A wave propagation method for three-dimensional hyperbolic conservation laws

J. O. Langseth * R. J. LeVeque †

January 28, 1997

Abstract

A class of wave propagation algorithms for three-dimensional conservation laws is developed. These unsplit finite volume methods are based on solving one-dimensional Riemann problems at the cell interfaces and applying flux-limiter functions to suppress oscillations arising from second derivative terms. Waves emanating from the Riemann problem are further split by solving Riemann problems in the transverse direction to model cross-derivative terms. Due to proper upwinding, the method is stable for Courant numbers up to one. Several examples using the Euler equations are included.

1 Introduction

This paper describes an unsplit method for solving three-dimensional conservation laws, i.e. equations of the form

$$u_t + f(u)_x + g(u)_y + h(u)_z = 0, \tag{1}$$

where $u \in \mathbb{R}^m$ is the conserved quantity. It is well known that these equations may develop shock waves or discontinuities. The existence of regions where the solution changes abruptly places special requirements on the numerical methods to be used. Traditionally, simple numerical schemes suffer from dissipative and dispersive effects, due to truncation, resulting in inappropriate representation of these discontinuities. This is reflected in the solution as spurious oscillations or excessive smearing in the vicinity of the discontinuity. Since the occurrence of discontinuous waves is a significant feature of hyperbolic problems, much effort has been made in order to construct robust methods producing sharp and monotone representations.

One special initial value problem is of major importance, both in its own and in the development of efficient numerical methods. This is the Riemann problem which consists of the equation $u_t + f(u)_x = 0$ with the initial condition

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

where u_l and u_r are constants. With certain assumptions on the flux functions f, it is always possible, in principle, to solve the Riemann problem if the states u_l and u_r are sufficiently "close", cf. [27]. The solution consists of waves travelling with finite velocities. These waves may either

*Norwegian Defence Research Establishment, P.O.Box 25, N-2007 Kjeller, Norway. This work has been supported by The Norwegian Research Council (NFR) through the program no. 101039/420.

†Departments of Mathematics and Applied Mathematics, University of Washington, Seattle, WA98195. This work has been supported by DOE grant DE-FG03-96ER25292 and NSF grants DMS-9505021 and DMS-9626645.

be discontinuous waves like shock waves or smooth rarefaction waves. The solution of this initial-value problem depends on the ratio x/t. Due to this simplification, it is possible to solve any scalar problem. Also for some important systems, the Riemann problem is solved. This is the case for the Euler equations in gas dynamics [9], [27]. The procedure for constructing the solution of a Riemann problem will be called a Riemann solver.

The history of the development of numerical methods for hyperbolic conservation laws is long and rich. The wave propagation method described in this paper falls into the tradition of methods based on solving Riemann problems. Godunov[8] used this mathematical problem as a building block for constructing a first order upwind scheme for the equations of gas dynamics. The key observation is that if the approximation is viewed as a piecewise constant function, local Riemann problems are introduced at cell boundaries. However, the low accuracy and the complexity of this method meant that other methods were mostly used for a long time. The methods in use were typically based on modifications of simple first or second order schemes to improve the representation of discontinuities. For second order schemes artificial viscosity was added in order to damp dispersive effects, cf. Richtmyer and Morton[20]. On the other hand, first order schemes were modified by reducing the dissipation near discontinuities. A typical example of the latter approach are the Flux Corrected Transport scheme by Boris and Bock [3]. These methods give higher order accuracy, but tend to involve problem dependent tuning.

Two decades after the initial work of Godunov, Van Leer [34],[35] resurrected Godunov's work. The accuracy was increased by constructing a piecewise linear approximation at the beginning of each time step. His MUSCL scheme was made monotone by the use of limiters, much in the same way as here. Colella and Woodward[5] used a similar strategy in the construction of the Piecewise Parabolic Method (PPM), which is formally third order accurate in space. Some relevant references describing numerical methods for one-dimensional conservation laws are found in [13].

A common approach when solving multi-dimensional hyperbolic problems is to apply dimensional splitting, cf. [6],[28]. The idea is to iterate on one-dimensional problems. The popularity of these algorithms is due to the fact that they produce surprisingly good results and that the strategy is very simple. Any one-dimensional scheme is easily extended to the multi-dimensional case using this approach. However, it is well known that dimensional splitting has several disadvantages. Since the strategy only involves flow in the coordinate directions, the solution is affected by the grid orientation. Discontinuities travelling obliquely to the grid experience more smearing than those travelling in the coordinate directions. The implementation of boundary conditions may also be complicated using this strategy. It is also well known that certain one-dimensional methods like the Random Choice Method may behave badly in split-operator applications. The reason is that the one-dimensional scheme simply is too good, and that artificial waves created in the splitting will survive [22].

In unsplit methods, information is propagated in a multi-dimensional way. The unsplit scheme to be presented was first described in an unfinished form in [10]. One-dimensional Riemann problems are solved at the interfaces. Limiter functions are applied to suppress spurious oscillations arising from second derivative terms. The left-going and right-going waves are split into parts propagating in the transverse direction by solving Riemann problems in coordinate directions tangential to the interfaces. This models cross-derivative terms necessary for obtaining both a stable and formally second order scheme. The scheme extends the approach used for two space-dimensions [14], [16], and the advection scheme for three-dimensional problems considered in [15]. The 3D wave propagation scheme is implemented in Fortran, and will be included in the software package `CLAWPACK` (Conservation LAW PACKage) [17]. This package includes routines for solving a wide range of hyperbolic problems, and contains a selection of different solvers, boundary conditions etc. `CLAWPACK` is freely available on the Web, from the URL http://www.amath.washington.edu/~rjl/clawpack.html

This method is similar in spirit to other multi-dimensional methods using one-dimensional Riemann problems at the cell interfaces, cf. [1], [2], [4], [12], [19], [23], [25], [26], [31]. However, an algorithmic difference is that the Riemann problem is based on the solution values at the beginning of the time step, and that cross-derivative terms are implemented due to the transverse propagation of the emanating waves. For example, in the method of Colella [4], the states involved in the Riemann problem are based on an interpolation in which the transverse terms are included.

The 3D algorithm will be derived by first presenting the one-dimensional and two-dimensional versions. The paper is organized as follows. In Section 2, the one-dimensional method is discussed in terms of wave propagation. The two-dimensional algorithm is presented in Section 3, and in Section 4, the algorithm is extended to the full three-dimensional case. The stability properties of the derived 3D schemes are analyzed in Section 5, using a von Neumann approach. In Section 6 numerical results are given for three test problems involving the Euler equations: a radially symmetric smooth solution where second order accuracy can be verified, an initially spherical shock wave expanding in a slab between two walls, and finally a problem involving a shock-bubble interaction. The algorithm is also applicable to problems not in conservation form, but this report will not focus on this subject. Nevertheless, this section also includes a problem involving a scalar advection equation with space and time dependent velocity field. In Section 7, some concluding remarks are given. Finally, in the Appendix, the complete 3D algorithm is written in a form close to the (present) CLAWPACK implementation. The problem-dependent routines provided by the user are briefly described.

2 One space dimension

Consider the one-dimensional conservation law,

$$u_t + f(u)_x = 0, \qquad u \in \mathbb{R}^m. \tag{3}$$

The problem is assumed to be hyperbolic, hence the Jacobian $f'(u)$ has real eigenvalues and a complete set of eigenvectors. Define a computational grid with interfaces $x_{i-1/2} = (i - 1/2)\Delta x$, time levels $t_n = n\Delta t$ and cells $C_i = [x_{i-1/2}, x_{i+1/2}]$. For simplicity, the grid size Δx and time step Δt are assumed to be constants. Integrating (3) over $C_i \times [t_n, t_{n+1}]$ gives,

$$\int_{x_{i-1/2}}^{x_{i+1/2}} u(x, t_{n+1})\, dx = \int_{x_{i-1/2}}^{x_{i+1/2}} u(x, t_n)\, dx - \left(\int_{t_n}^{t_{n+1}} f(u(x_{i+1/2}, t))\, dt - \int_{t_n}^{t_{n+1}} f(u(x_{i-1/2}, t))\, dt \right).$$

This integral form motivates the conservative numerical scheme

$$\bar{U}_i = U_i - \frac{\Delta t}{\Delta x}(F_{i+1} - F_i), \tag{4}$$

where U_i approximates the average of the exact solution on C_i at time $t = t_n$, and the numerical flux F_i approximates the time average of the exact flux taken at the interface between the cells C_{i-1} and C_i, i.e.

$$U_i \approx \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} u(x, t_n)\, dx, \qquad F_i \approx \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} f(u(x_{i-1/2}, t))\, dt.$$

2.1 Linear systems

Consider the linear system

$$f(u) = Au, \tag{5}$$

where A is a constant $m \times m$ matrix with eigenvalues λ^p and eigenvectors r^p. Since A has a complete set of eigenvectors, the matrix is diagonalizable, i.e.

$$A = R\Lambda R^{-1},$$

where $R = [r^1|r^2|\dots|r^m]$ is the eigenvector matrix and $\Lambda = \mathrm{diag}(\lambda^1, \lambda^2, \dots, \lambda^m)$. For $a \in \mathbb{R}$ let

$$\begin{aligned} a^+ &= \max(0, a) \\ a^- &= \min(0, a). \end{aligned} \tag{6}$$

Define the positive and negative parts of A as

$$A^+ = R\Lambda^+ R^{-1}, \qquad A^- = R\Lambda^- R^{-1}, \tag{7}$$

where

$$\Lambda^\pm = \mathrm{diag}(\lambda^{1\pm}, \lambda^{2\pm}, \dots, \lambda^{m\pm}).$$

The solution of the Riemann problem defined by U_{i-1} and U_i consists of m discontinuities moving with velocities λ^p. Each jump is a scalar multiple of the eigenvector, i.e. $\mathscr{W}_i^p = \alpha_i^p r^p$. Hence the jump $\Delta U_i = U_i - U_{i-1}$ may be written as

$$\Delta U_i = \sum_p \mathscr{W}_i^p.$$

Let $RP_x(u_l, u_r)$ denote the solution at $x/t = 0$ of the Riemann problem with left state u_l and right state u_r. The subscript indicate in which coordinate direction the Riemann problem is solved. (This notation is redundant here, but will be useful later.)

The first order Godunov flux is defined as

$$F_i^G = f(U_i^0),$$

where $U_i^0 = RP_x(U_{i-1}, U_i)$. For the linear system this intermediate state equals

$$\begin{aligned} U_i^0 &= U_{i-1} + \sum_{\lambda^p<0} \mathscr{W}_i^p & (8) \\ &= U_i - \sum_{\lambda^p>0} \mathscr{W}_i^p. & (9) \end{aligned}$$

Hence, the Godunov flux reads,

$$\begin{aligned} F_i^G &= AU_{i-1} + A^-\Delta U_i & (10) \\ &= AU_i - A^+\Delta U_i. & (11) \end{aligned}$$

For the numerical scheme (4) we get

$$\bar{U}_i = U_i - \frac{\Delta t}{\Delta x}(A^+\Delta U_i + A^-\Delta U_{i+1}). \tag{12}$$

The expression $A^+\Delta U_i = \sum_p (\lambda^p)^+ \mathscr{W}_i^p$ equals the flux contribution from the waves entering cell C_i from the Riemann problem at the left interface, while $A^-\Delta U_{i+1}$ equals the flux from the waves entering the same cell from the Riemann problem at the right interface. This clearly illustrates the upwind nature of the scheme. These piecewise constant waves, emanating from the Riemann problem, will be named increment waves.

2.2 Second order corrections

The scheme considered above is formally first order accurate only. In order to increase the order, (4) is rewritten as

$$\bar{U}_i = U_i + \Delta_i^{up} - \frac{\Delta t}{\Delta x}(\widetilde{F}_{i+1} - \widetilde{F}_i), \tag{13}$$

where Δ_i^{up} equals the upwind flux (or Godunov flux) obtained from (12), i.e.

$$\Delta_i^{up} = -\frac{\Delta t}{\Delta x}(A^+ \Delta U_i + A^- \Delta U_{i+1}). \tag{14}$$

The term $\widetilde{F}_i$ is used to update the solution so that second order accuracy is achieved.

The flux for the second order Lax-Wendroff scheme may be written as the Godunov flux plus a correction,

$$\begin{aligned} F_i &= \frac{1}{2}A(U_i + U_{i-1}) + \frac{1}{2}\frac{\Delta t}{\Delta x}A(U_i - U_{i-1}) \\ &= F_i^G + \frac{1}{2}|A|\left(1 - \frac{\Delta t}{\Delta x}|A|\right)\Delta U_i, \end{aligned}$$

where $|A| = A^+ - A^-$. Hence, a natural choice for $\widetilde{F}_i$ is

$$\begin{aligned} \widetilde{F}_i &= \frac{1}{2}|A|\left(1 - \frac{\Delta t}{\Delta x}|A|\right)\Delta U_i \\ &= \frac{1}{2}\sum_p |\lambda^p|\left(1 - \frac{\Delta t}{\Delta x}|\lambda^p|\right)\mathscr{W}_i^p. \end{aligned} \tag{15}$$

The Godunov scheme exhibits strong numerical dissipation, and discontinuities in the solution are smeared causing low accuracy. The Lax-Wendroff scheme, on the other hand, is more accurate in smooth parts of the solution. But near discontinuities, numerical dispersion generates oscillations, also reducing the accuracy.

A successful approach to suppress these oscillations is to apply flux limiting. This is obtained by replacing the wave $\mathscr{W}_i^p = \alpha_i^p r^p$ by

$$\widetilde{\mathscr{W}}_i^p = \Phi(\theta_i^p)\mathscr{W}_i^p, \tag{16}$$

where θ_i^p measures the smoothness of the solution. A standard way of doing this is to consider the ratio of wave strengths α_i^p in the upwind direction, i.e.

$$\theta_i^p = \begin{cases} \alpha_{i-1}^p/\alpha_i^p, & \lambda^p > 0 \\ \alpha_{i+1}^p/\alpha_i^p, & \lambda^p < 0. \end{cases} \tag{17}$$

The limiters used here are originally constructed in order to ensure both second order accuracy and TVD properties for scalar conservation laws. For a detailed discussion, cf. [13] and [30]. The TVD concept is not applicable in the system case directly, but applying this strategy on the characteristic equations, as done implicitly here, produces steep and monotone approximations of discontinuities.

Note that $\Phi = 1$ yields the Lax-Wendroff scheme, while $\Phi = 0$ yields the Godunov scheme. Some common limiters are:

$$\begin{aligned}
\text{minmod:} \quad & \Phi(\theta) = \max(0, \min(1, \theta)) \\
\text{superbee:} \quad & \Phi(\theta) = \max(0, \min(1, 2\theta), \min(2, \theta)) \\
\text{van Leer:} \quad & \Phi(\theta) = \frac{\theta + |\theta|}{1 + \theta} \\
\text{monotonized centered (MC):} \quad & \Phi(\theta) = \max(0, \min((1 + \theta)/2, 2, 2\theta))
\end{aligned}$$

Minmod is the most diffusive limiter of the above, in the sense that it adds less downwind contribution. This limiter selects the wave with the smallest norm of the two compared, provided the jumps across the waves are in the same direction, i.e. the wave strengths have common sign. If not, the wave is entirely suppressed. On the other hand, the superbee limiter is known to be "overcompressive", i.e. it tends to sharpen profiles into discontinuities. The MC-limiter seems to be a good choice in most situations.

The waves used in the second order update, i.e. $\frac{1}{2}|\lambda^p|(1 - \frac{\Delta t}{\Delta x}|\lambda^p|)\widetilde{\mathscr{W}}_i^p$, will be referred to as correction waves.

2.3 Nonlinear systems

Motivated by the notation used for linear systems, the numerical scheme to be used in the nonlinear case reads

$$\bar{U}_i = U_i - \frac{\Delta t}{\Delta x}(\mathscr{A}^+ \Delta U_i + \mathscr{A}^- \Delta U_{i+1}) - \frac{\Delta t}{\Delta x}(\widetilde{F}_{i+1} - \widetilde{F}_i). \tag{18}$$

The advantage of using this formulation instead of (4) will be discussed below. This scheme is conservative provided

$$f(U_i) - f(U_{i-1}) = \mathscr{A}^+ \Delta U_i + \mathscr{A}^- \Delta U_i. \tag{19}$$

This is trivially satisfies for (5) since $\mathscr{A}^\pm \Delta U_i = A^\pm \Delta U_i$ and $A = A^- + A^+$. The process of finding $\mathscr{A}^\pm \Delta U_i$ so that (19) is satisfied will be called *flux difference splitting*. In addition to this splitting, waves $\mathscr{W}_i^p$ and associated velocities λ^p have to be specified. There are M_w waves involved, not necessarily equal to m.

The flux splitting may be achieved in several ways. Here, two approaches will be briefly discussed.

Approach 1: In the Roe solver [13], [21], a local approximation A_i to the Jacobian $f'(u)$ is defined. This involves finding a (Roe-averaged) state U_i^*, depending on U_{i-1} and U_i so that if $A_i = f'(U_i^*)$, then

$$A_i(U_i - U_{i-1}) = f(U_i) - f(U_{i-1}).$$

This property ensures conservation. After constructing this matrix, the algorithm proceeds exactly as in the linear case.

This matrix is derived for several important problems like the Euler equations of gas dynamics. The flux difference splitting is defined in the obvious way as

$$\begin{aligned}
\mathscr{A}^+ \Delta U_i = A_i^+ \Delta U_i = \sum_p (\lambda_i^p)^+ \mathscr{W}_i^p \\
\mathscr{A}^- \Delta U_i = A_i^- \Delta U_i = \sum_p (\lambda_i^p)^- \mathscr{W}_i^p
\end{aligned}$$

Note that the eigenvalues and eigenvectors are space dependent. Since the solution only consists of discontinuities, rarefaction waves may cause problems. If these smooth waves are transonic, i.e. involves both positive and negative velocities, the numerical solution may contain physically incorrect shocks. In order to correct this, an entropy fix should be applied [13].

Approach 2: In some situations, it might be advantageous to use the exact solution of the Riemann problem. For example, when applied to the Euler equations, the Roe solver may produce negative density if the jump is large enough. Assume that U_i^0 has been found. Then the flux splitting can be defined as

$$\begin{aligned} \mathscr{A}^+\Delta U_i &= f(U_i) - f(U_i^0) \\ \mathscr{A}^-\Delta U_i &= f(U_i^0) - f(U_{i-1}) \end{aligned}$$

The waves and the associated velocities needed for the correction waves also have to be specified. The states in the exact Riemann solution naturally define the jumps $\mathscr{W}_i^p$. If the original waves are discontinuities, the velocities should be chosen according to the Rankine-Hugoniot condition. If the waves are rarefaction waves, an average velocity could be used.

A third approach is evident; a hybrid version in which the exact solver defines the flux splitting, and the Roe solver defines the waves and velocities. If needed, the entropy fix is added in the usual way.

In the numerical examples in Section 6, only the Roe solver is used. The exact solver or the hybrid solver is not tested. Due to the lack of experience using these approaches, we will only indicate how they may be implemented in the multi-dimensional case.

The advantage of using (18) is that the flux function does not have to be evaluated when the Roe solver is used. But more important is that since the flux function is not needed, the algorithm works equally well on hyperbolic equations in quasi-linear form, i.e.

$$u_t + A(u, x, t)u_x = 0.$$

The "flux splitting" (which is called "fluctuation splitting" more generally in [16]), the waves, and the velocities are easily defined. For a discussion on how the wave propagation algorithm is applied to non-conservative problems, see [16].

2.4 Flux limiters

In the linear case, it is reasonable to compare wave strengths α_i^p from neighboring interfaces, since the eigenvectors r^p are constants. In the nonlinear case, the waves $\mathscr{W}_{i-1}^p$, $\mathscr{W}_i^p$, and $\mathscr{W}_{i+1}^p$ are not in general parallel, viewed as vectors in $\mathbb{R}^m$. In `CLAWPACK`, the default limiting is based on a projection of $\mathscr{W}_{i\pm1}^p$ onto $\mathscr{W}_i^p$. The lengths of these projections are compared to the length of $\mathscr{W}_i^p$. The limiting is performed according to

$$\widetilde{\mathscr{W}}_i^p = \Phi(\theta_i^p)\mathscr{W}_i^p,$$

$$\theta_i^p = \begin{cases} (\mathscr{W}_{i-1}^p, \mathscr{W}_i^p)/(\mathscr{W}_i^p, \mathscr{W}_i^p), & \lambda_i^p > 0 \\ \\ (\mathscr{W}_{i+1}^p, \mathscr{W}_i^p)/(\mathscr{W}_i^p, \mathscr{W}_i^p), & \lambda_i^p < 0 \end{cases}$$

where $(\cdot,\cdot)$ represents the inner product in $\mathbb{R}^m$. Note that this limiting generalizes the limiting used on linear systems, i.e. (16) and (17).

3 Two space dimensions

In two space dimensions the general conservation law reads

$$u_t + f(u)_x + g(u)_y = 0, \qquad u \in \mathbb{R}^m. \tag{20}$$

This equation is hyperbolic if the matrix $\alpha f'(u) + \beta g'(u)$ has real eigenvalues and a complete set of eigenvectors for any real α, β with $|\alpha| + |\beta| > 0$. Define a Cartesian grid with fixed spacing Δx and Δy. Let C_{ij} denote the cell $[x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$, where $y_{j-1/2} = (j-1/2)\Delta y$. A conservative scheme for (20) reads,

$$\bar{U}_{ij} = U_{ij} - \frac{\Delta t}{\Delta x}(F_{i+1,j} - F_{ij}) - \frac{\Delta t}{\Delta y}(G_{i,j+1} - G_{ij}), \tag{21}$$

where U_{ij} approximates the average of u at C_{ij} at time t_n. F_{ij} approximates the flux average at the interface between $C_{i-1,j}$ and C_{ij} in the time interval $[t_n, t_{n+1})$, i.e.

$$F_{ij} \approx \frac{1}{\Delta t \Delta y} \int_{t_n}^{t_{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} f(u(x_{i-1/2}, y, t))\, dydt. \tag{22}$$

Similarly, G_{ij} is the numerical flux at the interface between cells $C_{i,j-1}$ and C_{ij}, i.e.

$$G_{ij} \approx \frac{1}{\Delta t \Delta x} \int_{t_n}^{t_{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} g(u(x, y_{j-1/2}, t))\, dxdt. \tag{23}$$

As for the one-dimensional case, the numerical scheme is implemented in a different form than (21),

$$\bar{U}_{ij} = U_{ij} + \Delta_{ij}^{up} - \frac{\Delta t}{\Delta x}(\widetilde{F}_{i+1,j} - \widetilde{F}_{ij}) - \frac{\Delta t}{\Delta y}(\widetilde{G}_{i,j+1} - \widetilde{G}_{ij}). \tag{24}$$

The upwind part is

$$\Delta_{ij}^{up} = -\frac{\Delta t}{\Delta x}(\mathscr{A}^+ \Delta_x U_{ij} + \mathscr{A}^- \Delta_x U_{i+1,j}) - \frac{\Delta t}{\Delta y}(\mathscr{B}^+ \Delta_y U_{ij} + \mathscr{B}^- \Delta_y U_{i,j+1}).$$

Here, $\Delta_x U_{ij} = U_{ij} - U_{i-1,j}$ and $\Delta_y U_{ij} = U_{ij} - U_{i,j-1}$. The terms $\mathscr{B}^{\pm} \Delta_x U_{ij}$ define flux difference splitting in the y-direction, i.e. $g(U_{ij}) - g(U_{i,j-1}) = \mathscr{B}^+ \Delta_y U_{ij} + \mathscr{B}^- \Delta_y U_{ij}$. $\widetilde{F}_{ij}$ and $\widetilde{G}_{ij}$ include second order terms and cross-derivative terms arising when modeling multi-dimensional wave propagation.

We only consider the waves emanating from the Riemann problem at the interface between the cells $C_{i-1,j}$ and C_{ij}, and how they update nearby fluxes. The expressions resulting from interfaces in the y-direction are found by proper switching between $\widetilde{G}_{ij}$ and $\widetilde{F}_{ij}$, Δx and Δy, etc.

3.1 Linear systems

Consider the linear constant coefficient problem

$$u_t + Au_x + Bu_y = 0, \tag{25}$$

where A and B are $m \times m$ matrices. Since the problem is assumed to be hyperbolic, both A and B are diagonalizable. Hence, we can define $B^{\pm}$ as we did for $A^{\pm}$ in (7). Let μ^p and w^p denote the eigenvalues and eigenvectors of B, respectively. Then define

$$B^{\pm} = WM^{\pm}W^{-1}, \tag{26}$$

where W is the eigenvector matrix, and M the diagonal eigenvalue matrix.

Before proceeding with the two-dimensional wave propagation method, we expand $u(x, y, t+\Delta t)$ in a Taylor series in order to derive the terms necessary for obtaining second order accuracy.

