BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 59, No. 1, 2011
DOI: 10.2478/v10175-011-0012-3

Varia

Numerical methods of solving equations of hydrodynamics from perspectives of the code FLASH

K. MURAWSKI[1]* and D. LEE[2]

[1] Faculty of Mathematics, Physics and Informatics, UMCS, 10 Radziszewskiego St., 20-031 Lublin, Poland
[2] ASC/Flash Center, The University of Chicago, 5640 S. Ellis Ave, Chicago, IL 60637, USA

Abstract. In this paper we review numerical methods for hydrodynamic equations. Internal complexity make numerical solutions of these equations a formidable task. We present results of advanced numerical simulations for a complex system with a use of a publicly available code, FLASH. These results proof that the numerical methods cope very well with this task.

Key words: numerical methods for hyperbolic equations, finite volume methods, Godunov-type methods.

1. Introduction

A variety of physical phenomena are described by fluid equations. Both soil, water, air and fire on some scales behave as fluids [1]. The mathematical description of fluid motion makes use of partial differential (e.g., the Boltzmann) equations which propagate the fluid variables in time. A simple mathematical tool to model a fluid is hydrodynamics (HD). Like solutions of nonlinear equations perturbations described by HD equations may result in large gradients which are difficult for any treatment. Finite-difference numerical methods are one of several different techniques available to solve the HD equations. They are simple to implement, easily adaptable to complex geometries, and well suited to handle nonlinear terms and large gradients. To represent adequately these gradients most numerical schemes were based on artificial viscosity which was originally introduced by Neumann and Richtmeyer [2], and later used by a number of authors [3, 4]. The use of standard numerical schemes of second-order accuracy or higher (e.g., the Lax-Wendroff method) generates spurious oscillations which destroy monotonicity of the solution. Lower-order schemes [5] are generally free of oscillations, but they are too dissipative to lead to acceptable results. Therefore, a development of more advanced schemes, which would adequately represent the large gradient profiles, was required.

Up to now many numerical schemes were developed for HD equations facing and trying to overcome the above reported problems. The aim of this paper is to review numerical methods for solving HD equations. We pay particular attention to those methods which were recently implemented in the code FLASH [6], described briefly in Sec. 4. HD equations are presented in Sec. 2. Mathematical properties of HD equations are discussed in Sec. 2. Finite-volume numerical methods for Euler equations are discussed in Sec. 3. Results of numerical simulations of waves in a strongly stratified atmosphere are presented in Sec. 4. This paper is completed by summary of the main results.

*e-mail: kmur@kft.umcs.lublin.pl

2. Mathematical properties of hydrodynamic equations

Hydrodynamic systems are described by Navier-Stokes equations which were derived in the first half of the nineteenth century independently by M. Navier and G. Stokes. In Cartesian coordinates, the Navier-Stokes equations for two-dimensional flows (with z being the invariable coordinate) can be written in the conservation form as

$$\begin{pmatrix} \varrho \\ \varrho u \\ \varrho v \\ E \end{pmatrix}_{,\mathrm{t}} + \begin{pmatrix} \varrho u \\ \varrho u^2 - \tau_{\mathrm{xx}} \\ \varrho uv - \tau_{\mathrm{xy}} \\ (E+p)u - u\tau_{\mathrm{xx}} - v\tau_{\mathrm{xy}} + h_{\mathrm{x}} \end{pmatrix}_{,\mathrm{x}} + \begin{pmatrix} \varrho v \\ \varrho vu - \tau_{\mathrm{yx}} \\ \varrho v^2 - \tau_{\mathrm{yy}} \\ (E+p)v - u\tau_{\mathrm{yx}} - v\tau_{\mathrm{yy}} + h_{\mathrm{y}} \end{pmatrix}_{,\mathrm{y}} = \mathbf{S}. \tag{1}$$

Here $\mathbf{S}$ is the source term which may contain external forces like gravity, $\mathbf{V} = [u, v, 0]$ denotes velocity, and E is the total energy density such that the gas pressure p is

$$p = (\gamma - 1)\left[E - \frac{1}{2}\varrho(u^2 + v^2)\right]. \tag{2}$$

The specific heats ratio $\gamma = c_p/c_{\mathrm{v}}$ is such that $\gamma = (m+2)/m$, where m is the number of internal degrees of freedom of the fluid molecules, c_p and c_{v} are the specific heats at constant pressure and volume, respectively. For a monoatomic fluid (diatomic molecules) $m = 3$ ($m = 5$), leading to $\gamma = 5/3$ ($\gamma = 1.4$). Heat transfer by thermal conduction is proportional to the local temperature T gradient

$$h_{\mathrm{x}} = -\kappa T_{,\mathrm{x}}, \qquad h_{\mathrm{y}} = -\kappa T_{,\mathrm{y}} \tag{3}$$

with κ being the thermal conductivity coefficient. Partial derivatives with respect to time t and spatial coordinates x, y are denoted by $_{,\mathrm{t}}$, $_{,\mathrm{x}}$, and $_{,\mathrm{y}}$, respectively.

In the late seventeenth century Isaac Newton discovered that shear-stress τ in a fluid is proportional to velocity gradients. Such fluids are called *Newtonian fluids* in opposite to non-Newtonian fluids (such as blood) for which this dependence does not exist. For Newtonian fluids Stokes showed that the normal ($\tau_{\rm xx}$, $\tau_{\rm yy}$) and shear ($\tau_{\rm xy}$, $\tau_{\rm yx}$) stresses are

$$\tau_{\rm xx}=\lambda\nabla\cdot\mathbf{v}+2\mu u_{,\rm x}, \qquad \tau_{\rm yy}=\lambda\nabla\cdot\mathbf{v}+2\mu v_{,\rm y},$$
$$\tau_{\rm xy}=\mu(u_{,\rm y}+v_{,\rm x}), \qquad \tau_{\rm yx}=\tau_{\rm xy}\,. \tag{4}$$

Here μ is the molecular viscosity coefficient and λ is the bulk viscosity coefficient. The normal stresses are related to the time rate of change of volume of a moving fluid element i.e., $\nabla\cdot\mathbf{v}$, whereas the shear stresses are associated with the time rate of change of the shearing deformation of the fluid element. Stokes made the hypothesis that $\lambda=-2\mu/3$. This relation is frequently used but is not definitely confirmed. Quirk [7] assumed that for gases the viscosity coefficient μ varies according to Sutherland's law,

$$\mu=\mu_0\left(\frac{T}{T_0}\right)^{3/2}\frac{T_0+110}{T+110}\,. \tag{5}$$

Here normal conditions are denoted by the subscript $_0$.

We discuss now the one-dimensional (1D) case by setting $v=0$ and $\partial/\partial y=0$ in Eq. (1). It is convenient to introduce the conservative state vector

$$\mathbf{q}(x,t)=\begin{pmatrix}q_1\\q_2\\q_3\end{pmatrix}=\begin{pmatrix}\varrho(x,t)\\\varrho u(x,t)\\E(x,t)\end{pmatrix} \tag{6}$$

and the flux

$$\mathbf{f}(\mathbf{q})=\begin{pmatrix}\varrho u\\\varrho u^2+p\\u(E+p)\end{pmatrix}= \begin{pmatrix}q_2\\\frac{1}{2}(3-\gamma)\frac{q_2^2}{q_1}+(\gamma-1)q_3\\\frac{q_2}{q_1}\left(\gamma q_3-\frac{\gamma-1}{2}\frac{q_2^2}{q_1}\right)\end{pmatrix}. \tag{7}$$

Then, by dropping all non-ideal ($\tau=h=0$) and source ($\mathbf{S}=0$) terms in Eq. (1) we get Euler equations

$$\mathbf{q}_{,\rm t}+\mathbf{f}(\mathbf{q})_{,\rm x}=0. \tag{8}$$

We rewrite these equations in the quasi-linear form

$$\mathbf{q}_{,\rm t}+\mathbf{A}_{\rm c}\mathbf{q}_{,\rm x}=0, \tag{9}$$

where $\mathbf{A}_{\rm c}$ is the Jacobian matrix

$$\mathbf{A}_{\rm c}=\mathbf{f}_{,\mathbf{q}}= =\begin{pmatrix}0&1&0\\\frac{1}{2}(\gamma-3)\frac{q_2^2}{q_1^2}&(3-\gamma)\frac{q_2}{q_1}&\gamma-1\\-\gamma\frac{q_2q_3}{q_1^2}+(\gamma-1)\frac{q_2^3}{q_1^3}&\gamma\frac{q_3}{q_1}-\frac{3}{2}(\gamma-1)\frac{q_2^2}{q_1^2}&\gamma\frac{q_2}{q_1}\end{pmatrix}= =\begin{pmatrix}0&1&0\\\frac{1}{2}(\gamma-3)u^2&(3-\gamma)u&\gamma-1\\u[(\gamma-1)\frac{u^2}{2}-H]&H-(\gamma-1)u^2&\gamma u\end{pmatrix}. \tag{10}$$

Here

$$H=\frac{E+p}{\varrho} \tag{11}$$

is the total specific enthalpy and E is the total energy density.

