Rankine-Hugoniot-Riemann solver for steady multidimensional conservation laws with source terms

Halvor Lund[a,b,c,*], Florian Müller[a], Bernhard Müller[b], Patrick Jenny[a]

[a] *Institute of Fluid Dynamics, ETH Zürich, Sonneggstrasse 3, 8092 Zürich, Switzerland*
[b] *Dept. of Energy and Process Engineering, Norwegian University of Science and Technology (NTNU), NO-7491 Trondheim, Norway*
[c] *SINTEF Energy Research, NO-7465 Trondheim, Norway*

Abstract

The *Rankine-Hugoniot-Riemann* (RHR) solver has been designed to solve steady multidimensional conservation laws with source terms. The solver uses a novel way of incorporating cross fluxes as source terms. The combined source term from the cross fluxes and normal source terms is imposed in the middle of a cell, causing a jump in the solution according to the Rankine-Hugoniot condition. The resulting Riemann problems at the cell faces are then solved by a conventional Riemann solver.

We prove that the solver is of second order accuracy for rectangular grids and confirm this by its application to the 2D scalar advection equation, the 2D isothermal Euler equations and the 2D shallow water equations. For these cases, the error of the RHR solver is comparable to or smaller than that of a standard Riemann solver with a MUSCL scheme. The RHR solver is also applied to the 2D full Euler equations for a channel flow with injection, and shown to be comparable to a MUSCL solver.

Keywords: finite volume methods, partial differential equations, conservation laws, Rankine-Hugoniot condition, source terms
2010 MSC: 76M12, 65N08, 35L65

1. Introduction

Our goal has been to develop a numerical method to solve systems of multidimensional hyperbolic partial differential equations (PDEs) with source terms, with an emphasis on calculating steady states accurately. Such systems of equations can describe a number of physical phenomena, e.g. combustion [1], multiphase flow with phase interaction in the form of mass or heat transfer [2, 3], water/vapour flow in nuclear reactors [4], cavitation

*Corresponding author. Phone +47 73597200.
Email addresses: halvor.lund@ntnu.no (Halvor Lund), florian.mueller@sam.math.ethz.ch (Florian Müller), bernhard.muller@ntnu.no (Bernhard Müller), jenny@ifd.mavt.ethz.ch (Patrick Jenny)

[5], shallow water flow over variable topography [6, 7], and fluid flow in a gravity field [8], to mention a few. In many cases, one can express the equations as balance laws consisting of a conservation law together with a source term, i.e. as

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{\nabla} \cdot \mathcal{F}(\boldsymbol{u}) = \boldsymbol{q}(\boldsymbol{u}, \boldsymbol{x}), \tag{1}$$

where $\boldsymbol{u}$ denotes the vector of conserved variables, $\mathcal{F}(\boldsymbol{u})$ the flux tensor and $\boldsymbol{q}(\boldsymbol{u}, \boldsymbol{x})$ the source term. The procedure of solving such equations numerically in multiple dimensions involves a number of challenges compared to solving a one-dimensional homogeneous ($\boldsymbol{q} = 0$) conservation law. First, the multidimensionality introduces new effects, which may be difficult to capture accurately by just using standard one-dimensional methods based on approximate Riemann solvers. Second, a stiff source term with a magnitude similar to the flux gradients may require a whole new approach, since approximate Riemann solvers for the numerical flux assume small or vanishing source terms.

One common approach to solving equations of the form (1) is to use a *fractional-step* or *operator-splitting* method, which is based on solving the conservation law $\boldsymbol{u}_t + \boldsymbol{\nabla} \cdot \mathcal{F}(\boldsymbol{u}) = 0$ and the ordinary differential equation (ODE) $\boldsymbol{u}_t = \boldsymbol{q}(\boldsymbol{u}, \boldsymbol{x})$ alternately to approximate the solution of the full problem (1). The advantage of such a splitting approach is that the operators can be approximated using well-proven methods developed for homogeneous conservation laws and for ODEs, respectively. However, as e.g. LeVeque [9, Chap. 17] points out, such splitting encounters difficulties, especially when the flux gradients and the source terms nearly or completely balance each other. This drawback of operator splitting has given rise to the development of *well-balanced* schemes, whose main aim is to well approximate the balance of the flux surface integral and the source volume integral in steady state.

Well-balanced schemes have been discussed by a number of authors, including Bale et al. [7], Bermudez and Vazquez [10], Donat and Martinez-Gavara [11], Gosse [4], Hubbard and García-Navarro [12], and LeVeque [6, 13]. Research on well-balanced schemes for the shallow water equations (SWE) has been particularly active field. Audusse et al. [14] present a scheme for the SWE with topography which is well-balanced for any numerical flux, ensures non-negative water height, and has been highly influential for other schemes in the past ten years. Murillo and García-Navarro [15] solve the SWE with source terms by adding an extra wave associated with the source term in their approximate Riemann solver. Noelle and co-workers [16–19] have focused on high order well-balanced methods and considered the application to the SWE and steady states with moving flow. Lukáčová-Medvid'ová et al. [20] present a finite volume evolution Galerkin (FVEG) scheme for the SWE which takes multidimensional effects explicitly into account by allowing wave propagation in all directions, not just parallel to the axes. Ricchiuto and Bollermann [21] develop a scheme for the SWE based on the quite recent Residual Distribution (RD) framework. Parés [22] gives a theoretical framework for designing well-balanced schemes, which also deals with the effects of source terms at cell boundaries.

LeVeque [6] proposed a method which incorporates the source term as a singular source in the centre of each grid cell, so that the flux difference exactly equals the source term integ-

ral approximation. This in turn leads to altered Riemann problems at the cell boundaries, which can be solved using a standard approximate Riemann solver with first or higher order reconstruction. Jenny and Müller [1] used a similar idea, but rather placed the source term at the cell boundary. Their work also introduced the concept for 2D problems of treating the flux gradients in the y direction as source terms when solving the Riemann problems in the x direction, and vice versa. This solver was coined the *Rankine-Hugoniot-Riemann* (RHR) solver, since a Rankine-Hugoniot condition is combined with a Riemann solver to calculate the new Riemann problem with source term at the cell boundary.

In this paper we build on the idea by Jenny and Müller [1] of treating cross fluxes as source terms, combined with placing the source term in the cell centre, as proposed by LeVeque [6]. This allows us to develop a numerical scheme with accurate treatment of multidimensional effects as well as source terms. Some stability problems reported by Jenny and Müller [1] for two-dimensional cases are eliminated here by introducing a novel limiter.

Our paper is organized as follows. In Section 2, we explain the Rankine-Hugoniot-Riemann solver for one and two dimensions. The method can easily be extended to three-dimensional problems. We then introduce a limiting procedure to preserve TVD-like properties and to eliminate instabilities. In Section 3 we present an analysis of the solver properties and show that it is of second order spatial accuracy for rectangular grids. Numerical investigations are presented in Section 4, where we apply the RHR solver to steady states for a 2D scalar advection equation, the 2D isothermal Euler equations, the 2D shallow water equations and the 2D full Euler equations. The numerical error is compared to that of a second-order MUSCL scheme. Finally, in Section 5 we draw some conclusions and outline further work.

2. Rankine-Hugoniot-Riemann solver

We are interested in solving a system of two-dimensional conservation laws with source terms, formulated in the steady case as

$$\frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} + \frac{\partial \boldsymbol{g}(\boldsymbol{u})}{\partial y} = \boldsymbol{q}(\boldsymbol{u}, \boldsymbol{x}), \tag{2}$$

where $\boldsymbol{u}$ denotes the vector of conserved variables, $\boldsymbol{f}$ and $\boldsymbol{g}$ the flux vectors in x- and y-direction, respectively, and $\boldsymbol{q}$ the source term vector. In [1], the RHR solver was applied to a 1D premixed laminar flame and a 2D laminar Bunsen flame, where the source term not only depends on the conserved variables $\boldsymbol{u}$, but also on $\nabla \boldsymbol{u}$. The homogeneous system is assumed to be hyperbolic, i.e. the matrix $n_x \boldsymbol{f}'(\boldsymbol{q}) + n_y \boldsymbol{g}'(\boldsymbol{q})$ is diagonalizable with real eigenvalues for all $(n_x, n_y) \in \mathbb{R}^2$. We will build on LeVeque's idea of implementing the source term as a singular source in the cell centre [6]. For simplicity and clarity, we shall first explain the RHR solver in one dimension. Then we continue with two dimensions, where the concept of cross fluxes as source terms is employed as suggested by Jenny and Müller [1].

2.1. One-dimensional solver

The Rankine-Hugoniot-Riemann (RHR) solver was first proposed by Jenny and Müller [1], while a similar method was presented by LeVeque [6]. They can both be applied to solve a one-dimensional conservation law with a source term, in the steady case written as

$$\frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} = \boldsymbol{q}(\boldsymbol{u}, x). \tag{3}$$

The two methods have the similarity that they incorporate the source term by placing it as a singular term at either the cell face or the cell centre, and then using the Rankine-Hugoniot condition to calculate the jump in the solution due to this singular source.

In this work we use the source term treatment by LeVeque [6], who distributed the source term as a singular term to the cell centre. The strength of the source term in cell i integrated over the whole cell is approximated by $\Delta x \boldsymbol{q}_i$. Therefore the Rankine-Hugoniot condition reads

$$\boldsymbol{f}(\boldsymbol{u}_{i,\mathrm{E}}) - \boldsymbol{f}(\boldsymbol{u}_{i,\mathrm{W}}) = \Delta x \boldsymbol{q}_i, \tag{4}$$

where $\boldsymbol{u}_{i,\mathrm{W}}$ and $\boldsymbol{u}_{i,\mathrm{E}}$ are the values in the western and eastern cell parts, respectively, cf. Fig. 1. To keep the method conservative, we also require that the average of the conservative variable in the cell is kept constant, i.e. that

$$\frac{1}{2}(\boldsymbol{u}_{i,\mathrm{E}} + \boldsymbol{u}_{i,\mathrm{W}}) = \boldsymbol{u}_i. \tag{5}$$

The reconstruction of $\boldsymbol{u}$ is illustrated in Figure 1. The new half-states are then used to solve the Riemann problems at each cell face, e.g. the Riemann problem at the face $I_{i+1/2}$ is given by $\boldsymbol{u}_{i,\mathrm{E}}$ and $\boldsymbol{u}_{i+1,\mathrm{W}}$ as the left and right states, respectively.

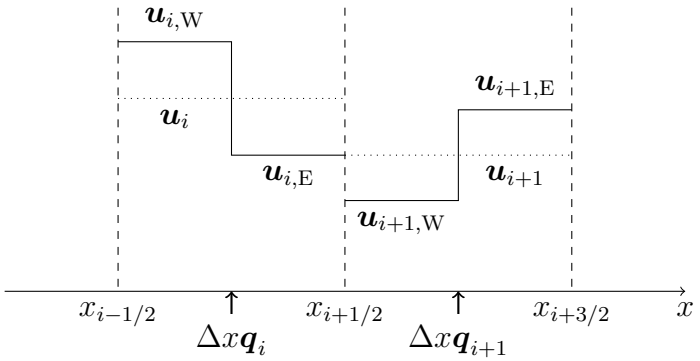


Figure 1: RHR solver in one dimension. The locations of the singular source terms are illustrated, as well as the cell averaged states (dotted lines) and the reconstruction (solid lines) of $\boldsymbol{u}$.

LeVeque [6] demonstrated that this approach is well-balanced, with an emphasis on the shallow water equations. Bale et al. [7] and LeVeque [23] argue that source term

singularities placed at the cell faces instead of in the cell centres are more robust and simpler to implement. However, according to our experience, the method with cell centred singularities introduced in the present section has proven both fruitful and relatively easy to implement, since one can use a standard Riemann solver at the cell faces.

2.2. Multidimensional solver

We will now extend the ideas presented in Section 2.1 to multidimensional problems, formulated for steady states. For simplicity, we will derive a solver for two dimensions, but it is straightforward to extend the method to three dimensions. In two dimensions, a system of conservation laws with source terms can be formulated as written in Eq. (2). Moving the last left-hand-side term to the right-hand-side yields

$$\frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} = -\frac{\partial \boldsymbol{g}(\boldsymbol{u})}{\partial y} + \boldsymbol{q}(\boldsymbol{u}, \boldsymbol{x}). \tag{6}$$

From this equation one can readily see that the y-directed flux term may be seen as a source term when solving the system in the x-direction, and vice versa. This idea of treating the cross flux as a source term was first introduced by Jenny and Müller [1], who placed the source terms at the cell faces. We will, however, continue to develop the idea of the cell centred source term described in Section 2.1, but incorporating both the source term $\boldsymbol{q}$ *and* the cross-flux term $-\partial \boldsymbol{g}(\boldsymbol{u})/\partial y$.

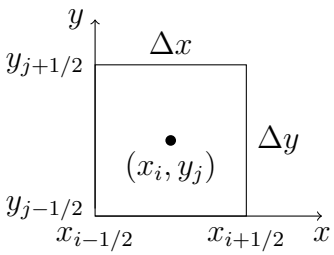


Figure 2: Sketch of cell $\mathcal{C}_{i,j}$

To find a finite volume formulation, we integrate Eq. (6) over a rectangular control volume $\mathcal{C}_{i,j}$ defined by the opposite corners $(x_{i-1/2}, y_{j-1/2})$ and $(x_{i+1/2}, y_{j+1/2}) = (x_{i-1/2} + \Delta x, y_{j-1/2} + \Delta y)$, cf. Fig. 2, which yields

$$\frac{1}{\Delta x}(\boldsymbol{f}_{i+1/2,j} - \boldsymbol{f}_{i-1/2,j}) = -\frac{1}{\Delta y}(\boldsymbol{g}_{i,j+1/2} - \boldsymbol{g}_{i,j-1/2}) + \boldsymbol{q}_{i,j}, \tag{7}$$

where $\boldsymbol{u}_{i,j}$ and $\boldsymbol{q}_{i,j}$ are the averages of $\boldsymbol{u}$ and $\boldsymbol{q}$, respectively, over control volume $\mathcal{C}_{i,j}$. The approximate averaged fluxes at the western and southern faces are given by

$$\boldsymbol{f}_{i-1/2,j} \approx \frac{1}{\Delta y} \int_{y_{j-1/2}}^{y_{j+1/2}} \boldsymbol{f}(\boldsymbol{u}(x_{i-1/2}, y))\, \mathrm{d}y, \tag{8}$$

$$\boldsymbol{g}_{i,j-1/2} \approx \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \boldsymbol{g}(\boldsymbol{u}(x, y_{j-1/2}))\, \mathrm{d}x. \tag{9}$$