In general we have that,

$$U(x, y, t+\Delta t) = U(x, y, t) + \Delta t\, U_t(x, y, t) + \frac{1}{2}\Delta t^2 U_{tt}(x, y, t) + \mathscr{O}(\Delta t^3). \tag{27}$$

From (25) we get

$$\begin{aligned} u_t &= -Au_x - Bu_y \\ u_{tt} &= A^2 u_{xx} + ABu_{yx} + BAu_{xy} + B^2 u_{yy}. \end{aligned}$$

Inserting this into (27) yields

$$u(x, y, t+\Delta t) = u - \Delta t(Au_x + Bu_y) + \frac{1}{2}\Delta t^2(A^2 u_{xx} + BAu_{xy} + ABu_{yx} + B^2 u_{yy}) + \mathscr{O}(\Delta t^3), \tag{28}$$

where the right hand side is evaluated at (x, y, t).

The first step in the algorithm is to solve the one dimensional Riemann problem normal to the interface, i.e. find waves $\mathscr{W}_{ij}^p$ so that $U_{ij} - U_{i-1,j} = \sum_p \mathscr{W}_{ij}^p$. This is exactly the same approach as used in the one-dimensional case. For example,

$$\mathscr{A}^- \Delta_x U_{ij} = A^- \Delta_x U_{ij} = \sum (\lambda^p)^- \mathscr{W}_{ij}^p, \qquad \mathscr{A}^+ \Delta_x U_{ij} = A^+ \Delta_x U_{ij} = \sum (\lambda^p)^+ \mathscr{W}_{ij}^p. \tag{29}$$

Doing the same in the y-direction yields a scheme in which waves only propagate normal to the cell interfaces.

For the scalar equation $u_t + au_x + bu_y = 0$, the stability limit is easily found to be $|a|\frac{\Delta t}{\Delta x} + |b|\frac{\Delta t}{\Delta y} \leq 1$. If the increment wave is propagated in the direction of the velocity vector (a, b), in the sense depicted in Figure 1a, the resulting flux updates will approximate the cross-derivative terms. This will increase both accuracy and stability. When the velocities are positive, the wave should move into cell $C_{i,j+1}$ causing an update of the $\widetilde{G}_{i,j+1}$ flux. Applying the definition of the G-flux in (23) to the increment wave, this update reads

$$\begin{aligned} \widetilde{G}_{i,j+1} &:= \widetilde{G}_{i,j+1} - \frac{1}{\Delta t}\frac{1}{\Delta x}\int_{t_n}^{t_{n+1}}\int_{x_{i-1/2}}^{x_{i-1/2}+a(t-t_n)} b\Delta_x U_{ij}\, dx dt, \\ &= \widetilde{G}_{i,j+1} - \frac{1}{2}\frac{\Delta t}{\Delta x} ab\Delta_x U_{ij}. \end{aligned} \tag{30}$$

Another way of deriving this expression is to consider the area in $C_{i,j+1}$ covered by the wave, which equals $\frac{1}{2}ab\Delta t^2$. Since the wave carries the constant jump $-\Delta_x U_{ij}$, the appropriate flux update equals (30).

The flux in the y-direction contains a difference in the x-direction. Hence, this update will account for the $\frac{1}{2}\Delta t^2 BAu_{xy}$ term in (28). The increment wave emanating from the interface between the cells $C_{i-1,j-1}$ and $C_{i-1,j}$ will update the flux $\widetilde{F}_{ij}$ by,

$$\widetilde{F}_{ij} := \widetilde{F}_{ij} - \frac{1}{2}\frac{\Delta t}{\Delta y} ab\Delta_y U_{i-1,j}. \tag{31}$$

Consider the scheme obtained where the cell-wise constant solution defined at the grid at time t_n is advected for a time step Δt. Next, the obtained function is averaged back onto the grid, cf.

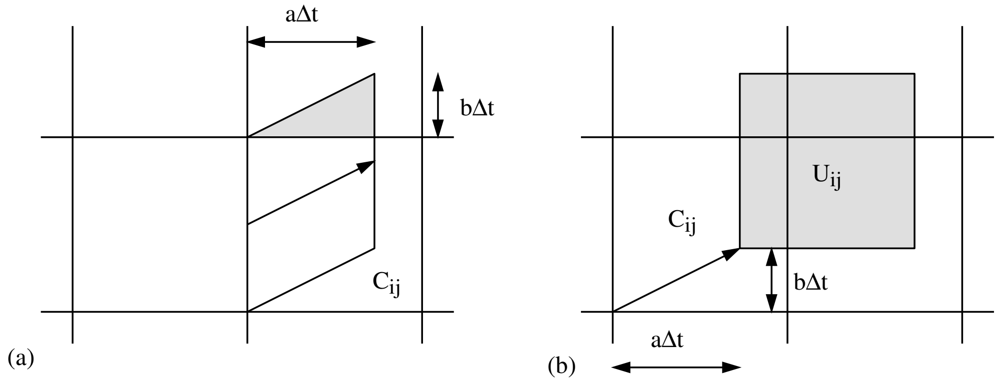


Figure 1: (a) Transverse propagation of the increment wave. (b) The flux updates define a scheme which equals the “shift and averaging” scheme, as long as the one-dimensional Courant numbers are less than one.

Figure 1. It is easily seen that the method defined by the fluxes (29), (30), and (31) equals this “shift and average” scheme provided

$$\max\left(|a|\frac{\Delta t}{\Delta x}, |b|\frac{\Delta t}{\Delta y}\right) \leq 1.$$

Hence, this is the stability limit for the wave propagation scheme.

In the system case, every increment wave is split into waves moving in the transverse direction, i.e.

$$\mathscr{W}_{ij}^p = \sum_q \beta_{ij}^{pq} w^q. \tag{32}$$

Each $\mathscr{W}_{ij}^p$ will update any of the surrounding $\widetilde{G}$ fluxes depending on the sign of λ^p. For example, if λ^p and μ^q are both positive, then

$$\widetilde{G}_{i,j+1} := \widetilde{G}_{i,j+1} - \frac{1}{2}\frac{\Delta t}{\Delta x}\lambda^p \mu^q \beta_{ij}^{pq} w^q,$$

similar to (30). The sum of all contributions to this flux gives

$$\widetilde{G}_{i,j+1} := \widetilde{G}_{i,j+1} - \frac{1}{2}\frac{\Delta t}{\Delta x}\sum_q \sum_p (\lambda^p)^+ (\mu^q)^+ \beta_{ij}^{pq} w^q. \tag{33}$$

It is easily seen that this is equal to

$$\widetilde{G}_{i,j+1} := \widetilde{G}_{i,j+1} - \frac{1}{2}\frac{\Delta t}{\Delta x} B^+ A^+ \Delta_x U_{ij}. \tag{34}$$

The proper updates for the rest of the $\widetilde{G}$-fluxes are

$$\begin{aligned} \widetilde{G}_{i-1,j} &:= \widetilde{G}_{i-1,j} - \frac{1}{2}\frac{\Delta t}{\Delta x} B^- A^- \Delta_x U_{ij} \\ \widetilde{G}_{ij} &:= \widetilde{G}_{ij} - \frac{1}{2}\frac{\Delta t}{\Delta x} B^- A^+ \Delta_x U_{ij} \\ \widetilde{G}_{i-1,j+1} &:= \widetilde{G}_{i-1,j+1} - \frac{1}{2}\frac{\Delta t}{\Delta x} B^+ A^- \Delta_x U_{ij}, \end{aligned} \tag{35}$$

These four possibilities are depicted in Figure 2. For the $\widetilde{F}$ flux there are similar updates. Note that these updates generalize (30). In the scalar case, only one of the updates in (34), (35) is nonzero.

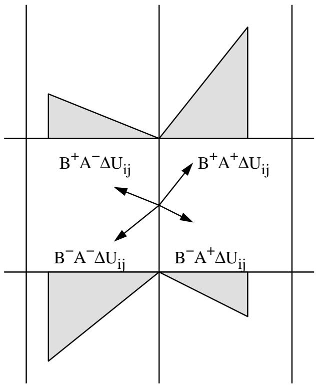


Figure 2: The four transverse flux differences for a linear system.

In an implementation, (33) would require the solution of m Riemann problems, in addition to the one necessary for obtaining $\mathscr{W}^p$. In general, this involves too much work. It is also in general too expensive to derive the $A^{\pm}$ and $B^{\pm}$ matrices explicitly, and then perform matrix multiplications. Instead the left-going flux difference $A^-\Delta_x U_{ij}$ and the right-going flux difference $A^+\Delta_x U_{ij}$ are split into transverse moving waves. This *transverse flux difference splitting* is defined by

$$A^-\Delta_x U_{ij} = \sum_q (\delta_{ij}^q)^- w^q, \qquad A^+\Delta_x U_{ij} = \sum_q (\delta_{ij}^q)^+ w^q. \tag{36}$$

The computations of the flux updates in (34) and (35) are significantly simplified, for example

$$B^+A^+\Delta_x U_{ij} = \sum_q (\mu^q)^+ (\delta_{ij}^q)^+ w^q, \tag{37}$$

with similar expressions for the rest of the updates. The number of Riemann problems per interface is now three, regardless of the size of the system.

The second order correction wave is applied as in one dimension, i.e.

$$\widetilde{F}_{ij} := \widetilde{F}_{ij} + \frac{1}{2}\sum_p |\lambda^p| \left(1 - \frac{\Delta t}{\Delta x}|\lambda^p|\right) \widetilde{\mathscr{W}}_{ij}^p. \tag{38}$$

A potential weakness of the method is that the limiter only takes into account waves propagating normal to the interfaces. At least for scalar problems there exist ways of doing multi-dimensional limiting, but they are more expensive in use [12], [24], and [36]. However, numerical experiments show that this simple one dimensional limiting is able to control oscillations in a satisfactory way.

In the two dimensional Lax-Wendroff scheme, the u_{xy} term is approximated using the centered difference

$$u_{xy} \approx \frac{1}{4\Delta x \Delta y}((U_{i+1,j+1} - U_{i-1,j+1}) - (U_{i+1,j-1} - U_{i-1,j-1})).$$

The resulting approximation of the term $\frac{1}{2}\Delta t^2 BAu_{xy}$ may be written in a flux formulation as

$$\widetilde{G}_{ij} := \widetilde{G}_{ij} + \frac{1}{8}\frac{\Delta t}{\Delta x} BA(\Delta_x U_{ij} + \Delta_x U_{i,j-1} + \Delta_x U_{i+1,j-1} + \Delta_x U_{i+1,j}).$$

The $\widetilde{G}_{ij}$ flux is updated by the average contribution from the four surrounding increment waves in the x-direction, regardless of the sign. In comparison, the updates given in (34), (35) results in the flux contribution

$$\widetilde{G}_{ij} := \widetilde{G}_{ij} + \frac{1}{2}\frac{\Delta t}{\Delta x}(B^-A^+\Delta_x U_{ij} + B^+A^+\Delta_x U_{i,j-1} + B^+A^-\Delta_x U_{i+1,j-1} + B^-A^-\Delta_x U_{i+1,j}).$$

Hence, the flux is only updated by the transverse propagating waves moving towards the interface. Note that since

$$B^+A^+ + B^-A^+ + B^+A^- + B^-A^- = BA \tag{39}$$

the flux updates above will also result in an approximation to the $\frac{1}{2}\Delta t^2 BAu_{xy}$ term in the Taylor expansion. This upwinding increases the stability compared to the Lax-Wendroff scheme. (For some results on the stability of second order schemes applied to (25) cf. [11].) The stability for the wave propagation scheme is

$$\max_{p,q}\left(|\lambda_p|\frac{\Delta t}{\Delta x}, |\mu_q|\frac{\Delta t}{\Delta y}\right) \le 1. \tag{40}$$

Surprisingly, in three dimensions, the use of upwinding instead of centered differences for cross-derivatives like BAu_{xy} does not increase the stability. In fact, the stability in the Lax-Wendroff scheme is destroyed. We will return to this topic in Section 5.

3.2 Transverse propagation of the correction wave

In some cases it is also advantageous to propagate the correction wave in the transverse direction. First consider the scalar equation $u_t + au_x + bu_y = 0$, with positive velocities. The natural approach is to move the correction wave in the same way as the increment wave (30). Let $c(x,t) = s(x - x_{i-1} - a(t - t_n))$ denote the correction wave associated cell $C_{i-1,j}$, where s equals the unlimited slope $\Delta_x U_{ij}/\Delta x$. Then we have that

$$\begin{aligned} \widetilde{G}_{i,j+1} &:= \widetilde{G}_{i,j+1} - \frac{1}{\Delta t}\frac{1}{\Delta x}\int_{t_n}^{t_{n+1}}\int_{x_{i-1/2}}^{x_{i-1/2}+a(t-t_n)} b\, c(x,t)\, dxdt, \\ &= \widetilde{G}_{i,j+1} - \frac{\Delta t}{\Delta x}ab\left(\frac{1}{4} - \frac{1}{6}\frac{\Delta t}{\Delta x}a\right)\Delta_x U_{ij}. \end{aligned} \tag{41}$$

For reasons to be made clear below, an update using other coefficients than those in (41) will be used. The proper updates will be $\widetilde{G}_{i,j+1} := \widetilde{G}_{i,j+1} + \frac{1}{2}ab\frac{\Delta t}{\Delta x}(1 - a\frac{\Delta t}{\Delta x})\Delta_x U_{ij}$. The geometrical interpretation of this update is that instead of propagating the wave in both directions simultaneously as in (41), the wave is first moved a distance $a\Delta t$ in the x-direction, and afterwards a distance $b\Delta t$ in the y-direction. The motion of this wave is shown in Figure 3. Note that a part of the wave will update the $\widetilde{G}$-flux in the downstream direction. The proper update is $\widetilde{G}_{i-1,j+1} := \widetilde{G}_{i-1,j+1} - \frac{1}{2}ab\frac{\Delta t}{\Delta x}(1 - a\frac{\Delta t}{\Delta x})\Delta_x U_{ij}$. Note that these updates equals b times the update this wave makes on the $\widetilde{F}_{ij}$ flux.

For systems, the correction waves are split into waves moving in the y-direction. Define

$$\mathscr{S} = \sum_p |\lambda^p|\left(1 - \frac{\Delta t}{\Delta x}|\lambda^p|\right)\widetilde{\mathscr{W}}_{ij}^p, \tag{42}$$

and split this correction term into an up-going and a down-going part, i.e. solve

$$\mathscr{S} = \sum_q \varepsilon_{ij}^q w^q. \tag{43}$$

The wave $\varepsilon_{ij}^q w^q$ will update two of the surrounding $\widetilde{G}$ fluxes depending on the sign of μ^q. For example,

$$\widetilde{G}_{i,j+1} = \widetilde{G}_{i,j+1} + \frac{1}{2}\frac{\Delta t}{\Delta x}\sum_q (\mu^q)^+\varepsilon_{ij}^q w^q, \tag{44}$$

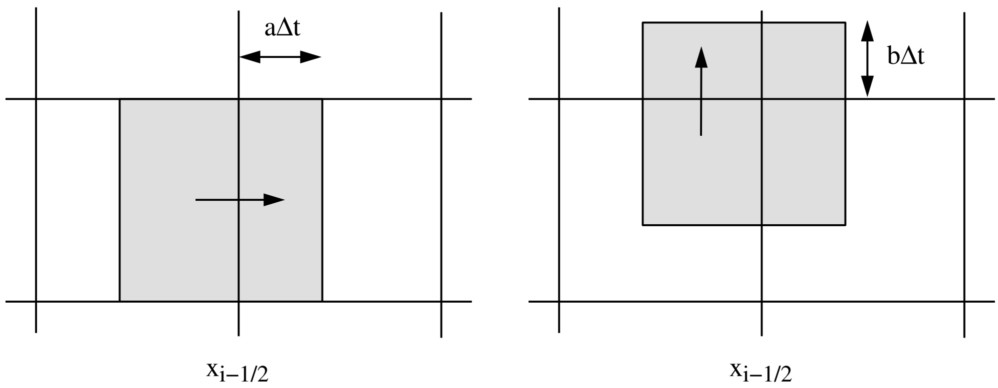


Figure 3: The transverse propagation of the correction wave when the velocities are positive. The wave is first shifted a distance $a\Delta t$ in the x-direction, then a distance $b\Delta t$ in the y-direction.

$$\widetilde{G}_{i-1,j+1} = \widetilde{G}_{i-1,j+1} - \frac{1}{2}\frac{\Delta t}{\Delta x}\sum_q (\mu^q)^+ \varepsilon^q_{ij} w^q, \tag{45}$$

as seen in Figure 3. Down-going waves would instead affect $\widetilde{G}_{i,j}$ and $\widetilde{G}_{i-1,j}$. In matrix notation, the complete updates are

$$\begin{aligned}
\widetilde{G}_{i,j+1} &:= \widetilde{G}_{i,j+1} + \frac{1}{2}\frac{\Delta t}{\Delta x}B^+|A|\left(1-|A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ij} \\
\widetilde{G}_{ij} &:= \widetilde{G}_{ij} + \frac{1}{2}\frac{\Delta t}{\Delta x}B^-|A|\left(1-|A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ij} \\
\widetilde{G}_{i-1,j+1} &:= \widetilde{G}_{i-1,j+1} - \frac{1}{2}\frac{\Delta t}{\Delta x}B^+|A|\left(1-|A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ij} \\
\widetilde{G}_{i-1,j} &:= \widetilde{G}_{i-1,j} - \frac{1}{2}\frac{\Delta t}{\Delta x}B^-|A|\left(1-|A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ij}.
\end{aligned} \tag{46}$$

It is possible to avoid the computation of the eigenvector expansion (or Riemann problem) in (43), by including $\mathscr{S}$ in the flux differences (29). The proper modifications are

$$\begin{aligned}
A^+\Delta_x U_{ij} &:= A^+\Delta_x U_{ij} - \mathscr{S} \\
A^-\Delta_x U_{ij} &:= A^-\Delta_x U_{ij} + \mathscr{S}
\end{aligned} \tag{47}$$

Since $\frac{1}{2}\mathscr{S}$ already is computed in (38), including this transverse propagation requires virtually no extra work.

In two dimensions, the inclusion of the transverse propagation of the correction wave neither increases the order nor the stability limit of the method, but the accuracy may be improved. However, in three dimensions these waves are essential for stability, cf. Section 5.

Assume that we want to construct a third order accurate scheme. Then we have to add appropriate terms to the Taylor expansion (28), which are

$$\begin{aligned}
-\frac{1}{6}\Delta t^3 (&A^3 u_{xxx} + A^2 B u_{yxx} + ABA u_{xyx} + AB^2 u_{yyx} \\
&+ BA^2 u_{xxy} + BAB u_{yxy} + B^2 A u_{xyy} + B^3 u_{yyy}).
\end{aligned} \tag{48}$$

We need to approximate these terms, and also those arising in the truncation error in the approximations of the first and second order terms already derived.

For example, the truncation error in the approximation of $\frac{1}{2}\Delta t^2 BAu_{xy}$ (35) is dominated by

$$-\frac{1}{4}\Delta t^2\Delta x(B^+A^+ - B^+A^- + B^-A^+ - B^-A^-)u_{xxy} = -\frac{1}{4}\Delta t^2\Delta x B|A|u_{xxy}. \tag{49}$$

Then approximating

$$\Delta t^2\Delta x B|A|\left(\frac{1}{4} - \frac{1}{6}\frac{\Delta t}{\Delta x}|A|\right)u_{xxy}, \tag{50}$$

will cancel the terms in (49) and add the u_{xxy} term from (48). Deriving similar expressions for the rest of the first and second order terms yields,

$$\begin{aligned}\frac{1}{6}\Delta t^3 ABAu_{xyx} + \Delta t^2\Delta x B|A|\left(\frac{1}{4} - \frac{1}{6}\frac{\Delta t}{\Delta x}|A|\right)u_{xxy} + \Delta t^2\Delta x\left(\frac{1}{4} - \frac{1}{6}\frac{\Delta t}{\Delta x}|A|\right)|A|Bu_{yxx} - \\ \frac{1}{6}\Delta t^3 BABu_{yxy} + \Delta t^2\Delta x A|B|\left(\frac{1}{4} - \frac{1}{6}\frac{\Delta t}{\Delta x}|B|\right)u_{yyx} + \Delta t^2\Delta x\left(\frac{1}{4} - \frac{1}{6}\frac{\Delta t}{\Delta x}|B|\right)|B|Au_{xyy} + \\ \frac{1}{6}\Delta t^3 A^3u_{xxx} + \frac{1}{6}\Delta t^3 B^3 u_{yyy}\end{aligned} \tag{51}$$

We need to approximate this expression in order to achieve a third order scheme.

Assume that A and B commute, i.e. that $AB = BA$. Then the expression above, with u_{xxx} and u_{yyy} omitted reduces to

$$\frac{1}{2}\Delta t^2\Delta x B|A|\left(1 - \frac{\Delta t}{\Delta x}|A|\right)u_{xxy} \quad + \quad \frac{1}{2}\Delta t^2\Delta y A|B|\left(1 - \frac{\Delta t}{\Delta y}|B|\right)u_{yyx}. \tag{52}$$

The transverse propagation of the correction wave as accomplished in (46), yields the desired approximation of the first term in (52). The second term is approximated by the similar transverse propagation of the correction wave emanating from interfaces in the y-direction.

For scalar advection problems with constant coefficients, a third order accurate scheme is easily obtained. When transverse propagation of the correction waves is included, only the terms u_{xxx} and u_{yyy} need to be approximated, cf. [15].

3.3 Nonlinear systems

In order to solve nonlinear systems, one has to specify the flux difference splitting $\mathscr{A}^{\pm}\Delta_x U_{ij}$, the waves $\mathscr{W}^p_{ij}$, and the associated velocities λ^p_{ij}. In addition, the transverse flux splitting has to be defined. Motivated by the linear case, let the symbols $\mathscr{B}^{\pm}\mathscr{A}^{\pm}\Delta_x U_{ij}$ denote the transverse flux differences in the nonlinear case. The propagation of the correction wave also has to be defined. There are two different approaches for obtaining the quantities needed, depending on whether exact or approximative Riemann solvers are applied.

If a Roe solver is used, we proceed essentially as in the linear case. The matrix A_{ij} satisfying $A_{ij}(U_{ij} - U_{i-1,j}) = f(U_{ij}) - f(U_{i-1,j})$ is constructed. This matrix defines $\mathscr{W}^p_{ij}$ and λ^p_{ij}. The flux difference splitting is defined as

$$\mathscr{A}^{\pm}\Delta_x U_{ij} = A^{\pm}_{ij}\Delta_x U_{ij},$$

where $A^{\pm}_{ij}$ is defined as in (7).

The transverse flux difference splitting is implemented using the matrix B_{ij}. This matrix is simply taken as the Jacobian $g'(u)$ evaluated at a state "close" to $U_{i-1,j}$ and U_{ij}. A good choice is to use the Roe-averaged state U^*_{ij} already computed in the construction of A_{ij}. The matrices $B^{\pm}_{ij}$ are defined using (26). The transverse flux splitting is then defined in the obvious way, for example

$$\mathscr{B}^{+}\mathscr{A}^{+}\Delta_x U_{ij} = B^{+}_{ij}A^{+}_{ij}\Delta_x U_{ij}. \tag{53}$$

Each product between a matrix and a vector corresponds to a Riemann problem. For Riemann problems in the normal direction, an entropy fix should be applied as in the 1D case. However, computations indicate that there is no need for this correction in the transverse direction. Hence, the Riemann problems used in computing these flux updates require less work than the Riemann problems in the normal direction. No entropy fix is needed, and the state used in evaluating the Jacobian is already computed.

As discussed in Section 2.3, there are situations where it is preferable to use an exact Riemann solver. They are in general more accurate and more robust than the approximate solvers. The drawback is that they are more computationally expensive.

For the flux splitting in the normal direction, $\mathscr{A}^{\pm}\Delta_x U_{ij}$ is defined exactly as in 1D, i.e.

$$\mathscr{A}^{+}\Delta_x U_{ij} = f(U_{ij}) - f(U_{ij}^{0}), \qquad \mathscr{A}^{-}\Delta_x U_{ij} = f(U_{ij}^{0}) - f(U_{i-1,j}),$$

where $U_{ij}^0 = RP_x(U_{i-1,j}, U_{ij})$.

It is not evident how to include exact solvers in the transverse direction. An approach blending an exact and an approximate solver is to apply the matrices $B_{ij}^{\pm}$ derived above, i.e.

$$\mathscr{B}^{+}\mathscr{A}^{+}\Delta_x U_{ij} = B_{ij}^{+}\left(f(U_{ij}) - f(U_{ij}^{0})\right).$$

Note that this expression generalizes the transverse flux differences for linear systems. It is straight forward to include the transverse propagation of the correction wave. This may be obtained as for linear systems by modifying the transverse propagation of the increment wave (47).

In Algorithm 1 on page 16, one time step of the two-dimensional wave propagation scheme is written. The approach using the Roe solver for computing the transverse fluxes is given. The number of Riemann problems to be solved for each interface is three, which is comparable to the work required in the method of Colella [4]. (See [16] for more discussion of this.)

