High-Speed Compressible Flow and Gas Dynamics

Michael Sielemann
Deutsches Zentrum für Luft- und Raumfahrt, Robotics and Mechatronics Center,
System Dynamics and Control, Münchner Strasse 20, 82234 Wessling, Germany

Abstract

Discretization schemes suitable for gas dynamics are reviewed and applied to the declarative concepts of Modelica. Here, a suitable connector definition is introduced to enable both robust simulation and higher-order schemes, which require larger stencils than typically available on established thermo-fluid dynamics connectors.

Keywords: Finite volume method, shock waves, monotone flux, total variation diminishing, essentially non-oscillatory

1 Introduction

System-level simulation of thermo-fluid dynamics using Modelica is a wide topic yet relatively mature. Several authors present applications using the language in various technical domains. For instance, Casella [3, 4] considers power plant simulation, Pfafferott [20], Tummescheit et al. [36], Richter [24], and Prölß [21] study applications in sub-critical vapor compression cycles, Casas [2, 1] addresses air conditioning using desiccant assisted cycles, and Vasel and Schmitz [40] consider air conditioning using transcritical cycles.

In all of the given applications, the governing equations are adapted to the specifics of the underlying flow phenomena. With the exception of López [5], the assumptions are identical for all applications reported in literature. The corresponding flow, which allows to make these assumptions, is called a *low-speed compressible flow* herein. All authors referenced in the first paragraph assume that the flow is incompressible with respect to the flow phenomena, as it is low-speed. Density variation is only encountered due to heat transfer and in lumped parameter components such as compressors. Density variation due to flow phenomena is neglected, i.e., the Mach number is typically below 0.3.

In particular, an analysis of model code revealed that the difference between static and total pressure is neglected as the dynamic pressure is considered small and not of interest. For the given applications in power plants or vapor compression cycle refrigeration systems this is reasonable. Only in special devices, which involve large variations in flow cross-section such as adapters between different pipe diameters or nozzles, total pressure is of interest. Total or stagnation enthalpy is often treated similarly; the kinetic term $v^2/2$ is neglected. A typical argument is that the order of magnitude of change in specific enthalpy due to heat transfer is larger than that of such kinetic terms.

If kinetic terms in pressure and specific enthalpy are not neglected for such applications and the common assumption of a steady-state momentum balance is made then coupled nonlinear algebraic equation systems arise for density, which is required to establish flow velocity. These coupled equation systems deteriorate simulation performance.

Certain applications involve a different type of flow, which is called *high-speed compressible flow* herein. Kinetic terms and dynamic pressure may not be neglected and have to be included in compressible formulations. Density variation is also encountered with respect to flow phenomena, in particular dynamic conservation of momentum is relevant and also shock waves may be part of the solution. The Mach number may be > 0.3 (including the supersonic regime). The term “gas dynamics” refers to the same type of flow.

The key theoretical area to enable applications involving high-speed compressible flow is the discretization method for the governing equations. The foundations of numerical solution methods in thermo-fluid dynamics are well understood. However, in the framework of equation-based, object-oriented modeling languages, only methods suitable for low-speed compressible flow have been applied. The classic finite volume method has been studied in par-

DOI
10.3384/ecp1207681

ticular by Tummescheit [35]. Moving boundary methods have been applied by Jensen [14, 15] and Tummescheit [35]. Casella [4] proposed a mean density discretization, which is non-conservative but results in continuous and continuously differentiable thermodynamic properties at phase boundaries of two-phase flow. Prölß and Schmitz [22] discretized the governing equations for frost formation on heat exchanger surfaces.

López [5] proposed an approach to model and simulate gas dynamics. Due to robustness issues, which are certainly linked to deficiencies in the connector definition used in [5] (c.f. reference [7]), the approach did not become widely supported. In an attempt to finally extend the applicability of Modelica also to high-speed compressible flow and gas dynamics, this paper and reference [29] contribute to the state of the art in the following areas.

- Relevant concepts of the theory in numerical solution methods for high-speed compressible flow are reviewed and translated from the algorithmic perspective taken in literature to the acausal concepts of equation-based, object-oriented modeling languages.

- The elements of discretization schemes are decomposed in an object-oriented fashion and implemented in a generic library. Object-oriented concepts are exploited for increased flexibility such as parametric polymorphism for exchangeable thermodynamic property models.

2 The governing equations in compact flux form

To address high-speed compressible flow, a compact flux formulation as described by Toro [34] is considered. It is posed using conserved variables u and flux f.

$$u_t(x,t) + f\left(u(x,t)\right)_x = s\left(u(x,t)\right) \tag{1}$$

$$u(x,t) = \begin{pmatrix} \rho \\ \rho v \\ \rho u_0 \end{pmatrix} \tag{2}$$

$$f\left(u(x,t)\right) = \begin{pmatrix} \rho v \\ \rho v^2 + p \\ v\left(\rho u_0 + p\right) \end{pmatrix} \tag{3}$$

If the cross-sectional area A is supposed to vary smoothly with time and position, then the following source term including heat transfer and viscous wall friction can be used [34].

$$s\left(u(x,t)\right) = \begin{pmatrix} 0 \\ \Delta p_{fr} \\ \rho \dot{q}_e \end{pmatrix} - \frac{1}{A}\frac{dA}{dt}\begin{pmatrix} \rho \\ \rho v \\ \rho u_0 + p \end{pmatrix} \tag{4}$$

3 Conservative methods

An approach to discretize the governing equations of thermo-fluid dynamics is now introduced based on Toro [34]. It is formulated in conserved variables and therefore called a conservative method.

The use of conservative methods is motivated by the presence of discontinuities such as shock waves in the solution of certain problems such as gas dynamics. Hou and LeFloch [13] have shown that formulations based on variables other than the conserved ones fail to correctly predict the solution at shock waves. They result in wrong jump conditions and thus wrong shock strength, speed, and location. The theorem of Lax and Wendroff [17] in turn states that conservative methods, if convergent, do converge to the weak solution of the conservation law. Consequently, conservative methods are an obvious choice if shock waves are potentially contained in the solution.

In this section, the compact formulation of the conservation laws introduced in equation (1) is used. The vector of conserved quantities is denoted by $u(x,t) = (\rho, \rho v, \rho u_0)$. In order to include weak solutions of (1), an integral form of the equations is used, a finite volume method.

As done in several numerical methods, the problem domain is discretized on a suitable computational mesh. The control volumes are defined based on a grid of cell side coordinates on an interval $[a,b]$

$$a = x_{1/2} < x_{3/2} < \ldots < x_{n-1/2} < x_{n+1/2} = b \tag{5}$$

Based on it, cells, cell centers and cell sizes are defined for $i = 1, 2, \ldots, n$.

$$\begin{aligned} I_i &= \left[x_{i-1/2}, x_{i+1/2}\right] \\ x_i &= \tfrac{1}{2}\left(x_{i-1/2} + x_{i+1/2}\right) \\ \Delta x_i &= x_{i+1/2} - x_{i-1/2} \end{aligned} \tag{6}$$

In this notation, $x_{i+1/2}$ is the coordinate of the right side of a computational cell I_i with cell center x_i. This grid is colocated. Furthermore, the maximum cell size is defined as follows.

$$\Delta x = \max_{1 \leqslant i \leqslant n}\left(\Delta x_i\right) \tag{7}$$

DOI
10.3384/ecp1207681

The discretization scheme allows to deduce algebraic equations or differential algebraic equations that properly approximate the governing equations. Note that, in the context of Modelica, the goal is to deduce differential algebraic equations and thus the equations have only to be discretized in space, not in time ("semi-discretized").

The set of cell centers, which is used in a discretization scheme to deduce such equations for each cell, is called the stencil. For the most simple schemes, the stencil for cell I_i includes I_i itself and the cells to the left and to the right,

$$S(i) = \{I_{i-1}, I_i, I_{i+1}\} \tag{8}$$

Therefore, equation (1) is integrated over the interval I_i to obtain

$$\begin{aligned}\frac{d\bar{u}(x_i,t)}{dt} =& s(\bar{u}(x_i,t)) - \\ & \frac{1}{\Delta x_i}\left(f\left(u\left(x_{i+1/2},t\right)\right) - f\left(u\left(x_{i-1/2},t\right)\right)\right)\end{aligned} \tag{9}$$

Herein, a cell average is used

$$\bar{u}(x_i,t) = \frac{1}{\Delta x_i}\int_{x_{i-1/2}}^{x_{i+1/2}} u(\xi,t)\,d\xi$$

Equation (9) is approximated by a semi-discretized conservative scheme, which results in a differential algebraic equation,

$$\frac{d\overline{u_i}(t)}{dt} = s(\overline{u_i}(t)) - \frac{1}{\Delta x_i}\left(f_{i+1/2} - f_{i-1/2}\right) \tag{10}$$

Herein, $\overline{u_i}(t)$ is a numerical approximation of the exact cell average $\bar{u}(x_i,t)$, and $f_{i\pm1/2}$ is a numerical flux, an approximation of the physical flux $f\left(u\left(x_{i\pm1/2},t\right)\right)$.

