

Numerical simulation of a highly underexpanded carbon dioxide jet

Zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften

von der Fakultät für Mathematik des
Karlsruher Instituts für Technologie (KIT)

genehmigte Dissertation

von
Carlos Galeano

Hauptreferent: Prof. Dr. Willy Dörfler
Korreferent: Prof. Dr. Christian Wieners

Tag der mündlichen Prüfung: 15.02.2017

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

Karlsruhe, 15.02.2017

.....
(Carlos Galeano)

Contents

1. Underexpanded jets	1
1.1. Overview of underexpanded jets	1
1.2. Rapid Expansion of Supercritical Solutions (<i>RESS</i>)	1
1.3. Scope of this work	4
2. Conservation Laws	7
2.1. The Navier–Stokes equations	7
2.1.1. The transport theorem	7
2.1.2. The continuity equation (mass conservation)	8
2.1.3. The momentum equations (momentum conservation)	8
2.1.4. The energy equation (energy conservation)	10
2.1.5. The set of Navier–Stokes equations for fluid dynamics	11
2.1.6. Equation of state	11
2.1.7. Dimensionless form of Navier–Stokes equations	12
2.1.8. Navier–Stokes equations in cylindrical coordinates	13
2.2. Inviscid case: Euler equations	15
2.2.1. Quasilinear form of Euler equations	15
2.2.2. Homogeneity property of the inviscid flux vector	16
2.2.3. Hyperbolicity of Euler equations	17
2.2.4. Rotational invariance of Euler equations	20
2.2.5. Riemann problem for the Euler equations	21
2.3. Boundary conditions	22
2.4. Equations of state for carbon dioxide	24
2.4.1. Span and Wagner equation of state	24
2.4.2. IUPAC equation of state	24
2.4.3. Extended and generalized Bender equation of state	25
2.4.4. Peng–Robinson equation of state	25
2.5. Thermophysical properties of CO ₂	25
2.5.1. Physical constants for CO ₂	26
2.5.2. Isochoric specific heat	26
2.5.3. Isobaric specific heat	28
2.5.4. Heat capacity ratio	29
2.5.5. Speed of sound	29
2.5.6. Specific internal energy	30
2.5.7. Dynamic viscosity	31
2.5.8. Thermal conductivity	32
3. Spatial semidiscretization for Navier–Stokes equations	35
3.1. Meshing	35
3.2. Weak formulation of inviscid problems using discontinuous Galerkin finite element method (DGFEM)	36
3.2.1. Broken Sobolev spaces	36

3.2.2.	Weak formulation	36
3.2.3.	Discrete approximation	37
3.3.	Numerical flux	37
3.3.1.	Central schemes	38
3.3.1.1.	Lax–Friedrichs numerical flux	38
3.3.1.2.	Lax–Wendroff numerical flux	38
3.3.2.	Upwind schemes	39
3.3.2.1.	Flux difference splitting schemes	40
3.3.2.2.	Flux vector splitting schemes	42
3.3.3.	The <i>AUSM⁺-up</i> scheme	45
3.3.4.	Hybrid-flux methods	47
3.4.	Shock capturing and Gibbs phenomenon	48
3.4.1.	Discontinuity indicator	48
3.4.2.	Artificial viscosity method	49
3.5.	Adaptive grid strategy	50
3.6.	Inviscid boundary conditions	52
3.6.1.	Inlet-outlet boundary condition	52
3.6.1.1.	Subsonic inlet/outlet boundary conditions	53
3.6.1.2.	Supersonic inlet/outlet boundary conditions	53
3.6.2.	Impermeable wall boundary condition	54
3.7.	Numerical treatment of viscous terms	54
3.7.1.	Scalar convection-diffusion problem	54
3.7.2.	Interior penalty method applied to Navier–Stokes problem	56
3.7.3.	Implementation of the viscous flux	57
3.7.4.	Viscous boundary conditions	58
3.7.4.1.	Subsonic inlet boundary condition	58
3.7.4.2.	Subsonic outlet boundary condition	59
3.7.4.3.	Supersonic inlet boundary condition	59
3.7.4.4.	Supersonic outlet boundary condition	60
3.7.4.5.	No-slip wall boundary condition	60
4.	Time integration	61
4.1.	Runge–Kutta methods	62
4.2.	Strong stability preserving Runge–Kutta methods	63
4.3.	Exponential integrators	63
4.3.1.	Exponential Runge-Kutta methods	63
4.3.1.1.	Exponential Euler method	66
4.3.1.2.	Convergence of higher order exponential Runge–Kutta methods	66
4.3.1.3.	Second order exponential Runge-Kutta method	67
4.3.2.	Rosenbrock-type methods	68
4.3.2.1.	Convergence order of the exponential Rosenbrock methods	68
4.3.2.2.	Exponential Rosenbrock–Euler method	69
4.3.2.3.	Higher order Rosenbrock-type methods	69
4.4.	Approximation of the matrix exponential operators	71
4.4.1.	Approximation of matrix exponential operators using Krylov subspaces	71
4.4.2.	Computation of the exponential of \mathbf{H}	73
4.4.2.1.	Rational Chebyshev approximations	73
4.4.2.2.	Diagonal Padé approximations	74
4.4.3.	Computation of $\phi_k(\mathbf{H})$ -functions	75
4.4.4.	Reorthogonalization process	76
4.4.5.	Error control and size of the Krylov subspace	77

5. Compressible Navier-Stokes solver	81
5.1. Characteristics of the compressible Navier-Stokes solver	81
5.2. Algorithm description	82
5.3. Numerical tests	84
5.3.1. Two dimensional rotating Gaussian pulse	84
5.3.2. Sod's shock tube problem	91
5.3.3. GAMM channel problem	96
5.3.3.1. Subsonic case	96
5.3.3.2. Transonic case	101
5.3.3.3. Supersonic case	105
5.3.4. Supersonic forward-facing step problem	109
5.3.5. Axisymmetric free underexpanded air-jet	116
5.4. Highly underexpanded CO ₂ -jet	122
6. Conclusions	137
6.1. Summary	137
6.2. Contributions	138
6.3. Future research	138
6.3.1. Mathematical model	139
6.3.2. Numerical implementation	139
Bibliography	141
Appendix	159
A. Specific internal energy equations	159
B. Analytic expressions derived from egB-EOS	160
C. Two dimensional rotating Gaussian pulse	160
D. Solution using an uniform grid with \mathbb{P}^1 <i>DG</i> -elements	163
E. Axisymmetric free underexpanded air-jet	165
F. Highly underexpanded CO ₂ -jet	169

1. Underexpanded jets

1.1. Overview of underexpanded jets

The jet formed by the exhausting of a fluid through a nozzle can be classified as an *underexpanded jet* if the pressure at the nozzle exit (p_e) is higher than the ambient pressure (p_∞) where the jet develops. Otherwise the jet is considered *overexpanded*.

Indeed, the exhausting of a high-pressure fluid flow into a quiescent and low-pressure medium produces a penetration and rapid expansion of the flow into the low-pressure chamber. This adiabatic expansion is characterized by supersonic (or even hypersonic) speeds in the core of the jet, an fast increase in the cross-section area of it and a dramatic dropping in the values of pressure and temperature, reaching values even much below the ambient conditions.

The expansion fan, generated at the nozzle lip, originates and complex interaction of expansion and compression waves, which finally produce a well known shock-structure in the jet. In Sec. 5.3.5 a description about the formation mechanism and the configuration of this shock-pattern can be found. The behavior of this shock-structure can change in depends of the pressure ratio $NPR = p_e/p_\infty$. According with the NPR parameter, [FPGB15] classified this kind of jets as moderately, highly and very highly underexpanded.

In case of highly and very highly underexpanded jets the most important element of the shock-structure is the named *Mach disk*. This is a normal shock, slightly curved, located at the end of the adiabatic expansion and through which a recompression process of the fluid takes place. The size and position of this discontinuity has been extensively studied and empirical expressions for different ranges of NPR ratio can be found [FPGB15].

The *underexpanded jets* are in a wide variety of applications such as in the rocket propulsion [SB10], in mass spectrometry [JGT⁺04], in the combustion processes (e.g. in direct-injection of hydrogen [BVM13]), in the drilling and rock breaking [LLK⁺13], or in the production of small particles by the process called rapid expansion of supercritical solutions (*RESS*) [THH⁺02, HTS00].

1.2. Rapid Expansion of Supercritical Solutions (*RESS*)

The *RESS* is a process, mainly used for the micronization (or microencapsulation) of solids, characterized by the production of very uniform size-particles less than 1 μ m [HTS00]. In

this process a substance in supercritical state is used to dissolve a solid solute, taking advantage of the high dissolving power of the solvent at this state. At this high pressure-temperature conditions, the solution is then expanded, through a micronized channel, into a chamber at atmospheric conditions. The dropping in the pressure and temperature, result of the adiabatic expansion, greatly reduces the solvent power, generating a strong supersaturation in the solution and forming of a large number of nuclei. Later, as product of the coagulation, the nuclei grow and the solute is precipitated as fine particles from the solution, now at gas state. [HHH⁺01].

The *RESS* process is a relatively new technology used for example in the production of high purity and ultrafine powders for the pharmaceutical industry [GGKS16, THH⁺02], or for obtaining precursors materials as polymer microspheres or ceramic powders [YF15].

Because of its multiple advantages, the CO₂ is one of the most common solvents used in the *RESS* process. Indeed, the carbon dioxide is nontoxic, nonflammable, chemically inert, relatively inexpensive, environmentally friendly, and principally it has a relatively low critical point [LADQ14, GGKS16].

The modeling of this physical-chemical process is not a simple task. On one side, the high ratio between the supercritical pressure and the atmospheric conditions at expansion chamber configures, at the nozzle exit, a highly (or very highly) underexpanded jet. Thus, a complex shock-structure in the near field of the jet is formed, and hence strong variations in the thermophysical properties of the solvent in this region. Here, the precise prediction of the fields of velocity, pressure, temperature and density, not only in the supersonic-core of the jet but also in the rest of the expansion chamber, and even into the nozzle, is crucial for the correct numerical modeling of the process. In each of these regions occur important particle phenomena.

In fact, although the nucleation can start near to the nozzle exit, as soon as the pressure cross down the critical value, several works report that this process starts early in the pre-expansion located within the throttle device [HTS03, HHH⁺01, YF15]. Furthermore, the particle growth, which mainly takes place in the supersonic region of the jet, continues behind the Mach disk along the subsonic regions of the expansion chamber [HHH⁺01].

On the other hand, the modeling of the thermophysical properties of the fluid is another important, but not simple issue here. At supercritical conditions, small changes in the pressure values produce strong variations of the thermophysical properties of the solvent, specially at conditions near to the critical point. Thus, the use of a precise thermophysical model for the solvent is mandatory not only in the context of the hydrodynamic model but also for the particle formation prediction. Indeed, this thermophysical relation, called equation of state, in addition to complete the hydrodynamic model also allows to compute caloric properties such as the speed of sound, the specific heats and the solubility, all them required for modeling this type of processes.

There exist different equations of state to model the thermodynamic behavior of the CO₂. Although, the Span & Wagner *EOS* [SW96] is considered the most precise equation available for the carbon dioxide [LADQ14, WWFF13], its complexity does not allow to use it efficiently in the context of computational fluid dynamic. Additionally, while the use of this equation is restricted to conditions above of the triple point, multiple experiments have shown that, during the adiabatic expansion occurred in the *RESS* process, the jet can find states below the triple point [WWFF13]. On the other hand, there is a type of mathematically simpler equations, called cubic *EOS*, such as the Peng–Robinson equation

[PR76]. Nevertheless, the cubic-type *EOS* does not allow to predict accurately caloric properties, specially near to the critical point [Tür99, Tür00], and then these relations are not advisable in the context of *RESS* process. A third type of *EOS* is the named extended generalized Bender equation (*egB-EOS*) [Ben71]. This model, although is considerably more complex than the cubic-type equations, has a wider range of validity than the previous *EOS* and is strongly recommended for modeling of the *RESS* process [Tür99, HTS00]. Nevertheless, here a good computational implementation is required in order to get an efficient numerical model.

Although the underexpanded jets have been extensively studied both experimental and numerically (see e.g. [CSG66, OKM⁺08, FPG15]), there exist at present limited information about experimental results and accurate numerical models about the *RESS* process, given that the called supercritical fluid technologies were only introduced since the end of 20th century [GGKS16]. In this topic, the several works published by Türk et al. should be highlighted. In these publications the experimental results are compared with numerical approximations obtained from 1D models are used to predict the hydrodynamic behavior of the jet, within the capillary nozzle [Tür99, HTS00, HHH⁺01] and in the expansion chamber [HTS03]. In these works, the CO₂ is mainly used as solvent and substances as naphthelene, benzoic acid and cholesterol are used as solute.

More recently, some 2D models for the *RESS* process have been developed, although with relative success. In [YF15] a hydrodynamic model for CO₂ based on finite volumes along with a preconditioned flux splitting scheme is presented. For this work the thermo-physical properties for the solvent are computed using the *PROPATH* library [PRO08], which employs the *EOS* defined by the International Union of Pure and Applied Chemistry (*IUPAC*). Nevertheless, the pressure ratio *NPR* used for the numerical experiments is relatively low ($p_o = 20.5\text{MPa}$ and $p_\infty = 2.5\text{MPa}$ are used) and it does not correspond to real operational conditions of the *RESS* process. On the other hand, the solution presented exhibits some spurious oscillations originated at Mach disk and propagated downstream. Additionally, the *EOS* employed can be only used above of the triple point. For conditions below this point a linearization for the *EOS* was incorporated.

In [GGKS16] experimental results are complemented with a numerical model built on the commercial software ANSYS-Fluent®, which uses finite volumes for the spatial discretization. For the experiments reported in this work, real operational condition for the *RESS* process were assumed. Thus, whereas a high-pressure condition (p_o) was taken equal to 25.0MPa, atmospheric pressure ($p_\infty = 0.1\text{MPa}$) was considered inside of the expansion chamber. Here, the ideal gas assumption was incorporated into the hydrodynamic model. The numerical results reported for the velocity field show an excessively diffusive solution with a very poor resolution in the shocks captured.

In [LDQA15] the simulation of the rapid expansion of supercritical carbon dioxide was developed by using of \mathbb{P}^1 continuous finite elements along with the characteristic-based split (*CBS*) method. Here, the time integration was developed using the first-order forward Euler scheme. The system of Navier–Stokes equation was completed using the extended generalized Bender *EOS* (*egB-EOS*), which, as previously, is strongly recommended for modeling CO₂ in this kind of process. Nevertheless, the numerical results reported were computed employing a pressure at nozzle inlet $p_o = 7.5\text{MPa}$ and atmospheric pressure in the expansion chamber. These values configure a low and not a realistic value for the *NPR* ratio into the *RESS* process.

Finally, in [WWFF13] the set on Navier–Stokes equations, complemented by a modified

Peng–Robinson *EOS*, is solved using finite volumes with a second-order Godunov method for the time marching scheme.

1.3. Scope of this work

The objective of this work is to develop an accurate, efficient and stable numeric strategy to simulate the hydrodynamic component of the *RESS* process using real high nozzle pressure ratios. The numerical model should allow the precise capture of the shock-structure in the underexpanded jet formed, a accurate prediction of the thermophysical properties of the fluid, as well as provided a efficient time marching scheme. The process parameters used in the present work correspond to the case analyzed in [HTS03], which considers the rapid expansion of supercritical CO₂ from a micronized nozzle, with inlet pressure $p_o = 20\text{MPa}$, to an expansion chamber at atmospheric pressure ($p_\infty = 0.1\text{MPa}$).

Henceforth, this document is organized in four chapters. In Chapter 2 the conservation equations for the modeling of the hydrodynamic component of *RESS* process are formulated. These balance equations, in form of the Navier–Stokes equations, are simplified for the case of problems with radial symmetric conditions, as in the free expanded jets. Because of the high-pressure condition at supercritical state, then the fluid cannot be assumed an ideal gas, and hence an equation of state (*EOS*) for real gas should be included in the numerical model. Considering its relatively low complexity, as well as the wide range of states in which the *EOS* is valid, including supercritical state, liquid, gas and liquid-gas mixtures conditions, the extended generalized Bender *EOS* (*egB-EOS*) was implemented in this work. Additionally, in this chapter the mathematical effects on the inviscid terms of Navier–Stokes equations by the use of real gas equations are analyzed.

In Chapter 3 the formulation used for the spatial semidiscretization, via discontinuous Galerkin elements, is discussed. Here, a complete analysis about numerical fluxes suitable for using with real gas equations is presented. Besides, in order to find stable and high-resolution approximation in the regions with non-smooth behavior, e.g. near to the Mach disk, the shock capturing and the adaptive mesh strategies are introduced. At the end of this chapter the implementation of the numerical boundary conditions is described.

The rapid expansion of fluid produced by a very high pressure gradient, as in the case of *RESS* process, multiple characteristic times are expected. On one hand, high speed regions in the core of the jet, that corresponds to supersonic or even hypersonic regimens, and on the other, regions with very low Mach numbers in a vast volume of the expansion chamber. Thus, the maximum allowable time step with explicit time-schemes will be strongly restricted by the fastest characteristics, whereas the physical time scale will be defined by slower characteristics. Here, even stronger restriction in the time step will arise if a local mesh refinement (adaptive refinement) is introduced. It configures a stiff problem and hence the implementation of a stable and efficient time marching scheme is necessary. In Chapter 4 an introduction to the exponential integrators used is included. Here, both exponential Runge–Kutta and exponential Rosenbrock methods are discussed. In these kind of methods the computation of the exponential of large-scale matrices is required, then a method based on projections onto Krylov subspaces using Arnoldi’s algorithm is described. Additional implementation details, as the reorthogonalization scheme, the stopping criteria for Arnoldi’s algorithm, and the computation of the matrix exponential via rational Chebyshev polynomials, are presented at the end of this chapter.

The numeric strategies described in previous chapter were implemented in a computational code, written in C++ language, using several libraries such as `deal.ii` [BHK07],

Trilinos [HBH⁺05] and **Intel**[®] **TBB** [Int16]. In Chapter 5 a general description of the compressible flow solver implemented is introduced. Besides, the results achieved in the test problems simulated with the compressible flow solver implemented are presented. Here, different benchmark problems were employed and different characteristics were analyzed. Initially, a linear diffusion-advection 2D-problem is solved and a convergence analysis for space and time is reported. Information about the computational performance of the code is also included. After the classic Sod's shock tube problem is used to evaluate the convergence rates in space and time for nonlinear problems. Later, the *GAMM*-channel problem and the forward facing step problem are used to evaluate the resolution in which the shocks are captured. The last test problem included consists in an underexpanded air-jet with a moderate pressure ratio. Finally the results obtained in the simulation of a very highly underexpanded CO₂-jet are shown. Here, multiple plots for Mach number, temperature, density and pressure are reported. Additionally, an analysis of the changes of phase occurred in the jet is presented using phase diagrams.

At the end of the document some conclusion and future works are listed.

2. Conservation Laws

2.1. The Navier–Stokes equations

2.1.1. The transport theorem

Consider a fluid that occupies a domain $\Omega_t \subset \mathbb{R}^d$ with $d \in \{1, 2, 3\}$ at time t , and let $\mathcal{V}(t) \subset \Omega_t$ be a bounded fluid volume called *control volume*. Thus, if $t_0 \in (t_1, t_2)$ is a fixed time instant and $\mathcal{V}(t_0) \subset \Omega_{t_0}$, the volume $\mathcal{V}(t)$ is defined by

$$\mathcal{V}(t) = \{\varphi(\mathbf{X}, t) : \mathbf{X} \in \mathcal{V}(t_0), t \in (t_1, t_2)\}. \quad (2.1)$$

Here, $\mathbf{X} \in \mathcal{V}(t_0) \rightarrow \mathbf{x} = \varphi(\mathbf{X}, t) \in \mathcal{V}(t)$ will be consider a continuously differentiable one-to-one mapping of $\mathcal{V}(t_0)$ onto $\mathcal{V}(t)$ with a continuous and bounded Jacobian $\det \nabla \varphi(\cdot, t)$ that satisfies the condition

$$\det \nabla \varphi(\mathbf{X}, t) > 0, \text{ for all } \mathbf{X} \in \mathcal{V}(t_0), t \in (t_1, t_2).$$

Let $F : \mathcal{M} \rightarrow \mathbb{R}$ be the Eulerian function that represents a physical quantity transported by the fluid, with $\mathbf{x} = \varphi(\mathbf{X}, t)$ and $\mathcal{M} = \{(\mathbf{x}, t) : t \in (0, T), \mathbf{x} \in \Omega_t\} \subset \mathbb{R}^{d+1}$ being the set that defines the domain of the function F . Thus, the total amount of F contained in the volume $\mathcal{V}(t)$ at time t is defined by

$$\mathcal{F}(t) = \int_{\mathcal{V}(t)} F(\mathbf{x}, t) d\mathbf{x},$$

and its variation respect to the time can be expressed by

$$\frac{d\mathcal{F}}{dt}(t) = \frac{d}{dt} \int_{\mathcal{V}(t)} F(\mathbf{x}, t) d\mathbf{x}.$$

Definition 2.1.1. If $\mathbb{A}(\mathbf{x}) \in \mathbb{R}^{m,n}$, $m, n \in \mathbb{N}$, is a tensor function on $\mathbf{x} \in \Omega_t$, then its *divergence operator* is defined by

$$\operatorname{div} \mathbb{A} = \left(\sum_{j=1}^n \frac{\partial A_{1j}}{\partial x_j}, \dots, \sum_{j=1}^n \frac{\partial A_{mj}}{\partial x_j} \right)^T.$$

Theorem 2.1.2 (The transport theorem). *Let the function F have continuous and bounded first order derivatives into \mathcal{M} and let $\mathbf{v} : \mathcal{M} \rightarrow \mathbb{R}^d$, $\mathbf{v}(\mathbf{x}, t) = \frac{\partial}{\partial t} \boldsymbol{\varphi}(\mathbf{X}, t)$, be the velocity field of the fluid. Then, the time derivative of \mathcal{F} can be expressed by*

$$\frac{d\mathcal{F}}{dt}(t) = \int_{\mathcal{V}(t)} \left[\frac{\partial F}{\partial t}(\mathbf{x}, t) + \operatorname{div}(F\mathbf{v})(\mathbf{x}, t) \right] d\mathbf{x}, \quad (2.2)$$

where the divergence operator applied to a tensor was presented in Def. 2.1.1.

