





Review

# Convergence towards High-Speed Steady States Using High-Order Accurate Shock-Capturing Schemes

Juan C. Assis <sup>1</sup>, Ricardo D. Santos <sup>1</sup>, Mateus S. Schuabb <sup>2</sup>, Carlos E. G. Falcão <sup>3</sup>, Rômulo B. Freitas <sup>4</sup>  
and Leonardo S. de B. Alves <sup>3,\*</sup>

<sup>1</sup> Programa de Pós-Graduação em Engenharia Mecânica, Universidade Federal Fluminense, Niterói 24210-240, RJ, Brazil

<sup>2</sup> Department of Mechanical and Aerospace Engineering, The Ohio State University, Columbus, OH 43210, USA

<sup>3</sup> Departamento de Engenharia Mecânica, Universidade Federal de Santa Maria, Santa Maria 97105-900, RS, Brazil

<sup>4</sup> Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Nova Iguaçu 26041-271, RJ, Brazil

\* Correspondence: lsbalves@id.uff.br

**Abstract:** Creating time-marching unsteady governing equations for a steady state in high-speed flows is not a trivial task. Residue convergence in time cannot be achieved when using most low- and high-order spatial discretization schemes. Recently, high-order, weighted, essentially non-oscillatory schemes have been specially designed for steady-state simulations. They have been shown to be capable of achieving machine precision residues when simulating the Euler equations under canonical coordinates. In the present work, we review these schemes and show that they can also achieve machine residues when simulating the Navier–Stokes equations under generalized coordinates. This is carried out by considering three supersonic flows of perfect fluids, namely the flow upstream a cylinder, the flow over a blunt wedge, and the flow over a compression ramp.

**Keywords:** TVD schemes; WENO schemes; residue convergence; increment convergence



**Citation:** Assis, J.C.; Santos, R.D.; Schuabb, M.S.; Falcão, C.E.G.; Freitas, R.B.; Alves, L.S.d.B. Convergence towards High-Speed Steady States Using High-Order Accurate Shock-Capturing Schemes. *Fluids* **2024**, *9*, 133. <https://doi.org/10.3390/fluids9060133>

Academic Editors: D. Andrew S. Rees and Giuliano De Stefano

Received: 4 April 2024

Revised: 17 May 2024

Accepted: 27 May 2024

Published: 1 June 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The high-speed flows under discussion in this work are modeled by unsteady and compressible Navier–Stokes equations for perfect fluids. These governing equations are solved using the method of lines. This represents a two-step simulation process that first generates a system of ordinary differential equations from the partial differential equations composing the original model governing equations and then marches it forward in time. A wide range of methods can be applied to generate this system. A review of them is beyond the scope of this work, but several relevant books can be used as a starting point [1–5]. This system is obtained here by applying finite-difference schemes under a generalized coordinate framework to the spatial operators of the model. The code developed to run all the simulations presented here is called 3D4S [6,7]. It stands for one-, two-, and three-dimensional structured steady-state solvers and uses the C++ programming language [8].

### 1.1. Literature Review

#### 1.1.1. Spatial Discretization

Apart from unconventional source terms that are not employed in the present work, all spatial operators in the unsteady and compressible Navier–Stokes equations for perfect fluids are applied to inviscid and viscous fluxes. Their discretizations are usually treated using different methodologies, whose implementations are briefly discussed below.

On the one hand, the discretization of viscous flux spatial gradients employs the traditional second-, fourth-, and sixth-order central-difference schemes in conservative form. It is important to note that the third one usually renders the resulting code too numerically unstable [9], and hence, it is not used here.

On the other hand, an accurate discretization of inviscid fluxes that can capture shocks is significantly more complex. This is due to the nonlinear stability properties that are required to prevent the Gibbs phenomenon near discontinuities. Monotone schemes are capable of doing so. This property states that (i) new minima and maxima cannot be created, (ii) a local minimum cannot be reduced, and (iii) a local maximum cannot be increased. They are, however, first-order accurate. High-order accuracy requires the use of special techniques to control the oscillations induced by discretizing across the shock. One possibility is to relax the monotonicity requirement for something else less restrictive. Following this path [10] leads to second-order accurate schemes that are total-variation diminishing (TVD). This property states that the solution's total variation at a given time step is always smaller than or equal to the solution's total variation at the previous time step. These schemes also satisfy the entropy condition, which guarantees the convergence of weak solutions of hyperbolic conservation laws. Furthermore, all monotone schemes are TVD and all TVD schemes are monotonicity-preserving [11]. This property states that a solution that is monotone at a given time step will remain monotone at the next time step. Furthermore, TVD schemes work by employing flux-limiters, which are weights placed on the forward and backward fluxes. They are designed to locally reduce the scheme to first-order accurate near discontinuities [12]. There are many flux limiters. Superbee [13] and Minmod [14–16] are the ones that introduce the least and most numerical diffusion, respectively. All others fall in between these two, such as the Monotonized Central Difference [17], van Leer [18], Koren [19], Sweby [20], Osher [21], Ospre [22] and van Albada [23]. 3D4S can use either the first-order accurate upwind/downwind discretization or the second-order accurate TVD scheme developed by [10], coupled with any of these flux limiters.

Unfortunately, it is well known that TVD schemes suffer from an accuracy order reduction near discontinuities [18,24,25]. Furthermore, they are first-order accurate for two-dimensional problems [26]. A path towards higher-order accuracy is to relax the TVD requirement for something that is less restrictive. This was performed by introducing the class of essentially non-oscillatory (ENO) schemes [27]. They represent higher-order generalizations of the first-order Godunov scheme [28] and its second-order extension known as the Monotonic Upstream-centered Scheme for Conservation Laws (MUSCL) [29]. ENO schemes choose between different candidate stencils in order to avoid including discontinuity in the discretization. The selection is performed by comparing the local smoothness of the discretization on these different stencils, measured using divided differences [30,31]. A variation of these schemes, known as weighted ENO (WENO) schemes, was then developed to achieve higher-order accuracy and smoother numerical fluxes [32]. Instead of using only one of the candidate stencils, as in ENO, WENO uses a convex combination of all candidate stencils based on nonlinear weights determined by smoothness indicators. 3D4S can use several different fifth-order WENO schemes for unsteady simulations, such as the original WENO5-JS [33], as well as WENO5-M [34], WENO5-Z [35] and WENO5-Z+ [36], and steady simulations, such as WENO5-ZQ [37] and its variants [38–41].

A distinguishing feature of the aforementioned upwind/downwind schemes is that their discretization must be performed according to the propagation direction of the inviscid flux characteristics. There are essentially two different ways to identify these directions, which are known as the flux-difference and vector-splitting methods [42]. The former are known as Riemann solvers whereas the latter are known as Boltzmann solvers, although they can also be considered Riemann solvers in a wider sense. On one hand, flux-difference splitting first discretizes the inviscid fluxes using their cell-faced values and only then separates their propagation directions to evaluate these values. Flux-vector splitting, on the other hand, first separates the inviscid fluxes according to their propagation directions and only then discretizes each part accordingly. It is important to note that they can also be categorized as either complete or incomplete solvers. The former resolves all waves that are present, whereas the latter neglects some of these waves. 3D4S can currently employ many of these methods, such as Roe [43], HLL [44] and Roe-HLL [45] based on

flux-difference splitting as well as Steger–Warming [46] and Lax–Friedrichs [33] based on flux-vector splitting, which are applied in either conservative or characteristic form.

### 1.1.2. Temporal Discretization

Having applied the spatial discretization of choice on a given grid to the unsteady and compressible Navier–Stokes equations for perfect fluids, the resulting system can then be marched in time. This is performed in 3D4S using different methodologies to generate either time-accurate unsteady or steady states. Since physically stable flows are studied in this work, explicit time-marching schemes can be employed to reach their steady states.

The explicit Euler scheme is a good choice for time-marching the unsteady governing equations towards time-asymptotic stable steady states not only due to its simplicity but also due to its parallelization efficiency. This is the approach used when employing 3D4S in this scenario. When time accuracy is required for shock-capturing, however, high-order temporal integration should be used. Furthermore, this should be performed in such a way as to preserve the nonlinear numerical stability properties of the chosen spatial discretization schemes. All time-marching schemes with this property are known today as Strong-Stability-Preserving (SSP), although they were originally called TVD as well because this property does not allow the solution’s total variation to increase in time. High-order time-marching schemes with this property preserve the nonlinear numerical stability properties of the Euler scheme, as long as their maximum time step is restricted. The SSP property was first proposed for an explicit Runge–Kutta (RK) scheme [30]. Although this SSP-RK scheme was originally presented in a format that did not belong to the traditional Butcher formulation [5], it can be written in this way using appropriate algebraic procedures [47,48]. The nonlinear upper bound imposed on the maximum time step of high-order SSP marching schemes, which is smaller than its linear counterpart, led to the optimization of these schemes so they can achieve the largest possible maximum time steps. Optimal explicit SSP-RK schemes have been derived with up to fourth-order accuracy and five stages [49]. They are the ones used in 3D4S for time-accurate simulations.

It is important to mention here that implicit SSP-RK schemes exist as well [50], where optimal diagonally implicit schemes have been developed with up to fourth-order accuracy and eight-stages [51,52] and with up to sixth-order accuracy and eleven stages [53]. Second- and third-order accurate implicit–explicit (IMEX) RK schemes also exist [54], but only the explicit part is SSP. However, there is strong evidence that the nonlinear upper bound imposed on the time steps of implicit and IMEX SSP-RK schemes renders them inefficient when compared to their explicit counterparts [55]. Hence, they are not used in 3D4S for time-accurate simulations.

### 1.1.3. Convergence towards Steady States

As far as the authors are aware, the effect that high-order WENO schemes have on convergence towards steady-state was first evaluated almost two decades ago [56]. This study shed light on the existence of small-amplitude post-shock oscillations, which prevented residue convergence to machine precision. It also led to the development of WENO schemes specially designed for steady-state simulations [37–40]. The use of these schemes did lead to residue convergence towards machine precision zero by reducing the amplitude of post-shock oscillations.

## 1.2. Present Contributions

Nevertheless, there are still issues that must be addressed on this front. For instance, steady-state convergence in the aforementioned studies was evaluated when considering the Euler equations, which do not include viscous effects. Furthermore, the test cases simulated in these studies considered quite simple geometries; i.e., they could be simulated using standard coordinate systems. Both issues are addressed in the present paper, namely the ability to achieve steady-state convergence to machine precision residues when using

WENO schemes (i) in the presence of viscous diffusion and (ii) when performing numerical simulations in generalized coordinates.

In order to do so, a comparison of several shock-capturing schemes is performed using the Navier–Stokes equations over a framework with generalized coordinates. In Section 2, the methodology used to implement the numerical discretization of both spatial and temporal terms of the governing equations is discussed. It also contains a brief review of the grid-generation techniques employed here. In Section 3, the results obtained for three different test cases are presented, namely the two-dimensional supersonic flow over a compression ramp, over a planar blunt body and upstream of a cylinder. Finally, the major conclusions derived from this work are highlighted in Section 4.