The remainder of this section is concerned with the construction of numerical fluxes. All these fluxes consist of a monotone flux and a reconstruction. Practically, a monotone flux is a flux free of spurious oscillations. Due to Godunov's Theorem such linear fluxes are however first-order accurate only. Therefore, these monotone fluxes are often used together with reconstructions in order to build higher-order schemes. The reconstruction provides an approximation of the vector of conserved variables u (or any other variable of interest) based on the cell averages. Its higher-order accuracy yields, together with a first-order monotone flux, higher-order numerical flux.

3.1 Monotone flux and first-order schemes

A monotone numerical flux is defined using a function g,

$$f_{i+1/2} = g\left(u^-_{i+1/2}, u^+_{i+1/2}\right) \tag{11}$$

Here, $u^-_{i+1/2}$ is in general an approximation of the vector of conserved variables at $x_{i+1/2}$ in the left limit, and $u^+_{i+1/2}$ in the right limit. Each monotone flux can be used without reconstruction with the approximation $u^-_{i+1/2} \approx u_i$ and $u^+_{i+1/2} \approx u_{i+1}$. The results are first-order schemes. Alternatively, any more sophisticated approach may be used to reconstruct $u^\pm_{i+1/2}$.

In the following presentation of monotone fluxes, q_r will refer to the right limit $q^+_{i+1/2}$ of a quantity q. Similarly, $q^-_{i+1/2}$ is abbreviated as q_l.

Monotone fluxes are classified as either upwind methods or central methods. Upwind methods discretize equations on a mesh according to the direction of propagation of information on that mesh. Central methods do not make a distinction based on the direction of information propagation. Within the upwind methods, both Godunov-type methods and flux vector splitting methods are presented based on [34].

3.1.1 Godunov-type Upwind Methods

These methods are also called flux difference splitting methods or Riemann approach methods. In the general case, $u^-_{i+1/2} \neq u^+_{i+1/2}$, i.e., at position $x_{i+1/2}$ a discontinuity is present. The original Godunov monotone flux therefore interpreted this as Riemann problem and provided the conserved variables at $x_{i+1/2}$, $u_{i+1/2}$. This is the state that will be present instantly at this position and will remain constant over a time step. Then, the flux can be evaluated at this position, $f\left(u_{i+1/2}\right)$. The result is the Godunov monotone flux.

As the Godunov monotone flux uses the exact solution to the Riemann problem, the resulting method is computationally relatively expensive and is rarely used for practical computations. Godunov-type monotone fluxes follow the approach of the Godunov monotone flux but employ an approximate Riemann solver. This reduces the computational expense significantly and results in rather accurate monotone fluxes.

Roe's Monotone Flux: This Godunov-type flux uses one of the most well-known approximate Riemann solvers. The approximate Riemann solver is due to Roe [26] and works as follows. The original Riemann problem is replaced by an approximate Riemann problem, which is solved exactly. The approximate problem is based on linearized conservation laws, $u_t + A_{lr}u_x = 0$.

The linearized problem has to be established for each combination of governing equations (e.g., Euler equations) and thermodynamic property model (e.g., ideal gas).

Roe [26] established a methodology using averaged values such that $A_{lr}\left(u^{+}_{i+1/2}-u^{-}_{i+1/2}\right)=A_{lr}\left(\overline{u}\right)$ fulfills the given conditions. The vector $\overline{u}$ is the vector of Roe average values. For the one-dimensional Euler equations and ideal gas, the Roe average values are as follows.

$$\overline{\rho}=\frac{\rho_r+\rho_l}{\sqrt{\rho_r}+\sqrt{\rho_l}}$$

$$\overline{v}=\frac{\sqrt{\rho_r}v_r+\sqrt{\rho_l}v_l}{\sqrt{\rho_r}+\sqrt{\rho_l}}$$

$$\overline{h_0}=\frac{\sqrt{\rho_r}h_{0,r}+\sqrt{\rho_l}h_{0,l}}{\sqrt{\rho_r}+\sqrt{\rho_l}}$$

and

$$\overline{c}^2=(\kappa-1)\left(\overline{h_0}-\frac{1}{2}\overline{v^2}\right)$$

Due to specific properties [26], the linearized system can be transformed into a system of independent transport equations. The data difference $\Delta u=u_r-u_l$ is projected onto the right eigenvectors of A_{lr}. This establishes the wave strengths α_i. Proper integral relations allow to establish the numerical flux as

$$g_{Roe}\left(u_l,u_r\right)=\frac{1}{2}\left(f_l+f_r\right)-\frac{1}{2}\sum_{i=1}^{3}\alpha_i\left|\lambda_i\right|K_i$$

with eigenvalues λ_i and right eigenvectors K_i.

For the problem of interest, the wave strengths are

$$\alpha_1=\frac{1}{2\overline{c}}\left[\Delta m-\Delta\rho\left(\overline{v}+\overline{c}\right)\right]-\frac{1}{2}\alpha_2$$

$$\alpha_2=-\frac{\kappa-1}{\overline{c}^2}\left[\Delta\rho\left(\overline{v}^2-h\right)-\overline{v}\Delta m+\Delta\overline{e}\right]$$

$$\alpha_3=\Delta\rho-\alpha_1-\alpha_2$$

Here, the data difference Δm for example refers to the difference in momentum.

HLLE Monotone Flux: The Harten, Lax and van Leer [12] monotone flux simplifies the approximate Riemann problem even further. It neglects the contact surfaces and consequently assumes that between the shock and the expansion fan only a single homogeneous state is present. For hyperbolic systems of two equations this is correct, but for the Euler equations addressed herein this is a rough approximation. Even if the resolution of contact surfaces is poor, this monotone flux is still a robust and efficient one, whose accuracy is, on global level, often sufficient.

An advantage of this flux is that it can be applied easily to different thermodynamic property models. The approximate Riemann solver of Roe for example is not straight-forward to apply to several problems such as ones involving real gas equations. It is therefore a relevant candidate for equation-based, object-oriented modeling languages applications, as the specific thermodynamic property models are often factored out of the component models, in which the discretized Euler equations are implemented.

The scheme is implemented via an a-priori estimation for the fastest signal speeds and its monotone flux is defined as

$$\begin{aligned}g_{HLLE}\left(u_l,u_r\right)=&\frac{c_r^+ f\left(u_l\right)-c_l^- f\left(u_r\right)}{c_r^+-c_l^-}\\&+\frac{c_r^+c_l^-}{c_r^+-c_l^-}\left(u_r-u_l\right)\end{aligned}$$

Here, the signal speeds are $c_r^+=\max\left(0,v_r+c_r,\overline{v}+\overline{c}\right)$ and $c_l^-=\min\left(0,v_l-c_l,\overline{v}-\overline{c}\right)$ respectively. In these equations the Roe average velocity $\overline{v}$ and the Roe average speed of sound $\overline{c}$ have been used.

3.1.2 Flux Vector Splitting Upwind Methods

In Patankar [19] for instance, a simple first-order upwind scheme in primitive variables was introduced. Based of the sign of a characteristic quantity (usually, this is a velocity normal to the cell boundary), any variable on the boundary was established to have either the value from the left or the right side. In the context of the present approach to conservative methods and high-speed compressible flow, there is no simple scheme of this type. This becomes obvious from the hyperbolicity of the Jacobian $\partial f/\partial u$ and its eigenvalues.

In general, the real part of the eigenvalues can have any sign and a simple one-sided differencing scheme will be appropriate only if the real parts of all eigenvalues have the same sign. The general system will however have some eigenvalues with a positive real part, and one side will be upwind for them, while the others have a negative sign on the real part and consequently the upwind side will be opposite for them. A typical way to resolve this problem is to split such a system into one with a positive real part of the eigenvalues and one with a negative real part and to treat them separately. These are the flux vector splitting methods discussed in this section.

The flux vector splitting approach is also called Boltzmann approach and works as follows [34]. As before, the Jacobian of the system of nonlinear hyperbolic conservation laws (1) is of interest.

$$A\left(u\right)=\frac{\partial f\left(u\right)}{\partial u}$$

Due to hyperbolicity, it may be expressed as

$$A = K\Lambda K^{-1} \tag{12}$$

Here, Λ is the diagonal matrix of eigenvalues λ_i of A. The matrix K is the matrix of right column eigenvectors K_i. The flux vector splitting methods aim at splitting the flux $f(u)$ into components $f^+(u)$ and $f^-(u)$ based on the following equality.

$$f(u) = f^+(u) + f^-(u)$$

Following the introduction of this section, the split fluxes are established such that the eigenvalues $\hat{\lambda}_i^+$, $\hat{\lambda}_i^-$ of the Jacobian

$$\hat{A}^+ = \frac{\partial f^+(u)}{\partial u},$$

$$\hat{A}^- = \frac{\partial f^-(u)}{\partial u}$$

fulfill $Re\left(\hat{\lambda}_i^+\right) \geq 0$ and $Re\left(\hat{\lambda}_i^-\right) \leq 0$.