We can express Euler equations in its non-conservative form. To do so we specify the non-conservative state vector, $\mathbf{w}$, such as

$$\mathbf{w}=\begin{pmatrix}w_1\\w_2\\w_3\end{pmatrix}=\begin{pmatrix}\varrho\\u\\p\end{pmatrix}. \tag{12}$$

We rewrite Euler equations in their quasi-linear form

$$\mathbf{w}_{,\rm t}+\mathbf{A}_{\rm n}\mathbf{w}_{,\rm x}=0, \tag{13}$$

where the non-conservative Jacobian is

$$\mathbf{A}_{\rm n}=\begin{pmatrix}u&\varrho&0\\0&u&\frac{1}{\varrho}\\0&\varrho c_{\rm s}^2&u\end{pmatrix}.$$

The eigenvalue problem for $\mathbf{A}_{\rm c}$ or $\mathbf{A}_{\rm n}$ is

$$\mathbf{A}_{\rm m}\mathbf{r}^{\rm i}_{\rm m}=\lambda^{\rm i}\mathbf{r}^{\rm i}_{\rm m}, \qquad m=c,n, \qquad i=1,2,3. \tag{14}$$

Here $\mathbf{r}^{\rm i}_{\rm m}$ is the right eigenvector and $\lambda^{\rm i}$ denotes the corresponding eigenvalue. We present the former for the case of the conservative form of Euler equations for which we have the diagonalizable equation

$$\mathbf{R}_{\rm c}^{-1}\mathbf{A}_{\rm c}\mathbf{R}_{\rm c}=\mathbf{\Lambda} \tag{15}$$

with the eigenvector matrix $\mathbf{R}_{\rm c}$ whose columns consist of the right eigenvectors, viz.

$$\mathbf{R}_{\rm c}=(\mathbf{r}^1_{\rm c},\mathbf{r}^2_{\rm c},\mathbf{r}^3_{\rm c})=\begin{pmatrix}1&1&1\\u-c_{\rm s}&u&u+c_{\rm s}\\H-uc_{\rm s}&\frac{u^2}{2}&H+uc_{\rm s}\end{pmatrix}. \tag{16}$$

Here a sound speed $c_{\rm s}=\sqrt{\gamma p/\varrho}$ and the matrix $\mathbf{\Lambda}$ is

$$\mathbf{\Lambda}=\begin{pmatrix}\lambda^1&0&0\\0&\lambda^2&0\\0&0&\lambda^3\end{pmatrix}=\begin{pmatrix}u-c_{\rm s}&0&0\\0&u&0\\0&0&u+c_{\rm s}\end{pmatrix}. \tag{17}$$

The full set of eigenvectors for $\mathbf{A}_{\rm n}$ is

$$\mathbf{R}_{\rm n} = (\mathbf{r}^1_{\rm n}, \mathbf{r}^2_{\rm n}, \mathbf{r}^3_{\rm n}) = \begin{pmatrix} 1 & 1 & 1 \\ -c_{\rm s}/\varrho & 0 & c_{\rm s}/\varrho \\ c_{\rm s}^2 & 0 & c_{\rm s}^2 \end{pmatrix}. \tag{18}$$

Note that the right eigenvectors of $\mathbf{A}_{\rm c}$ and $\mathbf{A}_{\rm n}$ differs.

The eigenvalues of $\mathbf{A}_{\rm c}$ and $\mathbf{A}_{\rm n}$ are identical and they are

$$\lambda^1(\mathbf{q}) = \frac{q_2}{q_1} - c_{\rm s},$$

$$\lambda^2(\mathbf{q}) = \frac{q_2}{q_1}, \tag{19}$$

$$\lambda^3(\mathbf{q}) = \frac{q_2}{q_1} + c_{\rm s}.$$

As these eigenvalues are real and they correspond to a set of linearly independent eigenvectors $\mathbf{r}^1_{\rm m}, \mathbf{r}^2_{\rm m}, \mathbf{r}^3_{\rm m}$, $m = c, n$, we conclude that Euler equations are hyperbolic. The above eigenvalues are associated with the fact that information from any point in the flow propagates according to the characteristic equations

$$\frac{dx}{dt} = u, \tag{20}$$

$$\frac{dx}{dt} = u \pm c_{\rm s}. \tag{21}$$

We define now the entropy $s(x,t)$ as

$$s = c_{\rm v} \ln\left(\frac{p}{\varrho^\gamma}\right) + s_0, \tag{22}$$

where s_0 is a constant. Then, Eq. (20) defines a trajectory $x = x(t)$, along which the entropy s is constant. As a result, evolution of the entropy is governed by the advection equation

$$s_{,\rm t} + u s_{,\rm x} = 0. \tag{23}$$

Following Toro [8] we define now *generalized Riemann invariants* as relations which are satisfied across the wave,

$$\frac{dw_1}{r^{\rm i}_{\rm n1}} = \frac{dw_2}{r^{\rm i}_{\rm n2}} = \frac{dw_3}{r^{\rm i}_{\rm n3}}, \qquad i = 1, 2, 3. \tag{24}$$

Here $r^{\rm i}_{\rm nj}$, $j = 1, 2, 3$, is the j-th component of the i-th right eigenvector of the Jacobian $\mathbf{A}_{\rm n}$.

Equation (21) defines then the Riemann invariants

$$R_\pm = u \pm \int \frac{\gamma dp}{\varrho c_{\rm s}}, \tag{25}$$

which are constant along the trajectories $x = x(t)$ defined by Eq. (21). These trajectories follow forward and backward sound waves in the frame moving with the speed u.

We find that

$$\frac{\partial \lambda^2}{\partial \mathbf{q}} = \lambda^2{}_{,\mathbf{q}} = \left(-\frac{q_2}{q_1^2}, \frac{1}{q_1}, 0\right). \tag{26}$$

As

$$\lambda^2{}_{,\mathbf{q}} \cdot \mathbf{r}^2_{\rm c} = 0, \tag{27}$$

we infer that the right eigenvector $\mathbf{r}^2_{\rm c}$ is linearly degenerate. As a result, with $\mathbf{r}^2_{\rm c}$ is not associated neither shocks nor rarefaction waves but contact discontinuities across which there is a jump in the mass density but the gas pressure and flow velocity are **continuous** [8]. The other eigenvalues, $\mathbf{r}^1_{\rm c}$ and $\mathbf{r}^3_{\rm c}$, might be either shocks or rarefaction waves. The scenario depends on a choice of the left, $\mathbf{q}_{\rm l}$, and right, $\mathbf{q}_{\rm r}$, states in a discontinuous representation of $\mathbf{q}$. For the shock wave (rarefaction wave) all of the state variables are discontinuous (continuous).

2.1. Rankine-Hugoniot condition. We consider now a discontinuity in a solution of Eq. (8) which propagates with the speed c. This speed depends on the jump in the solution $\mathbf{q}(x,t)$ across the discontinuity. Suppose a discontinuity moves from left to right; at time $t = t_1$ ($t = t_2$) the discontinuity is at the spatial position $x = x_1$ ($x = x_2$), where $x_1 < x_2$ and $t_1 < t_2$. Let the values of $\mathbf{q}$ be given as $\mathbf{q}_{\rm l}$ on the left hand side of the discontinuity and as $\mathbf{q}_{\rm r}$ on the right hand side of it. We integrate Eq. (8) to get

$$(\mathbf{q}_{\rm l} - \mathbf{q}_{\rm r})(x_2 - x_1) = [\mathbf{f}(\mathbf{q}_{\rm l}) - \mathbf{f}(\mathbf{q}_{\rm r})](t_2 - t_1). \tag{28}$$

In the limits $x_2 \to x_1$ and $t_2 \to t_1$ we have

$$c[\mathbf{q}] = [\mathbf{f}] \tag{29}$$

with $c = (x_2 - x_1)/(t_2 - t_1)$ and

$$[\mathbf{q}] = \mathbf{q}_{\rm r} - \mathbf{q}_{\rm l}, \qquad [\mathbf{f}] = \mathbf{f}(\mathbf{q}_{\rm r}) - \mathbf{f}(\mathbf{q}_{\rm l}). \tag{30}$$

Equation (29) is called the *Rankine-Hugoniot* jump condition which can be applied to shocks and contact discontinuities.

For a scalar equation we get

$$c = \frac{f(q_{\rm r}) - f(q_{\rm l})}{q_{\rm r} - q_{\rm l}}. \tag{31}$$

Hence we infer that any discontinuity between $q_{\rm l}$ and $q_{\rm r}$ is allowed, and it propagates with the speed c that is expressed by the above formula. For the case of a system of equations, only selected jumps in $\mathbf{q}$ are allowed, namely those for which vectors $[\mathbf{q}]$ and $[\mathbf{f}]$ are parallel one to each other. For a system of linear equations with the flux $\mathbf{f}(\mathbf{q}) = \mathbf{A}\mathbf{q}$, from Eq. (29) we obtain

$$c[\mathbf{q}] = \mathbf{A}[\mathbf{q}]. \tag{32}$$

Hence we conclude that only jumps are allowed, for which $[\mathbf{q}]$ and c are respectively an eigenvector and the associated eigenvalue of the matrix $\mathbf{A}$.