4 Three space dimensions

In three dimensions consider the conservation law

$$u_t + f(u)_x + g(u)_y + h(u)_z = 0.$$

This equation is hyperbolic if the matrix $\alpha f'(u) + \beta g'(u) + \gamma h'(u)$ has real eigenvalues and a complete set of eigenvectors for any real α, β, and γ, with $|\alpha| + |\beta| + |\gamma| > 0$. Define a regular grid with constant spacing Δx, Δy and Δz. Let C_{ijk} denote the cell $[x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}] \times [z_{k-1/2}, z_{k+1/2}]$, where $z_{k-1/2} = (k-1/2)\Delta z$. The conservation law will be approximated by the scheme

$$\bar{U}_{ijk} = U_{ijk} + \Delta_{ijk}^{up} - \frac{\Delta t}{\Delta x}(\widetilde{F}_{i+1,jk} - \widetilde{F}_{ijk}) - \frac{\Delta t}{\Delta y}(\widetilde{G}_{i,j+1,k} - \widetilde{G}_{ijk}) - \frac{\Delta t}{\Delta z}(\widetilde{H}_{ij,k+1} - \widetilde{H}_{ijk}).$$

Now the first order upwind part reads

$$\begin{aligned} \Delta_{ijk}^{up} &= -\frac{\Delta t}{\Delta x}(\mathscr{A}^{+}\Delta_x U_{ijk} + \mathscr{A}^{-}\Delta_x U_{i+1,jk}) - \frac{\Delta t}{\Delta y}(\mathscr{B}^{+}\Delta_y U_{ijk} + \mathscr{B}^{-}\Delta_y U_{i,j+1,k}) \\ &- \frac{\Delta t}{\Delta z}(\mathscr{C}^{+}\Delta_z U_{ijk} + \mathscr{C}^{-}\Delta_z U_{ij,k+1}), \end{aligned} \tag{54}$$

where $\Delta_z U_{ijk} = U_{ijk} - U_{ij,k-1}$. The terms $\mathscr{C}^{\pm}\Delta_z U_{ijk}$ denote the flux difference splitting in the z-direction, i.e. $h(U_{ijk}) - h(U_{ij,k-1}) = \mathscr{C}^{+}\Delta_z U_{ijk} + \mathscr{C}^{-}\Delta_z U_{ijk}$. The $\widetilde{F}$, $\widetilde{G}$, and $\widetilde{H}$ terms

Algorithm 1: One time step of the algorithm for a general two-dimensional problem where a Roe solver is used in the transverse direction.

#initialize fluxes:

for each i, j do

$\quad \widetilde{F}_{ij} := 0, \quad \widetilde{G}_{ij} := 0, \quad \Delta_{ij}^{up} := 0$

update fluxes based on interfaces in the x-direction:

for each interface $x = x_{i-1/2}$ do

$\quad$ <Compute flux difference splitting $\mathscr{A}^{\pm}\Delta_x U_{ij}$, wave $\mathscr{W}^p$ and velocities λ^p>

$$\Delta_{ij}^{up} := \Delta_{ij}^{up} - \frac{\Delta t}{\Delta x}\mathscr{A}^{+}\Delta_x U_{ij}, \qquad \Delta_{i-1,j}^{up} := \Delta_{i-1,j}^{up} - \frac{\Delta t}{\Delta x}\mathscr{A}^{-}\Delta_x U_{ij}$$

$\quad$ <Apply limiter to $\mathscr{W}^p$ to obtain $\widetilde{\mathscr{W}}^p$>

$$\mathscr{S} := \sum_p |\lambda^p| \left(1 - \frac{\Delta t}{\Delta x}|\lambda^p|\right) \widetilde{\mathscr{W}}^p$$

$$\widetilde{F}_{ij} := \widetilde{F}_{ij} + \frac{1}{2}\mathscr{S}$$

$\quad$ if transverse propagation of the correction wave then

$$\mathscr{A}^{+}\Delta_x U_{ij} := \mathscr{A}^{+}\Delta_x U_{ij} - \mathscr{S}, \quad \mathscr{A}^{-}\Delta_x U_{ij} := \mathscr{A}^{-}\Delta_x U_{ij} + \mathscr{S}$$

$\quad$ endif

$\quad$ # Split $\mathscr{A}^{+}\Delta_x U_{ij}$ and $\mathscr{A}^{-}\Delta_x U_{ij}$ into eigenvectors of $B \approx g'(u)$:

$$\mathscr{A}^{+}\Delta_x U_{ij} = \sum_q (\delta^q)^+ w^q, \quad \mathscr{A}^{-}\Delta_x U_{ij} = \sum_q (\delta^q)^- w^q$$

$\quad$ # Form the terms

$$\mathscr{B}^{\pm}\mathscr{A}^{*}\Delta_x U_{ij} := \sum_q (\mu^q)^{\pm}(\delta^q)^* w^q, \qquad \text{where } * \text{ equals } + \text{ or } -.$$

$\quad$ # Use these terms to update neighboring $\widetilde{G}$ fluxes:

$$\widetilde{G}_{i-1,j} := \widetilde{G}_{i-1,j} - \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{B}^{-}\mathscr{A}^{-}\Delta_x U_{ij}$$

$$\widetilde{G}_{ij} := \widetilde{G}_{ij} - \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{B}^{-}\mathscr{A}^{+}\Delta_x U_{ij}$$

$$\widetilde{G}_{i,j+1} := \widetilde{G}_{i,j+1} - \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{B}^{+}\mathscr{A}^{+}\Delta_x U_{ij}$$

$$\widetilde{G}_{i-1,j+1} := \widetilde{G}_{i-1,j+1} - \frac{1}{2}\frac{\Delta t^{+}}{\mathscr{B}}\mathscr{A}^{-}\Delta_x U_{ij}$$

Update fluxes based on interfaces in the y-direction:

Similar to the above, but roles of i and j, Δx and Δy, $\widetilde{F}$ and $\widetilde{G}$ switched.

Finally, update U:

for each i, j do

$$\bar{U}_{ij} := U_{ij} + \Delta_{ij}^{up} - \frac{\Delta t}{\Delta x}(\widetilde{F}_{i+1,j} - \widetilde{F_{ij}}) - \frac{\Delta t}{\Delta y}(\widetilde{G}_{i,j+1} - \widetilde{G}_{ij})$$

are used for including second order updates, and for including approximations of the necessary cross-derivative terms.

The outline is the same as in 2D. In this description we only consider the waves emanating from the Riemann problem at the interface between cells $C_{i-1,jk}$ and C_{ijk}, and how these waves update the surrounding $\widetilde{G}$ and $\widetilde{H}$ fluxes. Analogous procedures are applied in the y- and z-directions.

4.1 Propagation of the increment wave for scalar advection

Before continuing with systems of equations, the scalar advection equation

$$u_t + au_x + bu_y + cu_z = 0, \tag{55}$$

is considered. This equation has been studied thoroughly in [15]. In this section, only the propagation of the increment wave is discussed, leaving the motion of the correction wave to the more general system case.

For simplicity, assume that the velocities are positive. Then the upwind part reads

$$\Delta_{ijk}^{up} = -\frac{\Delta t}{\Delta x} aU_{i-1,jk} - \frac{\Delta t}{\Delta y} bU_{i,j-1,k} - \frac{\Delta t}{\Delta z} cU_{ij,k-1}.$$

The stability condition of this scheme is $a\frac{\Delta t}{\Delta x} + b\frac{\Delta t}{\Delta y} + c\frac{\Delta t}{\Delta z} \leq 1$. In order to increase both stability and accuracy, the increment wave is propagated in the transverse direction. This is accomplished in much the same way as in the two-dimensional case. In fact, all updates derived in the previous section will be reused with obvious changes in the indices.

The updates made on neighboring cells may be obtained by considering the volume of the part of the wave entering the cells times the jump carried by the wave. In Figure 4a, the increment wave is depicted when the velocities are positive. In this case, the cells $C_{i,j+1,k}$, $C_{ij,k+1}$, $C_{i,j+1,k+1}$ are affected. The corresponding volumes covered by the wave are

$$\begin{aligned}
C_{i,j+1,k}: &\quad \frac{1}{2}ab\Delta t^2 \Delta z - \frac{1}{3}abc\Delta t^3 \\
C_{ij,k+1}: &\quad \frac{1}{2}ac\Delta t^2 \Delta z - \frac{1}{3}abc\Delta t^3 \\
C_{i,j+1,k+1}: &\quad \frac{1}{3}abc\Delta t^3
\end{aligned}$$

These volumes are shown in Figure 4b. The actual change in the solution equals these volumes times the jump $-\Delta_x U_{ijk}$, and are obtained by changing the fluxes. There are several ways of distributing the solution updates between the fluxes in the scalar case. Here, a different approach than in [15] will be used, since this will be in agreement with the appropriate terms in the system case.

When the velocities are positive, the flux updates are

$$\begin{aligned}
\widetilde{G}_{i,j+1,k} &:= \widetilde{G}_{i,j+1,k} - \frac{1}{2}ab\frac{\Delta t}{\Delta x}\Delta_x U_{ijk} + \frac{1}{6}abc\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta z}\Delta_x U_{ijk} \\
\widetilde{G}_{i,j+1,k+1} &:= \widetilde{G}_{i,j+1,k+1} - \frac{1}{6}abc\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta z}\Delta_x U_{ijk} \\
\widetilde{H}_{ij,k+1} &:= \widetilde{H}_{ij,k+1} - \frac{1}{2}ac\frac{\Delta t}{\Delta x}\Delta_x U_{ijk} + \frac{1}{6}abc\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y}\Delta_x U_{ijk} \\
\widetilde{H}_{i,j+1,k+1} &:= \widetilde{H}_{i,j+1,k+1} - \frac{1}{6}abc\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y}\Delta_x U_{ijk}.
\end{aligned} \tag{56}$$

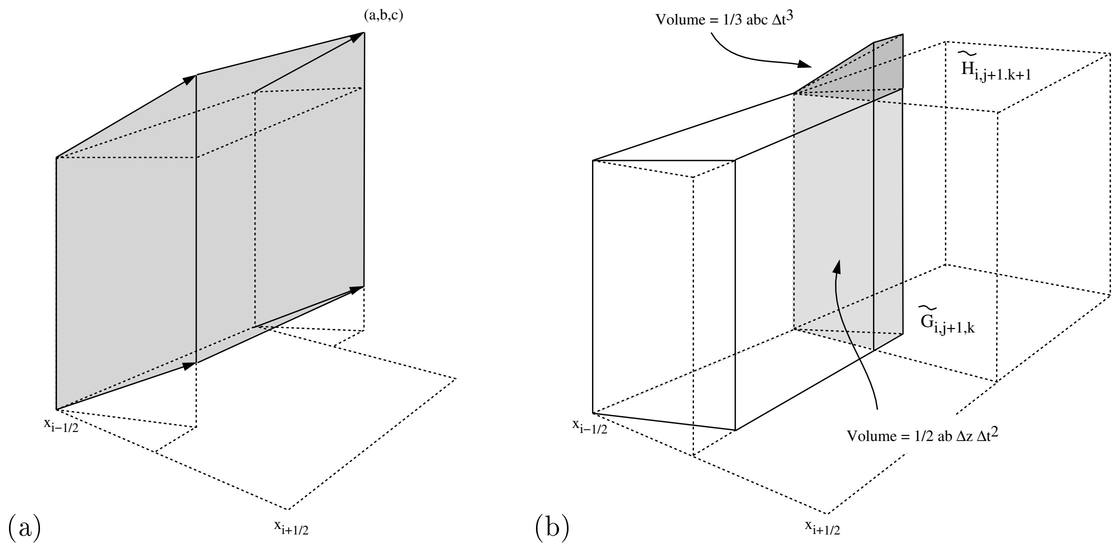


Figure 4: (a) The volume covered by the increment wave, when the velocities are positive. (b) The volume used in updating the cells $C_{ij,k+1}$ and $C_{i,j+1,k+1}$.

Next, consider the "shift and average" scheme in the three-dimensional case. The wave propagation scheme derived above equals this scheme provided

$$\max\left(a\frac{\Delta t}{\Delta x}, b\frac{\Delta t}{\Delta y}, c\frac{\Delta t}{\Delta z}\right) \leq 1,$$

i.e. as long as the waves do not move through more than one cell. Hence, the first order scheme described above has this stability limit.

Note that the update of $\widetilde{G}_{i,j+1,k}$ equals the update of $\widetilde{G}_{i,j+1}$ in (30) plus an extra term which has to do with the propagation in the third coordinate direction. The first term corresponds to the part of the wave moving into cells sharing an interface with either of the cells defining the Riemann problem, i.e. $C_{i-1,jk}$ and C_{ijk}. The last update is caused by the part of the wave moving into cells only sharing an edge with one of these cells.

It is advantageous to allow the correction wave to move in the transverse direction too. In fact, doing this, it is very easy to obtain a third order accurate scheme when applied to scalar advection, cf. [15].

4.2 Linear systems

Consider the linear hyperbolic problem,

$$u_t + Au_x + Bu_y + Cu_z = 0, \tag{57}$$

where A, B, and C are constant $m \times m$ matrices. C has eigenvalues ν^l and associated eigenvectors s^l. Let S denote the eigenvector matrix, and N the diagonal matrix with the eigenvalues as entries. Then define the positive and negative part of C by

$$C^{\pm} = SN^{\pm}S^{-1}.$$

Consider the Taylor series of $u(x, y, z, t + \Delta t)$ for (57) including terms up to third order in Δt. Due to the large number of terms, we only consider those which are differentiated last in x . The flux updates resulting from a single interface in x should lead to approximations of these terms, which are

$$\Delta t A u_x + \frac{1}{2}\Delta t^2 (A^2 u_{xx} + BA u_{xy} + CA u_{xz}) - \frac{1}{6}\Delta t^3 (A^3 u_{xxx} + ABA u_{xyx} + ACA u_{xzx} + BA^2 u_{xxy} + B^2 A u_{xyy} + BCA u_{xzy} + CA^2 u_{xxz} + CBA u_{xyz} + C^2 A u_{xzz}). \tag{58}$$

The wave propagation scheme to be derived will cover all these terms, except for the $A^3 u_{xxx}$ term. In addition, some fourth order terms will also be accounted for.

The algorithm starts as in 1D and 2D by considering one-dimensional Riemann problems at the interfaces. Hence, the jump $\Delta_x U_{ijk}$ is expanded in terms of the eigenvectors of A, yielding $\Delta_x U_{ijk} = \sum_p \mathscr{W}^p_{ijk}$. The upwind part (54) is defined in the obvious way, replacing $\mathscr{A}^\pm$ by $A^\pm$, $\mathscr{B}^\pm$ by $B^\pm$ and finally $\mathscr{C}^\pm$ by $C^\pm$. As noted in the previous section, the stability properties of the scheme involving only the upwind part, are rather poor. In order to increase the stability, $\widetilde{G}$ and $\widetilde{H}$ fluxes are updated in order to model multi-dimensional wave propagation.

The updates of the fluxes in the three-dimensional case may be viewed as consisting of an update as in 2D, augmented by a purely 3D correction.

First, let the increment wave move into cells sharing interfaces with either of the cells C_{ijk} or $C_{i-1,jk}$. By letting the waves move in the y-direction we get the same update as in 2D (35), with obvious changes in indices. This will account for the $\frac{1}{2}\Delta t^2 BA u_{xy}$ term in (58).

Next, consider the part of the increment wave moving in the z-direction. This results in an update of the $\widetilde{H}$ fluxes similar to (35),

$$\begin{aligned}
\widetilde{H}_{i-1,jk} &:= \widetilde{H}_{i-1,jk} - \frac{1}{2}\frac{\Delta t}{\Delta x} C^- A^- \Delta_x U_{ijk} \\
\widetilde{H}_{ijk} &:= \widetilde{H}_{ijk} - \frac{1}{2}\frac{\Delta t}{\Delta x} C^- A^+ \Delta_x U_{ijk} \\
\widetilde{H}_{ij,k+1} &:= \widetilde{H}_{ij,k+1} - \frac{1}{2}\frac{\Delta t}{\Delta x} C^+ A^+ \Delta_x U_{ijk} \\
\widetilde{H}_{i-1,j,k+1} &:= \widetilde{H}_{i-1,j,k+1} - \frac{1}{2}\frac{\Delta t}{\Delta x} C^+ A^- \Delta_x U_{ijk},
\end{aligned} \tag{59}$$

which account for $\frac{1}{2}\Delta t^2 CA u_{xz}$ in the Taylor expansion. Including these updates will increase the stability, but not to the best obtainable.

So far the updates have been exactly as in 2D. In order to get full 3D wave propagation, waves should also move into cells only sharing an edge with the cells defining the Riemann problem. This yields approximations of cross-derivatives like $\frac{1}{6}\Delta t^3 CBA u_{xyz}$. For this term, the proper updates are

$$\begin{aligned}
\widetilde{H}_{ij,k+1} &:= \widetilde{H}_{ij,k+1} + \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y} C^+ |B| A^+ \Delta_x U_{ijk} \\
\widetilde{H}_{ijk} &:= \widetilde{H}_{ijk} + \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y} C^- |B| A^+ \Delta_x U_{ijk} \\
\widetilde{H}_{i,j+1,k+1} &:= \widetilde{H}_{i+1,j,k+1} - \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y} C^+ B^+ A^+ \Delta_x U_{ijk} \\
\widetilde{H}_{i,j+1,k} &:= \widetilde{H}_{i,j+1,k} - \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y} C^- B^+ A^+ \Delta_x U_{ijk} \\
\widetilde{H}_{i,j-1,k+1} &:= \widetilde{H}_{i,j-1,k+1} + \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y} C^+ B^- A^+ \Delta_x U_{ijk} \\
\widetilde{H}_{i,j-1,k} &:= \widetilde{H}_{i,j-1,k} + \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y} C^- B^- A^+ \Delta_x U_{ijk}.
\end{aligned} \tag{60}$$

Six additional updates are also required. These are obtained by replacing i by $i-1$ in the fluxes above, and replacing $A^+\Delta_x U_{ijk}$ by $A^-\Delta_x U_{ijk}$. Note that the updates in (59) and (60) generalize the changes in $\widetilde{H}$ in (56). In order to approximate $BCAu_{xzy}$, similar updates apply to the $\widetilde{G}$ fluxes.

The $C^+B^+A^+\Delta_x U_{ijk}$ term in (60) may be interpreted as follows. Each wave $\mathscr{W}^p_{ijk}$ in the expansion of $\Delta_x U_{ijk}$ is split into waves moving in the y-direction, i.e. $\mathscr{W}^p = \sum_q \beta^{pq} w^q$. Then each $\beta^{pq} w^q$ is split into waves moving in the z-direction, i.e. $\beta^{pq} w^q = \sum_l \gamma^{pql} s^l$. It is easily seen that

$$C^+B^+A^+\Delta_x U_{ijk} = \sum_l \sum_q \sum_p (\lambda^p)^+ \, (\mu^q)^+ \, (\nu^l)^+ \, \gamma^{pql} s^l. \tag{61}$$

Computing these updates either using this approach or by performing matrix multiplications is too computationally expensive. Instead we do the same as done in 2D.

The flux differences $A^\pm \Delta_x U_{ijk}$ are split in the y-direction according to (36), replacing the subscripts ij by ijk. This yields the transverse wave strengths $(\delta^p)^\pm$. The transverse flux difference $B^+A^+\Delta_x U_{ijk}$ is defined as in (37). Note that this term already is computed when approximating the BAu_{xy} term. Next, split this flux difference into waves propagating in the z-direction, i.e. solve

$$B^+A^+\Delta_x U_{ijk} = \sum_l \eta^l s^l. \tag{62}$$

Using this, the update of $\widetilde{H}_{i,j+1,k+1}$ reads

$$\widetilde{H}_{i,j+1,k+1} := \widetilde{H}_{i,j+1,k+1} - \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y}\sum_l (\nu^l)^+ \eta^l s^l.$$

In order to compute all updates in (60), also $B^-A^+\Delta_x U_{ijk}$ and $B^\pm A^-\Delta_x U_{ijk}$ need to be split in the z-direction. Hence, in addition to the Riemann problems needed in approximating the second order cross-derivative terms like BAu_{xy}, four new Riemann problems are introduced by the third order terms.

The scheme considered so far is only first order accurate. In order to achieve second order accuracy, the correction wave is applied as in 1D and 2D, i.e.

$$\widetilde{F}_{ijk} := \widetilde{F}_{ijk} + \frac{1}{2}\sum_p |\lambda^p| \left(1 - \frac{\Delta t}{\Delta x}|\lambda^p|\right) \widetilde{\mathscr{W}}^p_{ijk}. \tag{63}$$

From observations in 2D, one expect the stability of this scheme to be superior to the three-dimensional Lax-Wendroff scheme. The use of upwind approximations instead of centered differences, should increase stability. In fact, in 3D this is not the case. The scheme considered so far is unconditionally unstable, for a linear system if no limiters are used. However, with limiters it is capable of producing good results. This is discussed in Section 5.

4.3 Transverse propagation of the correction wave

In order to restore the stability, the correction wave has to move in the transverse direction too.

We follow the same approach as for the increment wave. The three dimensional propagation of the correction wave is split into two parts. In the first one, the wave travels as in the 2D case, i.e. into cells sharing an interface with $C_{i-1,jk}$ or C_{ijk}. Second, the wave propagates into cells only sharing an edge with one of these two cells forming the original Riemann problem.

The update associated with the part of the wave moving in the y-direction, is covered by (46) with the obvious changes in the indices. These flux updates lead to approximations of terms in

(58) involving two derivatives in x and one in y, cf. (52). The updates caused by the part of the wave moving in the z-direction are

$$
\begin{aligned}
\widetilde{H}_{ij,k+1} &:= \widetilde{H}_{ij,k+1} + \frac{1}{2}\frac{\Delta t}{\Delta x} C^{+}|A|\left(1 - |A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ijk} \\
\widetilde{H}_{ijk} &:= \widetilde{H}_{ijk} + \frac{1}{2}\frac{\Delta t}{\Delta x} C^{-}|A|\left(1 - |A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ijk} \\
\widetilde{H}_{i-1,j,k+1} &:= \widetilde{H}_{i-1,j,k+1} - \frac{1}{2}\frac{\Delta t}{\Delta x} C^{+}|A|\left(1 - |A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ijk} \\
\widetilde{H}_{i-1,jk} &:= \widetilde{H}_{i-1,jk} - \frac{1}{2}\frac{\Delta t}{\Delta x} C^{-}|A|\left(1 - |A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ijk}.
\end{aligned}
\tag{64}
$$

These updates account for the terms in the Taylor expansion which include one derivative in z and two derivatives in x. In 2D, the correction wave (42) was included into the flux differences (47) in order to save one Riemann problem. However, in 3D it is beneficial to include this wave in other waves.

The scheme as described up to this point has rather good stability properties, section 5, but it is possible to improve it. In order to increase the stability further, propagation of the correction wave in the third direction has to be implemented.

In a Taylor expansion of the scalar problem (55), all derivatives of fourth order involving two derivatives in x, and one in both y and z, add up to

$$
-\frac{1}{2}\Delta t^3 \Delta x \; abc \left(1 - \frac{\Delta t}{\Delta x} a\right) u_{xxyz}. \tag{65}
$$

Recall from the discussion in Section 3.2 that the proper updates of the correction wave was found assuming that the matrices commute. The same is assumed here, and (65) is split evenly between the $\widetilde{G}$ fluxes and the $\widetilde{H}$ fluxes in the system case, i.e.

$$
-\frac{1}{4}\Delta t^3 \Delta x \; CB|A|\left(1 - \frac{\Delta t}{\Delta x}|A|\right) u_{xxyz} - \frac{1}{4}\Delta t^3 \Delta x \; BC|A|\left(1 - \frac{\Delta t}{\Delta x}|A|\right) u_{xxzy}. \tag{66}
$$

The first term is approximated by the following flux updates

$$
\begin{aligned}
\widetilde{H}_{ij,k+1} &:= \widetilde{H}_{ij,k+1} - \frac{1}{4}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y} C^{+}|B||A|\left(1 - |A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ijk} \\
\widetilde{H}_{ijk} &:= \widetilde{H}_{ijk} - \frac{1}{4}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y} C^{-}|B||A|\left(1 - |A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ijk} \\
\widetilde{H}_{i,j+1,k+1} &:= \widetilde{H}_{i+1,j,k+1} + \frac{1}{4}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y} C^{+}B^{+}|A|\left(1 - |A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ijk} \\
\widetilde{H}_{i,j+1,k} &:= \widetilde{H}_{i,j+1,k} + \frac{1}{4}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y} C^{-}B^{+}|A|\left(1 - |A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ijk} \\
\widetilde{H}_{i,j-1,k+1} &:= \widetilde{H}_{i,j-1,k+1} - \frac{1}{4}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y} C^{+}B^{-}|A|\left(1 - |A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ijk} \\
\widetilde{H}_{i,j-1,k} &:= \widetilde{H}_{i,j-1,k} - \frac{1}{4}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y} C^{-}B^{-}|A|\left(1 - |A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ijk}.
\end{aligned}
\tag{67}
$$

In addition to these updates, six additional updates are required in which the i is replaced by $i-1$ in the flux subscripts. The sign of the update is also switched.