The Steger-Warming Monotone Flux: In order to establish such a splitting, the homogeneity property of (1) may be exploited. If the system of hyperbolic conservation laws fulfills this property, then

$$f(u) = A(u)u \tag{13}$$

like in the linear constant coefficient case. The unsteady Euler equations fulfill this property and consequently the splitting may utilize the structure exposed in (12), that is, the splitting may be applied to the diagonal matrix Λ. Steger and Warming [30] proposed a splitting of the eigenvalues λ_i,

$$\lambda_i = \lambda_i^+ + \lambda_i^- \tag{14}$$

Here, $\lambda_i^+ \geq 0$ and $\lambda_i^- \leq 0$. Consequently, Λ is split as

$$\Lambda = \Lambda^+ + \Lambda^- \tag{15}$$

$\Lambda^\pm$ are the diagonal matrices of the split eigenvalues $\lambda_i^\pm$. This leads directly to the splitting of A.

$$A = A^+ + A^- \tag{16}$$

where $A^\pm = K\Lambda^\pm K^{-1}$. If the property (13) is fulfilled, one arrives at an expression for the flux splitting.

$$f(u) = f^+(u) + f^-(u) \tag{17}$$

Here, $f^\pm(u) = A^\pm u$.

The crucial question is how to choose $\lambda_i^\pm$ in (14). Steger and Warming [30] suggested to use to following equations.

$$\lambda_i^+ = \frac{1}{2}(\lambda_i + |\lambda_i|) = \max(\lambda_i, 0) \tag{18}$$

$$\lambda_i^- = \frac{1}{2}(\lambda_i - |\lambda_i|) = \min(\lambda_i, 0) \tag{19}$$

When exercising this approach, the following Steger-Warming monotone flux is established.

$$g_{SW}(u) = f^+(u) + f^-(u)$$

with

$$f^\pm(u) = \frac{\rho}{2\kappa} \begin{bmatrix} \lambda_1^\pm + 2(\kappa-1)\lambda_2^\pm + \lambda_3^\pm \\ (v-c)\lambda_1^\pm + 2(\kappa-1)v\lambda_2^\pm + (v+c)\lambda_3^\pm \\ (h-vc)\lambda_1^\pm + (\kappa-1)v^2\lambda_2^\pm + (h+vc)\lambda_3^\pm \end{bmatrix}$$

The eigenvalues are given by (18) and (19). The remaining variables have to be evaluated according to the definition of the flux, i.e., for $f^+(u)$ the values from the left such as ρ_l, u_l and for $f^-(u)$ the values from the right such as ρ_r, u_r.

3.1.3 Centered Methods

Schemes, whose support does not depend on the sign of the characteristic speeds, are called centered schemes.

The Rusanov Monotone Flux, a local Lax-Friedrichs Flux: The Lax-Friedrichs flux is one of the simplest and most approximate methods considered herein. It was originally developed in the context of finite-difference methods and later applied to the finite-volume context.

Similarly to the HLLE method, only an expansion and a compression wave are considered. In the original Lax-Friedrichs flux, the speed of each wave was assumed to be such that it reached the cell boundaries exactly within a time step Δt. For uniform grids, each wave of the global problem therefore had the same speed, which is an even more approximate solution than in the HLLE method. As, in the present context, no fully explicit scheme is employed but the method of lines, no time step Δt is defined. For this reason and to slightly improve accuracy, a local form of the Lax-Friedrichs monotone flux, the Rusanov monotone flux [27], is considered. In the Lax-Friedrichs flux,

$$g_{LF}(u_l, u_r) = \frac{1}{2}(f(u_r) + f(u_l)) - \frac{1}{2}\frac{\Delta x}{\Delta t}(u_r - u_l)$$

the signal speed $\Delta x/\Delta t$ is replaced by $\lambda_{max} = \max\left((|v|+c)_l, (|v|+c)_r\right)$. Then, the Rusanov monotone flux is defined as follows.

$$g_{Rus}\left(u_l, u_r\right) = \frac{1}{2}\left(f\left(u_r\right)+f\left(u_l\right)\right) - \frac{1}{2}\lambda_{max}\left(u_r - u_l\right)$$

First-Order Centered Monotone Flux: The First-Order Centered Monotone flux (FORCE scheme) [33] is obtained when replacing the random sampling of Riemann problems in Random Choice Methods with deterministic integral averages.

According to Toro [34], for fully explicit schemes, the result is the arithmetic mean of the Lax-Friedrichs and Richtmyer [25] fluxes. The Richtmyer flux is a second-order scheme with constant coefficients and is thus, according to Godunov's classic theorem [9], not monotone and results in spurious oscillations.

For the fully explicit version of the Richtmyer flux, an intermediate state is first defined,

$$u_{Ri} = \frac{1}{2}\left(u_l + u_r\right) + \frac{1}{2}\frac{\Delta t}{\Delta x}\left(f\left(u_l\right)+f\left(u_r\right)\right)$$

and then the flux is evaluated at it.

$$g_{Ri}\left(u_l, u_r\right) = f\left(u_{Ri}\right)$$

Then, the FORCE flux is the arithmetic mean of the Lax-Friedrichs and Richtmyer fluxes [34]

$$g_{Force}\left(u_l, u_r\right) = \frac{1}{2}\left(g_{LF}\left(u_l, u_r\right) + g_{Ri}\left(u_l, u_r\right)\right)$$

Again, the local version of the Lax-Friedrichs flux (the Rusanov flux presented in previous section) and a local version of the Richtmyer flux are used, is again obtained by replacing $\Delta x/\Delta t$ with λ_{max}.

After introducing some monotone numerical fluxes, methods to obtain higher-order approximations of the solution to (1) are considered.

3.2 Total Variation Diminishing schemes

Godunov's theorem [9] was mentioned already. It provides the theoretical foundation to the observation that *linear* second-order schemes are more accurate in smooth regions of a problem solution to (1) than first-order schemes. Near strong gradients and shocks, these methods produce spurious oscillations however. Monotone methods however do not exhibit such spurious oscillations. In case of linear schemes, their limited first-order accuracy is disadvantageous however.

One option to eliminate or reduce spurious oscillations for higher-order methods is to introduce artificial viscosity. This can be tuned such that it is large enough to suppress oscillations in the neighborhood of discontinuities and small elsewhere to maintain accuracy. A disadvantage of this approach is however, that the quantity of artificial viscosity is problem dependent and therefore requires fine-tuning by the user. This approach is not followed here and instead a less empirical approach to introduce viscosity is adopted.

Therefore, in order to circumvent the limitations formulated by Godunov's theorem, schemes with variable coefficients, i.e., nonlinear schemes, are considered. Such schemes can adapt themselves to the local nature of the solution.

Harten [10] defined High-Resolution Methods as numerical methods with the following properties.

1. Second or higher-order of accuracy in smooth parts of the solution
2. The solution is free of spurious oscillations.
3. The resolution of discontinuities in the solution is high, i.e., the number of cells containing the numerical reproduction of the discontinuity is smaller in comparison with that of first-order monotone schemes.

A class of methods fulfilling these properties is that of Total Variation Diminishing methods [10]. See this reference for a definition of the total variation. For brevity, only the case of a smooth function $u(t)$, for which the total variation is

$$TV\left(u\right) = \int_{-\infty}^{\infty}\left|u'\left(x\right)\right| dx$$

and the case of a mesh function $u^n = \{u_i^n\}$ are mentioned. For the latter, the total variation is defined as

$$TV\left(u^n\right) = \sum_{i=-\infty}^{\infty}\left|u_{i+1}^n - u_i^n\right|$$

Fundamental properties of the exact solution of the conservation law (1) such as no creation of new local extrema lead to the conclusion that the total variation $TV(u(t))$ is a decreasing function of time [10]. Consequently, Total Variation Diminishing methods mimic a property of the exact solution.

For a general scalar conservation law, Harten [10] provided a theorem on a sufficient condition for a particular class of nonlinear schemes with two coefficients to be Total Variation Diminishing (TVD). These conditions are essentially four inequalities on these two coefficients. As the coefficients may in general be data dependent, Harten's theorem provides a tool for the construction of nonlinear schemes that circumvent Godunov's theorem stated above.

The classic TVD approach to adaptively switch between the characteristics of a monotone first-order numerical flux g^{LO} and those of a higher-order constant coefficient flux g^{HI} is to make the following assumption [32].

$$g^{TVD} = g^{LO} + \varphi\left[g^{HI} - g^{LO}\right]$$

Here, φ is a flux limiter function that implements the adaptive algorithm. Analysis of Harten's theorem led to the identification of the Sweby TVD region [32]. In this region, various flux limiters have been defined such as the well-known limiters Superbee, Minbee, and Ultrabee.

In the following sections, this approach is not followed directly. Instead of flux limiters, slope limiters are used, which are analogous to the flux limiters.

For the reasons described in section 3.1.3, both an upwind TVD and a central TVD method are considered.

3.2.1 A MUSCL Upstream TVD Scheme

Van Leer [37, 38, 39] introduced a higher-order method along the concept of reconstruction mentioned in the introduction of this paper. MUSCL stands for Monotone Upstream-Centered Scheme for Conservation Laws.

The first-order schemes discussed so far use monotone fluxes directly by assuming piecewise constant data over the cells I_i, i.e., $u^-_{i+1/2} \approx u_i$ and $u^+_{i+1/2} \approx u_{i+1}$. In the simplest MUSCL scheme, piecewise linear local reconstructions are used. The reconstruction has to maintain the integral average, which is trivially fulfilled for piecewise linear local reconstructions.