The approximation of $-\partial \boldsymbol{g}(\boldsymbol{u})/\partial y$ on the right hand side of Eq. (7) quickly reveals that this term may be treated as a source term, similar to $\boldsymbol{q}_{i,j}$, when calculating the $\boldsymbol{f}$-fluxes in the x-direction. We place this source term as a singular source in the centre of the cell, in a similar fashion as explained in Section 2.1. Conservativity in control volume $\mathcal{C}_{i,j}$ and the Rankine-Hugoniot conditions can then be expressed as

$$\frac{1}{2}(\boldsymbol{u}_{i,j,\mathrm{W}} + \boldsymbol{u}_{i,j,\mathrm{E}}) = \boldsymbol{u}_{i,j}, \tag{10}$$

$$\frac{1}{2}(\boldsymbol{u}_{i,j,\mathrm{S}} + \boldsymbol{u}_{i,j,\mathrm{N}}) = \boldsymbol{u}_{i,j}, \tag{11}$$

$$\frac{1}{\Delta x}(\boldsymbol{f}(\boldsymbol{u}_{i,j,\mathrm{E}}) - \boldsymbol{f}(\boldsymbol{u}_{i,j,\mathrm{W}})) = \boldsymbol{q}_{x,i,j} \equiv \frac{\Delta \boldsymbol{g}_{i,j}}{\Delta y} + \boldsymbol{q}_{i,j}, \tag{12}$$

$$\frac{1}{\Delta y}(\boldsymbol{g}(\boldsymbol{u}_{i,j,\mathrm{N}}) - \boldsymbol{g}(\boldsymbol{u}_{i,j,\mathrm{S}})) = \boldsymbol{q}_{y,i,j} \equiv \frac{\Delta \boldsymbol{f}_{i,j}}{\Delta x} + \boldsymbol{q}_{i,j}. \tag{13}$$

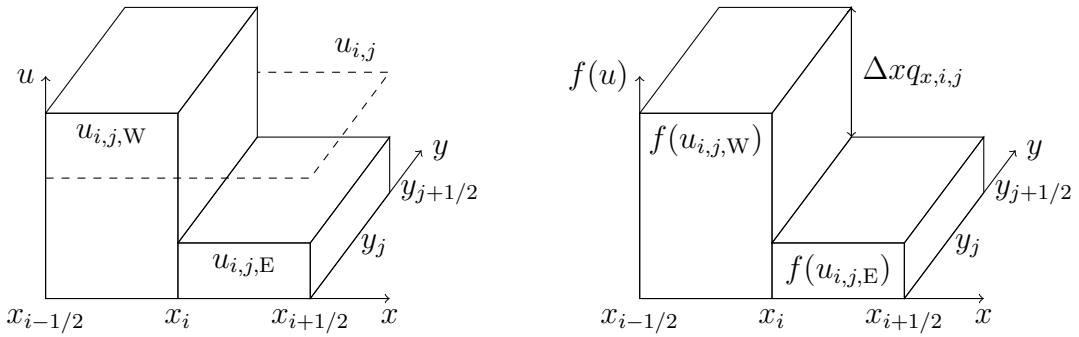
where the subscripts N/S/E/W denote the northern/southern/eastern/western parts of the cell. The flux differences are $\Delta \boldsymbol{f}_{i,j} = \boldsymbol{f}_{i-1/2,j} - \boldsymbol{f}_{i+1/2,j}$ and $\Delta \boldsymbol{g}_{i,j} = \boldsymbol{g}_{i,j-1/2} - \boldsymbol{g}_{i,j+1/2}$. We have also introduced $\boldsymbol{q}_{x,i,j}$ and $\boldsymbol{q}_{y,i,j}$ to denote the total source term in the x and y directions, respectively.

In a time stepping scheme, $\boldsymbol{q}_{i,j}$ is known from the old time level and the numerical fluxes $\boldsymbol{g}_{i,j\pm1/2}$ from the previous time level, as discussed in Section 4.1 below. Knowing the whole source term on the right hand side of Eq. (12), the half-states $\boldsymbol{u}_{i,j,\mathrm{E}}$ and $\boldsymbol{u}_{i,j,\mathrm{W}}$ of the 2D RHR solver can be determined in the same way as for the 1D RHR solver. To determine the half-states $\boldsymbol{u}_{i,j,\mathrm{S}}$ and $\boldsymbol{u}_{i,j,\mathrm{N}}$, the roles of $\boldsymbol{f}$ and $\boldsymbol{g}$ are simply reversed. The relations (10) and (12) defining $\boldsymbol{u}_{i,j,\mathrm{W}}$ and $\boldsymbol{u}_{i,j,\mathrm{E}}$ are sketched in Fig. 3. The states $\boldsymbol{u}_{i,j,\mathrm{E}}$ and $\boldsymbol{u}_{i+1,j,\mathrm{W}}$ then define the Riemann problem at the face $I_{i+1/2,j}$, which in turn can be used to calculate the flux $\boldsymbol{f}_{i+1/2,j}$ using a Riemann solver.

Since the fluxes depend on the adjacent states to be determined, the flux differences $\Delta \boldsymbol{g}_{i,j}$ in Eq. (12) and $\Delta \boldsymbol{f}_{i,j}$ in Eq. (13) are approximated by their known values at the previous time step when doing time stepping to reach the steady state, cf. Section 4.1. Thus, $\boldsymbol{q}_{x,i,j}$ in Eq. (12) and $\boldsymbol{q}_{y,i,j}$ in Eq. (13) are assumed to be known. In general, the conditions (10)–(13) may need to be solved numerically for $\boldsymbol{u}_{i,j,\mathrm{W}}$, $\boldsymbol{u}_{i,j,\mathrm{E}}$, $\boldsymbol{u}_{i,j,\mathrm{S}}$ and $\boldsymbol{u}_{i,j,\mathrm{N}}$, using e.g. Newton's method. However, for the 2D scalar linear advection equation and the 2D isothermal Euler equations presented later in this work, we are able to solve the conditions (10)–(13) analytically.

2.3. Limiting

The RHR solver presented by Jenny and Müller [1] was reported to have some stability problems when applied to two-dimensional balance laws. They handled these instabilities by introducing artificial numerical diffusion, which is rather arbitrary. In this paper we rather follow the ideas of Müller [24] and introduce a limiting of the western/eastern/southern/northern half-states.



(a) Condition (10) (b) Condition (12)

Figure 3: Sketch of the conditions (10) and (12) to define $\boldsymbol{u}_{i,j,\mathrm{W}}$ and $\boldsymbol{u}_{i,j,\mathrm{E}}$.

The variables we limit may either be the conserved variables or the primitive variables. For the isothermal Euler equations, the shallow water equations and the full Euler equations, we limit the primitive variables, i.e. the velocity components as well as the density and the water height, respectively. The limited state $\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{E}}$ is calculated as follows:

$$(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{E}})_k = \min\left[\max\left[(\boldsymbol{u}_{i,j,\mathrm{E}})_k - (\boldsymbol{u}_{i,j})_k, -|\delta_k|\right], |\delta_k|\right] + (\boldsymbol{u}_{i,j})_k \tag{14}$$

where $\delta_k = \mathrm{minmod}((\boldsymbol{u}_{i+1,j})_k - (\boldsymbol{u}_{i,j})_k, (\boldsymbol{u}_{i,j})_k - (\boldsymbol{u}_{i-1,j})_k)$. Here $(\cdot)_k$ denotes the k-th limited variable (i.e. the k-th component of the vector of the conserved or primitive variables), and $\boldsymbol{u}_{i,j,\mathrm{E}}$ is the unlimited eastern state. The limited western state $\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{W}}$ is then given by Eq. (10), so that the cell average is conserved. The limiter ensures that both $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{E}})_k$ and $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{W}})_k$ lie between $(\boldsymbol{u}_{i-1,j})_k$ and $(\boldsymbol{u}_{i+1,j})_k$, as long as $(\boldsymbol{u}_{i,j})_k$ also does so. The result would be completely equivalent if we limited the western state first, and then calculated the eastern state from this. An analogous requirement applies to the northern state $\boldsymbol{u}_{i,j,\mathrm{N}}$.

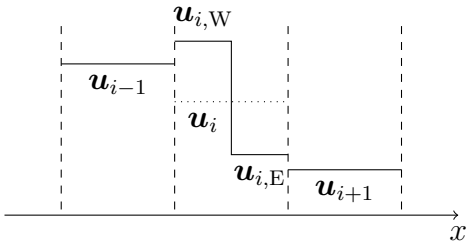
Figure 4 illustrates this limiting procedure. Fig. 4a shows an example of a possible result of solving the RHR relations (10)–(13). In this case, the state $\boldsymbol{u}_{i,\mathrm{W}}$ is out of bounds, since it is larger than both $\boldsymbol{u}_{i-1}$ and $\boldsymbol{u}_{i+1}$. The limiter is then applied, which results in the limited states $\boldsymbol{u}^{\mathrm{L}}_{i,\mathrm{W}} = \boldsymbol{u}_{i-1}$, and $\boldsymbol{u}^{\mathrm{L}}_{i,\mathrm{E}} = 2\boldsymbol{u}_i - \boldsymbol{u}^{\mathrm{L}}_{i,\mathrm{W}}$ (which follows from Eq. (10)), illustrated in Fig. 4b. A similar case is shown in Figs. 4c–4d, where $\boldsymbol{u}_{i,\mathrm{E}}$ is out of bounds, and hence the limiter reduces this to $\boldsymbol{u}^{\mathrm{L}}_{i,\mathrm{E}} = \boldsymbol{u}_{i-1}$.

In Figure 5a the instabilities without limiting are illustrated for the 2D steady scalar linear advection equation, i.e.

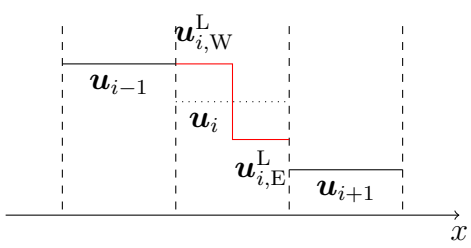
$$a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} = 0, \tag{15}$$

with constant velocity $\boldsymbol{v} = (a, b)^\top = (1, 0.5)^\top$ on a 20×20 grid with $\Delta x = \Delta y = 1.8$. At the boundaries $x = 0$ and $y = 0$ the scalar u is set according to a Gauss profile given by

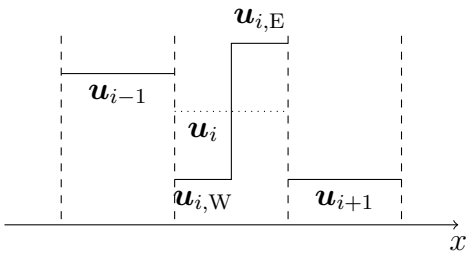
$$u(x, y) = \exp\left(-\frac{(y - \frac{b}{a} \cdot x)^2}{3^2}\right). \tag{16}$$



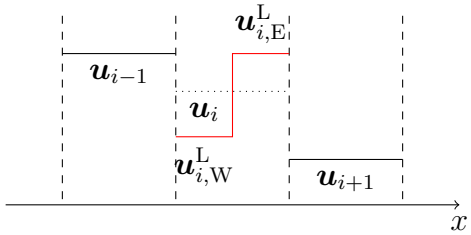
(a) Before limiting. $\boldsymbol{u}_{i,\mathrm{W}}$ lies outside the interval $[\boldsymbol{u}_{i+1}, \boldsymbol{u}_{i-1}]$ and needs to be limited.



(b) After limiting. $\boldsymbol{u}_{i,\mathrm{W}}$ is reduced, and $\boldsymbol{u}_{i,\mathrm{E}}$ is increased accordingly to conserve $\boldsymbol{u}_i$, as stated in Eq. (10).

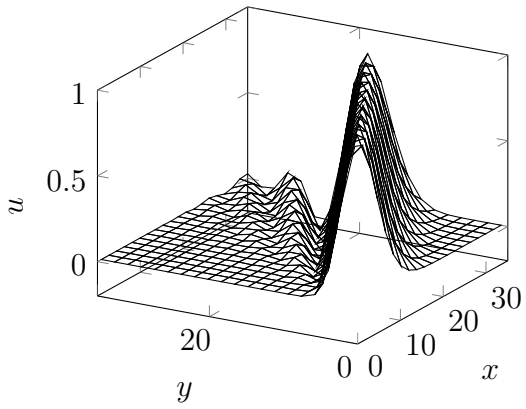


(c) Before limiting. $\boldsymbol{u}_{i,\mathrm{E}}$ lies outside the interval $[\boldsymbol{u}_{i+1}, \boldsymbol{u}_{i-1}]$ and needs to be limited.

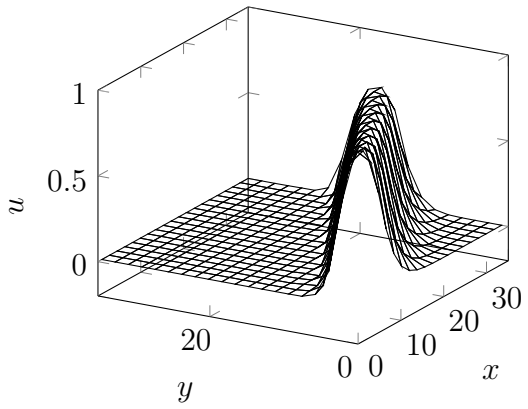


(d) After limiting. $\boldsymbol{u}_{i,\mathrm{E}}$ is reduced, and $\boldsymbol{u}_{i,\mathrm{W}}$ is increased accordingly to conserve $\boldsymbol{u}_i$, as stated in Eq. (10).

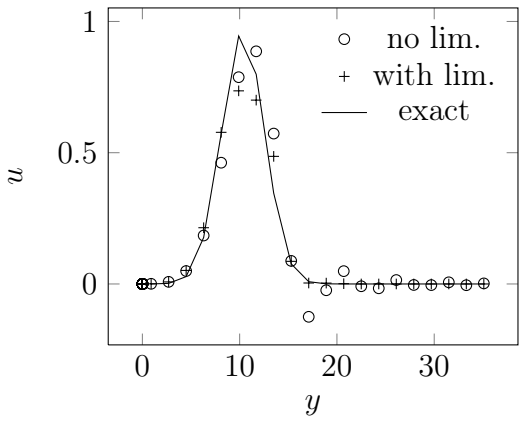
Figure 4: Illustration of the limiting procedure for the RHR solver for two different cases.



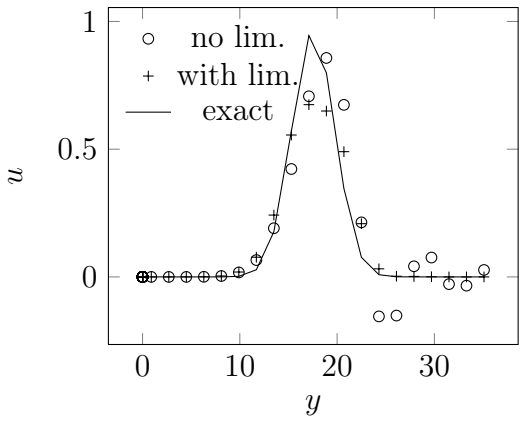
(a) Without limiting. Spurious oscillations emerge as the scalar u is advected through the domain.



(b) With limiting.



(c) Cross-section at $x = 20.7$.



(d) Cross-section at $x = 35.1$.

Figure 5: Advection of a scalar Gauss profile (16) with velocity $(a, b)^\top = (1, 0.5)^\top$, with (with lim.) and without limiting (no lim.), on a 20×20 grid with $\Delta x = \Delta y = 1.8$.