2.2. Riemann problem for Euler equations. The Riemann problem corresponds to the Cauchy's problem with a piecewise constant initial data

$$\mathbf{q}(x, t = 0) = \begin{cases} \mathbf{q}_{\rm l} & \text{for } x < 0, \\ \mathbf{q}_{\rm r} & \text{for } x > 0. \end{cases} \tag{33}$$

Here, $\mathbf{q}_{\rm l}$ and $\mathbf{q}_{\rm r}$ are constants which denote the left and right states, respectively. Such two adjacent arbitrary states will evolve into a set of left- and right-going shocks, rarefactions and an intermediate wave.

For Euler equations we can always solve exactly or approximately the Riemann problem [8]. The solution consists of waves traveling with finite velocities. These waves may either be discontinuous shock waves or smooth rarefaction

waves and a contact wave laying between these waves. The procedure for constructing the solution of a Riemann problem is called a Riemann solver. In the Riemann problem, we look for states $\mathbf{q}_{\rm l}^*$ and $\mathbf{q}_{\rm r}^*$ and speeds $c_1 < c_2 < c_3$ such that the Rankine-Hugoniot condition of Eq. (29) is satisfied

$$c_1(\mathbf{q}_{\rm l}^* - \mathbf{q}_{\rm l}) = \mathbf{f}(\mathbf{q}_{\rm l}^*) - \mathbf{f}(\mathbf{q}_{\rm l}), \tag{34}$$

$$c_2(\mathbf{q}_{\rm r}^* - \mathbf{q}_{\rm l}^*) = \mathbf{f}(\mathbf{q}_{\rm r}^*) - \mathbf{f}(\mathbf{q}_{\rm l}^*), \tag{35}$$

$$c_3(\mathbf{q}_{\rm r} - \mathbf{q}_{\rm r}^*) = \mathbf{f}(\mathbf{q}_{\rm r}) - \mathbf{f}(\mathbf{q}_{\rm r}^*). \tag{36}$$

According to LeVeque [9] we have $c_2 = u(\mathbf{q}_{\rm l}^*) = u(\mathbf{q}_{\rm r}^*)$, where u is the flow speed. As a result of that the intermediate wave, which is termed a *contact wave*, propagates with the flow speed $\lambda_2 = u$. Moreover, as $p(\mathbf{q}_{\rm l}^*) = p(\mathbf{q}_{\rm r}^*)$ the pressure is continuous across this wave, while mass density experiences a jump there, hence termed a 'contact wave'. Note that across shocks all dependent variables change discontinuously. The left most wave is a shock if the entropy condition is satisfied $\lambda_1(\mathbf{q}_{\rm l}) > c_1 > \lambda_1(\mathbf{q}_{\rm l}^*)$. Otherwise, this jump would be a rarefaction wave. Analogously, the right most wave is a shock if $\lambda_3(\mathbf{q}_{\rm r}^*) > c_3 > \lambda_3(\mathbf{q}_{\rm r})$. Otherwise, this wave would be a rarefaction wave.

We display schematically in the x- and t-coordinates a solution of the Riemann problem in Fig. 1. Here the left wave corresponds to a rarefaction wave, the intermediate wave is a contact wave and the right wave in a shock. In general, the left and right waves are combination of shocks and rarefaction waves, while the middle wave is always a contact wave. We can obtain a general solution involving these waves from a nonlinear algebraic equation [8]. This equation can be solved by any iterative method. As this is clearly a computationally expensive task, approximate methods are often required. These methods are described in the following part of the paper.

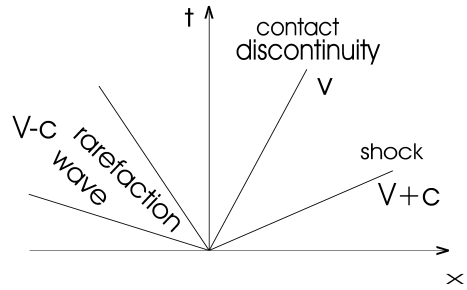

Fig. 1. Riemann fan for the one-dimensional Euler equations

HLL solver. Harten, Lax, and van Leer [10] devised a simple approximate Riemann solver which is commonly called HLL. In this solver the contact wave is removed from the system and the solution is approximated by two waves which propagate with their speeds c^- and c^+ such that they correspond to the minimum and maximum characteristic speeds of the system [11]. Strengths of these waves are

$$\mathbf{\Delta q}^1 = \mathbf{q}_* - \mathbf{q}_{\rm l}, \qquad \mathbf{\Delta q}^2 = \mathbf{q}_{\rm r} - \mathbf{q}_*. \tag{37}$$

We choose the middle state, $\mathbf{q}_*$, to satisfy the conservation constraint,

$$(c^+ - c^-)\mathbf{q}_* = c^+\mathbf{q}_{\rm r} - c^-\mathbf{q}_{\rm l} - (\mathbf{f}(\mathbf{q}_{\rm r}) - \mathbf{f}(\mathbf{q}_{\rm l})). \tag{38}$$

Hence we obtain the intermediate state

$$\mathbf{q}_* = \frac{1}{c^+ - c^-}[c^+\mathbf{q}_{\rm r} - c^-\mathbf{q}_{\rm l} - (\mathbf{f}(\mathbf{q}_{\rm r}) - \mathbf{f}(\mathbf{q}_{\rm l}))]. \tag{39}$$

As the Euler equations evolve three distinctive waves, with its speeds given by Eq. (19), obviously the HLL Riemann solver suffers from a drawback being the solver diffusive and a more appropriate solver is required. Although the HLL solver is known to be very diffusive, it provides robustness of numerical solutions as the solver always maintains positive states of density and pressures, especially in strong rarefaction regions. A solver which includes the intermediate wave is called HLLC. The details of this solver can be found in Toro [8]. In the following part of the paper we describe a very popular solver which was developed by Roe [12].

Roe solver. In the Riemann problem of Roe [12], we can use the quasi-linear system of Eq. (9). Roe [12] proposed to introduce an average Jacobian $\overline{\mathbf{A}}$, which approximates the Jacobian $\mathbf{A} = \mathbf{f}_{,\mathbf{q}}$. The average Jacobian (called also the Roe matrix) is such that for any given left and right pair of states $(\mathbf{q}_{\rm l}, \mathbf{q}_{\rm r})$ the U property is satisfied:

(i) $\overline{\mathbf{A}}$ is a linear mapping from the vector space $\mathbf{q}$ to the vector space $\mathbf{f}$;
(ii) $\overline{\mathbf{A}}(\mathbf{q}_{\rm l}, \mathbf{q}_{\rm r}) \to \mathbf{f}_{,\mathbf{q}}$ as $\mathbf{q}_{\rm l}$ and $\mathbf{q}_{\rm r} \to \mathbf{q}$;
(iii) $\overline{\mathbf{A}}(\mathbf{q}_{\rm l}, \mathbf{q}_{\rm r})$ has real eigenvalues and a complete set of linearly independent eigenvectors;
(iv) $\overline{\mathbf{A}}(\mathbf{q}_{\rm r} - \mathbf{q}_{\rm l}) = \mathbf{f}_{\rm r} - \mathbf{f}_{\rm l}$ for any $\mathbf{q}_{\rm l}$ and $\mathbf{q}_{\rm r}$.

In the original Roe scheme, the average state $\overline{\mathbf{q}}$ that was used to linearize the problem, is not $(\mathbf{q}_{\rm l} + \mathbf{q}_{\rm r})/2$. Instead, it is taken such as the property (iv) is satisfied. Such averaging leads to mass density as

$$\overline{\varrho} = \sqrt{\varrho_{\rm l}\varrho_{\rm r}}. \tag{40}$$

The remnant variables ($\mathbf{V}$, E), which are symbolically denoted here by ϕ, are averaged as [13]

$$\overline{\phi} = \frac{\sqrt{\varrho_{\rm l}}\phi_{\rm l} + \sqrt{\varrho_{\rm r}}\phi_{\rm r}}{\sqrt{\varrho_{\rm l}} + \sqrt{\varrho_{\rm r}}}. \tag{41}$$

Once all the averaged variables are obtained, the linearized Riemann problem is considered at each interface. The exact solution of this approximate problem can be expressed in terms of right eigenvector $\mathbf{r}^{\rm m}$ of $\overline{\mathbf{A}}$ as

$$[\mathbf{q}] = \sum_{\rm m=1}^{3} \alpha^{\rm m}\mathbf{r}^{\rm m}. \tag{42}$$

We determine the coefficient $\alpha^{\rm m}$ by multiplying the above equation by each left eigenvector $\mathbf{l}^{\rm j}$. With a use of

$$\mathbf{l}^{\rm j}\mathbf{r}^{\rm m} = \delta_{\rm jm}, \tag{43}$$

we get

$$\alpha^{\rm m} = \mathbf{l}^{\rm m}[\mathbf{q}]. \tag{44}$$

According to property (iv), with a use of Eq. (42) we get the flux increment, $[\mathbf{f}]$, expressed as a product of $[\mathbf{q}]$ and the corresponding eigenvalues $\lambda^{\rm m}$, *viz.*

$$[\mathbf{f}] = \sum_{\rm m=1}^{q} \alpha^{\rm m}\lambda^{\rm m}\mathbf{r}^{\rm m}. \tag{45}$$

Einfeld *et al.* [14] and Donat and Marquina [15] have pointed out that linearized Riemann solvers may fail in some extreme situations such as, for instance, in the case of strong rarefactions or shocks. Then mass densities or pressures can occasionally acquire negative values.

