{(1)}_{j-1/2,k}\text{minmod}\left[\sigma^+_{j-1/2,k}, \beta\sigma^+_{j+1/2,k}\right], \\
\Delta\bar{f}^-_{j+1/2,k} &= R^{(1)}_{j+1/2,k}\text{minmod}\left[\sigma^-_{j+1/2,k}, \beta\sigma^-_{j+3/2,k}\right], \\
\Delta\tilde{f}^+_{j+1/2,k} &= R^{(1)}_{j+1/2,k}\text{minmod}\left[\sigma^+_{j+1/2,k}, \beta\sigma^+_{j-1/2,k}\right], \\
\sigma^{\pm}_{j+1/2,k} &= \frac{1}{2}(D_1 \pm |D_1|)_{j+1/2,k}(R^{(1)}_{j+1/2,k})^{-1}\Delta_{j+1/2,k}w^n_{j,k}.
\end{aligned} \tag{22}
$$

$$
\begin{aligned}
\frac{1}{2}(D_m \pm |D_m|) = \text{Diag}\Bigg(&\frac{1}{2}(\lambda_1^{(m)} \pm |\lambda_1^{(m)}|), \frac{1}{2}(\lambda_2^{(m)} \pm |\lambda_2^{(m)}|), \frac{1}{2}(\lambda_3^{(m)} \pm |\lambda_3^{(m)}|), \\
&\frac{1}{2}(\lambda_4^{(m)} \pm |\lambda_4^{(m)}|\Bigg), \quad m = 1,2; \\
\text{minmod}(x,y) = \text{sgn}(x)&\cdot\max(0, \min(|x|, \text{sgn}(x)y)).
\end{aligned}
$$

The entries of matrices $R^{(1)}_{j+1/2,k}$, $(R^{(1)}_{j+1/2,k})^{-1}$, etc., were computed by using the cell interface values $\rho_{j+1/2,k}$, $u_{j+1/2,k}$, $v_{j+1/2,k}$, $c_{j+1/2,k}$ averaged in accordance with Roe's approach; the corresponding formulas may be found in [7].

The time step τ entering (16) was specified with regard for the results of [7] by formula

$$
\tau = \frac{4\theta}{5 - \phi + \beta(1+\phi)} \cdot \left[\max\left(\frac{|u| + c}{h_1}, \frac{|v| + c}{h_2}\right)\right]^{-1}, \tag{23}
$$

where θ is the safety factor, $0 < \theta \leq 1$, c is the local speed of sound, $c = (\gamma p/\rho)^{0.5}$, γ is the ratio of the gas specific heats entering (15).

The constant ϕ entering the correction (21) is a user-specified parameter, which regulates the upwindedness: $\phi = 1$ yields central differencing, $\phi = -1$ second-order accurate upwind-biased differencing, $\phi = 1/3$ third-order accurate upwind-biased differencing. The parameter β entering (23) is a "compression" parameter determined in the range given by [7] as

$$
1 < \beta \leq \frac{3 - \phi}{1 - \phi}. \tag{24}
$$

If a larger value of β satisfying (24) is taken then a switch of scheme (16) from a higher order of approximation to the first order occurs in a lesser number of spatial grid nodes.

5 Results

In this section we will present some results obtained by the Osher-Chakravarthy scheme applied to the above described shock reflection problem. Note that in our previous work [16], we have generated a code implementing the Jameson scheme applied to the same problem.

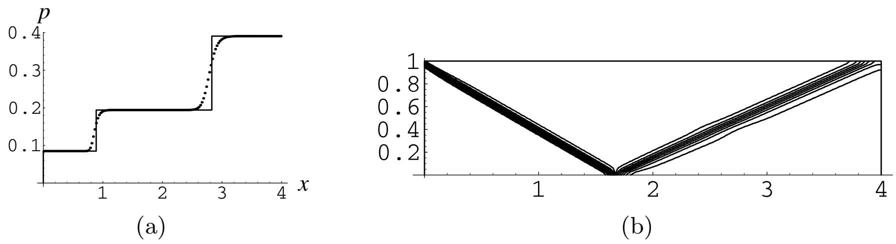

Fig. 5. Shock reflection problem, the Chakravarthy–Osher scheme (16): (a) pressure profile in the section $y = 0.4875$, (—) the exact solution, (○ ○ ○) the numerical solution; (b) predicted Mach number contours

In Fig. 5 we show the numerical results obtained on a mesh of 160×40 cells by the Chakravarthy–Osher scheme. It can be seen that this TVD scheme produces monotone solution profiles. It may be seen in Fig. 5, (b) that two oblique shocks are generated: one of them is the incident shock, and the other is the reflected shock. These shocks are well visible as the subregions, in which different Mach number contours coalesce.