2.1.2. The continuity equation (mass conservation)

Let $\rho : \mathcal{M} \rightarrow \mathbb{R}_{>0}$ be the density function, thus the *total mass* of fluid into the control volume $\mathcal{V}(t)$ can be expressed as

$$m(t) = \int_{\mathcal{V}(t)} \rho(\mathbf{x}, t) d\mathbf{x}$$

and its time derivative as

$$\frac{dm}{dt}(t) = \frac{d}{dt} \int_{\mathcal{V}(t)} \rho(\mathbf{x}, t) d\mathbf{x}.$$

Because of the fluid particles in the control volume are the same in any time, this time derivative is equal to zero.

$$\frac{dm}{dt}(t) = 0.$$

In other words, the mass into the control volume is conserved. Now, applying the Theorem 2.1.2, we get the equation

$$\int_{\mathcal{V}(t)} \left[\frac{\partial \rho}{\partial t}(\mathbf{x}, t) + \operatorname{div}(\rho\mathbf{v})(\mathbf{x}, t) \right] d\mathbf{x} = 0.$$

Considering that this expression is true for any arbitrary control volume $\mathcal{V}(t)$, then the equation

$$\frac{\partial \rho}{\partial t} + \operatorname{div}(\rho\mathbf{v}) = 0 \quad (2.3)$$

holds.

2.1.3. The momentum equations (momentum conservation)

Applying the second Newton's law to the control volume $\mathcal{V}(t)$, we get that the rate of change of the *total momentum* of $\mathcal{V}(t)$ is equal to the net force acting on this volume. Then, if $\mathcal{H}(\mathcal{V}(t)) \in \mathbb{R}^d$ is the total momentum of $\mathcal{V}(t)$, expressed as

$$\mathcal{H}(\mathcal{V}(t)) = \int_{\mathcal{V}(t)} \rho(\mathbf{x}, t) \mathbf{v}(\mathbf{x}, t) d\mathbf{x}$$

and $\mathcal{F}(\mathcal{V}(t)) = (\mathcal{F}_i(\mathcal{V}(t)))_i \in \mathbb{R}^d$ is the resultant force over the control volume, the motion equation for this body is

$$\frac{d\mathcal{H}(\mathcal{V}(t))}{dt} = \mathcal{F}(\mathcal{V}(t)).$$

Now, using the transport Theorem 2.1.2, we get the system of equations

$$\begin{aligned} \int_{\mathcal{V}(t)} \left[\frac{\partial}{\partial t} (\rho(\mathbf{x}, t) v_i(\mathbf{x}, t)) + \operatorname{div}(\rho(\mathbf{x}, t) v_i(\mathbf{x}, t) \mathbf{v}(\mathbf{x}, t)) \right] d\mathbf{x} \\ = \mathcal{F}_i(\mathcal{V}(t)), \quad i = 1, \dots, d. \end{aligned} \quad (2.4)$$

Further, we should distinguish two kind of forces acting on the volume, the *volume force* and the *surface force*. The volume force $\mathcal{F}_v(\mathcal{V}, t) \in \mathbb{R}^d$ acts on every fluid particle in the control volume in such way that if we define a force density $\mathbf{f}(\mathbf{x}, t) \in \mathbb{R}^d$, the total volume force is calculated by

$$\mathcal{F}_v(\mathcal{V}(t), t) = \int_{\mathcal{V}(t)} \rho(\mathbf{x}, t) \mathbf{f}(\mathbf{x}, t) d\mathbf{x}.$$

On the other hand, the resultant surface force is calculated using the loads on the surface of the control volume

$$\mathcal{F}_s(\mathcal{V}(t), t) = \int_{\partial\mathcal{V}(t)} \mathbf{T}(\mathbf{x}, t, \mathbf{n}(\mathbf{x}, t)) d\mathbf{x},$$

where $\mathbf{n}(\mathbf{x}, t) = (n_j)_j \in \mathbb{R}^d$ is the unit vector outer normal to $\partial\mathcal{V}$ and $\mathbf{T}(\mathbf{x}, t, \mathbf{n}) = (T_i(\mathbf{x}, t, \mathbf{n}))_i \in \mathbb{R}^d$ is the stress vector. This vector can be written using the stress tensor $\mathcal{T}(\mathbf{x}, t) = (\mathcal{T}_{ij}(\mathbf{x}, t))_{i,j} \in \mathbb{R}^{d,d}$ in the way

$$T_i(\mathbf{x}, t) = \sum_{j=1}^d \mathcal{T}_{ij}(\mathbf{x}, t) n_j(\mathbf{x}, t).$$

Introducing the previous definitions for \mathcal{F}_v and \mathcal{F}_s into the Eq. 2.4, we get

$$\begin{aligned} & \int_{\mathcal{V}(t)} \left[\frac{\partial}{\partial t} (\rho(\mathbf{x}, t) v_i(\mathbf{x}, t)) + \operatorname{div}(\rho(\mathbf{x}, t) v_i(\mathbf{x}, t) \mathbf{v}(\mathbf{x}, t)) \right] d\mathbf{x} \\ &= \int_{\mathcal{V}(t)} \rho(\mathbf{x}, t) f_i(\mathbf{x}, t) d\mathbf{x} + \int_{\partial\mathcal{V}(t)} \sum_{j=1}^d \mathcal{T}_{ij}(\mathbf{x}, t) n_j(\mathbf{x}, t) dS, \end{aligned}$$

or equivalently (using Green's theorem)

$$\begin{aligned} & \int_{\mathcal{V}(t)} \left[\frac{\partial}{\partial t} (\rho(\mathbf{x}, t) v_i(\mathbf{x}, t)) + \operatorname{div}(\rho(\mathbf{x}, t) v_i(\mathbf{x}, t) \mathbf{v}(\mathbf{x}, t)) \right] d\mathbf{x} \\ &= \int_{\mathcal{V}(t)} \left[\rho(\mathbf{x}, t) f_i(\mathbf{x}, t) + \sum_{j=1}^d \frac{\partial}{\partial x_j} \mathcal{T}_{ij}(\mathbf{x}, t) \right] d\mathbf{x}. \end{aligned}$$

Now, considering an arbitrary control volume in the last expression, it is possible to write

$$\frac{\partial}{\partial t} (\rho \mathbf{v}) + \operatorname{div}(\rho \mathbf{v} \otimes \mathbf{v}) = \rho \mathbf{f} + \operatorname{div} \mathcal{T}. \quad (2.5)$$

In Eq. 2.5 the stress tensor is composed by a *hydrostatic pressure* term plus a *viscous stress* component $\boldsymbol{\tau}(\mathbf{v})(\mathbf{x}, t) \in \mathbb{R}^{d,d}$, i.e.,

$$\mathcal{T} = -p \mathbb{I} + \boldsymbol{\tau}, \quad (2.6)$$

where $p(\mathbf{x}, t) \in \mathbb{R}$ is the *pressure* of the fluid. There are multiple physical models to describe the rheological behavior of a fluid, but a simple and commonly used model assumes that the viscous shear stress $\boldsymbol{\tau}$ is linearly proportional to the *deformation velocity tensor* $\mathbb{D}(\mathbf{v})(\mathbf{x}, t) \in \mathbb{R}^{d,d}$, which is defined as

$$\mathbb{D}_{ij}(\mathbf{v}) = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right), \quad i, j = 1, \dots, d.$$

If this condition is satisfied, the fluid is called *Newtonian* and the viscous stress can be expressed as

$$\boldsymbol{\tau} = 2\mu \left(\mathbb{D}(\mathbf{v}) - \frac{1}{3} \operatorname{div}(\mathbf{v}) \mathbb{I} \right), \quad (2.7)$$

where $\mu(\mathbf{x}, t) \in \mathbb{R}$ is the *dynamic viscosity* and $\mathbb{I} \in \mathbb{R}^{d,d}$ is the identity matrix.

Finally, including this definition for the stress tensor in Eq. 2.6, the *momentum equation* Eq. 2.5 is

$$\frac{\partial}{\partial t}(\rho \mathbf{v}) + \operatorname{div}(\rho \mathbf{v} \otimes \mathbf{v}) = \rho \mathbf{f} - \nabla p + \operatorname{div} \boldsymbol{\tau}. \quad (2.8)$$

2.1.4. The energy equation (energy conservation)

Writing the energy balance for the control volume $\mathcal{V}(t)$, we can express that the rate of change of the total energy $\mathcal{E}(\mathcal{V}(t)) \in \mathbb{R}$ is equal to the power developed by all forces (volume and surface forces) plus the total amount of heat transmitted to the control volume $\mathcal{Q}(\mathcal{V}(t)) \in \mathbb{R}$. This balance can be written as

$$\frac{d}{dt} \mathcal{E}(\mathcal{V}(t)) = \int_{\mathcal{V}(t)} \rho(\mathbf{x}, t) \mathbf{f} \cdot \mathbf{v} \, d\mathbf{x} + \int_{\partial \mathcal{V}(t)} \mathbf{T}(\mathbf{x}, t, \mathbf{n}(\mathbf{x}, t)) \cdot \mathbf{v}(\mathbf{x}, t) \, dS + \mathcal{Q}(\mathcal{V}(t)).$$

The total energy is calculated as the sum of the *specific energy* $E : \mathcal{M} \rightarrow \mathbb{R}$ over each fluid particle in $\mathcal{V}(t)$, i.e.

$$\mathcal{E}(\mathcal{V}(t)) = \int_{\mathcal{V}(t)} E(\mathbf{x}, t) \, d\mathbf{x},$$

while, the *specific energy* can be expressed as the sum of the *specific internal energy* $e : \mathcal{M} \rightarrow \mathbb{R}$ plus the *kinetic energy*

$$E(\mathbf{x}, t) = \rho(\mathbf{x}, t) e(\mathbf{x}, t) + \frac{1}{2} \rho(\mathbf{x}, t) |\mathbf{v}(\mathbf{x}, t)|^2, \quad (2.9)$$

and the total heat transferred is equal to the heat introduced by *internal heat sources* $q : \mathcal{M} \rightarrow \mathbb{R}$ plus the *conductive heat flux* $\mathbf{q} : \mathcal{M} \rightarrow \mathbb{R}^d$

$$\mathcal{Q}(\mathcal{V}(t)) = \int_{\mathcal{V}(t)} \rho(\mathbf{x}, t) q(\mathbf{x}, t) \, d\mathbf{x} - \int_{\partial \mathcal{V}(t)} \mathbf{q}(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}, t) \, dS,$$

or using the Fourier's law for thermal conductivity

$$\mathcal{Q}(\mathcal{V}(t)) = \int_{\mathcal{V}(t)} \rho(\mathbf{x}, t) q(\mathbf{x}, t) \, d\mathbf{x} + \int_{\partial \mathcal{V}(t)} k(\mathbf{x}, t) \nabla \theta(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}, t) \, dS,$$

where $k(\mathbf{x}, t) \in \mathbb{R}$ is the *thermal conductivity* and $\theta(\mathbf{x}, t) \in \mathbb{R}$ is the *temperature*.

Thus, the *energy balance equation* is written now as

$$\begin{aligned} & \frac{d}{dt} \int_{\mathcal{V}(t)} E(\mathbf{x}, t) \, d\mathbf{x} \\ &= \int_{\mathcal{V}(t)} \rho(\mathbf{x}, t) \mathbf{f}(\mathbf{x}, t) \cdot \mathbf{v}(\mathbf{x}, t) \, d\mathbf{x} + \int_{\partial \mathcal{V}(t)} \sum_{i,j=1}^d \mathcal{T}_{ij}(\mathbf{x}, t) n_j(\mathbf{x}, t) v_i(\mathbf{x}, t) \, dS \\ &+ \int_{\mathcal{V}(t)} \rho(\mathbf{x}, t) q(\mathbf{x}, t) \, d\mathbf{x} + \int_{\partial \mathcal{V}(t)} k(\mathbf{x}, t) \nabla \theta(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}, t) \, dS. \end{aligned}$$

Applying the Theorem 2.1.2, the Eq. 2.6, Green's theorem and proceeding as in the previous cases, we get

$$\begin{aligned} \frac{\partial}{\partial t} E + \operatorname{div}(E \mathbf{v}) &= \rho \mathbf{f} \cdot \mathbf{v} - \operatorname{div}(p \mathbf{v}) \\ &+ \operatorname{div}(\lambda \operatorname{div}(\mathbf{v}) \mathbf{v}) + \operatorname{div}(2\mu \mathbb{D}(\mathbf{v}) \mathbf{v}) + \operatorname{div}(k \nabla \theta) + \rho q. \end{aligned} \quad (2.10)$$

2.1.5. The set of Navier–Stokes equations for fluid dynamics

The set of equations (2.3), (2.8) and (2.10), that represent the balance of mass, momentum and energy, are called *compressible Navier–Stokes equations*. In a compact form, we can write the compressible Navier–Stokes problem as

$$\frac{\partial \mathbf{w}}{\partial t} + \operatorname{div}(\mathbb{F}^s(\mathbf{w})) = \operatorname{div}(\mathbb{F}^v(\mathbf{w}, \nabla \mathbf{w})) + \mathbf{F}(\mathbf{w})$$

$$\text{in } \mathcal{M} = \Omega_t \times (0, T), \quad (2.11)$$

subject to boundary conditions

$$\mathbf{B}(\mathbf{w}, \nabla \mathbf{w}) = 0 \quad \text{on } \partial\Omega \times (0, T), \quad (2.12)$$

and endowed with the initial condition

$$\mathbf{w}(\mathbf{x}, 0) = \mathbf{w}^0(\mathbf{x}) \quad \text{with } \mathbf{x} \in \Omega_0. \quad (2.13)$$

Here $\mathbf{w}(\mathbf{x}, t) \in \mathbb{R}^m$ is the vector of conserved quantities, $\mathbb{F}^s(\mathbf{w}) = (\mathbf{f}_1^s \dots \mathbf{f}_d^s) = (\mathbf{f}_j^s(\mathbf{w}))_j \in \mathbb{R}^{m,d}$ is the *inviscid flux tensor*, $\mathbb{F}^v(\mathbf{w}) = (\mathbf{f}_1^v \dots \mathbf{f}_d^v) = (\mathbf{f}_j^v(\mathbf{w}))_j \in \mathbb{R}^{m,d}$ is the *viscous flux tensor*, $\mathbf{F}(\mathbf{w}) \in \mathbb{R}^m$ is the load vector, $\mathbf{B}(\mathbf{w}, \nabla \mathbf{w})$ is some boundary operator and $m = d + 2$ is the number of system components. Particularly, for $d = 3$ these quantities are defined as

$$\mathbf{w} = \begin{pmatrix} \rho & \rho v_1 & \rho v_2 & \rho v_3 & E \end{pmatrix}^T, \quad (2.14)$$

$$\mathbb{F}^s(\mathbf{w}) = \begin{pmatrix} \rho v_1 & \rho v_2 & \rho v_3 \\ \rho v_1^2 + p & \rho v_1 v_2 & \rho v_1 v_3 \\ \rho v_2 v_1 & \rho v_2^2 + p & \rho v_2 v_3 \\ \rho v_3 v_1 & \rho v_3 v_2 & \rho v_3^2 + p \\ (E + p)v_1 & (E + p)v_2 & (E + p)v_3 \end{pmatrix}, \quad (2.15)$$

$$\mathbb{F}^v(\mathbf{w}, \nabla \mathbf{w}) = \begin{pmatrix} 0 & 0 & 0 \\ \tau_{11} & \tau_{21} & \tau_{31} \\ \tau_{12} & \tau_{22} & \tau_{32} \\ \tau_{13} & \tau_{23} & \tau_{33} \\ \sum_{i=1}^3 \tau_{1i} v_i + k \frac{\partial \theta}{\partial x_1} & \sum_{i=1}^3 \tau_{2i} v_i + k \frac{\partial \theta}{\partial x_2} & \sum_{i=1}^3 \tau_{3i} v_i + k \frac{\partial \theta}{\partial x_3} \end{pmatrix}, \quad (2.16)$$

$$\mathbf{F}(\mathbf{w}) = \begin{pmatrix} 0 & \rho f_1 & \rho f_2 & \rho f_3 & \rho \mathbf{f} \cdot \mathbf{v} + \rho q \end{pmatrix}^T, \quad (2.17)$$

where τ_{ij} , for $i, j = 1, \dots, d$, denotes the corresponding component of the viscous stress in Eq. 2.7.

2.1.6. Equation of state

This set of Navier–Stokes equations (Eq. 2.11) should be closed by a *equation of state* (EOS), usually with the form

$$p = p(\rho, \theta) \quad \text{or} \quad p = p(\rho, e). \quad (2.18)$$

Under gas conditions such that the work of the molecular interactions is much lower than the kinetic energy of the molecules, a basic gas model (called *ideal gas model*) can be used. Usually these ideal gas conditions are high temperatures and low pressures. For some gases like oxygen, nitrogen, hydrogen and carbon dioxide, the ideal gas model can be used under moderate temperature and pressure conditions. According to the ideal gas model

$$\frac{p}{\rho R \theta} = 1, \quad (2.19)$$

where $R = C_p - C_v \in \mathbb{R}_{>0}$ is the *specific gas constant*, $C_p \in \mathbb{R}_{>0}$ is the *isobaric specific heat* and $C_v \in \mathbb{R}_{>0}$ is the *isochoric specific heat*. By definition, for an ideal gas the specific internal energy can be expressed as

$$e = C_v \theta, \quad (2.20)$$

and thus the EOS Eq. 2.19 will be

$$p = (\gamma - 1) \rho e, \quad (2.21)$$

or using the conservative variables (through Eq. 2.9)

$$p = (\gamma - 1) \left(E - \frac{\rho |\mathbf{v}|^2}{2} \right), \quad (2.22)$$

where $\gamma = C_p/C_v \in \mathbb{R}_{>1}$ is the *heat capacity ratio*. From the previous equations also it is possible to write

$$\theta = \frac{1}{C_v} \left(E - \frac{\rho |\mathbf{v}|^2}{2} \right). \quad (2.23)$$

Nevertheless, to high pressures when the molecular interactions are important to describe the gas behavior, the gas ideal simplification is not valid and a real gas model should be used. Some of these more complex models for EOS are presented in Sec. 2.4.

2.1.7. Dimensionless form of Navier–Stokes equations

The set of Navier–Stokes equations is a dimensionally homogeneous system. However, to use these equations in a non-dimensional form results more convenient from different perspectives. From the physical point of view, a mathematical model built via the dimensionless Navier–Stokes equations is able to represent not only a particular physical case but also a wide set of similar cases. From a numerical perspective, the solution of this set of equations involve conservative variables with different orders of magnitude, this can cause numerical instabilities. To use a dimensionless form let to keep all the variables in a common range (ideally 0 to 1). There are multiple ways to reduce the Eq. 2.11 to a dimensionless form ([Mon13]). Here a technique commonly used for compressible flows is presented ([FFS03]). Defining reference quantities for length L^* , velocity U^* , density ρ^* , force density f^* , viscosity μ^* and thermal conductivity k^* , it is possible to write dimensionless variables as follows:

$$\begin{aligned} \tilde{\mathbf{x}} &= \frac{\mathbf{x}}{L^*}, & \tilde{t} &= \frac{tU^*}{L^*}, & \tilde{\mathbf{v}} &= \frac{\mathbf{v}}{U^*}, & \tilde{\rho} &= \frac{\rho}{\rho^*}, \\ \tilde{p} &= \frac{p}{\rho^*U^{*2}}, & \tilde{E} &= \frac{E}{\rho^*U^{*2}}, & \tilde{\theta} &= \frac{C_v\theta}{U^{*2}}, & \tilde{\mathbf{f}} &= \frac{\mathbf{f}}{f^*}, \\ \tilde{k} &= \frac{k}{k^*}, & \tilde{\mu} &= \frac{\mu}{\mu^*}, & \tilde{\lambda} &= \frac{\lambda}{\mu^*}, & \tilde{q} &= \frac{qL^*}{U^{*3}}. \end{aligned} \quad (2.24)$$

Introducing the definitions in Eq. 2.24, the Eq. 2.11 can be written using dimensionless variables

$$\frac{\partial \tilde{\mathbf{w}}}{\partial \tilde{t}} + \operatorname{div}(\mathbb{F}^s(\tilde{\mathbf{w}})) = \frac{1}{\operatorname{Re}} \operatorname{div}(\mathbb{F}^v(\tilde{\mathbf{w}}, \nabla \tilde{\mathbf{w}})) + \tilde{\mathbf{F}}(\tilde{\mathbf{w}}) \quad \text{in } \mathcal{M} = \tilde{\Omega}_{\tilde{t}} \times (0, \tilde{T}), \quad (2.25)$$

where, for $d = 3$

$$\mathbf{w} = \left(\tilde{\rho} \quad \tilde{\rho} \tilde{v}_1 \quad \tilde{\rho} \tilde{v}_2 \quad \tilde{\rho} \tilde{v}_3 \quad \tilde{E} \right)^T, \quad (2.26)$$

$$\mathbb{F}^v(\tilde{\mathbf{w}}, \nabla \tilde{\mathbf{w}}) = \begin{pmatrix} 0 & 0 & 0 \\ \tilde{\tau}_{11} & \tilde{\tau}_{21} & \tilde{\tau}_{31} \\ \tilde{\tau}_{12} & \tilde{\tau}_{22} & \tilde{\tau}_{32} \\ \tilde{\tau}_{13} & \tilde{\tau}_{23} & \tilde{\tau}_{33} \\ \sum_{i=1}^3 \tilde{\tau}_{1i} \tilde{v}_i + \frac{\gamma \tilde{k}}{\operatorname{Pr}} \frac{\partial \tilde{\theta}}{\partial \tilde{x}_1} & \sum_{i=1}^3 \tilde{\tau}_{2i} \tilde{v}_i + \frac{\gamma \tilde{k}}{\operatorname{Pr}} \frac{\partial \tilde{\theta}}{\partial \tilde{x}_2} & \sum_{i=1}^3 \tilde{\tau}_{3i} \tilde{v}_i + \frac{\gamma \tilde{k}}{\operatorname{Pr}} \frac{\partial \tilde{\theta}}{\partial \tilde{x}_3} \end{pmatrix}, \quad (2.27)$$

$$\tilde{\mathbf{F}}(\tilde{\mathbf{w}}) = \left(0 \quad \frac{\tilde{\rho} \tilde{f}_1}{\operatorname{Fr}^2} \quad \frac{\tilde{\rho} \tilde{f}_2}{\operatorname{Fr}^2} \quad \frac{\tilde{\rho} \tilde{f}_3}{\operatorname{Fr}^2} \quad \tilde{\rho} \tilde{\mathbf{f}} \cdot \tilde{\mathbf{v}} + \tilde{\rho} \tilde{q} \right)^T, \quad (2.28)$$

$$\tilde{\boldsymbol{\tau}} = \tilde{\lambda} \operatorname{div} \tilde{\mathbf{v}} \mathbb{I} + 2 \tilde{\mu} \mathbb{D}(\tilde{\mathbf{v}}), \quad (2.29)$$

and Fr , Re and Pr are the *Froude*, *Reynolds* and *Prandtl* numbers, respectively. These dimensionless numbers are defined as:

$$\operatorname{Fr} = \frac{U^*}{\sqrt{L^* f^*}}, \quad \operatorname{Re} = \frac{\rho^* U^* L^*}{\mu^*}, \quad \operatorname{Pr} = \frac{C_p \mu^*}{k^*}. \quad (2.30)$$

In Eq. 2.25 and Eq. 2.27 the operators ∇ and div , as well as the tensor \mathbb{D} (into the viscous stress tensor), are calculated with respect to the dimensionless coordinates \tilde{x} . In similar fashion, the thermodynamical relations (2.22) and (2.23) can be written using dimensionless variables as follows

$$\tilde{p} = (\gamma - 1) \left(\tilde{E} - \frac{\tilde{\rho} |\tilde{\mathbf{v}}|^2}{2} \right), \quad (2.31)$$

$$\tilde{\theta} = \frac{\tilde{E}}{\tilde{\rho}} - \frac{|\tilde{\mathbf{v}}|^2}{2}. \quad (2.32)$$

Onwards the Navier–Stokes equations in (2.25) will be considered, but for simplicity the tilde character will be omitted.