## 2. Methodology

### 2.1. Governing Equations

Using a Cartesian coordinate system, it is possible to write the Navier–Stokes equations in dimensionless form as

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{E}^i}{\partial x} + \frac{\partial \mathbf{F}^i}{\partial y} + \frac{\partial \mathbf{G}^i}{\partial z} = \frac{\partial \mathbf{E}^v}{\partial x} + \frac{\partial \mathbf{F}^v}{\partial y} + \frac{\partial \mathbf{G}^v}{\partial z} \quad , \quad (1)$$

where  $\mathbf{Q}$  is the vector of conservative variables and  $\mathbf{E}$ ,  $\mathbf{F}$  and  $\mathbf{G}$  are the conservative flux vectors in the  $x$ ,  $y$ , and  $z$  directions, respectively. The superscripts  $i$  and  $v$  stand for inviscid and viscous, respectively. These vectors are given by

$$\mathbf{Q} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{bmatrix} \quad , \quad \mathbf{E}^i = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho v u \\ \rho w u \\ (\rho E + p)u \end{bmatrix} \quad , \quad \mathbf{F}^i = \begin{bmatrix} \rho v \\ \rho v u \\ \rho v^2 + p \\ \rho w v \\ (\rho E + p)v \end{bmatrix} \quad , \quad \mathbf{G}^i = \begin{bmatrix} \rho w \\ \rho u w \\ \rho v w \\ \rho w^2 + p \\ (\rho E + p)w \end{bmatrix} \quad ,$$

$$\mathbf{E}^v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u \tau_{xx} + v \tau_{xy} + w \tau_{xz} - q_x \end{bmatrix} \quad , \quad \mathbf{F}^v = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u \tau_{yx} + v \tau_{yy} + w \tau_{yz} - q_y \end{bmatrix} \quad \text{and} \quad (2)$$

$$\mathbf{G}^v = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u \tau_{zx} + v \tau_{zy} + w \tau_{zz} - q_z \end{bmatrix} \quad .$$

where the viscous stress tensor is defined by

$$\tau_{xx} = \Pi_1 \left[ 2\mu \frac{\partial u}{\partial x} + \lambda \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \right] \quad , \quad \tau_{xy} = \tau_{yx} = \Pi_1 \left[ \mu \left( \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \right] \quad ,$$

$$\tau_{yy} = \Pi_1 \left[ 2\mu \frac{\partial v}{\partial y} + \lambda \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \right] \quad , \quad \tau_{xz} = \tau_{zx} = \Pi_1 \left[ \mu \left( \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \right) \right] \quad \text{and} \quad (3)$$

$$\tau_{zz} = \Pi_1 \left[ 2\mu \frac{\partial z}{\partial x} + \lambda \left( \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) \right] \quad \text{and} \quad \tau_{yz} = \tau_{zy} = \Pi_1 \left[ \mu \left( \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \right) \right] \quad ,$$

and the heat flux is defined by

$$q_x = -\Pi_2 k \frac{\partial T}{\partial x} \quad , \quad q_y = -\Pi_2 k \frac{\partial T}{\partial y} \quad \text{and} \quad q_z = -\Pi_2 k \frac{\partial T}{\partial z} \quad . \quad (4)$$

The aforementioned system of equations has more variables than equations. However, it is possible to employ the equation of state for an ideal gas, i.e.,

$$p = -\Pi_3 \rho T \quad , \quad (5)$$

where  $\Pi_k$  represents the dimensionless parameters, calculated with respect to the reference quantities. They can be written in generic form as follows:

$$\Pi_1 = \frac{\mu_R}{\rho_R U_R L_R} \quad , \quad \Pi_2 = \frac{k_R T_R}{\rho_R U_R^3 L_R} \quad \text{and} \quad \Pi_3 = \frac{RT_R}{U_R^2} \quad . \quad (6)$$

The reference quantities, on the other hand, are selected based on the free stream conditions. However, an exception is made for the reference velocity, which is derived from the free stream's speed of sound. Consequently, the dimensionless parameters become

$$\Pi_1 = \frac{Ma_\infty}{Re_\infty} \quad , \quad \Pi_2 = \frac{Ma_\infty}{(\gamma - 1) Re_\infty Pr_\infty} \quad \text{and} \quad \Pi_3 = \frac{1}{\gamma} \quad , \quad (7)$$

where  $Re$  is the Reynolds number,  $Pr$  is the Prandtl number,  $Ma$  is the Mach number, and the subscript  $\infty$  indicates free stream quantities. Having selected these three dimensionless parameters, one must still select the dynamic viscosity and thermal conductivity. The dimensionless dynamic viscosity is assumed to follow Sutherland's law, i.e.,

$$\mu(T) = \frac{1 + T_0}{T + T_0} T^{3/2} \quad , \quad (8)$$

and the dimensionless thermal conductivity is computed assuming that the Prandtl number is constant  $Pr = Pr_\infty$ . Hence, one obtains

$$\mu(T) = k(T) \quad . \quad (9)$$

since both heat capacities are constant in ideal gases.

The numerical methods utilized here were originally designed for rectangular domains. In order to use them in complex geometries, a coordinate transformation is applied to the governing equations to map the generalized physical space  $(x, y, z)$  to the rectangular computational space  $(\xi, \eta, \zeta)$ . It must be noted that this transformation ensures that the transformed equations retain their conservative behavior. Within the computational space, the governing equations are reformulated as

$$\frac{\partial q}{\partial t} + \frac{\partial e^i}{\partial \xi} + \frac{\partial f^i}{\partial \eta} + \frac{\partial g^i}{\partial \zeta} = \frac{\partial e^v}{\partial \xi} + \frac{\partial f^v}{\partial \eta} + \frac{\partial g^v}{\partial \zeta} \quad , \quad (10)$$

noting that the new coordinate system depends on the old one, i.e.,

$$\xi = \xi(x, y, z) \quad , \quad \eta = \eta(x, y, z) \quad \text{and} \quad \zeta = \zeta(x, y, z) \quad , \quad (11)$$

and, hence, the transformation Jacobian

$$\mathbf{J} = \frac{\partial(\xi, \eta, \zeta)}{\partial(x, y, z)} = \begin{bmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix} \quad , \quad (12)$$

has a nonzero determinant, i.e.,

$$J = \frac{1}{J^{-1}} = 1 \left/ \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)} \right. = 1 \left/ \begin{vmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{vmatrix} \right. \quad . \quad (13)$$

and the necessary metrics are computed using

$$\begin{bmatrix} \tilde{\zeta}_x & \tilde{\zeta}_y & \tilde{\zeta}_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{bmatrix} = J \begin{bmatrix} y_\eta z_\zeta - y_\zeta z_\eta & -(x_\eta z_\zeta - x_\zeta z_\eta) & x_\eta y_\zeta - x_\zeta y_\eta \\ -(y_\zeta z_\zeta - y_\zeta z_\zeta) & x_\zeta z_\zeta - x_\zeta z_\zeta & -(x_\zeta y_\zeta - x_\zeta y_\zeta) \\ y_\zeta z_\eta - y_\eta z_\zeta & -(x_\zeta z_\eta - x_\eta z_\zeta) & x_\zeta y_\eta - x_\eta y_\zeta \end{bmatrix} , \quad (14)$$

leading to the transformed conservative variables and fluxes

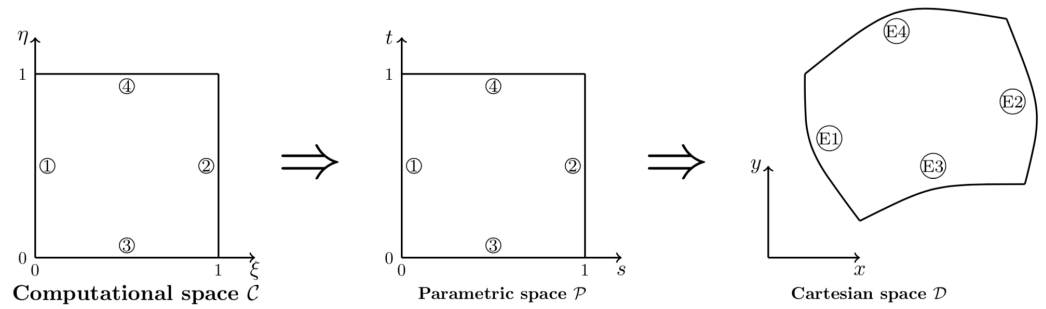
$$\begin{aligned} \mathbf{q} &= J^{-1} \mathbf{Q} \quad , \\ \mathbf{e}^i &= J^{-1} \left( E^i \tilde{\zeta}_x + F^i \tilde{\zeta}_y + G^i \tilde{\zeta}_z \right) \quad , \quad \mathbf{f}^i = J^{-1} \left( E^i \eta_x + F^i \eta_y + G^i \eta_z \right) \quad , \\ \mathbf{g}^i &= J^{-1} \left( E^i \zeta_x + F^i \zeta_y + G^i \zeta_z \right) \quad , \quad \mathbf{e}^v = J^{-1} \left( E^v \tilde{\zeta}_x + F^v \tilde{\zeta}_y + G^v \tilde{\zeta}_z \right) \quad , \\ \mathbf{f}^v &= J^{-1} \left( E^v \eta_x + F^v \eta_y + G^v \eta_z \right) \quad \text{and} \quad \mathbf{g}^v = J^{-1} \left( E^v \zeta_x + F^v \zeta_y + G^v \zeta_z \right) \quad . \end{aligned} \quad (15)$$

### 2.2. Grid Generation

All spatial discretization schemes described so far are implemented on a generalized coordinate framework using uniformly distributed points between 0 and 1 in all directions. Hence, the physical domain of interest must be mapped into this computational domain for all simulations. Analytical transformations are employed whenever they are available. If this is not the case, grid-generation tools are required. This is an important issue because grid quality is directly related to solution accuracy. For example, lack of convergence can be a consequence of poor grid quality [2].

There are two fundamental types of grids for multidimensional regions: structured and unstructured. They differ by the way in which the grid points are locally organized. Structured grids have regular connectivity, which is implicitly taken into account. They have quadrilateral elements in two-dimensional domains or hexahedron elements in three-dimensional ones. On the other hand, unstructured grids can be identified by irregular connectivity that must be explicitly described and stored. This type of grid quite often employs triangles and tetrahedral elements, respectively, in two- and three-dimensional problems [57]. Besides that, they also differ in the type of method that can use them. While finite difference methods require structured grids, finite volume and finite element methods allow both types of grids. Since 3D4S is based on finite differences schemes, this brief review focuses only on structured grid generation tools.

Since elliptic partial differential equations were introduced for grid generation [58], this approach has established itself as the one that arguably produces the best grids in terms of smoothness and grid point distribution. When these elliptic differential equations are homogeneous, the smoothness of the Laplace operator makes the grid evenly spaced throughout the domain. This makes grid refinement at specific locations, such as the wall for boundary-layer problems, more difficult. Hence, grids based solely on Laplace equations are unusable in practice. However, Poisson equations can be used instead so that control functions can be introduced through their source terms. The quality of an elliptic grid now depends on how well these control functions can be tailored [59]. They are usually employed to control grid clustering [60] as well as enforce orthogonality at boundaries [61]. The latter is essential to avoid coupling between wall grid points when wall-normal derivatives are required. The particular grid-generation procedure used in 3D4S [62] introduces an intermediate step between physical and computational spaces [63] so that both grid clustering and orthogonality can be enforced. Figure 1 shows an illustration of its mapping procedure.



**Figure 1.** Transformation from computational  $(\xi, \eta)$  space to a physical domain in Cartesian  $(x, y)$  space going through a parametric  $(s, t)$  space.