6 Appendix A

In this appendix we will present the implementation of TRS (RTVD1)–(RTVD6) in Maple and demonstrate its application to the model problem described in section 3.

Let us describe the TRS rules as follows:

```
> RTVD1:=F[i+1/2]=F[i+1/2]^L+phi[i+1/2]*(F[i+1/2]^H-F[i+1/2]^L);
> RTVD2:=F[i-1/2]=F[i-1/2]^L+phi[i-1/2]*(F[i-1/2]^H-F[i-1/2]^L);
> RTVD3:=F[i+1/2]^H=a[1]*beta[1]*(-u[n,j]+u[n,j+1]);
> RTVD5:=F[i-1/2]^H=a[1]*beta[0]*(u[n,j]-u[n,j-1]);
> RTVD4:=F[i+1/2]^L=a[1]*alpha[1]*(-u[n,j]+u[n,j+1]);
> RTVD6:=F[i-1/2]^L=a[1]*alpha[0]*(u[n,j]-u[n,j-1]);
```

The first rewriting step depicted in Fig. 3 is then obtained by:

```
> s1:=subs(RTVD1,RTVD2,RTVD3,RTVD4,RTVD5,RTVD6,
     F[i+1/2])-F[i-1/2];
```

$$s1 := a_1\alpha_1\left(-u_{n,j}+u_{n,j+1}\right) + \phi_{i+1/2}\left(a_1\beta_1\left(-u_{n,j}+u_{n,j+1}\right) - a_1\alpha_1\left(-u_{n,j}+u_{n,j+1}\right)\right) - F_{i-1/2}$$

In order to apply the bidirectional rule (RA1) to particular terms underlined in Fig. 3 at the second rewriting step we use the Maple commands `expand` and `collect`:

```
> s3:=collect(expand(s2),u[n,j+1]):
> s4:=collect(s3,u[n,j-1]):
> s5:=collect(s4,u[n,j]);
```

$$s5 := \left(-a_1\alpha_1 - \phi_{i+1/2}a_1\beta_1 + \phi_{i+1/2}a_1\alpha_1\right)u_{n,j} + \left(a_1\alpha_1 + \phi_{i+1/2}a_1\beta_1 - \phi_{i+1/2}a_1\alpha_1\right)u_{n,j+1} - F_{i-1/2}$$

And the similar application of (RTVD1)–(RTVD6), (RA1) but only to $F_{i-1/2}$:

```
> s1b:=subs(RTVD1,RTVD2,RTVD3, RTVD4, RTVD5,RTVD6,
      op(s5)-op(1,s5)-op(2,s5));
> s3b:=collect(s2b,u[n,j+1]):
> s4b:=collect(s3b,u[n,j-1]):
> s5b:=collect(s4b,u[n,j]):
```

As a result we have obtained the algebraic form of flux limiter method corresponding to Harten theorem:

$$\left(-a_1\alpha_0 - \phi_{i-1/2}\left(a_1\beta_0 - a_1\alpha_0\right)\right) u_{n,j} +$$
$$\left(a_1\alpha_0 - \phi_{i-1/2}\left(-a_1\beta_0 + a_1\alpha_0\right)\right) u_{n,j-1} +$$
$$\left(-a_1\alpha_1 - \phi_{i+1/2}a_1\beta_1 + \phi_{i+1/2}a_1\alpha_1\right) u_{n,j} +$$
$$\left(a_1\alpha_1 + \phi_{i+1/2}a_1\beta_1 - \phi_{i+1/2}a_1\alpha_1\right) u_{n,j+1}$$

