BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 60, No. 1, 2012
DOI: 10.2478/v10175-012-0008-7

Implementation of MUSCL-Hancock method into the C++ code for the Euler equations

K. MURAWSKI[1*], Jr. K. MURAWSKI[2], and P. STPICZYŃSKI[3,4]

[1] Institute of Informatics, UMCS, 1 M. Curie-Sklodowskiej St., 20-031 Lublin, Poland
[2] Faculty of Mathematics, Physics and Informatics, UMCS, 10 Radziszewskiego St., 20-031 Lublin, Poland
[3] Institute of Mathematics, UMCS, 1 M. Curie-Sklodowskiej St., 20-031 Lublin, Poland
[4] Institute of Theoretical and Applied Informatics of the Polish Academy of Sciences, 5 Baltycka St., 44-100 Gliwice, Poland

Abstract. In this paper we present implementation of the MUSCL-Hancock method for numerical solutions of the Euler equations. As a result of the internal complexity of these equations solving them numerically is a formidable task. With the use of the original C++ code, we developed and presented results of a numerical test that was performed. This test shows that our code copes very well with this task.

Key words: hyperbolic equations, numerical methods, Godunow methods.

1. Introduction

In this paper we concentrate on a first-order, one-dimensional hyperbolic system, a related form of hyperbolic equations,

$$\mathbf{q}_{,\mathrm{t}} + \mathbf{f}(\mathbf{q})_{,\mathrm{x}} = 0. \tag{1}$$

Here we specify a state vector $\mathbf{q}$ and a flux function $\mathbf{f}$ as

$$\mathbf{q} = [q_1, \ldots, q_\mathrm{m}]^\mathrm{T}, \quad \mathbf{f} = [f_1, \ldots, f_\mathrm{m}]^\mathrm{T}, \quad m = 1, 2, \ldots, \tag{2}$$

where $^\mathrm{T}$ denotes the vector transposition, and the derivatives with the respect to time t and the spatial coordinate x are expressed as

$$\mathbf{q}_{,\mathrm{t}} \equiv \frac{\partial \mathbf{q}}{\partial t}, \qquad \mathbf{f}_{,\mathrm{x}} \equiv \frac{\partial \mathbf{f}}{\partial x}. \tag{3}$$

The above equation is called *hyperbolic* if the Jacobian, $\mathbf{A} = \dfrac{\partial \mathbf{f}}{\partial \mathbf{q}}$, has real eigenvalues and is diagonalizable, i.e., has a complete set of linearly independent eigenvectors. Among others, examples of hyperbolic equations are the Euler equations which consist a set of three nonlinear equations (see Sec. 3). Because of their complexity, usually they cannot be solved analytically and therefore they must be calculated numerically. A numerical approach requires a model, which can be developed from a set of discretized differential equations (see Sec. 3).

These days, there is a number of numerical methods that attempt to solve hyperbolic equations. Godunov-type methods are considered as one of the most successful. These methods originate from the upwind scheme which was developed by Godunov [1]. In his approach a numerical approximation to a solution of hyperbolic equations is computed by means of a scheme which is derived from the integral form of the hyperbolic equations (see Eq. (5)). The original scheme of Godunov [1] uses the solution of the Riemann problem with piece-wise constant initial data within a numerical cell to compute the upwind numerical flux. The Riemann problem is defined as the initial problem with conditions represented by two constant states separated by a discontinuity [2]. At each jump of such a function a Riemann problem is solved for some time interval to fulfill the stability constraint. The resulting solution is averaged over the grid cells to get a piece-wise constant function, approximating the solution at the new time level.

Godunov [1] developed the first-order-accurate upwind scheme among a family of simple discretizations. It is noteworthy that Godunov [1] proved the theorem in which he stated that if an upwind scheme preserves the monotonicity of the solution this scheme is at most first-order accurate. This result could discourage anyone attempting to improve his scheme. Despite of that, the extension to second-order of accuracy in time and space was carried out by using a non-oscillatory piece-wise linear representation of data in a numerical cell. As a result of these efforts, nine years later the IBM researcher, J. Fromm, developed a higher-order scheme of low level of erroneous oscillations. By combining schemes with predominantly both positive and negative phase errors, Fromm received low dispersive errors [2]. Later on Kolgan [3] proposed to reduce spurious oscillations by applying the so-called principle of minimal values of derivatives, producing in this manner a non-oscillatory Godunov-type scheme of second order spatial accuracy. Van Leer [4] developed Monotone Upstream Scheme for Conservation Laws (MUSCL) in which he included a linear representation of a solution within each numerical cell. MUSCL was greatly simplified in 1980 by Steve Hancock but the modified method was actually published in 1984 by van Leer [5] and since that time it bears a common name MUSCL-Hancock method. See also van Leer [6].

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

A variant of the MUSCL-Hancock method was proposed by Falle [7]. This method was proved to be stable by Berthon [8, 9] and it was used in a number of applications, see for instance [10, 11], and references therein.

The goal of this paper is to present and adopt the MUSCL-Hancock method in numerical solutions of hyperbolic equations. The reason we pay our attention on this method is its robustness and accurate representation of complex solutions [10, 11]. We realize our aim by reviewing theory of the Euler equations in the following section. Here we introduce the Euler equations and specify the Riemann problem for them. In Sec. 3 we describe briefly the finite-volume and Godunov methods. We devote Sec. 4 to review the HLL and HLLC Riemann solvers. We present more accurate methods such as the MUSCL-Hancock method in Sec. 5. In the following section we present results of original studies of coding the above described numerical methods to solve one-dimensional Euler equations. This paper is completed by a summary of the main results and conclusions in Sec. 7.

2. Theory of the Euler equations

We recall that the *differential form* of the conservation law, Eq. (1), breaks down in the presence of discontinuities (shocks, contact waves). The *integral form* of the conservation law, which works for both continuous and discontinuous solutions is

$$\frac{d}{dt}\int\limits_{x_l}^{x_r}\mathbf{q}(x,t)\mathrm{dx}=\mathbf{f}(\mathbf{q}(x_l,t))-\mathbf{f}(\mathbf{q}(x_r,t)). \tag{4}$$

Consider the control volume $[x_l,x_r]\times[t_1,t_2]$. Integration in time of Eq. (4) leads to

$$\begin{aligned}\int\limits_{x_l}^{x_r}\mathbf{q}(x,t_2)\,\mathrm{dx}&=\int\limits_{x_l}^{x_r}\mathbf{q}(x,t_1)\,\mathrm{dx}\\&+\int\limits_{t_1}^{t_2}\mathbf{f}(\mathbf{q}(x_l,t))\,\mathrm{dt}-\int\limits_{t_1}^{t_2}\mathbf{f}(\mathbf{q}(x_r,t))\mathrm{dt}.\end{aligned} \tag{5}$$

We use the above form in *construction of finite-volume numerical methods* which are introduced in Sec. 3.