2.1.8. Navier–Stokes equations in cylindrical coordinates

In the previous analysis a rectangular system of coordinates was chosen, such that $\mathbf{e}_1 \times \mathbf{e}_2 = \mathbf{e}_3$, where the unit vector \mathbf{e}_i , for $i = 1, 2, 3$, defines the axis direction. Now, a cylindrical coordinate system will be chosen. In this system \mathbf{e}_1 represents the axial direction, \mathbf{e}_2 the radial direction and \mathbf{e}_3 the azimuthal direction. Considering this coordinate system, the operators ∇ and div can be expressed as follows [KCG05].

Definition 2.1.3. If $(\mathbb{A})_{ij} = (a_{ij})_{ij} \in \mathbb{R}^{m,3}$ is a tensor function on $\mathbf{x} \in \mathbb{R}^d$, where the x -components correspond with a cylindrical coordinates system, then the $\overline{\nabla}$ operator is defined as

$$\overline{\nabla} = \left(\frac{\partial}{\partial x_1} \quad \frac{\partial}{\partial x_2} \quad \frac{1}{x_2} \frac{\partial}{\partial x_3} \right)$$

and the $\overline{\text{div}}$ operator is defined as

$$\overline{\text{div}}\mathbb{A} = \begin{pmatrix} \frac{\partial a_{11}}{\partial x_1} + \frac{\partial a_{12}}{\partial x_2} + \frac{1}{x_2} \frac{\partial a_{13}}{\partial x_3} \\ \vdots \\ \frac{\partial a_{m1}}{\partial x_1} + \frac{\partial a_{m2}}{\partial x_2} + \frac{1}{x_2} \frac{\partial a_{m3}}{\partial x_3} \end{pmatrix}.$$

Using cylindrical coordinates ($d = 3$) and the operators in Def. 2.1.3, the set of Navier–Stokes equations is rewritten as

$$\frac{\partial \mathbf{w}}{\partial t} + \overline{\text{div}}(\mathbb{F}^s(\mathbf{w})) = \frac{1}{\text{Re}} \overline{\text{div}}(\mathbb{F}^v(\mathbf{w}, \overline{\nabla} \mathbf{w})) - \frac{1}{x_2} \mathbf{D} + \mathbf{F}(\mathbf{w}), \quad (2.33)$$

where

$$\mathbf{D} = \begin{pmatrix} \rho v_2 \\ \rho v_1 v_2 - \tau_{21}/\text{Re} \\ \rho v_2^2 - \rho v_3^2 - (\tau_{22} - \tau_{33})/\text{Re} \\ 2(\rho v_2 v_3 - \tau_{32}/\text{Re}) \\ v_2(E + p) - \frac{1}{\text{Re}} \left(\sum_{i=1}^3 \tau_{2i} v_i + \frac{\gamma^k}{\text{Pr}} \frac{\partial \theta}{\partial x_2} \right) \end{pmatrix}.$$

Some 3D-problems have a behavior symmetric respect to an axis. These models, called axisymmetric models, can be reduced to a 2D-problem analyzing only the plane that by revolution generates the full volume. This symmetry condition implies

$$\frac{\partial}{\partial x_3} = 0. \quad (2.34)$$

Thus, Navier–Stokes equations for an axisymmetric model can be getting from Eq. 2.33 and applying the condition Eq. 2.34. Further, if the velocity component in the azimuthal direction (swirl component) can be neglected, the system of Navier–Stokes equations becomes [Kim03]

$$\frac{\partial \mathbf{w}}{\partial t} + \overline{\text{div}}(\overline{\mathbb{F}}^s(\mathbf{w})) = \frac{1}{\text{Re}} \overline{\text{div}}(\overline{\mathbb{F}}^v(\mathbf{w}, \overline{\nabla} \mathbf{w})) - \frac{1}{x_2} \overline{\mathbf{D}} + \mathbf{F}(\mathbf{w}), \quad (2.35)$$

where

$$\mathbf{w} = \begin{pmatrix} \rho & \rho v_1 & \rho v_2 & E \end{pmatrix}^T, \quad (2.36)$$

$$\overline{\mathbb{F}}^s(\mathbf{w}) = \begin{pmatrix} \rho v_1 & \rho v_2 \\ \rho v_1^2 + p & \rho v_1 v_2 \\ \rho v_2 v_1 & \rho v_2^2 + p \\ (E + p)v_1 & (E + p)v_2 \end{pmatrix}, \quad (2.37)$$

$$\overline{\mathbb{F}}^v(\mathbf{w}, \overline{\nabla} \mathbf{w}) = \begin{pmatrix} 0 & 0 \\ \tau_{11} & \tau_{21} \\ \tau_{12} & \tau_{22} \\ \sum_{i=1}^2 \tau_{1i} v_i + \frac{\gamma^k}{\text{Pr}} \frac{\partial \theta}{\partial x_1} & \sum_{i=1}^2 \tau_{2i} v_i + \frac{\gamma^k}{\text{Pr}} \frac{\partial \theta}{\partial x_2} \end{pmatrix}, \quad (2.38)$$

$$\overline{\mathbf{D}} = \begin{pmatrix} \rho v_2 \\ \rho v_1 v_2 - \tau_{21}/\text{Re} \\ \rho v_2^2 - (\tau_{22} - \tau_{33})/\text{Re} \\ v_2(E + p) - \frac{1}{\text{Re}} \left(\sum_{i=1}^2 \tau_{2i} v_i + \frac{\gamma k}{\text{Pr}} \frac{\partial \theta}{\partial x_2} \right) \end{pmatrix} \quad (2.39)$$

and, $\overline{\text{div}}$ and τ_{ij} defined as previous with $\partial/\partial x_3 = 0$ and $v_3 = 0$.

2.2. Inviscid case: Euler equations

Previously the conservation equations for mass, momentum and energy for a fluid were presented. Now, considering an adiabatic (i.e., the heat conduction is neglected) and inviscid fluid (i.e., all the viscous stress are neglected), and ignoring any body force, the Navier–Stokes problem Eq. 2.11 is reduced to

$$\frac{\partial \mathbf{w}}{\partial t} + \text{div}(\mathbb{F}^s(\mathbf{w})) = 0 \quad \text{in } \mathcal{M} = \Omega_t \times (0, T), \quad (2.40)$$

subject to the boundary conditions

$$\mathcal{B}^s(\mathbf{w}) = \mathcal{B}^s(\mathbf{w}^D) \quad \text{on } \partial\Omega \times (0, T) \quad (2.41)$$

and equipped with the initial condition

$$\mathbf{w}(\mathbf{x}, 0) = \mathbf{w}^0(\mathbf{x}) \quad \text{with } \mathbf{x} \in \Omega_0. \quad (2.42)$$

Here \mathbf{w}^D represents the prescribed boundary values and \mathcal{B}^s a boundary operator used to enforce properly the boundary conditions, i.e. in such a way that the problem Eq. 2.40–Eq. 2.42 is well-posed. More about inviscid boundary conditions is discussed in Sec. 2.3 and Sec. 3.6. The system composed by Eq. 2.40–Eq. 2.42 is called *compressible Euler equations*.

2.2.1. Quasilinear form of Euler equations

Let $\mathbf{F}^n(\mathbf{w}, \mathbf{n}) \in \mathbb{R}^m$ be the projection of $\mathbb{F}^s(\mathbf{w}) = (\mathbf{f}_1^s \dots \mathbf{f}_d^s) = (\mathbf{f}_j^s(\mathbf{w}))_j \in \mathbb{R}^{m,d}$, over an arbitrary direction $\mathbf{n} = (n_j)_j \in \mathbb{R}^d$, i.e.

$$\mathbf{F}^n(\mathbf{w}, \mathbf{n}) = \sum_{j=1}^d \mathbf{f}_j^s(\mathbf{w}) n_j,$$

and let $\mathbf{P}(\mathbf{w}, \mathbf{n}) \in \mathbb{R}^{m,m}$ be the Jacobi matrix of \mathbf{F}^n

$$\mathbf{P}(\mathbf{w}, \mathbf{n}) = \frac{\partial \mathbf{F}^n(\mathbf{w}, \mathbf{n})}{\partial \mathbf{w}} = \sum_{j=1}^d \mathbb{A}_j(\mathbf{w}) n_j, \quad (2.43)$$

where

$$\mathbb{A}_j(\mathbf{w}) = \frac{\partial \mathbf{f}_j^s(\mathbf{w})}{\partial \mathbf{w}}.$$

Thus, using the chain rule in the Euler equations Eq. 2.40 it is possible to write this set of equations in a quasilinear form

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{j=1}^d \mathbb{A}_j(\mathbf{w}) \frac{\partial \mathbf{w}}{\partial x_j} = 0. \quad (2.44)$$

2.2.2. Homogeneity property of the inviscid flux vector

Lemma 2.2.1. *Let $d = 3$ and let $p = p(\rho, e)$ be the EOS with derivatives $\frac{\partial p}{\partial \rho} = p_\rho$ and $\frac{\partial p}{\partial e} = p_e$, thus the matrix $\mathbf{P}(\mathbf{w}, \mathbf{n})$ can be written as a sum of homogeneous and inhomogeneous parts ([LLS90])*

$$\mathbf{P}(\mathbf{w}, \mathbf{n}) = \mathbf{P}^h(\mathbf{w}, \mathbf{n}) + \mathbf{P}^\rho(\mathbf{w}, \mathbf{n}), \quad (2.45)$$

where

$$\mathbf{P}^h(\mathbf{w}, \mathbf{n}) = \begin{pmatrix} 0 & n_1 & n_2 & n_3 & 0 \\ -v_1 v_n + \frac{p}{\rho} n_1 & v_n + v_1 n_1 & v_1 n_2 & v_1 n_3 & 0 \\ -v_2 v_n + \frac{p}{\rho} n_2 & v_2 n_1 & v_n + v_2 n_2 & v_2 n_3 & 0 \\ -v_3 v_n + \frac{p}{\rho} n_3 & v_3 n_1 & v_3 n_2 & v_n + v_3 n_3 & 0 \\ -v_n \frac{E}{\rho} & h n_1 & h n_2 & h n_3 & v_n \end{pmatrix} \quad (2.46)$$

$$- \frac{p_e}{\rho} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ (e - |\mathbf{v}|^2/2) n_1 & v_1 n_1 & v_2 n_1 & v_3 n_1 & -n_1 \\ (e - |\mathbf{v}|^2/2) n_2 & v_1 n_2 & v_2 n_2 & v_3 n_2 & -n_2 \\ (e - |\mathbf{v}|^2/2) n_3 & v_1 n_3 & v_2 n_3 & v_3 n_3 & -n_3 \\ (e - |\mathbf{v}|^2/2) v_n & v_1 v_n & v_2 v_n & v_3 v_n & -v_n \end{pmatrix},$$

$$\mathbf{P}^\rho(\mathbf{w}, \mathbf{n}) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ (p_\rho - p/\rho) n_1 & 0 & 0 & 0 & 0 \\ (p_\rho - p/\rho) n_2 & 0 & 0 & 0 & 0 \\ (p_\rho - p/\rho) n_3 & 0 & 0 & 0 & 0 \\ (p_\rho - p/\rho) v_n & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (2.47)$$

h is the specific enthalpy defined by

$$h = \frac{1}{\rho} (E + p) \quad (2.48)$$

and v_n the normal velocity

$$v_n = \sum_{j=1}^d v_j n_j. \quad (2.49)$$

Remark 2.2.2. *Using the EOS for ideal gas (Eq. 2.21) into the matrix $\mathbf{P}(\mathbf{w}, \mathbf{n})$ in Lemma 2.2.1, it is found that $\mathbf{P}^\rho(\mathbf{w}, \mathbf{n}) = 0$ and thus the complete matrix takes the values ([Roh01])*

$$\mathbf{P}(\mathbf{w}, \mathbf{n}) = \begin{pmatrix} 0 & n_1 & n_2 & n_3 & 0 \\ \gamma_1 \frac{|\mathbf{v}|^2}{2} n_1 - v_1 v_n & v_n - \gamma_2 v_1 n_1 & v_1 n_2 - \gamma_1 v_2 n_1 & v_1 n_3 - \gamma_1 v_3 n_1 & \gamma_1 n_1 \\ \gamma_1 \frac{|\mathbf{v}|^2}{2} n_2 - v_2 v_n & v_2 n_1 - \gamma_1 v_1 n_2 & v_n - \gamma_2 v_2 n_2 & v_2 n_3 - \gamma_1 v_3 n_2 & \gamma_1 n_2 \\ \gamma_1 \frac{|\mathbf{v}|^2}{2} n_3 - v_3 v_n & v_3 n_1 - \gamma_1 v_1 n_3 & v_3 n_2 - \gamma_1 v_2 n_3 & v_n - \gamma_2 v_3 n_3 & \gamma_1 n_3 \\ \left(\gamma_1 \frac{|\mathbf{v}|^2}{2} - h \right) v_n & h n_1 - \gamma_1 v_1 v_n & h n_2 - \gamma_1 v_2 v_n & h n_3 - \gamma_1 v_3 v_n & \gamma v_n \end{pmatrix}, \quad (2.50)$$

where $\gamma_1 = \gamma - 1$ and $\gamma_2 = \gamma - 2$.

Theorem 2.2.3. *Let $p = p(\rho, e)$ be a general EOS, then the inviscid flux vector $\mathbf{f}_j^s(\mathbf{w})$ in Eq. 2.40 is a homogeneous function of degree one, and therefore can be written as*

$$\mathbf{f}_j^s(\mathbf{w}) = \mathbb{A}_j(\mathbf{w})\mathbf{w}, \quad (2.51)$$

if and only if the pressure $p(\rho, e)$ depends linearly of ρ [FFS03], such as in the case of a thermally perfect gas, for which the EOS is given by

$$p(\rho, e) = \rho\theta(e). \quad (2.52)$$

Proof. Apply the matrix $\mathbf{P}(\mathbf{w}, \mathbf{n})$ in Eq. 2.45 to the vector \mathbf{w}

$$\begin{aligned} \mathbf{P}(\mathbf{w}, \mathbf{n})\mathbf{w} &= \mathbf{P}^h(\mathbf{w}, \mathbf{n})\mathbf{w} + \mathbf{P}^\rho(\mathbf{w}, \mathbf{n})\mathbf{w} \\ &= \sum_{j=1}^d \mathbb{A}_j^h(\mathbf{w})n_j\mathbf{w} + \sum_{j=1}^d \mathbb{A}_j^\rho(\mathbf{w})n_j\mathbf{w}, \end{aligned} \quad (2.53)$$

where $\mathbb{A}_j^h(\mathbf{w})$ and $\mathbb{A}_j^\rho(\mathbf{w})$ are defined by Eq. 2.43, and verify that

$$\mathbb{A}_j^h(\mathbf{w})\mathbf{w} = \mathbf{f}_j^s.$$

As from Eq. 2.43 and Eq. 2.53

$$\mathbb{A}_j(\mathbf{w}) = \mathbb{A}_j^h(\mathbf{w}) + \mathbb{A}_j^\rho(\mathbf{w}),$$

and, in order to satisfy the expression Eq. 2.51, the non-homogeneous component $\mathbb{A}_j^\rho(\mathbf{w})$ should be equal to zero. Thus, using Eq. 2.47 it is concluded that the pressure function should satisfy the condition

$$p(\rho, e) = \rho \frac{\partial p}{\partial \rho}.$$

This condition is clearly satisfied by a thermally perfect gas. \square

Remark 2.2.4. *The ideal gas can be considered a perfect gas and therefore the inviscid flux vectors can be expressed as in Eq. 2.51. This condition can be easily verified using the expression in Remark 2.2.2, from where $\mathbb{A}_j(\mathbf{w})$ is only composed by the homogeneous component.*

2.2.3. Hyperbolicity of Euler equations

Lemma 2.2.5. *Let $d = 3$ and let $p = p(\rho, e)$ be a general EOS with the derivatives $\frac{\partial p}{\partial \rho} = p_\rho$ and $\frac{\partial p}{\partial e} = p_e$, then the matrix $\mathbf{P}(\mathbf{w}, \mathbf{n})$ has a set of eigenvalues defined by ([Gla89])*

$$\boldsymbol{\lambda} = \begin{pmatrix} v_n - c & v_n & v_n + c & v_n & v_n \end{pmatrix}, \quad (2.54)$$

where c is the speed of sound expressed as

$$c^2 = \frac{p}{\rho^2} p_e + p_\rho. \quad (2.55)$$

Further, if we compute the eigenvalues of matrix components $\mathbf{P}^h(\mathbf{w}, \mathbf{n})$ and $\mathbf{P}^\rho(\mathbf{w}, \mathbf{n})$, we get respectively ([LLS90])

$$\boldsymbol{\lambda}^h = \begin{pmatrix} v_n - c_e & v_n & v_n + c_e & v_n & v_n \end{pmatrix} \quad (2.56)$$

and

$$\boldsymbol{\lambda}^\rho = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad (2.57)$$

where $c_e^2 = c^2 - \left(p_\rho - \frac{p}{\rho}\right)$.

Lemma 2.2.6. *Let $d = 3$ and let $p = p(\rho, e)$ be a general EOS with the derivatives $\frac{\partial p}{\partial \rho} = p_\rho$ and $\frac{\partial p}{\partial e} = p_e$, then the matrix $\mathbb{A}_1(\mathbf{w})$ has a set of eigenvalues given by*

$$\boldsymbol{\lambda} = \begin{pmatrix} v_1 - c & v_1 & v_1 + c & v_1 & v_1 \end{pmatrix}, \quad (2.58)$$

and a set of eigenvectors

$$\begin{aligned} \mathbf{R} &= \begin{pmatrix} \mathbf{R}_1 & \mathbf{R}_2 & \mathbf{R}_3 & \mathbf{R}_4 & \mathbf{R}_5 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ v_1 - c & v_1 & v_1 + c & 0 & 0 \\ v_2 & v_2 & v_2 & 1 & 0 \\ v_3 & v_3 & v_3 & 0 & 1 \\ h - v_1 c & \frac{E}{\rho} - \rho \frac{p_\rho}{p_e} & h + v_1 c & v_2 & v_3 \end{pmatrix}. \end{aligned} \quad (2.59)$$

Further, the inverse of the eigenvector matrix \mathbf{R} is given by

$$\mathbf{R}^{-1} = \frac{1}{2h_o c} \begin{pmatrix} |\mathbf{v}|^2 c + h_o v_1 - e_k c & -(h_o + v_1 c) & -v_2 c & -v_3 c & c \\ 2(h - |\mathbf{v}|^2) c & 2v_1 c & 2v_2 c & 2v_3 c & -2c \\ |\mathbf{v}|^2 c - h_o v_1 - e_k c & h_o - v_1 c & -v_2 c & -v_3 c & c \\ -2h_o v_2 c & 0 & 2h_o c & 0 & 0 \\ -2h_o v_3 c & 0 & 0 & 2h_o c & 0 \end{pmatrix}, \quad (2.60)$$

where $e_k = \frac{E}{\rho} - \rho \frac{p_\rho}{p_e}$ and $h_o = h - e_k$.

Remark 2.2.7. *If the EOS for ideal gas (Eq. 2.21) is used into matrix \mathbb{A}_1 , then the expressions for eigenvalues Eq. 2.58 and eigenvectors Eq. 2.59-Eq. 2.60 are now calculated using*

$$c^2 = \frac{\gamma p}{\rho} = \gamma \gamma_1 e, \quad (2.61)$$

$$e_k = \frac{|\mathbf{v}|^2}{2}, \quad (2.62)$$

$$h = h_o + \frac{|\mathbf{v}|^2}{2} \quad (2.63)$$

and

$$h_o = \gamma e. \quad (2.64)$$

Definition 2.2.8. Let \mathcal{D} be an open set in \mathbb{R}^m . We say that system Eq. 2.44 is hyperbolic in the region \mathcal{D} , if every linear combination of the matrices $\mathbb{A}_j(\mathbf{w})$ has a set of eigenvalues $\boldsymbol{\lambda} = (\lambda_i)_i \in \mathbb{R}^m$ and a complete set of eigenvectors for any $\mathbf{w} \in \mathcal{D}$ ([HLLM98]). Or in other words, if every matrix $\mathbf{P}(\mathbf{w}, \mathbf{n})$ (see Eq. 2.43) is diagonalizable over \mathbb{R} , i.e., there exist a nonsingular matrix $\mathbf{R}(\mathbf{w}, \mathbf{n})$ such that

$$\mathbf{R}^{-1} \mathbf{P} \mathbf{R} = \text{diag}(\lambda_1, \dots, \lambda_m)$$

for any $\mathbf{n} = (n_j)_j \in \mathbb{R}^d$ and $\mathbf{w} \in \mathcal{D}$.

Theorem 2.2.9. *Let $p = p(\rho, e)$ be a general EOS with the derivatives $\frac{\partial p}{\partial \rho} = p_\rho$ and $\frac{\partial p}{\partial e} = p_e$, hence the system of Euler equations is an hyperbolic system only if*

$$c^2 = \frac{p}{\rho^2} p_e + p_\rho > 0. \quad (2.65)$$

Proof. Using Def. 2.2.8 and the result shown in Eq. 2.54, it is possible to conclude that the set of eigenvalues of $\mathbf{P}(\mathbf{w}, \mathbf{n})$ is real only if $c \in \mathbb{R}$, which requires that $c^2 \geq 0$. Additionally, the value $c = 0$ can be excluded in order to ensure a nonsingular matrix $\mathbf{R}(\mathbf{w}, \mathbf{n})$, how can be verified in the results of Lemma 2.2.6. Then, the hyperbolicity condition holds for $c^2 > 0$. \square

Remark 2.2.10. For the ideal gas EOS the speed of sound is given by Eq. 2.61 and the condition Eq. 2.65 is clearly satisfied.