Note that these updates are similar to the updates used when propagating the increment wave in the third direction (60). The updates above require the solution of four additional

Riemann problems. But due to the similarities, they may be included in the computation of 3D propagation of the increment wave. The proper updates for this inclusion are

$$\begin{aligned}
B^+A^+\Delta_x U_{ijk} &:= B^+A^+\Delta_x U_{ijk} - \frac{3}{2}B^+|A|\left(1-|A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ijk}\\
B^+A^-\Delta_x U_{ijk} &:= B^+A^-\Delta_x U_{ijk} + \frac{3}{2}B^+|A|\left(1-|A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ijk}\\
B^-A^+\Delta_x U_{ijk} &:= B^-A^+\Delta_x U_{ijk} - \frac{3}{2}B^-|A|\left(1-|A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ijk}\\
B^-A^-\Delta_x U_{ijk} &:= B^-A^-\Delta_x U_{ijk} + \frac{3}{2}B^-|A|\left(1-|A|\frac{\Delta t}{\Delta x}\right)\Delta_x U_{ijk}
\end{aligned} \tag{68}$$

With these changes, both the corner transport of the increment wave and the correction wave are covered by the updates in (60).

Finally, doing similar updates for the $\widetilde{G}$ flux will give approximations to the u_{xxzy} term in (66).

4.4 Nonlinear systems

The extension from linear to nonlinear problems is done in exactly the same way as in two dimensions. The flux differences $\mathscr{A}^{\pm}\Delta_x U_{ijk}$, the waves $\mathscr{W}^p_{ijk}$, and velocities are defined using either an approximate or an exact Riemann solver.

If the Roe solver is used, the states $U_{i-1,jk}$ and U_{ijk} define the Roe-averaged value U^* used in obtaining the Jacobian A_{ijk}. This value may be used in defining the transverse wave propagation. Let $B_{ijk} = g'(U^*)$ and $C_{ijk} = h'(U^*)$. The positive and negative parts of these matrices are defined as in the previous sections. The part of the increment wave causing flux updates as in 2D is implemented as in Section 3.3. The three-dimensional propagation is defined by terms like

$$\mathscr{C}^+\mathscr{B}^+\mathscr{A}^+\Delta_x U_{ijk} = C^+_{ijk}B^+_{ijk}\mathscr{A}^+\Delta_x U_{ijk}.$$

It is not evident how to define these terms when exact solvers are used in all three directions. But to use an exact solver in the normal direction and approximate solvers in the other two directions is easily implemented, as indicated above. The propagation of the correction wave in the transverse direction are done in the obvious way using the same updates as for linear systems, except now the matrices and velocities are space dependent.

In the Appendix, a complete algorithm for the three-dimensional wave propagation scheme is given.

In the `CLAWPACK` code, three parameters are used in specifying how the different waves propagate. For example, they can be used to specify whether the increment wave only travels as in 2D or if the wave also travels in the third coordinate direction.

These parameters are needed in the next section where the stability properties are discussed, and their meaning is listed below.

$$m_1 = \begin{cases} 1 & \text{The second order correction wave is not included.} \\ & \text{Hence the method is formally first order accurate.} \\ 2 & \text{The correction wave is included.} \end{cases}$$

$$m_2 = \begin{cases} 0 & \text{No wave travels in transverse direction.} \\ 1 & \text{Increment waves propagate in transverse direction as in 2D.} \\ 2 & \text{Both increment and correction waves propagate in transverse} \\ & \text{direction as in 2D.} \end{cases}$$

$$m_3 = \begin{cases} 0 & \text{No waves propagate in the 3D way.} \\ 1 & \text{Three-dimensional propagation of the increment wave.} \\ & \text{Involves 3D extension only, requires } m_2 > 0. \\ 2 & \text{Three-dimensional propagation of increment and correction wave.} \\ & \text{Involves 3D extension only, requires } m_2 > 0. \end{cases}$$

Let Method (m_1, m_2, m_3) denote the scheme involving wave propagation defined by these parameters. For example, Method (1,1,0) define the first order scheme in which the increment wave moves as in 2D. Likewise, the complete wave propagation will be denoted Method (2,2,2).

4.5 Boundary conditions

The boundary conditions are implemented by extending the computational grid. Two cells are used outside the physical domain in each dimension. These cells, named "ghost cells" are assigned values at the beginning of each time step. Only one cell is needed outside the physical domain in order to compute the flux difference splitting $\mathscr{A}^{\pm}\Delta_x U$, the wave $\mathscr{W}^p$, and the associated velocities λ^p. However, if the waves are limited, an additional block of ghost cells are needed. Recall that the limiting is based on comparing waves from the same family emanating from neighboring cells. This comparison is done in the upwind direction. Hence, if the boundary condition models a general inflow situation, two ghost cells are needed outside the boundary.

In `CLAWPACK`, the user may specify a different number of ghost cells if a larger stencil is to be used. In the software package, several kinds of boundary conditions are provided; among them are inflow conditions, periodic conditions, and solid wall conditions for gas dynamical problems.

In many situations it is desirable to apply a boundary condition which affects the outflow as little as possible. A typical situation from gas dynamics is caused by fast moving waves like sound waves and shock waves. If these waves are not of special interest, and the aim is to advance the numerical solution to a relatively large time value, the area spanned by these waves is so large that it is not manageable to include them in the computational domain with the desired accuracy. Hence, these waves should leave the computational domain, and do so without causing non-physical reflections corrupting the computation. In connection with a Godunov method, there exist fairly simple strategies for incorporating these "open" or "non-reflecting" boundary conditions. The simplest one is to use a zeroth order extrapolation, i.e. the ghost cells are assigned the value in the cells adjacent to the boundary. The effect is that the Riemann problem at the boundary does not generate any waves. Hence, no waves will enter the computational domain. No waves will leave the area either, but these waves would only update the ghost cells, which are overwritten at the beginning of the next time step anyway. Note that waves will enter the boundary cells from the remaining interfaces, hence the solution value will change as it is advanced in time. This simple procedure seems to work well, even when waves hit the boundary at an angle other than normal to the boundary.

4.6 Relation to other approaches

A popular method for solving multi-dimensional conservation laws is to apply dimensional splitting (Strang[28]). This is a simple way of extending one-dimensional methods to problems in several space-dimensions. Due to its simplicity, this is a computationally very efficient approach. As mentioned in Section 1, the dimensional splitting suffer from grid orientation effect resulting in loss of accuracy.

In [4], Colella constructs a multi-dimensional unsplit method, and applies it successfully on the Euler equations in 2D. Saltzman extends this approach to three-dimensional problems in [25]. This class of methods is somewhat different then the wave propagation methods considered in this paper. These methods are based on the observation that $F_{ijk} = f(u^0)$ yields a second order accurate approximation to the exact flux integral

$$\frac{1}{\Delta t \Delta y \Delta z} \int_{t_n}^{t_{n+1}} \int_{z_{k-1/2}}^{z_{k+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} f(u(x_{i-1/2}, y_j, z_k, t))\, dy\, dz\, dt.$$

if $u^0 \approx u(x_{i-1/2}, y_j, z_k, t_n + \Delta t/2)$.

Two approximations u_l and u_r of $u(x_{i-1/2}, y_j, z_k, t_n + \Delta t/2)$ are made. Both are constructed using information from one side of the interface $x_{i-1/2}$, u_l from the left and u_r from the right. After establishing these approximations, we set $u^0 = RP_x(u_l, u_r)$.

The construction of these two states involves tracing characteristics backwards to the beginning of the time step, where the solution is considered to be piecewise linear. One-dimensional limiting is performed at this stage, analogous to what is done here. Both u_l and u_r contain approximations to $g(u)_y$ and $h(u)_z$, derived by solving appropriate Riemann problems in the transverse direction.

The main difference between Colella's method and the method considered here is that in the former method a lot of preprocessing is done in order to define the states used in the Riemann problem defining the numerical flux. The limiting is performed before the Riemann problem normal to the interface is solved. The number of Riemann problems required is comparable to what is needed in the wave propagation approach. However, it is possible to reduce this number significantly by proper implementation. In both Colella's approach and in the wave propagation method, the major part of the Riemann problems needed for multi-dimensional upwinding involves Riemann problems in the third coordinate direction. For example, consider the update of a flux in the x-direction. A single Riemann problem is required in the direction normal to the interface, and an increasing number of these problems are needed when computing the influence from interfaces in the y and z directions. In Colella's scheme, the Riemann problems in the z direction are based on data from the beginning of the time step, i.e. U^n, and it is therefore possible to reuse the solution when updating other fluxes in the x-direction. Contributions for the Riemann problems in the y-direction need some preprocessing, but this is most present in the computation of the Riemann problem in the x direction.

For the wave propagation method the situation is reversed. The Riemann problem in the x-direction involves the solution at the beginning of the time step, while the numerous Riemann problems in the z-direction need preprocessing. The total number of Riemann problems needed at each interface for the Colella/Saltzman method may be reduced to five. Exact Riemann solvers are most often used for this type of method.

In the wave propagation method, the one-dimensional Riemann problem is defined exactly as in 1D. After solving this problem, limiters are applied. Next these waves are propagated in the transverse direction, causing updates of the neighboring cell values. Recall that the number of Riemann problems needed is 15 per interface for the full method. But only one involves a complete solution of the Riemann problem. This makes these two approaches somewhat difficult to compare. In general, the use of an exact Riemann solver makes any method more robust than if an approximate solver is used.

Even though there are major differences in these two approaches, there are also similarities. If the Colella/Saltzman scheme is applied to a linear system with no limiters, the scheme is similar to Method (2,1,1). However, the former scheme does not embed terms corresponding to the transverse propagation of the correction wave.

Radvogin[19] has formulated an upwind scheme which is based on splitting the coefficient matrices in the same way as in (39). When applied to linear systems, this scheme is quite similar to the one presented here, though the limiters are applied differently. For other multi-dimensional unsplit schemes, consider for example [1],[2], and [31].

5 Stability

In this section the stability of the wave propagation schemes derived in the previous sections is discussed. To do this, consider the linear constant coefficient problem

$$u_t + Au_x + Bu_y + Cu_z = 0, \tag{69}$$

with periodic boundary conditions. If no limiters are used, the schemes are linear and it is appropriate to analyze the stability using a von Neumann approach, cf. [29]. In [15], the stability of the scalar two-dimensional equation is thoroughly discussed using the same tool.

In this section, let $i = \sqrt{-1}$. The Fourier mode

$$U_{IJK} = e^{i(\xi I + \eta J + \theta K)} v, \qquad v = [1, 1, \ldots, 1]^T \tag{70}$$

is inserted into the difference schemes. The wave numbers ξ, η, θ are in the range between $-\pi$ and π. For all schemes, this results in an expression of the form

$$\bar{U}_{IJK} = T(\xi, \eta, \theta) U_{IJK}, \tag{71}$$

where T in general is a complex $m \times m$ matrix. This matrix also depends on the space and time discretization, and the matrices A, B, and C. Even for scalar problems, the expression for T gets complicated as the waves involved update an increasing number of adjacent cells. Due to this it is cumbersome to derive analytical expressions for the stability, if at all possible. Therefore the stability will be determined by examining the amplification matrix numerically for a particular set of wave numbers.

Define ω_a, ω_b, and ω_c as the Courant numbers in each coordinate direction, i.e.

$$\omega_a = \frac{\Delta t}{\Delta x} \max_p |\lambda_p|, \quad \omega_b = \frac{\Delta t}{\Delta y} \max_q |\mu_q|, \quad \omega_c = \frac{\Delta t}{\Delta z} \max_l |\nu_l|. \tag{72}$$

For given wave numbers ξ, η, and θ, the scheme is stable provided $||T^n||$ stays bounded for increasing n. If T is non-defective (diagonalizable), this is the case if the spectral radius $r(T) \leq 1$. For the class of wave propagation schemes considered, the amplification matrix is not proved to be non-defective. This property is investigated numerically, and in major parts of the wave number domain, the amplification matrix T is determined to be non-defective. For those wave numbers for which the matrix is indicated to be defective, $||T^n||$ is computed for different values of n. In these cases, $\|T^n\|$ stays nicely bounded, and the spectral radius is always less than unity. For all schemes considered, $T(0,0,0) = 1$. Hence, the wave propagation schemes is said to be stable for given values of ω_a, ω_b, and ω_c if

$$\max_{\xi,\eta,\theta} r(T(\xi, \eta, \theta)) = 1,$$

where maximum is taken for $-\pi \leq \xi, \eta, \theta \leq \pi$.

The numerical von Neumann analysis is performed as follows. For the remainder of this section, let ω denote the Courant number in the special case when $\omega = \omega_a = \omega_b = \omega_c$. For given ω, $\max_{\xi,\eta,\theta} r(T)$ is estimated on a regular $10 \times 10 \times 10$ grid in the wave numbers ξ, η and θ.

ω	$\max_{\xi,\eta,\theta}(r(T))$ for								
	T^{100}	T^{110}	T^{111}	T^{210}	T^{211}	T^{220}	T^{221}	T^{222}	LW
0.1	1.00	1.00	1.00	1.06	1.05	1.00	1.00	1.00	1.00
0.2	1.00	1.00	1.00	1.24	1.18	1.00	1.00	1.00	1.00
0.3	1.00	1.00	1.00	1.54	1.32	1.00	1.00	1.00	1.03
0.4	1.40	1.00	1.00	1.96	1.45	1.00	1.00	1.00	1.17
0.5	2.00	1.00	1.00	2.50	1.50	1.00	1.50	1.00	1.57
0.6	2.60	1.72	1.00	3.16	1.43	1.06	2.02	1.00	2.21
0.7	3.20	2.68	1.00	3.94	1.26	1.37	2.33	1.00	3.01
0.8	3.80	3.88	1.00	4.84	1.11	2.00	2.33	1.00	3.97
0.9	4.40	5.32	1.00	5.86	1.02	3.92	1.92	1.00	5.08
1.0	5.00	7.00	1.00	7.00	1.00	7.00	1.00	1.00	6.32
1.01	5.06	7.18	1.06	7.12	1.12	7.37	1.14	1.12	6.45

Table 1: The spectral radius $\max(r(T))$ versus the CFL number $\omega = \omega_a = \omega_b = \omega_c$ for the scalar problem.

Below, this procedure will be applied on two problems, one scalar and one system. The numerical schemes will be identified using the three parameters presented in the previous section. For example T^{210} equals the amplification matrix obtained when $m_1 = 2$, $m_2 = 1$, and $m_3 = 0$.

Example 1: The first problem to be considered is $u_t + au_x + bu_y + cu_z = 0$, with $a = b = c = 1$. For this simple equation, it is possible, to some extent, to derive analytical results. The expressions for T given below are derived under the assumption of positive velocities. The stability results are given in Table 1.

In the simplest case, i.e. Method (1,0,0), the increment wave only moves normal to the cell interfaces. In the case of positive velocities

$$T^{100} = 1 - \omega_a(1 - e^{-i\xi}) - \omega_b(1 - e^{-i\eta}) - \omega_c(1 - e^{-i\theta}).$$

Consider the special case when $\omega = \omega_a = \omega_b = \omega_c$. Then $\max_{\xi,\eta,\theta}(|T^{100}|) = |1 - 6\omega|$, and the scheme is stable provided $\omega \leq 1/3$ This is in agreement with the well known stability result for this scheme, $a\frac{\Delta t}{\Delta x} + b\frac{\Delta t}{\Delta y} + c\frac{\Delta t}{\Delta z} \leq 1$.

When the increment wave propagates into adjacent cells as in 2D, the amplification matrix reads

$$\begin{aligned} T^{110} &= T^{100} + \omega_a\omega_b(1 - e^{-i\xi})(1 - e^{-i\eta}) + \omega_a\omega_c(1 - e^{-i\xi})(1 - e^{-i\theta}) \\ &+ \omega_b\omega_c(1 - e^{-i\eta})(1 - e^{-i\theta}). \end{aligned} \tag{73}$$

Consider again the special case in which the one-dimensional Courant numbers are equal. Inserting $\xi = \eta = \theta = \pi$ yields $T^{110} = |1 - 6\omega + 12\omega^2|$. This expression gives the values in the table as long as the scheme is unstable. The maximum values for $|T^{110}|$ are attained for these wave numbers. According to the table, the stability is increased to $\omega \leq 1/2$.

When the increment wave is allowed to move in all three directions, it is possible to write T^{111} in the compact form

$$T^{111} = (1 - \omega_a(1 - e^{-i\xi}))(1 - \omega_b(1 - e^{-i\eta}))(1 - \omega_c(1 - e^{-i\theta})). \tag{74}$$

It follows easily that $|T^{111}| \leq 1$ as long as $\max(\omega_a, \omega_b, \omega_c) \leq 1$. Hence, compared to the simple (1,0,0) method, the stability is increased by a factor 3.

In the scalar case, the amplification matrix for the formally second order Lax-Wendroff scheme reads

$$
\begin{aligned}
T^{LW} &= T^{100} + \frac{1}{2}\omega_a(1-\omega_a)(1-e^{-i\xi})(1-e^{i\xi}) + \frac{1}{2}\omega_b(1-\omega_b)(1-e^{-i\eta})(1-e^{i\eta}) \\
&+ \frac{1}{2}\omega_c(1-\omega_c)(1-e^{-i\theta})(1-e^{i\theta}) \\
&+ \frac{1}{4}\omega_a\omega_b(e^{i\xi}-e^{-i\xi})(e^{i\eta}-e^{-i\eta}) + \frac{1}{4}\omega_a\omega_c(e^{i\xi}-e^{-i\xi})(e^{i\theta}-e^{-i\theta}) \\
&+ \frac{1}{4}\omega_b\omega_c(e^{i\eta}-e^{-i\eta})(e^{i\theta}-e^{-i\theta}).
\end{aligned} \tag{75}
$$

When all the one-dimensional Courant numbers are equal, the stability limit of this scheme has been determined numerically to be $\omega \leq 0.1925$.

In the 2D case, changing the approximation of the cross-derivative terms from centered to upwind, increases the stability. In Method (2,1,0), the same upwind handling has been used for the cross-derivative terms in 3D. For this scheme, the amplification matrix reads

$$
\begin{aligned}
T^{210} &= T^{110} + \frac{1}{2}\omega_a(1-\omega_a)(1-e^{-i\xi})(1-e^{i\xi}) + \frac{1}{2}\omega_b(1-\omega_b)(1-e^{-i\eta})(1-e^{i\eta}) \\
&+ \frac{1}{2}\omega_c(1-\omega_c)(1-e^{-i\theta})(1-e^{i\theta}).
\end{aligned} \tag{76}
$$

In the special case when $\omega = \omega_a = \omega_b = \omega_c$, inserting $\xi = \eta = \theta = \pi$ into the expression above gives $|T^{210}| = 1 + 6\omega^2$. Hence the scheme is unconditionally unstable, which is also reflected in Table 1. This is a rather surprising result.

In order to add stability, the increment wave is propagated as in Method (1,1,1). The expression for T^{211} equals (76) with T^{111} replaced by T^{110}. In this case we have $T^{211}(\pi,\pi,\pi) = 1 + 6\omega^2 - 8\omega^3$. The scheme is still unconditionally unstable, but the instability is less pronounced than for Method (2,1,0). From the table, note that the scheme has an isolated point of stability at $\omega = 1.0$. For this value the scheme happens to be exact.

The transverse propagation of the correction wave has to come into play in order to obtain a stable second order accurate scheme. In Method (2,2,0) both the increment wave and the correction wave travel as in the two-dimensional case. For this scheme

$$
\begin{aligned}
T^{220} &= T^{210} + \frac{1}{2}\omega_a\omega_b(1-\omega_a)(e^{i\xi}-1)(1-e^{-i\xi})(1-e^{-i\eta}) \\
&+ \frac{1}{2}\omega_a\omega_c(1-\omega_a)(e^{i\xi}-1)(1-e^{-i\xi})(1-e^{-i\theta}) \\
&+ \frac{1}{2}\omega_a\omega_b(1-\omega_b)(e^{i\eta}-1)(1-e^{-i\eta})(1-e^{-i\eta}) \\
&+ \frac{1}{2}\omega_b\omega_c(1-\omega_b)(e^{i\eta}-1)(1-e^{-i\eta})(1-e^{-i\theta}) \\
&+ \frac{1}{2}\omega_b\omega_c(1-\omega_c)(e^{i\theta}-1)(1-e^{-i\theta})(1-e^{-i\xi}) \\
&+ \frac{1}{2}\omega_b\omega_c(1-\omega_c)(e^{i\theta}-1)(1-e^{-i\theta})(1-e^{-i\eta}).
\end{aligned} \tag{77}
$$

According to Table 1, this scheme is stable for $\omega \leq 0.5$ In Method (2,2,1), the increment wave propagates as in Method (1,1,1). The amplification matrix equals the expression above, with T^{210} replaced by T^{211}. In this case $T^{221}(\pi,\pi,\pi) = 1 - 18\omega^2 + 16\omega^3$. The maximum value is also attained for these arguments in the unstable regime. Hence the scheme is stable as long as $\omega \leq (1+\sqrt{33})/16 \approx 0.4215$. Compared to Method (2,2,0), the stability region is slightly reduced, but the instability is less pronounced. As for Method (2,1,1), the scheme is exact for $\omega = 1.0$.

Finally, if also the correction wave propagates in a three-dimensional manner, the amplification matrix reads

$$\begin{aligned} T^{222} &= T^{221} - \frac{1}{2}\omega_a\omega_b\omega_c(1-\omega_a)(e^{i\xi}-1)(1-e^{-i\xi})(1-e^{-i\eta})(1-e^{-i\theta}) \\ &- \frac{1}{2}\omega_a\omega_b\omega_c(1-\omega_b)(e^{i\eta}-1)(1-e^{-i\eta})(1-e^{-i\eta})(1-e^{-i\theta}) \\ &- \frac{1}{2}\omega_a\omega_b\omega_c(1-\omega_c)(e^{i\theta}-1)(1-e^{-i\theta})(1-e^{-i\xi})(1-e^{-i\eta}). \end{aligned} \tag{78}$$

According to Table 1, the wave propagation scheme is stable for $\omega \leq 1$, which is the best one can expect.

To summarize, in 2D it is sufficient to include transverse propagation of the increment wave to achieve a second order scheme with stability limit $\omega \leq 1$. In 3D however, it is necessary to also propagate the correction wave in the transverse direction in order to achieve the same. Speaking in terms of Taylor expansions, in 2D terms like u_{xy}, u_{yx} are sufficient, while in 3D terms up to $u_{xxyz}, u_{yyxz}, \ldots$ have to be included.

Example 2: In the second problem, consider the Euler equations in the primitive variables ρ, u, v, w, and p. As usual, ρ and p denote the density and the pressure, while u, v, and w are the velocity components in the three coordinate directions. Finally, γ is a gas constant. The equations are linearized using the constant state $(\rho_0, u_0, v_0, w_0, p_0)$. The matrices involved are

$$A = \begin{bmatrix} u_0 & \rho_0 & 0 & 0 & 0 \\ 0 & u_0 & 0 & 0 & 1/\rho_0 \\ 0 & 0 & u_0 & 0 & 0 \\ 0 & 0 & 0 & u_0 & 0 \\ 0 & \gamma p_0 & 0 & 0 & u_0 \end{bmatrix}, \quad B = \begin{bmatrix} v_0 & 0 & \rho_0 & 0 & 0 \\ 0 & v_0 & 0 & 0 & 0 \\ 0 & 0 & v_0 & 0 & 1/\rho_0 \\ 0 & 0 & 0 & v_0 & 0 \\ 0 & 0 & \gamma p_0 & 0 & u_0 \end{bmatrix}, \tag{79}$$

$$C = \begin{bmatrix} w_0 & 0 & 0 & \rho_0 & 0 \\ 0 & w_0 & 0 & 0 & 0 \\ 0 & 0 & w_0 & 0 & 0 \\ 0 & 0 & 0 & w_0 & 1/\rho_0 \\ 0 & 0 & 0 & \gamma p_0 & w_0 \end{bmatrix}.$$

The eigenvalues are $\lambda = \{u_0 - c_0, u_0, u_0, u_0, u_0 + c_0\}$, $\mu = \{v_0 - c_0, v_0, v_0, v_0, v_0 + c_0\}$, and $\nu = \{w_0 - c_0, w_0, w_0, w_0, w_0 + c_0\}$. Here $c_0 = \sqrt{\gamma p_0/\rho_0}$ denotes the sound speed. The stability depends on these constant values. If $u_0 = v_0 = w_0 = 0$, $\gamma p_0 = \rho_0 = 1$ are used, the stability results are nearly identical to what was achieved for the scalar problem.

In order to get a different picture, the following values are used in the computations below, $u_0 = v_0 = w_0 = 1$, $\gamma p_0 = 2$ and $\rho_0 = 1$.