2.1. The Euler equations. We can express the Euler equations in the conservative form of Eq. (1) with the conservative state vector,

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

and the flux function,

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

Here ϱ is a mass density, u velocity, and E energy density such as

$$E=\frac{p}{\gamma-1}+\frac{\varrho u^2}{2}. \tag{8}$$

The symbol γ denotes the specific heats ratio.

We rewrite now the Euler equations in the quasi-linear form,

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

with the Jacobian,

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

where $H=(E+p)/\varrho$ is the total specific enthalpy.

With a use of the *non-conservative state vector*

$$\mathbf{w}=\begin{bmatrix}w_1\\w_2\\w_3\end{bmatrix}=\begin{bmatrix}\varrho\\u\\p\end{bmatrix} \tag{11}$$

we rewrite the Euler equations in the quasi-linear form,

$$\mathbf{w}_{,\mathrm{t}}+\mathbf{A}_\mathrm{n}\mathbf{w}_{,\mathrm{x}}=0\,. \tag{12}$$

Here the matrix $\mathbf{A}_\mathrm{n}$ is

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

with the sound speed $c_\mathrm{s}=\sqrt{\gamma p/\varrho}$.

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

$$\mathbf{A}_\mathrm{l}\mathbf{r}_\mathrm{l}^{(\mathrm{i})}=\lambda^{(\mathrm{i})}\mathbf{r}_\mathrm{l}^{(\mathrm{i})},\qquad \mathrm{l}=\mathrm{c},\mathrm{n},\qquad \mathrm{i}=1,2,3 \tag{13}$$

with $\mathbf{r}_\mathrm{l}^{(\mathrm{i})}$ being the right eigenvector and $\lambda^{(\mathrm{i})}$ denoting the corresponding eigenvalue.

We have

$$\mathbf{R}_\mathrm{c}^{-1}\mathbf{A}_\mathrm{c}\mathbf{R}_\mathrm{c}=\mathbf{\Lambda} \tag{14}$$

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

$$\begin{aligned}\mathbf{R}_\mathrm{c}&=\left[\mathbf{r}_\mathrm{c}^{(1)},\mathbf{r}_\mathrm{c}^{(2)},\mathbf{r}_\mathrm{c}^{(3)}\right]\\&=\begin{bmatrix}1&1&1\\u-c_\mathrm{s}&u&u+c_\mathrm{s}\\H-uc_\mathrm{s}&\frac{u^2}{2}&H+uc_\mathrm{s}\end{bmatrix}.\end{aligned} \tag{15}$$

Here the matrix $\mathbf{\Lambda}$ is

$$\mathbf{\Lambda} = \begin{bmatrix} \lambda^{(1)} & 0 & 0 \\ 0 & \lambda^{(2)} & 0 \\ 0 & 0 & \lambda^{(3)} \end{bmatrix} = \begin{bmatrix} u - c_\mathrm{s} & 0 & 0 \\ 0 & u & 0 \\ 0 & 0 & u + c_\mathrm{s} \end{bmatrix}. \tag{16}$$

From Eq. (13) we find the right eigenvectors of $\mathbf{A}_\mathrm{n}$ as

$$\mathbf{R}_\mathrm{n} = \left[\mathbf{r}_\mathrm{n}^{(1)}, \mathbf{r}_\mathrm{n}^{(2)}, \mathbf{r}_\mathrm{n}^{(3)}\right] = \begin{bmatrix} 1 & 1 & 1 \\ -c_\mathrm{s}/\varrho & 0 & c_\mathrm{s}/\varrho \\ c_\mathrm{s}^2 & 0 & c_\mathrm{s}^2 \end{bmatrix}. \tag{17}$$

Note that the right eigenvectors of $\mathbf{A}_\mathrm{c}$ and $\mathbf{A}_\mathrm{n}$ differ one from each other, while the eigenvalues of $\mathbf{A}_\mathrm{c}$ and $\mathbf{A}_\mathrm{n}$ are identical and they are

$$\lambda^{(1)} = u - c_\mathrm{s}, \qquad \lambda^{(2)} = u, \qquad \lambda^{(3)} = u + c_\mathrm{s}. \tag{18}$$

As these eigenvalues are real and they correspond to a set of linearly independent eigenvectors $\mathbf{r}_\mathrm{l}^{(1)}, \mathbf{r}_\mathrm{l}^{(2)}, \mathbf{r}_\mathrm{l}^{(3)}$, $\mathrm{l} = \mathrm{c}, \mathrm{n}$, we conclude that the Euler equations are hyperbolic.

2.2. The Riemann problem for the Euler equations. For the Euler equations we can always solve the Riemann problem. The solution consists of three simple waves (a shock, a contact wave, and a rarefaction wave) traveling with finite velocities which are specified by Eq. (18). The procedure for constructing the solution of the Riemann Problem is called a Riemann solver. We discuss below these simple waves separately.

For *a shock* we can apply either the Rankine-Hugoniot condition [2],

$$\mathbf{f}(\mathbf{q}_\mathrm{r}) - \mathbf{f}(\mathbf{q}_\mathrm{l}) = \lambda_\mathrm{s}(\mathbf{q}_\mathrm{r} - \mathbf{q}_\mathrm{l}), \tag{19}$$

or the Riemann invariants [2]. Note that a shock moving with its speed λ_s should satisfy the entropy condition

$$\lambda^{(\mathrm{i})}(\mathbf{q}_\mathrm{l}) > \lambda_\mathrm{s} > \lambda^{(\mathrm{i})}(\mathbf{q}_\mathrm{r}). \tag{20}$$

Hence, we infer that the characteristics are convergent.

We can describe *a contact wave* either by the Rankine-Hugoniot condition,

$$\mathbf{f}(\mathbf{q}_\mathrm{r}) - \mathbf{f}(\mathbf{q}_\mathrm{l}) = \lambda_\mathrm{c}(\mathbf{q}_\mathrm{r} - \mathbf{q}_\mathrm{l}), \tag{21}$$

or by the Riemann invariants. Here λ_c is the contact wave speed. For this wave characteristics are parallel

$$\lambda^{(\mathrm{i})}(\mathbf{q}_\mathrm{l}) = \lambda^{(\mathrm{i})}(\mathbf{q}_\mathrm{r}) = \lambda_\mathrm{c}. \tag{22}$$

For *a rarefaction wave* we adopt the generalized Riemann invariants. For this wave characteristics are divergent as we have

$$\lambda^{(\mathrm{i})}(\mathbf{q}_\mathrm{l}) < \lambda^{(\mathrm{i})}(\mathbf{q}_\mathrm{r}). \tag{23}$$