The steady state solution exhibits spurious oscillations propagating downstream on the left side of the advected crest.

After applying the limiting to the 2D steady scalar linear advection equation, the spurious oscillations are eliminated, cf. Fig. 5b. Figures 5c and 5d show two cross-sections of the numerical solutions depicted in Figures 5a and 5b, as well as the exact solution. We recognize that the limiter causes the oscillations to vanish, but also leads to more diffusive solutions, e.g. the peak values are smaller than without limiting.

In the setting of the 2D unsteady linear advection equation, i.e.

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} = 0, \tag{17}$$

the RHR solver combined with the proposed limiter complies with the minimum/maximum principle. Here, the minimum/maximum principle states that for a pure initial value problem with initial conditions $u_0(x, y)$ specified for $-\infty < x, y < \infty$ we have $\min(u_0) \leq u \leq \max(u_0)$ for all times t. We first observe that for a locally maximal state $u_{i,j}$, the limiter does not allow the corresponding half-states $u_{i,j,\mathrm{N}}$, $u_{i,j,\mathrm{S}}$, $u_{i,j,\mathrm{E}}$ and $u_{i,j,\mathrm{W}}$ to be different from $u_{i,j}$. Moreover, the limiter ensures that the adjacent half-states of the neighbouring cells do not outreach $u_{i,j}$. Since Riemann problems between the half-states at the cell interfaces reduce to simple upwinding, we see that the locally maximal state $u_{i,j}$ cannot increase as time evolves. Certainly, new local maxima can emerge but due to the previous argument, these new maxima can no longer increase after their creation. Therefore, the upper bound provided by the initial global maximum is not violated. Along similar lines, the minimum principle is met. It is emphasized that time integration must be sufficiently stable for this reasoning to hold. We would also like to point out that although the limiter (in itself) does not allow extrema to increase, source terms in a balance law might cause them to increase when the balance law is evolved in time, see Section 4.1.

3. Analysis of the RHR solver for the 2D steady scalar linear advection equation

In this section, in order to highlight some important properties of the RHR solver without limiter, we present an analysis of the solver for the 2D steady scalar linear advection equation with a linear source term, i.e.

$$a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} = cu + d, \tag{18}$$

where $\boldsymbol{v} = (a, b)^\top$ is the (constant) velocity vector and c and d are additional constants. For this equation, the flux functions are simply given by $f(u) = au$ and $g(u) = bu$. We assume without loss of generality that the two velocities are positive, i.e. $a, b > 0$. In this case, the numerical fluxes at the faces are simply chosen as the upwind fluxes, i.e.

$$f_{i+1/2,j} = au_{i,j,\mathrm{E}}, \text{ and} \tag{19}$$

$$g_{i,j+1/2} = bu_{i,j,\mathrm{N}}. \tag{20}$$

With the given flux functions and numerical fluxes, the conservativity and Rankine-Hugoniot conditions from Eqs. (10)–(13) read

$$\frac{1}{2}(u_{i,j,\mathrm{W}} + u_{i,j,\mathrm{E}}) = u_{i,j}, \tag{21}$$

$$\frac{1}{2}(u_{i,j,\mathrm{S}} + u_{i,j,\mathrm{N}}) = u_{i,j}, \tag{22}$$

$$\frac{a}{\Delta x}(u_{i,j,\mathrm{E}} - u_{i,j,\mathrm{W}}) = \frac{b}{\Delta y}(u_{i,j-1,\mathrm{N}} - u_{i,j,\mathrm{N}}) + cu_{i,j} + d, \tag{23}$$

$$\frac{b}{\Delta y}(u_{i,j,\mathrm{N}} - u_{i,j,\mathrm{S}}) = \frac{a}{\Delta x}(u_{i-1,j,\mathrm{E}} - u_{i,j,\mathrm{E}}) + cu_{i,j} + d, \tag{24}$$

and the finite volume scheme (7) in the steady case reads

$$\frac{a}{\Delta x}(u_{i-1,j,\mathrm{E}} - u_{i,j,\mathrm{E}}) + \frac{b}{\Delta y}(u_{i,j-1,\mathrm{N}} - u_{i,j,\mathrm{N}}) + cu_{i,j} + d = 0. \tag{25}$$

We now wish to show how the stencil for the solution in cell (i, j), $u_{i,j}$, depends on the solution in the neighbouring cells. To this end, we solve Eq. (21) for $u_{i,j,\mathrm{W}}$ and Eq. (22) for $u_{i,j,\mathrm{S}}$ and substitute the results into Eqs. (23) and (24), respectively, which yields

$$\frac{2a}{\Delta x}(u_{i,j,\mathrm{E}} - u_{i,j}) = \frac{b}{\Delta y}(u_{i,j-1,\mathrm{N}} - u_{i,j,\mathrm{N}}) + cu_{i,j} + d, \tag{26}$$

$$\frac{2b}{\Delta y}(u_{i,j,\mathrm{N}} - u_{i,j}) = \frac{a}{\Delta x}(u_{i-1,j,\mathrm{E}} - u_{i,j,\mathrm{E}}) + cu_{i,j} + d. \tag{27}$$

Using Eq. (25), we replace the right-hand side of Eqs. (26)–(27), which leads to

$$\frac{a}{\Delta x}(u_{i,j,\mathrm{E}} + u_{i-1,j,\mathrm{E}} - 2u_{i,j}) = 0, \tag{28}$$

$$\frac{b}{\Delta y}(u_{i,j,\mathrm{N}} + u_{i,j-1,\mathrm{N}} - 2u_{i,j}) = 0. \tag{29}$$

Finally, we solve Eq. (28) for $u_{i-1,j,\mathrm{E}}$ and Eq. (29) for $u_{i,j-1,\mathrm{N}}$ and substitute the results into Eq. (25), which yields

$$\frac{a}{\Delta x}(u_{i,j} - u_{i,j,\mathrm{E}}) + \frac{b}{\Delta y}(u_{i,j} - u_{i,j,\mathrm{N}}) + \frac{c}{2}u_{i,j} + \frac{d}{2} = 0. \tag{30}$$

We now add the following equations to retrieve a stencil for $u_{i,j}$: Eqs. (28), (29) and (30), Eq. (28) with shifted indices $(i, j) \to (i, j-1)$, Eq. (29) with shifted indices $(i, j) \to (i-1, j)$, Eq. (30) with shifted indices $(i, j) \to (i-1, j)$, Eq. (30) with shifted indices $(i, j) \to (i, j-1)$, Eq. (30) with shifted indices $(i, j) \to (i-1, j-1)$. After solving the resulting expression for $u_{i,j}$, we get

$$\begin{aligned} u_{i,j} = {} & \frac{c\Delta x\Delta y + 2a\Delta y - 2b\Delta x}{2a\Delta y + 2b\Delta x - c\Delta x\Delta y}u_{i-1,j} + \frac{c\Delta x\Delta y - 2a\Delta y + 2b\Delta x}{2a\Delta y + 2b\Delta x - c\Delta x\Delta y}u_{i,j-1} \\ & + \frac{c\Delta x\Delta y + 2a\Delta y + 2b\Delta x}{2a\Delta y + 2b\Delta x - c\Delta x\Delta y}u_{i-1,j-1} + \frac{4d\Delta x\Delta y}{2a\Delta y + 2b\Delta x - c\Delta x\Delta y}. \end{aligned} \tag{31}$$

The stencil may be viewed as an operator which maps the solution to the south, west and south-west to the location (i, j); note for example that for $\frac{\Delta x}{\Delta y} = \frac{a}{b}$ and $c = d = 0$, the stencil reduces to

$$u_{i,j} = u_{i-1,j-1}, \tag{32}$$

i.e. the RHR solver propagates the solution *exactly* diagonally to the grid. Figure 6 shows the numerical results for a case with advection of a Gauss profile given by Eq. (16) on a 10×10 grid with $\Delta x = 1.8$ and $\Delta y = 3.6$ and $2a = b$. As expected the numerical solution is exact. Although this example of advection with constant velocity is rather trivial, the result illustrates the capability of the RHR solver to capture fluxes in oblique direction with respect to the grid orientation.

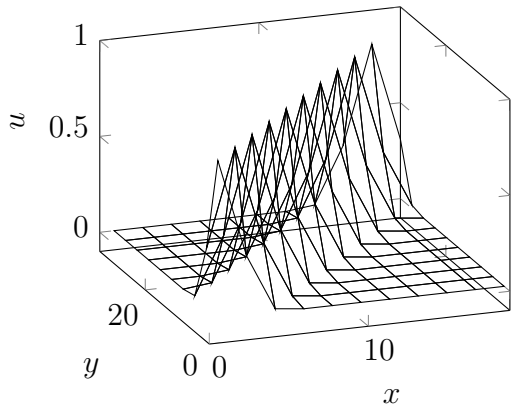


Figure 6: Advection of a scalar Gauss profile (16) with $2a = b$ and $\Delta x = 1.8$ and $\Delta y = 3.6$ on a 10×10 grid, which is solved exactly by the RHR solver (without limiter).

3.1. Error analysis

In this section, we wish to analyse the spatial order of accuracy of the RHR solver (without limiter) for the 2D steady advection equation with source term (18). We define the local error by the difference between the numerical solution $u_{i,j}$ and the exact solution $\tilde{u}_{i,j}$, where the numerical solution is evaluated with the exact solution values $\tilde{u}_{i-1,j}$, $\tilde{u}_{i,j-1}$ and $\tilde{u}_{i-1,j-1}$, i.e.

$$\begin{aligned} E_{\text{local}} = &\frac{c\Delta x\Delta y - 2a\Delta y - 2b\Delta x}{2a\Delta y + 2b\Delta x - c\Delta x\Delta y}\tilde{u}_{i,j} + \frac{c\Delta x\Delta y + 2a\Delta y - 2b\Delta x}{2a\Delta y + 2b\Delta x - c\Delta x\Delta y}\tilde{u}_{i-1,j} \\ &+ \frac{c\Delta x\Delta y - 2a\Delta y + 2b\Delta x}{2a\Delta y + 2b\Delta x - c\Delta x\Delta y}\tilde{u}_{i,j-1} + \frac{c\Delta x\Delta y + 2a\Delta y + 2b\Delta x}{2a\Delta y + 2b\Delta x - c\Delta x\Delta y}\tilde{u}_{i-1,j-1} \\ &+ \frac{4d\Delta x\Delta y}{2a\Delta y + 2b\Delta x - c\Delta x\Delta y}, \end{aligned} \tag{33}$$

where we have used the stencil (31) to express $u_{i,j}$ as a function of the exact solution in the neighbouring cells. We now assume that the solution $\tilde{u}$ is sufficiently smooth such that

$\tilde{u}_{i-1,j}$, $\tilde{u}_{i,j-1}$ and $\tilde{u}_{i-1,j-1}$ in Eq. (33) can be expressed as a Taylor series around (x_i, y_j). Since $\tilde{u}$ is an exact solution of Eq. (18), we find that the y derivative is given by

$$\frac{\partial \tilde{u}}{\partial y} = -\frac{a}{b}\frac{\partial \tilde{u}}{\partial x} + \frac{c}{b}\tilde{u} + \frac{d}{b}. \tag{34}$$

We utilize this to replace all y derivatives stemming from the Taylor series expansion in Eq. (33). This causes the zeroth, first and second order terms to cancel, leaving

$$\begin{aligned} E_{\text{local}} = \frac{\Delta x \Delta y}{3b^2\left(2a\Delta y + 2b\Delta x - c\Delta x\Delta y\right)}\Big(&- a^3\Delta y^2 \frac{\partial^3 \tilde{u}}{\partial x^3}\Big|_{i,j} + ab^2\Delta x^2 \frac{\partial^3 \tilde{u}}{\partial x^3}\Big|_{i,j} + 3a^2c\Delta y^2 \frac{\partial^2 \tilde{u}}{\partial x^2}\Big|_{i,j} \\ &- 3ac^2\Delta y^2 \frac{\partial \tilde{u}}{\partial x}\Big|_{i,j} + c^3\Delta y^2 \tilde{u}_{i,j} + c^2 d\Delta y^2\Big) + \text{higher order terms.} \end{aligned} \tag{35}$$

If we assume that the ratio $\Delta x/\Delta y$ is fixed, we find

$$E_{\text{local}} = \mathcal{O}(\Delta x^3), \tag{36}$$

i.e. the local spatial error of the RHR solver is of third order. To find the global error, we realize that in order to advect the solution from the boundary to a certain cell, the stencil (31) is applied a certain number of times proportional to $1/\Delta x$. Therefore the global error is of second order,

$$E_{\text{global}} = \mathcal{O}(\Delta x^2). \tag{37}$$

We would like to point out the fact that the scheme achieves second order with a compact stencil that is only dependent on the solution value and the fluxes in the nearest neighbouring cells. This is in contrast to e.g. a MUSCL scheme, which requires two cells in all directions to achieve second order.

In this paper we mainly focus on showing the spatial properties of the RHR solver for the steady case, thus we do not investigate the temporal properties in detail. The time integration procedure is outlined in the following section.

4. Numerical investigation

In this section, we numerically investigate how the RHR solver behaves for steady states for a two-dimensional advection equation, the two-dimensional isothermal Euler equations, the two-dimensional shallow water equations and the two-dimensional full Euler equations. We start by describing in a general way how the time stepping is performed, which we need to arrive at the steady states.

4.1. Time integration/solution algorithm

In general, we wish to solve a multidimensional system of conservation laws with source terms. For simplicity, we consider the two-dimensional case, i.e.

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} + \frac{\partial \boldsymbol{g}(\boldsymbol{u})}{\partial y} = \boldsymbol{q}(\boldsymbol{u}, \boldsymbol{x}). \tag{38}$$

For this system of balance laws, the finite volume scheme (7) can (in the unsteady case) be rearranged as

$$\frac{\partial \boldsymbol{u}_{i,j}}{\partial t} = \frac{1}{\Delta x}(\boldsymbol{f}_{i-1/2,j} - \boldsymbol{f}_{i+1/2,j}) + \frac{1}{\Delta y}(\boldsymbol{g}_{i,j-1/2} - \boldsymbol{g}_{i,j+1/2}) + \boldsymbol{q}_{i,j}. \tag{39}$$

From the RHR relations in Eqs. (10)–(13), we see that the system of ordinary differential equations (ODEs) for $\boldsymbol{u}_{i,j,\mathrm{E}}$ and $\boldsymbol{u}_{i,j,\mathrm{W}}$ is highly coupled between cells, since the interface fluxes $\boldsymbol{f}_{i\pm1/2,j}$ and $\boldsymbol{g}_{i,j\pm1/2}$ (in general) depend on the states in neighbouring cells on both sides. This presents a challenge when implementing a time integration scheme. Hence we choose to calculate the cross-flow fluxes in the total source terms $\boldsymbol{q}_{x,i,j}$ and $\boldsymbol{q}_{y,i,j}$ based on the *previous* time step when solving the RHR relations for the next time step.

We then propose to move the solution forward in time using the following algorithm.