2.3. Numerical Methods

2.3.1. Viscous Flux Discretization

In the 3D4S implementation, standard central differences are applied to discretize the viscous fluxes. To illustrate this approach, consider the scalar viscous flux

$$\frac{\partial f}{\partial x} = \frac{\partial}{\partial x} \left( \mu \frac{\partial u}{\partial x} \right) \quad , \quad (16)$$

which can be approximated using a generic central difference scheme,

$$\left. \frac{\partial f}{\partial x} \right|_i = \bar{\delta}_x^{(p)} f_i + \mathbf{O}(\Delta x^p) \quad , \quad (17)$$

also applied to its scalar flux, i.e.,

$$f_i = \mu_i \bar{\delta}_x^{(p)} u_i + \mathbf{O}(\Delta x^p) \quad , \quad (18)$$

where  $p$  is the scheme accuracy order and  $\delta_x$  is the central-difference operator, which can be developed for any order  $p$ . For instance, the second-order ( $p = 2$ ) operator is

$$\bar{\delta}_x^{(2)} f_i = \frac{f_{i+1} - f_{i-1}}{2 \Delta x} \quad , \quad (19)$$

the fourth-order ( $p = 4$ ) operator is

$$\bar{\delta}_x^{(4)} f_i = \frac{f_{i-2} - 8f_{i-1} + 8f_{i+1} - f_{i+2}}{12 \Delta x} \quad , \quad (20)$$

and the sixth-order ( $p = 6$ ) operator is

$$\bar{\delta}_x^{(6)} f_i = \frac{-f_{i-3} + 9f_{i-2} - 45f_{i-1} + 45f_{i+1} - 9f_{i+2} + f_{i+3}}{12 \Delta x} \quad . \quad (21)$$

Introducing Equation (18) into Equation (17) yields

$$\left. \frac{\partial f}{\partial x} \right|_i = \delta_x (\mu_i \delta_x u_i + \mathbf{O}(\Delta x^q)) + \mathbf{O}(\Delta x^p) = \delta_x (\mu_i \delta_x u_i) + \mathbf{O}(\Delta x^p, \Delta x^{q-1}) \quad , \quad (22)$$

which means that the latter operation in Equation (22) can lead to odd–even decoupling as well as order loss whenever  $\mu$  is not a smooth enough function [9]. Hence, high-order operators, such as the one in Equation (21), must be used with care.

Figure 2 illustrates the stencil dependence associated with the central-difference schemes discussed previously. It is important to recognize that, as the order of the scheme increases, so does the number of points required for the approximation. Near boundary schemes must be biased upwind near the inlet and downwind near the outlet due to the lack of external grid points. Generally, the scheme’s accuracy near boundaries is reduced



to maintain numerical stability. Nevertheless, the overall accuracy can be preserved by ensuring that the boundary order is one less than the inner domain order [64].

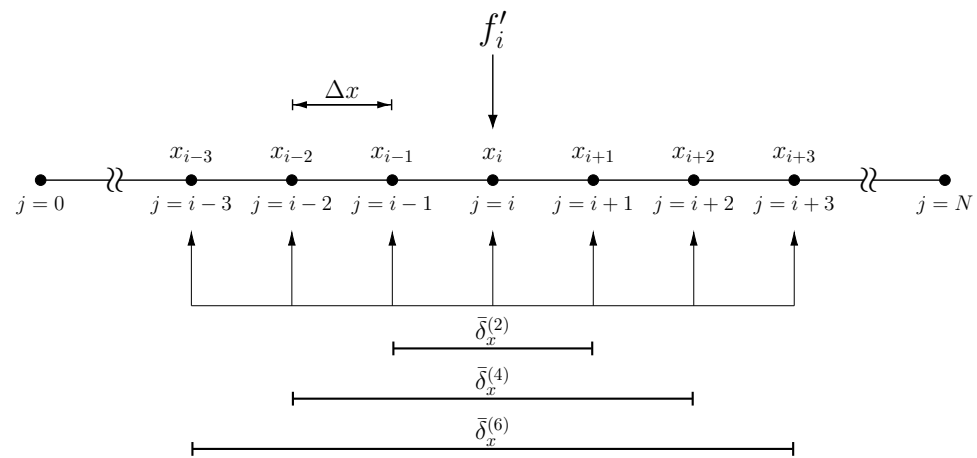


Figure 2. Stencil dependence of central-difference operators.

### 2.3.2. Inviscid Flux Discretization

The discretization of inviscid fluxes in 3D4S is significantly more complex. It is described using the scalar hyperbolic conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0 \quad , \tag{23}$$

where  $x$  is the position in a one-dimensional space,  $t$  is time, and  $u = u(x, t)$  and  $f = f(u(x, t))$  are the conservative variable and the conservative flux function, respectively. The goal is to describe the evolution of  $u$  within  $t \in [0, \infty]$  and  $x \in [x_0, x_N]$ .

#### Conservative Schemes

A numerical scheme used to solve Equation (23) is called conservative if it ensures that  $u$  is conserved across all cells. Thus, numerical conservation is achieved using a single flux function that describes the flow of  $u$  between neighboring cells. Using a semi-discretized form of Equation (23), namely

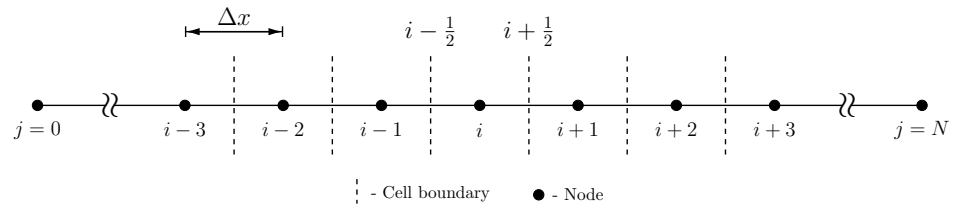
$$\frac{du}{dt} \Big|_i = - \frac{\partial f}{\partial x} \Big|_i \quad , \tag{24}$$

a conservative finite-difference scheme used to approximate the right-hand-side term of Equation (24) must be written as

$$\frac{df}{dx} \Big|_i = \frac{h_{i+\frac{1}{2}} - h_{i-\frac{1}{2}}}{\Delta x} \quad , \tag{25}$$

where  $h_{i+1/2} = h(x = x_{i+1/2})$  is a numerical flux function describing the flux crossing the interface  $x_{i+1/2}$  between  $x_i$  and  $x_{i+1}$ . This can be visualized in Figure 3, which shows an illustration of a typical computational grid. The numerical flux function is defined such that Equation (25) involves no truncation error. Traditionally [31], the numerical flux function has been implicitly defined as





**Figure 3.** Illustration of a typical computational grid.

$$f(x) = \frac{1}{\Delta x} \int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} h(\xi) d\xi \quad , \quad (26)$$

which leads, after integration, to

$$f(x) = \frac{H(x + \frac{\Delta x}{2}) - H(x - \frac{\Delta x}{2})}{\Delta x} \quad , \quad (27)$$

where  $H(x)$  is the primitive of  $h(\xi)$ , i.e.,  $H(x) = \int_{-\infty}^x h(\xi) d\xi$ . Furthermore, taking the derivative of Equation (27) and evaluating it at  $x_i$  results in Equation (25). One must note that the numerical flux function is not a numerical approximation of the flux function, even though the flux function derivative is exactly equal to a finite difference approximation of the numerical flux function. Unfortunately, it is not possible to obtain the exact numerical flux function, and, hence, a numerical approximation must be used. High-order conservative numerical schemes can be constructed by employing high-order approximations of the numerical flux function. Defining  $\hat{f}(x)$  as this arbitrary polynomial approximation of  $h(x)$ , its coefficients can be determined from the flux function by replacing  $h(\xi)$  by  $\hat{f}(\xi)$  in Equation (26). Hence, a numerical approximation of Equation (25) can be written as

$$\left. \frac{df}{dx} \right|_i \approx \frac{\hat{f}_{i+\frac{1}{2}} - \hat{f}_{i-\frac{1}{2}}}{\Delta x} \quad . \quad (28)$$

### Flux Function Splitting

The previous discussion did not take into account the propagation direction, i.e.,  $\partial f / \partial u$ , of variable  $u$ . In the particular case of a nonlinear flux function, the variable  $u$  can propagate both downstream, where  $\partial f / \partial u > 0$ , and upstream, where  $\partial f / \partial u < 0$ , simultaneously. Hence, the flux function must be split into positive and negative parts, since they require different spatial discretization techniques. In other words,

$$f = f^+ + f^- \quad \text{with} \quad \frac{\partial f^+}{\partial u} \geq 0 \quad \text{and} \quad \frac{\partial f^-}{\partial u} \leq 0 \quad , \quad (29)$$

which allows the numerical flux function to be computed as

$$\hat{f} = \hat{f}^+ + \hat{f}^- \quad . \quad (30)$$

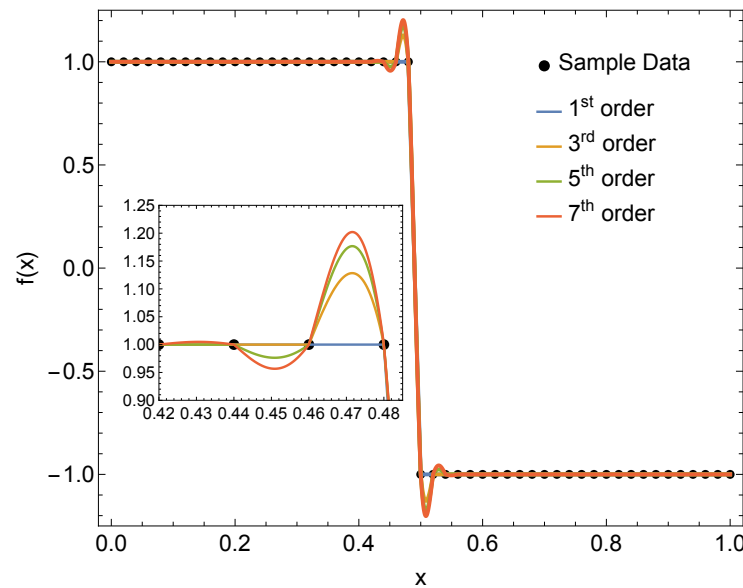
The Lax–Friedrichs flux splitting is a simple and inexpensive example of flux splitting. It splits the fluxes according to

$$f^\pm = \frac{1}{2}(f \pm \alpha u) \quad \text{with} \quad \alpha = \max_u \left| \frac{\partial f}{\partial u} \right| \quad . \quad (31)$$

### WENO Schemes

In the vicinity of discontinuities, high-order polynomial interpolations inherently exhibit oscillatory behavior. This is illustrated in Figure 4, which shows spurious oscillations from high-order polynomial fits across a step function discontinuity. Shock waves present in high-speed compressible flows are an example of such a discontinuity, which leads to

spurious oscillations in simulations employing high-order schemes. They not only lead to accuracy loss but also negatively impact the numerical stability of said schemes.



**Figure 4.** Polynomial interpolations of data taken from a step function.

Weighted Essentially Non-Oscillatory (WENO) schemes were developed to prevent such oscillations while preserving the high-order accuracy of the approximation. The foundational principle of a WENO scheme is its use of a convex combination of numerical flux approximations from lower-order polynomials to construct a high-order approximation. In smooth regions, WENO schemes become upstream central schemes. Near discontinuities, on the other hand, the stencil that includes the discontinuity is assigned a lower weight in the convex combination, minimizing its impact.