As a rarefaction wave corresponds to a smooth flow it is convenient to rewrite the Euler equations with the use of the state vector,

$$\mathbf{w} = [\varrho, u, s]^\mathrm{T}, \tag{24}$$

where $s \sim p/\varrho^\gamma$ is the entropy. Then we have

$$\mathbf{w}_{,\mathrm{t}} + \mathbf{A}(\mathbf{w})\mathbf{w}_{,\mathrm{x}} = \mathbf{0} \tag{25}$$

with

$$\mathbf{A}(\mathbf{w}) = \begin{bmatrix} u & \varrho & 0 \\ c_\mathrm{s}^2/\varrho & u & p_{,\mathrm{s}}/\varrho \\ 0 & 0 & u \end{bmatrix}. \tag{26}$$

The eigenvalue, $\lambda^{(\mathrm{i})}$, and the right eigenvector, $\mathbf{r}^{(\mathrm{i})}$, $i = 1, 2, 3$, of $\mathbf{A}(\mathbf{w})$ are

$$\lambda^{(1)} = u - c_\mathrm{s}, \qquad \lambda^{(2)} = u, \qquad \lambda^{(3)} = u + c_\mathrm{s}, \tag{27}$$

$$[\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \mathbf{r}^{(3)}] = \begin{bmatrix} 1 & -p_{,\mathrm{s}} & 1 \\ -c_\mathrm{s}/\varrho & 0 & c_\mathrm{s}/\varrho \\ 0 & c_\mathrm{s}^2 & 0 \end{bmatrix}. \tag{28}$$

Equation (25) can be used for a description of rarefaction waves. Indeed the Riemann invariants for the left propagating rarefaction wave lead to [2]

$$\frac{d\varrho}{1} = \frac{du}{-c_\mathrm{s}/\varrho} = \frac{ds}{0}. \tag{29}$$

Hence, we conclude that

$$s = \text{const}, \qquad u + \int c_\mathrm{s}\frac{d\varrho}{\varrho} = u + \frac{2c_\mathrm{s}}{\gamma - 1} = \text{const}. \tag{30}$$

By performing similar computations but for the rarefaction wave propagating with $\lambda^{(3)}$ we get

$$s = \text{const}, \qquad u - \int c_\mathrm{s}\frac{d\varrho}{\varrho} = u - \frac{2c_\mathrm{s}}{\gamma - 1} = \text{const}. \tag{31}$$

Equations (30) and (31) are useful in finding solutions of the Riemann problem in the case of the rarefaction waves.

We consider the following cases:

1. a left propagating wave which is:
   (a) a shock. This wave can be described by the Rankine-Hugoniot conditions. It is convenient to rewrite these conditions in the reference frame moving with the shock speed, λ_s. Then we have these conditions written as
   $$\varrho_\mathrm{l}\widehat{u}_\mathrm{l} = \varrho_{*\mathrm{l}}\widehat{u}_{*\mathrm{l}}, \tag{32}$$
   $$\varrho_\mathrm{l}\widehat{u}_\mathrm{l}^2 + p_\mathrm{l} = \varrho_{*\mathrm{l}}\widehat{u}_{*\mathrm{l}}^2 + p_*, \tag{33}$$
   $$\widehat{u}_\mathrm{l}(\widehat{E}_\mathrm{l} + p_\mathrm{l}) = \widehat{u}_*(\widehat{E}_{*\mathrm{l}} + p_*), \tag{34}$$
   where
   $$\widehat{u}_\mathrm{l} = u_\mathrm{l} - \lambda_\mathrm{s}, \qquad \widehat{u}_* = u_* - \lambda_\mathrm{s}; \tag{35}$$
   (b) a rarefaction wave which is described by the isentropic law (entropy is constant),
   $$p = C\varrho^\gamma, \qquad C = \text{const}, \tag{36}$$
   and the generalized Riemann invariant which follows from Eq. (30),
   $$u_\mathrm{l} + \frac{2c_{\mathrm{sl}}}{\gamma - 1} = u_* + \frac{2c_{\mathrm{s}*}}{\gamma - 1}; \tag{37}$$
2. a right propagating wave, which consists either of a shock or rarefaction wave, is described by analogous conditions to the left wave.

We can use Eqs. (32)–(37) to solve the Riemann problem for the Euler equations. As a result, we arrive at a transcendental algebraic equation for a gas pressure p_*. See Eq. (4.5) of Toro [2]. This equation can be solved numerically with the use of an iterative method.

3. The finite-volume and Godunov's methods

With the use of Eq. (5) we discretize the system of hyperbolic equations as

$$\mathbf{q}_{\rm i}^{\rm n+1} = \mathbf{q}_{\rm i}^{\rm n} + \frac{\Delta t}{\Delta x}(\mathbf{f}_{\rm i-1/2} - \mathbf{f}_{\rm i+1/2}). \tag{38}$$

Here we introduced a uniform grid size Δx and a time-step Δt as

$$x_{\rm i} = i\Delta x, \quad i = 0, 1, \ldots, i_{\max}, \quad \Delta x = x_{\rm i+1} - x_{\rm i}, \tag{39}$$

$$t^{\rm n} = n\Delta t, \quad n = 0, 1, \ldots, n_{\max}, \quad \Delta t = t^{\rm n+1} - t^{\rm n} \tag{40}$$

and implemented

$$\begin{aligned} \mathbf{q}_{\rm i}^{\rm n} &= \frac{1}{\Delta x}\int\limits_{x_{\rm i-1/2}}^{x_{\rm i+1/2}} \mathbf{q}(x, t^{\rm n})\,{\rm dx}, \\ \mathbf{f}_{\rm i\pm1/2}^{\rm n} &= \frac{1}{\Delta t}\int\limits_{t}^{t+\Delta t} \mathbf{f}(\mathbf{q}(x_{\rm i\pm1/2}, t))\,{\rm dt}. \end{aligned} \tag{41}$$

The indices $i \pm 1/2$ correspond to the intercells. Equation (38) results from the finite-volume method which mimics conservative properties of the Euler equations.