1. Calculate the total source terms based on the fluxes from the previous time step:

$$\boldsymbol{q}^n_{x,i,j} = \frac{\boldsymbol{g}^{n-1/2}_{i,j-1/2} - \boldsymbol{g}^{n-1/2}_{i,j+1/2}}{\Delta y} + \boldsymbol{q}^n_{i,j}, \qquad \boldsymbol{q}^n_{y,i,j} = \frac{\boldsymbol{f}^{n-1/2}_{i-1/2,j} - \boldsymbol{f}^{n-1/2}_{i+1/2,j}}{\Delta x} + \boldsymbol{q}^n_{i,j} \tag{40}$$

   For the first time step, the fluxes at time step $n-1/2$ are unknown, but are assumed to be zero.
2. Compute the half-states $\boldsymbol{u}^n_{i,j,\mathrm{S}}$, $\boldsymbol{u}^n_{i,j,\mathrm{N}}$, $\boldsymbol{u}^n_{i,j,\mathrm{W}}$ and $\boldsymbol{u}^n_{i,j,\mathrm{E}}$ using Eqs. (10)–(13) based on $\boldsymbol{u}^n_{i,j}$ and the total source terms $\boldsymbol{q}^n_{x,i,j}$ and $\boldsymbol{q}^n_{y,i,j}$ given by (40).
3. Calculate the limited states $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{N}})^n$ and $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{E}})^n$ according to Eq. (14). The limited states $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{S}})^n$ and $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{W}})^n$ are then given by Eqs. (10) and (11), respectively.
4. Solve the Riemann problems defined by the limited values $(\boldsymbol{u}^{\mathrm{L}}_{i-1,j,\mathrm{E}})^n$ and $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{W}})^n$, $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{E}})^n$ and $(\boldsymbol{u}^{\mathrm{L}}_{i+1,j,\mathrm{W}})^n$, $(\boldsymbol{u}^{\mathrm{L}}_{i,j-1,\mathrm{N}})^n$ and $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{S}})^n$, $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{N}})^n$ and $(\boldsymbol{u}^{\mathrm{L}}_{i,j+1,\mathrm{S}})^n$, to obtain the Riemann fluxes $\boldsymbol{f}^n_{i-1/2,j}$, $\boldsymbol{f}^n_{i+1/2,j}$, $\boldsymbol{g}^n_{i,j-1/2}$ and $\boldsymbol{g}^n_{i,j+1/2}$, respectively.
5. Calculate an intermediate state $\boldsymbol{u}^{n+1/2}$ given by

$$\boldsymbol{u}^{n+1/2}_{i,j} = \boldsymbol{u}^n_{i,j} + \frac{\Delta t}{\Delta x}(\boldsymbol{f}^n_{i-1/2,j} - \boldsymbol{f}^n_{i+1/2,j}) + \frac{\Delta t}{\Delta y}(\boldsymbol{g}^n_{i,j-1/2} - \boldsymbol{g}^n_{i,j+1/2}) + \Delta t \boldsymbol{q}^n_{i,j}, \tag{41}$$

6. Compute the half-states $\boldsymbol{u}^{n+1/2}_{i,j,\mathrm{S}}$, $\boldsymbol{u}^{n+1/2}_{i,j,\mathrm{N}}$, $\boldsymbol{u}^{n+1/2}_{i,j,\mathrm{W}}$ and $\boldsymbol{u}^{n+1/2}_{i,j,\mathrm{E}}$ using Eqs. (10)–(13) based on $\boldsymbol{u}^{n+1/2}_{i,j}$ and the total source terms $\boldsymbol{q}^n_{x,i,j}$ and $\boldsymbol{q}^n_{y,i,j}$ given by (40).
7. Calculate the limited states $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{N}})^{n+1/2}$ and $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{E}})^{n+1/2}$ according to Eq. (14). The limited states $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{S}})^{n+1/2}$ and $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{W}})^{n+1/2}$ are then given by Eqs. (10) and (11), respectively.
8. Solve the Riemann problems defined by the limited values $(\boldsymbol{u}^{\mathrm{L}}_{i-1,j,\mathrm{E}})^{n+1/2}$ and $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{W}})^{n+1/2}$, $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{E}})^{n+1/2}$ and $(\boldsymbol{u}^{\mathrm{L}}_{i+1,j,\mathrm{W}})^{n+1/2}$, $(\boldsymbol{u}^{\mathrm{L}}_{i,j-1,\mathrm{N}})^{n+1/2}$ and $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{S}})^{n+1/2}$, $(\boldsymbol{u}^{\mathrm{L}}_{i,j,\mathrm{N}})^{n+1/2}$ and $(\boldsymbol{u}^{\mathrm{L}}_{i,j+1,\mathrm{S}})^{n+1/2}$, to obtain the Riemann fluxes $\boldsymbol{f}^{n+1/2}_{i-1/2,j}$, $\boldsymbol{f}^{n+1/2}_{i+1/2,j}$, $\boldsymbol{g}^{n+1/2}_{i,j-1/2}$ and $\boldsymbol{g}^{n+1/2}_{i,j+1/2}$, respectively.

9. Advance time by Δt to reach $\boldsymbol{u}_{i,j}^{n+1}$, i.e.

$$\boldsymbol{u}_{i,j}^{n+1} = \boldsymbol{u}_{i,j}^{n} + \frac{\Delta t}{2}\left(\frac{\boldsymbol{f}_{i-1/2,j}^{n+1/2} - \boldsymbol{f}_{i+1/2,j}^{n+1/2}}{\Delta x} + \frac{\boldsymbol{g}_{i,j-1/2}^{n+1/2} - \boldsymbol{g}_{i,j+1/2}^{n+1/2}}{\Delta y} + \boldsymbol{q}_{i,j}^{n+1/2} + \frac{\boldsymbol{f}_{i-1/2,j}^{n} - \boldsymbol{f}_{i+1/2,j}^{n}}{\Delta x} + \frac{\boldsymbol{g}_{i,j-1/2}^{n} - \boldsymbol{g}_{i,j+1/2}^{n}}{\Delta y} + \boldsymbol{q}_{i,j}^{n} \right), \tag{42}$$

This scheme is quite similar to Heun's method, a two-stage Runge-Kutta method. In our scheme, however, the source terms $\boldsymbol{q}_x^n$ and $\boldsymbol{q}_y^n$ are used in *both* stages, and are calculated based on the fluxes in the previous half time step, given by Eq. (40). An alternative to this scheme would have been a simple first-order forward Euler scheme, i.e. steps 1 to 5 above with half steps $n - 1/2$ in (40) and $n + 1/2$ in (41) replaced by the old and new time levels $n - 1$ and $n + 1$, respectively. However, the scheme presented above exhibits better stability properties and can handle larger time steps. Strong stability preserving (SSP) schemes [25] might be an option to consider in the future, but this will probably also require careful treatment of the updating of the source terms $\boldsymbol{q}_x^n$ and $\boldsymbol{q}_y^n$.

Whenever a MUSCL scheme was used for comparison, the time integration was performed with a standard Heun's method. A CFL number of $C = 0.3$ was used for all the numerical computations, which was chosen as a safe value to avoid any possible instabilities in time, and since our focus was not on the time integration itself. The CFL number is defined as

$$C = \Delta t \max_{p,k} \frac{|\lambda_{p,k}|}{\Delta x_k}, \tag{43}$$

where $\lambda_{p,k}$ is the pth eigenvalue in the kth dimension of the hyperbolic system, and Δx_k is the grid spacing in the kth dimension.

The time stepping scheme presented above is formally not of second order for the RHR scheme, since the total source terms $\boldsymbol{q}_{x,i,j}^n$ and $\boldsymbol{q}_{y,i,j}^n$ depend on the fluxes at the previous time step, but it is certainly more accurate than a first order method in time. However, the present goal of the time integration is to reach some steady state, so the exact order of the scheme is of less importance. For each time step, the RHR solver involves solving the RHR relations (10)–(13) which are not solved in the MUSCL scheme, hence we expect that each time step will be more costly for the RHR scheme. We will show results for the computational time of a single iteration for each system of equations in the following sections. It should be pointed out that we have not put any effort in accelerating the convergence of the RHR solver to steady state.

4.2. Method of manufactured solutions

To compute the exact error of a numerical solution, one needs to know the exact solution to the problem, given the boundary (and possibly initial) conditions. For more complex systems of PDEs, domains and boundary conditions, an exact solution may be out of reach. In these cases, the method of manufactured solutions can often be useful [26]. Instead of searching for the exact solution to the original problem (38), one rather makes the ansatz

that the solution is $\boldsymbol{u}^*$, which can be an arbitrary sufficiently smooth function, preferably close to an exact solution. We then assume that the ansatz solves the modified equation

$$\frac{\partial \boldsymbol{f}(\boldsymbol{u}^*)}{\partial x} + \frac{\partial \boldsymbol{g}(\boldsymbol{u}^*)}{\partial y} = \boldsymbol{q}(\boldsymbol{u}^*, \boldsymbol{x}) + \boldsymbol{R}(\boldsymbol{u}^*, \boldsymbol{x}), \tag{44}$$

where $\boldsymbol{R}$ is the residual, caused by the fact that $\boldsymbol{u}^*$ is not an exact solution to the original problem. This residual is simply calculated by inserting $\boldsymbol{u}^*$ into Eq. (44) and solving for $\boldsymbol{R}$. If $\boldsymbol{R}$ were zero, $\boldsymbol{u}^*$ would be an exact solution to Eq. (2) or the steady version of Eq. (38). This problem has essentially the same structure as the steady version of the original problem (38), and can thus be used to investigate the accuracy properties of the numerical method. We solve Eq. (44) numerically using the modified source term $\boldsymbol{q}^* = \boldsymbol{q} + \boldsymbol{R}$, and since we now know that the manufactured solution $\boldsymbol{u}^*$ is the exact solution to the modified problem, we can compute the numerical error exactly. The method of manufactured solutions will be used to calculate the numerical error for an isothermal Euler case in Section 4.4 and a shallow water case in Section 4.5.

In the following sections we will present a number of numerical cases, each of which has either a known exact solution or a manufactured solution, except for the channel flow case in Section 4.6. Knowing the solution, we can set the boundary conditions to the exact solution, avoiding any possible issues of employing characteristic or non-reflecting boundary conditions. The characteristic Riemann solvers will automatically take the characteristic variables from the exterior, i.e. the given boundary conditions, or from the interior, i.e. from the solution in the adjacent cell at the previous time level or previous stage, depending on whether the characteristic is entering or leaving the domain.

4.3. Advection of a scalar

In this section we present some findings with the RHR solver applied to a two-dimensional scalar linear advection case with constant velocity $(a, b)^\top$, which in the steady case is given by

$$a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} = 0. \tag{45}$$

The simplicity of this equation makes it suitable to illustrate some important properties of the RHR solver.

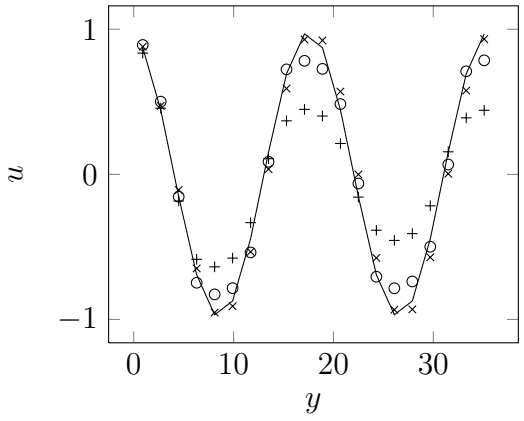
4.3.1. Numerical order of convergence

As shown in Section 3.1, the RHR solver is expected to be of second order for a smooth solution. To confirm this numerically, we consider a cosine shaped solution,

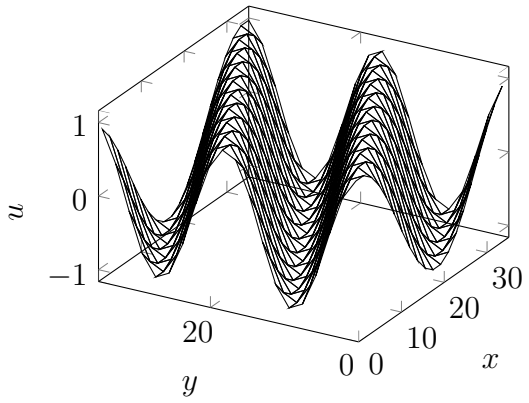
$$u(x, y) = \cos(\omega_0(-bx + ay)) \tag{46}$$

where $\omega_0 = \pi/9$, $a = 1.0$, $b = 0.5$, and the grid has dimensions $[0, 36] \times [0, 36]$. The solution (46) is used to set the boundary conditions at $x = 0$ and $y = 0$. We solve Eq. (17) with initial condition given by Eq. (46) using the time integration scheme in Section 4.1 and wait for the solution to reach steady state. The numerical solution is then compared to

the exact solution to determine the error. For illustration, Figure 7 shows the solution for the RHR solver with and without limiter and a MUSCL upwind solver with van Albada limiter for a grid of 20×20 cells. We recognize that the RHR solver with limiter does a significantly better job than MUSCL in resolving the problem on this grid, while the RHR solver without limiter is even more accurate. The plot in Figure 8 shows the L^2 errors for



(a) Cross-section at $x = 35.1$, with 20×20 cells. RHR with limiter ($\circ$), RHR without limiter ($\times$), MUSCL with van Albada limiter ($+$) and exact solution (solid line).



(b) Exact solution.

Figure 7: Advection of a scalar cosine-shaped boundary condition (46) with velocity $(a, b)^\top = (1, 0.5)^\top$, on a 20×20 grid with $\Delta x = \Delta y = 1.8$.

the RHR solver with and without limiter, and the MUSCL upwind solver with minmod and van Albada limiters, as functions of grid size $n_x = n_y$, where n_x and n_y are the number of grid cells in x- and y-direction, respectively. As seen in the figure, the RHR solver has an error significantly smaller than that of a MUSCL solver with van Albada limiter, while the RHR solver without limiter is even more accurate.

Figure 9 shows the L^1 norm of the residual as a function of the number of time steps, given by

$$R^n = \frac{1}{n_x n_y n_v} \sum_{i,j,k} |(\boldsymbol{u}^n_{i,j})_k - (\boldsymbol{u}^{n-1}_{i,j})_k|, \tag{47}$$

where $(\boldsymbol{u}^n_{i,j})_k$ is the kth component of $\boldsymbol{u}^n$ at the grid point i, j, and n_v is the number of components of $\boldsymbol{u}^n$. Note that in our general definition (47), $n_v = 1$ for the scalar advection equation.