Lemma 2.2.11. Let $d = 3$ and let $p = p(\rho, e)$ be the ideal gas EOS Eq. 2.21, then from Eq. 2.54, the matrix $\mathbf{P}(\mathbf{w}, \mathbf{n})$ has a set of eigenvalues defined by

$$\boldsymbol{\lambda} = \begin{pmatrix} v_n - c & v_n & v_n + c & v_n & v_n \end{pmatrix} \quad (2.66)$$

with the speed of sound c calculated according with Eq. 2.61. Further, this matrix has a complete set of eigenvectors given by ([Roh01])

$$\begin{aligned} \mathbf{R} &= \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{r}_4 & \mathbf{r}_5 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ v_1 - cn_1 & v_1 & v_1 + cn_1 & n_2 & -n_3 \\ v_2 - cn_2 & v_2 & v_2 + cn_2 & -n_1 & 0 \\ v_3 - cn_3 & v_3 & v_3 + cn_3 & 0 & n_1 \\ h - cv_n & e_k & h + cv_n & v_1 n_2 - v_2 n_1 & v_3 n_1 - v_1 n_3 \end{pmatrix}, \end{aligned} \quad (2.67)$$

with h, e_k defined according with Eq. 2.62–Eq. 2.64. The eigenvectors $\mathbf{r}_2, \mathbf{r}_4$ and \mathbf{r}_5 span a subspace, thus any vector in this subspace is also an eigenvector. Therefore, it is also a proper set of eigenvectors

$$\mathbf{R} = \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{r}_4 & \mathbf{r}_6 \end{pmatrix} \quad (2.68)$$

or

$$\mathbf{R} = \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{r}_5 & \mathbf{r}_6 \end{pmatrix}, \quad (2.69)$$

with

$$\mathbf{r}_6 = \begin{pmatrix} 0 & 0 & n_3 & -n_2 & v_2 n_3 - v_3 n_2 \end{pmatrix}^T. \quad (2.70)$$

The matrix \mathbf{R} in Eq. 2.67 is a nonsingular matrix for $n_1 \neq 0$ and its inverse is given by ([Roh01])

$$\mathbf{R}^{-1} = \begin{pmatrix} \frac{\gamma_1 e_k + cv_n}{2c^2} & -\frac{\gamma_1 v_1 + cn_1}{2c^2} & -\frac{\gamma_1 v_2 + cn_2}{2c^2} & -\frac{\gamma_1 v_3 + cn_3}{2c^2} & \frac{\gamma_1}{2c^2} \\ \frac{c^2 - \gamma_1 e_k}{c^2} & \frac{\gamma_1 v_1}{c^2} & \frac{\gamma_1 v_2}{c^2} & \frac{\gamma_1 v_3}{c^2} & -\frac{\gamma_1}{c^2} \\ \frac{\gamma_1 e_k - cv_n}{2c^2} & -\frac{\gamma_1 v_1 - cn_1}{2c^2} & -\frac{\gamma_1 v_2 - cn_2}{2c^2} & -\frac{\gamma_1 v_3 - cn_3}{2c^2} & \frac{\gamma_1}{2c^2} \\ \frac{v_2 - v_n n_2}{n_1} & n_2 & \frac{n_2^2 - 1}{n_1} & \frac{n_2 n_3}{n_1} & 0 \\ \frac{v_n n_3 - v_3}{n_1} & -n_3 & -\frac{n_2 n_3}{n_1} & \frac{1 - n_3^2}{n_1} & 0 \end{pmatrix}. \quad (2.71)$$

Analogously, the eigenvector matrix in Eq. 2.68 is nonsingular for $n_2 \neq 0$ and its inverse

is defined by

$$\mathbf{R}^{-1} = \begin{pmatrix} \frac{\gamma_1 e_k + cv_n}{2c^2} & -\frac{\gamma_1 v_1 + cn_1}{2c^2} & -\frac{\gamma_1 v_2 + cn_2}{2c^2} & -\frac{\gamma_1 v_3 + cn_3}{2c^2} & \frac{\gamma_1}{2c^2} \\ \frac{c^2 - \gamma_1 e_k}{c^2} & \frac{\gamma_1 v_1}{c^2} & \frac{\gamma_1 v_2}{c^2} & \frac{\gamma_1 v_3}{c^2} & -\frac{\gamma_1}{c^2} \\ \frac{\gamma_1 e_k - cv_n}{2c^2} & -\frac{\gamma_1 v_1 - cn_1}{2c^2} & -\frac{\gamma_1 v_2 - cn_2}{2c^2} & -\frac{\gamma_1 v_3 - cn_3}{2c^2} & \frac{\gamma_1}{2c^2} \\ \frac{v_n n_1 - v_1}{n_2} & \frac{1 - n_1^2}{n_2} & -n_1 & -\frac{n_1 n_3}{n_2} & 0 \\ \frac{v_3 - v_n n_3}{n_2} & \frac{n_1 n_3}{n_2} & n_3 & \frac{n_3^2 - 1}{n_2} & 0 \end{pmatrix}, \quad (2.72)$$

while the matrix Eq. 2.69 is nonsingular for $n_3 \neq 0$ and its inverse is given by

$$\mathbf{R}^{-1} = \begin{pmatrix} \frac{\gamma_1 e_k + cv_n}{2c^2} & -\frac{\gamma_1 v_1 + cn_1}{2c^2} & -\frac{\gamma_1 v_2 + cn_2}{2c^2} & -\frac{\gamma_1 v_3 + cn_3}{2c^2} & \frac{\gamma_1}{2c^2} \\ \frac{c^2 - \gamma_1 e_k}{c^2} & \frac{\gamma_1 v_1}{c^2} & \frac{\gamma_1 v_2}{c^2} & \frac{\gamma_1 v_3}{c^2} & -\frac{\gamma_1}{c^2} \\ \frac{\gamma_1 e_k - cv_n}{2c^2} & -\frac{\gamma_1 v_1 - cn_1}{2c^2} & -\frac{\gamma_1 v_2 - cn_2}{2c^2} & -\frac{\gamma_1 v_3 - cn_3}{2c^2} & \frac{\gamma_1}{2c^2} \\ \frac{v_1 - v_n n_1}{n_3} & \frac{n_1^2 - 1}{n_3} & \frac{n_1 n_2}{n_3} & n_1 & 0 \\ \frac{v_n n_2 - v_2}{n_3} & -\frac{n_1 n_2}{n_3} & \frac{1 - n_2^2}{n_3} & -n_2 & 0 \end{pmatrix}. \quad (2.73)$$

2.2.4. Rotational invariance of Euler equations

Given the vector state $\mathbf{w}(\mathbf{x}, t)$ that satisfies the Euler system

$$\frac{\partial \mathbf{w}}{\partial t} + \sum_{k=1}^d \frac{\partial \mathbf{f}_k^s(\mathbf{w})}{\partial x_k} = 0 \quad \text{in } \mathcal{M} = \Omega_t \times (0, T), \quad (2.74)$$

and, considering a change in the frame of reference such that for $\mathbf{x} \in \mathbb{R}^3$ we have

$$\bar{\mathbf{x}} = \mathbf{Q}(\mathbf{n}) \mathbf{x}, \quad (2.75)$$

where

$$\mathbf{n} = \begin{pmatrix} \cos \alpha \cos \beta & \sin \alpha \cos \beta & \sin \beta \end{pmatrix}^T \quad (2.76)$$

and $\mathbf{Q}_o(\mathbf{n}) \in \mathbb{R}^{3,3}$ is the rotation matrix

$$\mathbf{Q}_o(\mathbf{n}) = \begin{pmatrix} \cos \alpha \cos \beta & \sin \alpha \cos \beta & \sin \beta \\ -\sin \alpha & \cos \alpha & 0 \\ -\cos \alpha \sin \beta & -\sin \alpha \sin \beta & \cos \beta \end{pmatrix}, \quad (2.77)$$

then it is possible to prove that, under the new Cartesian frame $\bar{\mathbf{x}}$, the Euler system remains unchanged, i.e.,

$$\frac{\partial \mathbf{q}}{\partial t} + \sum_{k=1}^d \frac{\partial \mathbf{f}_k^s(\mathbf{q})}{\partial \bar{x}_k} = 0, \quad (2.78)$$

where $\mathbf{q}(\bar{\mathbf{x}}, t)$ is the vector state $\mathbf{w}(x, t)$ expressed under the new frame of reference

$$\mathbf{q}(\bar{\mathbf{x}}, t) = \mathbf{Q}(\mathbf{n}) \mathbf{w}(\mathbf{Q}_o^{-1}(\mathbf{n}) \bar{\mathbf{x}}, t), \quad (2.79)$$

with

$$\mathbf{Q}(\mathbf{n}) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \cos \alpha \cos \beta & \sin \alpha \cos \beta & \sin \beta & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 & 0 \\ 0 & -\cos \alpha \sin \beta & -\sin \alpha \sin \beta & \cos \beta & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.80)$$

Analogously, it is possible to write

$$\mathbf{f}_k^s(\mathbf{w}) \cdot \mathbf{n} = \mathbf{Q}^{-1}(\mathbf{n}) \mathbf{f}_k^s(\mathbf{Q}(\mathbf{n}) \mathbf{w}). \quad (2.81)$$

This property is called *rotational invariance of Euler equations*. The proof of this property can be found in [Tor99].

2.2.5. Riemann problem for the Euler equations

We consider the one dimensional system of Euler equations (Eq. 2.40) for an ideal gas and subject to the initial condition

$$\mathbf{w}(x, 0) = \begin{cases} \mathbf{w}_L, & x < 0, \\ \mathbf{w}_R, & x > 0 \end{cases}, \text{ for } \mathbf{w}_L, \mathbf{w}_R \in \mathbb{R}^3. \quad (2.82)$$

According to Lemma 2.2.11, for this case the Jacobian matrix of the vector flux has a set of eigenvalues given by

$$\boldsymbol{\lambda} = \begin{pmatrix} v_1 - c & v_1 & v_1 + c \end{pmatrix}, \quad (2.83)$$

as well as a complete set of eigenvectors defined by

$$\mathbf{R} = \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ v_1 - c & v_1 & v_1 + c \\ h - v_1 c & \frac{v_1^2}{2} & h + v_1 c \end{pmatrix}, \quad (2.84)$$

with h defined according with Eq. 2.63. Thus, it is possible to conclude that three characteristic fields are present in the solution of this problem. For non-linear systems these characteristic fields can be entropy discontinuity waves or rarefactions waves. At the same time, some of these fields can be genuine nonlinear or linearly degenerate, also named contact discontinuities.

Definition 2.2.12. If $\mathbf{r}_k(\mathbf{w})$ and $\lambda_k(\mathbf{w})$ are, correspondingly, the eigenvector and the associated eigenvalue that define k -th characteristic field for a non-linear hyperbolic system, then this characteristic field (and its corresponding eigenvector) is called *linearly degenerate* if

$$\nabla \lambda_k(\mathbf{w}) \cdot \mathbf{r}_k(\mathbf{w}) = 0, \text{ for all } \mathbf{w} \in \mathbb{R}^m,$$

otherwise, if

$$\nabla \lambda_k(\mathbf{w}) \cdot \mathbf{r}_k(\mathbf{w}) \neq 0, \text{ for all } \mathbf{w} \in \mathbb{R}^m,$$

it will be named *genuinely nonlinear* ([Lax57]). In other words, the characteristic field will be *genuinely nonlinear* if the corresponding eigenvalue increase or decrease monotonically as \mathbf{w} varies along the integral curves of \mathbf{r}_k , but if the eigenvalue stills constant along this curves, the field is *linearly degenerate*.

Using Eq. 2.83 and Eq. 2.84, the following products are obtained

$$\nabla\lambda_1 \cdot \mathbf{r}_1 = -\frac{\gamma + 1}{2} \frac{c}{\rho}, \quad (2.85)$$

$$\nabla\lambda_2 \cdot \mathbf{r}_2 = 0, \quad (2.86)$$

$$\nabla\lambda_3 \cdot \mathbf{r}_3 = \frac{\gamma + 1}{2} \frac{c}{\rho}, \quad (2.87)$$

and, from Def. 2.2.12, it is possible conclude that two of the characteristic fields in this system are genuinely nonlinear (first and third fields), while the second field is linearly degenerate (see Fig. 2.1). Specifically the characteristic fields can be classified as follows:

- *1st-characteristic field*: from result in Eq. 2.85 this field is genuinely non-linear, where the states \mathbf{w}_L and \mathbf{w}_L^* (see Fig. 2.1) are connected by a rarefaction.
- *2nd-characteristic field*: as previous, the second field is a contact discontinuity, where the value of $\lambda_2(\mathbf{w}) = v_1$ is constant along the integral curves belonging to $\mathbf{r}_2(\mathbf{w})$. From it follows that \mathbf{r}_2 , that is only in terms of v_1 , is constant along this curves, and then these integral curves are lines on the phase diagram. Thus, examining the *Rankine-Hugoniot* condition between the states \mathbf{w}_L^* and \mathbf{w}_R^* (see Fig. 2.1), the following expressions are found:

$$\begin{aligned} \rho_L^* v_{1,L}^* - \rho_R^* v_{1,R}^* &= s(\rho_L^* - \rho_R^*), \\ \left(\frac{1}{2}\rho_L^* v_{1,L}^{*2} + p_L^*\right) - \left(\frac{1}{2}\rho_R^* v_{1,R}^{*2} + p_R^*\right) &= s(\rho_L^* v_{1,L}^* - \rho_R^* v_{1,R}^*), \\ v_{1,L}^*(E_L^* + p_L^*) - v_{1,R}^*(E_R^* + p_R^*) &= s(E_L^* - E_R^*), \end{aligned} \quad (2.88)$$

where s is the propagation speed of the wave. This set of conditions are satisfied if $s = v_{1,L}^* = v_{1,R}^*$ and $p_L^* = p_R^*$. From this result it is possible to conclude that the contact discontinuity travel to the same speed that the fluid, and on the other hand, that both velocity and pressure are constant through the contact discontinuity and only there exist a jump in the density. Because this density discontinuity, other quantities, as momentum, temperature and entropy, also present a jump across the contact discontinuity. Precisely, the jump in entropy shows that this discontinuity is an entropy discontinuity wave. Further, as there is no a jump in the pressure, then there is no a net force over the fluid particles to cross from the left region to the right one, and thus the particles on this line will move parallel to the contact discontinuity.

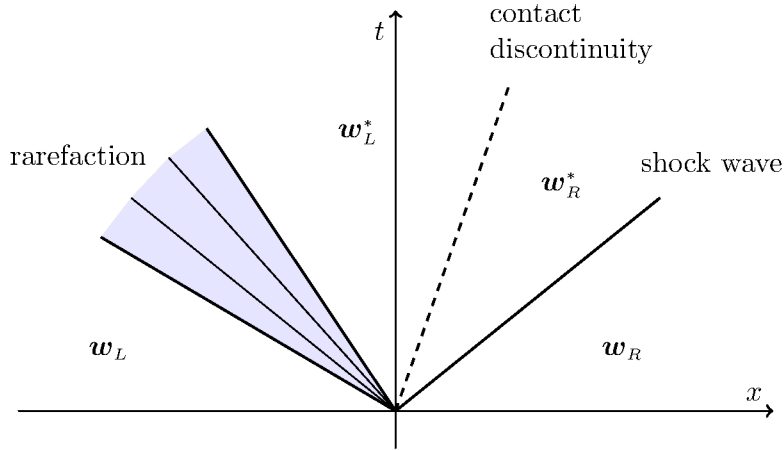
- *3rd-characteristic field*: from result in Eq. 2.87 this field is a non-linear wave. In this case, the states \mathbf{w}_R^* and \mathbf{w}_R (see Fig. 2.1) are connected by a shock wave.

A complete solution and analysis of this problem can be found in [FFS03, Wes01].

2.3. Boundary conditions

The Navier–Stokes system in Eq. 2.25 should be equipped with a set of boundary conditions, such that these correspond with the physical nature of the problem and, at the same time, with the character of the equations. For the viscous Navier–Stokes problem it is usual to classify the borders as either Dirichlet boundary (Γ^D) or Neumann boundary (Γ^N), in such a way that $\partial\Omega = \Gamma^D \cup \Gamma^N$ and $\Gamma^D \cap \Gamma^N = \emptyset$. Particularly, the boundary will be called to be of Dirichlet-type if at least one of the conservative variables are prescribed on this border, otherwise the border will be Neumann-type.

Figure 2.1.: Characteristic fields in the solution of the Riemann problem for Euler equations.



From a physical point of view, in a flow problem can be distinguished three different boundaries: inlet flow (Γ_{IN}), outlet flow (Γ_{O}) and walls (Γ_{W}). Further, the inflow/outflow borders can be subsonic or supersonic, while the wall boundary condition can be adiabatic or isothermal. From the theory of characteristics (see Sec. 3.6) it is possible to say that only on the supersonic outlet boundary should be not prescribed any boundary value, and thus

$$\Gamma^D = \Gamma_{\text{IN}}^{\text{sup}} \cup \Gamma_{\text{IN}}^{\text{sub}} \cup \Gamma_{\text{O}}^{\text{sub}} \cup \Gamma_{\text{W}}^{\text{iso}} \cup \Gamma_{\text{W}}^{\text{adia}}$$

and

$$\Gamma^N = \Gamma_{\text{O}}^{\text{sup}},$$

where $\Gamma_{\text{W}}^{\text{iso}}$ corresponds to the isothermal wall boundary and $\Gamma_{\text{W}}^{\text{adia}}$ to the adiabatic wall boundary.

Thus, the complete Navier–Stokes problem is formulated through the system of equations Eq. 2.25

$$\frac{\partial \mathbf{w}}{\partial t} + \text{div}(\mathbb{F}^s(\mathbf{w})) = \frac{1}{\text{Re}} \text{div}(\mathbb{F}^v(\mathbf{w}, \nabla \mathbf{w})) + \mathbf{F}(\mathbf{w}) \quad \text{in } \mathcal{M} = \Omega_t \times (0, T), \quad (2.89)$$

endowed with the boundary conditions

$$\mathcal{B}(\mathbf{w}) = \mathcal{B}(\mathbf{w}^D) \quad \text{on } \Gamma_{\text{IN}}^{\text{sup}} \cup \Gamma_{\text{IN}}^{\text{sub}} \cup \Gamma_{\text{O}}^{\text{sub}}, \quad (2.90)$$

$$\mathbb{F}^v(\mathbf{w}, \nabla \mathbf{w}) \cdot \mathbf{n} = 0 \quad \text{on } \Gamma_{\text{O}}^{\text{sup}}, \quad (2.91)$$

$$\left. \begin{array}{l} \mathbf{v} = 0 \\ \theta = \theta_{\text{W}} \end{array} \right\} \text{on } \Gamma_{\text{W}}^{\text{iso}} \quad (2.92)$$

and

$$\left. \begin{array}{l} \mathbf{v} = 0 \\ \nabla \theta \cdot \mathbf{n} = 0 \end{array} \right\} \text{on } \Gamma_{\text{W}}^{\text{adia}}, \quad (2.93)$$

and subject to the initial condition

$$\mathbf{w}(\mathbf{x}, 0) = \mathbf{w}^0(\mathbf{x}), \quad \text{for all } \mathbf{x} \in \Omega. \quad (2.94)$$

Here, \mathbf{w}^D is the boundary value vector, θ_{W} is the temperature on the isothermal wall and $\mathcal{B}(\mathbf{w})$ denotes a boundary operator that enforces properly the boundary values. A complete discussion about numeric imposition of this boundary condition is developed in Sec. 3.6 for inviscid flows, and in Sec. 3.7.4 for viscous problems.

2.4. Equations of state for carbon dioxide

Some of the most used EOS for carbon dioxide are:

2.4.1. Span and Wagner equation of state

The Span and Wagner EOS ([SW96]) is considered the most precise model to describe the behavior of CO₂ ([Tür99, LDQA15]), even in the supercritical region. Unfortunately its complexity makes it extremely expensive to use into computational models. In this model the equation of state is derived from the fundamental equation [PN98]

$$p(\rho, \theta) = \rho^2 \frac{\partial A}{\partial \rho}(\rho, \theta), \quad (2.95)$$

where $A(\rho, \theta) \in \mathbb{R}$ is the *Helmholtz free energy*, which is defined by

$$A(\rho, \theta) = \int_0^\rho \frac{p(r, \theta) - R\theta r}{r^2} dr + R\theta \ln(\rho) + A^\circ(\theta), \quad (2.96)$$

with $A^\circ(\theta)$ being the Helmholtz energy for the ideal gas state and R the *specific gas constant*.

Particularly, in [SW96] the *dimensionless Helmholtz energy* $\phi(\delta, \zeta)$ is expressed in terms of an ideal gas part $\phi^\circ(\delta, \zeta)$ and a residual energy $\phi^r(\delta, \zeta)$

$$\phi(\delta, \zeta) = \frac{A(\rho, \theta)}{R\theta} = \phi^\circ(\delta, \zeta) + \phi^r(\delta, \zeta),$$

where

$$\delta = \frac{\rho}{\rho_c}, \quad \zeta = \frac{\theta_c}{\theta}, \quad (2.97)$$

and ρ_c and θ_c are the *critical density* and the *critical temperature*, respectively. The expressions for $\phi^\circ(\delta, \zeta)$ and $\phi^r(\delta, \zeta)$, and the coefficients to compute them, as well as the accuracy of this model in the different regions of the pressure-temperature diagram, can be found in [SW96].

2.4.2. IUPAC equation of state

The International Union of Pure and Applied Chemistry (*IUPAC*) standardized the CO₂ properties and defined as model for the EOS

$$\frac{p(\rho, \theta)}{\rho R \theta} = 1 + \delta \sum_{i=0}^9 \sum_{j=0}^{J_i} a_{ij} (\zeta - 1)^j (\delta - 1)^i, \quad (2.98)$$

where δ and ζ are defined in Eq. 2.97, and the coefficients a_{ij} and the numbers J_i can be found in [AAR76]. The *IUPAC-EOS* is able to determine accurately pressure values for CO₂ in liquid and gas phase, as well as in the supercritical region, nevertheless it is not able to work under supercooled conditions ([AAR76, YF15]). The *IUPAC-EOS* and the Span and Wagner model are included in the `PROPATH` library ([PRO08]) for the computing of the thermophysical properties of CO₂.

2.4.3. Extended and generalized Bender equation of state

In ([PM89]) a generalized version of the Bender EOS ([Ben71]) is presented. This model, named *extended generalized Bender EOS (egB-EOS)*, is described by the equation

$$\begin{aligned} \frac{p(\rho, \theta)}{\rho R \theta} = & 1 + \delta (\epsilon_1 - \epsilon_2 \zeta - \epsilon_3 \zeta^2 - \epsilon_4 \zeta^3 - \epsilon_5 \zeta^4) + \delta^2 (\epsilon_6 + \epsilon_7 \zeta + \epsilon_8 \zeta^2) \\ & + \delta^3 (\epsilon_9 + \epsilon_{10} \zeta) + \delta^4 (\epsilon_{11} + \epsilon_{12} \zeta) + \delta^5 (\epsilon_{13} \zeta) \\ & + \delta^2 [\epsilon_{14} \zeta^3 + \epsilon_{15} \zeta^4 + \epsilon_{16} \zeta^5 + \delta^2 (\epsilon_{17} \zeta^3 + \epsilon_{18} \zeta^4 + \epsilon_{19} \zeta^5)] e^{-\delta^2}, \end{aligned} \quad (2.99)$$

where δ and ζ were defined in Eq. 2.97. One of the most relevant characteristics of this model is that only four parameters are required to characterize the substance: critical density (ρ_c), critical temperature (θ_c), *acentric factor* (ω) and *polar factor* (χ). In fact, all the ϵ -coefficients into Eq. 2.99 are computed using 95 general parameters (see [PM89]) and the four specific parameters mentioned. The *egB-EOS* is adequate to describe the behavior of the CO₂ in a wide range of states in liquid phase as well as in gas phase. This range is bounded by $0.3 \leq 1/\zeta \leq 16$, $\delta \leq 3.3$ and $p/p_c \leq 3.3$, where p_c is the value of the pressure in the critical point. Therefore this model can be used for supercritical and subcritical states, and also for supercooled conditions. According with several researches ([Tür99, LDQA15, KTSG96]), the *egB-EOS* is able to derive the thermodynamical properties of the CO₂ with sufficient accuracy, even under supercritical conditions. In Sec. B the *egB-EOS*, with the ϵ -coefficients computed for CO₂, is presented.