First, slope vectors $\Delta_{i\pm 1/2}$ are defined as follows.

$$\Delta_{i-1/2} = u_i - u_{i-1} \tag{20}$$

$$\Delta_{i+1/2} = u_{i+1} - u_i \tag{21}$$

Strictly speaking, these slopes are not slopes but differences of the vector of conserved quantities in adjacent cells. The terminology used in literature is adopted however and therefore $\Delta_{i\pm 1/2}$ are called slope vectors. In order to implement a TVD scheme, the approach of limited slopes described by Quirk [23] is used.

$$\hat{\Delta}_i = \begin{cases} \max[0, & \\ \quad \min\left(\beta\Delta_{i-1/2}, \Delta_{i+1/2}\right), & \\ \quad \min\left(\Delta_{i-1/2}, \beta\Delta_{i+1/2}\right)] & \Delta_{i+1/2} > 0 \\ \min[0, & \\ \quad \max\left(\beta\Delta_{i-1/2}, \Delta_{i+1/2}\right), & \\ \quad \max\left(\Delta_{i-1/2}, \beta\Delta_{i+1/2}\right)] & \Delta_{i+1/2} < 0 \end{cases}$$

The value $\beta = 1$ does, in the scalar case, reproduce the Minbee flux limiter, and $\beta = 2$ the Superbee flux limiter.

Based on the piecewise linear local reconstruction,

$$u_i(x,t) = \overline{u_i}(t) + \frac{x - x_i}{\Delta x_i}\hat{\Delta}_i$$

The values at the extreme points of the cell I_i are established.

$$u^+_{i-1/2} = \overline{u_i} - \frac{1}{2}\widehat{\Delta}_i \tag{22}$$

$$u^-_{i+1/2} = \overline{u_i} + \frac{1}{2}\widehat{\Delta}_i \tag{23}$$

In order to finally obtain the second-order accurate upstream flux, some first-order monotone upstream flux is employed with the reconstructed values $u^-_{i+1/2}$, $u^+_{i+1/2}$.

$$g^{TVDu}_{i+1/2} = g^{mu}_{i+1/2}\left(u^-_{i+1/2}, u^+_{i+1/2}\right)$$

Note that $u^-_{i+1/2}$ is obtained from a reconstruction in cell I_i, and $u^+_{i+1/2}$ from a reconstruction in cell I_{i+1}.

3.2.2 A MUSCL Centered TVD Scheme

As mentioned before, also a second-order TVD centered scheme is introduced. It also follows the concept of the MUSCL scheme but uses a first-order monotone *centered* flux.

This approach is base on a slope limiter ξ_i, for which the following equation holds.

$$\hat{\Delta}_i = \xi_i \Delta_i$$

Here, the slope vector of the cells, Δ_i, is used.

$$\Delta_i = \frac{1}{2}(1+\omega)\Delta_{i-1/2} + \frac{1}{2}(1-\omega)\Delta_{i+1/2}$$

This is a weighted average of the side slope vectors $\Delta_{i\pm 1/2}$, see (20) and (21). The weighting parameter has to fulfill $\omega \in [-1,1]$. In computations conducted for this paper, the value of $\omega = 0$ was used. Additionally, the ratio r_i of the cell side slope vectors is introduced.

$$r_i = \frac{\Delta_{i-1/2}}{\Delta_{i+1/2}}$$

Then, a slope limiter analogous to the Superbee flux limiter is [34]

$$\xi_{sb}(r) = \begin{cases} 0 & r \leqslant 0 \\ 2r & 0 \leqslant r \leqslant \frac{1}{2} \\ 1 & \frac{1}{2} \leqslant r \leqslant 1 \\ \min(r, \xi_r(r), 2) & r \geqslant 1 \end{cases}$$

A van Leer-type slope limiter is [34]

$$\xi_{vl}(r) = \begin{cases} 0 & r \leqslant 0 \\ \min\left(\frac{2r}{1+r}, \xi_r(r)\right) & r \geqslant 0 \end{cases}$$

A Minbee-type slope limiter is [34]

$$\xi_{mb}(r) = \begin{cases} 0 & r \leqslant 0 \\ r & 0 \leqslant r \leqslant 1 \\ \min(1, \xi_r(r)) & r \geqslant 1 \end{cases}$$

Above, $\xi_r(r)$, a TVD region limit that is defined as follows, was used.

$$\xi_r(r) = \frac{2}{1 - \omega + (1 + \omega) r}$$

As before, the conservative variable vector is approximated via the limited slope $\hat{\Delta}_i$ and equations (22) and (23). Then, the second-order accurate centered flux is obtained via a first-order monotone centered flux with the reconstructed values $u^-_{i+1/2}$, $u^+_{i+1/2}$. For this purpose, the FORCE flux can be used.

$$g^{TVDc}_{i+1/2} = g^{Force}_{i+1/2}\left(u^-_{i+1/2}, u^+_{i+1/2}\right)$$

Note again that $u^-_{i+1/2}$ is obtained from a reconstruction in cell I_i, and $u^+_{i+1/2}$ from a reconstruction in cell I_{i+1}.

3.3 Weighted Essentially Non-Oscillatory schemes

One disadvantage of TVD schemes is that the accuracy near discontinuities is reduced. In the schemes presented above, this was directly visible in the slope for example. Also, the accuracy necessarily is reduced to first-order near *smooth* extrema.

In this section, both Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory schemes are presented, which are self-similar (i.e., there is no mesh size dependent parameter), uniformly high-order accurate, yet essentially non-oscillatory for piecewise smooth functions (i.e., the magnitude of the oscillations decays with order of accuracy of the scheme). Piecewise smooth functions are smooth except at finitely many isolated points. At these points, the function and its derivatives are assumed to have finite left and right limits.

The key element of these schemes is the reconstruction. This is a specific interpolation technique, which was developed for piecewise smooth functions. It works by automatically choosing the locally smoothest stencil, and by that avoiding crossing discontinuities in the interpolation procedure as much as possible.

The Essentially Non-Oscillatory reconstruction algorithm starts with a stencil containing one or two cells only. It then adds either the cell to the right or the one to the left of the stencil, depending on which results in the less oscillatory interpolant.

Instead of choosing one of the candidate stencils and discarding the others, Weighted Essentially Non-Oscillatory reconstruction uses a convex combination of the interpolant through all candidate stencils.

First, the given two reconstructions are presented and then it is described how to establish a numerical flux from the corresponding reconstructions. This section is based on Shu [28].

3.3.1 Essentially Non-Oscillatory Reconstruction

Before describing the Essentially Non-Oscillatory (ENO) reconstruction, an important detail of interpolation methods used for reconstruction has to be addressed. In section 3.2 it was mentioned that linear interpolation in the MUSCL scheme was uncritical with respect to maintaining the proper cell average of the interpolant. In the context of the present methods, higher-order interpolation is considered and therefore the interpolant must be established in a way that maintains the cell average.

Assume that some function, say, velocity, is considered. The cell averages $\overline{v_i}$ of that function $v(x)$ are given on a grid. One is interested in a polynomial $p_i(x)$ of degree $k-1$ for each cell I_i. This then forms a k-th order approximation to $v(x)$ in the cell I_i. The polynomial shall be constructed such that its cell average shall agree with that of the original function $\overline{v_i}$.

Assume that, additionally to the cell I_i and the order of accuracy k, one is given a stencil $S(i)$ of k consecutive cells. The stencil is given via the left shift r, i.e., the stencil includes r cells to the left and s cells to the right of I_i, with $r+s+1=k$.

$$S(i) = \{I_{i-r}, \ldots, I_{i+s}\} \tag{24}$$

In order to preserve the cell average, the interpolant over the stencil is established via the primitive function of $v(x)$.

$$V(x) = \int_{-\infty}^{x} v(\xi)\, d\xi$$

Then, the interpolant can be established. In computational implementations, this interpolation step is usually accelerated via the computation of so-called reconstruction coefficients. This is possible, because one is usually not interested in the complete interpolant but only in values of it at specific stations such as $x_{i+1/2}$. Due to the linearity of the mapping from the cell averages $\overline{v_i}$ to the interpolated values, these reconstruction coefficients depend on the left shift of the stencil r, the order k, and the mesh spacing Δx_i, but not on the function v itself.

The actual ENO approximation is addressed next. Here, an adaptive stencil is used. This means that the left shift r is not constant. A left shift r that is constant over the cells I_i would lead to a fixed stencil approximation (e.g., a central stencil) for which it was shown that it leads to spurious oscillations if of order two or higher with constant coefficients. In ENO approximation, the left shift is thus established for each cell I_i in a way that avoids including a cell with a discontinuous change in the stencil.

Harten et al. [11] showed that a robust criterion to identify the stencil with the "smoother" interpolant is to choose the one with the smaller absolute value of the Newton divided difference.