We see that all methods converge to machine precision. The RHR solver with limiter uses 30-50% more time steps to reach steady state than the MUSCL upwind solver with van Albada and minmod limiters, respectively. As the upwind method with the less dissipative van Albada limiter needs more time levels to converge to steady state than the more dissipative minmod limiter, the reason for the larger number of time levels needed by the

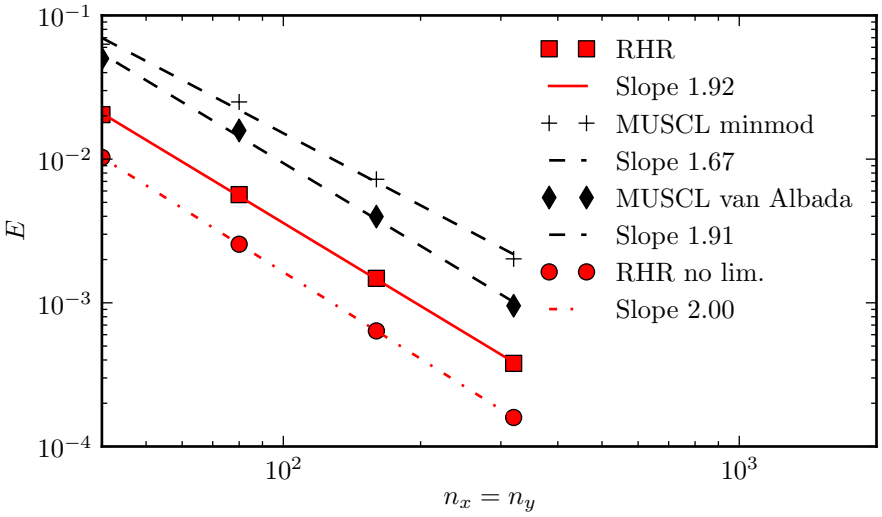


Figure 8: L^2 error E as a function of grid size $n_x = n_y$ for a case with advection of a cosine shaped scalar profile.

RHR solver seems to be that the RHR solver is less dissipative than the MUSCL upwind methods.

4.3.2. RHR compared to upwind and MUSCL

In this section we present results for scalar linear advection of a Gaussian curve, given by Eq. (16), in order to illustrate the good properties of the RHR solver when it comes to transport in directions oblique to the grid lines. Figure 10 shows results calculated with a first-order upwind method, a MUSCL upwind scheme with van Albada limiter, and the RHR solver with limiter. The RHR solver is seen to be less diffusive than the other two methods, which is also illustrated by the scalar cross-section profiles shown in Fig. 10d.

The computational cost for one time step of each scheme is shown in Table 1 for a PC with a clock rate of 2.4 GHz. As expected, the RHR scheme has a similar cost as the MUSCL scheme, since solving the RHR relations (10)–(13) only involves solving two simple linear equations.

Method	Cost (ms)
RHR no lim.	8.8
RHR lim.	8.8
MUSCL minmod	8.6
MUSCL van Albada	8.7

Table 1: Computational cost per time step for advection of a scalar on a 80×80 grid

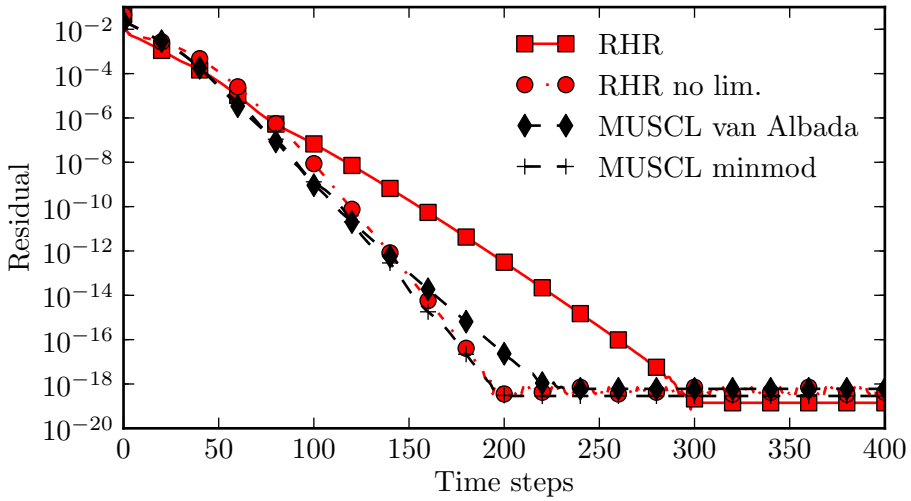


Figure 9: L^1 norm of the residual as a function of the number of time steps for a case with advection of a cosine shaped scalar profile, for a 10×10 grid, $C = 0.3$.

4.4. Isothermal Euler equations

As a slightly more complex numerical example, we now present results for the two-dimensional isothermal Euler equations,

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho u\\ \rho v\end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix}\rho u\\ \rho u^2 + p\\ \rho uv\end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2 + p\end{bmatrix} = \mathbf{0}, \tag{48}$$

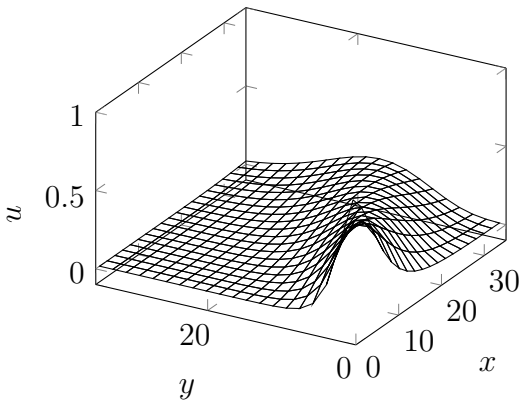
where ρ is the density, and u and v are velocity components in the x- and y-directions, respectively. We close the system with a simple equation of state, $p = \rho c^2$, where c is the constant speed of sound. In the following we first show how the RHR relations are solved for the isothermal Euler equations, followed by a brief description of the characteristic solver used to solve the resulting Riemann problems. Finally, we present numerical results for a manufactured steady solution demonstrating second order.

4.4.1. Solving the RHR relations

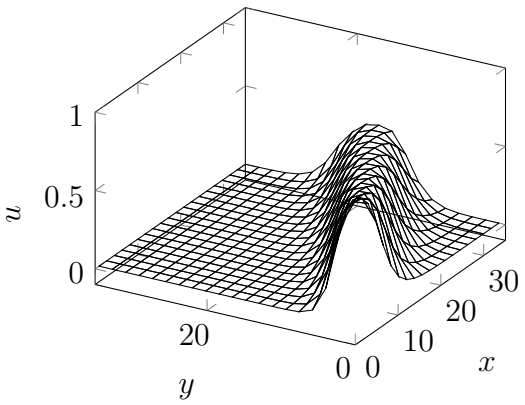
To calculate the half-states $\boldsymbol{u}_{i,j,\mathrm{E}}$, $\boldsymbol{u}_{i,j,\mathrm{W}}$, $\boldsymbol{u}_{i,j,\mathrm{S}}$ and $\boldsymbol{u}_{i,j,\mathrm{N}}$, we need to solve the RHR relations given by Eqs. (10)–(13). This is a straightforward process for the advection equation we have considered so far, but slightly more complex for the isothermal Euler equations.

In the following, we only consider the solution procedure for $\boldsymbol{u}_{i,j,\mathrm{E}}$, as the procedure for $\boldsymbol{u}_{i,j,\mathrm{N}}$ is completely analogous. The states $\boldsymbol{u}_{i,j,\mathrm{W}}$ and $\boldsymbol{u}_{i,j,\mathrm{S}}$ are then given by Eqs. (10)–(11). Using Eq. (10), we replace $\boldsymbol{u}_{i,j,\mathrm{W}}$ in Eq. (12), which then reads

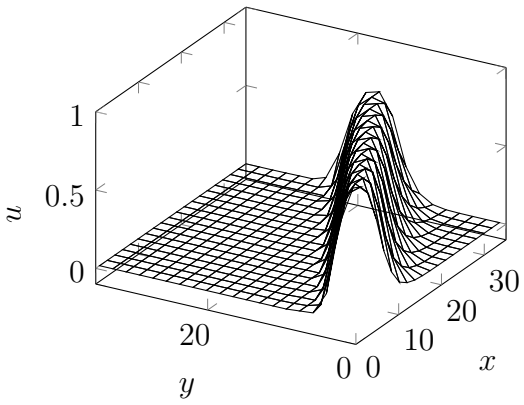
$$\frac{1}{\Delta x}(\boldsymbol{f}(\boldsymbol{u}_{\mathrm{E}}) - \boldsymbol{f}(2\boldsymbol{u} - \boldsymbol{u}_{\mathrm{E}})) = \boldsymbol{q}_x, \tag{49}$$



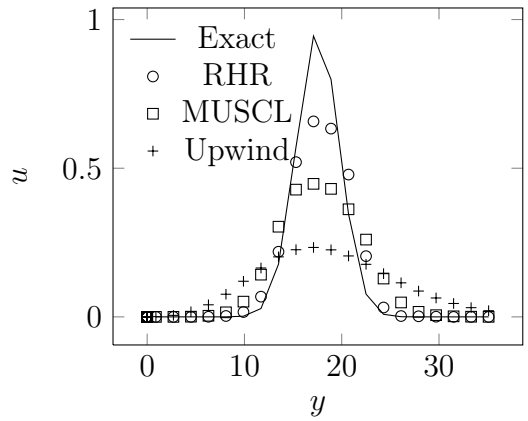
(a) Upwind.



(b) MUSCL with van Albada limiter.



(c) RHR solver with limiter.



(d) Cross-section at $x = 35.1$ for the three numerical solutions, together with the exact solution.

Figure 10: Advection of a scalar Gaussian profile (16) with velocity $(a, b)^\top = (1, 0.5)^\top$, on a 20×20 grid with $\Delta x = \Delta y = 1.8$.

where we have omitted the spatial indices i and j. The flux function $\boldsymbol{f}$ is given by

$$\boldsymbol{f}(\boldsymbol{u}) = \begin{bmatrix} \rho u \\ \rho(u^2 + c^2) \\ \rho u v \end{bmatrix} = \begin{bmatrix} u_2 \\ u_2^2/u_1 + u_1 c^2 \\ u_2 u_3/u_1 \end{bmatrix}, \tag{50}$$

where $u_1 = \rho$, $u_2 = \rho u$ and $u_3 = \rho v$. Substitution into Eq. (49) leads to

$$\begin{bmatrix} u_{2,\mathrm{E}} \\ u_{2,\mathrm{E}}^2/u_{1,\mathrm{E}} + u_{1,\mathrm{E}} c^2 \\ u_{2,\mathrm{E}} u_{3,\mathrm{E}}/u_{1,\mathrm{E}} \end{bmatrix} - \begin{bmatrix} 2u_2 - u_{2,\mathrm{E}} \\ (2u_2 - u_{2,\mathrm{E}})^2/(2u_1 - u_{1,\mathrm{E}}) + (2u_1 - u_{1,\mathrm{E}})c^2 \\ (2u_2 - u_{2,\mathrm{E}})(2u_3 - u_{3,\mathrm{E}})/(2u_1 - u_{1,\mathrm{E}}) \end{bmatrix} = \Delta x \boldsymbol{q}_x. \tag{51}$$

The first component of Eq. (51) is easily solved for $u_{2,\mathrm{E}}$, i.e.

$$u_{2,\mathrm{E}} = \frac{\Delta x}{2} q_{x,1} + u_2. \tag{52}$$

Next, we solve the second component of Eq. (51) for $u_{1,\mathrm{E}}$. After substituting $\xi = u_1 - u_{1,\mathrm{E}}$ one obtains the cubic equation

$$\xi^3 + \frac{\Delta x q_{x,2}}{2c^2}\xi^2 + \frac{2u_{2,\mathrm{E}}^2 - 2u_1^2 c^2 + 4u_2(u_2 - u_{2,\mathrm{E}})}{2c^2}\xi + \frac{-4u_2 u_1(u_2 - u_{2,\mathrm{E}}) - \Delta x q_{x,2} u_1^2}{2c^2} = 0 \tag{53}$$

for ξ. This equation may be solved either numerically using Newton's method with good starting values, or analytically using an exact solution formula, of which we have used the latter. In our analytical solution of equation (53), we choose the root yielding a positive density $u_{1,\mathrm{E}}$ and minimising the jump ξ between $u_{1,\mathrm{E}}$ and u_1.

Having found ξ and thus $u_{1,\mathrm{E}}$, one can calculate $u_{3,\mathrm{E}}$ using the third component of Eq. (51) and obtains

$$u_{3,\mathrm{E}} = \frac{\frac{(2u_2 - u_{2,\mathrm{E}})2u_3}{2u_1 - u_{1,\mathrm{E}}} + \Delta x q_{x,3}}{\frac{u_{2,\mathrm{E}}}{u_{1,\mathrm{E}}} + \frac{(2u_2 - u_{2,\mathrm{E}})}{2u_1 - u_{1,\mathrm{E}}}}. \tag{54}$$

In summary, Eqs. (52)–(54) yield $\boldsymbol{u}_{i,j,\mathrm{E}}$, and similarly $\boldsymbol{u}_{i,j,\mathrm{N}}$ can be calculated.

4.4.2. Characteristic solver

Here a characteristic-based Riemann solver, similar to the one by Sesterhenn et al. [27, 28], is employed. The full derivation can be found in Appendix A. The central state is given by

$$\rho_{\mathrm{C}} = \frac{c(\rho_{\mathrm{R}} + \rho_{\mathrm{L}})}{2c + u_{\mathrm{R}} - u_{\mathrm{L}}}, \tag{55}$$

$$u_{\mathrm{C}} = \frac{c(\rho_{\mathrm{L}} - \rho_{\mathrm{R}}) + \rho_{\mathrm{C}} u_{\mathrm{R}} + \rho_{\mathrm{C}} u_{\mathrm{L}}}{2\rho_{\mathrm{C}}}. \tag{56}$$

In two dimensions, the velocity component parallel to the face is simply advected from the upwind side, i.e. $v_{\mathrm{C}} = v_{\mathrm{L}}$ if $u_{\mathrm{C}} > 0$, and $v_{\mathrm{C}} = v_{\mathrm{R}}$ if $u_{\mathrm{C}} < 0$. The Riemann flux is then given by $\boldsymbol{f}(\boldsymbol{u}_{\mathrm{C}})$.

This solver works well for small Mach numbers, but can be replaced by an exact Riemann solver or e.g. Roe's approximate Riemann solver for higher Mach numbers.

4.4.3. Order of convergence

To check the order of convergence of the RHR solver for the isothermal Euler equations, we apply the solver to a problem with a manufactured solution. For the solution we make the ansatz

$$\rho = \rho_0 \exp[\frac{-\frac{1}{2}(x^2+y^2)B^2}{c^2}], \tag{57}$$

$$u = Bx, \tag{58}$$

$$v = -By, \tag{59}$$

where B and ρ_0 are some constants; here we chose $B = 0.1$ and $\rho_0 = 1.0$. For the speed of sound, we chose $c = 1.0$. We then insert this into the (steady) isothermal Euler equations to find the source terms that result from the presented ansatz.

$$\frac{\partial}{\partial x}\begin{bmatrix}\rho u\\ \rho(u^2+c^2)\\ \rho uv\end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix}\rho v\\ \rho uv\\ \rho(v^2+c^2)\end{bmatrix} = \begin{bmatrix}\rho\\ \rho u\\ \rho v\end{bmatrix}\frac{B}{c^2}(v^2-u^2) \equiv \boldsymbol{q}. \tag{60}$$

We have now derived a manufactured solution with a corresponding source term, which we use to analyse the order of convergence. The potential flow field specified by Eqs. (58)–(59) is illustrated in Figure 11. When we solve this case numerically, the solution in Eqs. (57)–(59) is used to set the boundary conditions exactly on all boundaries, while the source term $\boldsymbol{q}$ is computed from Eq. (60) in all cells. The numerical solution is then compared with the exact solution to calculate the error.