A relaxation solver. Jin and Xin [16] in their relaxation scheme replaced Eq. (8) by the coupled equations,

$$\mathbf{q}_{,\mathrm{t}} + \mathbf{v}_{,\mathrm{x}} = 0, \tag{46}$$

$$\mathbf{v}_{,\mathrm{t}} + \mathbf{B}^2 \mathbf{q}_{,\mathrm{x}} = \frac{1}{\tau}(\mathbf{f}(\mathbf{q}) - \mathbf{v}). \tag{47}$$

Here $\mathbf{q}$, $\mathbf{v} \in R^q$ and $\mathbf{B}^2 \in R^{q\times q}$ is a positive definite matrix. In the original scheme this matrix was chosen to be a diagonal matrix with positive diagonal elements [16]. The relaxation time is denoted by $\tau > 0$.

We consider now the condition

$$|\lambda| \le b_{\max}, \tag{48}$$

where λ is an eigenvalue of the Jacobian matrix $\mathbf{f}_{,\mathbf{q}}$ and $b_{\max} = max_{\mathrm{m}}\{b^{\mathrm{m}}\}$ is the spectral radius of $\mathbf{B}$ which corresponds to positive eigenvalues of $\mathbf{B}$,

$$b^{\mathrm{m}} > 0, \qquad m = 1, 2, 3. \tag{49}$$

Now, as $\tau \to 0$ from Eq. (47) we get

$$\mathbf{v} \to \mathbf{f}(\mathbf{q}) \tag{50}$$

if Eq. (48) is satisfied.

We advance Eqs. (46) and (47) with a use of a fractional step method [17]. First, we advance over time step Δt the homogenous equations

$$\mathbf{q}_{,\mathrm{t}} + \mathbf{v}_{,\mathrm{x}} = 0, \tag{51}$$

$$\mathbf{v}_{,\mathrm{t}} + \mathbf{B}^2 \mathbf{q}_{,\mathrm{x}} = 0. \tag{52}$$

This step leads to $\mathbf{q}^*$ and $\mathbf{v}^*$ which we update to $\mathbf{q}^{n+1}$ and $\mathbf{v}^{n+1}$ by solving

$$\mathbf{q}_{,\mathrm{t}} = 0, \tag{53}$$

$$\mathbf{v}_{,\mathrm{t}} = \frac{1}{\tau}(\mathbf{f}(\mathbf{q}) - \mathbf{v}). \tag{54}$$

From Eq. (53) we have

$$\mathbf{q}^{n+1} = \mathbf{q}^*. \tag{55}$$

We can treat Eq. (54) implicitly

$$\mathbf{v}^{n+1} = \mathbf{f}(\mathbf{q}^{n+1}) + e^{-\Delta t/\tau}[\mathbf{v}^* - \mathbf{f}(\mathbf{q}^{n+1})]. \tag{56}$$

In the limit of $\tau \to 0$ we simplify this expression to

$$\mathbf{v}^{n+1} = \mathbf{f}(\mathbf{q}^{n+1}). \tag{57}$$

In summary, the relaxation scheme consists of solving Eq. (51) with the use of Eq. (57). An approximate Riemann solver is defined then as follows. Given values $\mathbf{q}_{\mathrm{l}}$ and $\mathbf{q}_{\mathrm{r}}$ we compute $\mathbf{v}_{\mathrm{l}} = \mathbf{f}(\mathbf{q}_{\mathrm{l}})$ and $\mathbf{v}_{\mathrm{r}} = \mathbf{f}(\mathbf{q}_{\mathrm{r}})$ and then solve the Riemann problem for Eq. (46) with the data [18]

$$\begin{cases} \mathbf{q}_{\mathrm{l}}\,, \\ \mathbf{f}(\mathbf{q}_{\mathrm{l}})\,, \end{cases} \qquad \begin{cases} \mathbf{q}_{\mathrm{r}}\,, \\ \mathbf{f}(\mathbf{q}_{\mathrm{r}})\,. \end{cases} \tag{58}$$

3. Finite-volume numerical methods

For physically motivated equations, it is important to insure that a numerical method satisfies conservation of the physical quantities such as mass, momentum, and energy. Therefore, it is often advisable to use a finite-volume method in which $\mathbf{q}_{\mathrm{i}}^n$ is considered as an approximation to the average value of $\mathbf{q}(\mathbf{x}, t)$ over a grid cell rather than a pointwise value of $\mathbf{q}$. This average value is the integral of $\mathbf{q}$ over the cell divided by its volume. A finite-volume scheme is based on the discrete equations which are constructed by expressing the integral conservation law on a discrete set of control volumes. The spatial derivative terms are expressed as a surface integral of fluxes which are approximated with a use of solutions at two adjacent finite-volumes. We approximate the state vector $\mathbf{q}$ by its average value over the i-th numerical cell at a discrete time $t_n = n\Delta t$ as, for the case of 1D,

$$\overline{\mathbf{q}}_{\mathrm{i}}^n = \frac{1}{\Delta x_{\mathrm{i}}} \int\limits_{\mathrm{x_{i-1/2}}}^{x_{\mathrm{i+1/2}}} \mathbf{q}(x, t_n)\, dx. \tag{59}$$

Here $\Delta x_{\mathrm{i}} = x_{\mathrm{i+1/2}} - x_{\mathrm{i-1/2}}$, the index i corresponds to the cell center and $i - 1/2$ ($i + 1/2$) to its left (right) interface.

We consider Eq. (8). The integral form of this equation, applied to the numerical cell of index i, over a single time-step dt is

$$\int\limits_{\mathrm{x_{i-1/2}}}^{x_{\mathrm{i-1/2}}+\Delta x_{\mathrm{i}}} [\mathbf{q}(x, t_{\mathrm{n+1}}) - \mathbf{q}(x, t_n)]dx = \int\limits_{\mathrm{t_n}}^{t_{\mathrm{n+1}}} [\mathbf{f}(\mathbf{q}(x_{\mathrm{i-1/2}}, t)) - \mathbf{f}(\mathbf{q}(x_{\mathrm{i-1/2}} + \Delta x_{\mathrm{i}}, t))]dt. \tag{60}$$

Hence, dividing by Δx_{i} and using Eq. (59) we get

$$\overline{\mathbf{q}}_{\mathrm{i}}^{n+1} = \overline{\mathbf{q}}_{\mathrm{i}}^n - \Delta t \frac{\overline{\mathbf{f}}_{\mathrm{i+1/2}}^n - \overline{\mathbf{f}}_{\mathrm{i-1/2}}^n}{\Delta x_{\mathrm{i}}}. \tag{61}$$

Here $\overline{\mathbf{f}}_{\mathrm{i\pm1/2}}^n$ is an approximation to the average flux

$$\overline{\mathbf{f}}_{\mathrm{i\pm1/2}}^n = \frac{1}{\Delta t} \int\limits_{\mathrm{t_n}}^{t_{\mathrm{n+1}}} \mathbf{f}(q(x_{\mathrm{i\pm1/2}}, t))dt. \tag{62}$$

From the above discussion it follows that finite-volume methods mimic conservational properties of the evolution equations. Henceforth these methods are reviewed in this paper. For simplicity reasons, we use q_{i}^n instead of $\overline{q}_{\mathrm{i}}^n$.