The WENO reconstruction [33] can be written as

$$\hat{f}_{i+1/2} = \sum_{k=0}^{r-1} \omega_k \hat{f}_{i+1/2}^k \quad , \quad (32)$$

where  $r$  is the number of stencils candidates,  $\hat{f}_{i+1/2}^k$  is the numerical flux approximation using the  $k$ th stencil, and  $\omega_k$  is the nonlinear weight given to the  $k$ th stencil. Figure 5 illustrates the classical fifth-order WENO-JS [33], where each  $\hat{f}_{i+1/2}^k$  is formulated as a third-order polynomial approximation of the numerical flux function using different stencils. Following this approach, the numerical flux approximations based on upstream central schemes are given by

$$\begin{aligned} \hat{f}_{i+1/2}^0 &= \frac{1}{6}(2f_{i-2} - 7f_{i-1} + 11f_i) \quad , \\ \hat{f}_{i+1/2}^1 &= \frac{1}{6}(-f_{i-1} + 5f_i + 2f_{i+1}) \quad \text{and} \\ \hat{f}_{i+1/2}^2 &= \frac{1}{6}(2f_i + 5f_{i+1} - 1f_{i+2}) \quad , \end{aligned} \quad (33)$$

and the weights  $\omega_k$  in WENO-JS are given by

$$\omega_k^{(JS)} = \frac{\alpha_k}{\sum_{n=0}^{r-1} \alpha_n} \quad \text{with} \quad \alpha_k = \frac{\bar{\omega}_k}{(\epsilon + IS_k)^p} \quad , \quad (34)$$

where  $\epsilon = 10^{-6}$  is used to prevent division by zero,  $IS_k$  is the indicator of smoothness for the  $k$ th stencil,  $\bar{\omega}_k$  is the linear weight, and  $p$  is a constant used to accelerate the rate at which  $\omega_k^{(JS)}$  goes to zero in non-smooth regions. On one hand, the linear weighted

combination of numerical flux approximations must yield a fifth-order upstream central scheme, i.e.,

$$\bar{\omega}_0 \hat{f}_{i+\frac{1}{2}}^0 + \bar{\omega}_1 \hat{f}_{i+\frac{1}{2}}^1 + \bar{\omega}_2 \hat{f}_{i+\frac{1}{2}}^2 = \frac{1}{60} (2f_{i-2} - 13f_{i-1} + 47f_i + 27f_{i+1} - 3f_{i+2}) \quad , \quad (35)$$

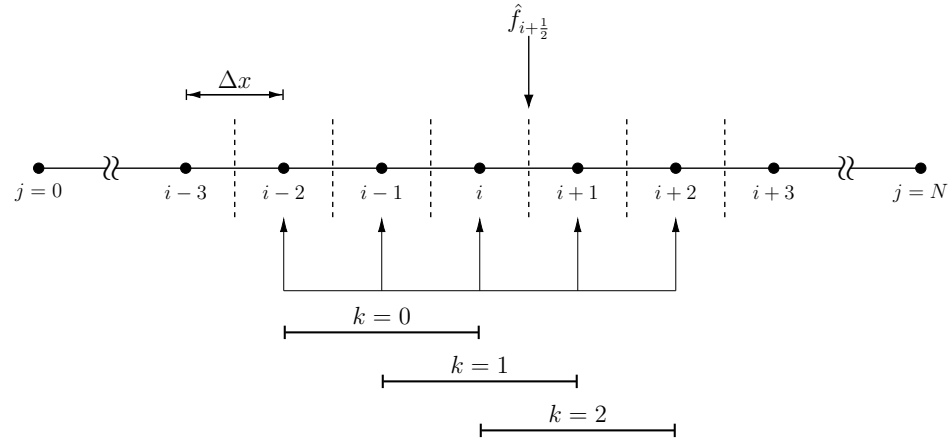


Figure 5. Fifth-Order WENO-JS scheme stencil dependence.

and, hence, the linear weights must be

$$\bar{\omega}_0 = \frac{1}{10} \quad , \quad \bar{\omega}_1 = \frac{6}{10} \quad \text{and} \quad \bar{\omega}_2 = \frac{3}{10} \quad . \quad (36)$$

On the other hand, the smoothness indicator should converge to one, making the linear and nonlinear weights equivalent. This can be achieved using the following definition [33]:

$$IS_k = \sum_{j=1}^2 \Delta x^{2j-1} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left( \frac{d^j \hat{f}^k}{dx^j} \right)^2 dx \quad , \quad (37)$$

which depends on the polynomial approximations employed. In the particular case of the WENO-JS scheme, these indicators of smoothness become

$$\begin{aligned} IS_0 &= \frac{13}{12} (f_{j-2} - 2f_{j-1} + f_j)^2 + \frac{1}{4} (f_{j-2} - 4f_{j-1} + 3f_j)^2 \quad , \\ IS_1 &= \frac{13}{12} (f_{j-1} - 2f_j + f_{j+1})^2 + \frac{1}{4} (-f_{j-1} + f_{j+1})^2 \quad \text{and} \quad (38) \\ IS_2 &= \frac{13}{12} (f_j - 2f_{j+1} + f_{j+2})^2 + \frac{1}{4} (3f_j - 4f_{j+1} + f_{j+2})^2 \quad . \end{aligned}$$

After the early seminal contributions [32,33], a diverse spectrum of WENO schemes has emerged. For instance, the WENO-Z scheme [35] introduces a new procedure for the calculation of the nonlinear weights, i.e.,

$$\omega_k^{(Z)} = \frac{\alpha_k^z}{\sum_{n=0}^{r-1} \alpha_n^z} \quad \text{with} \quad \alpha_k^z = \bar{\omega}_k \left[ 1 + \left( \frac{\tau}{\varepsilon + IS_k} \right)^p \right] \quad , \quad (39)$$

where  $\tau = |IS_0 - IS_2|$ . Another example is the WENO-ZP [36], which proposes a different formulation for the weights, namely

$$\alpha_k^{zp} = \bar{\omega}_k \left[ 1 + \left( \frac{\tau + \varepsilon}{\varepsilon + IS_k} \right)^p + \lambda \left( \frac{IS_{k'} + \varepsilon}{\tau + \varepsilon} \right) \right] \quad , \quad (40)$$

noting that it reduces to the WENO-Z scheme when  $\lambda = 0$ , although  $\lambda = \Delta x^{2/3}$  is suggested instead. For the sake of simplicity, the variants that modify the reconstruction polynomials are omitted. Nevertheless, their derivation can be found elsewhere [37,39,56].

### Extension to a System of Equations

In the case of a system of hyperbolic equations, i.e.,

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial x} = 0 \quad , \tag{41}$$

$q = q(x, t)$  is the vector of conservative variables and  $f = f(q(x, t))$  is the vector of conservative fluxes. It is possible rewrite this equation as

$$\frac{\partial q}{\partial t} + \frac{\partial f}{\partial q} \frac{\partial q}{\partial x} = 0 \quad , \tag{42}$$

using the chain rule. Solving Equation (42) is known as the component-wise approach. A different approach, known as the characteristic-wise approach, can be pursued by taking advantage of Equation (41)'s hyperbolicity. As a consequence,  $\partial f / \partial q$  has a complete set of real eigenvalues, and also, Equation (42) can be decoupled.

When using a high-order discretization, the characteristic-wise approach has provided more accurate results than its component-wise counterpart. It can be implemented by first defining  $R(q)$  and  $L(q) = R^{-1}$ , which are the right and left eigenvector matrixes of  $\partial f / \partial q$ , respectively. Hence, the Lax–Friedrichs flux split for systems of equations is written as

$$f_j^\pm = \frac{1}{2} \left( f_j \pm R_{i+\frac{1}{2}} \Lambda_{i+\frac{1}{2}} L_{i+\frac{1}{2}} q_j \right) \quad \text{with } j \in [i-l, i+l] \quad , \tag{43}$$

where  $l$  is the stencil width. Furthermore,  $R_{i+1/2} = R(q_{i+1/2})$  and  $L_{i+1/2} = L(q_{i+1/2})$ , where  $q_{i+1/2}$  is an average state between  $i$  and  $i + 1$ , such as the Roe average [12], and

$$\Lambda_{i+\frac{1}{2}} = \text{Diag} \left[ \lambda^{(0)}, \lambda^{(1)}, \dots, \lambda^{(m)} \right] \quad \text{with } \lambda^{(p)} = \max_q \left| \lambda^{(p)}(q_{i+\frac{1}{2}}) \right| \quad , \tag{44}$$

where  $m$  is the number of equations and  $\lambda^{(p)}$  is the  $p$ th eigenvalue of  $\partial f / \partial q$ . Multiplying Equation (43) by  $L_{i+\frac{1}{2}}$  as well as denoting  $\tilde{f}_j = L_{i+\frac{1}{2}} f_j$  and  $w_j = L_{i+\frac{1}{2}} q_j$ , yields

$$\tilde{f}_j^\pm = \frac{1}{2} \left( \tilde{f}_j \pm \Lambda_{i+\frac{1}{2}} w_j \right) \quad , \tag{45}$$

which is the projection of positive and negative fluxes into the characteristic space. A WENO reconstruction of  $\tilde{f}_j^\pm$  generates the numerical flux function approximation  $\hat{\tilde{f}}_{i+\frac{1}{2}}^\pm$ . Projecting back to the physical space with  $\hat{f}_{i+\frac{1}{2}}^\pm = R_{i+\frac{1}{2}} \hat{\tilde{f}}_{i+\frac{1}{2}}^\pm$ , it is possible to build the scheme using Equation (28). Multi-dimensional problems are built in the same way, one dimension at a time. The eigenvalues and eigenvectors can be found elsewhere [65].

### 2.3.3. Temporal Integration

The spatial discretization of both viscous and inviscid terms present in Equation (10) leads to the system of ordinary differential equations

$$\frac{dq}{dt} = \mathcal{F}(q) \quad , \tag{46}$$

where  $\mathcal{F}(\mathbf{q})$  represents discrete residue obtained after spatial discretization. Following the method of lines, the time integration of Equation (46) can be now discussed. The simplest way of marching it forward in time is

$$\mathbf{q}^{n+1} = \mathbf{q}^n + \Delta t \mathcal{F}(\mathbf{q}^n) + \mathbf{O}(\Delta t) \quad , \quad (47)$$

known as the explicit Euler method, which is first-order-accurate in time. Higher-order time integration schemes also exist. They are designed to increase scheme accuracy in time, reducing the CPU time required to generate highly accurate solutions. Runge–Kutta (RK) schemes are arguably the most used multi-stage high-order marching schemes. However, classical schemes [5] that do not satisfy the nonlinear proprieties required by a WENO discretization lead to spurious oscillations near discontinuities.

Strong-Stability-Preserving (SSP) RK schemes [47] ensure the preservation of these nonlinear stability proprieties. They can be written as

$$\begin{aligned} \mathbf{k}^{(0)} &= \mathbf{q}^n \quad , \\ \mathbf{k}^{(i)} &= \sum_{k=0}^{i-1} \left( \alpha_{i,k} \mathbf{k}^{(k)} + \beta_{i,k} \Delta t \mathcal{F}(\mathbf{k}^{(k)}) \right) \quad \text{with } 1 \leq i \leq s \quad , \\ \mathbf{q}^{n+1} &= \mathbf{k}^{(s)} \quad , \end{aligned} \quad (48)$$

where  $s$  is the number of stages and  $\mathbf{k}^{(i)}$  is the intermediate stage variable vector. It is important to note that SSPRK schemes are subject to a nonlinear time-step restriction that is more stringent than its linear counterpart. In other words, the aforementioned nonlinear stability properties will only be preserved when  $\Delta t \leq \mathcal{C} \Delta t_{EE_{max}}$ , for a given  $\mathcal{C}$  constant.