We specify the Riemann problem as

$$\mathbf{q}_{,\rm t} + \mathbf{f}(\mathbf{q})_{,\rm x} = \mathbf{0}, \tag{42}$$

$$\mathbf{q}(x, 0) = \begin{cases} \mathbf{q}_{\rm l}\,, & x < 0, \\ \mathbf{q}_{\rm r}\,, & x \geq 0. \end{cases} \tag{43}$$

Its solution is

$$\mathbf{q}_{\rm i+1/2}(x/t), \tag{44}$$

which determines Godunov's numerical flux

$$\mathbf{f}_{\rm i+1/2}^{\rm n} = \frac{1}{\Delta t}\int\limits_{t}^{t+\Delta t} \mathbf{f}(\mathbf{q}(0/t))\,{\rm dt} = \mathbf{f}(\mathbf{q}(0/t)). \tag{45}$$

For a system of linear equations the flux f(q) is

$$\mathbf{f}(\mathbf{q}) = \mathbf{A}\mathbf{q}. \tag{46}$$

Then, Godunov's flux is

$$\mathbf{f}_{\rm i+1/2} = \mathbf{A}\mathbf{q}_{\rm i+1/2}(0/t). \tag{47}$$

For a linear system we find the stability condition as [2]

$$\Delta t = c_{\rm cfl}\frac{\Delta x}{s}, \qquad s = \max(|\lambda^{\rm (i)}|), \qquad {\rm i} = 1, 2, 3, \tag{48}$$

where we implemented the Courant-Friedrichs-Lewy (CFL or Courant) number $c_{\rm cfl}$ as

$$c_{\rm cfl} = \frac{s\Delta t}{\Delta x} = \frac{s}{\Delta x/\Delta t} = \frac{\text{advection speed}}{\text{grid speed}}. \tag{49}$$

We conclude this part of the paper saying that a careful care in choosing maximum wave speeds s for CFL condition is required, and Godunov's numerical methods can be successfully adopted both to linear and nonlinear systems.

4. Approximate Riemann solvers

As we already showed in Subsec. 2.2 the exact Riemann solver is available for the Euler equations but, as it leads to a transcendental equation, it is numerically expensive. Therefore, approximate Riemann solvers can be constructed instead. We present below two examples of approximate Riemann solvers.

4.1. The HLL Riemann solver. We review here the Harten-Lax-van Leer (HLL) Riemann solver which takes into account the fastest waves only. To do so, we apply the integral form of the conservation laws in the control volume of Fig. 1. The integral form of the set of hyperbolic equations, in the control volume $[x_{\rm l}, x_{\rm r}] \times [0, T]$ is

$$\begin{aligned} \int\limits_{\rm x_l}^{\rm x_r} \mathbf{q}(x, T){\rm dx} &= \int\limits_{\rm x_l}^{\rm x_r} \mathbf{q}(x, 0){\rm dx} \\ &+ \int\limits_{0}^{\rm T} \mathbf{f}(\mathbf{q}(x_{\rm l}, t)){\rm dt} - \int\limits_{0}^{\rm T} \mathbf{f}(\mathbf{q}(x_{\rm r}, t)){\rm dt}. \end{aligned} \tag{50}$$

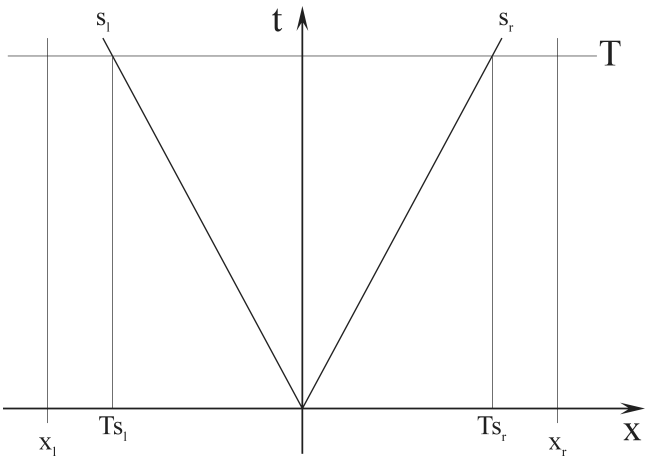

Fig. 1. Control volume for the HLL Riemann solver

Evaluating the right hand side of this expression, we obtain

$$\int\limits_{\rm x_l}^{\rm x_r} \mathbf{q}(x, T){\rm dx} = x_{\rm r}\mathbf{q}_{\rm r} - x_{\rm l}\mathbf{q}_{\rm l} + T(\mathbf{f}_{\rm l} - \mathbf{f}_{\rm r}), \tag{51}$$

where $\mathbf{f}_{\rm l} = \mathbf{f}(\mathbf{q}_{\rm l})$ and $\mathbf{f}_{\rm r} = \mathbf{f}(\mathbf{q}_{\rm r})$. We express now the integral

$$\begin{aligned} \int\limits_{\rm x_l}^{\rm x_r} \mathbf{q}(x, T){\rm dx} &= \int\limits_{\rm x_l}^{\rm Ts_l} \mathbf{q}(x, T){\rm dx} \\ &+ \int\limits_{\rm Ts_l}^{\rm Ts_r} \mathbf{q}(x, T){\rm dx} + \int\limits_{\rm Ts_r}^{\rm x_r} \mathbf{q}(x, T){\rm dx}. \end{aligned} \tag{52}$$

Hence evaluating the first and third integrals on the right hand side, we find

$$\int_{x_l}^{x_r} \mathbf{q}(x,T)\mathrm{dx} = \int_{Ts_l}^{Ts_r} \mathbf{q}(x,T)\mathrm{dx} + (Ts_l - x_l)\mathbf{q}_l + (x_r - Ts_r)\mathbf{q}_r. \tag{53}$$

Comparing Eqs. (51) and (53), we get

$$\int_{Ts_l}^{Ts_r} \mathbf{q}(x,T)\mathrm{dx} = T(s_r\mathbf{q}_r - s_l\mathbf{q}_l + \mathbf{f}_l - \mathbf{f}_r). \tag{54}$$

We divide now the above expression by $T(s_r - s_l)$ and introduce an average state

$$\mathbf{q}^{\mathrm{HLL}} \equiv \frac{1}{T(s_r - s_l)} \int_{Ts_l}^{Ts_r} \mathbf{q}(x,T)\,\mathrm{dx} = \frac{s_r\mathbf{q}_r - s_l\mathbf{q}_l + \mathbf{f}_l - \mathbf{f}_r}{s_r - s_l}. \tag{55}$$

Note that the above HLL average state, $\mathbf{q}^{\mathrm{HLL}}$, is not used for flux evaluation. We express this flux as follows. Applying the integral form of the conservation law to the control volume $[x_l, 0] \times [0, T]$, we get

$$\int_{Ts_l}^{0} \mathbf{q}(x,T)\mathrm{dx} = -Ts_l\mathbf{q}_l + T(\mathbf{f}_l - \mathbf{f}_{0l})\,. \tag{56}$$

Here $\mathbf{f}_{0l}$ is the flux at $x = 0$. Hence we get

$$\mathbf{f}_{0l} = \mathbf{f}_l - s_l\mathbf{q}_l - \frac{1}{T}\int_{Ts_l}^{0} \mathbf{q}(x,T)\mathrm{dx}. \tag{57}$$