3.1. Von Neumann stability analysis. In the numerical schemes, there are typically some restrictions on the time-step Δt and the spatial grid size Δx. We explain this for the advection equation

$$q_{,\mathrm{t}} + cq_{,\mathrm{x}} = 0. \tag{63}$$

Although this equations consists an oversimplified version of the Euler equations the idea of studying stability of a numerical scheme remains same for the latter. We discretize the

advection equation by a numerical scheme with a use of a uniform grid. Temporal and spatial derivatives are approximated by a forward (in time) Euler scheme,

$$q_{,\mathrm{t}}|_{\mathrm{i,n}} = \frac{q_{\mathrm{i}}^{n+1} - q_{\mathrm{i}}^{n}}{\Delta t} + \mathcal{O}(\Delta t), \tag{64}$$

and a centered (in space) Euler scheme

$$q_{,\mathrm{x}}|_{\mathrm{i,n}} = \frac{q_{\mathrm{i+1}}^{n} - q_{\mathrm{i-1}}^{n}}{2\Delta x} + \mathcal{O}(\Delta x^2). \tag{65}$$

Substituting (64) and (65) into (63), we obtain a forward time, centered space discrete representation

$$\frac{q_{\mathrm{i}}^{n+1} - q_{\mathrm{i}}^{n}}{\Delta t} + c\frac{q_{\mathrm{i+1}}^{n} - q_{\mathrm{i-1}}^{n}}{2\Delta x} = 0. \tag{66}$$

To check whether this scheme is stable we introduce a Fourier mode

$$q_{\mathrm{i}}^{n} = A^n e^{jki\Delta x}, \tag{67}$$

where $j^2 = -1$, k is a wave vector and $A = A(k)$ is called the *amplification factor* which in general is a complex function of wavenumber k. Note that a difference equation is called stable in the *von Neumann* sense if

$$|A(k)| \leq 1 \tag{68}$$

for some k [2]. To find $A(k)$ we substitute (67) into (66). Dividing by A^n, we get

$$A(k) = 1 - j\frac{c\Delta t}{\Delta x}\sin(k\Delta x). \tag{69}$$

Hence, we get that for all k

$$|A(k)| > 1. \tag{70}$$

As there is no such Δt which would satisfy stability condition (68) we infer that the discretization of Eq. (66) is unconditionally unstable and therefore it is useless in practice. Stable schemes can be easily constructed. In the following part of the paper we present examples of such stable schemes.

3.2. Godunov-type numerical schemes. In the original Godunov [5] approach, the upwind finite-volume flow solver was implemented and the solution was considered to be piecewise constant over each grid cell at a fixed time, *i. e.*

$$q(x) = q_{\mathrm{i}}, \qquad x_{\mathrm{i-1/2}} < x < x_{\mathrm{i+1/2}}. \tag{71}$$

As a result, discontinuities are placed at cell interfaces $x_{\mathrm{i}\pm 1/2}$. Fluid evolves to the next time-step through wave interactions, which originate at adjacent cell boundaries and specify Riemann problem. As such approach mimics much of the relevant physics, Godunov schemes suppose to result in accurate and well-behaved treatment of shock waves. However, as the Godunov method is first-order accurate it exhibits strong numerical dissipation, and discontinuities in the solution are considerably smeared over several grid zones.

The low accuracy and the complexity of the Godunov method meant that other methods needed to be developed. Such development effort was undertaken by Kolgan [19] who proposed to suppress spurious oscillations and produced in this way a non-oscillatory Godunov-type scheme of second order spatial accuracy. Further, more well-known, developments are due to van Leer [20] who extended to second-order spatial accuracy by the MUSCL approach. His approach consists of two key steps: (a) an interpolation (projection or reconstruction) step where, within each cell, the data is approximated by linear functions; (b) an upwind step where the average fluxes at each interface are evaluated by taking into account the wind direction. A great deal of effort was spent to enhance the accuracy of the interpolation step, and to improve the efficiency and robustness of the upwind step [12, 21]. Accurate interpolations are derived by assuming that the data is smooth. However, in the presence of a shock, these interpolations lead to oscillations which can be prevented by an introduction of a monotonicity constraint for a numerical scheme [20]. In this scheme the accuracy was increased by constructing a piecewise linear approximation of $q(x,t)$ at the beginning of each time-step, *viz.*

$$\overline{q}(x,t) = q_{\mathrm{i}} + s_{\mathrm{i}}(x - \overline{x}_{\mathrm{i}}), \qquad x_{\mathrm{i-1/2}} < x < x_{\mathrm{i+1/2}}. \tag{72}$$

Here s_{i} is a slope and $\overline{x}_{\mathrm{i}} = (x_{\mathrm{i}} + x_{\mathrm{i+1}})/2 = x_{\mathrm{i}} + \Delta x/2$ is the center of the grid cell. So, $\overline{q}(\overline{x}_{\mathrm{i}}, t) = q_{\mathrm{i}}$. Moreover, it is required that the average value of $\overline{q}(x,t)$ over the cell is equal to q_{i}. The slope s_{i} can be constructed by many ways such as

$$s_{\mathrm{i}} = \frac{q_{\mathrm{i+1}} - q_{\mathrm{i-1}}}{2\Delta x} \text{ (centered slope, Fromm's scheme)}, \tag{73}$$

$$s_{\mathrm{i}} = \frac{q_{\mathrm{i}} - q_{\mathrm{i-1}}}{\Delta x} \text{ (upwind slope, Beam-Warming scheme)}, \tag{74}$$

$$s_{\mathrm{i}} = \frac{q_{\mathrm{i+1}} - q_{\mathrm{i}}}{\Delta x} \text{ (downwind slope, Lax-Wendroff scheme)}, \tag{75}$$

$$s_{\mathrm{i}} = \mathrm{minmod}\left(\frac{q_{\mathrm{i}} - q_{\mathrm{i-1}}}{\Delta x}, \frac{q_{\mathrm{i+1}} - q_{\mathrm{i}}}{\Delta x}\right) \text{ (minmod slope)}. \tag{76}$$

Here the minmod function,

$$\mathrm{minmod}(a,b) = \begin{cases} a & \text{for } |a| < |b| \text{ and } ab > 0, \\ b & \text{for } |a| > |b| \text{ and } ab > 0, \\ c & \text{for } ab \leq 0, \end{cases} \tag{77}$$

returns the smallest argument in magnitude if the arguments are of the same sign, and zero if they are not. Note that choosing $s_{\mathrm{i}} = 0$ in the above expressions leads to the Godunov method.

A PPM scheme is a further extension of the MUSCL scheme. The key difference from MUSCL is that $\mathbf{q}$ is allowed to be piecewise parabolic within a cell, rather than piecewise linear. Second-order accuracy in time is again achieved in the same way as in MUSCL, via characteristic tracing and solving Riemann problems.

Essentially non-oscillatory (ENO) schemes are again extensions of the basic Godunov approach. Arbitrarily high-order polynomials are allowed to define $\mathbf{q}$ within a cell, yielding arbitrarily high-order spatial accuracy [22].

Various shock-capturing schemes were compared by Woodward and Colella [23] by computing a blast wave interaction problem in one dimension. The result of that test was an ordering of the schemes in terms of the accuracy.

With the most accurate schemes listed first, that ordering was as follows: PPM, MUSCL, and the Godunov scheme.

Godunov-type schemes are very robust and give reliable results for a wide range of problems without needing to be retuned. However, even these modern schemes are far from being perfect. There are few instances in which a particular scheme produces inappropriate results [14, 15]. For instance, most Godunov-type schemes lead to the generation of a long wavelength noise, downstream nearly stationary shock. This noise is not effectively damped by the dissipation of the scheme [15]. In few cases Roe solvers exhibit nonlinear instability, producing unphysical local features which are called carbuncles. These features become more pronounced for finer grid. In multi-dimensions a problem occurs if a wave is far from aligned with the grid. Then, a grid-oblique wave may be represented by grid-aligned waves, enhancing numerical dissipation and leading to a loss of resolution.

Another deficiency of a Godunov-type scheme is that while computing rarefaction waves, the scheme can produce nonphysical expansion shocks in the computed flow. In this case the true Riemann solution contains a transonic wave with characteristic speeds that increase from negative to positive values through the rarefaction fan. Then, the eigenvalue of the average Jacobian $\overline{\mathbf{A}}$ is such that $\lambda_{\rm l} < 0$ to the left of the wave while $\lambda_{\rm r} > 0$ to the right of the wave. It leads to information traveled partly to the left and partly to the right, affecting cell averages on both sides. The Roe solver approximates every wave by a single discontinuity that propagates at a speed given by an eigenvalue $\overline{c}$ of $\overline{\mathbf{A}}$. In the transonic rarefaction case this speed is approximately zero and the proper spreading does not occur. This can lead to numerical approximations with entropy violating discontinuities.

Several ways to fix the problem of prevention Roe method from admitting expansion shocks exist in the literature. For instance, we can replace values of the numerical viscosity μ smaller than some tolerance ε with higher values μ' such that

$$\mu' = \begin{cases} \mu & \text{for } |\mu| \geq \varepsilon, \\ \dfrac{\mu}{2}\left(\dfrac{2\mu^2}{\varepsilon} + \dfrac{\varepsilon}{2}\right) & \text{for } |\mu| < \varepsilon. \end{cases} \tag{78}$$

For typical simulations ε is set to 0.2. This modification is only applied to rarefaction waves. Although the dependence on ε is small and this entropy works well, it suffers from a drawback that a tunable parameter ε was introduced into the scheme and there is little physical justification for its use.

Note that in Roe method [12] one only needs an entropy fix at sonic points.