Recall the definition of the Newton divided differences. For the primitive function $V(x)$ the 0-th degree divided difference is

$$V\left[x_{i-1/2}\right] = V\left(x_{i-1/2}\right)$$

and the general j-th degree divided difference with $j \geq 1$ is defined as

$$V\left[x_{i-1/2},\dots,x_{i+j-1/2}\right] = \frac{V\left[x_{i+1/2},\dots,x_{i+j-1/2}\right] - V\left[x_{i-1/2},\dots,x_{i+j-3/2}\right]}{x_{i+j-1/2} - x_{i-1/2}}$$

Similarly, the divided differences of the cell averages are

$$\overline{v}\left[x_i\right] = \overline{v_i}$$

and in general

$$\overline{v}\left[x_i,\dots,x_{i+j}\right] = \frac{\overline{v}\left[x_{i+1},\dots,x_{i+j}\right] - \overline{v}\left[x_i,\dots,x_{i+j-1}\right]}{x_{i+j} - x_i} \tag{25}$$

Note that the zeroth degree divided difference of $\overline{v_i}$ is identical to the first degree divided difference of $V(x)$ due to the definition of the primitive function.

$$V\left[x_{i-1/2},x_{i+1/2}\right] = \frac{V\left(x_{i+1/2}\right) - V\left(x_{i-1/2}\right)}{x_{i+1/2} - x_{i-1/2}} = \overline{v_i} \tag{26}$$

This equality allows to express the divided differences of $V(x)$ of degree $j \geq 1$ by those of $\overline{v_i}$ of degree $j \geq 0$. Taking the derivative of the k-th degree interpolation polynomial $P(x)$ to approximate $V(x)$, one finds that only divided difference of $\overline{v_i}$ of degree $j \geq 1$ are required to express $p(x)$.

The ENO approximation thus identifies the "smoothest" stencil in $\overline{v_i}$ via a stencil of $V(x)$, which is labeled $\hat{S}(i)$. Notice that from the latter the corresponding stencil in $\overline{v_i}$ can be identified via (26). First, the divided differences of the primitive function $V(x)$ are computed using (26) and, for degrees $j \geq 2$, using (25). Then, the algorithm starts with a two point stencil in $V(x)$,

$$\hat{S}_2(i) = \left\{x_{i-1/2},x_{i+1/2}\right\}$$

This stencil is then consecutively enlarged for $l = 2,\dots,k$. From the preceding step the following stencil is known

$$\hat{S}_l(i) = \left\{x_{i+1/2},\dots,x_{j+l-1/2}\right\}$$

and one of the neighboring points $x_{j-1/2}$ and $x_{j+l+1/2}$ is added to the stencil. If

$$\left|V\left[x_{j-1/2},\dots,x_{j+l-1/2}\right]\right| < \left|V\left[x_{j+1/2},\dots,x_{j+l+1/2}\right]\right|$$

then $x_{j-1/2}$ is added to $\hat{S}_l(i)$ to obtain $\hat{S}_{l+1}(i)$. If the inequality is not fulfilled, then $x_{j+l+1/2}$ is added to the stencil.

As soon as the stencil is completely established, Lagrange or Newton interpolation can be used to find the interpolants. In computational implementations the reconstruction coefficients mentioned at the beginning of this section are usually used instead. By the choice of the stencil the left shift r is established. Then, the proper reconstruction coefficients can be used to instantly establish the interpolated values at the interface locations.

Figure 1 illustrates the interpolants chosen by Essentially Non-Oscillatory schemes. For the example $\overline{v} = \{10, 10.4, 10.25, 10, 3, 2.5, 2.25, 2\}$ and $x = \{1,2,3,4,5,6,7,8\}$ were assumed. First, consider the resulting interpolant for cell 3. The scheme described above starts the stencil with this cell and extends it twice (i.e., *order* − 1 times) to the left or right. As described, the schemes includes either neighbor point that results in a smoother interpolant according to the criterion of divided differences. For cell 3, the scheme once selects a cell to the left and once a cell to the right for inclusion in the stencil. For cell 4 in turn, including the right cell (cell 5) would lead to rather large gradients in the interpolant each time. Therefore, the stencil is extended twice to the left. The interpolant for cell 4

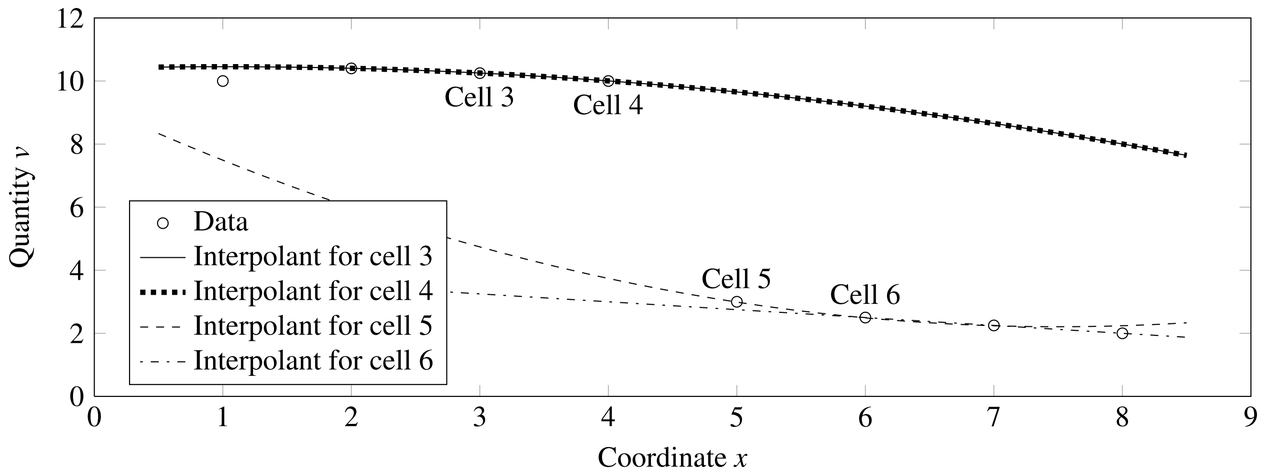

Figure 1: Third-order ENO reconstruction

is therefore identical to that of cell 3. For cells 5 and 6, the stencil is only extended to points to the right for similar reasons.

The left limit of $v_{4+1/2}$ is established based on the interpolant of cell 4, i.e., $v^{-}_{4+1/2} = 9.84$. The right limit is $v^{+}_{4+1/2} = 3.33$.

3.3.2 Weighted Essentially Non-Oscillatory Reconstruction

ENO schemes are uniformly high-order accurate right up to the discontinuity, which is achieved by adaptively switching the stencil used for interpolation. However, certain properties leave room for improvements [28]:

- The stencil may change near zeros of the solution even by a round-off error perturbation.
- As the left shift of the stencil may change at neighboring points, the resulting numerical flux is not smooth.
- To the reconstruction scheme, $2k-1$ cells are available. In the end, only k cells are used. This results in k-th order accuracy when $2k-1$-th order accuracy is theoretically possible in smooth regions of the solution.

The idea of Weighted Essentially Non-Oscillatory (WENO) reconstruction is to use a convex combination of the interpolants through several stencils. If, however, a candidate stencil contains a discontinuity, its weight shall be close to zero to mimic the successful properties of ENO schemes.

For each cell I_i k candidate stencils are consequently available.

$$S_r(i) = \{x_{i-r}, \dots, x_{i-r+k-1}\}$$

with $r = 0, \dots, k-1$. Using the reconstruction coefficients, each stencil produces a different reconstruction of $v_{i+1/2}$, which is labeled $v^{(r)}_{i+1/2}$. A convex combination of these values is used to define the reconstruction using the WENO method.

$$v_{i+1/2} = \sum_{r=0}^{k-1} \omega_r v^{(r)}_{i+1/2}$$

For stability and consistency, $\omega_r \geq 0$ and $\sum\limits_{r=0}^{k-1} \omega_r = 1$ need to be imposed. In smooth regions, these weights should approximate optimal high-order weights to $k-1$-th order, which would imply $(2k-1)$-th order of the complete reconstruction scheme. The question is now what these optimal weights are. In the general case, this leads to an overdetermined system of equations, which can be solved, e.g., by using a least-squares algorithm. In the case of a uniform mesh, the equation system becomes square and an explicit solution can be computed. Jiang and Shu [16] gave optimal weights d_r for uniform grids and $1 \leq k \leq 3$. Herein, $k = 3$ is considered. For this value of k, the following optimal weights have been established.

$$d_0 = \frac{3}{10}, \qquad d_1 = \frac{3}{5}, \qquad d_2 = \frac{1}{10}$$

Furthermore, Jiang and Shu [16] suggested the following form of the weights

$$\omega_r = \frac{\alpha_r}{\sum\limits_{s=0}^{k-1} \alpha_s}$$

for $r = 0,\ldots,k-1$. Coefficients α_r in turn are defined as follows

$$\alpha_r = \frac{d_r}{(\varepsilon + \beta_r)^2}$$

Here, $\varepsilon > 0$ is introduced to avoid division by zero. Following Jiang and Shu [16], $\varepsilon = 10^{-6}$ was used in computations. β_r are called smooth indicators in the given reference and have been defined as follows

$$\beta_r = \sum_{l=1}^{k-1} \int_{x_{i-1/2}}^{x_{i+1/2}} \Delta x^{2l-1} \left(\frac{\partial^l p_r(x)}{\partial x^l} \right)^2 dx$$

This is the sum of the squares of the scaled L^2 norms for all derivatives of the interpolation polynomial $p_r(x)$ over the interval $\left(x_{i-1/2}, x_{i+1/2}\right)$. For $k = 3$, the result is a $2k-1 = 5$-th order accurate reconstruction.

Figure 2 illustrates Weighted Essentially Non-Oscillatory reconstruction on the same example as figure 1. The reconstruction of the left limit of $v_{4+1/2}$ is considered, i.e., $v^-_{4+1/2}$. For this, the scheme uses three stencils $S_r(4)$ with increasing left-shift r. The interpolants based on these stencils are illustrated in the figure. Note the strong gradients in the interpolants using $S_0(4)$ and $S_1(4)$. This is also an illustration that the stencil selection of the ENO scheme shown in figure 1 for cell 4 was reasonable.