Both the optimal second-order SSPRK scheme, given by

$$\begin{aligned} \mathbf{k}^{(1)} &= \mathbf{q}^n + \Delta t \mathcal{F}(\mathbf{q}^n) \quad \text{and} \\ \mathbf{q}^{n+1} &= \frac{1}{2} \mathbf{q}^n + \frac{1}{2} \mathbf{k}^{(1)} + \frac{\Delta t}{2} \mathcal{F}(\mathbf{k}^{(1)}) \quad , \end{aligned} \quad (49)$$

and the optimal third-order SSPRK scheme, given by

$$\begin{aligned} \mathbf{k}^{(1)} &= \mathbf{q}^n + \Delta t \mathcal{F}(\mathbf{q}^n) \quad , \\ \mathbf{k}^{(2)} &= \frac{3}{4} \mathbf{q}^n + \frac{1}{4} \mathbf{k}^{(1)} + \frac{\Delta t}{4} \mathcal{F}(\mathbf{k}^{(1)}) \quad \text{and} \\ \mathbf{q}^{n+1} &= \frac{1}{3} \mathbf{q}^n + \frac{2}{3} \mathbf{k}^{(2)} + \frac{2 \Delta t}{3} \mathcal{F}(\mathbf{k}^{(2)}) \quad , \end{aligned} \quad (50)$$

are currently implemented in 3D4S [49].

### 3. Results

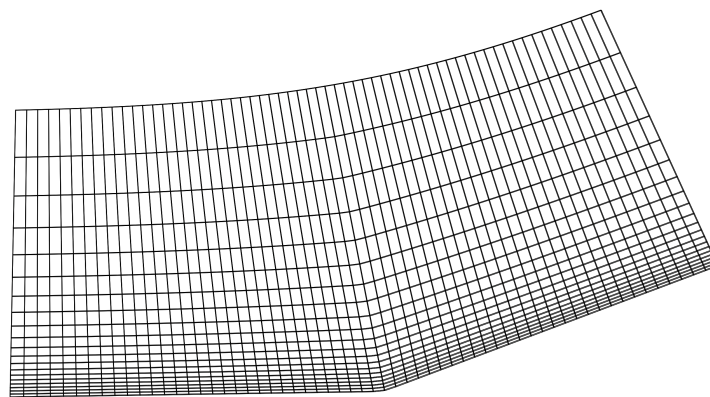
Three different test cases are presented to analyze the ability of high-order schemes to achieve accurate steady states when simulating the Navier–Stokes equations. First, a Mach 3 flow over a compression ramp is discussed. The importance of such problem comes from the preponderance of corners in high-speed flows. The adverse (favorable) pressure gradient upstream (downstream) of the corner separates (reattaches) the boundary layer. At these same regions of the flow, compression fans appear and lead to the formation of separation and reattachment shocks. Hence, the main goal here is to evaluate steady-state convergence in the presence of shock boundary layer interactions (SBLIs). Subsequently, a Mach 6 flow over a blunt body is discussed. The body geometry is constructed by connecting a cylinder to a wedge through a  $C^1$  curve. A bow shock appears near the leading edge and propagates away from the body. The relevance of this geometry arises from the

effects that blunted noses have in the laminar–turbulent boundary layer transition. Finally, a Mach 6 flow upstream of a cylinder is discussed. This case is employed here to highlight the difference between different WENO schemes regarding steady-state convergence. 3D4S code verification can be found elsewhere in the literature [66–68].

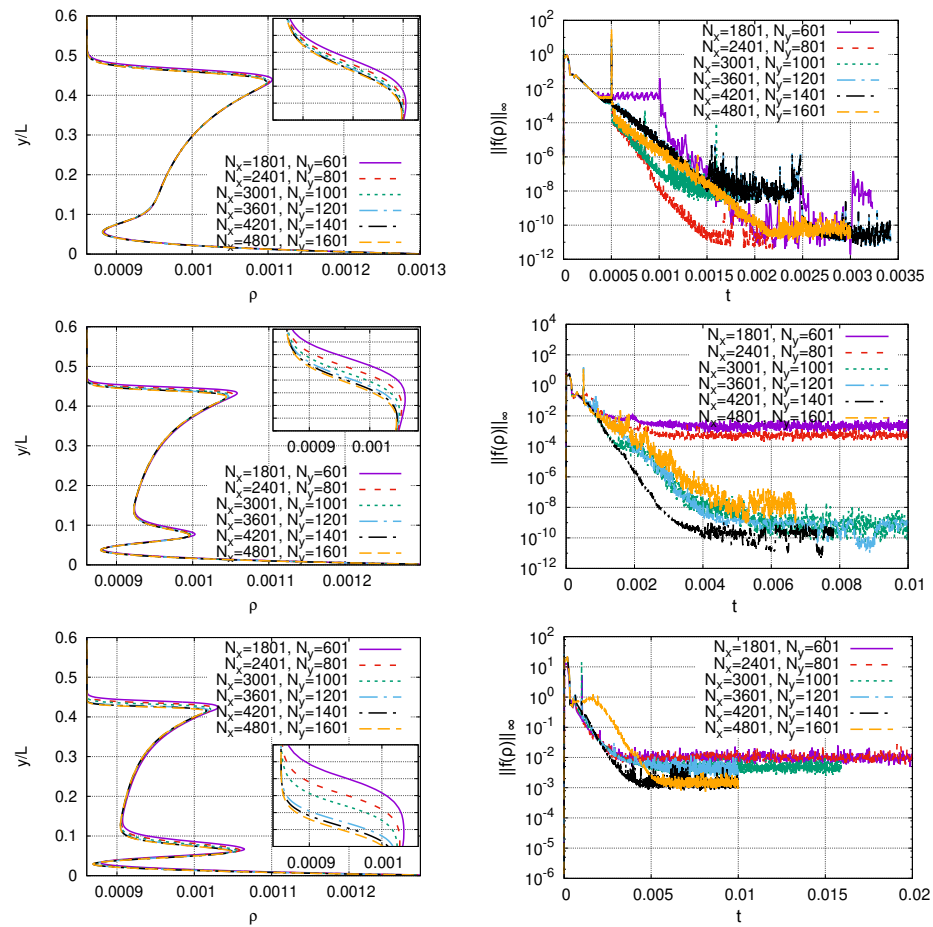
### 3.1. Compression Ramp

The first attempts to obtain such steady states used low-order schemes, which are here based on the explicit Euler scheme for time integration and a second-order accurate spatial discretization, which employed a TVD scheme with the Roe flux-difference splitting and the SuperBee flux limiter for inviscid fluxes as well as a conservative central-difference scheme for the viscous fluxes. No slip and isothermal walls are considered with  $T_w = T_\infty = 216.67$ . Figure 6 presents a sketch of the analytic grid used in these simulations, and a complete discussion of grid generation can be found elsewhere [6]. The density wall-normal profile at the corner (left) and density-maximum residue convergence in time (right) obtained for three different free stream Reynolds numbers, namely  $Re_\infty = 2 \times 10^3$  (top),  $5 \times 10^3$  (middle), and  $10 \times 10^3$  (bottom), are shown in Figure 7 for the isotherm wall under different grid sizes. Simulations require an increasingly larger number of grid points for their spatial profiles to converge towards the same tolerances as the free stream Reynolds number increases. Nevertheless, the number of grid points used in this figure shows spatial profiles that are grid-converged for graphical accuracy. On the other hand, residue convergence in time has a much stronger dependence on the free-stream Reynolds number. Residue converges in time for all grids employed when  $Re_\infty = 2 \times 10^3$ , for only the four largest grids employed when  $Re_\infty = 5 \times 10^3$ , and for none of the grids employed when  $Re_\infty = 10 \times 10^3$ . Convergence is quantified here by how much the maximum density residue decreases in time. It does so by approximately 10 orders of magnitude in the former two cases. In the latter case, however, it does so by only approximately four orders of magnitude, suggesting a six-digit accuracy loss. These low-order trends were also observed when using OpenFOAM, even though its second-order accurate finite-volume solver used approximately 16 million grid points to simulate the  $Re_\infty = 10^4$  case [66].

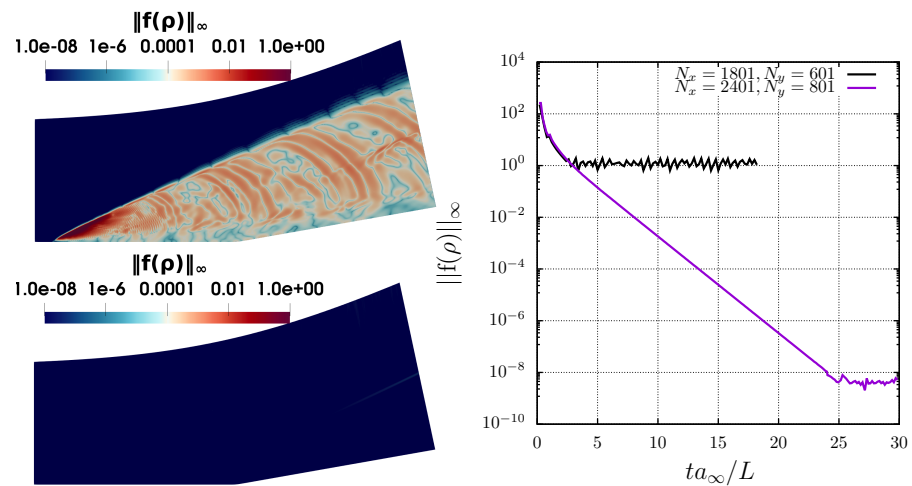
In order to emphasize the impact of higher-order schemes in such a steady-state simulation, Figure 8 (left) shows the convergence history for  $Re_\infty = 10 \times 10^3$  considering an adiabatic wall when a fifth-order WENO-JS and a fourth-order central difference scheme are applied to inviscid and viscous flux discretizations, respectively. Unlike the low-order simulations, a 10-orders-of-magnitude residue decrease in time is observed when using a  $N_x = 2401$ ,  $N_y = 801$  grid, while temporal content is still present in the residue when using a  $N_x = 1801$ ,  $N_y = 601$  grid. Figure 8 (right) shows the residue convergence in time for both grids. One can note in the latter case that numerical oscillations are generated at the leading-edge shock and propagate downstream, preventing residue convergence in the time-to-machine precision. Furthermore, wall-properties of case  $Re_\infty = 10 \times 10^3$  are presented in Appendix A.1.



**Figure 6.** Compression ramp grid using  $N_x = 61$  and  $N_y = 21$  points.



**Figure 7.** Density wall-normal profile at the corner (left) and density maximum residue convergence in time (right) obtained with  $Re_\infty = 2 \times 10^3$  (top),  $5 \times 10^3$  (middle), and  $10 \times 10^3$  (bottom) from low-order 3D4S for the isothermal wall case under different grid sizes.



**Figure 8.** Convergence history (right) for the compression ramp simulation using different grid sizes, as well as the steady-state residue of mass conservation equation using (left top)  $N_x = 1801, N_y = 601$  and (left bottom)  $N_x = 2401, N_y = 801$ .