3.3. Dimensionally split and unsplit schemes. We consider a two-dimensional wave problem which is described by a conservative system of equations:

$$\mathbf{q}_{,\rm t} + \nabla \cdot \mathbf{F} = 0. \tag{79}$$

Here $\mathbf{q}$ stands for a vector of conserved quantities $(q_1, q_2, q_3)^T$ and $\mathbf{F}$ is the flux which is a 2×3-vector. Integrating Eq. (79) with respect to time over time-step Δt, and with respect to space over a cell surface $\Delta x \Delta y$, we find

$$\mathbf{q}_{\rm ij}^{n+1} = \mathbf{q}_{\rm ij}^{n} - \frac{\Delta t}{\Delta x \Delta y} \sum_{\rm i=1}^{4} \mathbf{F}(\overline{\mathbf{u}}_{\rm i}^{\rm n}) \cdot \mathbf{l}_{\rm i}, \tag{80}$$

where Gauss' theorem was applied to replace surface integral by integral along the four cell edges, $m = 1, ..., 4$. The vector $\mathbf{l}_{\rm m}$ is normal to the cell edge m and has the length of this edge. In this case the cell of four edges is chosen. The quantity $\mathbf{q}_{\rm ij}^{n}$ stands for the average of cell ij, taken at time-step $n\Delta t$, and $\overline{\mathbf{q}}_{\rm i}^{\rm n}$ denotes $\mathbf{q}^{\rm n}$ which is evaluated at the cell edge $\rm m$.

Note that discretization (80) mimics the integral form of Eq. (79). This property is of vital importance as the discretization should be able to capture shocks. Differential form (79) does not make sense for discontinuous profiles. However, the corresponding integral form is still valid. The surface integral, $\int \mathbf{F} \cdot d\mathbf{l}$, in Eq. (80) has been approximated by taking at each cell edge an average flux $\mathbf{F} = \mathbf{F}(\overline{\mathbf{u}}_{\rm i}^{\rm n})$, with $\overline{\mathbf{u}}_{\rm i}^{\rm n}$ a suitably chosen average for the particular cell edge i. The most accurate choice for $\overline{\mathbf{u}}_{\rm i}^{\rm n}$ would be to take the average of the left $\mathbf{q}_{\rm l}^{\rm n}$ and right $\mathbf{q}_{\rm r}^{\rm n}$ values, i.e.

$$\overline{\mathbf{u}}_{\rm i}^{\rm n} = \frac{\mathbf{q}_{\rm l}^{\rm n} + \mathbf{q}_{\rm r}^{\rm n}}{2}.$$

Operator splitting schemes. A common approach when solving multi-dimensional hyperbolic equations is to apply an operator splitting method [24]. The idea of the operator splitting method is to iterate sequentially one dimensional equations; in each time-step, multi-dimensional derivatives are split into a set of one-dimensional derivatives, with variations in other directions ignored temporarily. Then, each row and column in the grid is treated as a one-dimensional problem. Updating the flow quantities along each row is done using the one-dimensional solver. The popularity of these methods is a consequence of the fact that the numerical schemes lead to surprisingly good results [3] and that the strategy is very simple as any multi-dimensional scheme consists of a system of the one-dimensional problems. We explain this strategy for the two-dimensional system of equations

$$\mathbf{q}_{,\rm t} + \mathbf{A}\mathbf{q}_{,\rm x} + \mathbf{B}\mathbf{q}_{,\rm y} = 0 \tag{81}$$

which can be split into one-dimensional subequations

$$\mathbf{q}_{,\rm t} + \mathbf{A}\mathbf{q}_{,\rm x} = 0, \qquad x-\text{sweep}, \tag{82}$$

$$\mathbf{q}_{,\rm t} + \mathbf{B}\mathbf{q}_{,\rm y} = 0, \qquad y-\text{sweep}. \tag{83}$$

In the x-sweep we would solve Eq. (82) along $y = \text{const}$, updating $\mathbf{q}$ to $\mathbf{q}^*$. In the y-sweep we then use u^* for solving Eq. (83) along $x = \text{const}$.

Such procedure will generally introduce a splitting error unless $\mathbf{AB} = \mathbf{BA}$. We explain this in the following way. Define the exact solution operator $\mathbf{S}_{\Delta \rm t}$ as the operator which advances the exact solution $\mathbf{q}(x, y, t)$ of Eq. (81) by a time-step Δt. Analogously $\mathbf{S}^x_{\Delta \rm t}$ and $\mathbf{S}^y_{\Delta \rm t}$ are the operators which advance exact solutions of Eq. (82) and (83), respectively.

With a use of the Taylor expansion these operators can be written as

$$\mathbf{S}^x_{\Delta \mathrm{t}} \simeq e^{-\Delta t \mathbf{A} \frac{\partial}{\partial x}},$$

$$\mathbf{S}^y_{\Delta \mathrm{t}} \simeq e^{-\Delta t \mathbf{B} \frac{\partial}{\partial y}}, \tag{84}$$

$$\mathbf{S}_{\Delta \mathrm{t}} \simeq e^{-\Delta t (\mathbf{A} \frac{\partial}{\partial x} + \mathbf{B} \frac{\partial}{\partial y})},$$

where we introduced the notation

$$e^{a \frac{\partial}{\partial x}} = 1 + a \frac{\partial}{\partial x} + \frac{1}{2!} \left(a \frac{\partial}{\partial x} \right)^2 + \frac{1}{3!} \left(a \frac{\partial}{\partial x} \right)^3 + \cdots . \tag{85}$$

A simple first-order (in time) splitting approach can be written as:

$$\mathbf{S}^x_{\Delta \mathrm{t}} \mathbf{S}^y_{\Delta \mathrm{t}} = \mathbf{S}_{\Delta \mathrm{t}} + \mathcal{O}((\Delta t)^2). \tag{86}$$

Strang [25] showed that

$$\mathbf{S}^x_{\frac{\Delta \mathrm{t}}{2}} \mathbf{S}^y_{\Delta \mathrm{t}} \mathbf{S}^x_{\frac{\Delta \mathrm{t}}{2}} = \mathbf{S}_{\Delta \mathrm{t}} + \mathcal{O}((\Delta t)^3). \tag{87}$$

So, the above combination of one-dimensional solution operators approximates the full evolution operator within second-order accuracy.

The splitting error does not often exceed errors in other numerical schemes, and the dimensional splitting can be a very effective approach [24]. However, the operator splitting methods have several disadvantages. For example, discontinuities traveling obliquelly to the grid are smeared more than those traveling in the coordinate directions, and implementation of boundary conditions may also be complicated using this method.

Operator unsplit methods. In unsplit methods, information is propagated in a genuinely multi-dimensional way. One-dimensional Riemann problems can be solved at the interfaces. Limiter functions are applied to suppress numerically induced oscillations which are usually generated by higher-order derivative terms. The left-going and right-going waves are split into parts propagating in the transverse direction by solving Riemann problems in coordinate directions tangential to the interfaces.

A class of conservative finite difference schemes for hyperbolic conservation laws in multi-dimensional spaces has been developed by Colella [26]. These schemes do not make use of operator splitting and instead the multidimensional wave properties of the solution are used to calculate fluxes. In these schemes some of the second-order terms are limited to suppress oscillations. Although the same Riemann problems appear in these schemes as in the operator split methods, these schemes are somewhat more expensive, requiring twice as many solutions to the Riemann problems as the corresponding operator split algorithm.

3.4. Adaptive mesh refinement. If a numerical solution of a flow-field containing complex flows which occur at high-gradient regions has to be determined, an appropriate resolution of these phenomena is of central importance for the overall quality of the solution. The positions of these high-gradient regions are usually unknown. At the same time it is not possible to refine the whole discretization domain because of limited computer power and required cost efficiency. A possible solution is to refine the grid locally where the high-gradient regions are detected [27, 28, 29]. The adaptive grid methodology makes it possible to achieve very high resolution in the most interesting regions. The algorithm is particularly well suited to unsteady flows.

The adaptive mesh refinement method (AMR) employs a hierarchical grid structure which changes dynamically and which is composed of grids of varying resolution. The grid which covers the entire computational domain is called the level 0 grid. There are also few additional levels of grid, each finer than the rest. These finer grids do not cover the whole domain but only those regions where more resolution is determined to be needed. The mesh is refined locally based on an estimate of the solution error [30]. By concentrating mesh points where they are most needed, high-quality solutions can be obtained at reasonable computational cost. By that way, careful attention is paid to resolving efficiently the disparate length scales. Various refinement criteria may be employed. For example, the local velocity divergence can be used to detect compressive phenomena, whereas the velocity curl can be used to detect shear [30]. That is, for a cell with a characteristic size Δx, the compressibility d_{c} and shear d_{s} detectors are

$$d_{\mathrm{c}} = |\nabla \cdot \mathbf{v}|(\Delta x)^{3/2}, \qquad d_{\mathrm{s}} = |\nabla \times \mathbf{v}|(\Delta x)^{3/2}. \tag{88}$$

Cells are refined or coarsened if d_{c} and d_{s} is above or below a specified threshold. Complex geometries can be treated with the use of the cut-cell approach [31]. In this case, a Cartesian grid is superimposed on the physical domain. Grid points which fall outside of the flow field are discarded, resulting in a series of intersected cells along the boundary delimiting the flow field. While the discretization in the interior of the flow field is unchanged, the discretization at the boundaries must be altered to account for the cut-cells. For complex flows, adaptive meshing can easily be implemented by simple cell-subdivision.