The WENO scheme proceeds with the different reconstruction values $v^{(0)}_{4+1/2}$ to $v^{(2)}_{4+1/2}$, which are each marked with a filled circle in figure 2. For this particular example, the scheme results in weights $\omega_0 = 1.3 \cdot 10^{-6}$, $\omega_1 = 15.6 \cdot 10^{-6}$, $\omega_2 = 0.999983$. This means, that the interpolant with left-shift $r = 2$ dominates and $v^-_{4+1/2} \approx v^{(2)}_{4+1/2}$.

3.3.3 ENO and WENO numerical fluxes

So far, two different algorithms for the reconstruction of piecewise smooth functions were introduced. The question is now how to construct corresponding higher-order numerical fluxes for the system of hyperbolic conservation laws (1) from these reconstructions.

Probably, the easiest way to do this is to apply the reconstruction to each component of the vector of conserved variables u separately and thus reconstruct the left and right limit $u^{\pm}_{i+1/2}$ at the location $x_{i+1/2}$. Then, a monotone first-order flux can be used to establish an essentially non-oscillating higher-order numerical flux.

Shu [28] remarks that only low-order schemes are highly sensitive to the choice of first-order monotone flux. This sensitivity decreases with increasing order of accuracy and therefore a simple Lax-Friedrichs monotone flux is used in the given reference to construct higher-order WENO numerical fluxes.

The given component-wise approach to construct a numerical flux based on ENO and WENO reconstructions is simple to implement. Also, the resulting schemes work reasonably well for many applications, in particular if the order of the scheme is not high. Shu [28] mentions “second or sometimes third-order”.

If the order of the scheme is high or a more demanding test problem shall be solved, the following characteristic decomposition is much more robust and should be implemented instead.

Recall the diagonal decomposition of the Jacobian of the flux in section 3.1.2 on flux vector splitting, (12). A change of variables $v = K^{-1}u$ leads to a decoupling of the system of conservation laws (1). Then, the component-wise application of the ENO or WENO reconstruction is fundamentally justified. The reconstructed values $v^{\pm}_{i+1/2}$ are then transformed back into the physical space of conserved variables,

$$u^{\pm}_{i+1/2} = K v^{\pm}_{i+1/2}$$

A remaining question is the choice of K, which depends on u, $K = K(u)$. For this purpose, the Roe averages introduced in section 3.1.1 were used, as this leads to advantageous properties such as the satisfaction of the mean value theorem.

Based on the reconstructed left and right limit $u^{\pm}_{i+1/2}$ at the location $x_{i+1/2}$, a monotone first-order flux is used again to establish an essentially non-oscillating higher-order numerical flux.

4 Object-oriented implementation

Two libraries for object-oriented modeling and simulation of gas dynamics were developed for [29] and this paper. Both were written in Modelica. The first one is a library specific to ideal gases, which allows several simplifications and results in little computational overhead. The second one is a gas dynamics library for generic thermodynamic property models. These thermodynamic property models are implemented according to the object-oriented interface MODELICA.MEDIA [6]. This interface had to be extended with two additional methods to be suitable for applications in gas dynamics. These and other implementation aspects are discussed in this section.

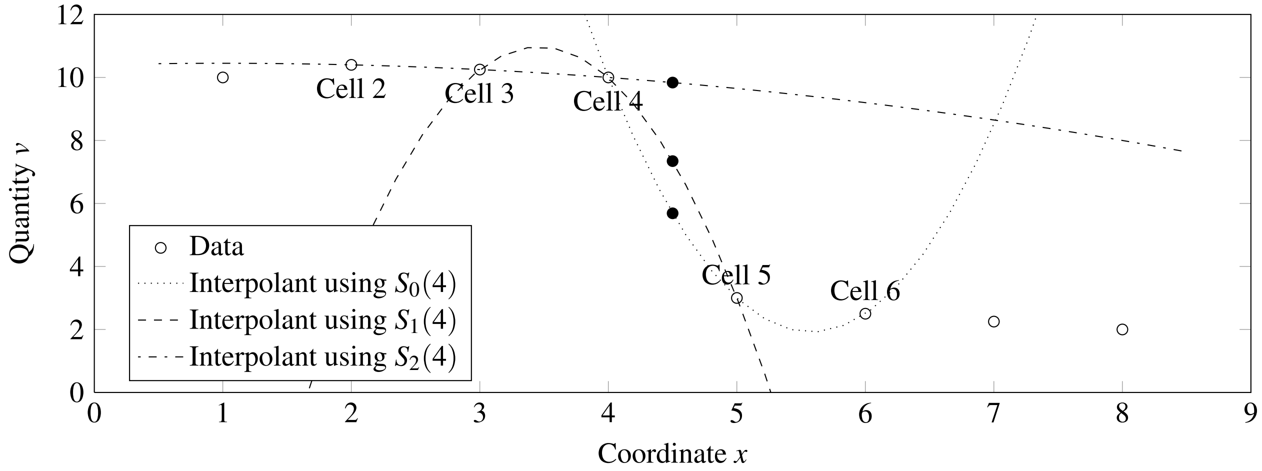

Figure 2: Fifth-order WENO reconstruction

4.1 Ideal gas and generic thermodynamic property models

A large fraction of the literature on discretization methods using conservative methods considers ideal gas equations of state only. Discretizations using real gas[1] equations of state in turn consider non-ideal media, too. Several articles make assumptions on the structure of the real gas equations of state however (e.g,. Liou et al. [18] assume a “general pressure function” but require that is be explicit in density, specific internal energy, and mass fractions, and Gallouët et al. [8] explicitly assume Tammann and van der Waals equations of state).

In equation-based, object-oriented modeling and simulation, one aims to encapsulate the equations of state in separate classes and implement discretization methods independently using a generic interface. As the given real gas schemes require structural assumptions on the equations of state, too, a generic interface had to be extended with several methods specific to these structural assumptions. A clean separation between discretization scheme and equation of state appears to be difficult in this case.

A large fraction of the methods described in the previous section 3 are specific to ideal gases with constant specific heat capacity c_p. Specialized Riemann solvers can be constructed easily for some of these methods (such as the HLLE method described in section 3.1.1). In the context of equation-based, object-oriented modeling languages, such approximate Riemann solvers had to be exchanged *concurrently* with the equations of state. A more practical solution is the use of centered schemes. These schemes are independent of any Riemann solver and can thus be used with any thermodynamic property model. As described in section 3.1.3, the support of these schemes does not depend on the sign of the characteristic speeds. While the upwind schemes as discussed in sections 3.1.1 and 3.1.2 are more accurate in several cases than their centered counterparts, they are usually more complex and computationally expensive [34]. Therefore, in the libraries described herein, monotone and TVD centered schemes as well as schemes using higher-order reconstruction with a centered scheme are implemented for general thermodynamic property models and upwind methods are restricted to ideal gases.

4.2 Generic interface to thermodynamic property computations

As described above, the object-oriented interface of MODELICA.MEDIA [6] is used for thermodynamic property computations. In order to be applicable to gas dynamics, this interface has to be extended with two additional methods.

The first extension is required for the conversion of conserved variables to primitive variables. In the gas dynamics library for generic equations of state the primitive variables are velocity v and the thermodynamic state record of the medium[2]. For the conversion of the vector u as defined in equation (2) to the primitive variables an additional `setState` function is thus required. From u, density and specific internal energy can be established. Therefore, a function

[1] In this thesis, a real gas is one that is not both thermally and calorically ideal.

[2] In place of the velocity the mass flow rate could have been used, too. This selection is ambiguous and was eventually made for similarity with conventional implementations of gas dynamics.

`setState_duX` is used.

The second extension is required for the conversion of the classic primitive variables $\{\rho, v, p\}$ to the ones used in the object-oriented implementation for generic thermodynamic property computations, the thermodynamic state record and velocity. This is necessary in case of a characteristic decomposition such as the one discussed in section 3.3.3. For this purpose, a function `setState_pdX` is required. Note that this is only required if a gas dynamics library for generic thermodynamic property models shall also be used with ideal gases.

4.3 Conservative and non-conservative formulations

In order to obtain valid simulation results, the conserved quantities in the governing equations and the conservation statements they imply have to make physical sense [34]. Formulations that are conservative purely in a mathematical sense (i.e., formally, they can be expressed as (1), but there is no corresponding conservation law in physics) will, in case of shock waves, result in wrong shock speeds and therefore wrong solutions [34].

In the context of equation-based, object-oriented modeling languages, a simple solution is to explicitly select the conserved variables themselves as state variables, i.e., $u(x,t)$. This is done in the gas dynamics library specific to ideal gases. For ideal gases that are both thermally and calorically ideal (in particular, c_p is not a function of temperature), all intensive quantities can be established in closed form based on any two thermodynamic potentials. Therefore, no distinction between independent and dependent variables is required for such media.

For generic thermodynamic property models this is different. In general, such models are explicit in a number of thermodynamic potentials only (e.g., pressure and specific enthalpy). As long as the physical flux is not changed, it is then possible to use the independent variables of a thermodynamic property model as state variables instead. This is the approach followed in the gas dynamics library for generic thermodynamic property models.