References

1. Baader, F., Nipkow, T.: Term Rewriting and *All That*. Cambridge University Press (1998)
2. Blinkov, Y.A., Mozshilkin, V.V.: Finite volume method for high order equations. In: *Collection of Papers Dedicated to the 75th Anniversary of the Mechanics Chair of Saratov State University*, Saratov State University (1996) 146–154 (in Russian)
3. Buchberger, B.: Basic features and development of the critical pair completion procedure. In: *Rewriting Techniques and Applications*, J.P. Jouannaud (Ed.), volume 202 of LNCS, Springer Verlag (1986)
4. Buchberger, B.: History and basic features of the critical-pair/completion procedure. *J. Symb. Comp.* **3** (1987) 3–38
5. Buchberger, B., Loos, R.: Algebraic simplification. *Computing*, Supplement 4 (1982) 11–43
6. Caughey, D.A., Hafez, M.M.: A review of Jameson's contributions to computational fluid dynamics. In: *Frontiers in Computational Fluid Dynamics 1994*, D.A. Caughey, M.M. Hafez (Eds.), John Wiley & Sons, Chichester, New York (1995) 1–42
7. Chakravarthy, S.R., Osher, S.: *A New Class of High Accuracy TVD Schemes for Hyperbolic Conservation Laws*, AIAA Paper No. 85-0363 (1985)
8. Chen, P.: The entity-relationship model–toward a unified view of data. *ACM Trans. on Database Systems* **1** (1976) 9 – 36
9. Chibisov, D., Ganzha, V., Zenger, C.: *Objekt-Orientierte Modellierung fuer Wissenschaftliches Rechnen in Maple*. Preprint of the Institute of Informatics. Technical University of Munich (2002, in press)
10. Dadone, A.: Symmetry techniques for the numerical solution of the 2D Euler equations at impermeable boundaries. *Internat. J. for Numer. Methods in Fluids* **28** (1998) 1093–1108
11. Engelen, van R.: CTADEL: *A Generator of Efficient Numerical Code*. Ph.D. Thesis, LIACS, Leiden University (1998)
12. Gallopoulos, E., Houstis, E., and Rice, J.R.: *Future Research Directions in Problem Solving Environments for Computational Science*, Report of a Workshop on Research Directions in Integrating Numerical Analysis, Symbolic Computing, Computational Geometry, and Artificial Intelligence for Computational Science. Technical Report 1259, Center for Supercomputing Research and Development, University of Illinois at Urbana – Champaign (1992)
13. Ganzha, V.G., Vorozhtsov, E.V.: Diagram design and analysis of aerodynamics problems with Mathematica. *Selçuk J. Appl. Math.* **1** (2000) 21–46
14. Ganzha, V.G., Vorozhtsov, E.V.: Stability investigation of Runge-Kutta schemes with artificial dissipator on curvilinear grids for the Euler equations. *Math. and Computers in Simulation* **58** (2001) 1–36
15. Ganzha, V.G., Chibisov, D., Vorozhtsov, E.V.: GROOME – tool supported graphical object oriented modelling for computer algebra and scientific computing. In: *Computer Algebra in Scientific Computing/ CASC 2001*, V.G. Ganzha, E.W. Mayr, E.V. Vorozhtsov (Eds.), Springer-Verlag, Berlin (2001) 213–232
16. Ganzha, V.G., Chibisov, D., Vorozhtsov, E.V.: Problem solving for scientific computing: data modelling instead of algorithms ? *Selçuk J. Appl. Math.* **2** (2001) 53–72
17. Jameson, A., Schmidt, W.: Some recent developments in numerical methods for transonic flows. *Comput. Meth. Appl. Mech. and Engng.* **51** (1985) 467–493
18. Jameson, A., Schmidt, W., Turkel, E.: *Numerical Solution of the Euler Equations by Finite Volume Methods Using Runge Kutta Time Stepping Schemes*. AIAA Paper 81-1259 (1981)
19. Kintsch, W., Greeno, J.G.: Understanding and solving word arithmetic problems. *Psychological Review* **92** (1995) 109–129
20. Laney, C.B.: *Computational Gasdynamics*, Cambridge University Press, Berlin (1998)
21. LeVeque, R.J.: *Numerical Methods for Conservation Laws*, Birkhäuser Verlag, Basel, Boston, Berlin (1992)
22. Loos, R.: Term reduction systems and algebraic algorithms. In: *Proc. 5th German Workshop on Artifical Intelligence, GWAI'81*, volume 47 of *Informatik Fachberichte*, J. Seikmann (Ed.), Springer-Verlag (1981)

23. Steinberg, S., Akers, R.L., Kant, E., Randall, C.J., Young, R.L.: SciNapse: A problem-solving environment for partial differential equations. *IEEE Computational Science and Engineering* **4** (1997) 32–42
24. Toro, E.F.: *Riemann Solvers and Numerical Methods for Fluid Dynamics. A Practical Introduction*, Springer-Verlag, Berlin, New York (1999)
25. Yee, H.C., Warming, R.F.: Implicit total variation diminishing (TVD) schemes for steady-state calculations. *J. Comput. Phys.* **57** (1985) 327–360