2.4.4. Peng–Robinson equation of state

The Peng–Robinson is a well known EOS that belong to the family of cubic equations of state. This model defines the pressure as ([PR76])

$$p(\rho, \theta) = \frac{R_m \theta}{V_m - b} - \frac{a\alpha}{V_m^2 + 2bV_m - b^2}, \quad (2.100)$$

where $R_m = RM$ is the *universal gas constant*, $V_m = M/\rho$ is the *molar volume* and the coefficients a , b and α are defined as

$$\begin{aligned} a &= 0.457235 \frac{R_m^2 \theta_c^2}{p_c}, \\ b &= 0.077796 \frac{R_m \theta_c}{p_c} \end{aligned}$$

and

$$\alpha = \left(1 + (0.37464 + 1.54226\omega - 0.26992\omega^2) \left(1 - \frac{1}{\sqrt{\zeta}} \right) \right)^2,$$

with M being the *molar mass* and ω is the *acentric factor*. Although this model is able to describe densities in the liquid-gas phases, even into the supercritical region, it is not able to describe accurately some caloric properties ([DCT98, Tür99, LDQA15]).

2.5. Thermophysical properties of CO₂

The equation of state in Eq. 2.18, either in form of an ideal gas equation or using some real gas equation, is mainly used as a constitutive relation to close the Navier–Stokes system. Nevertheless, in this set of conservation equations some additional *thermodynamic properties*, which are derived from the EOS, are involved. In fact, properties as the specific heat (C_v or C_p), the specific heat ratio (γ), the temperature (θ) and the speed of sound (c) are

also required for the modeling of these problems. These properties, called *caloric properties*, are deduced from thermodynamic relations and using the EOS. A complete analysis of these thermodynamic properties, as well as the mathematical relations between them, can be found in advanced thermodynamic books, e.g. in [HK07, JD96].

When the ideal gas EOS is employed, the values of C_p , C_v and γ can be considered constant, whereas the other properties can be easily obtained from Eq. 2.21. Thus, Eq. 2.23 for temperature and Eq. 2.61 for the speed of sound are derived. Nevertheless, in case of real gases, which have substantially more complex EOS, the caloric properties should be derived for each particular EOS used. Next, the deduction of the thermodynamic properties of the CO_2 , calculated from the *egB-EOS*, is presented.

2.5.1. Physical constants for CO_2

Table 2.1 include the most relevant physical constants for CO_2 .

Table 2.1.: Physical constants for CO_2 (taken from [SW96]).

M :	Molar mass	$M = 44.0098 \pm 0.0016$ g/mol
R :	Specific gas constant	$R = 0.1889241 \pm 0.0000116$ kJ/kgK
θ_c :	Critical temperature	$\theta_c = 304.1282 \pm 0.015$ K
p_c :	Critical pressure	$p_c = 7.3773 \pm 0.0030$ MPa
ρ_c :	Critical density	$\rho_c = 467.6 \pm 0.6$ kg/m ³
θ_t :	Triple point temperature	$\theta_t = 216.592 \pm 0.002$ K
p_t :	Triple point pressure	$p_t = 0.51795 \pm 0.00010$ MPa
θ_0 :	Reference temperature	$\theta_0 = 298.15$ K
p_0 :	Reference pressure	$p_0 = 0.101325$ MPa
e_0 :	Reference specific internal energy	$e_0 = 0$ kJ/kg

2.5.2. Isochoric specific heat

Considering an EOS with the form $p = p(\rho, \theta)$, the isochoric specific heat (or *heat capacity at constant volume*) C_v , can be computed using the expression ([PN98])

$$C_v(\rho, \theta) = -\theta \left(\frac{\partial^2 A}{\partial \theta^2}(\rho, \theta) \right). \quad (2.101)$$

Introducing Eq. 2.96 into Eq. 2.101 the following expression is found

$$C_v(\rho, \theta) = -\theta \int_0^\rho \frac{\partial^2 p}{\partial \theta^2}(r, \theta) \frac{dr}{r^2} + C_v^\circ(\theta). \quad (2.102)$$

Here, the term

$$C_v^\circ(\theta) = -\theta \frac{d^2 A^\circ}{d\theta^2}(\theta)$$

is the isochoric specific heat for the ideal gas state. According to [SW96], this term can be approximated by the relation

$$\frac{C_v^\circ(\theta)}{R} = 1 + a_1 + \sum_{i=2}^6 a_i (\Theta_i \zeta)^2 \frac{\exp(\Theta_i \zeta)}{(\exp(\Theta_i \zeta) - 1)^2}, \quad (2.103)$$

along with the coefficients in Table 2.2. Here, as previous, $\zeta = \theta_c/\theta$. The behavior of C_v in the $p - \theta$ diagram, computed from Eq. 2.102 and using the *egB-EOS*, is shown in Fig. 2.2. On the other hand, a comparison between the C_v values found using *egB-EoOS* and *Span & Wagner-EOS* is presented in Fig. 2.3.

Table 2.2.: List of numeric coefficients to compute C_v° ([SW96]).

i	1	2	3	4	5	6
a_i	2.50000000	1.99427042	0.62105248	0.41195293	1.04028922	0.08327678
Θ_i		3.15163	6.11190	6.77708	11.32384	27.08792

Figure 2.2.: Isochoric specific heat for CO₂ calculated using *egB-EOS*.

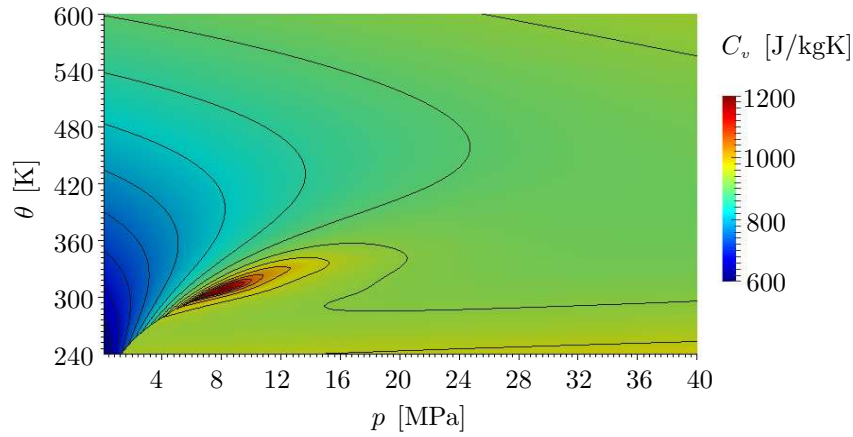
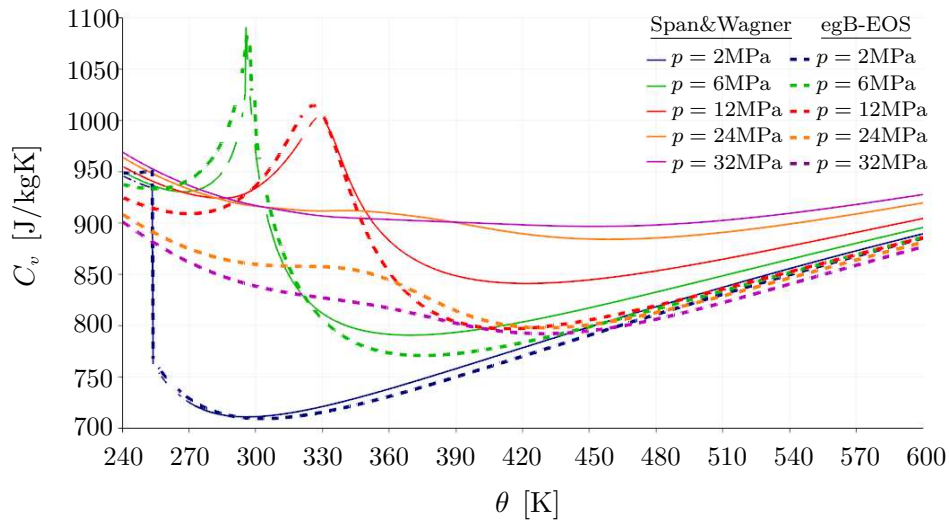


Figure 2.3.: Differences in the values of the isochoric specific heat for CO₂ using two different EOS (*Span&Wagner-EOS* and *egB-EOS*).



2.5.3. Isobaric specific heat

In order to compute the isobaric specific heat, also called *heat capacity at constant pressure*, the expression ([PN98])

$$C_p(\rho, \theta) = C_v(\rho, \theta) + \frac{\theta \left(\frac{\partial p}{\partial \theta}(\rho, \theta) \right)^2}{\rho^2 \left(\frac{\partial p}{\partial \rho}(\rho, \theta) \right)} \quad (2.104)$$

is used. In Fig. 2.4 is shown the behavior of C_p , calculated using Eq. 2.104 and the *egB-EOS*, for different values of pressure and temperature. The differences in the C_p values for two different EOS (*egB-EOS* and *Span & Wagner-EOS*) are presented in Fig. 2.5.

Figure 2.4.: Isobaric heat capacity for CO₂ calculated using *egB-EOS*.

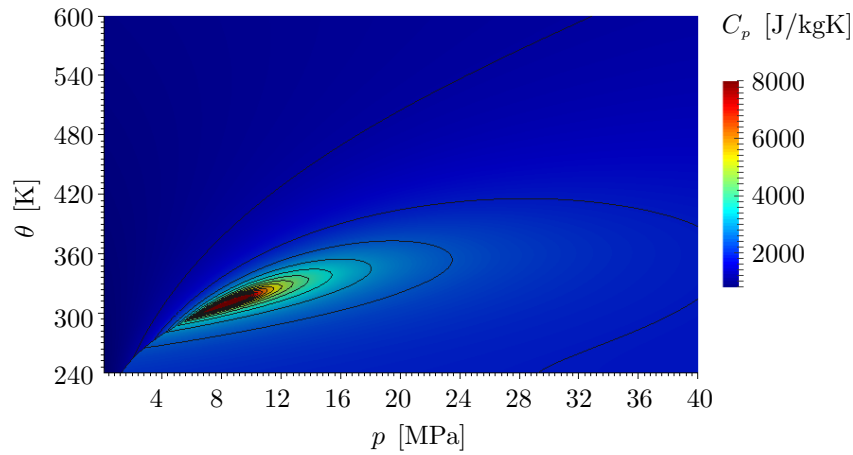
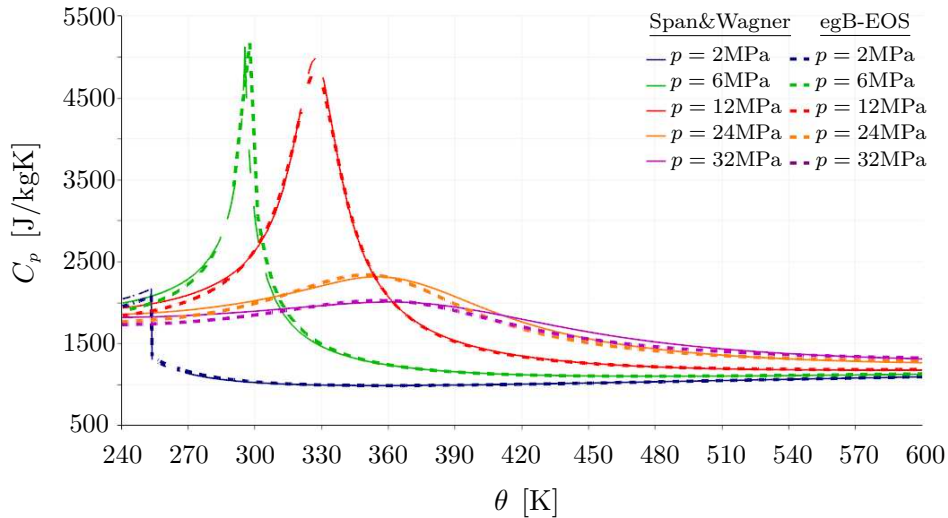


Figure 2.5.: Differences in the values of the isobaric specific heat for CO₂ using two different EOS (*Span&Wagner-EOS* and *egB-EOS*).



2.5.4. Heat capacity ratio

From Eq. 2.104, the heat capacity ratio can be calculated using the expression

$$\gamma(\rho, \theta) = 1 + \frac{\theta \left(\frac{\partial p}{\partial \theta}(\rho, \theta) \right)^2}{\rho^2 C_v(\rho, \theta) \left(\frac{\partial p}{\partial \rho}(\rho, \theta) \right)}. \quad (2.105)$$

Using the last expression it is possible to plot the behavior of γ in the $p - \theta$ diagram, as in the Fig. 2.6. Additionally, differences in γ values, computed using both *Span & Wagner-EOS* and *egB-EOS*, are showed in Fig. 2.7.

Figure 2.6.: Heat capacity ratio for CO₂ calculated using *egB-EOS*.

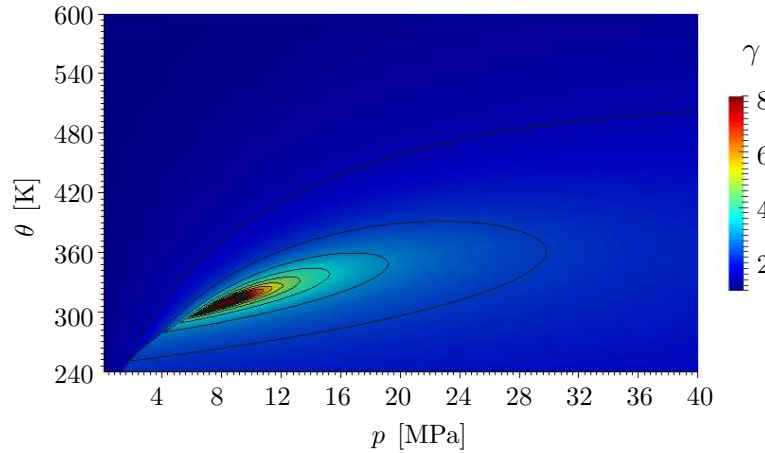
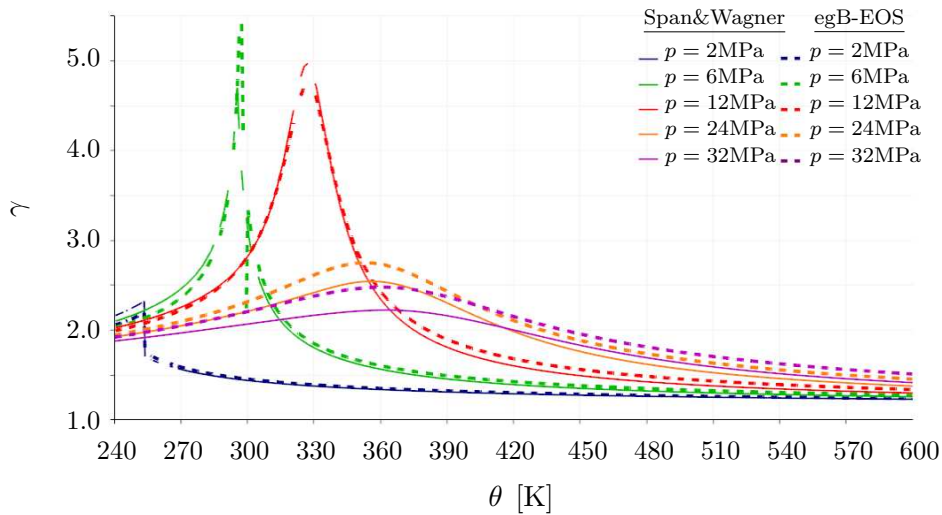


Figure 2.7.: Differences in the values of the specific heat ratio for CO₂ using two different EOS (*Span&Wagner-EOS* and *egB-EOS*).



2.5.5. Speed of sound

One of the most important properties for compressible flows is the speed of sound, here this quantity can be computed from the expression ([PN98])

$$c(\rho, \theta) = \sqrt{\gamma(\rho, \theta) \frac{\partial p}{\partial \rho}(\rho, \theta)}. \quad (2.106)$$

The Fig. 2.8 shows the behavior of the speed of sound c , computed using Eq. 2.106 and the *egB-EOS*, in terms of pressure and temperature. On the other hand, in Fig. 2.9 are shown the differences in the values of the speed of sound computed using both *egB-EOS* and *Span & Wagner-EOS*.

Figure 2.8.: Speed of sound for CO_2 calculated using *egB-EOS*.

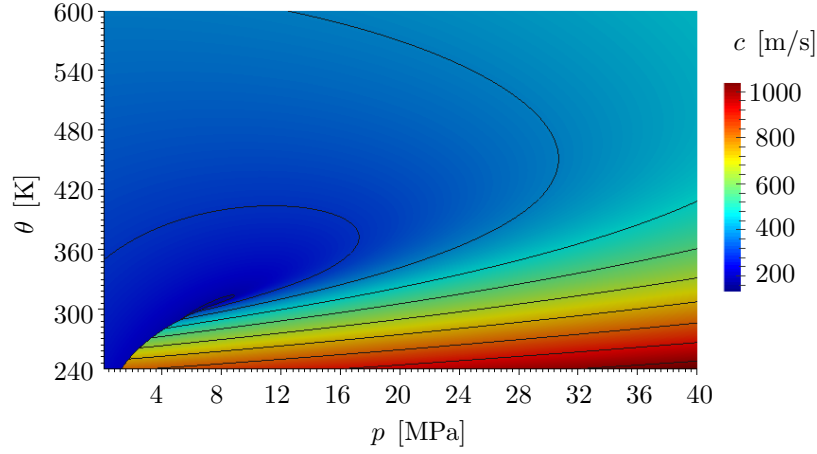
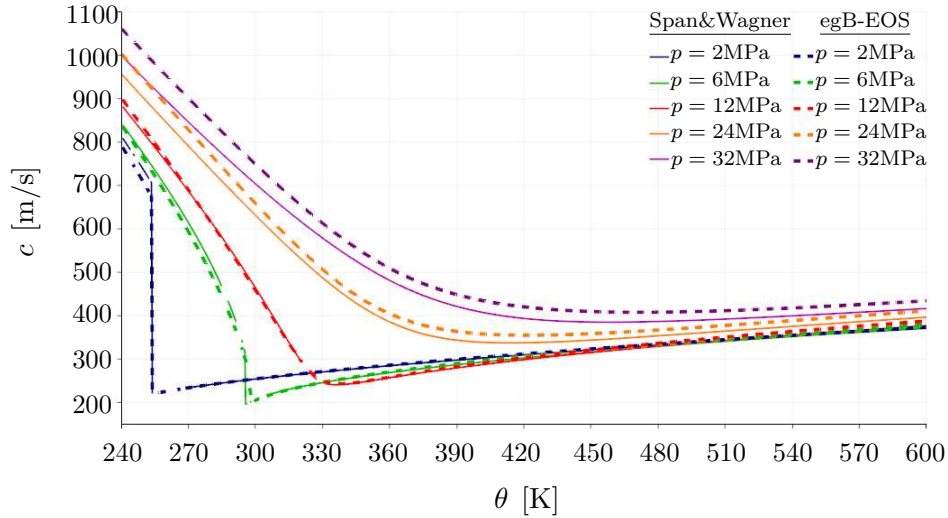


Figure 2.9.: Differences in the values of speed of sound in CO_2 using two different EOS (*Span&Wagner-EOS* and *egB-EOS*).



2.5.6. Specific internal energy

The specific internal energy is computed using the equation ([HK07])

$$e(\rho, \theta) = e_0 - \theta \int_{\rho_0}^{\rho} \frac{\partial}{\partial \theta} \left(\frac{p}{\theta} \right) \frac{dr}{r^2} + \int_{\theta_0}^{\theta} C_v^o(T) dT. \quad (2.107)$$

Here, the reference values for temperature θ_0 , pressure p_0 and specific internal energy e_0 , are listed in Table 2.1, while the reference density is calculated using the relation for ideal gas

$$\rho_0 = \frac{p_0}{R\theta_0}. \quad (2.108)$$

The analytic expression for e obtained from Eq. 2.107, as well as its partial derivatives $\partial e/\partial\rho$ and $\partial e/\partial\theta$, are included in Sec. A .

The corresponding graphic for the specific internal energy in the $\theta - p$ diagram, and using the *egB-EOS*, is plotted in Fig. 2.10. The differences introduced by using two different EOS are shown in Fig. 2.11 for *egB-EOS* and *Span & Wagner-EOS*.

Figure 2.10.: Specific internal energy for CO₂ calculated using *egB-EOS*.

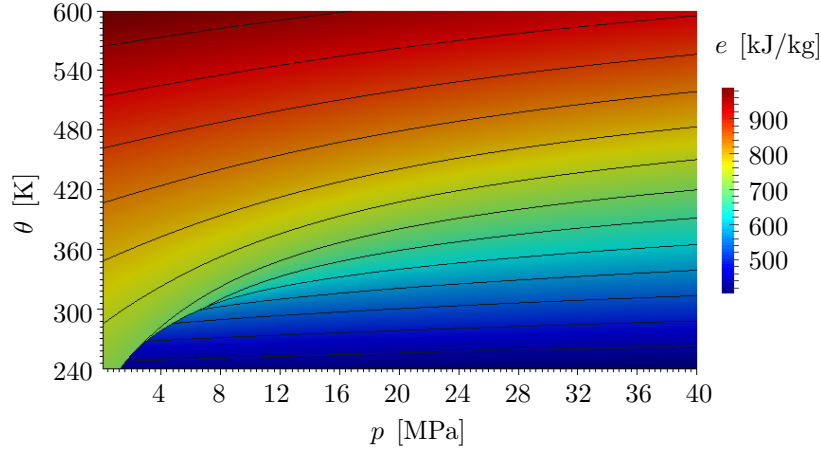
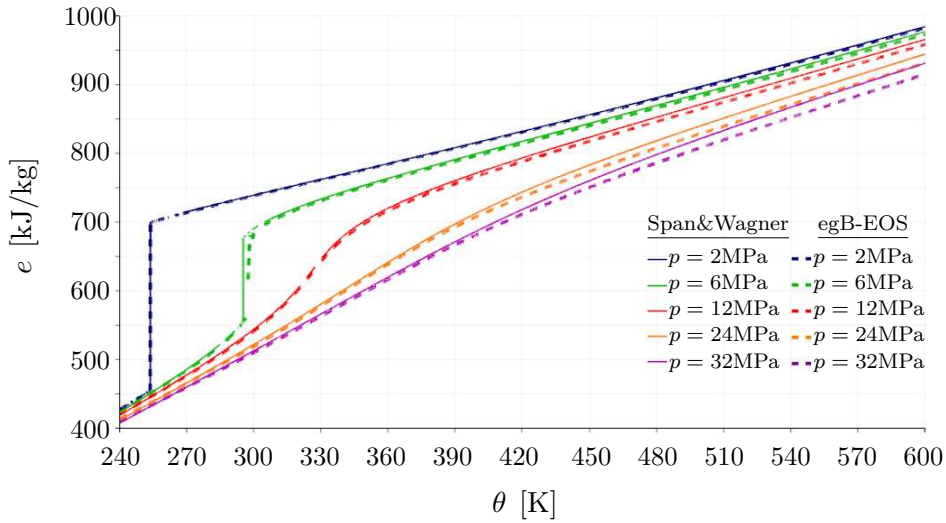


Figure 2.11.: Differences in the values of the specific internal energy for CO₂ using two different EOS (*Span&Wagner-EOS* and *egB-EOS*).