In the above expression we replace $\mathbf{q}(x,T)$ by $\mathbf{q}^{\mathrm{HLL}}$. As a result of that we get

$$\mathbf{f}_{0l} = \mathbf{f}_l + s_l(\mathbf{q}^{\mathrm{HLL}} - \mathbf{q}_l). \tag{58}$$

Note that this expression can be obtained from the Rankine-Hugoniot conditions

$$\mathbf{f}_{0l} - \mathbf{f}_l = s_l(\mathbf{q}^{\mathrm{HLL}} - \mathbf{q}_l), \tag{59}$$

$$\mathbf{f}_r - \mathbf{f}_{0l} = s_r(\mathbf{q}_r - \mathbf{q}^{\mathrm{HLL}}). \tag{60}$$

Replacing $\mathbf{f}_{0l}$ by $\mathbf{f}^{\mathrm{HLL}}$ in Eq. (58) and using Eq. (55), we find that the flux, corresponding to the HLL state, is

$$\mathbf{f}^{\mathrm{HLL}} = \frac{s_r\mathbf{f}_l - s_l\mathbf{f}r + s_rs_l(\mathbf{q}_r - \mathbf{q}_l)}{s_r - s_l}. \tag{61}$$

In summary, the HLL intercell numerical flux is

$$\mathbf{f}^{\mathrm{HLL}}_{i+1/2} = \begin{cases} \mathbf{f}_l, & 0 \le s_l, \\ \mathbf{f}^{\mathrm{HLL}}, & s_l \le 0 \le s_r, \\ \mathbf{f}_r, & 0 \ge s_r. \end{cases} \tag{62}$$

The HLL solver is very simple and entropy satisfying [2]. This solver performs well at critical (sonic) rarefactions, the left (right) hand side of which moves to the left (right) with velocities higher than the sound speed. As middle waves are ignored in the HLL solver an excessive smearing of contact waves and vortices occur. The HLL Riemann solver is exact for a system of two equations such as shallow water equations [2]. For this solver wave speed estimates are still required. We discuss this issue in Subsec. 4.3.

4.2. The HLLC Riemann solver. The HLLC Riemann solver is a modification of the HLL Riemann solver. Here C stands for a contact wave. Contact and shear waves, missing in the HLL solver, are taken into account in the HLLC solver [2]. As a result, the star region contains two sub-regions, q_{*l}, q_{*r}, (Fig. 2). We have

$$\int_{Ts_l}^{Ts_r} \mathbf{q}(x,T)\mathrm{dx} = \int_{Ts_l}^{Ts_*} \mathbf{q}(x,T)\mathrm{dx} + \int_{Ts_*}^{Ts_r} \mathbf{q}(x,T)\mathrm{dx}. \tag{63}$$

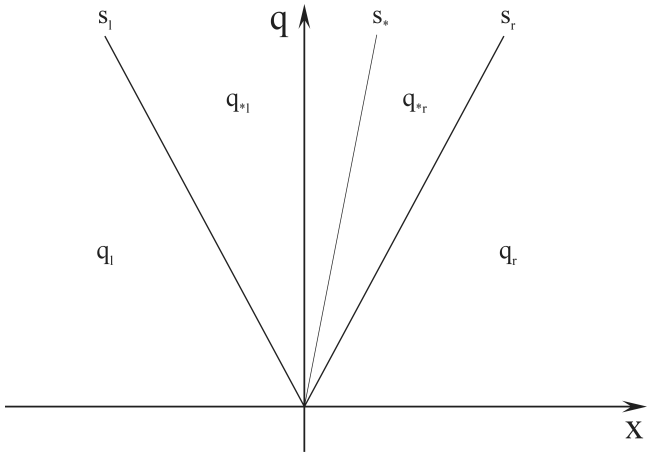

Fig. 2. The HLLC Riemann fan

We define now

$$\mathbf{q}_{*l} = \frac{1}{T(s_* - s_l)} \int_{Ts_l}^{Ts_*} \mathbf{q}(x,T)\mathrm{dx}, \tag{64}$$

$$\mathbf{q}_{*r} = \frac{1}{T(s_r - s_*)} \int_{Ts_*}^{Ts_r} \mathbf{q}(x,T)\mathrm{dx}. \tag{65}$$

Applying the Rankine-Hugoniot conditions across each of the waves of speeds s_l, s_* and s_r, we get

$$\mathbf{f}_{*l} = \mathbf{f}_l + s_l(\mathbf{q}_{*l} - \mathbf{q}_l), \tag{66}$$

$$\mathbf{f}_{*r} = \mathbf{f}_{*l} + s_*(\mathbf{q}_{*r} - \mathbf{q}_{*l}), \tag{67}$$

$$\mathbf{f}_r = \mathbf{f}_{*r} + s_r(\mathbf{q}_r - \mathbf{q}_{*r}). \tag{68}$$

Note that there are three equations for the four unknown vectors $\mathbf{q}_{*l}$, $\mathbf{f}_{*l}$, $\mathbf{q}_{*r}$, and $\mathbf{f}_{*r}$. So, there are more unknowns than equations.

The additional conditions for the HLLC Riemann solver are

$$u_{*l} = u_{*r} = u_* = s_*, \qquad p_{*l} = p_{*r} = p_*. \tag{69}$$

From Eqs. (66) and (68) we find that

$$s_{\rm l}\mathbf{q}_{*\rm l} - \mathbf{f}_{*\rm l} = s_{\rm l}\mathbf{q}_{\rm l} - \mathbf{f}_{\rm l}, \tag{70}$$

$$s_{\rm r}\mathbf{q}_{*\rm r} - \mathbf{f}_{*\rm r} = s_{\rm r}\mathbf{q}_{\rm r} - \mathbf{f}_{\rm r}. \tag{71}$$

From the first and second components of these equations we get

$$\begin{aligned} p_{*\rm l} &= p_* = p_{\rm l} + \varrho_{\rm l}(s_{\rm l} - u_{\rm l})(s_* - u_{\rm l}),\\ p_{*\rm r} &= p_* = p_{\rm r} + \varrho_{\rm r}(s_{\rm r} - u_{\rm r})(s_* - u_{\rm r}). \end{aligned} \tag{72}$$

Hence we obtain

$$s_* = \frac{p_{\rm r} - p_{\rm l} + \varrho_{\rm l}u_{\rm l}(s_{\rm l} - u_{\rm l}) - \varrho_{\rm r}u_{\rm r}(s_{\rm r} - u_{\rm r})}{\varrho_{\rm l}(s_{\rm l} - u_{\rm l}) - \varrho_{\rm r}(s_{\rm r} - u_{\rm r})}. \tag{73}$$

From Eqs. (70) and (71) with the use of Eq. (72) we find

$$\mathbf{f}_{*\rm k} = \mathbf{f}_{\rm k} + s_{\rm k}(\mathbf{q}_{*\rm k} - \mathbf{q}_{\rm k}), \qquad k = l, r. \tag{74}$$