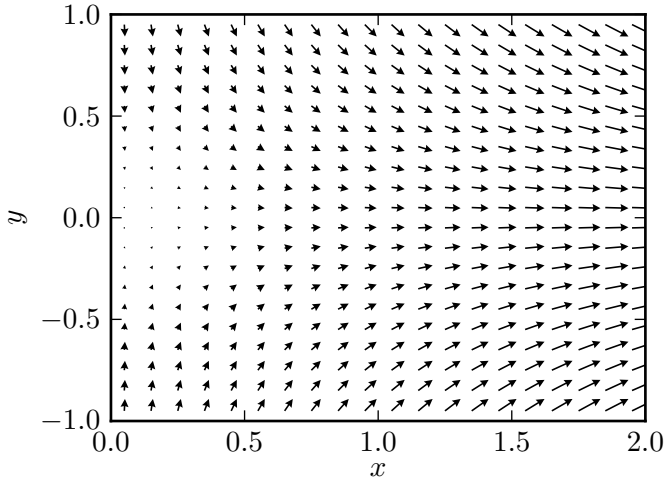


Figure 11: Velocity field for the 2D isothermal Euler case.

Figure 12 shows the L^2 error for density, $\|\rho - \rho_{\text{exact}}\|_2$, as a function of grid size n for a $n \times n$ grid, which demonstrates second order convergence for both the MUSCL upwind

scheme with minmod and monotonized central (MC) limiters, and the RHR solver with limiter. The error of the RHR solver is clearly smaller than the error of the MUSCL MC scheme, and almost one order of magnitude smaller than that of the MUSCL minmod scheme.

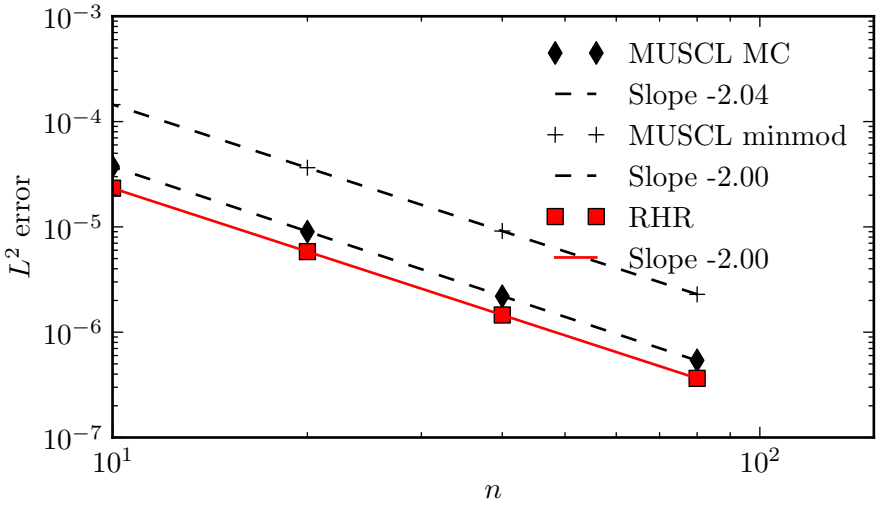


Figure 12: Grid convergence of the L^2 error of density for the 2D isothermal Euler equations.

Figure 13 shows the L^1 norm of the residual as a function of the number of time steps for the same case, which shows that all schemes converge to machine accuracy. The slower convergence exhibited by the RHR solver is expected to be caused by its less dissipative character than the upwind MUSCL methods as in the application to the 2D linear advection equation, cf. the discussion of Fig. 9 in Section 4.3.1. Please note that no attempt has been made to accelerate the convergence to steady state for any of the schemes.

The computational cost for each time step is shown in Table 2. The RHR scheme is expected to be more costly than a MUSCL scheme, since solving the RHR relations (10)–(13) involves (among other operations) solving a cubic equation (53). One might be able to linearize this cubic equation in some way, thereby reducing the cost for solving it.

Method	Cost (ms)
RHR	44.4
MUSCL minmod	18.6
MUSCL MC	18.3
MUSCL van Albada	19.1

Table 2: Computational cost per time step for the isothermal Euler equations on a 80×80 grid

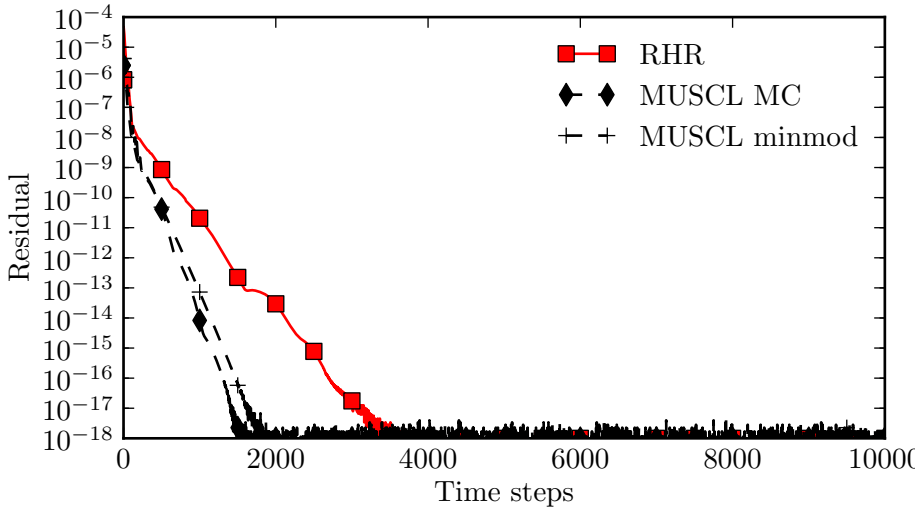


Figure 13: L^1 norm of the residual as the function of the number of time steps for the 2D isothermal Euler equations, 10×10 grid, $C = 0.3$.

4.5. Shallow water equations

The shallow water equations may include source terms due to bottom topography and bottom friction. Those source terms have been an important motivation to develop well-balanced methods. In this section, however, we will focus on solving the homogeneous shallow water equations to demonstrate how the RHR solver performs to maintain a flux balance in the steady state. The homogeneous shallow water equations read

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ hu \\ hv \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix} = \mathbf{0}, \tag{61}$$

where h is the water height above the bottom surface, and u and v are velocity components in the x- and y-directions, respectively. In the following we first show how the RHR relations are solved, derive a characteristic solver, and then present numerical results for a steady case demonstrating the order of convergence.

4.5.1. Solving the RHR relations

To calculate the half-states $\boldsymbol{u}_{i,j,\mathrm{E}}$, $\boldsymbol{u}_{i,j,\mathrm{W}}$, $\boldsymbol{u}_{i,j,\mathrm{S}}$, $\boldsymbol{u}_{i,j,\mathrm{N}}$, we need to solve the RHR relations (10)–(13). For the shallow water equations we choose to linearize these relations. By writing $\boldsymbol{u}_{\mathrm{E}} = \boldsymbol{u} + \boldsymbol{\epsilon}$ and replacing $\boldsymbol{u}_{\mathrm{W}}$ using Eq. (10), we can write Eq. (12) as

$$\boldsymbol{f}(\boldsymbol{u} + \boldsymbol{\epsilon}) - \boldsymbol{f}(\boldsymbol{u} - \boldsymbol{\epsilon}) = \Delta x \boldsymbol{q}_x. \tag{62}$$

We then linearize this equation, which yields

$$\boldsymbol{f}'(\boldsymbol{u})\boldsymbol{\epsilon} = \frac{\Delta x}{2}\boldsymbol{q}_x \tag{63}$$

where $\boldsymbol{f}'(\boldsymbol{u})$ is the Jacobian of $\boldsymbol{f}(\boldsymbol{u})$,

$$\boldsymbol{f}'(\boldsymbol{u}) = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{u_2^2}{u_1^2} + gu_1 & 2\frac{u_2}{u_1} & 0 \\ -\frac{u_2 u_3}{u_1^2} & \frac{u_3}{u_1} & \frac{u_2}{u_1} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -u^2 + gh & 2u & 0 \\ -uv & v & u \end{bmatrix}, \tag{64}$$

where $\boldsymbol{u} = (u_1, u_2, u_3) = (h, hu, hv)$. The first component of the equation system (63) can easily be solved for ϵ_2,

$$\epsilon_2 = \Delta x \frac{q_{x,1}}{2}. \tag{65}$$

We then solve for ϵ_1 from the second component of (63),

$$\epsilon_1 = \frac{1}{-u^2 + gh}(\Delta x \frac{q_{x,2}}{2} - 2u\epsilon_2) \tag{66}$$

Finally, the third component of (63) can be solved for ϵ_3,

$$\epsilon_3 = \frac{1}{u}\Delta x \frac{q_{x,3}}{2} + v\epsilon_1 - \frac{v}{u}\epsilon_2. \tag{67}$$

Should u be zero, the Jacobian matrix $\boldsymbol{f}'(\boldsymbol{u})$ is singular, and this final equation cannot be solved, in which case we assume ϵ_3 to be zero. The same applies to $u = \pm\sqrt{gh}$, in which case we must assume $\epsilon_1 = 0$. Having solved for $\boldsymbol{\epsilon}$, we can then calculate $\boldsymbol{u}_{\mathrm{E}}$ and $\boldsymbol{u}_{\mathrm{W}}$, and a similar procedure is used to calculate $\boldsymbol{u}_{\mathrm{N}}$ and $\boldsymbol{u}_{\mathrm{S}}$.

4.5.2. Characteristic solver

For the shallow water equations, we employ a characteristic based Riemann solver, similar to the one presented in Section 4.4.2. The full derivation is given in Appendix B. The central state is given by

$$h_{\mathrm{C}} = \frac{\sqrt{h_{\mathrm{L}}h_{\mathrm{R}}}(u_{\mathrm{L}} - u_{\mathrm{R}} + \sqrt{gh_{\mathrm{R}}} + \sqrt{gh_{\mathrm{L}}})}{\sqrt{gh_{\mathrm{R}}} + \sqrt{gh_{\mathrm{L}}}}, \tag{68}$$

$$u_{\mathrm{C}} = \frac{gh_{\mathrm{L}} - gh_{\mathrm{R}} + \sqrt{gh_{\mathrm{R}}}u_{\mathrm{R}} + \sqrt{gh_{\mathrm{L}}}u_{\mathrm{L}}}{\sqrt{gh_{\mathrm{R}}} + \sqrt{gh_{\mathrm{L}}}}. \tag{69}$$

In two dimensions, the velocity component parallel to the face is simply advected from the upwind side, i.e. $v_{\mathrm{C}} = v_{\mathrm{L}}$ if $u_{\mathrm{C}} > 0$, and $v_{\mathrm{C}} = v_{\mathrm{R}}$ if $u_{\mathrm{C}} < 0$. The Riemann flux is then given by $\boldsymbol{f}(\boldsymbol{u}_{\mathrm{C}})$.

4.5.3. Order of convergence

To demonstrate the order of convergence of the RHR solver for the shallow water equations, we apply the solver to a manufactured solution. We make the ansatz

$$h = h_0 - \frac{B^2x^2}{2g} - \frac{B^2y^2}{2g}, \tag{70}$$

$$u = Bx, \tag{71}$$

$$v = -By, \tag{72}$$

where B and h_0 are some constants; here we chose $B = 0.1$ and $\rho_0 = 1.0$. By inserting this ansatz into the steady shallow water equations, we find the source terms associated with this manufactured solution.

$$\frac{\partial}{\partial x}\begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix} = \begin{bmatrix} 1 \\ Bx \\ -By \end{bmatrix} \frac{B^3}{g}(y^2 - x^2) \equiv \boldsymbol{q}. \tag{73}$$

We will now use the given manufactured solution with the corresponding source term to investigate the order of convergence. We use the exact solution (70)–(72) to set the boundary conditions on all boundaries, while the source term $\boldsymbol{q}$ is computed from Eq. (73) in all cells. The numerical solution is then compared with the exact solution to calculate the error.

Figure 14 shows the L^2 error for height, $\|h - h_{\text{exact}}\|_2$, as a function of grid size n for a $n \times n$ grid, which demonstrates a second order convergence for both the MUSCL scheme with MC and minmod limiter, and the RHR solver with limiter. The picture is very similar to that in Fig. 12: The error of the RHR solver is systematically smaller than that of the MUSCL MC scheme, and almost one order of magnitude smaller than that of the MUSCL minmod scheme.

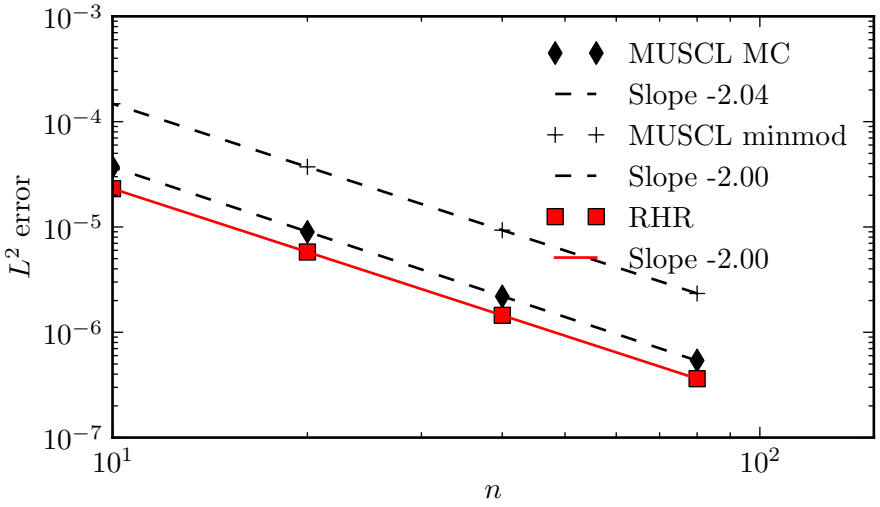


Figure 14: Grid convergence of the L^2 error of height for the 2D shallow water equations.

Figure 15 shows the L^1 norm of the residual as a function of the number of time steps for the same case, which confirms that all methods converge to machine precision. The RHR solver uses more time steps to reach steady state than the MUSCL upwind schemes. Again we attribute the slower convergence of the RHR solver to its lower numerical dissipation, cf. the discussions at the ends of sections 4.3.1 and 4.4.3.