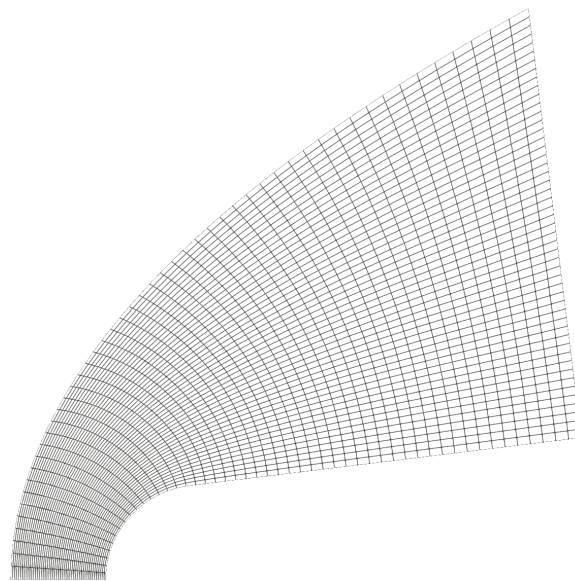
### 3.2. Blunt Wedge

WENO schemes developed for unsteady simulations are known to introduce post-shock oscillations, with a magnitude proportional to the local truncation error, that prevent residue convergence in time toward machine zero [38]. This issue was overcome with



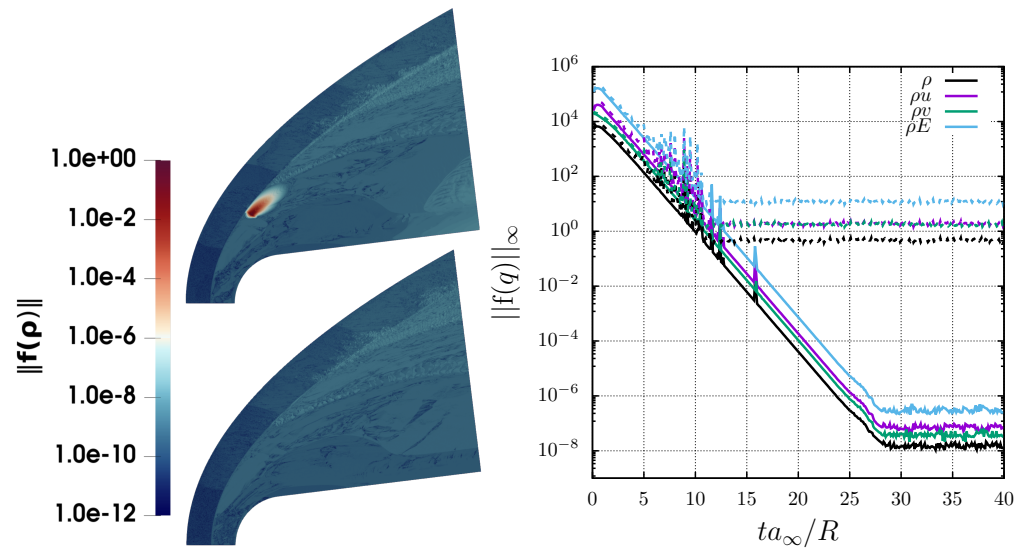
the development of WENO5-ZQ for steady simulations [37,39–41]. Its ability to do so is verified here by simulating the hypersonic flow over a blunt wedge.

Flow conditions are set by imposing the free stream Mach number  $Ma_\infty = 6$ , the Reynolds number based on the nose radius  $Re_R = 1.5 \times 10^3$ , the free stream Prandtl number  $Pr_\infty = 0.72$ , the ratio between constant specific heats  $\gamma = 1.4$ , the free stream temperature  $T_\infty = 200$  K, the dynamic viscosity defined by Sutherland's law, and the thermal conductivity defined by the constant Prandtl assumption. Furthermore, a no-slip and isothermal wall with  $T_w = T_0$  is considered. Grid-converged simulations were achieved with  $(N_x, N_y) = (1201, 801)$  using a time step of  $\Delta t = 5 \times 10^{-6}$ . Time integration was performed using the third-order accurate and three-stage ERK scheme, and the viscous fluxes were discretized with the fourth-order accurate conservative central-difference scheme. Figure 9 shows a sketch of the elliptic grid used in these simulations. A complete discussion of grid generation can be found elsewhere [63].



**Figure 9.** Blunt-body grid using  $N_x = 51$  and  $N_y = 51$  points.

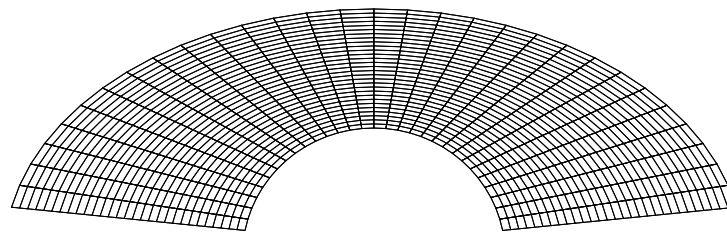
Figure 10 (right) shows the convergence in time of the  $L_\infty$  norm of the residue for the mass, momentum and energy conservation equations using WENO5-JS (dashed) and WENO5-ZQ (solid) schemes. As already expected, post-shock oscillations prevent the maximum residue calculated with the former from converging in time to machine zero. Maximum residues calculated with the latter, on the other hand, decrease by approximately 10 orders of magnitude in time. Figure 10 (left) shows the spatial distribution of the steady mass conservation equation residue obtained while using WENO-JS (top) and WENO-ZQ (bottom) schemes. One can note that the maximum residues are, in fact, located at the shock. Elsewhere, these residues are two to five orders of magnitude smaller. There was no need to either align the grid and the shock or cluster grid points around the shock whenever using WENO. This is in contrast with said literature, which simulated this flow using a second-order TVD-type scheme instead. They spent a large amount of effort trying to improve the alignment between the grid and steady shock, as well as clustering grid points around this shock due to the strong gradients in the stagnation region. Moreover, wall-properties of case  $Re_\infty = 1.5 \times 10^3$  are presented in Appendix A.2.



**Figure 10.** Instantaneous spatial distribution of the mass conservation residue obtained while using WENO-JS (left top) and WENO-ZQ (left bottom). Residue time convergence to the steady state (right) for the blunt-body simulation using WENO5-JS (dashed) and WENO5-ZQ (solid) schemes.

### 3.3. Quarter Cylinder

WENO-ZQ, which is especially designed for steady simulations, has shown better results than the classical WENO-JS scheme so far. This is also true for other WENO schemes as well; i.e., they also have a worse performance for steady simulations when compared to WENO-ZQ. In order to illustrate this, a set of simulation results is presented for a Mach 6 flow upstream of a cylinder using four different WENO schemes, namely WENO-JS [33], WENO-ZP [36], WENO-ZPs [36,56] and WENO-ZQ [37]. The reference Reynolds number is  $Re_D = 1 \times 10^4$  and the free-stream temperature  $T_\infty = 271.155$  K. Furthermore, a no-slip and adiabatic wall is considered. Figure 11 shows a sketch of the analytic grid used in these simulations, where the grid generation formulas can be found elsewhere [33].



**Figure 11.** Cylinder grid using  $N_x = 31$  and  $N_y = 31$  points.

Figure 12 shows the mass conservation equation’s maximum residue convergence in time obtained while employing WENO-JS (top left), WENO-ZP (top right), WENO-ZPs (bottom left), and WENO-ZQ (bottom right). For each case, a grid-independence study was performed. While the former three schemes were only able to reduce this residue in time by approximately 3 orders of magnitude, WENO-ZQ was able to do so for 12 orders of magnitude, essentially reaching machine precision. Figure 13 presents the spatial distribution of the steady density field obtained for these four cases. One can note that slight post-shock oscillations prevent the residue convergence to machine precision in the three leftmost plots, as occurred in the compression ramp and blunt-wedge cases. Furthermore, wall-properties of case  $Re_\infty = 10 \times 10^3$  are presented in Appendix A.3.

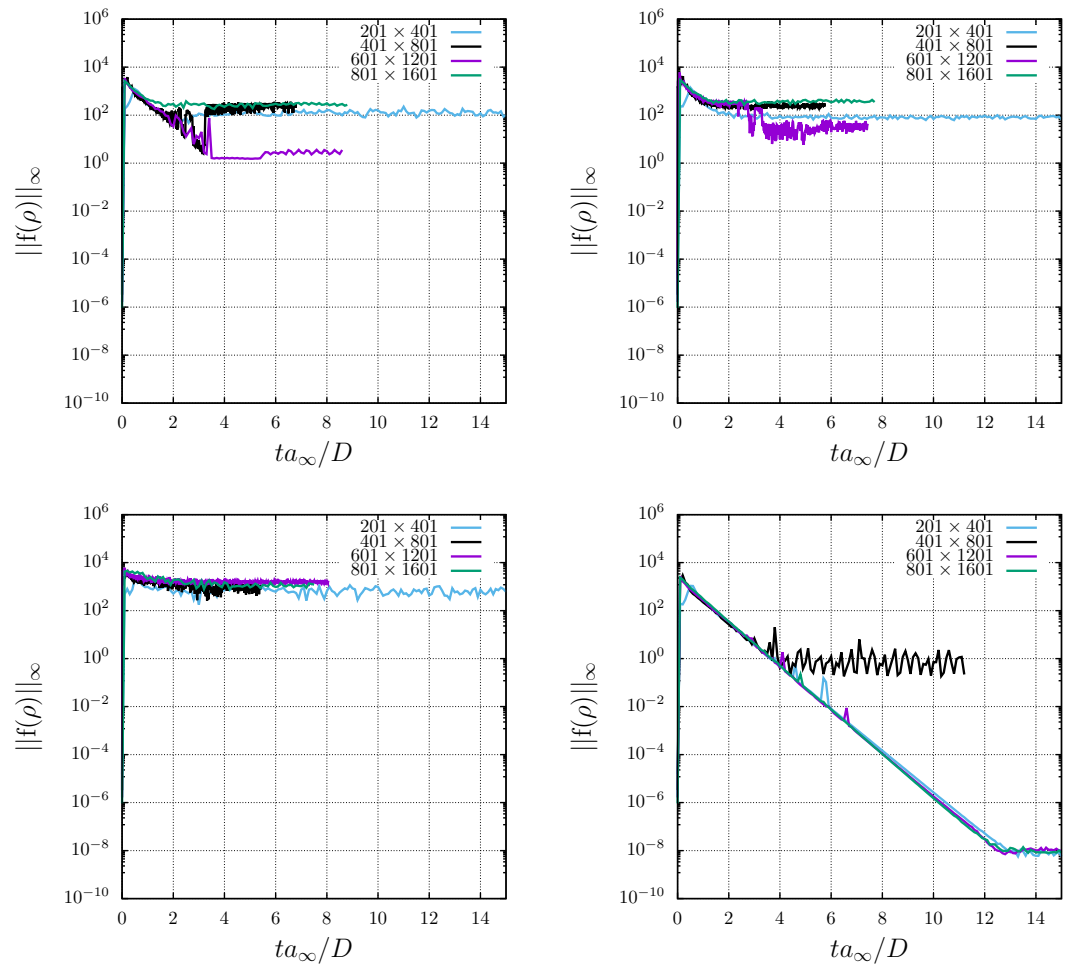


Figure 12. Density maximum residue convergence in time obtained with WENO-JS (top left), WENO-ZP (top right), WENO-ZPs (bottom left), and WENO-ZQ (bottom right) for different grid sizes.