Here the intermediate states $\mathbf{q}_{*\rm k}$ are

$$\mathbf{q}_{*\rm k} = \varrho_{\rm k}\frac{s_{\rm k} - u_{\rm k}}{s_{\rm k} - s_*}\begin{bmatrix} 1 \\ s_* \\ \dfrac{E_{\rm k}}{\varrho_{\rm k}} + (s_* - u_{\rm k})\left(s_* + \dfrac{p_{\rm k}}{\varrho_{\rm k}(s_{\rm k} - u_{\rm k})}\right) \end{bmatrix}, \quad k = l, r. \tag{75}$$

In summary, the HLLC intercell numerical flux is

$$\mathbf{f}^{\rm HLLC}_{\rm i+1/2} = \begin{cases} \mathbf{f}_{\rm l}, & 0 < s_{\rm l},\\ \mathbf{f}_{*\rm l}, & s_{\rm l} \le 0 \le S_*,\\ \mathbf{f}_{*\rm r}, & S_* \le 0 \le s_{\rm r},\\ \mathbf{f}_{\rm r}, & 0 \ge s_{\rm r}. \end{cases} \tag{76}$$

4.3. Wave speed estimates. We need estimates for wave speeds $s_{\rm l}$ and $s_{\rm r}$. The use of the eigenvalues,

$$\lambda^{(1)} = u - c_{\rm s}, \qquad \lambda^{(2)} = u, \qquad \lambda^{(3)} = u + c_{\rm s}, \tag{77}$$

is unproductive and therefore it is not recommended. Instead, we can use information from other Riemann solvers, e.g., the Roe average eigenvalues [12], which works well [2]. As a result, we imply

$$s_1 = \widetilde{u} - \widetilde{c}_{\rm s}, \qquad s_{\rm r} = \widetilde{u} + \widetilde{c}_{\rm s}, \tag{78}$$

where $\widetilde{u}$ and $\widetilde{c}_{\rm s}$ are the Roe-averaged quantities as

$$\begin{aligned} \widetilde{u} &= \frac{\sqrt{\varrho_{\rm l}}u_1 + \sqrt{\varrho_{\rm r}}u_{\rm r}}{\sqrt{\varrho_{\rm l}} + \sqrt{\varrho_{\rm r}}},\\ \widetilde{c}_{\rm s}^2 &= (\gamma - 1)\left(\widetilde{H} - \frac{\widetilde{u}^2}{2}\right) \end{aligned} \tag{79}$$

with $\varrho_{\rm l}$ and $\varrho_{\rm r}$ denoting the left and right (to the interface) values of mass density. Here the enthalpy $H = (E + p)/\varrho$ is averaged as

$$\widetilde{H} = \frac{\sqrt{\varrho_{\rm l}}H_1 + \sqrt{\varrho_{\rm r}}H_{\rm r}}{\sqrt{\varrho_{\rm l}} + \sqrt{\varrho_{\rm r}}}. \tag{80}$$

The other possibility is to adopt the expression for the gas pressure p_*. See [2] for details.

5. Higher-order numerical schemes for hyperbolic equations

As the first-order numerical methods, introduced in Sec. 3, are too diffusive and therefore impractical to use we aim here to construct high-resolution schemes. These schemes should be at least second-order accurate in smooth regions of the solution, be free from spurious oscillations, and give high-resolution of large-gradient regions. There are a number of higher-order methods. For a review see, e.g. [2, 13]. In this paper we limit our discussion to the MUSCL-Hancock method as this method is recognized as very robust and therefore it is widely used in many modern numerical codes. Examples of such codes are NUMERIKA [2], FLASH [14], ATHENA [15], GAMER [16], and others.

5.1. MUSCL method. The low accuracy and the complexity of Godunov's method meant that other methods needed to be developed. Such development effort was undertaken by Kolgan [3] 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 were due to van Leer [4] who extended Godunov's approach to second-order spatial accuracy by the MUSCL approach. See also [5]. Van Leer's 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 upwind 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 (e.g. [12]). 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 [4]. In this scheme the accuracy was increased by constructing a piecewise linear approximation of $\mathbf{q}(x,t)$, viz.

$$\begin{aligned} \overline{\mathbf{q}}(x,t) &= \mathbf{q}_{\rm i} + \mathbf{s}_{\rm i}(x - \overline{x}_{\rm i}),\\ x_{\rm i-1/2} &< x < x_{\rm i+1/2}. \end{aligned} \tag{81}$$

Here $\mathbf{s}_{\rm i}$ is a slope and $\overline{x}_{\rm i} = (x_{\rm i} + x_{\rm i+1})/2 = x_{\rm i} + \Delta x/2$ is the center of the grid cell. So, $\overline{\mathbf{q}}(\overline{x}_{\rm i}, t) = \mathbf{q}_{\rm i}$. Moreover, it is required that the average value of $\overline{\mathbf{q}}(x,t)$ over the cell is equal to $\mathbf{q}_{\rm i}$. The slope $\mathbf{s}_{\rm i}$ can be constructed by a number of ways, e.g.

$$\mathbf{s}_{\rm i} = \frac{\mathbf{q}_{\rm i+1} - \mathbf{q}_{\rm i-1}}{2\Delta x} \quad \text{(centered slope, Fromm's scheme)}, \tag{82}$$

$$\mathbf{s}_{\rm i} = \frac{\mathbf{q}_{\rm i} - \mathbf{q}_{\rm i-1}}{\Delta x} \quad \text{(upwind slope, Beam-Warming scheme)}, \tag{83}$$

$$\mathbf{s}_{\rm i} = \frac{\mathbf{q}_{\rm i+1} - \mathbf{q}_{\rm i}}{\Delta x} \quad \text{(downwind slope, Lax-Wendroff scheme)}, \tag{84}$$

$$\mathbf{s}_i = \text{minmod}\left(\frac{\mathbf{q}_i - \mathbf{q}_{i-1}}{\Delta x}, \frac{\mathbf{q}_{i+1} - \mathbf{q}_i}{\Delta x}\right) \quad (\text{minmod slope}). \tag{85}$$

Here the minmod function,

$$minmod(a,b) = \begin{cases} a, & \text{for } |a| < |b| \text{ and } ab > 0, \\ b, & \text{for } |a| > |b| \text{ and } ab > 0, \\ 0, & \text{for } ab \le 0, \end{cases} \tag{86}$$

returns the smallest argument in magnitude if the arguments are of the same sign, and zero if they are not.

For other choices of slopes see, for instance, [2, 12]. Note that choosing $\mathbf{s}_i = 0$ in the above expressions leads to Godunov's method [1].