The expressions for $T(\xi, \eta, \theta)$ are not hard to derive, but involve a large number of terms as the algorithm includes an increasing number of updates of neighboring cells. For example, the amplification matrix for Method (1,0,0) reads

$$\begin{aligned} T^{100} &= I - \frac{\Delta t}{\Delta x}[A^+(1-e^{-i\xi}) + A^-(e^{i\xi}-1)] \\ &- \frac{\Delta t}{\Delta y}[B^+(1-e^{-i\eta}) + B^-(e^{i\eta}-1)] \\ &- \frac{\Delta t}{\Delta z}[C^+(1-e^{-i\theta}) + C^-(e^{i\theta}-1)]. \end{aligned}$$

ω	$\max_{\xi,\eta,\theta}(r(T))$ for								
	T^{100}	T^{110}	T^{111}	T^{210}	T^{211}	T^{220}	T^{221}	T^{222}	LW
0.1	1.00	1.00	1.00	1.03	1.02	1.00	1.00	1.00	1.00
0.2	1.00	1.00	1.00	1.10	1.08	1.00	1.00	1.00	1.00
0.3	1.00	1.00	1.00	1.23	1.15	1.00	1.00	1.00	1.00
0.4	1.00	1.00	1.00	1.41	1.22	1.00	1.00	1.00	1.02
0.5	1.32	1.00	1.00	1.63	1.28	1.00	1.00	1.00	1.12
0.6	1.79	1.00	1.00	1.91	1.31	1.00	1.05	1.00	1.37
0.7	2.25	1.01	1.00	2.24	1.31	1.00	1.37	1.00	1.80
0.8	2.72	1.53	1.00	2.62	1.37	1.01	1.55	1.00	2.34
0.9	3.18	2.19	1.00	3.05	1.42	1.17	1.53	1.00	2.99
1.0	3.65	2.98	1.00	3.54	1.46	1.63	1.23	1.00	3.74
1.01	3.69	3.07	1.02	3.59	1.47	1.72	1.23	1.04	3.82

Table 2: The spectral radius $\max_{\xi,\eta,\theta}(r(T))$ versus the CFL number ω for the Euler system.

The spectral radius of the amplification matrix is given in Table 2.

The tendency is the same in this problem as in the previous. Propagating the increment waves, as in 2D, adds stability compared to the simple Method (1,0,0). If the increment wave also updates cells sharing only an edge with the two cells defining the original Riemann problem, stability is obtained if $\omega \leq 1$. Hence, Method (1,1,1) is the optimal first order method when it comes to stability.

Replacing the centered approximations of cross-derivatives in the Lax-Wendroff scheme with upwind approximations results in an unconditionally unstable method. The degree of instability is reduced in Method (2,1,1), but still the scheme is unconditionally unstable. Introducing transverse propagation of the correction wave adds stability. Still, the stability area of Method (2,2,0) is larger than for Method (2,2,1), but the latter has less pronounced instability, as observed for the scalar example. When both the increment wave and the correction wave propagate in a three dimensional way, the scheme is stable provided $\omega \leq 1$.

Note that the difference in stability between Method (2,2,0) and Method (2,2,2) is less than for the scalar problem. From experiments involving other parameters and equations, the most restrictive stability limit for Method (2,2,0) appears to be $\omega \leq 1/2$. As discussed in Section 4, the updates from one single interface in Method (2,2,2) involve 15 Riemann problems, while the same number for Method (2,2,0) is 5. Hence, Method (2,2,0) may, for certain applications, be computationally more efficient than Method (2,2,2).

Recall from the previous sections that the transverse propagation of the correction wave consists of several terms from a Taylor expansion added together. This is obtained under the assumption that the involved matrices commute. The matrices in the system (79) do not commute, but the stability does not seem to suffer from this. This is also experienced when considering other systems. It seems like the stability is not sensitive to the commutation of the matrices.

5.1 Wave number dependency and limiter influence

From numerical experiments we have observed that the unconditionally unstable methods (2,1,0) and (2,1,1) do not behave as badly as the discussion above may indicate. This is especially true for Method (2,1,1). But even Method (2,1,0) may produce nice results when applied on nonlinear problems *if* limiters are used. This indicates that the instability is strongest for large

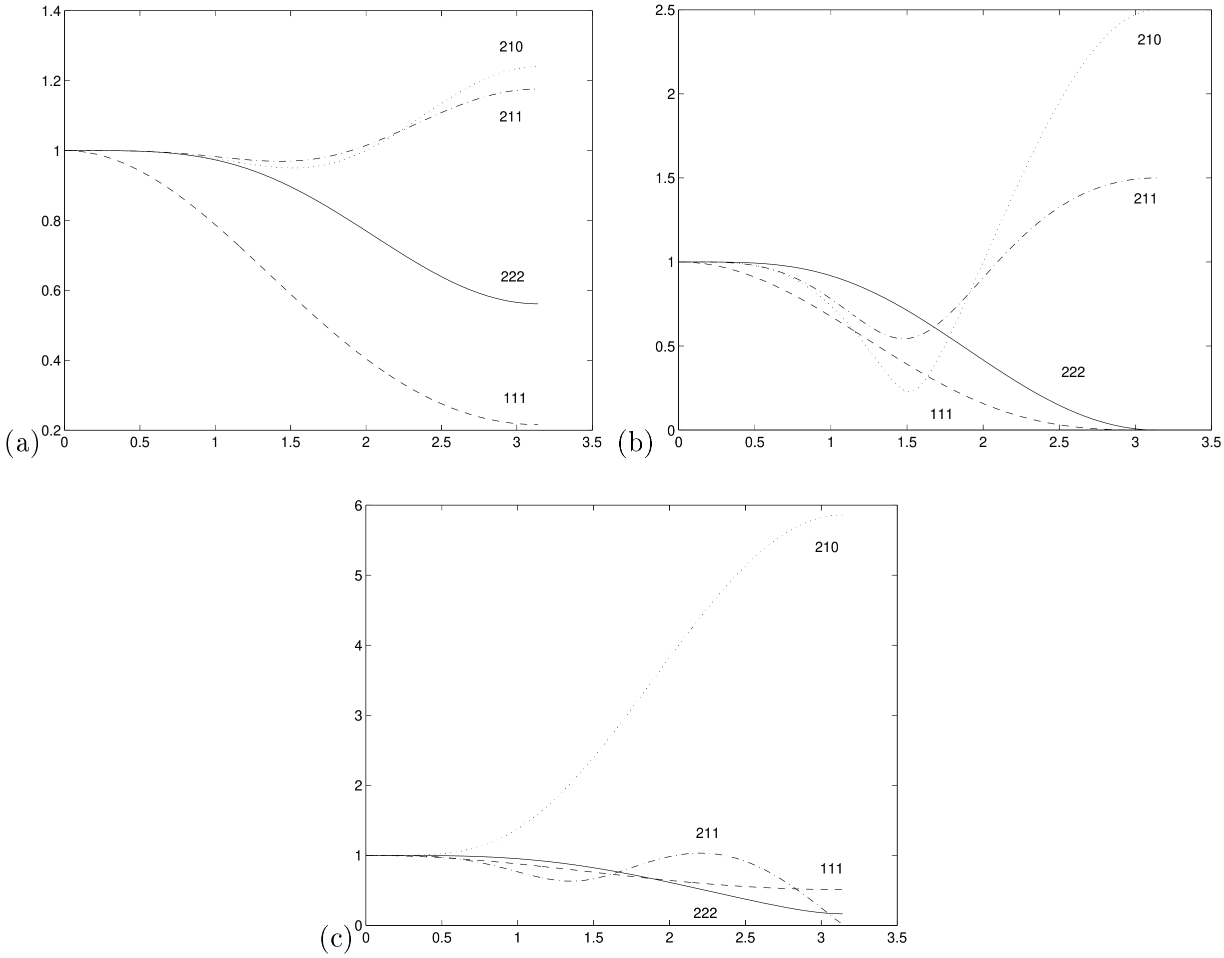


Figure 5: The absolute value of the amplification factor versus the wave number for the scalar problem. The Courant numbers are held fixed: (a) $\omega = 0.2$, (b) $\omega = 0.5$, (c) $\omega = 0.9$.

wave numbers. Limiters are assumed to be most effective in this range. Here we reconsider the two test problems from the previous section, and examine how the stability depends on wave numbers and limiters.

In order to simplify the discussion, we consider the case when the wave numbers are equal, i.e. $\xi = \eta = \theta$. In Figure 5, the absolute value of the amplification factor versus the wave number for the scalar problem is depicted for three different values of the Courant number ω. The analytical expressions derived in the previous section are used. For all schemes, $|T|$ is symmetric about $\xi = 0$, and in the figures only $0 \leq \xi \leq \pi$ is given.

As expected, the stable methods (1,1,1) and (2,2,2) are uniformly bounded by one, and the former scheme is more dissipative than the latter. Note that both schemes are "infinitely" dissipative near the wave number π when $\omega = 0.5$. When it comes to the unstable schemes, Method (2,1,0) is more unstable than Method (2,1,1) for nearly all wave numbers and Courant numbers. Also note that Method (2,1,1) behaves well near π when $\omega = 0.9$.

These results may be confirmed numerically by considering the initial condition

$$u_0(x, y, z) = \sin(2\pi nx + 2\pi ny + 2\pi nz), \tag{80}$$

where n is a positive integer. Assume that the computational domain is $[0,1] \times [0,1] \times [0,1]$ and

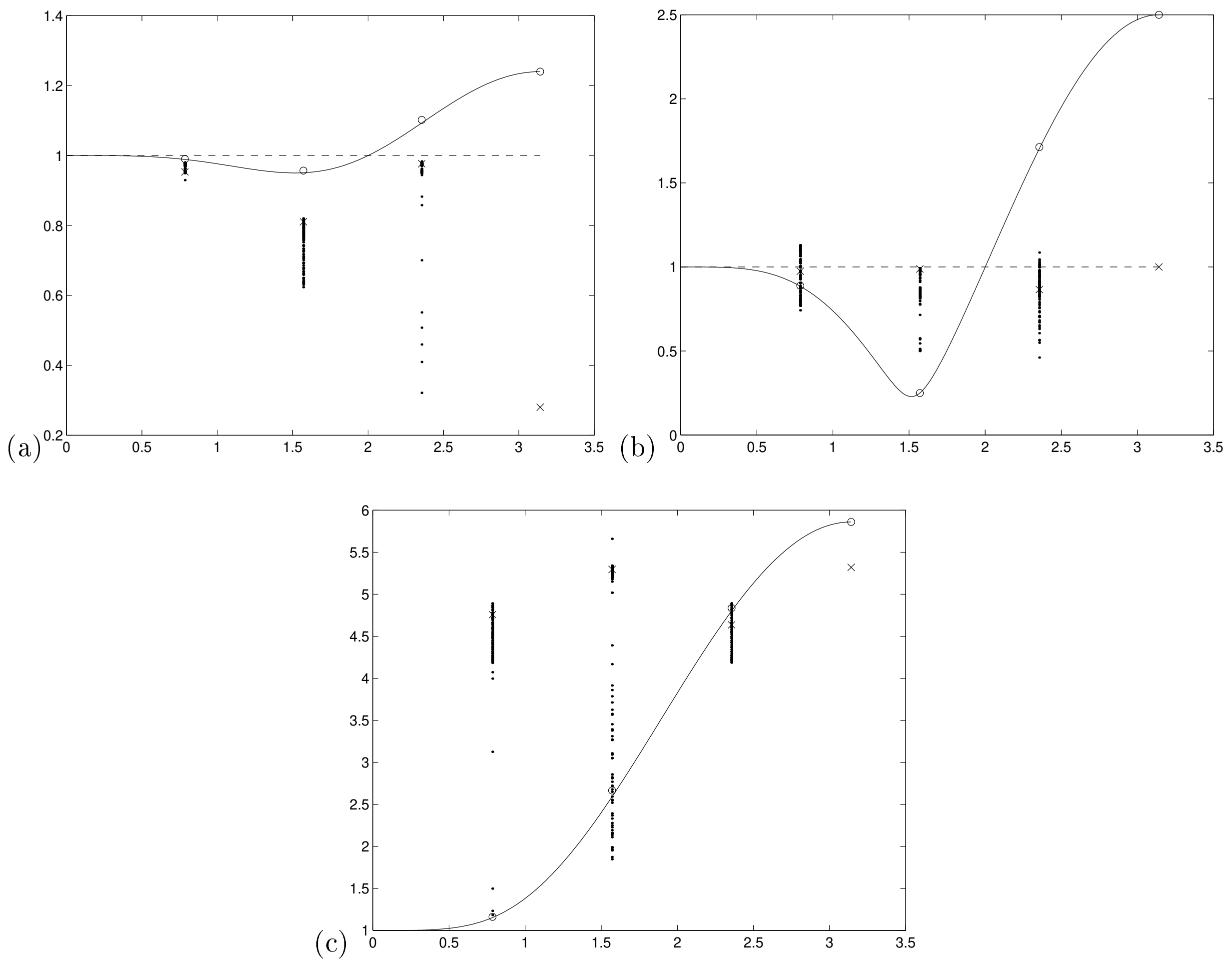


Figure 6: The absolute value of the amplification factor versus the wave number for Method 210 applied to the scalar problem. The Courant numbers are: (a) $\omega = 0.2$, (b) $\omega = 0.5$, (c) $\omega = 0.9$. The circles (o) correspond to numerical experiments, while the dots and the x-marks correspond to the use of limiters (see text).

that $\Delta x = \Delta y = \Delta z$. The wavelength in the Fourier mode (70) is $2\pi\Delta x/\xi$, and in (80) the wavelength is $1/n$. Hence

$$\xi = \frac{2\pi n}{N},$$

where N is the number of cells in each coordinate direction. Since the shortest wavelength obtainable on the grid is $2\Delta x$, acceptable values for n are $1, 2, \ldots, N/2$, assuming that N is even.

In Figure 6, the estimated amplification factor is depicted for Method (2,1,0). In these computations $N = 40$ and $n = 5, 10, 15, 20$ are used. The associated value for the amplification matrix is marked with circles (o) in the plot. The agreement between the analytical curve and the estimated values are good.

It is of interest to consider what the use of limiters does to this picture. The computations are redone, but now with the MC-limiter. Since the expression (71) is no longer valid, the ratio

$$\max_{IJK} |\bar{U}_{IJK}| / \max_{IJK} |U_{IJK}|$$

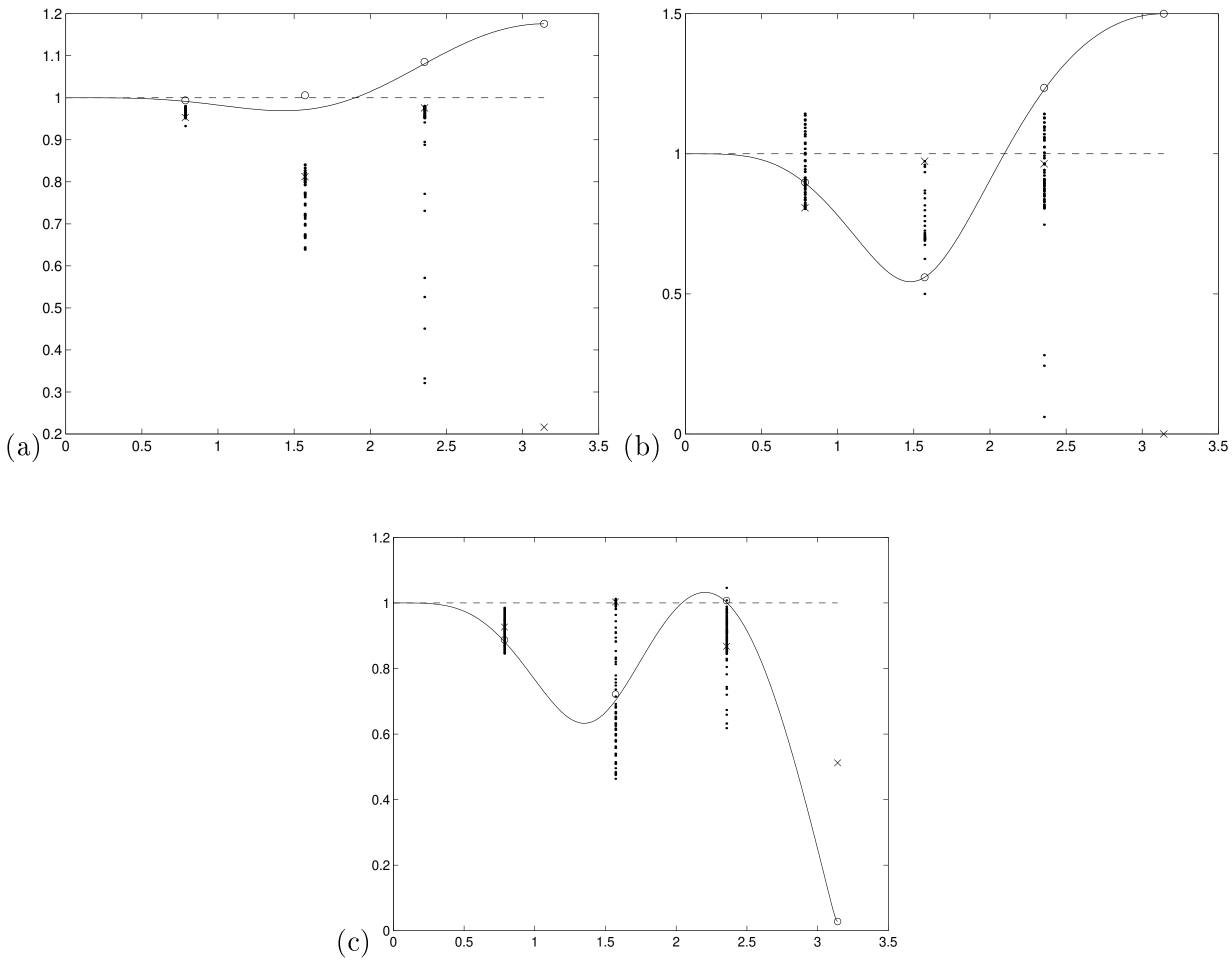


Figure 7: The absolute value of the amplification factor versus the wave number for Method 211 applied to the scalar problem. The Courant numbers are: (a) $\omega = 0.2$, (b) $\omega = 0.5$, (c) $\omega = 0.9$. The circles (o) correspond to numerical experiments, while the dots and the x-marks correspond to the use of limiters (see text).

is plotted as a dot in the figure for each of the first 100 time steps. The ratio for the last time step is marked with a cross (x). The overall picture is that the limiters may prevent the solution from going unstable when the Courant numbers are small enough. For larger Courant numbers, the limiter does not help. Note that when $n = 20$, yielding a wavelength of $2\Delta x$, the gradient in the numerical solution changes sign from one cell to the other. In this case the limiter becomes zero, the high order terms are switched off and the scheme is equivalent to Method (1,1,0).

In Figure 7, results from similar computations using Method (2,1,1) are given. Computations using the initial condition (80) confirm the results obtained from the analytical expressions. In general it seems like the use of limiters decreases the instability, which is in agreement with what is observed in computations. At wavelength $2\Delta x$ ($n = 20$), the Method (2,1,1) reduces to Method (1,1,1) since the limiter in this case turns off higher order terms.

The numerical example to be presented in Section 6.2 models an initially spherical shock wave expanding in a slab between two parallel walls. This problem was also studied in [10]. The method used was (2,1,1) at a Courant number of about 0.9, and the computation did not suffer from any instability problems.

In the previous section we also considered the linearized Euler equations. In Figure 8, the

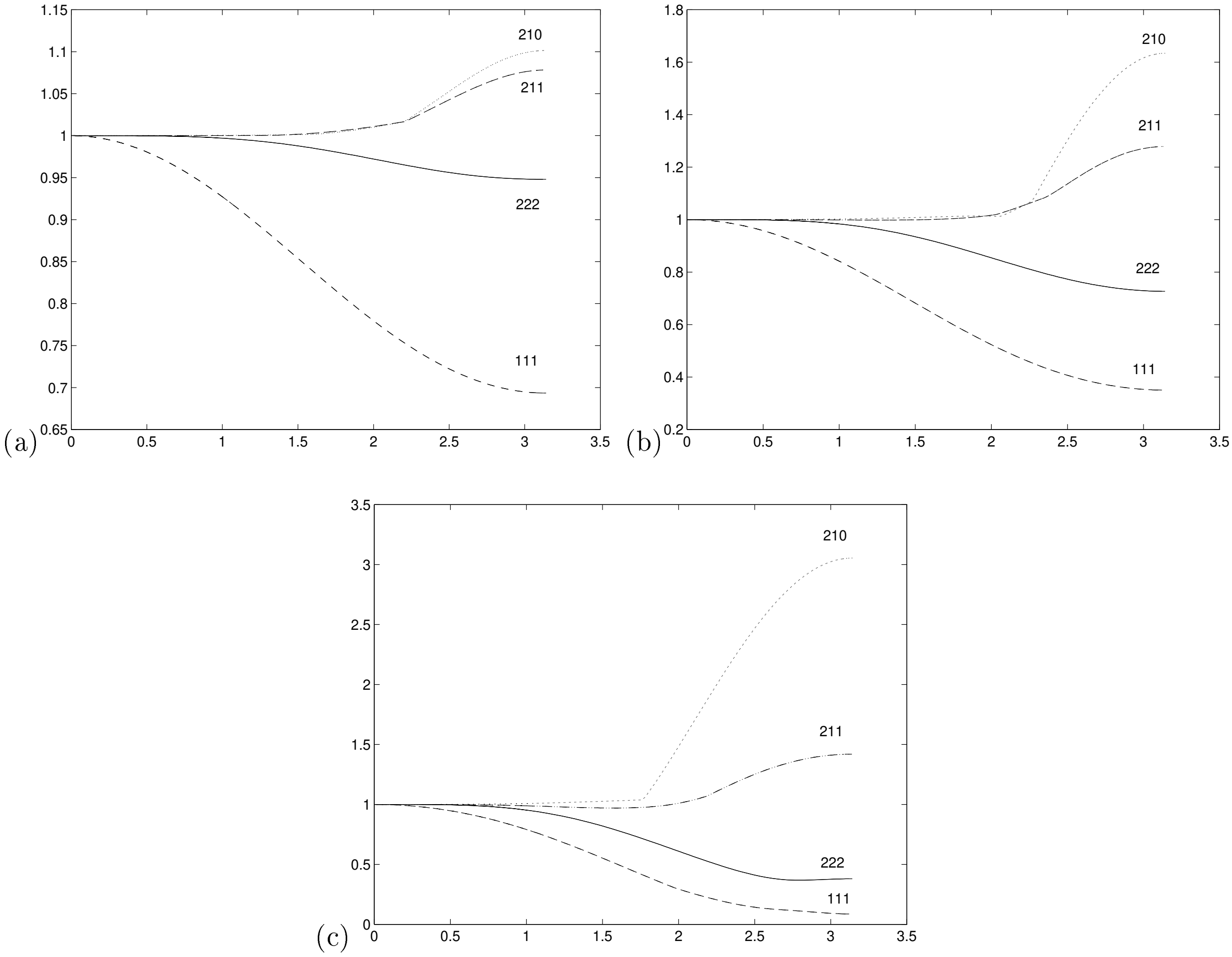


Figure 8: The spectral radius of the amplification matrix versus the wave number for the linearized Euler equations. The Courant numbers are held fixed: (a) $\omega = 0.2$, (b) $\omega = 0.5$, (c) $\omega = 0.9$.

wave number dependency is depicted for some of the schemes for three different Courant numbers. Compared to the behavior of the schemes when applied to the scalar problem (Figure 5), the situation is clearer. For a given wave number, the tendency is that $r(T^{111}) < r(T^{222}) < r(T^{211}) < r(T^{210})$. Also note that for both the unstable schemes, $r(T)$ stays close to unity except for "large" wave numbers. This is especially the case for Method (2,1,1). Since limiters tend to damp the influence of large wave numbers, the expected instability may be hard to observe when limiters are used.

Limiters are constructed in order to reduce oscillations near discontinuities. They are not intended to increase the stability range of the scheme. Nevertheless, it seems like the use of limiters has a positive effect on Method (2,1,1), and to some extent on method (2,1,0), when it comes to this.

5.2 Alternative approximation of second order terms

Features of the MC-limiter motivates an alternative approximation of second order derivatives. When the solution is sufficiently smooth, the MC-limiter yields $\phi(\theta) = (1+\theta)/2$. For the one-dimensional advection problem $u_t + au_x = 0$, $a > 0$, this gives the flux

$$\widetilde{F}_{i+1} = aU_i - \frac{1}{4}a\left(1 - a\frac{\Delta t}{\Delta x}\right)(U_{i+1} - U_{i-1}).$$

ω	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	1.01
$\max(r(T^{210}))$	1.00	1.00	1.00	1.00	1.27	1.92	2.75	3.88	5.32	7.00	7.18
$\max(r(T^{211}))$	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.07

Table 3: The spectral radius $\max(r(T))$ versus the CFL number ω for the scalar problem, when u_{xx} is approximated alternatively. The modified versions of Method (2,1,0) and Method (2,1,1) are considered.