Table 3 shows the computational cost of each time step. The RHR scheme is only slightly more expensive than the MUSCL scheme, since the RHR solver has the extra cost of

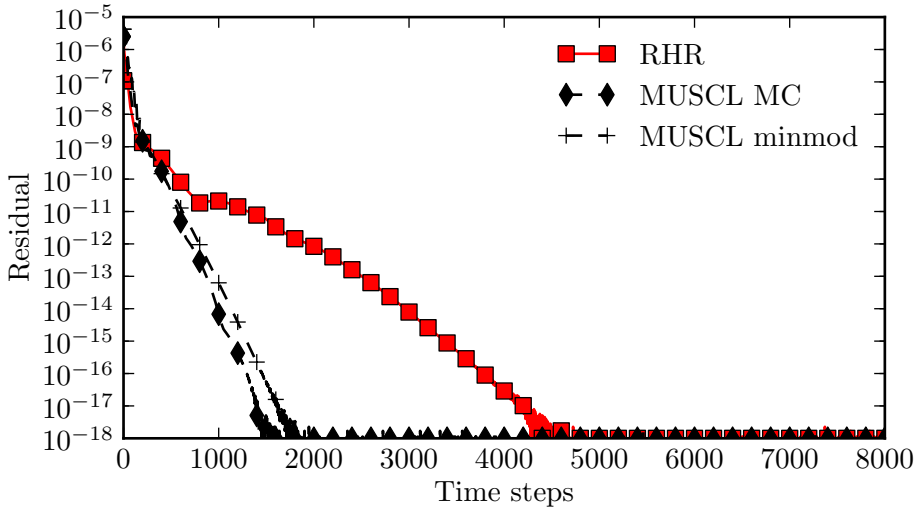


Figure 15: Residual as a function of number of time steps for the 2D shallow water equations, 10×10 grid, $C = 0.3$.

solving the RHR relations (10)–(13), which involves solving the three linear equations (65)–(67) for the eastern/western states, and three equivalent ones for the northern/southern states.

Method	Cost (ms)
RHR	16.5
MUSCL minmod	13.8
MUSCL MC	13.7
MUSCL van Albada	14.1

Table 3: Computational cost per time step for the shallow water equations on an 80×80 grid

4.6. The full Euler equations

In this section, we derive the RHR solver for the full Euler equations, and apply it to a channel flow case. The full Euler equations are given by

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ E\end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix}\rho u\\ \rho u^2 + p\\ \rho uv\\ u(E+p)\end{bmatrix} + \frac{\partial}{\partial y}\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2 + p\\ v(E+p)\end{bmatrix} = 0 \tag{74}$$

where $E = \rho(e + \frac{1}{2}(u^2 + v^2))$ is the total energy per unit volume. We use the following relation for a perfect gas:

$$E = \frac{p}{\gamma - 1} + \frac{\rho}{2}(u^2 + v^2), \tag{75}$$

where γ is the ratio of specific heats, which we set for air at standard conditions to 1.4 in our simulations.

4.6.1. Solving the RHR relations

Similar to the approach for the shallow water equations in Section 4.5.1, we linearize the RHR relations, and instead solve

$$\boldsymbol{f}'(\boldsymbol{u})\boldsymbol{\epsilon} = \frac{\Delta x}{2}\boldsymbol{q}_x, \tag{76}$$

where $\boldsymbol{u}_{\mathrm{E}} = \boldsymbol{u} + \boldsymbol{\epsilon}$. The Jacobian of the x-flux function $\boldsymbol{f}$ is

$$\boldsymbol{f}'(\boldsymbol{u}) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -u^2 + \frac{\gamma-1}{2}(u^2+v^2) & (3-\gamma)u & -(\gamma-1)v & \gamma-1 \\ -uv & v & u & 0 \\ -\frac{u}{\rho}(\gamma E - (\gamma-1)\rho(u^2+v^2)) & \frac{1}{\rho}(\gamma E - \frac{\gamma-1}{2}\rho(3u^2+v^2)) & -(\gamma-1)uv & \gamma u \end{bmatrix} \tag{77}$$

The linear system (76) can be solved relatively easily for $\boldsymbol{\epsilon}$, which also gives the half-states $\boldsymbol{u}_{\mathrm{E}} = \boldsymbol{u} + \boldsymbol{\epsilon}$ and $\boldsymbol{u}_{\mathrm{W}} = \boldsymbol{u} - \boldsymbol{\epsilon}$.

When solving the linear system (76), one might encounter situations where the Jacobian $\boldsymbol{f}'(\boldsymbol{u})$ is singular or close to singular. This will happen if we are at or close to a sonic point ($|u| = c$) or stagnation point ($u = 0$). For such situations, we introduce a singularity fix. It simply involves setting $\boldsymbol{\epsilon}$ to zero in those cells where the modulus of the determinant $D = \det \boldsymbol{f}'(\boldsymbol{u})$ is smaller than some threshold value δ, i.e. if $|D| = |u^2(u^2 - c^2)| < \delta$, which was tuned to $\delta = 2 \cdot 10^7$ for the test case presented in the next section. When we set $\boldsymbol{\epsilon}$ to zero, the RHR solver reduces to the ordinary upwind method. For the channel flow case presented in the next section, this singularity fix resolved an issue with lack of convergence around stagnation points in the y-direction when $\boldsymbol{g}'(\boldsymbol{u})$ is singular for $v = 0$.

4.6.2. A channel flow case

In this section, we present results for a channel flow with strong incoming cross flow. The channel is described by a rectangular domain $[0.5, 17] \times [0.5, 4]$. At the left edge ($x = 0.5$), there is inflow with velocity $u = 20$ and density $\rho = 1$. The upper and lower edges are solid walls, except in the range $2.25 < x < 9.75$, where there is incoming flow from the lower edge, with a velocity profile given by

$$v(x) = \begin{cases} 20 & \text{if } x < 6, \\ 20 - \frac{20}{7.5^2}(2x - 12)^2 & \text{if } x \geq 6. \end{cases} \tag{78}$$

On the outflow boundary ($x = 17$), the pressure is set to $p = 10^5$. The case is illustrated in Figure 16.

The simulations were run with $\Delta x = \Delta y = 0.5$, a CFL number of $C = 0.3$, and the same characteristic Riemann solver as in Ref. [1], derived along similar lines as in Appendix A. Figures 17, 18 and 19 show streamlines and isocontours for the total pressure

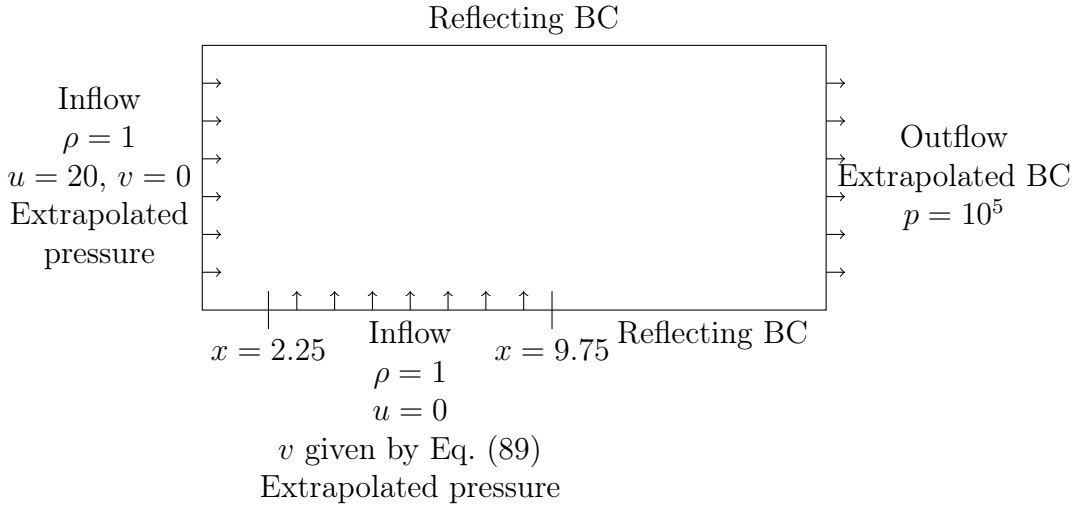


Figure 16: Channel flow case with boundary conditions (BC).

$p_{\text{tot}} = p + \frac{1}{2}\rho(u^2 + v^2)$ for an upwind solver, MUSCL with minmod limiter, and the RHR solver, respectively.

In this case, we do not know the exact solution. However, we do know that a correct solution should have isocontours of total pressure parallel to the streamlines. For the upwind method, the isocontours do not follow the streamlines very well, while the RHR solver and the MUSCL scheme seem to have far better agreement. Thus, the RHR solver performs well regarding the turning of the streamlines of the injected flow, although due to the singularity fix the RHR solver reduces to the upwind method in the regions where the flow is parallel or nearly parallel to one of the axes.

Table 4 shows the computational cost for each time step for the RHR solver, MUSCL minmod and the first-order upwind method. The RHR solver has a slightly higher cost than MUSCL, due to the cost of solving the linear system of equations (76).

Method	Cost (ms)
RHR	16.9
Upwind	13.1
MUSCL minmod	15.9
MUSCL van Albada	16.3

Table 4: Computational cost per time step for the full Euler equations on an 80×80 grid

5. Conclusions

We have developed a *Rankine-Hugoniot-Riemann* (RHR) solver for steady multidimensional conservation laws with source terms. The cross fluxes are treated as source terms,

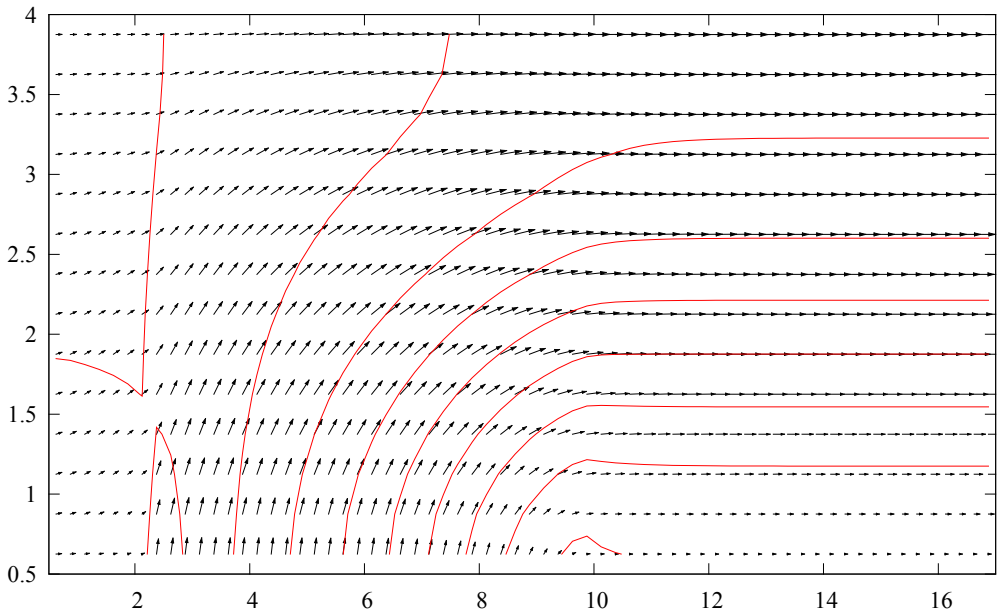


Figure 17: Contours of total pressure $p + \frac{1}{2}\rho(u^2 + v^2)$ (in red), and velocity vectors, for a first order upwind scheme on a 66×14 grid.

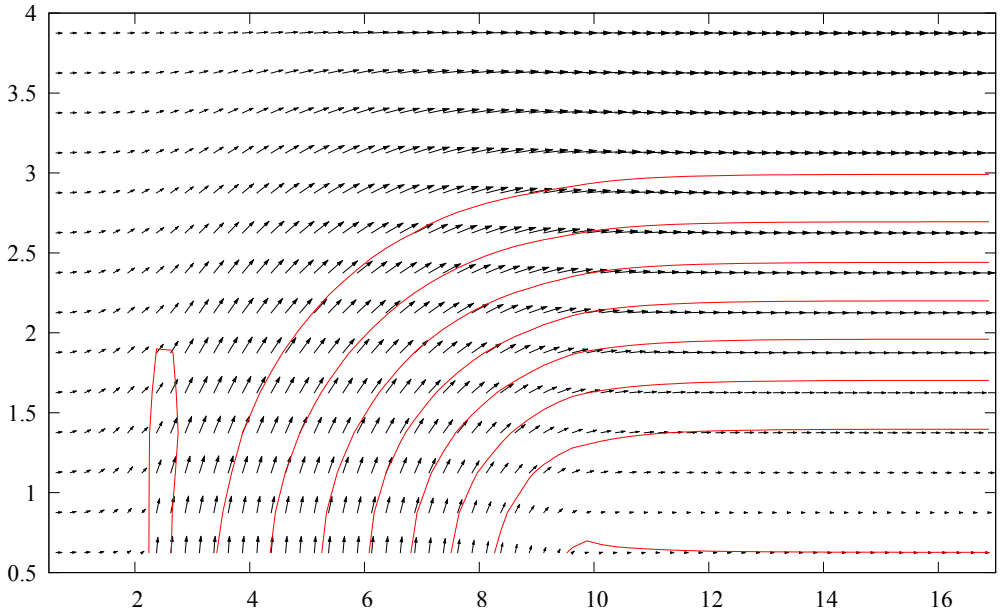


Figure 18: Contours of total pressure (in red) and velocity vectors, for MUSCL with minmod limiter on a 66×14 grid.

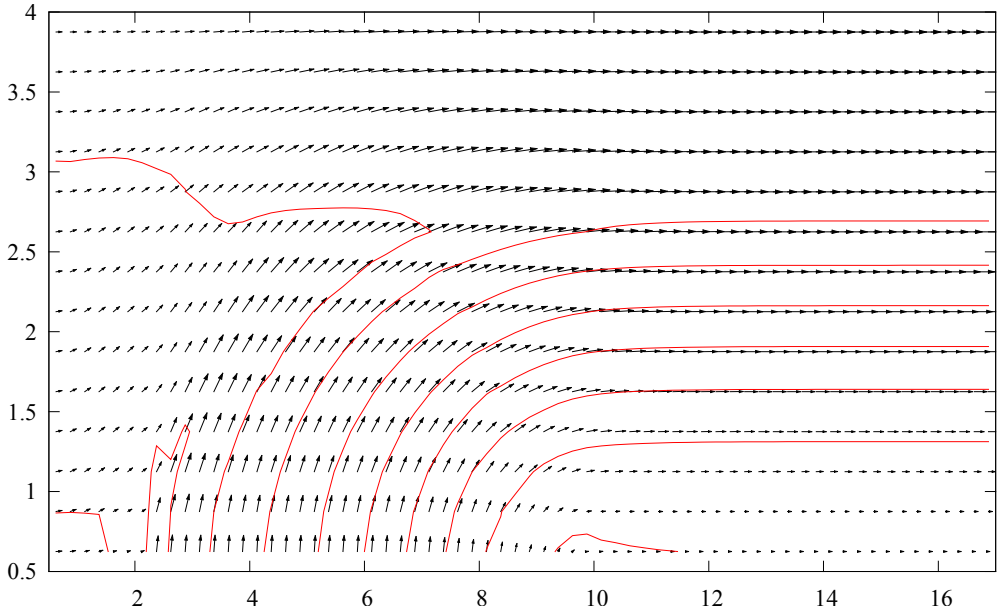


Figure 19: Contours of total pressure (in red) and velocity vectors, for the RHR solver with singularity fix on a 66×14 grid.

which are distributed as singular sources in the middle of each cell, leading to a jump in the solution given by a Rankine-Hugoniot condition. The resulting Riemann problems at the cell faces are then solved using a standard Riemann solver. In contrast to many other schemes treating multidimensionality and source terms, there is no need for special Riemann solvers. We introduced a limiting procedure similar to a total variation diminishing (TVD) enforcement, to avoid instabilities.