It is noteworthy here that the remeshing technique is relatively easy for triangular grids [32]. The refinement can be performed through triangle subdivision, where a triangle is branched into two triangles by cutting it on its longest side. The coarsening can be obtained through node removal. The node is flagged for removal if all its neighboring triangles are to be coarsened. This node removal leaves an open polygon, which is then remeshed [32].

4. Numerical results

As a consequence of complexity of hydrodynamic waves in highly inhomogeneous medium it is necessary to understand simpler phenomena which may play the role of elementary building blocks in the construction of a more elaborated theory. As a result, our strategy is to develop simpler models at the initial stage of the research and progressively extend and generalize them to more complex models at subsequent stages. Therefore in this study, we assume that at the equilibrium the solar atmosphere is settled in a two-dimensional and

still ($\mathbf{V} = 0$) environment. At the equilibrium, equilibrium gas pressure and mass density are given as

$$p(y) = p_0 \ \exp\left(-\int_{y_r}^{y} \frac{dy'}{\Lambda(y')}\right), \tag{89}$$

$$\varrho(y) = \frac{p(y)}{g\Lambda(y)}. \tag{90}$$

Here

$$\Lambda(y) = k_B T(y)/(mg) \tag{91}$$

is the pressure scale-height, and p_0 denotes the gas pressure at the reference level, $y = y_r$.

We adopt a temperature profile $T(y)$ which is displayed in Fig. 2. Note that T attains a value of about 5700 K at the top of the photosphere which corresponds to $y = 0.5$ Mm. At higher altitudes $T(y)$ falls off until it reaches its minimum of 4350 K at the altitude of $y \simeq 0.95$ Mm. Higher up $T(y)$ grows gradually with height up to the transition region which is located at $y \simeq 2.7$ Mm. Here $T(y)$ experiences a sudden growth up to the coronal value of 1.5 MK at $y = 10$ Mm. Having specified $T(y)$ with a use of Eqs. (89) and (90) we can determine mass density and gas pressure profiles.

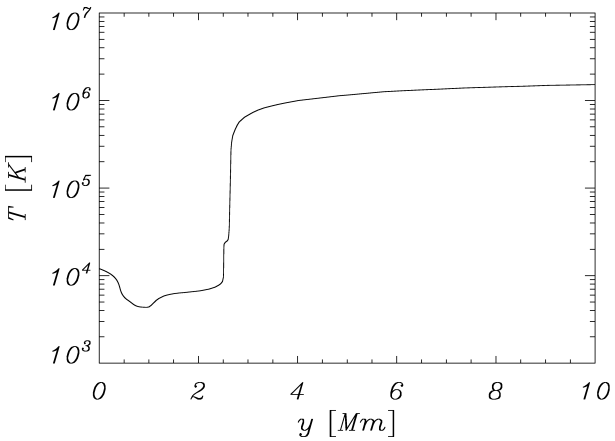

Fig. 2. Temperature (in Kelvins, logarithmic scale) profile vs. altitude y (in Mm) in the solar atmosphere.

We excite waves in the solar atmosphere by launching initially, at $t = 0$, the impulse in a vertical component of velocity V_y, i.e.

$$V_y(x, y, t = 0) = A_v \exp\left[-\frac{x^2}{w^2} - \frac{y^2}{w^2}\right]. \tag{92}$$

Here A_v is the amplitude of the initial Gaussian pulse and $w = 0.3$ Mm is its width.

Euler equations of Eq. (8) are solved numerically using the code FLASH [6, 27] which implements a second-order unsplit Godunov solver and AMR which originates from Paramesh, the AMR mesh package used in FLASH. FLASH is a component-based massively parallel multiphysics simulation code with a wide user base. The time integration in FLASH was originally designed using Strang operator splitting for hydrodynamics. A directionally unsplit solver is a new high-order Godunov hydrodynamics solver in FLASH. The second-order accurate MUSCL-Hancock type method [8] is implemented in the code as the default. The unsplit method is able to solve 1D, 2D and 3D problems by switching between different types of Riemann solvers (Roes linearized solver, HLL, HLLC), slope limiters (Monotonized central-difference, Minmod, van Leers).

In our simulations, we set the simulation box as $(-15, 15)$ Mm $\times(-0.5, 29.5)$ Mm. At all boundaries we fixed all plasma quantities to their equilibrium values. In our studies we use AMR grid with a minimum (maximum) level of refinement blocks set to 5 (8). The refinement strategy is based on controlling numerical errors in a gradient of mass density. Such settings result in an excellent resolution of steep spatial profiles, which significantly reduces numerical diffusion within the simulation region.

Figure 3 illustrates spatial profiles of $\log\varrho$ and velocity vectors at $t = 200$ s (left top), $t = 300$ s (right top), $t = 400$ s (left bottom), $t = 600$ s (right bottom) for the initial pulse amplitude $A_v = 5$ km s^{-1}. The initial pulse splits in a usual way into counter-propagating waves. The wave propagating upwards grows in its amplitude as a result of the rapid decrease of mass density in the chromosphere. As a consequence of that a shock results in. Photospheric and chromospheric plasma is lifted up by underpressure which settles in below the shock. The pressure gradient force overwhelms gravity and it pushes the photospheric and chromospheric material towards the solar corona. This scenario is clearly seen at $t = 250$ s. At a later time the plasma becomes attracted by gravity and as a result is falls off towards the low layers. However, the secondary shock which results from the original pulse works against this fall off as it lifts up the photospheric and chromospheric plasma. As a result a complex bi-directional flows arises. The whole scenario bares many features of solar spicules. Note a development of vertices which are clearly seen at $t = 300$ s and later on. These vertices cascade into smaller scale vertices which propagate horizontally. Such a horizontal propagation can be justified on analytical basements [33].

This scenario consists the building block of 1D rebound shock model of Hollweg [34] who proposed that the secondary shock (or rebound shock) lifts up the transition region higher than the first shock thereby resulting in a spicule appearance at observed heights. The process is well studied in the frame of 1D numerical simulations. However, our 2D numerical simulations introduce new interesting features in comparison to the 1D rebound shock scenario.

There are few conclusions which result from our simulations:

a) according to the theory of Klein-Gordon equation an initial pulse generates a wave front and a trailing wake which oscillates with acoustic cut-off frequency [34];
b) even a small amplitude initial pulse launched at the top of the photosphere exhibits a tendency to generate shocks. These shocks result from a nonlinear wake.