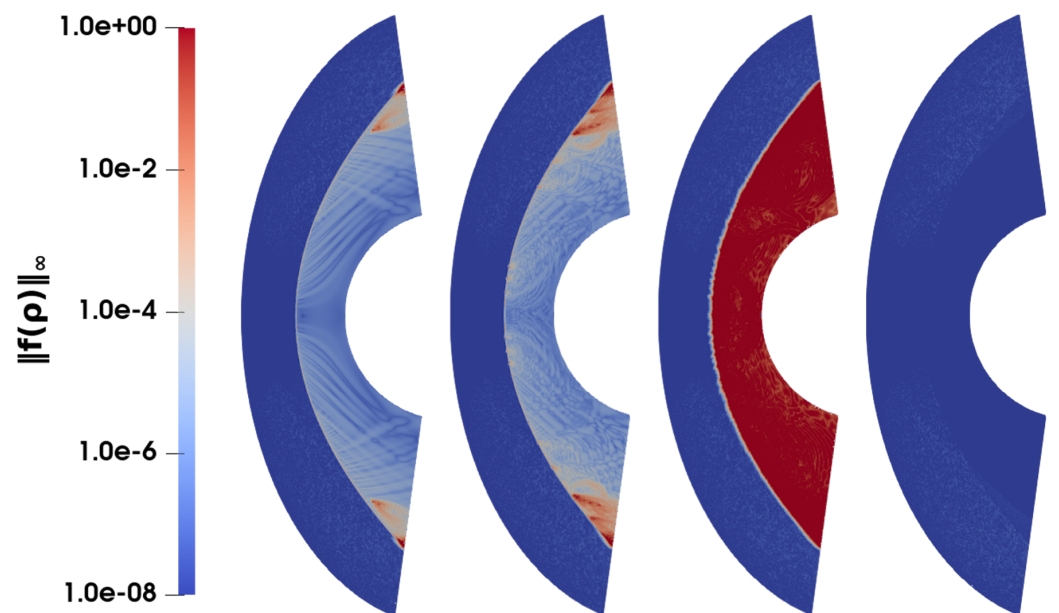


Figure 13. Steady-state density residue spatial distribution obtained using WENO-JS (first column), WENO-ZP (second column), WENO-ZPs (third column), and WENO-ZQ (fourth column). The results were obtained considering the  $(N_x, N_y) = (801, 1201)$  grid.

#### 4. Conclusions

The ability of high-order shock-capturing schemes to achieve residue convergence in time and, hence, generate accurate steady-states was discussed in the present work. A carefully analysis of three different supersonic flows showed the benefits of using higher-order schemes. They are more efficient when compared to their lower-order counterparts. As the Reynolds number increases, the high grid resolution required to achieve machine precision steady-states when using lower-order schemes makes them unfeasible. A considerably smaller grid resolution is required to do so when using high-order schemes. However, this is not true of all high-order schemes. Only the WENO-ZQ scheme, which is specially designed for steady-state simulations, was able to achieve machine precision residues in all test cases explored in this study. It is important to highlight that, even though a double-precision floating point was used, machine precision for the computation of residue  $\mathcal{F}$  was greater than  $O(10^{-16})$ . This is expected since the increment per time step  $\Delta t$  for the conservative variables is  $O(\Delta t \mathcal{F})$ , which leads to an increment of  $O(10^{-16})$ .

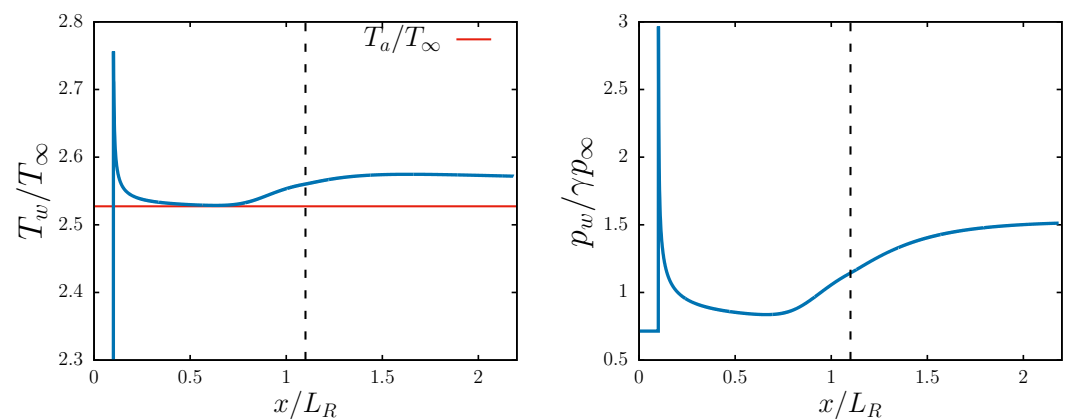
**Funding:** This research was funded by the US grants FA9550-18-1-0419, FA9550-22-1-0033 and FA8655-23-1-7031 under SOARD/AFOSR as well as the Brazilian agencies CAPES (graduate fellowships), FAPERJ (CNE and Bolsa Nota 10 SEI E-26/204.043/2022) and CNPq (PQ1D).

**Acknowledgments:** The authors would like to acknowledge Pedro Paredes (NIA/NASA LaRC) and Vassilis Theofilis (Technion) for their support during the development of this work.

**Conflicts of Interest:** The authors declare no conflicts of interest.

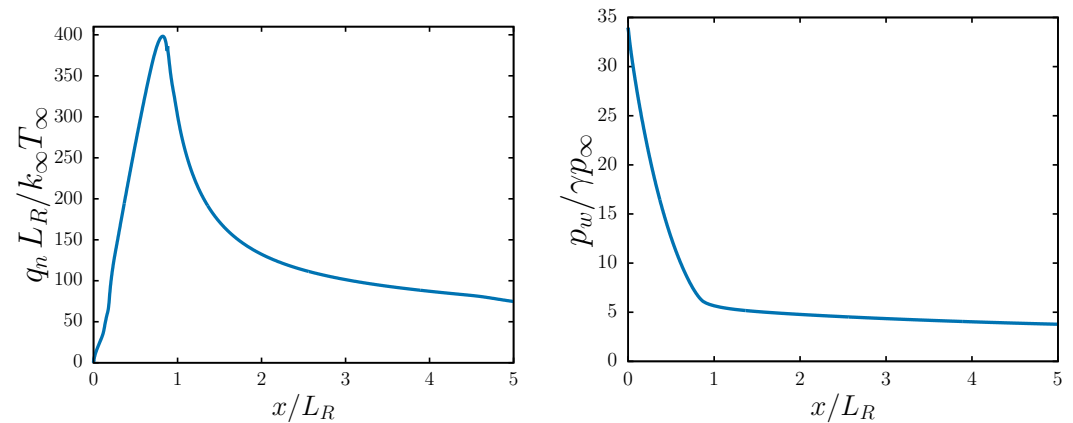
#### Appendix A. Wall Properties

##### Appendix A.1. Compression Ramp



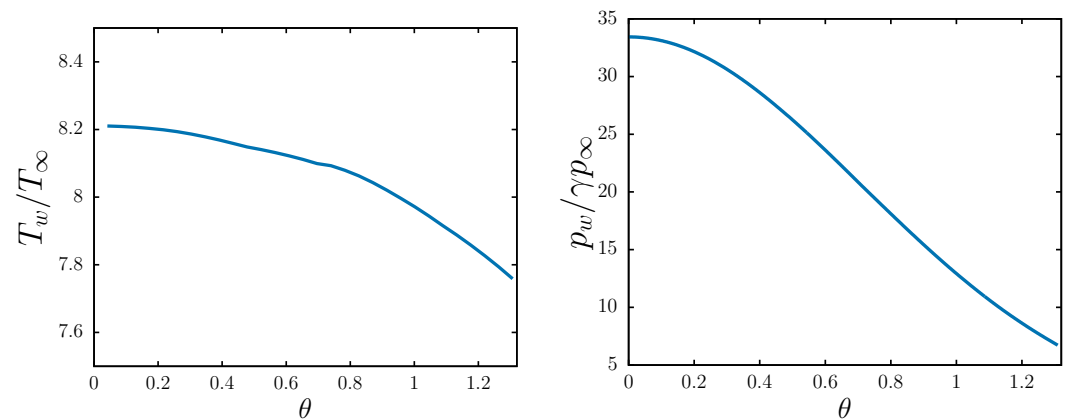
**Figure A1.** Dimensionless wall temperature (left) and pressure (right). In both figures, the vertical dashed black line marks the corner position. The solid red line represents the adiabatic wall temperature using the theoretical recovery temperature for laminar boundary layers. The leading edge is placed at  $x/L_R = 0.1$ , and the reference length is the distance from the leading edge to the corner.

## Appendix A.2. Blunt Body



**Figure A2.** Dimensionless wall heat flux (**left**) and pressure (**right**). The reference length  $L_R$  is the nose radius and  $x/L_R$  measures the distance along the wall.

## Appendix A.3. Cylinder



**Figure A3.** Dimensionless wall temperature (**left**) and pressure (**right**).  $\theta = 0$  corresponds to the stagnation point. Due to symmetry, only the upper half of the solution is presented.

## References

1. Versteeg, H.K.; Malalasekera, W. *An Introduction to Computational Fluid Dynamics. The Finite Volume Method*; Prentice Hall: London, UK, 1995.
2. Tannehill, J.C.; Anderson, D.A.; Pletcher, R.H. *Computational Fluid Mechanics and Heat Transfer*; Taylor & Francis: Philadelphia, PA, USA, 1997.
3. Lomax, H.; Pulliam, T.H.; Zingg, D.W. *Fundamentals of Computational Fluid Dynamics*; Scientific Computation, Springer & Verlag: Berlin/Heidelberg, Germany, 2001.
4. Hirsch, C. *Numerical Computation of Internal and External Flows; Fundamentals of Computational Fluid Dynamics*; Elsevier: Amsterdam, The Netherlands, 2007; Volume 1.
5. Butcher, J.C. *Numerical Methods for Ordinary Differential Equations*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2008.
6. Santos, R.D. A Study of Runge–Kutta Schemes with Strong Linear and Nonlinear Numerical Stability. Ph.D. Thesis, Fluminense Federal University, Niterói, Brazil, 2020.
7. Assis, J.C. Grid Shock Alignment, Boundary Conditions Axial Symmetry and 3D simulations of Hypersonic Flows over Blunt Bodies. Master's Thesis, Universidade Federal Fluminense, Niterói, Brazil, 2024.
8. Harbison, S.P.; Steele, G.L., Jr. *C—A Reference Manual*, 4th ed.; Tartan, Inc.: Monroeville, PA, USA, 1995.
9. Santiago, K.F.S.; Alves, L.S.B. Conservative and nondispersive schemes for diffusion terms with strong property variations. *Numer. Heat Transf. Part B Fundam.* **2017**, *71*, 133–145. [[CrossRef](#)]
10. Harten, A. High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.* **1983**, *49*, 357–393. [[CrossRef](#)]
11. Hirsch, C. *Numerical Computation of Internal and External Flows, Computational Methods for Inviscid and Viscous Flows*; Wiley: Hoboken, NJ, USA, 1991.
12. Laney, C.B. (Ed.) *Computational Gas Dynamics*; Cambridge University Press: Cambridge, UK, 1998.