This one-dimensional scheme is known as the Fromm scheme, and was used by van Leer [32] to construct a monotone and conservative second order method.

If this approximation is used in the wave propagation scheme, the stability properties are changed significantly. In Table 3, the spectral radius is shown for the modified version of Method (2,1,0) and Method (2,1,1) when applied to the scalar problem in Example 1. The same set of Courant numbers are used as in Table 1.

The stability properties are improved for both schemes. None of the schemes are unconditionally unstable. Method (2,1,0) is now stable for $\omega \leq 0.5$, while Method (2,1,1) has gained optimal stability, i.e. stability for $\omega \leq 1.0$. In Figure 9, the wave number dependency of the modified version of Method (2,1,1) is depicted for the same Courant numbers as used in Figure 7.

When it comes to stability, the overall impression is that the alternative way of approximating second order terms is superior to the one used so far. This observation also indicates why the MC-limiter obviously adds stability. Unfortunately, it is not evident how to make a non-oscillatory version of the scheme for nonlinear systems, when the second order term u_{xx} is approximated in a Fromm sense. This is a problem studied by van Leer in [33].

6 Numerical results

In this section we consider the Euler equations which models an inviscid, compressible and non-heat conducting gas. The system may be written as

$$\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}_t + \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(E+p) \end{bmatrix}_x + \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(E+p) \end{bmatrix}_y + \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(E+p) \end{bmatrix}_z = 0, \tag{81}$$

where ρ and p denote the density and pressure. The velocity components in the x, y, and z directions are u, v, and w, respectively.
If we assume that the gas is polytropic, the total energy E is

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

where γ is a gas constant taken to be 1.4 in the examples below. In all examples to be considered, the Roe solver is used, cf. [21].

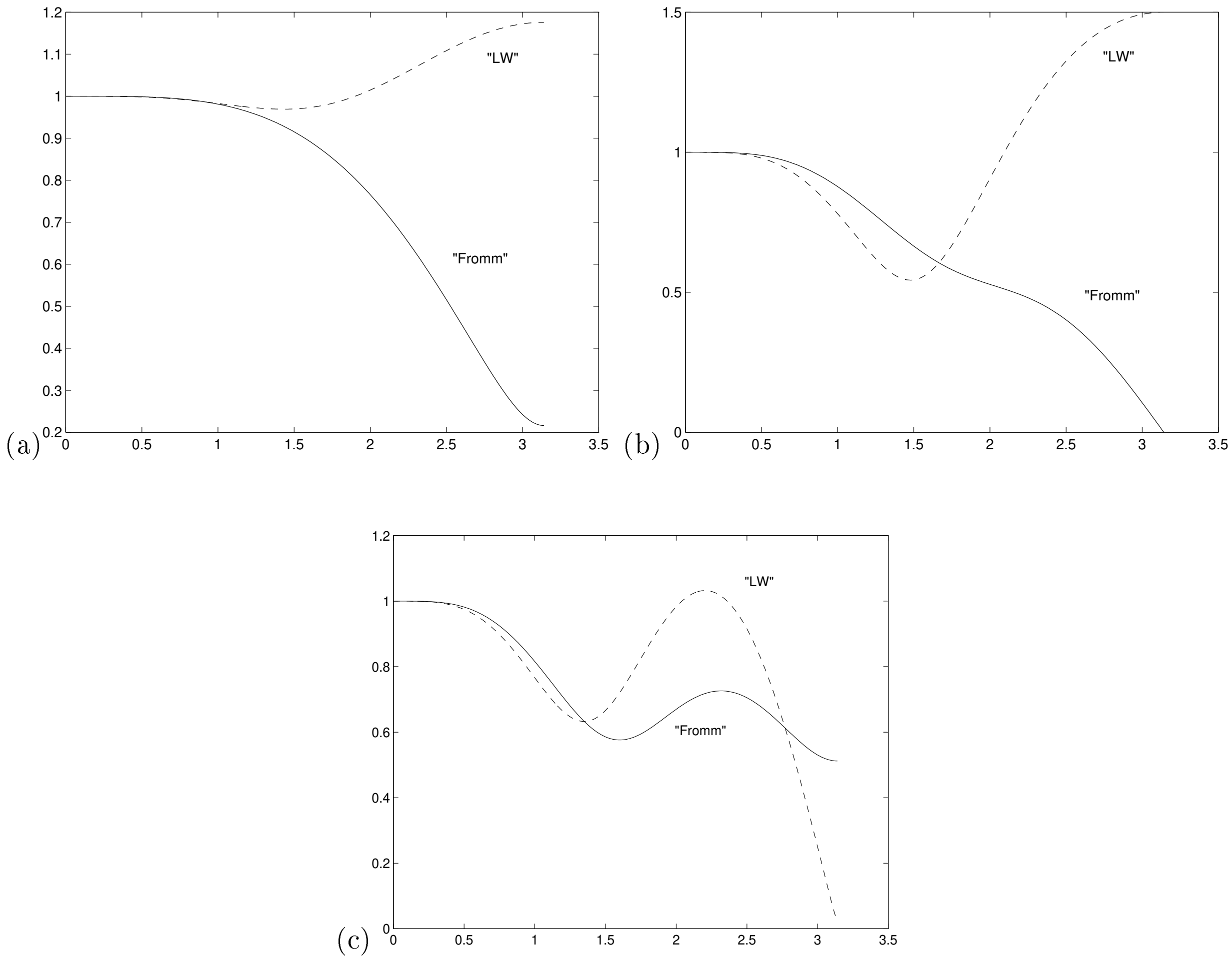


Figure 9: The absolute value of the amplification factor versus the wave number for Method (2,1,1) applied to the scalar problem, when different flux updates are used for the term u_{xx}. The dashed curve represents the version in which the discretization is done according to Law-Wendroff, i.e. $\phi(\theta) = 1$. The solid curve represents the use of the Fromm discretization, i.e. $\phi = (1 + \theta)/2$. The Courant numbers are: (a) $\omega = 0.2$, (b) $\omega = 0.5$, (c) $\omega = 0.9$.

	1-norm errors			max-norm errors		
N	ρ	ρu	E	ρ	ρu	E
20	$6.482 \cdot 10^{-3}$	$2.498 \cdot 10^{-3}$	$9.425 \cdot 10^{-3}$	$9.398 \cdot 10^{-3}$	$7.496 \cdot 10^{-3}$	$1.373 \cdot 10^{-2}$
40	$3.492 \cdot 10^{-3}$	$1.338 \cdot 10^{-3}$	$5.074 \cdot 10^{-3}$	$5.292 \cdot 10^{-3}$	$4.389 \cdot 10^{-3}$	$7.738 \cdot 10^{-3}$
80	$1.831 \cdot 10^{-3}$	$7.003 \cdot 10^{-4}$	$2.658 \cdot 10^{-3}$	$2.729 \cdot 10^{-3}$	$2.350 \cdot 10^{-3}$	$3.984 \cdot 10^{-3}$
order	0.93	0.93	0.93	0.96	0.90	0.96

Table 4: Errors for the smooth Euler problem computed with Method (1,1,1) on a $N \times N \times N$ grid. The errors in ρv and ρw equals the errors in ρu. The convergence order is estimated on the two finest grids.

6.1 Smooth Euler solution

In order to verify second order convergence, we consider an initial value problem with a smooth solution. Initially, the gas is at rest and

$$\rho(x, y, z, 0) = E(x, y, z, 0) = 1 + e^{-30(r-1)^2},$$

where $r = \sqrt{x^2 + y^2 + z^2}$. The solution will remain spherically symmetric. Due to this, it is possible to formulate (81) as a one-dimensional conservation law with a source term, i.e.

$$\begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix}_t + \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{bmatrix}_r = -\frac{1}{r} \begin{bmatrix} \rho u \\ \rho u^2 \\ u(E + p) \end{bmatrix}. \tag{82}$$

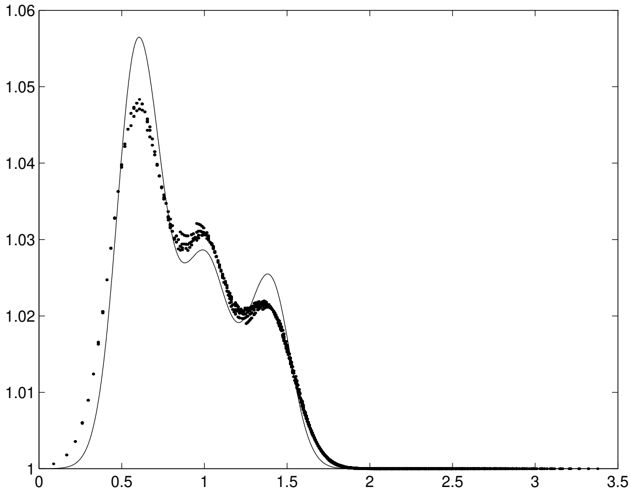
Here, u denotes the radial velocity. An accurate reference solution is made by solving this equation using `CLAWPACK` routines. Due to the symmetry, the computational domain for the three-dimensional scheme is $(x, y, z) \in [0, 2] \times [0, 2] \times [0, 2]$. At the boundaries $x = 0$, $y = 0$, and $z = 0$, symmetric boundary conditions are used. The ghost cells adjacent to the boundary is assigned the same value as in the boundary cell, except for the normal velocity which is negated. The same procedure is in fact used when solid walls are modelled. The remaining boundaries are all of the outflow type described in Section 4.5.

The solution is computed on a $N \times N \times N$ grid at time $t = 0.5$. In Table 4, the errors for Method (1,1,1) are given for $N = 20$, 40 and 80. The errors are computed by comparing with results computed with the one-dimensional scheme. The convergence rate q is estimated using the two finest grids according to the formula

$$q = \frac{1}{\ln 2} \ln \left(\frac{\text{error}(\Delta x)}{\text{error}(\Delta x/2)} \right).$$

The errors in ρv and ρw equals the errors in ρu. As expected, this scheme is approximately first order accurate. In Figure 10, scatter plots of the density using $N = 20$ and $N = 40$ are shown. Every value in the $N \times N \times N$ grid is plotted against r, the distance from the origin. The solid curve corresponds to the accurate computation produced by the one-dimensional problem.

In Table 5 and Figure 11, the errors and scatter plots are given for the complete wave propagation scheme, i.e. Method (2,2,2). Since the solution is smooth, no limiting is performed. According to the figures in the table, the scheme is second order accurate. In both computations fixed time steps are used giving a Courant number of approximately 0.78.



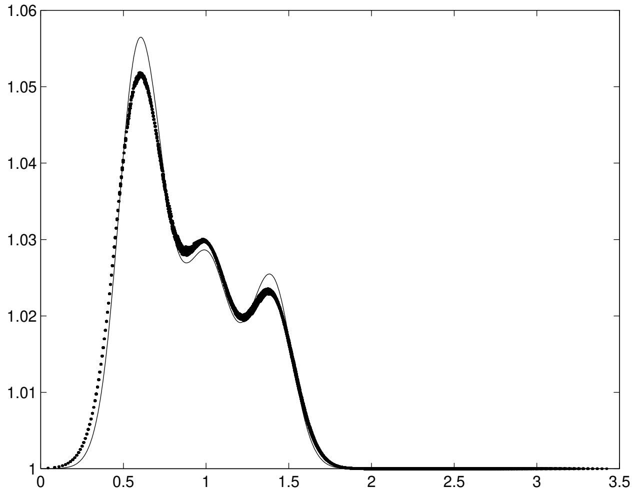


Figure 10: Scatter plot for the Euler equation with smooth solution, computed with Method (1,1,1). The density is depicted against the distance from origin. In the left picture, $N = 20$, while in the right picture $N = 40$. The solid curve corresponds to the reference solution.

	1-norm errors			max-norm errors		
N	ρ	ρu	E	ρ	ρu	E
20	$4.334 \cdot 10^{-3}$	$1.705 \cdot 10^{-3}$	$6.290 \cdot 10^{-3}$	$6.985 \cdot 10^{-3}$	$5.140 \cdot 10^{-3}$	$1.012 \cdot 10^{-2}$
40	$1.361 \cdot 10^{-3}$	$4.767 \cdot 10^{-4}$	$1.961 \cdot 10^{-3}$	$2.331 \cdot 10^{-3}$	$1.761 \cdot 10^{-3}$	$3.368 \cdot 10^{-3}$
80	$3.541 \cdot 10^{-4}$	$1.254 \cdot 10^{-4}$	$5.091 \cdot 10^{-4}$	$6.387 \cdot 10^{-4}$	$4.931 \cdot 10^{-4}$	$9.216 \cdot 10^{-4}$
order	1.94	1.93	1.95	1.87	1.84	1.87

Table 5: Errors for the smooth Euler problem computed with Method (2,2,2) on a $N \times N \times N$ grid. The errors in ρv and ρw equals the errors in ρu. The convergence order is estimated on the two finest grids.

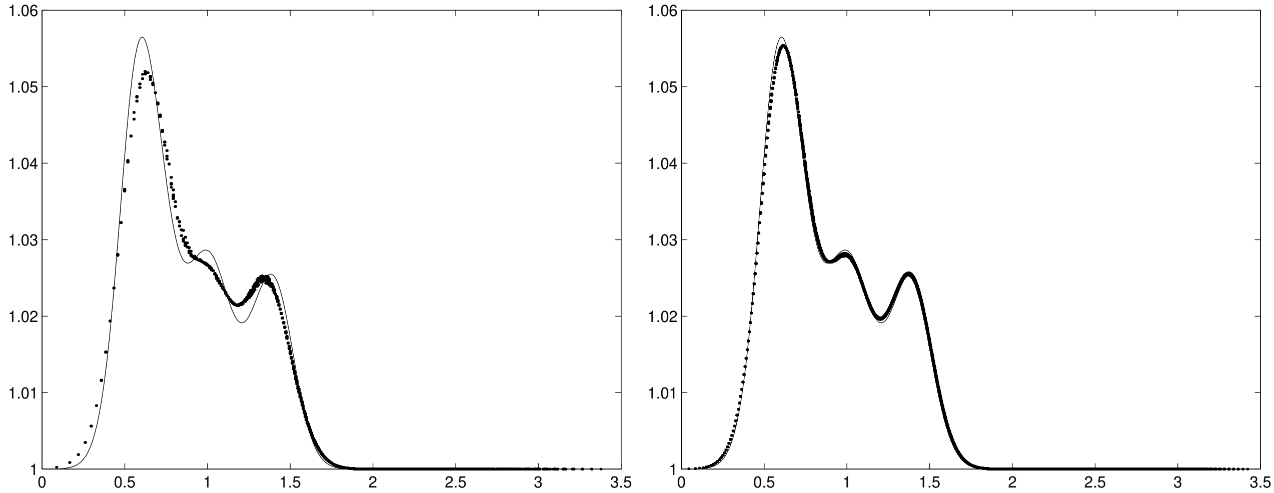


Figure 11: Scatter plots for the Euler equation with smooth solution, computed with Method (2,2,2). The density is depicted against the distance from origo. In the left picture, $N = 20$, while in the right picture $N = 40$. The solid curve corresponds to the reference solution.

6.2 A spherical Riemann problem between two walls

In this example we consider a spherical Riemann problem between two walls. The sphere is centered at $(0, 0, 0.4)$ with radius 0.2. The walls are placed at $z = 0$ and $z = 1$. Initially the gas is at rest. Inside the sphere $\rho_{in} = 1$ and $p_{in} = 5$, while outside $\rho_{out} = 1$ and $p_{out} = 1$. The initial condition results in a strong outward moving shock wave and contact discontinuity, and an inward moving rarefaction wave. This inward moving wave causes a local "implosion", and a second outward moving shock wave is created. Until the initial shock wave reaches the lower wall, the solution is spherically symmetric. After this, the solution will remain cylindrically symmetric. Hence, it is possible to formulate this as a two-dimensional problem with a source term, i.e.

$$\begin{bmatrix} \rho \\ \rho u \\ \rho w \\ E \end{bmatrix}_t + \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho u w \\ u(E+p) \end{bmatrix}_r + \begin{bmatrix} \rho w \\ \rho u w \\ \rho w^2 + p \\ w(E+p) \end{bmatrix}_z = -\frac{1}{r} \begin{bmatrix} \rho u \\ \rho u^2 \\ \rho u w \\ u(E+p) \end{bmatrix}, \tag{83}$$

where u now denotes the radial velocity. This equation is solved using `CLAWPACK` routines, and the results is used for comparison. In Figure 12 and Figure 13, the pressure is shown for a series of t-values. The two-dimensional problem is solved on a 600×400 grid. Due to the symmetry, the computational domain is $(r, z) = [0, 1.5] \times [0, 1]$, where $r = \sqrt{x^2 + y^2}$. However, in the figures, also the symmetric part is depicted. From the initial condition, a low pressure region is developed. Due to this, a spherical layer of the gas starts to move towards the center of the sphere. This creates a second shock wave moving outwards. The main features of the solution is the interactions between these shock waves and between these waves and the walls. At $t = 0.7$, Mach reflections are seen in the first shock. Another significant feature is the development of a near stationary low density region in the center of the domain. This region is clearly seen in Figure 14, which depicts the density at $t = 0.7$ using a schlieren type image. This picture emulates the photographic technique used in physical experiments. The magnitude of the density gradient is depicted. The larger the gradient, the darker is the region. Nonlinear shading is used

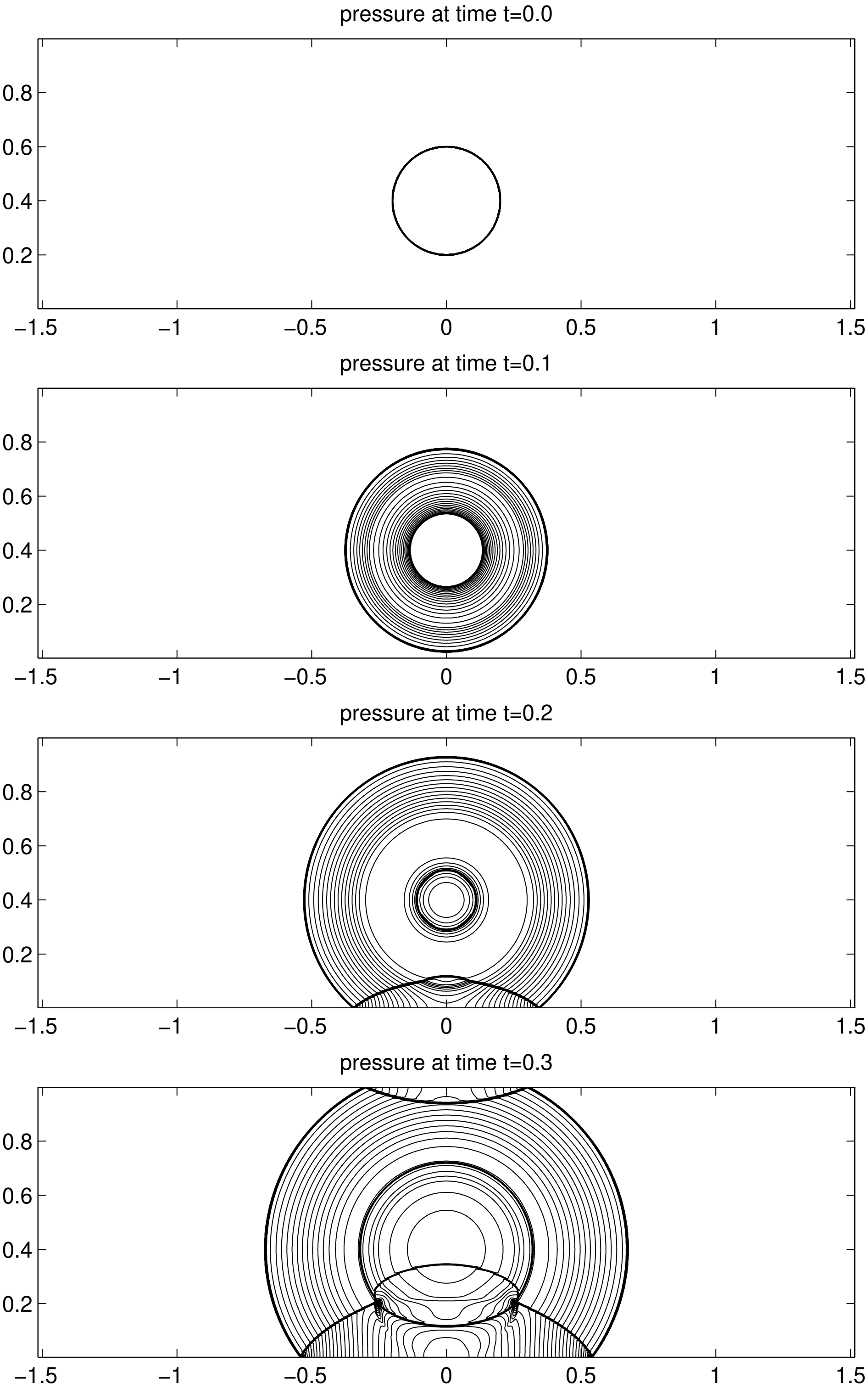


Figure 12: Results of a 2D calculation with cylindrical symmetry. A 600×400 grid is used in the computational domain $0 \leq r \leq 1.5$, $0 \leq z \leq 1$. The symmetric image is also plotted.

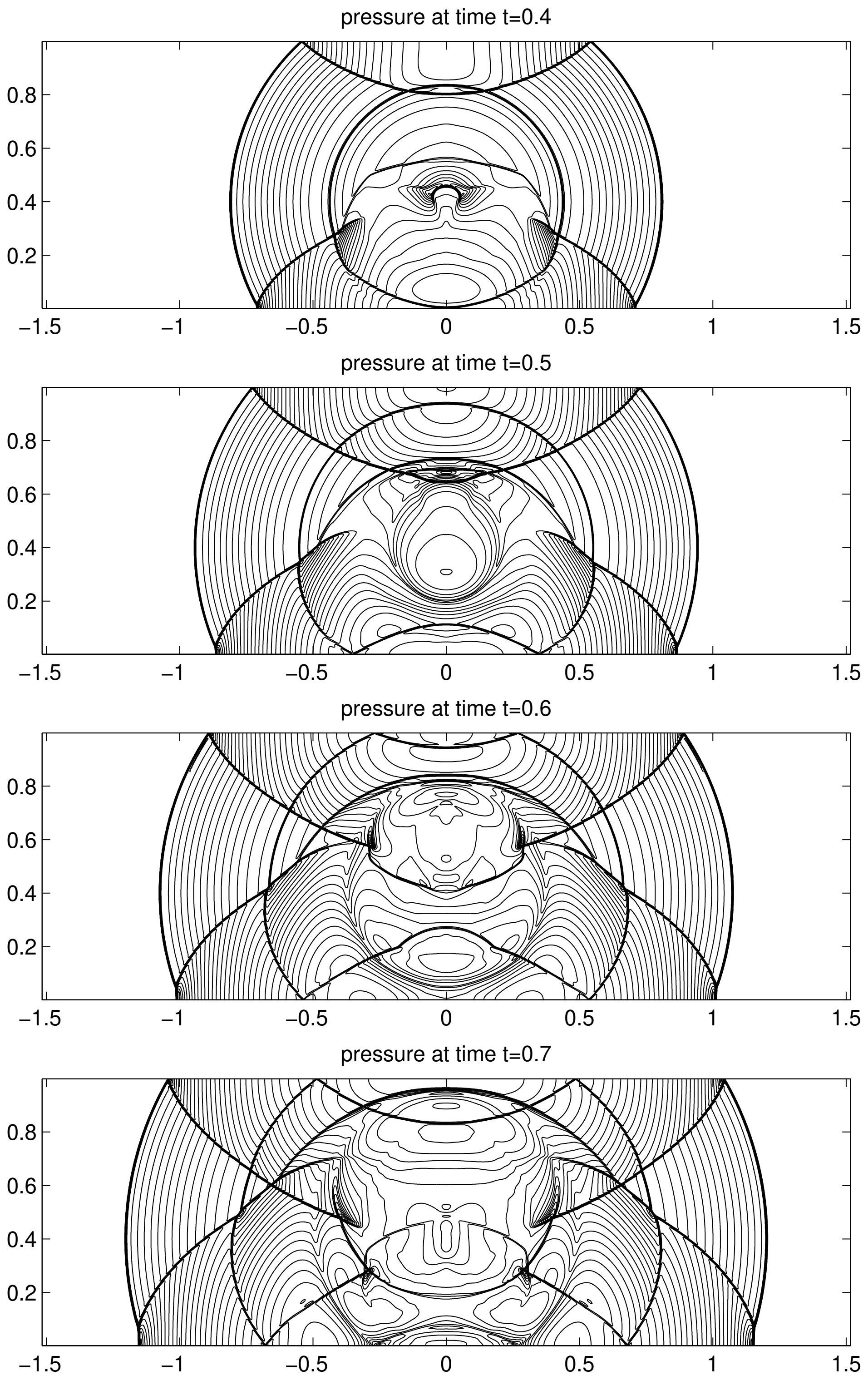


Figure 13: Results of a 2D calculation with cylindrical symmetry. The continuation of Figure 12.