5.2. MUSCL-Hancock scheme. As the original MUSCL method [4] was modified in 1980 by UC Berkeley graduate student of fluid mechanics, Steve Hancock, the modified scheme bears a common name MUSCL-Hancock method. See [6].

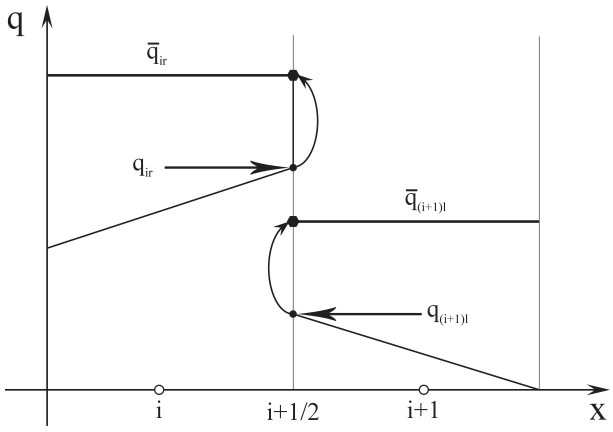

Fig. 3. Boundary extrapolated values

The MUSCL-Hancock method consists of the following steps:

1. **Data reconstruction and boundary extrapolated values.** At this stage we reconstruct linear subcell distributions of a (non-conservative or conservative) vector state $\mathbf{w}_i$ and compute intercell values from

$$\mathbf{w}_{il} = \mathbf{w}_i - \frac{1}{2}\Delta x\, \mathbf{s}_i, \qquad \mathbf{w}_{ir} = \mathbf{w}_i + \frac{1}{2}\Delta x\, \mathbf{s}_i. \tag{87}$$

Here $\mathbf{w}_{il}$ ($\mathbf{w}_{ir}$) denotes $\mathbf{w}$ at the left (right) intercell of the i-th grid, $\mathbf{s}_i$ is a limited slope which is introduced in Eq. (81);

2. **Evolution of the boundary values.** We let the boundary extrapolated values evolve by a time $\frac{1}{2}\Delta t$ according to

$$\overline{\mathbf{w}}_{il} = \mathbf{w}_{il} + \frac{\Delta t}{2\Delta x}\left[\mathbf{f}\left(\mathbf{w}_{il}\right) - \mathbf{f}\left(\mathbf{w}_{ir}\right)\right], \tag{88}$$

$$\overline{\mathbf{w}}_{ir} = \mathbf{w}_{ir} + \frac{\Delta t}{2\Delta x}\left[\mathbf{f}\left(\mathbf{w}_{il}\right) - \mathbf{f}\left(\mathbf{w}_{ir}\right)\right]. \tag{89}$$

3. **The Riemann problem.** We solve the Riemann Problem with data

$$\mathbf{w}_{il*} = \overline{\mathbf{w}}_{ir}, \qquad \mathbf{w}_{ir*} = \overline{\mathbf{w}}_{(i+1)l}. \tag{90}$$

Hence, we compute the intercell flux $\mathbf{f}_{i+\frac{1}{2}}\left(\mathbf{w}_{il*}, \mathbf{w}_{ir*}\right)$ with a use of the similarity solution, $\mathbf{w}_{i+\frac{1}{2}}\left(\frac{x}{t}\right)$,

$$\mathbf{f}_{i+\frac{1}{2}} = \mathbf{f}\left(\mathbf{w}_{i+\frac{1}{2}}\left(0\right)\right). \tag{91}$$

This flux is used in the discretized version of the Euler equations to evaluate the updated value of the vector state, $\mathbf{w}_i^{n+1}$.

6. C++ coding of the MUSCL-Hancock method for the Euler equations

We organize this section by presenting the MUSCL-Hancock method for one-dimensional Euler equations which is the set of three nonlinear hyperbolic equations given by Eqs. (1), (6), (7). First we review the numerical code and later on present results of the numerical test we perform.

All results showed in this section are obtained with the use of the original code which was written in C++. All numerical simulations are performed on a PC computer with the use of g++ compiler. A typical numerical session lasted a dozen or so minutes on Intel Core 2 Duo T8300 (2.4 GHz) CPU with 3 GB of RAM. The obtained numerical data was visualized with the use of Interactive Data Language (IDL)[1] scripts which allowed to visualize the results of numerical simulations in the form af animations (stored here in avi format) or at given moments of time in the form of postscript files which were adopted for presentation of the results in this paper.

The code Euler1d_MH.cpp which solves the Euler equations consists of the following routines:

- readIniFile – reads parameters of the problem from the file advect.ini
- setInitialConditions – specifies the initial conditions
- setBoundaryConditions – specifies the boundary conditions
- setCFLConditions – imposes Courant-Friedrichs-Lewy (CFL) condition
- estimate – computes wave speed estimates for HLLC Riemann solver
- fluxVal – computes flux vector components from the components of the vector state of conserved variables
- computeIntercellFluxes – computes intercell numerical fluxes
- evolve_q – evolves the solution to a new time level using the explicit conservative formula
- output – dump the numerical data to files euler.xxx
- evolveInTime – main loop, sets the boundary conditions and performs iteration in time
- saveConfigIdl – writes to file config.ini the number of output files used by the IDL script movie.pro
- main – calls the required routines

[1] http://www.ittvis.com/ProductServices/IDL.aspx

This code implements several slope limiters, such as: Godunov, Fromm, Superbee, van Leer, van Albada, MinMod and MinMax. Other slope limiters can be easily implemented. As a default initial condition a user can optionally set values of mass density, velocity and gas pressure in the left, middle and right sections which are divided by two diaphragms. Spatial positions of diaphragms can be set arbitrarily. For the numerical results presented in this paper we set these diaphragms at $x = 4$ and $x = 6$. See Eq. (94).

We perform numerical simulations with the use of the code Euler1d_MH.cpp. At $t = 0$ s we specify initial conditions as

$$\varrho(x, t = 0) = 1, \tag{92}$$

$$u(x, t = 0) = 0, \tag{93}$$

$$p(x, t = 0) = \begin{cases} 1 & x < 4, \\ 1.25, & 4 \leq x \leq 6, \\ 1, & x > 6. \end{cases} \tag{94}$$

Such initial conditions, displayed in Fig. 4 (top-left panel), results in counter-propagating waves. The leading signals correspond to shocks represented by discontinuities in gas pressure profiles (middle-left). At $t = 1$ s (top-right panel) and $t = 2$ s (middle-left) these shocks are located at $x \cong 2.5$, $x \cong 7.5$ and $x \cong 1.5$, $x \cong 8.5$, respectively. These shocks are followed by rarefaction waves which at $t = 2$ s are located at $x \cong 3$ and $x \cong 7$ (middle-left). For $t > 5$ s behind the shocks and rarefaction waves contact waves are present. These contact (or entropy) waves settle as stationary structures which are well observed at $x = 4$ and $x = 6$ for $t \geq 1$ s. Note that the final product of the initial pressure perturbation is a rarefied gas in the region where the initial perturbation was launched. This rarefied region corresponds to a new equilibrium which remains a good evidence of the initial perturbation. In conclusion, the initial perturbation of Eqs. (92)–(94) results in stationary contact (or entropy) waves which bounds a region of permanently rarefied plasma. Therefore, the above proposed test can serve as a good check of a performance of a numerical code.