4.4 Inhomogeneous problems

In several references on computational methods for gas dynamics, fully explicit conservative methods are considered in contrast to (10). In the context of equation-based, object-oriented modeling, it is natural however to use a semi-discretized formulation. Furthermore, this has advantages for inhomogeneous problems. No source term splitting schemes [31] are required for the present approach. With the semi-discretization (also called method of lines) both the numeric fluxes and the source term are algebraic expressions and no further complications arise for inhomogeneous problems.

4.5 Library design

In this section, the design of the two gas dynamics libraries is sketched. The one considering generic thermodynamic property models is emphasized and some remarks are made on the one specific to ideal gases. For readability, the code illustrates single-substance media only. Mass fractions of multiple-substance media can be covered analogously to the other primitive variables, because they are similarly dominated by convection.

The connector has to implement the stencil defined in equation (8). Its length depends on the stencil length required by the discretization scheme. If the stencil for a flux computation has to include n cells, then at least $n/2$ of these cells are inside the domain modeled by the respective component and need not be accessed via the connector. This implies that at most $n/2$ cells of the stencil have to be provided by the connector. Therefore, the connector definition given in listing 1 is used.

Note the replaceable discretization package (“`Discretization`”) in the connector definition in addition to the replaceable package containing the thermodynamic property model (“`Medium`”). A vector of thermodynamic states and one of velocities of the given length are defined twice. Different causal prefixes are used to handle how one component “prescribes” and “reads” which variables[3]. The library considering ideal gases only uses density and pressure vectors in place of the thermodynamic state.

Additionally, information about the computational mesh has to be included in the connector. In the proposed connector definition, the coordinates of the sides of the cells are used. They are defined in a local coordinate system, whose origin is set to the side shared by two components connected together. The coordinate of this shared side can thus be omitted and the same number of side coordinates and cell center variables on

[3] The causal prefixes are used in the acausal modeling language just to define a nominal causality, not an actual one.

```
connector Stencil_a
  "Interface for quasi one-dimensional high-speed flow"

  replaceable package Medium =
    Modelica.Media.Interfaces.PartialMedium "Medium model";

  replaceable package Discretization =
      GasDynamics.Discretizations.Partial.PartialDiscretization
    "Discretization";

  output Medium.ThermodynamicState
    state_a[Discretization.halfStencilLength]
    "Thermodynamic state stencil";
  output SI.Velocity v_a[Discretization.halfStencilLength]
    "Velocity stencil";
  output SI.Length x_side_a[Discretization.halfStencilLength]
    "Cell side coordinate";

  input Medium.ThermodynamicState
    state_b[Discretization.halfStencilLength]
    "Thermodynamic state stencil";
  input SI.Velocity v_b[Discretization.halfStencilLength]
    "Velocity stencil";
  input SI.Length x_side_b[Discretization.halfStencilLength]
    "Cell side coordinate";
end Stencil_a;
```

Listing 1: Connector for high-speed compressible flow

the thermodynamic state and velocity is included. The side coordinates for `Stencil_a` are defined strictly positive; those for `Stencil_b` strictly negative.

Analogous to the `Stencil_a` connector definition in listing 1, a connector `Stencil_b` is defined. It differs only in inverted causality prefixes (`input` instead of `output` and vice versa).

The discretization package contains structural parameters including the stencil length, conversion functions, an exchangeable thermodynamic properties model, and flux functions. Its interface is defined in listings 2 to 4.

The structural parameters of a Discretization are its name, whether it uses equations applicable to ideal gases, its order of accuracy, and the stencil length.

The conversion functions of a Discretization convert the set of primitive variables (thermodynamic state record and velocity) to the vector of conserved variables as defined in equation (2) and vice versa. Note that these functions need not be replaceable, because the implementations are generally valid. Note that in the second conversion function in listing 3 one of the additional functions mentioned in section 4.2 is used (`setState_duX()`).

The key elements of a Discretization are the flux functions. Their interfaces are described in listing 4. For readability, interfaces are defined for both a monotone first-order flux and the arbitrary-order numerical flux. This allows to clearly separate the reconstruction and the Riemann solver for instance. In models, only the arbitrary-order numerical flux is used and therefore the use of the monotone flux function is optional. The monotone flux arguments are the left and right thermodynamic state and the flow velocities. It returns the flux vector. The arbitrary-order flux function has a stencil of thermodynamic states and of velocity as well as the cell side coordinates as inputs and also returns the flux vector. The Discretization package also contains a replaceable package implementing thermodynamic properties. This is not shown in listings 2 to 4. Discretization packages were implemented using the Local Lax-Friedrichs flux, Roe's Riemann solver, the HLLE Riemann solver, the Steger-Warming flux vector splitting, the First-Order Centered flux, the Muscl-

```
partial package PartialDiscretization
  "Interface for discretization in compact flux form"

  // Description
  constant String discretizationName =
    "unusablePartialDiscretization"
    "Name of the discretization";

  // Type of discretization
  constant Boolean idealGasOnly = false
    " = true, if contains specifics of ideal gases";
  constant Integer order(min=1) = 1
    "Order of discretization method";

  // Stencil definition
  constant Integer halfStencilLength = 1
    "Half of length of stencil for flux f_(i+1/2)";
  final constant Integer stencilLength = 2*halfStencilLength
    "Length of stencil for flux f_(i+1/2)";

  // ...

end PartialDiscretization;
```

Listing 2: Discretization interface, structural parameters

Hancock TVD scheme with several limiters and monotone fluxes both in upstream and in centered versions, third- to ninth-order ENO schemes and several fifth-order WENO schemes with and without characteristic decomposition.

The implementation of a Discretization is illustrated for a second-order Muscl-Hancock scheme with a Superbee limiter and a Local Lax-Friedrichs flux in [29] and omitted here due to space constraints.

4.6 Applications

Results of a Sod-type problem are shown in figure 3. Here, the results of computations using the Local Lax-Friedrichs scheme (a first-order monotone centered method) are compared to those using a fifth-order WENO scheme (using Roe's first-order monotone flux and a characteristic decomposition). The figure illustrates the generally accepted result that proper higher-order reconstructions lead to higher resolution of shock waves, expansion fans, and contact discontinuities [34]. That is, such phenomena are smeared over fewer computational cells.

5 Conclusions

A conceptually meaningful structure for numerical gas dynamics using Modelica was introduced. The reviewed discretization schemes were implemented in the resulting framework and delivered robust and efficient simulation of the corresponding thermo-fluid dynamics problems.

References

[1] W. Casas. *Untersuchung und Optimierung sorptionsgestützter Klimatisierungsprozesse*. PhD thesis, Technical University of Hamburg-Harburg, Institute of Thermo-Fluid Dynamics, 2006.

[2] W. Casas and G. Schmitz. Experiences with a gas driven, desiccant assisted air conditioning system with geothermal energy for an office building. *Energ. Buildings.*, 37(5):493–501, 2005.

[3] F. Casella and A. Leva. Modelica open library for power plant simulation: design and experimental validation. In P. Fritzson, editor, *Proceedings*

```
partial package PartialDiscretization
  "Interface for discretization in compact flux form"

  // ...

  function primitiveToConserved
    "Convert primitive variables to conserved variables"
    input Medium.ThermodynamicState state "Thermodynamic state";
    input SI.Velocity v "Velocity";
    output Real u[3] "Vector of conserved variables";
  algorithm
    u := {Medium.density(state), Medium.density(state)*v,
      Medium.density(state)*
        (Medium.specificInternalEnergy(state) + 1/2*v*v)};
  end primitiveToConserved;

  function conservedToPrimitive
    "Convert conserved variables to primitive variables"
    input Real u[3] "Vector of conserved variables";
    output Medium.ThermodynamicState state "Thermodynamic state";
    output SI.Velocity v "Velocity";
  algorithm
    v := u[2]/u[1];
    state := Medium.setState_duX(u[1], u[3]/u[1]-1/2*v*v,
               Medium.X_default);
  end conservedToPrimitive;

  // ...

end PartialDiscretization;
```

Listing 3: Discretization interface, conversion functions

of the Third International Modelica Conference, pages 41–50, Linköping, Sweden, 2003.

[4] F. Casella and A. Leva. Modelling of thermo-hydraulic power generation processes using Modelica. *Math. Comput. Model. Dyn. Syst.*, 12(1):19–33, 2006.

[5] J. Díaz López. Shock wave modeling for Modelica.Fluid library using oscillation-free logarithmic reconstruction. In *Proceedings of the Fifth International Modelica Conference*, pages 641–649, 2006.

[6] H. Elmqvist, H. Tummescheit, and M. Otter. Object-oriented modeling of thermo-fluid systems. In P. Fritzson, editor, *Proceedings of the Third International Modelica Conference*, pages 269–286, Linköping, Sweden, 2003.

[7] R. Franke, F. Casella, M. Otter, M. Sielemann, S.-E. Mattson, H. Olsson, and H. Elmqvist. Stream connectors—an extension of Modelica for device-oriented modeling of convective transport phenomena. In F. Casella, editor, *Proceedings of the seventh International Modelica conference*, pages 108–121, Como, September 2009.

[8] T. Gallouët, J. Hérard, and N. Seguin. Some recent finite volume schemes to compute euler equations using real gas eos. *Int. J. Numer. Meth. Fl.*, 39(12):1073–1138, 2002.

[9] S. K. Godunov. A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sbornik.*, 47:357–393, 1959.