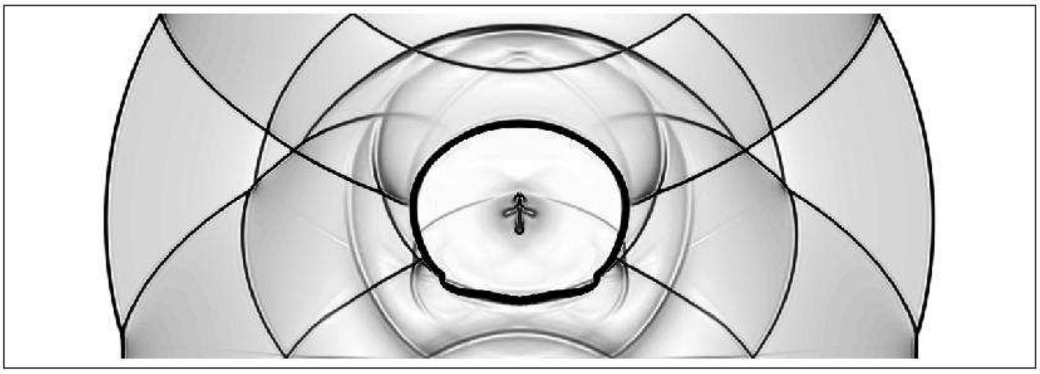


Figure 14: A schlieren type image showing the density at $t = 0.7$. The two-dimensional problem is solved using a 600×400 grid.

in order to enhance weak structures in the flow.

Due to the symmetry, the computational domain for the three-dimensional algorithm is $(x, y, z) \in [0, 1.5] \times [0, 1.5] \times [0, 1]$. In Figure 15, the pressure in the $x - z$ plane at $t = 0.7$ is shown using an increasing number of cells. The two dimensional results are also given for comparison. Another way of graphically indicating convergence is to consider scatter plots, like those used in the previous example. In Figure 16, the pressure is plotted against the distance from the z-axis. The plane used is $z = 0.4$ at $t = 0.7$. These pictures indicate that not only are the main features well resolved, but also the finer structures appear to be converging. In both the two-dimensional and the three-dimensional computations MC-limiters are used on all waves. In these computations the Courant number was approximately 0.9.

6.3 Shock-bubble interaction

In this example we consider the interaction between a planar shock and a spherical region with a density jump across the surface. This example has served as an illustration on the induced vorticity and mixing when a shock wave runs through a inhomogeneous media. Thorough numerical studies of problems of this type are found in [7] and [18]. In these papers, 2D problems involving a two-component gas are considered. Here we look at the 3D problem with a single gas.

The experimental set up is as follows, cf. Figure 17. The sphere is centered at (0.3,0,0) with radius 0.2. The gas has zero velocity and the pressure is unity in the whole domain initially. Inside the sphere, the density is 0.1, while outside the density is 1.0. The incoming shock wave propagates in the x-direction, and starts at $x = 0$. The pressure behind the shock is 10, yielding a 2.95 Mach shock.

As in the previous example, the problem is cylindrically symmetric, and it is possible to rewrite the problem into a 2D source term problem. Due to numerical viscosity, different grid sizes yield qualitative different results in the vortices. Hence it is not that meaningful to compute a reference solution using the 2D code in this case. Nevertheless, 2D `CLAWPACK` has been used to produce some high resolution results. In Figure 17, four numerical schlieren images are shown, resulting from a computation using a 1500×500 grid on the computational domain $[0, 1.5] \times [0, 0.5]$. Also the symmetric image is displayed.

When the inscident shock wave hits the bubble, it separates into a reflected smooth wave and a penetrating shock wave. Due to the higher sound speed inside the bubble, the latter wave will speed up, hit the bubble wall ahead, and reflect. At $t = 0.1$, this reflected wave is seen inside the

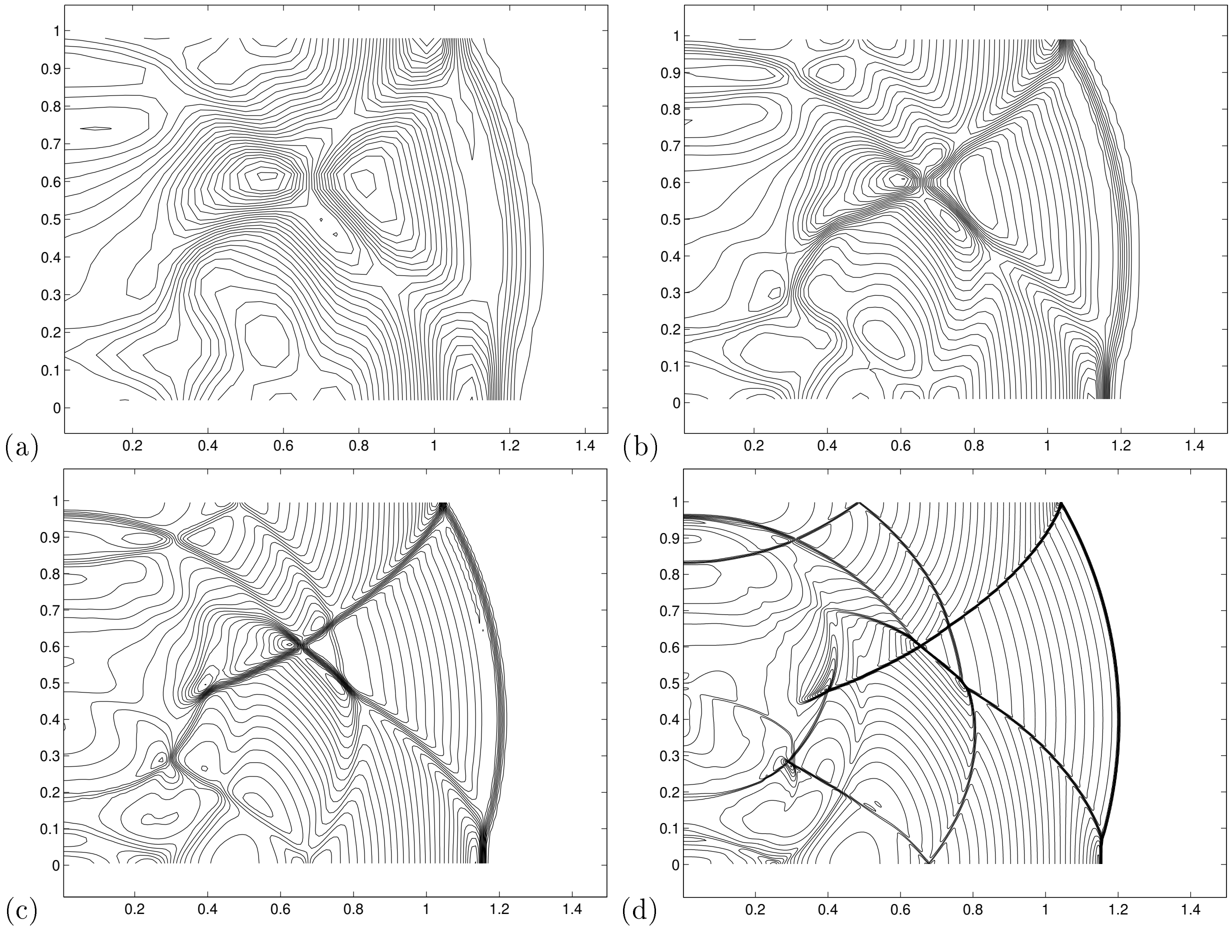


Figure 15: The pressure in the $x-z$ plane at $t = 0.7$ computed on different grids: (a) $37\times37\times25$, (b) $75 \times 75 \times 50$, (c) $150 \times 150 \times 100$. In (d) results from the corresponding 2D computation using a 600×400 grid is shown.

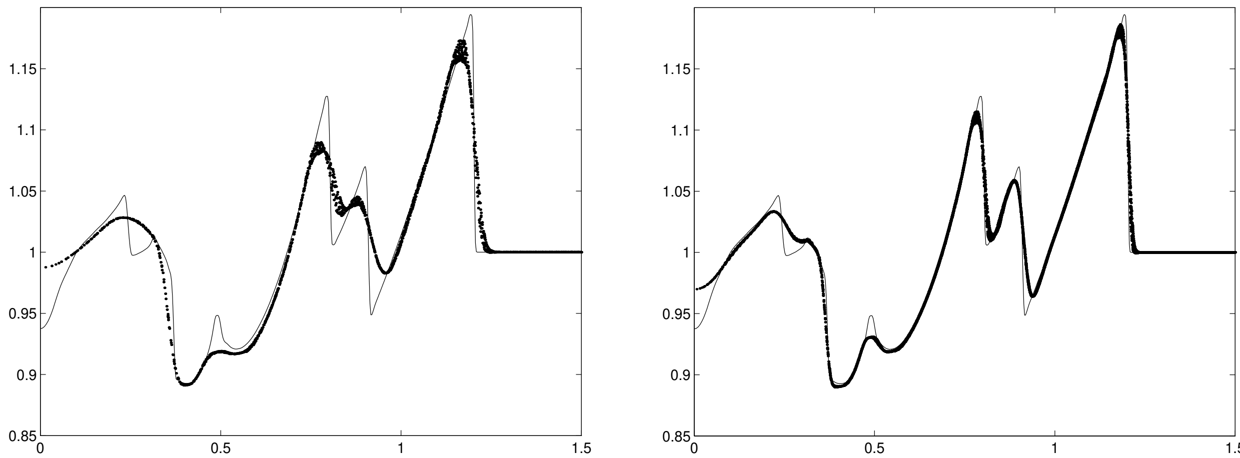


Figure 16: Scatter plots of the pressure versa the distance from the z-axis in the plane $z = 0.4$. In the left picture the computational grid is $75\times75\times50$, while in the right a $150\times150\times50$ grid is used. The solid curve corresponds to the accurate 2D computation.

deformed bubble. At the same time the first tendencies to vortex formation is seen at the back wall of the bubble. At this time, the inscident shock has also totally captured the bubble. At the "top" of the bubble, a complex pattern of discontinuities has formed. At $t = 0.2$, the bubble has collapsed, and the remnants are trapped inside a rotating ring (or a vortex tube). At later times, secondary vortices are seen on the rim of the main vortices.

For the 3D code, the symmetries in the problem are utilized, and the computational domain is $[0, 1.5] \times [0, 0.5] \times [0, 0.5]$. (The unstable nature of the vortices will eventually result in non-symmetries in the solution.) Symmetric boundary conditions are applied at the boundaries $y = 0$ and $z = 0$. At $x = 0$, inflow conditions are used. Open boundary conditions are used on the remaining boundaries. In this computation a $240 \times 80 \times 80$ grid is used. In the last of the small schlieren images in Figure 17, the solution for $y = 0$ is shown for $t = 0.4$. Naturally, the level of details is better in the 2D computation, but the main features of the flow are present.

Figure 18 shows a scatter plot comparing the 2D and the 3D computations. The density is plotted against the distance from the x-axis for $x = 0.6$ at $t = 0.2$. In order to avoid too much influence from the vortices, a relatively early stage of the computation is selected. The cross-section used goes through the center of the vortex. In Figure 18, the vortex surrounds the low density region. The main difference between the two solutions is that effects near the lower wall of the vortex is badly resolved in 3D. In 2D computations using the same grid size as in 3D, the same effects are present. Hence this is a resolution problem and does not indicate that there are major differences between the results given by the 2D and the 3D schemes.

The qualitative picture of the vortices is highly dependent on the numerical viscosity. As a consequence, different limiters produce different results. In these computations, the MC limiter is used.

6.4 Advection with non-constant velocities

Although this report does not focus on problems on non-conservative form, we include one such problem. The example is taken from [15], which deals with wave propagation schemes for advection in incompressible flow. The schemes considered in this report may be viewed as a generalization of the schemes presented in [15].

The problem to be considered is the scalar advection equation

$$\rho_t + u\rho_x + v\rho_y + w\rho_z = 0,$$

were the velocity field corresponds to a swirling deformation flow of the form

$$\begin{aligned} u(x,y,z) &= 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z)g(t) \\ v(x,y,z) &= -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z)g(t) \\ w(x,y,z) &= -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z)g(t). \end{aligned}$$

The time dependency $g(t)$ is given by

$$g(t) = \cos(\pi t/T).$$

The computational domain is the unit cube, and initially a discontinuity is placed at $x = 1/2$, i.e.

$$\rho(x,y,z,0) = \begin{cases} 1 & x \le 1/2 \\ 0 & x > 1/2, \end{cases}$$

Due to the velocity dependency, the flow reverses at $t = T/2$. At $T = t$, the initial discontinuity should be recovered. In the computation presented here $T = 1.5$, and a $80 \times 80 \times 80$ grid is used

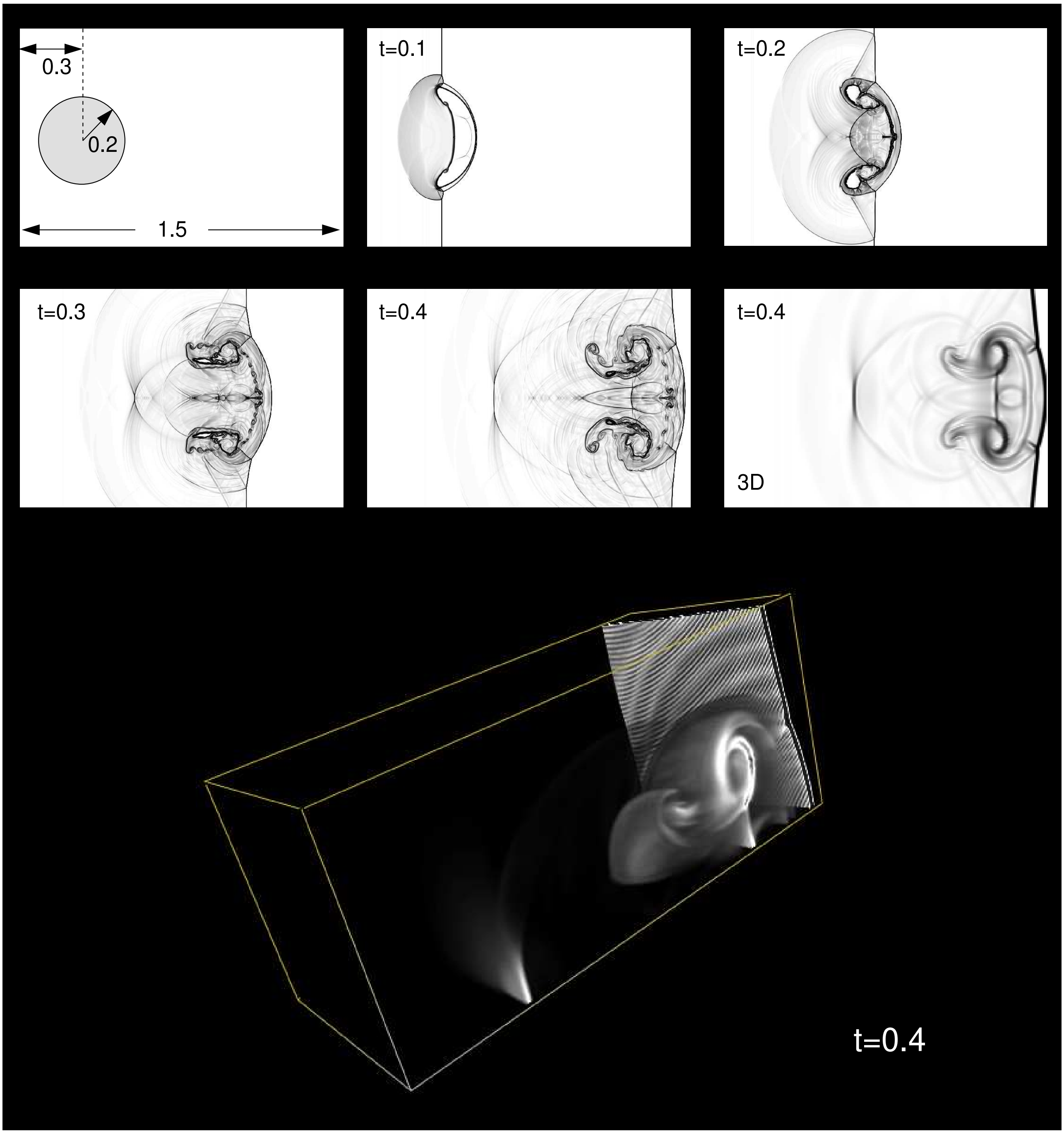


Figure 17: Computational results for the shock-bubble interaction in both 2D and 3D. The density gradients are shown in a schlieren type image.

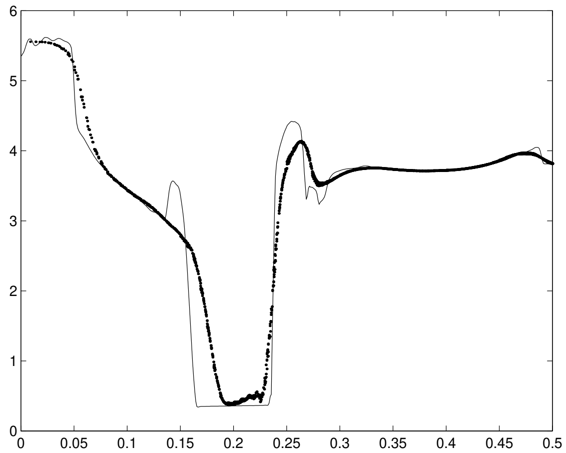


Figure 18: Scatter plot of the density versus the distance from the x-axis at $t = 0.2$ for the plane $x = 0.6$. A $240 \times 80 \times 80$ grid is used. The solid curve corrresponds to the 2D computation using a 1500×500 grid.

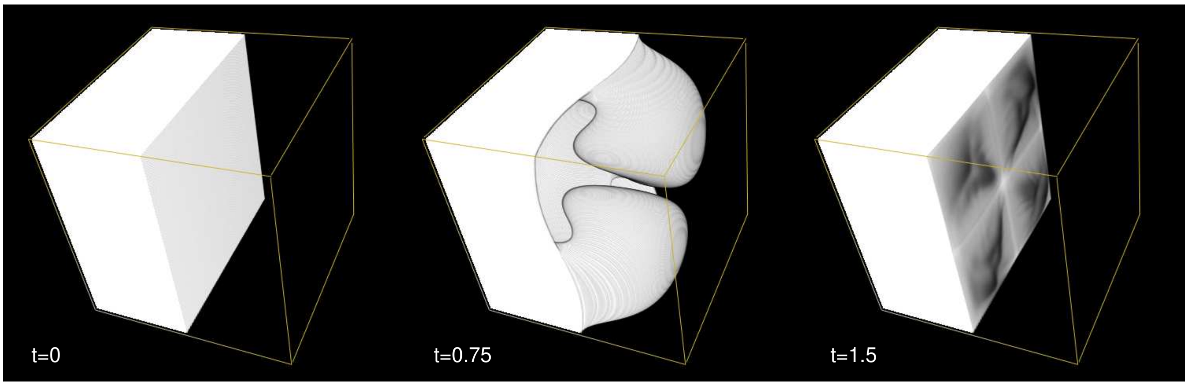


Figure 19: The solution of the scalar advection problem shown at three different time levels. The solution value one is non-transparent white, while the value zero is transparent.

along with the superbee limiter. In Figure 19, the initial condition is shown together with the solution at $t = 0.75$ and $t = 1.5$. The images show how the initial discontinuity get twisted, and returns to an approximation of the initial condition. In order to view the discontinuity, the value one is shown as non-transparent white, while the value zero is made transparent.

7 Concluding remarks

In this paper, a three-dimensional wave propagation method is constructed, and applied to the Euler equations. By using proper wave propagation, the derived schemes are stable provided the Courant number is less or equal to one. The 3D scheme generalizes the two-dimensional approach in [16]. The wave propagation method is implemented in Fortran, and will be included in the software package `CLAWPACK`, which is freely available on the Web.

Though the examples used in this paper mainly are written in conservative form, the scheme and the `CLAWPACK` software handle hyperbolic problems written in quasi-linear form $u_t + Au_x + Bu_y + Cu_z = 0$, where the matrices depend on u, x, y, z, and t. (Cf. the advection problem above.) See [16] for a discussion on how such problems are solved using `CLAWPACK`.

The 3D scheme will, in the near future, be generalized for solving the more general hyperbolic problem

$$\kappa(x,y,z)u_t + A(u,x,y,z,t)u_x + B(u,x,y,z,t)u_y + C(u,x,y,z,t)u_z = \psi(u,x,y,z,t).$$

The capacity function $\kappa = \kappa(x,y,z)$ may be used when solving the equation on a curvilinear grid. In this case κ equals the Jacobian for the grid mapping from the curvilinear grid to the rectangular computational grid, cf. [16]. Another example involving a capacity function is flow in porous media, where κ may represent porosity. The source term $\psi(u,x,y,z)$ will be included using fractional step methods.

The effect of applying different Riemann solvers are not addressed in this paper. Especially, the use of an exact Riemann solver should be studied. In two space dimensions, the number of Riemann problems solved at each interface is three. Hence, even if exact solvers are used in conjunction with the wave propagation method, the amount of work per interface is comparable to the work needed in the method of Colella[4]. However, in three dimensions, the number of Riemann problems per interface is 15 for the wave propagation scheme. This involves a significant amount of work, even if the approximate Roe solver is used. At present, the computational efficiency of this method is not compared to other approaches, but this will hopefully be done in future work. The possibility of reducing the number of Riemann problems per interface by lumping together appropriate terms should be studied.

Appendix

A The 3D algorithm and the `CLAWPACK` software

In the previous sections, the three-dimensional wave propagation scheme has been described in detail. However, a complete algorithm has not yet been presented. So far, the flux updates specifying the method has mostly been written in matrix notation, i.e. terms involving matrix multiplications. Computing these products explicitly is not in general an efficient way of implementing this scheme. Instead, the approach is used in which a wave group is split into waves moving in the transverse direction. In this section a complete algorithm is given in a form close to what is used in the implementation. Afterwards, the `CLAWPACK` routines used for realizing this are briefly discussed.

A.1 The 3D algorithm

In order to simplify the notation, the subscripts referring to space dependency are omitted. In addition, the "C syntax" $+=$ and $-=$ is used, i.e.

$$a \mathrel{+}= b, \quad a \mathrel{-}= b,$$

are used instead of

$$a = a + b, \quad a = a - b,$$

In Algorithm 2, one time step of the algorithm is shown. The integers m_1, m_2, and m_3 refer to the parameters introduced in Section 4.4.

The interfaces are considered coordinatewise, starting with interfaces in the x-direction. Below, the numbered parts in the algorithm are briefly discussed. So is the connection between the expressions here, and those in the previous sections.

The algorithm starts in step (i) by defining the left and right propagating flux differences $\mathscr{A}^{\pm}\Delta_x U_{ijk}$, the waves $\mathscr{W}^p$, and the associated velocities λ^p originating from the interface $x_{i-1/2}$. As mentioned in Section 2.3, there exist several ways of obtaining these quantities. A Roe solver or an exact solver may be used, but it is also possible to construct a hybrid version.

The upstream terms Δ^{up}_{ijk} associated with the two cells $C_{i-1,jk}$ and C_{ijk} are updated by these flux differences. In order to suppress spurious oscillations, the waves $\mathscr{W}^p$ may be limited. In step (ii), these possibly altered waves define $\mathscr{S}$. This term is used when implementing the motion of the correction wave. Next, the increment wave is propagated into cells sharing an interface with either $C_{i-1,jk}$ or C_{ijk}, i.e. the two-dimensional approach. In step (iii), this wave type is propagated in the y-direction. The updates of the $\widetilde{G}$-fluxes correspond to the updates in (34) and (35). Note that the flux differences rather than the increment waves are split into waves moving in the transverse direction, as discussed in Section 3.1. In a similar way, the increment waves are propagated in the z-direction, step (iv). The resulting updates equal those given in (59).

So far, methods "up to" (2,1,0) are described. As discussed in Section 5, this scheme is unconditionally unstable. In order to restore the stability, the correction wave is also propagated in the transverse direction. In step (v) and step (vi), these waves are propagated in the two-dimensional sense, as done for the increment wave in step (iii) and step (iv). Note the similarities in the expressions in step (iii-iv) and steps (v-vi). Both the increment waves and the correction waves are propagated in same way. In 2D, cf. Algorithm 1, these two sets of updates are added together, saving one Riemann problem. The same strategy could easily have been used also in the 3D case, but the advantage is greater if other terms are lumped together.