2.5.7. Dynamic viscosity

The dynamic viscosity is calculated using the correlation presented by [AS72]

$$\mu(\rho, \theta) = \mu^\circ(\theta) \exp\left(\sum_{i=1}^4 \sum_{j=0}^1 a_{ij} \delta^i \zeta^j\right), \quad (2.109)$$

where the coefficients a_{ij} are listed in Table 2.3,

$$\mu^\circ(\theta) = \frac{1}{\sqrt{\zeta}} (27.2246461 - 16.6346068\zeta + 4.66920556\zeta^2). \quad (2.110)$$

and, as in Eq. 2.97, $\zeta = \theta_c/\theta$ and $\delta = \rho/\rho_c$. Here, all values are given in *SI* units, i.e., θ is given in K, ρ in kg/m^3 and μ in $\mu\text{Pa}\cdot\text{s}$. Eq. 2.109 is used to show, in Fig. 2.12 and

Table 2.3.: List of numeric coefficients to compute μ° ([AS72]).

a_{10}	0.248566120	a_{30}	0.363854523
a_{11}	0.004894942	a_{31}	-0.774229021
a_{20}	-0.373300660	a_{40}	-0.0639070755
a_{21}	1.22753488	a_{41}	0.142507049

Fig. 2.13, the behavior of μ on a wide range of pressure and temperature.

Figure 2.12.: Behavior of the dynamic viscosity of CO_2 on the $\theta - p$ domain.

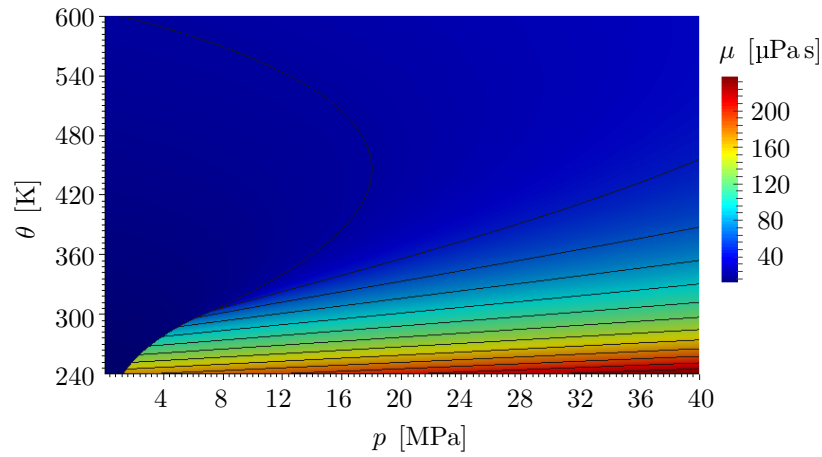
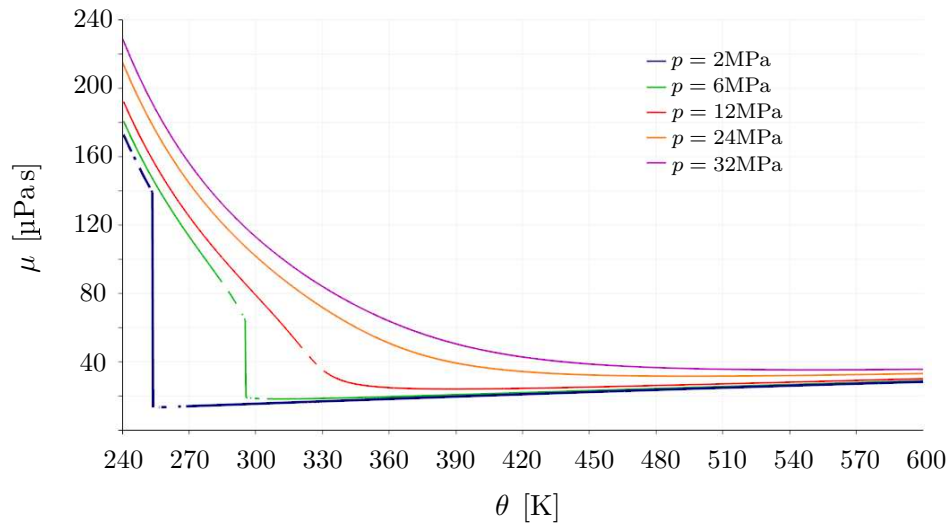


Figure 2.13.: Dynamic viscosity for CO_2 .



2.5.8. Thermal conductivity

According to [VWO⁺90], the thermal conductivity can be expressed as a zero-density basic approximation $k^\circ(\theta)$, plus a term to incorporate the effect of the density $\Delta k(\rho)$,

Table 2.4.: List of numeric coefficients to compute μ° ([AS72]).

i	a_i	b_i	c_i	i	a_i	b_i	c_i
0	0.4226159	—	—	4	0.0	7.981590	6.594919E-11
1	0.6280115	0.02387869	2.447164E-2	5	0.0	-1.940558	—
2	-0.5387661	4.350794	8.705605E-5	6	-0.4362677	—	—
3	0.6735941	-10.33404	-6.547950E-8	7	0.2255388	—	—

and an additional correction term to enhance the approximation near to the critical point $\Delta_c k(\rho, \theta)$, i.e.

$$k(\rho, \theta) = k^\circ(\theta) + \Delta k(\rho) + \Delta_c k(\rho, \theta). \quad (2.111)$$

Here, the basic approximation is given by the expression

$$k^\circ(\theta) = 0.475598 \frac{\sqrt{\theta}(1 + 0.4k^*)}{\mathcal{G}(\theta^*)}, \quad (2.112)$$

with $\theta^* = \theta/251.196$,

$$\mathcal{G}(\theta^*) = \sum_{i=0}^7 \frac{a_i}{\theta^{*i}} \quad (2.113)$$

and

$$k^* = 1.0 + \exp\left(-\frac{183.5}{\theta}\right) \sum_{i=1}^5 b_i \left(\frac{\theta}{100}\right)^{2-i}. \quad (2.114)$$

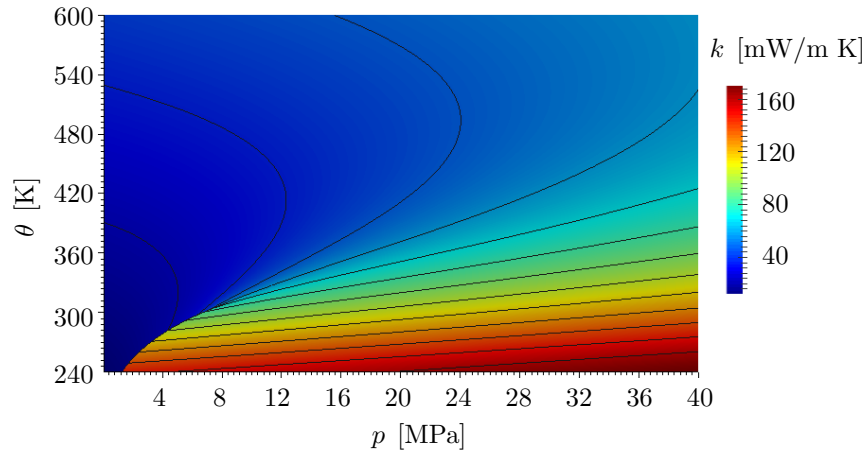
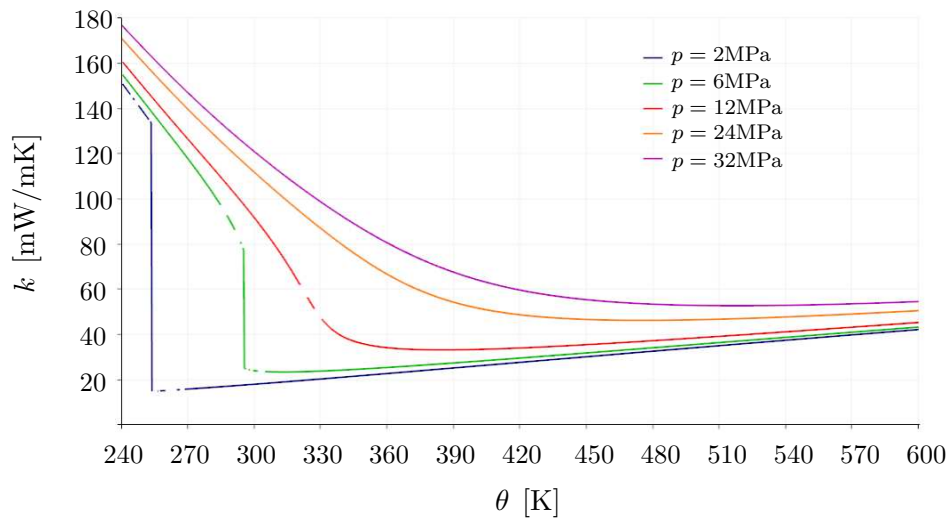
On the other hand, the first correction term is estimated through the equation

$$\Delta k(\rho) = \sum_{i=1}^4 c_i \rho^i. \quad (2.115)$$

For Eq. 2.112–Eq. 2.115 all quantities are expressed in *SI*-units, i.e. θ in K, ρ in kg/m³ and k in mW/mK. Likewise, for these equations the list of coefficients a_i , b_i and c_i is given in Table 2.4.

In [VWO⁺90] a second correction term $\Delta_c k$ is included in order to improve the quality of the correlation near to the critical point, nevertheless this term is of high complexity, and its use will reduce efficiency in the computational sense, for this reason it will not take in account to compute the thermal conductivity in this work.

From Eq. 2.111–Eq. 2.115, the behavior of k can be plotted in terms of temperature and pressure, as in Fig. 2.14 and Fig. 2.15.

Figure 2.14.: Behavior of the thermal conductivity for CO₂ in the $\theta - p$ domain.Figure 2.15.: Thermal conductivity for CO₂.

3. Spatial semidiscretization for Navier–Stokes equations

3.1. Meshing

Let Ω_h be a polygonal (for $d = 2$) or polyhedral (for $d = 3$) approximation of Ω , and let $\mathcal{T}_h = \{K_i : i \in I\}$, with $I = \{1, 2, \dots, N_c\}$, be the partition of the closure $\overline{\Omega}_h$ using a set K of N_c closed convex polygons (for $d = 2$) or polyhedra (for $d = 3$).

In this triangulation, two elements $K_i, K_j \in \mathcal{T}_h$ will be named *neighbors* if these share a common edge (for $d = 2$) or plane (for $d = 3$) Γ . Further, we will denote with Γ_{ij} the corresponding edge/face belongs to K_i , and with Γ_{ji} the equivalent overlap edge/face belongs to K_j (see Fig. 3.1). How no requirements about conforming mesh is demanded, the common edge/face between two neighbor elements will not necessarily correspond to the complete border of both elements. Additionally, $s(i)$ be the index set for the neighboring elements of K_i , i.e.,

$$s(i) = \{j \in I : K_j \text{ is a neighbor of } K_i\}.$$

Likewise, $\partial\Omega_h$ will denote the discrete approximation of $\partial\Omega$ composed by a finite set of

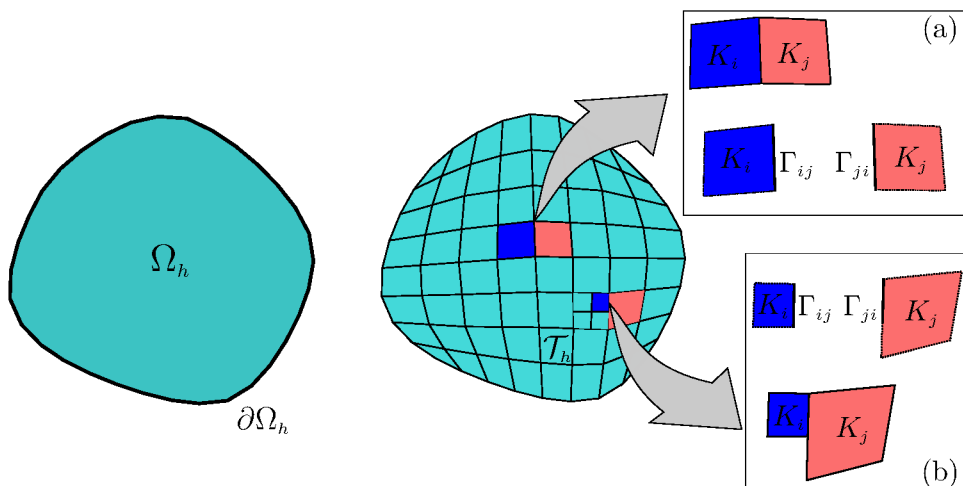


Figure 3.1.: Spatial discretization.

faces S_j adjacent to $\partial\Omega$. Thus, $\kappa(i)$ will denote an index set such that

$$\kappa(i) = \{j \in I_b : S_j \text{ is a face of } K_i\},$$

with $I_b = \{-1, -2, \dots\}$. Therefore, K_i will be named internal element if $\kappa(i) = \emptyset$. In similar fashion $\kappa^D(i) \subset \kappa(i)$ will denote the set of indices of faces adjacent to the Dirichlet border (Γ^D), i.e., where at least one conservative variable is prescribed, while $\kappa^N(i)$ will represent the set of faces adjacent to the Neumann boundary (Γ^N), thus $\kappa(i) = \kappa^D(i) \cup \kappa^N(i)$. Further, $S(i)$ will be the complete index set for the element K_i , i.e., $S(i) = s(i) \cup \kappa(i)$.

3.2. Weak formulation of inviscid problems using discontinuous Galerkin finite element method (DGFEM)

3.2.1. Broken Sobolev spaces

Defining a broken Sobolev space over the triangulation \mathcal{T}_h , such that

$$H^r(\Omega, \mathcal{T}_h) = \{v \in L^2(\Omega) : v|_K \in H^r(K) \text{ for all } K \in \mathcal{T}_h\}, \quad (3.1)$$

with $r \geq 0$. For $v \in H^1(\Omega, \mathcal{T}_h)$ the following notation is introduced:

- $v|_{\Gamma_{ij}}$ is the trace of $v|_{K_i}$ on the edge Γ_{ij} .
- $v|_{\Gamma_{ji}}$ is the trace of $v|_{K_j}$ on the edge Γ_{ji} . Note that in general $v|_{\Gamma_{ij}} \neq v|_{\Gamma_{ji}}$.
- The jump of v in the edge Γ_{ji} is defined by

$$[[v]]_{\Gamma_{ij}} = -[[v]]_{\Gamma_{ji}} = v|_{\Gamma_{ij}} - v|_{\Gamma_{ji}}. \quad (3.2)$$

- The average of v on Γ_{ji} is defined by

$$\{v\}_{\Gamma_{ij}} = \{v\}_{\Gamma_{ji}} = \frac{1}{2} (v|_{\Gamma_{ij}} + v|_{\Gamma_{ji}}). \quad (3.3)$$

3.2.2. Weak formulation

Taking the set of Euler equations Eq. 2.40, multiplying by the vector-valued function $\varphi \in [H^2(\Omega, \mathcal{T}_h)]^m$ and integrating over Ω_h one finds

$$\frac{d}{dt} \sum_{i \in I} \int_{K_i} \mathbf{w} \cdot \varphi \, d\mathbf{x} + \sum_{i \in I} \int_{K_i} \operatorname{div}(\mathbb{F}^s(\mathbf{w})) \cdot \varphi \, d\mathbf{x} = 0. \quad (3.4)$$

Now, applying Green's theorem, it is possible to write

$$\begin{aligned} \frac{d}{dt} \sum_{i \in I} \int_{K_i} \mathbf{w} \cdot \varphi \, d\mathbf{x} - \sum_{i \in I} \int_{K_i} \sum_{k=1}^d \mathbb{F}_k^s(\mathbf{w}) \cdot \frac{\partial \varphi}{\partial x_k} \, d\mathbf{x} \\ + \sum_{i \in I} \sum_{j \in S(i)} \int_{\Gamma_{ij}} \sum_{k=1}^d \mathbb{F}_k^s(\mathbf{w}) n_k \cdot \varphi \, dS = 0, \end{aligned} \quad (3.5)$$

or defining the forms

$$(\mathbf{w}, \varphi)_h = \sum_{i \in I} \int_{K_i} \mathbf{w} \cdot \varphi \, d\mathbf{x} \quad (3.6)$$

and

$$\begin{aligned} \mathbf{b}_h(\mathbf{w}, \varphi) = & - \sum_{i \in I} \int_{K_i} \sum_{k=1}^d \mathbb{F}_k^s(\mathbf{w}) \cdot \frac{\partial \varphi}{\partial x_k} dx \\ & + \sum_{i \in I} \sum_{j \in S(i)} \int_{\Gamma_{ij}} \sum_{k=1}^d \mathbb{F}_k^s(\mathbf{w}) n_k \cdot \varphi dS, \end{aligned} \quad (3.7)$$

the expression Eq. 3.4 is

$$\frac{d}{dt} (\mathbf{w}, \varphi)_h + \mathbf{b}_h(\mathbf{w}, \varphi) = 0. \quad (3.8)$$

3.2.3. Discrete approximation

Let $\mathbf{w} : \mathcal{M} \rightarrow \mathbb{R}^m$ be the exact solution of Eq. 3.8, and we will denote the function $\mathbf{w}(t) : \Omega \rightarrow \mathbb{R}^m$ such that $\mathbf{w}(t) = \mathbf{w}(\cdot, t)$. Then, approximating \mathbf{w} locally by discrete functions $\mathbf{w}_h \in C^1([0, T]; \mathbf{S}_h)$, where

$$\begin{aligned} \mathbf{S}_h &= [S_h]^m, \\ S_h &= \{v : \mathcal{T}_h \mapsto \mathbb{R} : v|_K \in \mathbb{P}^q \text{ for all } K \in \mathcal{T}_h\} \end{aligned} \quad (3.9)$$

and \mathbb{P}^q is the space of polynomials over K with degree lower than q , thus, the discrete version of Eq. 3.8 is

$$\frac{d}{dt} (\mathbf{w}_h(t), \varphi)_h + \mathbf{b}_h(\mathbf{w}_h(t), \varphi) = 0. \quad (3.10)$$

As usual, the normal flux component $\sum_{k=1}^d \mathbf{f}_k^s(\mathbf{w}_h) n_k$ in Eq. 3.7 will be approximated using a numerical flux \mathbf{H} . This term corresponds to a physical model that represents the interaction between neighboring cells. Hence, this numerical term should incorporate not only the values of the conservative vector \mathbf{w}_h over the interface of K_i , but also the values of \mathbf{w}_h over the neighbor K_j . Thus, the corresponding term in Eq. 3.7 is written now as

$$\int_{\Gamma_{ij}} \sum_{j=1}^d \mathbf{f}_j^s(\mathbf{w}_h) \cdot \varphi n_j dS \approx \int_{\Gamma_{ij}} \mathbf{H}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}) \cdot \varphi dS, \quad (3.11)$$

where $\mathbf{w}^+ = \mathbf{w}_h|_{\Gamma_{ij}}$, $\mathbf{w}^- = \mathbf{w}_h|_{\Gamma_{ji}}$ and \mathbf{n} is the unit vector normal to the interface Γ_{ij} . Thus, the operator $\mathbf{b}_h(\mathbf{w}_h(t), \varphi)$ now can be written as

$$\begin{aligned} \mathbf{b}_h(\mathbf{w}_h(t), \varphi) = & - \sum_{i \in I} \int_{K_i} \sum_{k=1}^d \mathbb{F}_k^s(\mathbf{w}_h) \cdot \frac{\partial \varphi}{\partial x_k} dx \\ & + \sum_{i \in I} \sum_{j \in S(i)} \int_{\Gamma_{ij}} \mathbf{H}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}) \cdot \varphi dS. \end{aligned} \quad (3.12)$$

The proper selection of a numerical flux is a main part for an efficient and accurate numerical solution, therefore this topic is discussed in a separate section.

3.3. Numerical flux

As previously was mentioned, a basic ingredient for a precise, stable and efficient numerical scheme is an adequate numerical flux. From the characteristic theory is well known that for a supersonic flow the waves travel from upstream to downstream, and correspondingly the information \mathbf{w} used by the numerical flux must come from upstream side of interface Γ . Nevertheless, for subsonic case the waves can travel from upstream to downstream or vice versa. In these cases the numerical flux should take the information from both sides

of the interface, i.e. it must take in account both $\mathbf{w}|_{\Gamma_{ij}} = \mathbf{w}^+$ and $\mathbf{w}|_{\Gamma_{ji}} = \mathbf{w}^-$. A bigger challenge to design (or select) an efficient numerical flux is found in cases where subsonic and supersonic flows there exist into the time-space domain, e.g. in the shock waves a subsonic region is found at one side of the shock, along with a supersonic region at the other.

Definition 3.3.1. Let $\mathbf{H}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n})$ be a generic numerical flux, let \mathcal{D} be the domain of the flux $\mathbf{f}_j^s(\mathbf{w})$ and let $\mathcal{S} = \{\mathbf{n} \in \mathbb{R}^d : |\mathbf{n}| = 1\}$, then the numerical flux \mathbf{H} should satisfy the next properties ([LeV02]):

- \mathbf{H} is defined and is Lipschitz continuous on the domain $\mathcal{D} \times \mathcal{D} \times \mathcal{S}$.
- \mathbf{H} is consistent, i.e.

$$\mathbf{H}(\mathbf{u}, \mathbf{u}, \mathbf{n}) = \mathbf{F}^n(\mathbf{u}, \mathbf{n}) = \sum_{j=i}^d \mathbb{F}_j^s(\mathbf{u}) n_j$$

for all $\mathbf{u} \in \mathcal{D}$ and $\mathbf{n} \in \mathcal{S}$.

- \mathbf{H} is conservative, i.e.

$$\mathbf{H}(\mathbf{u}, \mathbf{v}, \mathbf{n}) = -\mathbf{H}(\mathbf{v}, \mathbf{u}, -\mathbf{n})$$

for all $\mathbf{u}, \mathbf{v} \in \mathcal{D}$ and $\mathbf{n} \in \mathcal{S}$.

There exist a big number of research works focused on developing adequate numerical fluxes for different kind of problems. Here, these numerical fluxes are classified into two general groups: central schemes and upwind schemes.