[10] A. Harten. High resolution schemes for hy-

```
partial package PartialDiscretization
  "Interface for discretization in compact flux form"

  //...

  replaceable partial  function monotoneFlux
    "First-order flux approximation"
    input Medium.ThermodynamicState state_l
      "Stencil of thermodynamic states on left (i)";
    input Medium.ThermodynamicState state_r
      "Stencil of thermodynamic states on right (i+1)";
    input SI.Velocity v_l "Velocity in x-dir on left, v_(i)";
    input SI.Velocity v_r "Velocity in x-dir on right, v_(i+1)";
    output Real flux[3] "Fluxes f_(i+1/2)";
  end monotoneFlux;

  replaceable partial  function flux "Numeric flux approximation"
    input Medium.ThermodynamicState state[stencilLength]
      "Thermodynamic state stencil";
    input SI.Velocity v[stencilLength] "Velocity stencil";
    input Real x_side[stencilLength + 1]
      "Coordinates of cell sides (i-1/2), (i+1/2) etc.";
    output Real flux[3] "Fluxes f_(i+1/2)";
  end flux;

  //...

end PartialDiscretization;
```

Listing 4: Discretization interface, flux functions

perbolic conservation laws. *J. Comput. Phys.*, 49:357–393, 1983.

[11] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high order essentially non-oscillatory schemes, III. *J. Comput. Phys.*, 71:231–303, 1987.

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

[13] T. Y. Hou and P. LeFloch. Why non-conservative schemes converge to the wrong solutions: Error analysis. *Math. Comput.*, 62:497–530, 1994.

[14] J. Jensen, J. Jensen, and H. Tummescheit. Moving boundary models for dynamic simulations of two-phase flows. In *Proceedings of the Second International Modelica Conference*, 2002.

[15] J. M. Jensen. *Dynamic Modeling of Thermo-Fluid Systems with focus on evaporators for refrigeration*. PhD thesis, Technical University of Denmark, Department of Mechanical Engineering, 2003.

[16] G. Jiang and C.-W. Shu. Effcient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126:202–228, 1996.

[17] P. D. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure Appl. Math.*, 13:217–237, 1960.

[18] M. Liou, B. Leer, and J. Shuen. Splitting of inviscid fluxes for real gases. *J. Comput. Phys.*, 87(1):1–24, 1990.

[19] S. Patankar and D. Spalding. A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *Int. J. Heat. Mass. Tran.*, 15:1787–1806, 1972.

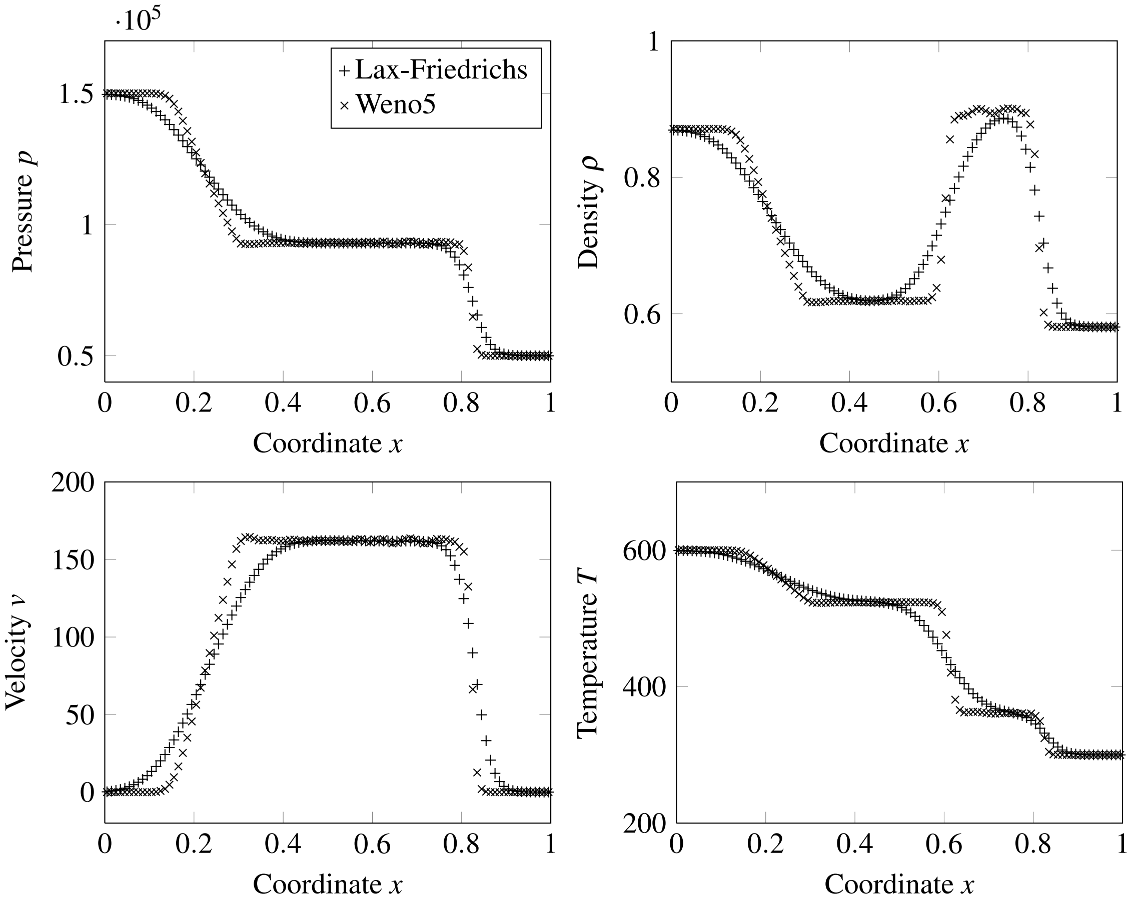

Figure 3: Comparison of Local Lax-Friedrichs and fifth-order WENO schemes on a Sod-type problem

[20] T. Pfafferott. *Dynamische Simulation von CO2-Kälteprozessen für mobile Anwendungen*. PhD thesis, Technical University of Hamburg-Harburg, Institute of Thermo-Fluid Dynamics, 2005.

[21] K. Prölß. *Untersuchung von Energie- und Massespeicherungsvorgängen in Pkw-Kälteanlagen*. PhD thesis, Technical University of Hamburg-Harburg, Institute of Thermo-Fluid Dynamics, 2009.

[22] K. Prölßand G. Schmitz. Modeling of frost growth on heat exchanger surfaces. In *Proceedings of the Fifth International Modelica Conference*, 2006.

[23] J. J. Quirk. An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two dimensional bodies. *Comput. Fluid.*, 23(1):125–142, 1994.

[24] C. C. Richter. *Proposal of New Object-Oriented Equation-Based Model Libraries for Thermodynamic Systems*. PhD thesis, Technical University Braunschweig, Institute for Thermodynamics, 2008.

[25] R. D. Richtmyer and K. W. Morton. *Difference Methods for Initial Value Problems*. Interscience-Wiley, New York, 1967.

[26] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43:357–372, 1981.

[27] V. V. Rusanov. Calculation of interaction of non-steady shock waves with obstacles. *USSR J. Comput. Math. Phys.*, 1:267–279, 1961.

[28] C.-W. Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. *Advanced numerical approximation of nonlinear hyperbolic equations*, 1697:325–432, 1998.

[29] M. Sielemann. *Device-Oriented Modeling and Simulation in Aircraft Energy Systems Design*. PhD thesis, Technical University of Hamburg-Harburg, Institute of Thermo-Fluid Dynamics, 2012.

[30] J. L. Steger and R. F. Warming. Flux vector splitting of the inviscid gasdynamic equations with applications to finite difference methods. *J. Comput. Phys.*, 40:263–293, 1981.

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

[32] P. K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM J. Numer. Anal.*, 21:995–1011, 1984.

[33] E. F. Toro. On two Glimm-related schemes for hyperbolic conservation laws. In *Proceedings of the Fifth Annual Conference of the CFD Society of Canada*, pages 3.49–3.54. University of Victoria, Canada, 1997.

[34] E. F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer, 1997.

[35] H. Tummescheit. *Design and Implementation of Object-Oriented Model Libraries using Modelica*. PhD thesis, Lund University, Department of Automatic Control, 2002.

[36] H. Tummescheit, J. Eborn, and K. Prölß. Airconditioning–a Modelica library for dynamic simulation of AC systems. In G. Schmitz, editor, *Proceedings of the Fourth International Modelica Conference*, Hamburg, Germany, 2005.

[37] B. van Leer. Towards the ultimate conservative difference scheme I. the quest for monotonicity. *Lect. Notes. Phys.*, 18:163–168, 1973.

[38] B. van Leer. Towards the ultimate conservative difference scheme II. monotonicity and conservation combined in a second order scheme. *J. Comput. Phys.*, 14:361–370, 1974.

[39] B. van Leer. Towards the ultimate conservative difference scheme III. upstream-centered finite difference schemes for ideal compressible flow. *J. Comput. Phys.*, 23:263–275, 1977.

[40] J. Vasel and G. Schmitz. Transient simulation of a direct-evaporating CO2 cooling system for an aircraft. In *25th International Congress of the Aeronautical sciences (ICAS), Proceedings of the*, Hamburg, Germany, September 2006.