In order to add truly three-dimensional propagation of the increment wave, these waves also have to move into cells only sharing an edge with either $C_{i-1,jk}$ and C_{ijk}. This involves approximating terms like $CBAu_{xyz}$ in a Taylor expansion. The appropriate updates using the matrix notation are found in Section 4.2. An efficient way of implementing this wave motion is to split $BA\Delta_x U_{ijk}$ into waves moving in the z-direction, cf. (61). These waves are then used in updating the $\widetilde{H}$-fluxes. The splitting of the term $\mathscr{B}\mathscr{A}\Delta_x U_{ijk}$ in the z-direction is given is step (viii). The notation using $\pm *$, where $*$ equals $+$ or $-$ as subscripts, does not refer to the positive or negative part of real numbers, but are used as shorthand notation referring to the sign of the velocities involved. For example, the coefficients η^l_{-+} are used in the splitting of the term $\mathscr{B}^-\mathscr{A}^+\Delta_x U_{ijk}$, i.e. negative velocities in y and positive velocities in x.

The 3D motion of the correction wave causes updates similar to those caused by the increment wave. Hence, it is advantageous to add these waves, saving the solution of four Riemann problems. The inclusion of 3D propagation of the correction wave involves splitting the term $\mathscr{B}\mathscr{S}$ in the z-direction. This is easily obtained by using the expansion computed in step (iv). In step (vii), $\mathscr{B}\mathscr{A}\Delta_x U_{ijk}$ is updated to account for this inclusion. These definitions correspond to those given in (68). The updates of the $\widetilde{H}$-flux in step (viii) correspond to those in (60), and approximates $CBAu_{xyz}$ and $CB|A|(1-\frac{\Delta t}{\Delta x}|A|)u_{xxyz}$, depending on m_3, in the linear case.. In steps (ix) and (x), similar approximations of $BCAu_{xzy}$ and $BC|A|(1-\frac{\Delta t}{\Delta x}|A|)u_{xxzy}$ are given.

Up to this point, only the waves emanating from interfaces in the x-direction have been considered. The associated updates account for approximations of derivatives differentiated last in x, for example Au_x, BAu_{xy}, $CBAu_{xyz}$, in the linear case. Step (xi) considers the updates caused by waves emanating from interfaces in the y-direction. Naturally, the corresponding updates are similar to those considered in steps (i-x). Hence with appropriate changes, the same steps are used. The correct coordinate change is $(x,y,z) \to (y,z,x)$.

In step (xi), the updates caused by waves emanating from interfaces in the z-direction are considered. In this case, the appropriate coordinate change is $(x,y,z) \to (z,x,y)$. An efficient implementation of this algorithm should take advantage of the similarities in the updates in the three coordinate directions, i.e. (i-x), (xi), and (xii). It is only necessary to write a single routine that treats one coordinate direction. The two remaining directions are covered by changing the arguments. Finally, in step (xiii), the solution U_{ijk} is advanced to the next time level.

A.2 The `CLAWPACK` software

The general structure of the `CLAWPACK` code is described in the user notes [17] and in the code itself. Here we briefly consider the routines the user needs to supply, and how Algorithm 2 is related to the code.

First, the user needs to write a driver routine which initializes the grid and calls `CLAWPACK` using the routine `claw3`. Next, a routine that adds the desired boundary conditions is needed. This is done as described in Section 4.5, by assigning appropriate values to the "ghost cells" outside the computational grid. A routine implementing Algorithm 2 advances the solution for one time step, and is called from `claw3`.

The user needs to supply the routines that solve the Riemann problems (or wave splitting). The routine `rpn3(inor,...)` solves the Riemann problem in the direction normal to the cell interfaces. If `inor` = 1,2, and 3, the normal directions are in x, y, and z, respectively. When `inor` = 1, the output from this routine is the flux splitting $\mathscr{A}^{\pm}\Delta_x U$, the waves $\mathscr{W}^p$ and the velocities λ^p. Step (i) in the algorithm, is obtained by calling `rpn3(1,...)`.

For the transverse wave propagation, the user needs to provide the routine `rpt3(inor,itran,...)`. The argument `inor` has the same meaning as for `rpn3`. The second argument `itran` specifies in which of the two remaining coordinate directions the wave is to be split. If `itran` = 2 and 3, the splitting is performed in the second and third coordinate direction in the specific coordinate system. If for example `inor` = 1, `itran` = 3, and the term (or vector in this context) to be split is $\mathscr{B}^+\mathscr{A}^+\Delta_x U_{ijk}$, the output is the terms $\mathscr{C}^-\mathscr{B}^+\mathscr{A}^+\Delta_x U_{ijk}$ and $\mathscr{C}^+\mathscr{B}^+\mathscr{A}^+\Delta_x U_{ijk}$. In the linear case the output from this call is used in approximating $CBAu_{xyz}$, and in addition $CB|A|(1 - \frac{\Delta t}{\Delta x}|A|)u_{xxyz}$ if $m_3 = 2$. Another example; if `inor` = `itran` = 3, and the input is $\mathscr{C}^+\Delta_x U$, the output is $\mathscr{B}^-\mathscr{C}^+\Delta_x U$ and $\mathscr{B}^+\mathscr{C}^+\Delta_x U$.

According to this, the splitting in the steps (iii), (v), and (x) are all performed by calling `rpt3(1,2,...)`, of course with different arguments. Similarly, the splitting in the steps (iv), (vi), and (viii) are achieved by calling `rpt3(1,3,...)`.

The `rpn3` routine is called once for every interface, while `rpt3` is called 14 times for the full wave propagation method. However, for a nonlinear problem using the Roe solver, `rpn3` involves more computational work than `rpt3`, since the former routine computes the Roe averaged quantities and applies the entropy fix. For systems occurring in gas dynamics, there exist explicit formulas for solving the involved linear system, cf. [21]. Hence, the computations in `rpt3` consists of simple evaluations of these expressions.

Algorithm 2: One time step of the algorithm for a general three-dimensional problem.

initialize fluxes:

for each i, j, k do

$\widetilde{F}_{ijk} := 0, \quad \widetilde{G}_{ijk} := 0, \quad \widetilde{H}_{ijk} := 0, \quad \Delta^{up}_{ijk} := 0$

update fluxes based on interfaces in the x-direction:

for each interface $x = x_{i-1/2}$ do

(i) <Compute flux difference splitting $\mathscr{A}^{\pm}\Delta_x U_{ijk}$, wave $\mathscr{W}^p$ and velocities λ^p>

$\Delta^{up}_{ijk} -= \frac{\Delta t}{\Delta x}\mathscr{A}^{+}\Delta_x U_{ijk}, \qquad \Delta^{up}_{i-1,jk} -= \frac{\Delta t}{\Delta x}\mathscr{A}^{-}\Delta_x U_{ijk}$

<Apply limiter to $\mathscr{W}^p$ to obtain $\widetilde{\mathscr{W}}^p$>

(ii) # Add correction wave ($m_1 > 1$)

$\mathscr{S} := \sum |\lambda^p|(1 - \frac{\Delta t}{\Delta x}|\lambda^p|)\widetilde{\mathscr{W}}^p$

$\widetilde{F}_{ijk} += \frac{1}{2}\mathscr{S}$

(iii) # Split $\mathscr{A}^{\pm}\Delta_x U_{ijk}$ into waves w^q propagating in y-direction

with velocities μ^q ($m_2 > 0$):

$\mathscr{A}^{+}\Delta_x U_{ijk} = \sum(\delta^q)^{+}w^q, \quad \mathscr{A}^{-}\Delta_x U_{ijk} = \sum(\delta^q)^{-}w^q$

Form the terms

$\mathscr{B}^{\pm}\mathscr{A}^{+}\Delta_x U_{ijk} := \sum(\mu^q)^{\pm}(\delta^q)^{+}w^q, \quad \mathscr{B}^{\pm}\mathscr{A}^{-}\Delta_x U_{ijk} := \sum(\mu^q)^{\pm}(\delta^q)^{-}w^q$

Use these terms to update neighboring $\widetilde{G}$ fluxes (as in 2D):

$\widetilde{G}_{i-1,jk} -= \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{B}^{-}\mathscr{A}^{-}\Delta_x U_{ijk}$

$\widetilde{G}_{ijk} -= \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{B}^{-}\mathscr{A}^{+}\Delta_x U_{ijk}$

$\widetilde{G}_{i,j+1,k} -= \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{B}^{+}\mathscr{A}^{+}\Delta_x U_{ijk}$

$\widetilde{G}_{i-1,j+1,k} -= \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{B}^{+}\mathscr{A}^{-}\Delta_x U_{ijk}$

(iv) # Split $\mathscr{A}^{\pm}\Delta_x U_{ijk}$ into waves s^l propagating in z-direction

with velocities ν^l ($m_2 > 0$):

$\mathscr{A}^{+}\Delta_x U_{ijk} = \sum(\varepsilon^l)^{+}s^l, \quad \mathscr{A}^{-}\Delta_x U_{ijk} = \sum(\varepsilon^l)^{-}s^l$

Form the terms

$\mathscr{C}^{\pm}\mathscr{A}^{+}\Delta_x U_{ijk} := \sum(\nu^l)^{\pm}(\varepsilon^l)^{+}s^l, \quad \mathscr{C}^{\pm}\mathscr{A}^{-}\Delta_x U_{ijk} := \sum(\nu^l)^{\pm}(\varepsilon^l)^{-}s^l$

Use these terms to update neighboring $\widetilde{H}$ fluxes:

$\widetilde{H}_{i-1,jk} -= \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{C}^{-}\mathscr{A}^{-}\Delta_x U_{ijk}$

$\widetilde{H}_{ijk} -= \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{C}^{-}\mathscr{A}^{+}\Delta_x U_{ijk}$

$\widetilde{H}_{ij,k+1} -= \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{C}^{+}\mathscr{A}^{+}\Delta_x U_{ijk}$

$\widetilde{H}_{i-1,j,k+1} -= \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{C}^{+}\mathscr{A}^{-}\Delta_x U_{ijk}$

(v) # Split the correction wave into waves propagating in y-direction ($m_2 = 2$):

$$\mathscr{S} = \sum \widetilde{\delta}^q \, w^q$$

Form the terms

$$\mathscr{B}^{\pm}\mathscr{S} := \sum (\mu^q)^{\pm} \, \widetilde{\delta}^q \, w^q$$

Use these terms to update neighboring $\widetilde{G}$ fluxes (as in 2D):

$$\widetilde{G}_{i-1,jk} \mathrel{-}= \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{B}^{-}\mathscr{S}, \qquad \widetilde{G}_{ijk} \mathrel{+}= \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{B}^{-}\mathscr{S}$$

$$\widetilde{G}_{i-1,j+1,k} \mathrel{-}= \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{B}^{+}\mathscr{S}, \qquad \widetilde{G}_{i,j+1,k} \mathrel{+}= \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{B}^{+}\mathscr{S}$$

(vi) # Split the correction wave into waves propagating in z-direction ($m_2 = 2$):

$$\mathscr{S} = \sum \widetilde{\varepsilon}^{\,l} \, s^l$$

Form the terms

$$\mathscr{C}^{\pm}\mathscr{S} = \sum (\nu^l)^{\pm} \, \widetilde{\varepsilon}^{\,l} \, s^l$$

Use these terms to update neighboring $\widetilde{H}$ fluxes:

$$\widetilde{H}_{i-1,jk} \mathrel{-}= \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{C}^{-}\mathscr{S}, \qquad \widetilde{H}_{ijk} \mathrel{+}= \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{C}^{-}\mathscr{S}$$

$$\widetilde{H}_{i-1,j,k+1} \mathrel{-}= \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{C}^{+}\mathscr{S}, \qquad \widetilde{H}_{ij,k+1} \mathrel{+}= \frac{1}{2}\frac{\Delta t}{\Delta x}\mathscr{C}^{+}\mathscr{S}$$

(vii) <u>if</u> $m_3 = 2$ <u>then</u>

$$\mathscr{B}^{\pm}\mathscr{A}^{+}\Delta_x U_{ijk} \mathrel{-}= 3\mathscr{B}^{\pm}\mathscr{S}, \quad \mathscr{B}^{\pm}\mathscr{A}^{-}\Delta_x U_{ijk} \mathrel{+}= 3\mathscr{B}^{\pm}\mathscr{S}$$

(viii) # Split these terms into waves propagating in the z-direction: ($m_3 > 0$)

$$\mathscr{B}^{\pm}\mathscr{A}^{+} = \sum \eta^l_{\pm+} \, s^l, \quad \mathscr{B}^{\pm}\mathscr{A}^{-} = \sum \eta^l_{\pm-} \, s^l$$

Form the terms

$$\mathscr{C}^{\pm}\mathscr{B}^{+}\mathscr{A}^{+}\Delta_x U_{ijk} = \sum (\nu^l)^{\pm} \, \eta^l_{++} s^l, \qquad \mathscr{C}^{\pm}\mathscr{B}^{+}\mathscr{A}^{-}\Delta_x U_{ijk} = \sum (\nu^l)^{\pm} \, \eta^l_{+-} s^l$$

$$\mathscr{C}^{\pm}\mathscr{B}^{-}\mathscr{A}^{+}\Delta_x U_{ijk} = \sum (\nu^l)^{\pm} \, \eta^l_{-+} s^l, \qquad \mathscr{C}^{\pm}\mathscr{B}^{-}\mathscr{A}^{-}\Delta_x U_{ijk} = \sum (\nu^l)^{\pm} \, \eta^l_{--} s^l$$

Use these terms to update neighboring $\widetilde{H}$ fluxes:

$$\widetilde{H}_{ijk} \mathrel{+}= \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y}\mathscr{C}^{-}(\mathscr{B}^{+} - \mathscr{B}^{-})\mathscr{A}^{+}\Delta_x U_{ijk}$$

$$\widetilde{H}_{i,j+1,k} \mathrel{-}= \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y}\mathscr{C}^{-}\mathscr{B}^{+}\mathscr{A}^{+}\Delta_x U_{ijk}$$

$$\widetilde{H}_{i,j-1,k} \mathrel{+}= \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y}\mathscr{C}^{-}\mathscr{B}^{-}\mathscr{A}^{+}\Delta_x U_{ijk}$$

$$\widetilde{H}_{ij,k+1} \mathrel{+}= \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y}\mathscr{C}^{+}(\mathscr{B}^{+} - \mathscr{B}^{-})\mathscr{A}^{+}\Delta_x U_{ijk}$$

$$\widetilde{H}_{i,j+1,k+1} \mathrel{-}= \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y}\mathscr{C}^{+}\mathscr{B}^{+}\mathscr{A}^{+}\Delta_x U_{ijk}$$

$$\widetilde{H}_{i,j-1,k+1} \mathrel{+}= \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta y}\mathscr{C}^{+}\mathscr{B}^{-}\mathscr{A}^{+}\Delta_x U_{ijk}$$

Six similar updates where i is replaced by $i-1$ and A^{+} is replaced by A^{-}.

(ix) if $m_3 = 2$ then

$$\mathscr{C}^\pm \mathscr{A}^+ \Delta_x U_{ijk} \mathrel{-}= 3\mathscr{C}^\pm \mathscr{S}, \quad \mathscr{C}^\pm \mathscr{A}^- \Delta_x U_{ijk} \mathrel{+}= 3\mathscr{C}^\pm \mathscr{S}$$

(x) # Split these waves into waves propagating in the y-direction ($m_3 > 0$):

$$C^\pm A^+ \Delta_x U_{ijk} = \sum \sigma^q_{\pm +}\, w^q, \quad C^\pm A^- \Delta_x U_{ijk} = \sum \sigma^q_{\pm -}\, w^q$$

Form the terms

$$\mathscr{B}^\pm \mathscr{C}^+ \mathscr{A}^+ \Delta_x U_{ijk} = \sum (\mu^q)^\pm\, \sigma^q_{++} w^q, \qquad \mathscr{B}^\pm \mathscr{C}^+ \mathscr{A}^- \Delta_x U_{ijk} = \sum (\mu^q)^\pm\, \sigma^q_{+-} w^q$$

$$\mathscr{B}^\pm \mathscr{C}^- \mathscr{A}^+ \Delta_x U_{ijk} = \sum (\mu^q)^\pm\, \sigma^q_{-+} w^q, \qquad \mathscr{B}^\pm \mathscr{C}^- \mathscr{A}^- \Delta_x U_{ijk} = \sum (\mu^q)^\pm\, \sigma^q_{--} w^q$$

Use these terms to update neighboring $\widetilde{G}$ fluxes:

$$\widetilde{G}_{ijk} \mathrel{+}= \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta z}\mathscr{B}^-(\mathscr{C}^+ - \mathscr{C}^-)\mathscr{A}^+ \Delta_x U_{ijk}$$

$$\widetilde{G}_{ij,k+1} \mathrel{-}= \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta z}\mathscr{B}^- \mathscr{C}^+ \mathscr{A}^+ \Delta_x U_{ijk}$$

$$\widetilde{G}_{ij,k-1} \mathrel{+}= \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta z}\mathscr{B}^- \mathscr{C}^- \mathscr{A}^+ \Delta_x U_{ijk}$$

$$\widetilde{G}_{i,j+1,k} \mathrel{+}= \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta z}\mathscr{B}^+(\mathscr{C}^+ - \mathscr{C}^-)\mathscr{A}^+ \Delta_x U_{ijk}$$

$$\widetilde{G}_{i,j+1,k+1} \mathrel{-}= \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta z}\mathscr{B}^+ \mathscr{C}^+ \mathscr{A}^\Delta_x U_{ijk}$$

$$\widetilde{G}_{i,j+1,k-1} \mathrel{+}= \frac{1}{6}\frac{\Delta t}{\Delta x}\frac{\Delta t}{\Delta z}\mathscr{B}^+ \mathscr{C}^- \mathscr{A}^+ \Delta_x U_{ijk}$$

Six similar updates where i is replaced by $i-1$ and A^+ is replaced by A^-.

(xi) # Update fluxes based on interfaces in the y-direction:

Similar to the above with the changes $(i,j,k) \to (j,k,i)$, $(\widetilde{F}, \widetilde{G}, \widetilde{H}) \to (\widetilde{G}, \widetilde{H}, \widetilde{F})$,

$\mathscr{A}^\pm \to \mathscr{B}^\pm$, and $(\Delta x, \Delta y, \Delta z) \to (\Delta y, \Delta z, \Delta x)$.

(xii) # Update fluxes based on interfaces in the z-direction:

Similar to the above with the changes $(i,j,k) \to (k,i,j)$, $(\widetilde{F}, \widetilde{G}, \widetilde{H}) \to (\widetilde{H}, \widetilde{F}, \widetilde{G})$,

$\mathscr{A}^\pm \to \mathscr{C}^\pm$, and $(\Delta x, \Delta y, \Delta z) \to (\Delta z, \Delta x, \Delta y)$.

(xiii) # Update U:

for each i, j, k do

$$\bar{U}_{ijk} = U_{ijk} + \Delta^{up}_{ik} - \frac{\Delta t}{\Delta x}(\widetilde{F}_{i+1,jk} - \widetilde{F}_{ijk}) - \frac{\Delta t}{\Delta y}(\widetilde{G}_{i,j+1,k} - \widetilde{G}_{ijk}) - \frac{\Delta t}{\Delta z}(\widetilde{H}_{ij,k+1} - \widetilde{H}_{ijk})$$

References

[1] J. B. Bell, C. N. Dawson, and G. R. Shubin, *An unsplit, higher order Godunov method for scalar conservation laws in multiple dimensions*, J. Comp. Phys., 74 (1988), pp. 1-24.

[2] S. J. Billett, *A Class of Upwind Methods for Conservation Laws*, Ph.D. Thesis, Cranfield University, College of Aeronautics, 1994.

[3] J. P. Boris, D. L. Book, *Flux corrected transport I, SHASTA, a fluid transport algorithm that works*, J. Comput. Phys., 11 (1973), pp. 38-69.

[4] P. Colella, *Multidimensional upwind schemes for hyperbolic conservation laws*, J. Comput. Phys., 87 (1990), pp. 171-200.

[5] P. Colella, P. Woodward, *The piecewise-parabolic method (PPM) for gas-dynamical simulations*, J. Comput. Phys., 54 (1984), pp. 174-201.

[6] M. Crandall and A. Majda, *The method of fractional steps for conservation laws*, Numer. Math. 34 (1980), 285-314.

[7] Wai Sun Don, C. B. Quillen, *Numerical simulation of shock-cylinder interactions: I. Resolution*, J. Comp. Phys., 122 (1995), pp. 244-265.

[8] S. K. Godunov, *Finite difference methods for numerical computations of discontinuous solutions of equations of fluid dynamics*, Mat. Sb., 47 (1959), pp. 271-306 (in Russian).

[9] J. J. Gottlieb, P. T. Groth, *Assessment of Riemann solvers for unsteady one-dimensional inviscid flows of perfect gases* J. Comput. Phys. 78 (1988) pp. 437-458.

[10] J. O. Langseth, R. J. LeVeque, *Three-dimensional Euler computations using CLAWPACK*, in Conf. on Numer. Meth. for Euler and Navier-Stokes Eq., Montreal, P. Arminjon, ed., 1995, to appear.
(`ftp://amath.washington.edu/pub/rjl/papers/jlo-rjl:montreal95.ps.Z`)

[11] P. D. Lax and B. Wendroff, *Difference schemes for hyperbolic equations with high order of accuracy*, Comm. Pure Appl. Math., 18 (1964), pp. 381-398.

[12] B. P. Leonard, M. K. MacVean, and A. P. Lock, *Positivity-preserving numerical schemes for multidimensional advection.* NASA Technical Memorandum 106055, ICOMP-93-05, 1993.

[13] R. J. LeVeque, *Numerical Methods for Conservation Laws*, Birkhäuser Verlag, (1990).

[14] R. J. LeVeque, *Simplified multi-dimensional flux limiter methods*, in Numerical Methods for Fluid Dynamics 4. M. J. Baines and K. W. Morton, eds., Oxford University Press, 1993, pp. 175-190.

[15] R. J. LeVeque, *High-resolution conservative algorithms for advection in incompressible flow*, SIAM J. Numer. Anal., 33 (1996), pp. 627-665.

[16] R. J. LeVeque, *Wave propagation algorithms for multi-dimensional hyperbolic systems*, to appear in J. Comput. Phys, 1997.
(`ftp://amath.washington.edu/pub/rjl/papers/wpalg.ps.Z`)

[17] R. J. LeVeque, `CLAWPACK` User Notes. Available from `netlib.att.com` in `netlib/pdes/claw/doc` or on the Web at the URL `ftp://amath.washington.edu/pub/rjl/programs/clawpack.html.`

[18] J. J. Quirk, S. Karni, *On the dynamics of a shock-bubble interaction*, ICASE Report No. 94-75.

[19] Y. B. Radvogin, *Quasi-monotonous multidimensional differential schemes with second order accuracy.* Soviet Academy of Science Preprint (in Russian), 1991.

[20] R. D. Richtmyer, K. W. Morton, *Difference Methods for Initial-value Problems,* Wiley-Interscience, 1967.

[21] P. L. Roe, *Approximate Riemann solvers, parameter vectors, and difference schemes,* J. Comput. Phys., 43 (1981), pp. 357-372.

[22] P. L. Roe, *Discontinuous solutions to hyperbolic systems under operator splitting*, Num. Meth. for PDE, 7 (1991), pp. 277-297.

[23] P. L. Roe and D. Sidilkover, *Optimum positive linear schemes for advection in two and three dimensions*, J. Numer. Anal., 29 (1992), pp. 1542-1568.

[24] J. Saltzman, *Monotonic difference schemes for the linear advection equation in two and three dimensions.* Los Alamos Report LA-UR-87-2479, 1987

[25] J. Saltzman, *An unsplit 3-D upwind method for hyperbolic conservation laws*, J. Comput. Phys., 115 (1994), pp. 153-168.

[26] P. K. Smolarkiewicz and W. W. Grabowski, *The multidimensional positive definite advection transport algorithm: nonoscillatory option*, J. Comput. Phys., 86 (1990), pp. 355-375.

[27] J. Smoller, *Shock waves and Reaction-Diffusion Equations*, Springer, 1983.

[28] G. Strang, *On the construction and comparison of difference schemes*, SIAM J. Num. Anal., 5 (1968), pp. 506-517.

[29] J. C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, Wadsworth & Brooks/Cole, 1989.

[30] P. K. Sweby, *High resolution schemes using flux limiters for hyperbolic conservation laws*, SIAM J. Num. Anal., 21 (1984), pp. 995-1011.

[31] J. T. Trangenstein, *An unsplit Godunov method for three-dimensional polymer flooding*, preprint, 1993.

[32] B. van Leer, *Towards the ultimate conservative difference scheme II. Monotonicity and conservation combined in a second-order scheme*, J. Comput. Phys., 14 (1974), pp. 361-370.

[33] B. van Leer, *Towards the ultimate conservative difference scheme III. Upstream-centered finite-difference schemes for ideal compressible flow*, J. Comput. Phys., 23 (1977), pp. 263-275.

[34] B. van Leer, *Towards the ultimate conservative difference scheme IV. A new approach to numerical convection*, J. Comput. Phys., 23 (1977), pp. 276-299.

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

[36] S. T. Zalesak, *Fully multidimensional flux corrected transport algorithms for fluids*, J. Comput. Phys., 31 (1979), pp. 335-362.