3.3.1. Central schemes

The basic prototype for a central scheme is given by

$$\mathbf{H}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}) = \frac{1}{2}(\mathbf{F}^n(\mathbf{w}^+, \mathbf{n}) + \mathbf{F}^n(\mathbf{w}^-, \mathbf{n})) + \mathbf{F}_{ad}, \quad (3.13)$$

where the first term at right hand side represents a simple average flux, which is generally instable for hyperbolic problems ([LeV02]), while the term \mathbf{F}_{ad} represents a stabilization term, necessary in order to damp (via numeric diffusion) the oscillations produced by the first term.

3.3.1.1. Lax–Friedrichs numerical flux

One of the most popular central schemes is the Lax–Friedrichs numerical flux, which is defined as

$$\mathbf{H}_{\text{LF}}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}) = \frac{1}{2}(\mathbf{F}^n(\mathbf{w}^+, \mathbf{n}) + \mathbf{F}^n(\mathbf{w}^-, \mathbf{n})) - \alpha(\mathbf{w}^- - \mathbf{w}^+), \quad (3.14)$$

where α is defined as the spectral radii of the Jacobi matrix of \mathbf{F}^n , i.e.

$$\alpha = \frac{1}{2} \max_{\mathbf{v}=\mathbf{w}^+, \mathbf{w}^-} \{|\lambda(\mathbf{P}(\mathbf{v}, \mathbf{n}))|\}.$$

3.3.1.2. Lax–Wendroff numerical flux

Unlike Lax–Friedrichs numerical flux (with first order of accuracy), a scheme with second order of accuracy in space and time can be formulated using ([LW60]):

$$\begin{aligned} \mathbf{H}_{\text{LW}}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}) &= \frac{1}{2}(\mathbf{F}^n(\mathbf{w}^+, \mathbf{n}) + \mathbf{F}^n(\mathbf{w}^-, \mathbf{n})) \\ &\quad - \frac{\tau}{2h_K} \mathbf{P} \left(\frac{1}{2}(\mathbf{w}^+ + \mathbf{w}^-), \mathbf{n} \right) (\mathbf{F}^n(\mathbf{w}^-, \mathbf{n}) - \mathbf{F}^n(\mathbf{w}^+, \mathbf{n})), \end{aligned} \quad (3.15)$$

where τ is the time step size and h_K is a measure for the mesh size.

3.3.2. Upwind schemes

The basic idea around this methods is to use the characteristics theory in order to predict how the waves are traveling, and then to identify from where the information are coming. Although for the scalar linear case the application of this strategy is straight, for nonlinear systems these techniques represent a harder challenge. These group of methods can be classified in three subgroups: flux difference splitting methods (FDSM), flux vector splitting methods (FVSM) and advection upstream splitting methods (AUSM).

Lemma 3.3.2. *Consider the Riemann problem given by the linear system*

$$\frac{\partial \mathbf{w}}{\partial t} + \mathbb{A} \frac{\partial \mathbf{w}}{\partial x} = 0, \quad (3.16)$$

with the initial condition

$$\mathbf{w}(x, 0) = \begin{cases} \mathbf{w}^+, & \text{if } x < 0, \\ \mathbf{w}^-, & \text{if } x > 0, \end{cases} \quad (3.17)$$

where $\mathbb{A} \in \mathbb{R}^{m,m}$ is a diagonalizable matrix. From it follows that there exist a matrix \mathbf{R} such that

$$\mathbb{A} = \mathbf{R} \mathbf{\Lambda} \mathbf{R}^{-1}$$

with

$$\mathbf{R} = (\mathbf{r}_1 \dots \mathbf{r}_m)$$

and

$$\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_m),$$

being $\lambda_i \in \mathbb{R}$, $i = 1, \dots, m$, the i -th eigenvalue and $\mathbf{r}_i \in \mathbb{R}^m$ the associated eigenvector. Thus, assuming that \mathbf{w} is constant in the domain $\Omega_i = \{(x, t); t > 0, \lambda_i < x/t < \lambda_{i+1}\}$ it is possible to show that for all $(x, t) \in \Omega_i$ the following flux expressions are equivalent

- $\mathbb{A} \mathbf{w}(x, t) = \mathbb{A}^+ \mathbf{w}^+ + \mathbb{A}^- \mathbf{w}^-$,
- $\mathbb{A} \mathbf{w}(x, t) = \mathbb{A} \mathbf{w}^+ + \int_{\mathbf{w}^+}^{\mathbf{w}^-} \mathbb{A}^- d\mathbf{w}$,
- $\mathbb{A} \mathbf{w}(x, t) = \mathbb{A} \mathbf{w}^- - \int_{\mathbf{w}^+}^{\mathbf{w}^-} \mathbb{A}^+ d\mathbf{w}$,
- $\mathbb{A} \mathbf{w}(x, t) = \frac{1}{2} (\mathbb{A} \mathbf{w}^- + \mathbb{A} \mathbf{w}^+) - \frac{1}{2} \int_{\mathbf{w}^+}^{\mathbf{w}^-} |\mathbb{A}| d\mathbf{w}$,

where

$$\mathbb{A} = \mathbb{A}^+ + \mathbb{A}^-, \quad (3.18)$$

$$\mathbb{A}^\pm = \mathbf{R} \mathbf{\Lambda}^\pm \mathbf{R}^{-1}, \quad (3.19)$$

$$|\mathbb{A}| = \mathbf{R} |\mathbf{\Lambda}| \mathbf{R}^{-1} = \mathbb{A}^+ - \mathbb{A}^- \quad (3.20)$$

$$|\mathbf{\Lambda}| = \text{diag}(|\lambda_1|, \dots, |\lambda_m|) \quad (3.21)$$

and

$$\mathbf{\Lambda}^\pm = \text{diag}(\lambda_1^\pm, \dots, \lambda_m^\pm). \quad (3.22)$$

Proof. Proof of this lemma can be found in [FFS03]. □

Remark 3.3.3. *Considering the nonlinear Riemann problem*

$$\frac{\partial \mathbf{w}}{\partial t} + \frac{\partial}{\partial x} \mathbf{f}^s(\mathbf{w}) = 0, \quad (3.23)$$

with the initial condition Eq. 3.17, where $\mathbf{f}^s(\mathbf{w}) \in \mathbb{R}^m$. Thus, in analogy with the linear case in Lemma 3.3.2, it is possible to write the Godunov numerical flux \mathbf{H}_R as

$$\begin{aligned} \mathbf{H}_R(\mathbf{w}^+, \mathbf{w}^-) &= \mathbf{f}^s(\mathbf{w}^+) + \int_{\mathbf{w}^+}^{\mathbf{w}^-} \mathbb{A}^- d\mathbf{w} \\ &= \mathbf{f}^s(\mathbf{w}^-) - \int_{\mathbf{w}^+}^{\mathbf{w}^-} \mathbb{A}^+ d\mathbf{w} \\ &= \frac{1}{2}(\mathbf{f}^s(\mathbf{w}^+) + \mathbf{f}^s(\mathbf{w}^-)) - \frac{1}{2} \int_{\mathbf{w}^+}^{\mathbf{w}^-} |\mathbb{A}| d\mathbf{w}, \end{aligned} \quad (3.24)$$

where \mathbb{A}^\pm , and $|\mathbb{A}|$ are given by Eq. 3.19 and Eq. 3.20, respectively.

3.3.2.1. Flux difference splitting schemes

A natural idea consists in treating each interface jump as a 1D-Riemann problem, thus the exact solution allows to compute the numerical flux. This idea was presented by Godunov in his seminal paper ([God59]) and onwards it inspired a wide amount of numerical methods (also named Godunov methods). This scheme has shown a good accuracy, even in presence of shock waves and contact discontinuities ([Tor99]), preserving the positivity of density and pressure. Nevertheless, as in the Eq. 3.24, the exact solution of the Riemann problem implies the analytical integration of Riemann invariants and it involves two basic drawbacks:

- On the one hand, in practical 2D-3D models, to solve this problem on each interface can demand a huge computational cost and make the method unfeasible. Hence, a lot of efforts are focus on to develop approximate solutions to Riemann problems. Some of the most recognized methods were presented by Roe ([Roe81]), Osher and Solomon ([OS82]) and Harten et al. ([HL81]).
- The analytical formulation of integral in Eq. 3.24 should be built for each specific problem, but may not be possible for some problems. Particularly, in the context of inviscid gases obeying a non-ideal EOS, some authors have proposed exact Riemann solutions for specific cases, e.g. Colella and Graz ([CG85]) present an exact solution considering a general convex EOS, while Surel et al. ([SLL94]) develop a solution taking a EOS of the form $p = p(\rho, \theta)$ and Quartapelle et al. ([QCGQ03]) focus on the van der Waals EOS.

Roe linear approximation

In order to reduce complexity of computing the exact solution of the Riemann problem Eq. 3.24, involved in the numerical solution of Euler equations, Roe and Pike ([Roe81]) presented a linearization technique. The basic idea behind this approach, which is one of the most accurate methods available, is to replace the nonlinear Riemann problem by a linear case that approximates as well as the original nonlinear formulation. Thus, Roe's linear problem is

$$\frac{\partial \mathbf{w}}{\partial t} + \tilde{\mathbb{A}}(\mathbf{w}^+, \mathbf{w}^-) \frac{\partial \mathbf{w}}{\partial x} = \mathbf{0}, \quad (3.25)$$

with the initial condition Eq. 3.17. Here $\tilde{\mathbb{A}}$ is a matrix (locally constant) with the following properties:

- $\tilde{\mathbb{A}}(\mathbf{w}^+, \mathbf{w}^-)$ is diagonalizable.
- $\tilde{\mathbb{A}}(\mathbf{w}^+, \mathbf{w}^-)$ is consistent with the exact Jacobian, i.e.

$$\tilde{\mathbb{A}}(\mathbf{w}, \mathbf{w}) = \mathbb{A}(\mathbf{w}).$$

- $\tilde{\mathbb{A}}(\mathbf{w}^+, \mathbf{w}^-)$ is conservative across discontinuities, i.e.

$$\mathbf{f}^s(\mathbf{w}^+) - \mathbf{f}^s(\mathbf{w}^-) = \tilde{\mathbb{A}}(\mathbf{w}^+, \mathbf{w}^-)(\mathbf{w}^+ - \mathbf{w}^-).$$

For the Euler equations and using the ideal gas EOS, Roe defines the matrix $\tilde{\mathbb{A}}$ equal to matrix \mathbb{A}_1 in Remark 2.2.7, but evaluated in an average state $\hat{\mathbf{w}}$, i.e.,

$$\tilde{\mathbb{A}}(\mathbf{w}^+, \mathbf{w}^-) = \mathbb{A}_1(\hat{\mathbf{w}}),$$

where the average state is given by

$$\begin{aligned} \hat{\mathbf{w}} &= \left(\hat{\rho} \quad \hat{\rho} \hat{v}_1 \quad \hat{\rho} \hat{v}_2 \quad \hat{\rho} \hat{v}_3 \quad \hat{E} \right)^T, \\ \sqrt{\hat{\rho}} &= \frac{1}{2} \left(\sqrt{\rho^+} + \sqrt{\rho^-} \right), \\ \hat{v}_1 &= \frac{\sqrt{\rho^+} v_1^+ + \sqrt{\rho^-} v_1^-}{\sqrt{\rho^+} + \sqrt{\rho^-}}, \\ \hat{v}_2 &= \frac{\sqrt{\rho^+} v_2^+ + \sqrt{\rho^-} v_2^-}{\sqrt{\rho^+} + \sqrt{\rho^-}}, \\ \hat{v}_3 &= \frac{\sqrt{\rho^+} v_3^+ + \sqrt{\rho^-} v_3^-}{\sqrt{\rho^+} + \sqrt{\rho^-}}, \\ \hat{E} &= \frac{1}{\gamma} \hat{\rho} \hat{h} + \frac{\gamma - 1}{2\gamma} \hat{\rho} |\hat{v}|^2, \\ \hat{h} &= \frac{\sqrt{\rho^+} h^+ + \sqrt{\rho^-} h^-}{\sqrt{\rho^+} + \sqrt{\rho^-}}. \end{aligned} \tag{3.26}$$

Finally, knowing the eigenvalues and eigenvectors of $\mathbb{A}_1(\hat{\mathbf{w}})$ from Remark 2.2.7, and using the average speed of sound

$$\hat{a} = (\gamma - 1) \left(\hat{h} - \frac{1}{2} |\hat{v}|^2 \right),$$

the numerical flux can be computed using any of the expressions in Lemma 3.3.2. Some extra details to implement the method can be found in ([FFS03]).

As have been reported in multiple works, this scheme is able to reproduce linear waves (contact discontinuities) with a high resolution, whereby it is a suitable method to capture boundary layers when Navier–Stokes are used to model problems. Nevertheless, its formulation allows to have non-physical expansion waves which are clearly undesirable ([CL95]).

Some authors have extended this method to the case of real gases, e.g. Glaister ([Gla89]) solves 1D Euler equations along with a convex EOS using a linearized Riemann problem. Extensions to 2D–3D problems were developed by the same author using a splitting operator. Vinokur and Montagné ([VM90]) present a generalized Roe average approximation to Riemann problems to arbitrary equilibrium gas laws. Likewise, Grossman and Walters ([GW89a]), and later Liou and van Leer ([LLS90]), developed an approach to the solution of Euler equations for real gases adapting the Roe formulation through an equivalent function $\bar{\gamma}$ such that $\bar{\gamma} = \bar{\gamma}(\rho, e)$.

Later, Guardone and Vigevano ([GV02]) present an extension to Roe linearization for problems with van der Waal gases, and Cinnella ([Cin06]) proposes some simplifications to the Roe method applied to problems with dense gases. More recently, Moon et al. ([MSC11]) deduce an average state for Roe linearization considering that the compressibility factor is locally constant between the interfaces.

3.3.2.2. Flux vector splitting schemes

Considering the fluid flow as a group of particles moving in and out of each cell, then it is natural to express the total flux through a cell interface as a sum of a positive flux (leaving the cell) and a negative flux (going into the cell), i.e.

$$\mathbf{F}^n(\mathbf{w}, \mathbf{n}) = \mathbf{F}^+(\mathbf{w}, \mathbf{n}) + \mathbf{F}^-(\mathbf{w}, \mathbf{n}). \quad (3.27)$$

This family of methods leads to simpler techniques (normally supported by a less sophisticated theory), which are easier to implement and have lower computational cost than the Gudunov methods (FDSM). Nevertheless, the solutions found with this schemes are regularly less accurate and exhibit an excessive numerical diffusion, producing poor resolution in discontinuities for inviscid problems ([Tor99]) and smearing boundary layers in applications to the Navier–Stokes equations ([LTR87]).

Steger–Warming flux-vector splitting

Steger and Warming ([SW81]) represent the forward and backward fluxes, respectively, as

$$\mathbf{F}^\pm(\mathbf{w}^\pm, \mathbf{n}) = \mathbf{P}^\pm(\mathbf{w}^\pm, \mathbf{n})\mathbf{w}^\pm,$$

hence the numerical flux can be written as

$$\mathbf{H}_{\text{sw}}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}) = \mathbf{P}^+(\mathbf{w}^+, \mathbf{n})\mathbf{w}^+ + \mathbf{P}^-(\mathbf{w}^-, \mathbf{n})\mathbf{w}^-, \quad (3.28)$$

where

$$\mathbf{P}^\pm(\mathbf{w}, \mathbf{n}) = \sum_{j=1}^d \mathbb{A}_j^\pm(\mathbf{w})n_j.$$

A equivalent formulation, inspired by Eq. 3.24, is

$$\begin{aligned} \mathbf{H}_{\text{sw}}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}) &= \frac{1}{2} (\mathbf{F}^n(\mathbf{w}^+, \mathbf{n}) + \mathbf{F}^n(\mathbf{w}^-, \mathbf{n})) \\ &\quad - \frac{1}{2} (|\mathbf{P}(\mathbf{w}^-, \mathbf{n})|\mathbf{w}^- - |\mathbf{P}(\mathbf{w}^+, \mathbf{n})|\mathbf{w}^+), \end{aligned} \quad (3.29)$$

where $|\mathbf{P}(\mathbf{w}^\pm, \mathbf{n})|$ is expressed according to the definitions in Lemma 3.3.2. From Eq. 3.28 or Eq. 3.29 it follows that the homogeneity property in Lemma 2.2.3 should be satisfied, and then this split can only be applied in cases with flows of thermally perfect gases or ideal gases. Nevertheless, some extensions to real gases have been developed by Liou and Van Leer ([LLS90]), Montagné et al. ([MY89, VM90]) and Grossman and Walters ([GW89a, GW89b]).

Vijayasundaram flux-vector splitting

A partial upwind scheme can be derived from Eq. 3.28 evaluating the Jacobi matrix in a average state ([Vij86]), i.e.

$$\mathbf{H}_{\text{vy}}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}) = \mathbf{P}^+\left(\frac{\mathbf{w}^+ + \mathbf{w}^-}{2}, \mathbf{n}\right)\mathbf{w}^+ + \mathbf{P}^-\left(\frac{\mathbf{w}^+ + \mathbf{w}^-}{2}, \mathbf{n}\right)\mathbf{w}^-. \quad (3.30)$$

As in Steger–Warming splitting, the flux vector should be a homogeneous function of degree one in order to apply this scheme. Thus, this formulation is only reserved to thermally perfect gases.

Lemma 3.3.4. *Let $d = 1$ and let p be defined by the ideal gas EOS, then the inviscid flux term $\mathbb{F}^s(\mathbf{w})$ in Eq. 2.40 can be expressed as*

$$\mathbb{F}_M^s(\rho, M(\mathbf{w}), c(\mathbf{w})) = \begin{pmatrix} \rho c M \\ \frac{\rho c^2}{\gamma} (\gamma M^2 + 1) \\ \rho c^3 M \left(\frac{1}{\gamma-1} + \frac{M^2}{2} \right) \end{pmatrix}, \quad (3.31)$$

where

$$M(\mathbf{w}) = \frac{v_1}{c(\mathbf{w})} \quad (3.32)$$

is the dimensionless Mach number.

Proof. From Mach number definition to use $v_1 = Mc$, as well as $p = c^2 \rho / \gamma$ from Eq. 2.61, to change v_1 and p in $\mathbb{F}^s(\mathbf{w})$ (Eq. 2.15) by their equivalent expressions in terms of ρ , M and c . \square

Van Leer flux-vector splitting

Using the inviscid flux expression given in Lemma 3.3.4, van Leer ([L82]) proposes a numerical flux based on the splitting of the Mach number as follows:

- Mass flux splitting: the term $\rho c M$ is rewritten using a splitting for the Mach number such that

$$\rho c M \equiv f_m^+(\mathbf{w}^+) + f_m^-(\mathbf{w}^-), \quad (3.33)$$

where

$$f_m^\pm(\mathbf{w}^\pm) = \rho^\pm c(\mathbf{w}^\pm) \mathcal{M}_{(2)}^\pm(M(\mathbf{w}^\pm)), \quad (3.34)$$

$$\mathcal{M}_{(2)}^\pm(M(\mathbf{w})) = \begin{cases} \pm \frac{1}{4} (M \pm 1)^2, & \text{if } |M| < 1, \\ \mathcal{M}_{(1)}^\pm, & \text{otherwise.} \end{cases} \quad (3.35)$$

and

$$\mathcal{M}_{(1)}^\pm(M(\mathbf{w})) = \frac{1}{2} (M \pm |M|). \quad (3.36)$$

- Momentum flux splitting: the momentum flux is split using

$$\frac{\rho c^2}{\gamma} (\gamma M^2 + 1) \equiv f_v^+(\mathbf{w}^+) + f_v^-(\mathbf{w}^-), \quad (3.37)$$

where

$$f_v^\pm = \frac{\rho^\pm c^2(\mathbf{w}^\pm)}{\gamma} \mathcal{M}_{(3)}^\pm(M(\mathbf{w}^\pm)), \quad (3.38)$$

$$\mathcal{M}_{(3)}^+(M(\mathbf{w})) = \begin{cases} 0, & \text{if } M \leq -1, \\ \frac{1}{4} (M + 1)^2 ((\gamma - 1)M + 2), & \text{if } -1 < M < 1, \\ \gamma M^2 + 1, & \text{otherwise} \end{cases} \quad (3.39)$$

and

$$\mathcal{M}_{(3)}^-(M(\mathbf{w})) = \begin{cases} \gamma M^2 + 1, & \text{if } M \leq -1, \\ -\frac{1}{4} (M - 1)^2 ((\gamma - 1)M - 2) & \text{if } -1 < M < 1, \\ 0, & \text{otherwise.} \end{cases} \quad (3.40)$$

- Energy flux splitting: the energy flux is rewritten using the splitting

$$\rho M c^3 \left(\frac{1}{\gamma - 1} + \frac{M^2}{2} \right) \equiv f_E^+(\mathbf{w}^+) + f_E^-(\mathbf{w}^-), \quad (3.41)$$

where

$$f_E^+(\mathbf{w}) = \begin{cases} 0, & \text{if } M \leq -1, \\ \frac{\gamma^2}{2(\gamma^2-1)} \frac{f_M^{+2}}{f_m^+}, & \text{if } -1 < M < 1, \\ \rho M c^3 \left(\frac{1}{\gamma-1} + \frac{M^2}{2} \right), & \text{otherwise,} \end{cases} \quad (3.42)$$

and

$$f_E^-(\mathbf{w}) = \begin{cases} \rho M c^3 \left(\frac{1}{\gamma-1} + \frac{M^2}{2} \right), & \text{if } M \leq -1, \\ \frac{\gamma^2}{2(\gamma^2-1)} \frac{f_M^{-2}}{f_m^-}, & \text{if } -1 < M < 1, \\ 0, & \text{otherwise.} \end{cases} \quad (3.43)$$

Thus, for a one dimensional case, the final numerical flux formula can be expressed as

$$\mathbf{H}_{\text{VL}}(\mathbf{w}^+, \mathbf{w}^-) = \mathbf{F}_M^+(\mathbf{w}^+) + \mathbf{F}_M^-(\mathbf{w}^-), \quad (3.44)$$

with

$$\mathbf{F}_M^\pm(\mathbf{w}) = \begin{pmatrix} f_m^\pm(\mathbf{w}) \\ f_V^\pm(\mathbf{w}) \\ f_E^\pm(\mathbf{w}) \end{pmatrix}. \quad (3.45)$$

This strategy produces a total upwind formulation for supersonic flows and a kind of centered formulation for subsonic regime. The second order polynomial used to define the Mach number in the subsonic range allows to solve the continuity problems in sonic points ($M = 1$) and stagnation points ($|\mathbf{v}| = 0$) encountered with the Steger-Warming formulation. Nevertheless, it is well known that this scheme introduces excessive numerical diffusion over contact discontinuities ([L82]). An alternative multidimensional form for this numerical flux is given by ([FFS03])

$$\begin{aligned} \mathbf{H}_{\text{VL}}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}) &= \frac{1}{2} (\mathbf{F}^n(\mathbf{w}^+, \mathbf{n}) + \mathbf{F}^n(\mathbf{w}^-, \mathbf{n})) \\ &\quad - \frac{1}{2} \left| \mathbf{P} \left(\frac{\mathbf{w}^+ + \mathbf{w}^-}{2}, \mathbf{n} \right) \right| (\mathbf{w}^- - \mathbf{w}^+). \end{aligned} \quad (3.46)$$

Some extensions of this numerical flux to real gases are presented in ([LLS90, MY89, GW89a]).