13. Roe, P.L. Some contributions to the modelling of discontinuous flows. In *Large-Scale Computations in Fluid Mechanics*; Engquist, B.E., Osher, S., Somerville, R.C.J., Eds.; AMS-SIAM: Providence, RI, USA, 1985; Volume 22, pp. 163–193.
14. Roe, P.L. *Numerical Algorithms for the Linear Wave Equation*; Royal Aircraft Establishment: Farnborough, UK, 1981.
15. Roe, P.L.; Baines, M.J. Algorithms for advection and shock problems. In Proceedings of the 4th Conference on Numerical Methods in Fluid Mechanics, Paris, France, 7–9 October 1981.
16. Sweby, P.K.; Baines, M.J. On convergence of Roe’s scheme for the general non-linear scalar wave equation. *J. Comput. Phys.* **1984**, *56*, 135–148. [[CrossRef](#)]
17. van Leer, B. Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. *J. Comput. Phys.* **1977**, *23*, 276–299. [[CrossRef](#)]
18. van Leer, B. Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme. *J. Comput. Phys.* **1974**, *14*, 361–370. [[CrossRef](#)]
19. Koren, B. Chapter A Robust Upwind Discretization Method for Advection, Diffusion and Source Terms. In *Numerical Methods for Advection-Diffusion Problems*; Centrum voor Wiskunde en Informatica: Amsterdam, The Netherlands, 1993.
20. Sweby, P.K. High Resolution Schemes Using FLux Limiters for Hyperbolic Conservation Laws. *SIAM J. Numer. Anal.* **1984**, *21*, 995–1011. [[CrossRef](#)]
21. Chakravarthy, S.; Osher, S. High resolution applications of the Osher upwind scheme for the Euler equations. In Proceedings of the 6th Computational Fluid Dynamics Conference Danvers, Danvers, MA, USA, 13–15 July 1983.
22. Waterson, N.P.; Deconinck, H. *A Unified Approach to the Design and Applications of Bounded Higher-Order Convection Schemes*; Preprint/Von Karman Institute for Fluid Dynamics, VKI: Sint-Genesius-Rode, Belgium, 1995; pp. 182–207.
23. van Albada, G.D.; van Leer, B.; Roberts, W.W. A Comparative Study of Computational Methods in Cosmic Gas Dynamics. *Astron. Astrophys.* **1982**, *108*, 76–84.
24. Boris, J.P.; Book, D.L. Flux-Corrected Transport. I. SHASTA, A Fluid Transport Algorithm That Works. *J. Comput. Phys.* **1973**, *11*, 38–69. [[CrossRef](#)]
25. van Leer, B. Towards the ultimate conservative difference scheme I. The quest of monotonicity. In *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*; Lecture Notes in Physics; Springer: Berlin/Heidelberg, Germany, 1973; Volume Fundamental Numerical Techniques; pp. 163–168.
26. Goodman, J.B.; LeVeque, R.J. On the Accuracy of Stable Schemes for 2D Scalar Conservation Laws. *Math. Comput.* **1985**, *45*, 15–21. [[CrossRef](#)]
27. Harten, A.; Engquist, B.; Osher, S.; Chakravarthy, S.R. Uniformly High Order Accurate Essentially Non-Oscillatory Schemes, III. *J. Comput. Phys.* **1987**, *131*, 3–47. [[CrossRef](#)]
28. Godunov, S.K. A difference scheme for numerical solution of discontinuous solution of hydrodynamic equations. *Math. Sb.* **1959**, *47*, 271–306.
29. van Leer, B. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method. *J. Comput. Phys.* **1979**, *32*, 101–136. [[CrossRef](#)]
30. Shu, C.W.; Osher, S. Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes. *J. Comput. Phys.* **1988**, *77*, 439–471. [[CrossRef](#)]
31. Shu, C.W.; Osher, S. Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes II. *J. Comput. Phys.* **1989**, *83*, 32–78. [[CrossRef](#)]
32. Liu, X.D.; Osher, S.; Chan, T. Weighted Essentially Non-Oscillatory Schemes. *J. Comput. Phys.* **1994**, *115*, 200–212. [[CrossRef](#)]
33. Jiang, G.S.; Shu, C.W. Efficient Implementation of Weighted ENO Schemes. *J. Comput. Phys.* **1996**, *126*, 202–228. [[CrossRef](#)]
34. Henrick, A.K.; Aslam, T.D.; Powers, J.M. Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points. *J. Comput. Phys.* **2005**, *207*, 542–567. [[CrossRef](#)]
35. Borges, R.; Carmona, M.; Costa, B.; Don, W.S. An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. *J. Comput. Phys.* **2008**, *227*, 3191–3211. [[CrossRef](#)]
36. Acker, F.; Borges, R.; Costa, B. An improved WENO-Z scheme. *J. Comput. Phys.* **2016**, *313*, 726–753. [[CrossRef](#)]
37. Zhu, J.; Qiu, J. A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws. *J. Comput. Phys.* **2016**, *318*, 110–121. [[CrossRef](#)]
38. Zhu, J.; Shu, C.W. Numerical study on the convergence to steady state solutions of a new class of high order WENO schemes. *J. Comput. Phys.* **2017**, *349*, 80–96. [[CrossRef](#)]
39. Zhu, J.; Shu, C.W. A new type of multi-resolution WENO schemes with increasingly higher order of accuracy. *J. Comput. Phys.* **2018**, *375*, 659–683. [[CrossRef](#)]
40. Zhu, J.; Shu, C.W. Convergence to Steady-State Solutions of the New Type of High-Order Multi-resolution WENO Schemes: A Numerical Study. *Commun. Appl. Math. Comput.* **2019**, *2*, 429–460. [[CrossRef](#)]
41. Zhang, S.; Zhu, J.; Shu, C.W. A brief review on the convergence to steady state solutions of Euler equations with high-order WENO schemes. *Adv. Aerodyn.* **2019**, *1*, 25. [[CrossRef](#)]
42. Toro, E.F. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*; Springer-Verlag: Berlin/Heidelberg, Germany, 2009.
43. Roe, P.L. Approximate Riemann Solvers, Parameter Vectors and Difference Schemes. *J. Comput. Phys.* **1981**, *43*, 357–372. [[CrossRef](#)]

44. Harten, A.; Lax, P.D.; van Leer, B. On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws. *SIAM Rev.* **1983**, *25*, 35–61. [[CrossRef](#)]
45. Einfeldt, B. On Godunov-Type Methods for Gas Dynamics. *SIAM J. Numer. Anal.* **1988**, *25*, 294–318. [[CrossRef](#)]
46. Steger, J.L.; Warming, R.F. Flux vector splitting of the inviscid gas dynamic equations with application to finite-difference methods. *J. Comput. Phys.* **1981**, *40*, 263–293. [[CrossRef](#)]
47. Gottlieb, S.; Shu, C.W. Total-Variation-Diminishing Runge–Kutta Schemes. *Math. Comput.* **1998**, *67*, 73–85. [[CrossRef](#)]
48. Spiteri, R.J.; Ruuth, S.J. A new class of optimal high-order strong-stability- preserving time discretization methods. *SIAM J. Numer. Anal.* **2002**, *40*, 469–491. [[CrossRef](#)]
49. Gottlieb, S.; Ketcheson, D.; Shu, C.W. High-order strong stability preserving time discretizations. *J. Sci. Comput.* **2009**, *38*, 251–289. [[CrossRef](#)]
50. Ferracina, L.; Spijker, M.N. An Extension and Analysis of the Shu-Osher Representation of Runge–Kutta Methods. *Math. Comput.* **2004**, *74*, 201–219. [[CrossRef](#)]
51. Ferracina, L.; Spijker, M.N. *Computing Optimal Monotonicity-Preserving Runge–Kutta Methods*; Technical Report MI2005-07; Leiden University: Leiden, The Netherlands, 2005.
52. Ferracina, L.; Spijker, M.N. Strong Stability of Singly-Diagonally-Implicit Runge–Kutta Methods. *Appl. Numer. Math.* **2008**, *58*, 1675–1686. [[CrossRef](#)]
53. Ketcheson, D.I.; Macdonald, C.B.; Gottlieb, S. Optimal Implicit Strong Stability Preserving Runge–Kutta Methods. *Appl. Numer. Math.* **2009**, *59*, 373–392. [[CrossRef](#)]
54. Pareschi, L.; Russo, G. implicit–explicit Runge–Kutta Schemes and Applications to Hyperbolic Systems with Relaxation. *J. Sci. Comput.* **2005**, *25*, 129–155.
55. Santos, R.D.; Alves, L.S. A comparative analysis of explicit, IMEX and implicit strong stability preserving Runge–Kutta schemes. *Appl. Numer. Math.* **2021**, *159*, 204–220. [[CrossRef](#)]
56. Zhang, S.; Shu, C.W. A new smoothness indicator for the WENO schemes and its effect on the convergence to steady state solutions. *J. Sci. Comput.* **2007**, *31*, 273–305. [[CrossRef](#)]
57. Liseikin, V. *Grid Generation Methods*; Scientific Computation; Springer: Amsterdam, The Netherlands, 2009.
58. Thompson, J.F. Elliptic grid generation. *Appl. Math. Comput.* **1982**, *10–11*, 79–105. [[CrossRef](#)]
59. Thompson, J.F.; Soni, B.K.; Weatherill, N.P. *Handbook of Grid Generation*; CRC Press: Boca Raton, FL, USA, 1998.
60. Steger, J.L.; Sorenson, R.L. Automatic mesh-point clustering near a boundary in grid generation with elliptic partial differential equations. *J. Comput. Phys.* **1979**, *33*, 405–410. [[CrossRef](#)]
61. Hsu, K.; Lee, S.L. A numerical technique for two-dimensional grid generation with grid control at all of the boundaries. *J. Comput. Phys.* **1991**, *96*, 451–469. [[CrossRef](#)]
62. Spekrijse, S.P. Elliptic Grid Generation Based on Laplace Equations and Algebraic Transformations. *J. Comput. Phys.* **1995**, *118*, 38–61. [[CrossRef](#)]
63. Nunes, M.S.S. Generation of Steady-States for Stationary and Convectively Unstable Flows Using the Frequency Displacement Procedure. Master’s Thesis, Fluminense Federal University, Niterói, Brazil, 2021.
64. Poinot, T.J.; Lele, S.K. Boundary conditions for direct simulations of compressible viscous flows. *J. Comput. Phys.* **1992**, *101*, 104–129. [[CrossRef](#)]
65. Masatsuka, K. *I Do Like CFD*; Lulu. com: Morrisville, NC, USA, 2009; Volume 1.
66. Santos, R.D.; Alves, L.S.; Cerulus, N.; Theofilis, V. On Two-Dimensional Steady-States of Supersonic Flows over Compression Ramps and their Global Linear Instability. In Proceedings of the AIAA Scitech Forum, San Diego, CA, USA, 7–11 January 2019; pp. 2019–2321.
67. Burtsev, A.; Quintanilha, H.R.D.; Theofilis, V.; Santos, R.D.; Alves, L.S. Linear Instability Mechanisms of Supersonic Flow Past Blunt Bodies. In Proceedings of the AIAA Scitech Forum, Virtual, 11–15 and 19–21 January 2021; AIAA 2021-0050.
68. Santos, R.D.; Alves, L.S.; Cerulus, N.; Quintanilha, H.; Theofilis, V. Accurate Two Dimensional Steady-State for the Supersonic Flow over a Compression Corner. In Proceedings of the 33th ICAS Congress, Stockholm, Sweden, 4–9 September 2022.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.