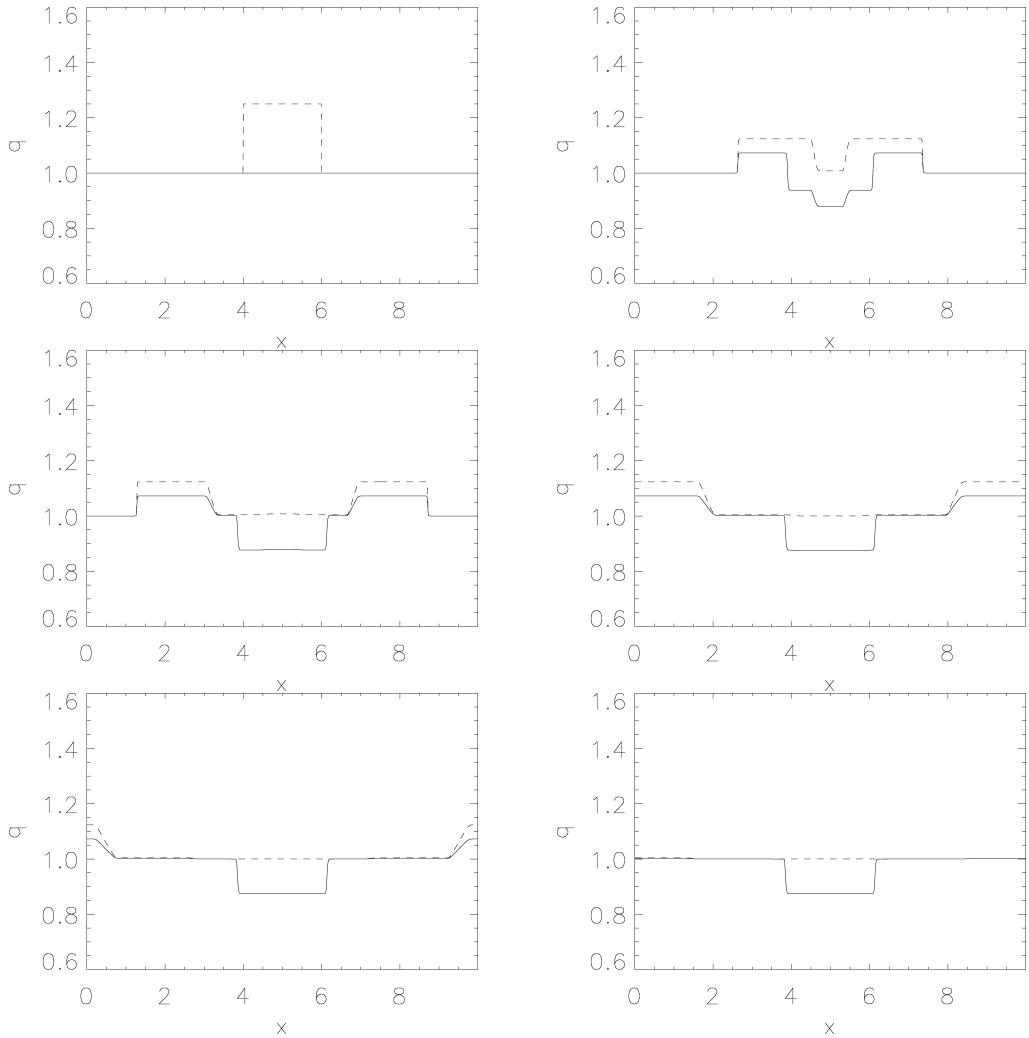

Fig. 4. Mass density (solid line) and gas pressure (dashed line) profiles of the Euler equations for SUPERBEE slope limiter and 1000 grid cells. Profiles are displayed at t=0 s (top-left panel), t=1 s, t=2 s, t=3 s, t=4 s and t=5 s (bottom-right panel)

7. Summary and conclusions

In this paper we described the MUSCL-Hancock method for numerical solutions of the initial-value problem for the Euler equations and its implementation into the C++ code. Additionally, we presented theory of the Euler equations. In particular, we introduced the Riemann problem for Euler equations and reviewed three solvers: Godunov's, HLL and HLLC. In the following section we presented the numerical methods we used, their original implementation and description of the original C++ code. We presented and described results of the numerical test we performed.

The approach presented in this paper can be extended along various ways. In particular, future works can consist of modification of the code for solving two- and three-dimensional Euler equations. Another option is to make it parallel and implement to work on graphical cards such as NVIDIA (CUDA) or ATi (OpenCL).

The authors express their thanks to the referee for his/her stimulating comment.

REFERENCES

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

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

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

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

[5] B. van Leer, "On the relation between the upwind-differencing schemes of Godunov, Engquist-Osher and Roe", *SIAM J. Sci. Statist. Comput.* 5 (1), 1–20 (1984).

[6] B. van Leer, "Upwind and high-resolution methods for compressible flow: from donor cell to residual distribution schemes", *Comm. Computational Physics* 1 (2), 192–206 (2006).

[7] S.A.E.G. Falle, "Self-similiar jets", *MNRAS* 250, 581–596 (1991).

[8] C. Berthon, "Stability of the MUSCL schemes for the Euler equations", *Comm. Math. Sciences* 3, 133–158 (2005).

[9] C. Berthon, "Why the MUSCL-Hancock scheme is L1-stable", *Numer. Math.* 104, 27–46 (2006).

[10] K. Murawski and D. Lee, "Numerical methods of solving equations of hydrodynamics from perspectives of the code FLASH", *Bull. Pol. Ac.: Tech.* 59 (3), 81–92 (2011).

[11] K. Murawski, "Numerical solutions of magnetohydrodynamic equations", *Bull. Pol. Ac.: Tech.* 59 (2), 219–226 (2011).

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

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

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

[15] J.M. Stone, "The Athena MHD code: extensions, applications, and comparisons to ZEUS", *Numerical Modeling of Space Plasma Flows: ASTRONUM-2008 ASP Conf. Series* 406, 277–281 (2009).

[16] H.-Yu Schive, Yu-C. Tsai, T. Chiueh, "GAMER: A graphic processing unit accelerated adaptive-mesh-refinement code for astrophysics", *Astrophys. J. Suppl.* 186 (2), 457–484 (2010).