We were able to prove that on rectangular grids, the RHR solver yields a second order accurate numerical solution for the 2D linear advection equation with a linear source term, and that the solution can be advected exactly if the advection velocity is diagonal on the grid. We have also investigated the properties of the RHR solver numerically, which confirmed that the scheme is of second order accuracy both for the 2D linear advection equation, the 2D isothermal Euler equations and the 2D shallow water equations. Furthermore, the RHR solver has an error which is smaller than that of a second-order MUSCL scheme for these cases. The solver was also applied to a channel flow case using the full Euler equations. A singularity fix was necessary to apply the RHR solver close to sonic or stagnation points. Future work will hopefully eliminate the need for this singularity fix.

The stencil of the RHR scheme has the advantage of being compact, as the numerical fluxes of a cell only depend on the numerical solutions of the cell and its neighbours which have a face or a corner in common with the cell. The basic ideas outlined here for the isothermal Euler equations, the shallow water equations and the full Euler equations should carry over to other conservation laws with source terms, such as the shallow water equations with a topography source term or two-phase flow equations.

Acknowledgements

The first author was financed through the CO_2 Dynamics project, and acknowledges the support from the Research Council of Norway (189978), Gassco AS, Statoil Petroleum AS and Vattenfall AB.

References

[1] P. Jenny, B. Müller, Rankine–Hugoniot–Riemann solver considering source terms and multidimensional effects, J. Comput. Phys. 145 (1997) 575–610.
[2] H. B. Stewart, B. Wendroff, Two-phase flow: Models and methods, J. Comput. Phys. 56 (1984) 363–409.
[3] H. Lund, P. Aursand, Two-phase flow of CO_2 with phase transfer, Energy Procedia 23 (2012) 246–255.
[4] L. Gosse, A well-balanced flux-vector splitting scheme designed for hyperbolic systems conservation laws with source terms, Comput. Math. Appl. 39 (2000) 135–159.
[5] R. Saurel, F. Petitpas, R. Abgrall, Modelling phase transition in metastable liquids: application to cavitating and flashing flows, J Fluid Mech 607 (2008) 313–350.
[6] R. J. LeVeque, Balancing source terms and flux gradients in high-resolution Godunov methods: The quasi-steady wave-propagation algorithm, J. Comput. Phys. 146 (1998) 346–365.
[7] D. S. Bale, R. J. LeVeque, S. Mitran, J. Rossmanith, A wave-propagation method for conservation laws and balance laws with spatially varying flux functions, SIAM J. Sci. Comput. 24 (3) (2002) 955–978.
[8] R. J. LeVeque, D. S. Bale, Wave propagation methods for conservation laws with source terms, in: International Series of Numerical Mathematics, Vol. 130, Birkhäuser, 1999.
[9] R. J. LeVeque, Finite Volume Methods for Hyperbolic Problems, Cambridge University Press, 2002.
[10] A. Bermudez, M. E. Vazquez, Upwind methods for hyperbolic conservation laws with source terms, Comput. Fluids 23 (8) (1994) 1049–1071.
[11] R. Donat, A. Martinez-Gavara, Hybrid second order schemes for scalar balance laws, J. Sci. Comput. 48 (2011) 52–69.
[12] M. E. Hubbard, P. García-Navarro, Flux difference splitting and the balancing of source terms and flux gradients, J. Comput. Phys. 165 (2000) 89–125.
[13] R. J. LeVeque, A well-balanced path-integral f-wave method for hyperbolic problems with source terms, J. Sci. Comput. 48 (2011) 209–226.
[14] E. Audusse, F. Bouchut, M.-O. Bristeau, R. Klein, B. Perthame, A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows, SIAM J. Sci. Comput. 25 (6) (2004) 2050–2065.
[15] J. Murillo, P. García-Navarro, Weak solutions for partial differential equations with source terms: Application to the shallow water equations, J. Comput. Phys. 229 (2010) 4327–4368.
[16] S. Noelle, N. Pankratz, G. Puppo, and J. R. Natvig, Well-balanced finite volume schemes of arbitrary order of accuracy for shallow water flows. J. Comput. Phys. 213 (2006) 474–499.
[17] S. Noelle, Y. Xing, and C.-W. Shu. High-order well-balanced finite volume WENO schemes for shallow water equation with moving water. J. Comput. Phys. 226 (2007) 29–58.
[18] S. Noelle, Y. Xing, and C.-W. Shu. High-order well-balanced schemes. In G. Puppo and G. Russo, editors, Numerical Methods for Balance Laws. Quaderni di Matematica 24 (2010), pages 1–66.
[19] Y. Xing, C.-W. Shu, and S. Noelle. On the advantage of well-balanced schemes for moving-water equilibria of the shallow water equations. J. Sci. Comput. 48 (2011) 339–349.
[20] M. Lukáčová-Medvid'ová, S. Noelle, M. Kraft, Well-balanced finite volume evolution Galerkin methods for the shallow water equations. J. Comput. Phys. 221 (1) (2007) 122–147.
[21] M. Ricchiuto, A. Bollermann, Stabilized residual distribution for shallow water simulations, J. Comput. Phys. 228 (4) (2009) 1071–1115.

[22] C. Parés, Numerical methods for nonconservative hyperbolic systems: a theoretical framework, SIAM J. Numer. Anal. 44 (1) (2006) 300–321.
[23] R. J. LeVeque, M. Pelanti, A class of approximate Riemann solvers and their relation to relaxation schemes, J. Comput. Phys. 172 (2001) 572–591.
[24] F. Müller, Rankine-Hugoniot-Riemann solver for two-dimensional conservation laws, Master's thesis, ETH Zürich (2010).
[25] S. Gottlieb, D. Ketcheson, C.-W. Shu, Strong stability preserving Runge-Kutta and multistep time discretizations, World Scientific, 2011.
[26] P. J. Roache, Verification and Validation in Computational Science and Engineering, Hermosa Publishers, Albuquerque, USA, 1998.
[27] J. Sesterhenn, B. Müller, H. Thomann, A simple characteristic flux evaluation for subsonic flow, in: 2nd ECCOMAS CFD Conf., Wiley, Chichester, 1994, p. 57.
[28] J. Sesterhenn, Zur numerischen Berechnung kompressibler Strömungen bei kleinen Mach-Zahlen, Ph.D. thesis, ETH Zürich, diss. ETH Nr. 11334 (1995).

Appendix A. Characteristic solver for the isothermal Euler equations

Here we derive a characteristic-based Riemann solver, similar to the one by Sesterhenn et al. [27, 28]. To derive the characteristic quantities, we consider the isothermal Euler equations in one dimension, written in the quasi-linear form

$$\begin{bmatrix}\rho\\ \rho u\end{bmatrix}_t + \underbrace{\begin{bmatrix}0 & 1\\ c^2-u^2 & 2u\end{bmatrix}}_{\boldsymbol{J}}\begin{bmatrix}\rho\\ \rho u\end{bmatrix}_x = 0. \tag{A.1}$$

We now rewrite this system to formulate it using the primitive variables $\boldsymbol{v} = [\rho, u]^\top$,

$$\begin{bmatrix}\rho\\ u\end{bmatrix}_t + \underbrace{\begin{bmatrix}u & \rho\\ \frac{c^2}{\rho} & u\end{bmatrix}}_{\boldsymbol{J'}}\begin{bmatrix}\rho\\ u\end{bmatrix}_x = 0. \tag{A.2}$$

The eigenvalues of the Jacobian matrix $\boldsymbol{J'}$ are $\lambda_1 = u - c$ and $\lambda_2 = u + c$. Solving for the eigenvectors of $\boldsymbol{J'}$ yields the right eigenvector matrix

$$\boldsymbol{R}(u) = \begin{bmatrix}\rho & \rho\\ -c & c\end{bmatrix}. \tag{A.3}$$

We then determine the inverse (left eigenvector) matrix $\boldsymbol{R}^{-1}$ and multiply $\boldsymbol{R}_0^{-1} = \boldsymbol{R}^{-1}(\boldsymbol{v}_0)$ by the primitive variables $\boldsymbol{v} = [\rho, u]^\top$, which yields the characteristic variables

$$\boldsymbol{w} = \boldsymbol{R}_0^{-1}\boldsymbol{v} = \frac{1}{2\rho_0 c}\begin{bmatrix}c & -\rho_0\\ c & \rho_0\end{bmatrix}\begin{bmatrix}\rho\\ u\end{bmatrix} = \frac{1}{2\rho_0 c}\begin{bmatrix}\rho c - \rho_0 u\\ \rho c + \rho_0 u\end{bmatrix}, \tag{A.4}$$

where we have evaluated the matrix $\boldsymbol{R}_0^{-1} = \boldsymbol{R}^{-1}(\boldsymbol{v}_0)$ at some point of linearization $\boldsymbol{v} = \boldsymbol{v}_0$.

In the context of Fig. A.20, we calculate the state in the region C by assuming that the characteristic variables are constant along the solid lines from the regions L and R

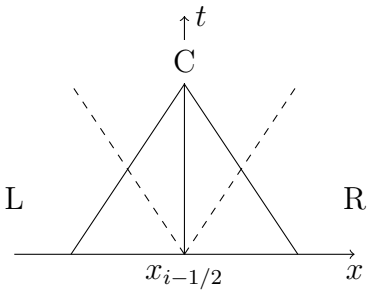


Figure A.20: A Riemann problem at $x_{i-1/2}$ giving rise to two waves, shown by dashed lines. The characteristics are shown by solid lines.

to region C. The dashed lines are the waves resulting from the Riemann problem. From Eq. (A.4) we then derive the approximate relations

$$\rho_{\mathrm{R}} c - \rho_{\mathrm{C}} u_{\mathrm{R}} = \rho_{\mathrm{C}} c - \rho_{\mathrm{C}} u_{\mathrm{C}}, \tag{A.5}$$

$$\rho_{\mathrm{L}} c + \rho_{\mathrm{C}} u_{\mathrm{L}} = \rho_{\mathrm{C}} c + \rho_{\mathrm{C}} u_{\mathrm{C}}, \tag{A.6}$$

which follows from the fact that the characteristic variables (A.4) are constant along the characteristics, shown by solid lines in Figure A.20. Here we have chosen to linearize w_1 and w_2 around $\boldsymbol{v}_0 = \boldsymbol{v}_{\mathrm{C}}$. Solving these two relations for u_{C} and ρ_{C} yields

$$\rho_{\mathrm{C}} = \frac{c(\rho_{\mathrm{R}} + \rho_{\mathrm{L}})}{2c + u_{\mathrm{R}} - u_{\mathrm{L}}}, \tag{A.7}$$

$$u_{\mathrm{C}} = \frac{c(\rho_{\mathrm{L}} - \rho_{\mathrm{R}}) + \rho_{\mathrm{C}} u_{\mathrm{R}} + \rho_{\mathrm{C}} u_{\mathrm{L}}}{2\rho_{\mathrm{C}}}. \tag{A.8}$$

Appendix B. Characteristic solver for the shallow water equations

To derive the characteristic quantities, we consider the shallow water equations in one dimension, written in the quasi-linear form

$$\begin{bmatrix} h \\ hu \end{bmatrix}_t + \underbrace{\begin{bmatrix} 0 & 1 \\ gh - u^2 & 2u \end{bmatrix}}_{\boldsymbol{J}} \begin{bmatrix} h \\ hu \end{bmatrix}_x = 0. \tag{B.1}$$

We then rewrite this system to one formulated using the primitive variables $\boldsymbol{v} = [h, u]^\top$,

$$\begin{bmatrix} h \\ u \end{bmatrix}_t + \underbrace{\begin{bmatrix} u & h \\ g & u \end{bmatrix}}_{\boldsymbol{J'}} \begin{bmatrix} h \\ u \end{bmatrix}_x = 0. \tag{B.2}$$

The eigenvalues of the Jacobian matrix $\boldsymbol{J'}$ are $\lambda_1 = u - \sqrt{gh}$ and $\lambda_2 = u + \sqrt{gh}$. We recognize that the quasi-linear formulation of the shallow water equations is identical to

that of the isothermal Euler equations if we only replace c by $\sqrt{gh}$. We then derive the characteristic variables

$$\boldsymbol{w} = \boldsymbol{R}_0^{-1}\boldsymbol{v} = \frac{1}{2h_0\sqrt{gh_0}}\begin{bmatrix}\sqrt{gh_0} & -h_0\\ \sqrt{gh_0} & h_0\end{bmatrix}\begin{bmatrix}h\\ u\end{bmatrix} = \frac{1}{2h_0\sqrt{gh_0}}\begin{bmatrix}h\sqrt{gh_0}-h_0u\\ h\sqrt{gh_0}+h_0u\end{bmatrix}, \tag{B.3}$$

where we have evaluated the matrix $\boldsymbol{R}_0^{-1} = \boldsymbol{R}^{-1}(\boldsymbol{v}_0)$ at some point of linearization $\boldsymbol{v} = \boldsymbol{v}_0$. From Eq. (B.3) we then derive the approximate relations

$$h_{\mathrm{C}}\sqrt{gh_{\mathrm{R}}} - h_{\mathrm{R}}u_{\mathrm{C}} = h_{\mathrm{R}}\sqrt{gh_{\mathrm{R}}} - h_{\mathrm{R}}u_{\mathrm{R}}, \tag{B.4}$$

$$h_{\mathrm{C}}\sqrt{gh_{\mathrm{L}}} + h_{\mathrm{L}}u_{\mathrm{C}} = h_{\mathrm{L}}\sqrt{gh_{\mathrm{L}}} + h_{\mathrm{L}}u_{\mathrm{L}}, \tag{B.5}$$

where we have linearized w_1 and w_2 at the right and left state, $\boldsymbol{v}_{\mathrm{R}}$ and $\boldsymbol{v}_{\mathrm{L}}$, respectively. We solve Eqs. (B.4) and (B.5) for h_{C} and u_{C},

$$h_{\mathrm{C}} = \frac{\sqrt{h_{\mathrm{L}}h_{\mathrm{R}}}(u_{\mathrm{L}} - u_{\mathrm{R}} + \sqrt{gh_{\mathrm{R}}} + \sqrt{gh_{\mathrm{L}}})}{\sqrt{gh_{\mathrm{R}}} + \sqrt{gh_{\mathrm{L}}}}, \tag{B.6}$$

$$u_{\mathrm{C}} = \frac{gh_{\mathrm{L}} - gh_{\mathrm{R}} + \sqrt{gh_{\mathrm{R}}}u_{\mathrm{R}} + \sqrt{gh_{\mathrm{L}}}u_{\mathrm{L}}}{\sqrt{gh_{\mathrm{R}}} + \sqrt{gh_{\mathrm{L}}}}. \tag{B.7}$$