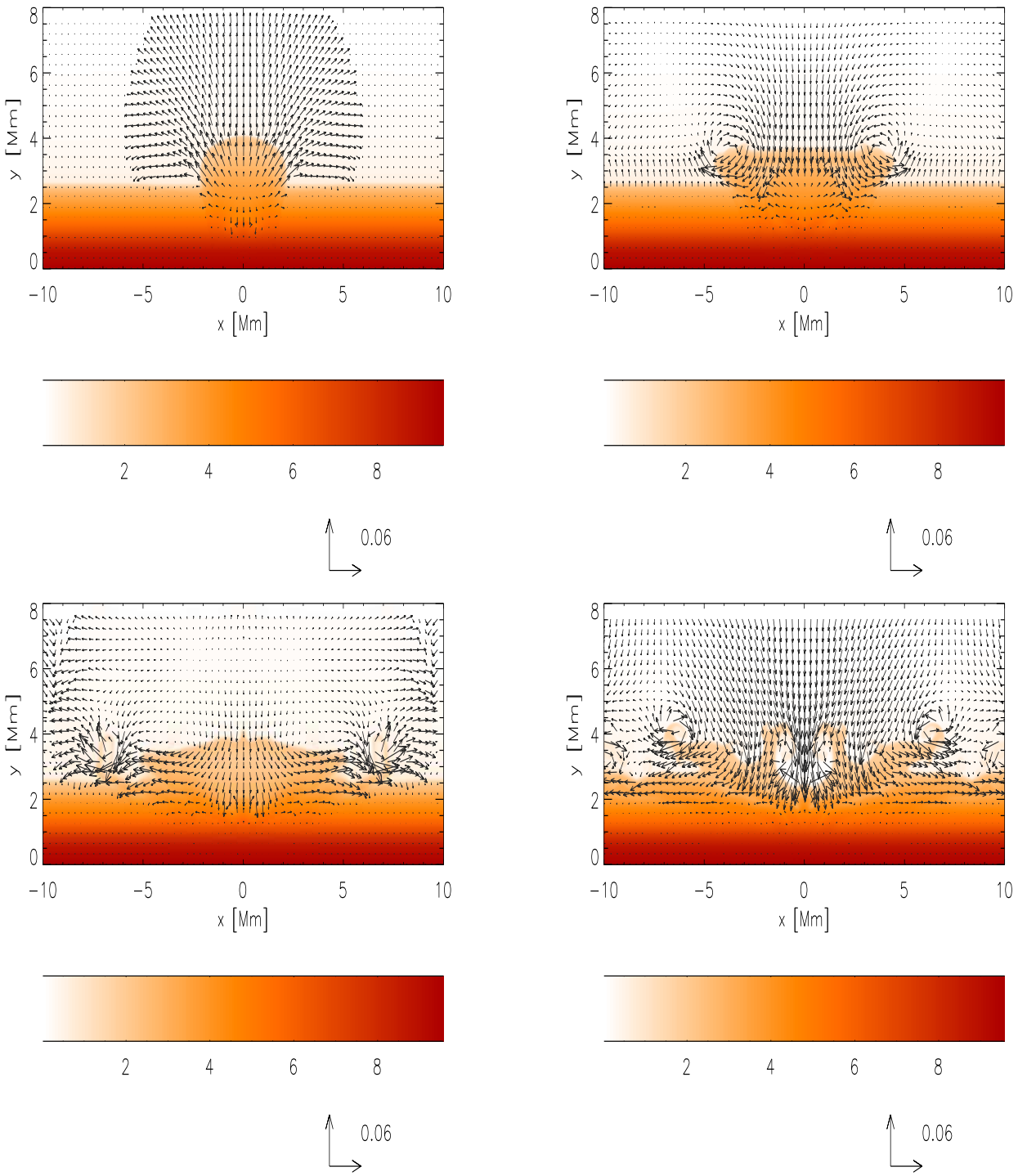

Fig. 3. Mass density (contour plots, log scale) and velocity (arrows) profiles at $t = 200$ s (top panel), $t = 300$ s, $t = 400$ s, and $t = 600$ s (bottom panel) for ($x_0 = 0$, $y_0 = 0.5$) Mm and $A_v = 2$ km s^{-1}. Mass density and velocity are expressed in units of 10^{-12} kg m^{-3} and 1 Mm s^{-1}, respectively.

5. Summary

This paper presents several mathematical aspects in numerical methods for hydrodynamic equations. Although this presentation is far from complete as the emphasis is on the methods which are the most effective and the best known to the authors.

There are several conditions that numerical schemes should satisfy: accuracy and speed of numerical simulations, adequate representation of complex flows and steep profiles, without generation spurious oscillations as well as *robustness*. A computer code is described as being robust if it has the virtue of giving reliable results to a wide range of problems

without needing to be retuned. Numerical schemes such as shock-capturing schemes described in this paper satisfy these conditions.

Existing numerical models such as was used in Sec. 4 with an adaptation of the FLASH code demonstrate the feasibility of fluid simulations in obtaining at least qualitative and, to some extent, quantitative features in the fluid media. With continued improvements in computational methods and computer resources, the usefulness and capability of the numerical approach should continue to improve.

The FLASH code has been developed by the DOE-supported ASC/Alliance Center for Astrophysical Thermonuclear Flashes at the University of Chicago.

REFERENCES

[1] J. Trangenstein, *Numerical Solution of Hyperbolic Partial Differential Equations*, Cambridge University Press, Cambridge, 2008.

[2] J. von Neumann and R.D. Richtmeyer, "A method for the numerical calculation of hydrodynamic shocks", *J. Appl. Phys.* 21, 232–237 (1950).

[3] J.M. Stone and M.L. Norman, "ZEUS-2D: a radiation magnetohydrodynamics code for astrophysical flows in two space dimensions. II. The magnetohydrodynamic algorithms and tests", *Astrophys. J. Suppl. Ser.* 80, 791–818 (1992).

[4] K. Murawski and R.S. Steinolfson, "Numerical modeling of the solar wind interaction with Venus", *Planet. Space Sci.* 44 (3), 243–252 (1996).

[5] S.K. Godunov, "A difference scheme for numerical solution of discontinuos solution of hydrodynamic equations", *Math. Sb.* 47, 271–306 (1959).

[6] D. Lee and A.E. Deane, "An unsplit staggered mesh scheme for multidimensional magnetohydrodynamics", *J. Comput. Phys.* 228 (4), 952–975 (2009).

[7] J.J. Quirk, "An adaptive grid algorithm for computational shock hydrodynamics", *PhD Thesis*, College of Aeronautics, Cranfield Institute of Technology, Cranfield, 1991.

[8] E. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer, Berlin, 2009.

[9] R.J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhäuser Verlag Basel, Berlin, 1990.

[10] A. Harten, P.D. Lax, and B. van Leer, "On upstream differencing and Godunov-type schemes for hyperbolic conservation laws", *SIAM Rev.* 25 (1), 35–61 (1983).

[11] B. Einfeld, "On Godunov-type methods for gas dynamics", *SIAM J. Num. Anal.* 25 (2), 294–318 (1988).

[12] P.L. Roe, "Approximate Riemann solvers, parameter vectors and difference schemes", *J. Comp. Phys.* 43, 357–372 (1981).

[13] N. Aslan, "Two-dimensional solutions of MHD equations with an adapted Roe method", *Int. J. Numer. Meth. Fluids* 23 (11), 1211–1222 (1996).

[14] B. Einfeld, C.D. Munz, P.L. Roe, amd B. Sjögreen, "On Godunov-type methods near low densities", *J. Comp. Phys.* 92 (2), 273–295 (1991).

[15] R. Donat and A. Marquina, "Capturing shock reflections: an improved flux formula", *J. Comp. Phys.* 125 (1), 42–58 (1996).

[16] S. Jin and Z.P. Xin, "The relaxation schemes for systems of conservation laws in arbitrary space dimensions", *Comm. Pure Appl. Math.* 48 (3), 235–276 (1995).

[17] R.J. LeVeque, *Finite-volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, 2002.

[18] R.J. LeVeque and M. Pelanti, "A class of approximate Riemann solvers and their relation to relaxation schemes", *J. Comput. Phys.* 172 (2), 572–591 (2001).

[19] V.P. Kolgan, "Application of the minimum-derivative principle in the construction of finite-diverence schemes for numerical analysis of discontinuous solutions in gas dynamics", *Uch. Zap. TsaGI* 3 (6), 68–77 (1972).

[20] B. van Leer, "Towards the ultimate conservative difference scheme. V-A second-order sequel to Godunov's method", *J. Comp. Phys.* 32, 101–136 (1979).

[21] P.L. Roe, "Sonic flux formulae", *SIAM J. Sci. Stat. Comput.* 13, 611–630 (1982).

[22] A. Harten, "ENO schemes with subcell resolution", *J. Comp. Phys.* 83, 148–184 (1989).

[23] P.R. Woodward and P. Colella, "The numerical simulation of two-dimensional fluid flow with strong shocks", *J. Comp. Phys.* 54, 115–173 (1984).

[24] K. Murawski and M. Goossens, "Operator splitting for multidimensional magnetohydrodynamics", *J. Geophys. Res.* 99 (A6), 11569–11573 (1994).

[25] G. Strang, "On the construction and comparison of difference schemes", *SIAM J. Num. Anal.* 5 (3), 506–517 (1968).

[26] P. Colella, "Multidimensional upwind methods for hyperbolic conservation laws", *J. Comp. Phys.* 87, 171–200 (1990).

[27] M.J. Berger and P. Colella, "Local adaptive mesh refinement for shock hydrodynamics", *J. Comp. Phys.* 82, 64–84 (1989).

[28] J. Bell, M. Berger, J. Saltzman, and M. Welcome, "Three dimensional adaptive mesh refinement for hyperbolic conservation laws", *SIAM J. Sci. Comput.* 15 (1), 127–138 (1994).

[29] D.F. Martin and P. Colella, "A cell-centered adaptive projection method for the incompressible Euler equations", *J. Comp. Phys.* 163 (2), 271–312 (2000).

[30] D. DeZeeuw and K.G. Powell, "An adaptively refined Cartesian mesh solver for the Euler equations", *J. Comp. Phys.* 104, 56–68 (1993).

[31] J.J. Quirk, "A cartesian grid approach with hierarchical refinement for compressible flows", *Computers and Fluids* 23 (1), 125–142 (1994).

[32] J.-Y. Trepanier, M. Reggio, and D. Ait-Ali-Yahia, "An implicit flux- difference splitting method for solving the Euler equations on adaptive traingular grids", *Int. J. Num. Meth. Heat Fluid Flow* 3 (1), 63–77 (1993).

[33] S. Arendt, "Vorticity in stratified fluids. I. General formulation", *Geophys. Astrophys. Fluid Dynamics* 68 (1), 59–83 (1993).

[34] J.V. Hollweg, "On the origin of solar spicules", *Astrophys. J.* 257, 345–353 (1982).