Lemma 3.3.5. *Let $d = 1$ and let $p = p(\mathbf{w})$ be a general EOS, then the inviscid flux term $\mathbb{F}^s(\mathbf{w})$ in Eq. 2.40 can be expressed as the sum of a convective and an acoustic term*

$$\mathbb{F}_{MP}^s(\mathbf{w}, M(\mathbf{w}), c(\mathbf{w})) = \rho c M \begin{pmatrix} 1 \\ v_1 \\ h \end{pmatrix} + \begin{pmatrix} 0 \\ p \\ 0 \end{pmatrix}, \quad (3.47)$$

where the Mach number is $M(\mathbf{w}) = v_1/c$ and the specific enthalpy $h(\mathbf{w})$ is defined in accordance with Eq. 2.48.

Liou–Steffen flux-vector splitting

Recognizing the physical differences between convective and acoustic waves, Liou and Steffen ([LS93]) propose an improved version of the van Leer flux-vector splitting. Thus, using the inviscid flux expression given in Lemma 3.3.5, a different splitting is designed for each flux component: a Mach number splitting for the convective flux and a pressure splitting for the acoustic term. Using this idea, the numerical flux for an one-dimensional problem can be written as

$$\mathbf{H}_{\text{LS}}(\mathbf{w}^+, \mathbf{w}^-) = \rho_{\frac{1}{2}}(\mathbf{w}^+, \mathbf{w}^-) c_{\frac{1}{2}}(\mathbf{w}^+, \mathbf{w}^-) M_{\frac{1}{2}}(\mathbf{w}^+, \mathbf{w}^-) \mathbf{F}_{\frac{1}{2}}(\mathbf{w}^+, \mathbf{w}^-) + \mathbf{F}_P(\mathbf{w}^+, \mathbf{w}^-), \quad (3.48)$$

where

$$M_{\frac{1}{2}}(\mathbf{w}^+, \mathbf{w}^-) = \mathcal{M}_{(2)}^+(\mathcal{M}(\mathbf{w}^+)) + \mathcal{M}_{(2)}^-(\mathcal{M}(\mathbf{w}^-)), \quad (3.49)$$

$$\rho_{\frac{1}{2}}(\mathbf{w}^+, \mathbf{w}^-) = \begin{cases} \rho^+, & \text{if } M_{\frac{1}{2}} \geq 0, \\ \rho^-, & \text{otherwise,} \end{cases} \quad (3.50)$$

$$c_{\frac{1}{2}}(\mathbf{w}^+, \mathbf{w}^-) = \begin{cases} c(\mathbf{w}^+), & \text{if } M_{\frac{1}{2}} \geq 0, \\ c(\mathbf{w}^-), & \text{otherwise,} \end{cases} \quad (3.51)$$

$$\mathbf{F}_{\frac{1}{2}}(\mathbf{w}^+, \mathbf{w}^-) = \begin{cases} \mathbf{F}_C(\mathbf{w}^+), & \text{if } M_{\frac{1}{2}} \geq 0, \\ \mathbf{F}_C(\mathbf{w}^-), & \text{otherwise,} \end{cases} \quad (3.52)$$

$$\mathbf{F}_C(\mathbf{w}) = \begin{pmatrix} 1 \\ v_1 \\ h \end{pmatrix}, \quad (3.53)$$

$$\mathbf{F}_P(\mathbf{w}^+, \mathbf{w}^-) = \begin{pmatrix} 0 \\ P^+(\mathcal{M}(\mathbf{w}^+))p^+ + P^-(\mathcal{M}(\mathbf{w}^-))p^- \\ 0 \end{pmatrix} \quad (3.54)$$

and

$$P^\pm(\mathcal{M}(\mathbf{w})) = \begin{cases} \frac{1}{2}(1 \pm \text{sign}(\mathcal{M})), & \text{if } |\mathcal{M}| \geq 1, \\ \frac{1}{4}(\mathcal{M} \pm 1)^2(2 \mp \mathcal{M}), & \text{otherwise.} \end{cases} \quad (3.55)$$

This scheme allows to get at the same time a simple formulation, as with the flux-vector splitting methods, and an accuracy comparable to Roe's method. Additionally, this method is easily extended to multiple dimensions and it does not require that the flux satisfies the homogeneity property of Lemma 2.2.3, hence it can be applied indistinctly to ideal or real gases. This scheme has inspired multiple extensions and enhanced versions, giving birth to a family of methods termed *AUSM*-schemes.

3.3.3. The *AUSM*⁺-up scheme

A third category of numerical fluxes are the called *advection upstream splitting methods* (*AUSM*). These methods share the good characteristics of *FDSM* and *FVSM*. On the one hand, these are as low dissipative and highly accurate as *FDSM* ([Lio01]), and on the other hand they exhibit an easy formulation and a high computational efficiency as the flux-vector splitting methods. Part of the simplicity of these methods lies in they do not require to compute any Jacobi matrix or to develop an eigen-decomposition in the inviscid flux term. Consequently the *AUSM* can be easily extended to real gases

([RCG13, EFL00, EFL99]) and to other systems of conservation laws as multiphase flows, chemical nonequilibrium flows or turbulence models ([CL07, GCP06, MEKS03]). In their seminal work, Liou and Steffen ([LS93]) present a double splitting: the first, based on a Mach number splitting, is applied to the convective flux, and a second splitting applied to the pressure flux. However, despite its good accuracy in many cases, both the *AUSM* and its improved version, called *AUSM*⁺ ([Lio96]), report pressure oscillations in regions with low Mach numbers ([Lio06]). Multiple versions have been developed, inspired in *AUSM* and *AUSM*⁺, looking for a robust method with a wide range of Mach numbers ([Lio10]). Here we only will present one of the most robust and accurate method of this family, the *AUSM*⁺-*up*. The scheme *AUSM*⁺-*up* ([Lio06]) is derived from the original splitting of Liou and Steffen, but some enhancement have been included in order to improve the accuracy specially with small Mach numbers. The numerical flux with this scheme can be expressed as

$$\mathbf{H}_{\text{AUSM}}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}) = \rho_{\frac{1}{2}}(\mathbf{w}^+, \mathbf{w}^-) c_{\frac{1}{2}}(\mathbf{w}^+, \mathbf{w}^-) \mathbf{M}_{\frac{1}{2}}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}) \mathbf{F}_{\frac{1}{2}}(\mathbf{w}^+, \mathbf{w}^-) + \mathbf{F}_P(\mathbf{w}^+, \mathbf{w}^-), \quad (3.56)$$

where

- The $\rho_{\frac{1}{2}}$ and $c_{\frac{1}{2}}$ terms are computed as a simple average

$$\rho_{\frac{1}{2}} = \frac{\rho^+ + \rho^-}{2}, \quad (3.57) \quad c_{\frac{1}{2}} = \frac{c(\mathbf{w}^+) + c(\mathbf{w}^-)}{2}. \quad (3.58)$$

- The average Mach number $\mathbf{M}_{\frac{1}{2}}$ is defined using a splitting with higher order polynomials, with respect to the Liou–Steffen scheme, in order to increase the accuracy of the *AUSM* scheme ([Lio96]). Additionally, a pressure diffusion term M_p was introduced with the purpose of conserving the coupling between velocity and pressure, specially at small Mach numbers. $\mathbf{M}_{\frac{1}{2}}$ is written as

$$\mathbf{M}_{\frac{1}{2}}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}) = \mathcal{M}_{(4)}^+(\mathbf{M}(\mathbf{w}^+)) + \mathcal{M}_{(4)}^-(\mathbf{M}(\mathbf{w}^-)) - M_p(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}), \quad (3.59)$$

where

$$\mathcal{M}_{(4)}^{\pm}(\mathbf{M}(\mathbf{w})) = \begin{cases} \mathcal{M}_{(2)}^{\pm}(1 \mp 16\beta\mathcal{M}_{(2)}^{\mp}), & \text{if } |\mathbf{M}| < 1, \\ \mathcal{M}_{(1)}^{\pm}, & \text{otherwise,} \end{cases} \quad (3.60)$$

$$M_p(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}) = \frac{K_p}{f_a} \max(1 - \sigma \bar{M}^2, 0) \frac{p^+ - p^-}{\rho_{\frac{1}{2}} a_{\frac{1}{2}}^2}, \quad (3.61)$$

$\mathcal{M}_{(2)}^{\pm}$ was defined in Eq. 3.35, $\beta = \frac{1}{8}$ ([Lio06]), $0 \leq K_p \leq 1$, $\sigma \leq 1$ and $f_a \in [0, 1]$ is a scaling factor that increases the pressure diffusion term as $\mathbf{M} \rightarrow 0$. This term is computed as

$$f_a(M_o) = M_o(2 - M_o), \quad (3.62)$$

with

$$M_o^2 = \min(1, \max(\bar{M}^2, M_{\infty}^2)), \quad (3.63)$$

$$\bar{M}^2 = \frac{v_n^{+2} + v_n^{-2}}{2c_{\frac{1}{2}}^2}, \quad (3.64)$$

and M_{∞} being the far field Mach number. As in Eq. 3.64, in this scheme each Mach number should be computed using the average speed of sound, i.e.

$$\mathbf{M} = \frac{v_n}{c_{\frac{1}{2}}}. \quad (3.65)$$

- The acoustic term \mathbf{F}_P is splitting using the expression

$$\mathbf{F}_P(\mathbf{w}^+, \mathbf{w}^-) = \mathcal{P}_{(5)}^+(\mathbf{M}(\mathbf{w}^+)) p^+ + \mathcal{P}_{(5)}^-(\mathbf{M}(\mathbf{w}^-)) p^- + p_u, \quad (3.66)$$

where

$$\mathcal{P}_{(5)}^\pm(\mathbf{M}(\mathbf{w})) = \begin{cases} \mathcal{M}_{(2)}^\pm((\pm 2 - \mathbf{M}) \mp 16\alpha \mathbf{M} \mathcal{M}_{(2)}^\mp), & \text{if } |\mathbf{M}| < 1, \\ \frac{1}{\mathbf{M}} \mathcal{M}_{(1)}^\pm, & \text{otherwise,} \end{cases} \quad (3.67)$$

$\alpha = 3/16$ ([Lio06]) and p_u is a diffusion term expressed as

$$p_u = -K_u \mathcal{P}_{(5)}^+(\mathbf{M}(\mathbf{w}^+)) \mathcal{P}_{(5)}^-(\mathbf{M}(\mathbf{w}^-)) f_a c_{\frac{1}{2}} (\rho^+ + \rho^-) (v_n^- - v_n^+), \quad (3.68)$$

with $0 \leq K_u \leq 1$. The terms $\mathcal{P}_{(5)}^+$ and $\mathcal{P}_{(5)}^-$ are used to switch off the velocity diffusion term in the supersonic regime.

Extensive numerical tests and applications to physical problems have shown that this scheme is reliable and effective in a wide range of Mach numbers, as in the subsonic level as in the supersonic regime. Additionally, Liou ([Lio06]) showed that this method allows to capture exactly contact discontinuities and shocks waves, and to preserve the positivity property in mass (density).

3.3.4. Hybrid-flux methods

In general, the *FDSM* are able to reproduce linear waves (contact discontinuities) with a high resolution, but normally with a high computational demand. On the other hand, the *FVSM* are very robust to capture genuinely non-linear waves in a computational efficient way, although it is well known that these methods introduce excessive numerical diffusion in contact discontinuities. Thus, a lot of efforts lead to develop robust and efficient hybrid schemes combining the strengths of two or more numerical fluxes. Coquel and Liou ([CL95]) proposed a hybrid upwind splitting (*HUS*) where an Osher scheme is used in fields where linear waves are presented, and a van Leer flux splitting in regions where nonlinear waves are localized. De Vuyst ([V04]) presents a numerical hybrid flux calculated using a weighted average of a Lax-Wendroff second order numerical flux and a first order Lax-Friedrichs flux, as follows

$$\mathbf{H}_{\text{HYB}} = \Theta \mathbf{H}_{\text{LF}} + (1 - \Theta) \mathbf{H}_{\text{LW}}, \quad (3.69)$$

where the combination coefficient $0 \leq \Theta \leq 1$, designed to locally control the amount of numerical entropy dissipation, switches off the high order flux in regions near to discontinuities, activating the lower order flux, looking for preserving the stability. In smooth regions one lets $\Theta \rightarrow 0$ and the numerical flux is close to Lax-Wendroff scheme. A similar technique is presented in [Hab95], but combining a *FDSM* and a *FVSM* to produce a hybrid scheme. In similar fashion to Eq. 3.69, these flux combination hybrid schemes can be written as

$$\mathbf{H}_{\text{HYB}} = \mathbf{H}_{\text{HO}} - \Theta (\mathbf{H}_{\text{HO}} - \mathbf{H}_{\text{LO}}), \quad (3.70)$$

where \mathbf{H}_{HO} and \mathbf{H}_{LO} represent a generic high order and low order numerical flux, respectively. Thus, these hybrid schemes are equivalent to the flux corrected transport schemes (*FCT*) proposed by Boris and Brook ([BB73]) and extended by Zalesak ([Zal79]). In this last scheme a first stage is used to compute the flux using a high order scheme, while in a second stage this flux is corrected using the difference with a lower order numerical flux (the difference is called antidiffusive flux).

In all this schemes a big part of the success consists in identifying regions (or cells) where

a higher diffusive scheme (with a lower order of accuracy) should be used. Normally these regions correspond to the vicinity of steep gradients or discontinuities. Thus, it is important to employ efficient and reliable discontinuity indicators to save the accuracy of the hybrid method. In [DFS03, QS05a, LQ10a, KXR⁺04] some discontinuity indicators, in the context of discontinuous Galerkin methods, are presented and evaluated.

As is natural, a lot of these hybrid formulations are employed to capture discontinuities saving high accuracy and preserving the stability of the method. In the following section will be presented more details about shock capturing.

3.4. Shock capturing and Gibbs phenomenon

One of the most challenging problems in the modeling of compressible flows is the existence of discontinuous solutions, even when smooth initial conditions are employed. It is well known that higher order schemes (with a low numerical diffusion) exhibit non-physical oscillations, wiggles, overshoots and undershoots near to steep gradients and discontinuities, this effect is named *Gibbs phenomenon*.

A generalized way to avoid this problem is to smooth the local discrete solution in the vicinity of discontinuities. This effect is found via addition, explicitly or implicitly, of some sort of extra dissipation into the numerical scheme. Normally, the numeric techniques developed to suppress the Gibbs oscillations are composed by two parts: a *trouble cell indicator*, or *discontinuity indicator*, that identifies the group of cells where the solution should be smoothed, and second, a method to smooth the approximation, or equivalently, to introduce some sort of numerical dissipation. As is natural, to introduce this extra numerical diffusion into the scheme tends to sacrifice the accuracy in the region where it was applied.

3.4.1. Discontinuity indicator

As was previously mentioned, the design of a efficient and precise indicator is important for the success of the numerical technique. Dolejší and Feistauer ([DFS03, DF03]) show that the jumps of the discrete solution through the interfaces are of order $O(h_K)$ in regions with discontinuities and of order $O(h_K^2)$ in zones where the solution is smooth. Thus, these jumps can be used as discontinuity sensors. Using this idea Dolejší and Feistauer propose as discontinuity indicator for the element $K_i \in \mathcal{T}_h$ the expression

$$g(i) = \sum_{j \in S(i)} \int_{\Gamma_{ij}} \frac{\llbracket w_h \rrbracket^2}{h_{K_i} |K_i|^{3/4}} dS, \quad (3.71)$$

for all $i \in I$ and where $|K_i|$ denotes the d-dimensional volume of K_i , $h_{K_i} = \text{diam}(K_i)$ and w_h corresponds to some representative component of the discrete vector solution \mathbf{w}_h . Subsequent works ([FK07, FKP10]) show that this formulation is suitable to detect discontinuities in the density of compressible flows. In these works the elements are marked as trouble cells when

$$g(i) = \sum_{j \in S(i)} \int_{\Gamma_{ij}} \frac{\llbracket \rho_h \rrbracket^2}{h_{K_i} |K_i|^{3/4}} dS > 1. \quad (3.72)$$

Krivodonova ([KXR⁺04]) developed an indicator based on the convergence properties of *DGFEM*. In accordance with this idea the indicator for the edge j , that belongs to element K_i , is defined by

$$g(i) = \sum_{j \in S(i)} \frac{\left| \int_{\Gamma_{ij}} \llbracket w_h \rrbracket dS \right|}{(h_{K_i})^{\frac{p+1}{2}} |\Gamma_{ij}| \|w_h\|}, \quad (3.73)$$

for all $i \in I$ and where $|\Gamma_{ij}|$ denote the $(d - 1)$ -dimensional volume of Γ_{ij} . An complete analysis of different discontinuity indicators can be found in [QS05a].

On the other hand, in the context of the multi-dimensional optimal order detection (*MOOD*) ([DL16, DZLD14, LDD14]) two detection criteria to identify trouble cells are employed: first a physical admissibility criteria, e.g. positivity in pressure and fluid density in the case of compressible flows, and second a numerical admissibility criteria, which normally uses a form of the discrete maximum principle.

3.4.2. Artificial viscosity method

With regard to the method to smooth the discrete solution in the problematic cells, in a coarse sense these methods can be classified in two groups: methods that introduce explicitly an artificial viscosity term and a second group that uses limiters to correct the discrete solution, e.g. the *FCT* method, presented in the previous section, the weighted essentially non-oscillatory schemes *WENO* ([LQ10a, QS05b]), or the *MOOD*-type methods.

The idea about introducing artificial viscosity to stabilize discrete solutions in presence of shock waves was originally presented by von Neumann and Richtmyer ([NR50]), and because of its efficiency and simplicity to be implemented, it has been extensively used, specially in the context of *DGFEM*. This method consists in to add the artificial term

$$\beta_h(\mathbf{w}_h(t), \varphi) = \sum_{i \in I} \int_{K_i} \boldsymbol{\mu}_a \sum_{k=1}^d \frac{\partial \mathbf{w}_h}{\partial x_k} \cdot \frac{\partial \varphi}{\partial x_k} d\mathbf{x} \quad (3.74)$$

to Eq. 3.10, where $\boldsymbol{\mu}_a \in \mathbb{R}^{m,m}$ is, in general, a matrix with the artificial-viscosity coefficients. Feistauer and Kučera ([FK07]) introduced a simple expression for the artificial viscosity based only on the size of the element

$$\beta_h(\mathbf{w}_h(t), \varphi) = \nu_1 \sum_{i \in I} h_{K_i} G(i) \int_{K_i} \sum_{k=1}^d \frac{\partial \mathbf{w}_h}{\partial x_k} \cdot \frac{\partial \varphi}{\partial x_k} d\mathbf{x}, \quad (3.75)$$

where $\nu_1 = O(1)$ and $G(i)$ is a switch-variable that turns equal to 1 when the discontinuity indicator exceeds a threshold and turns to 0 otherwise. In order to strengthen the effect of the neighboring elements into the viscosity term, and inspired on the interior penalty *DGFEM* (see Sec. 3.7), the authors in [FFS03] propose to add an extra term of the form

$$\mathbf{J}_h(\mathbf{w}_h(t), \varphi) = \nu_2 \sum_{i \in I} \sum_{j \in S(i)} \frac{1}{2} (G(i) + G(j)) \int_{\Gamma_{ij}} [[\mathbf{w}_h]] \cdot [[\varphi]] dS, \quad (3.76)$$

with $\nu_2 = O(1)$.

Authors as Hartmann ([Har06, HH02]) and Bassi and Rebay ([BR95]) use the residuals of the discrete solution to estimate the regions where an artificial viscosity should be added and to evaluate the amount of viscosity that should be introduced. According to Hartmann, a variant of Eq. 3.74 is given by

$$\beta_h(\mathbf{w}_h(t), \varphi) = \nu_3 \sum_{i \in I} h_{K_i}^{2-\beta} \int_{K_i} |\operatorname{div}(\mathbb{F}^s(\mathbf{w}_h))| \sum_{k=1}^d \frac{\partial \mathbf{w}_h}{\partial x_k} \cdot \frac{\partial \varphi}{\partial x_k} d\mathbf{x}, \quad (3.77)$$

where $\nu_3 > 0$, $0 < \beta < 1/2$. In a similar fashion, Guermond et al. ([GPP11, ZGMP13]) use the size of the local entropy production to define the intensity of this numeric dissipation. Note that in these last focus, as well as in Eq. 3.77, a switch to activate or deactivate

the shock capturing is not necessary because the stabilization term is applied in the full domain, but regulating locally the amount of numerical viscosity required. Some other formulations for μ_a can be found in ([BD10, Har06]).

For the capturing of strong shocks can be convenient to combine the introduction of the stabilization terms, e.g. Eq. 3.75–Eq. 3.76, with the use of the hybrid numerical flux formulation presented in Eq. 3.70. Thus, in the context of this work, the G -switch variable, activated by the shock indicator, will be incorporated with a double purpose. First, it introduces only in the critical elements the artificial viscosity via stabilization terms Eq. 3.75–Eq. 3.76. Second, if a hybrid numerical flux is used (see Sec. 3.3.4), the value of G is introduced instead of Θ in Eq. 3.70. As result, the lower-order numerical flux \mathbf{H}_{LO} is activated in the same elements where the stabilization terms were added. Here, critical elements should be understood as the cells with discontinuity indicator values upper than a limit value preset.

3.5. Adaptive grid strategy

In order to increase the resolution in the non-smooth regions of the solution a mesh grading is required. The idea of the adaptive mesh strategy is to locally refine cells such that the local error is constant on each element. Here no formal error estimate will be used and instead of we will use a simple and well known idea. According to that, if a field of values $f \in \mathbb{R}$ is to be accurately predicted, then the product between the preferable size of the cell $h_{K_i}^*$ and the field gradient on the cell should be constant for each cell, i.e.,

$$h_{K_i}^* \|\nabla f\|_{L^p(K_i)} \approx C, \quad \text{for } i = 1, \dots, N_c, \quad (3.78)$$

where N_c is the total number of cells in the actual triangulation and we will take $p = 2$. This idea was adopted by Bathe et al. [BZ09] to formulate an adaptive mesh strategy, which inspired the adaptive method used at the present work. Following the idea in [BZ09], the constant C in Eq. 3.78 can be computed as

$$C = \frac{\Lambda_r}{N_c} \sum_{i=1}^{N_c} h_{K_i} \|\nabla f\|_{L^2(K_i)}, \quad (3.79)$$

where h_{K_i} is the size of cell K_i in the actual triangulation and $\Lambda_r \in \mathbb{R}$ is a parameter to control the total cell number in the adapted mesh. Thus, the preferable size for the cell K_i can be calculated as

$$h_{K_i}^* = \frac{\Lambda_r}{N_c \|\nabla f\|_{L^2(K_i)}} \sum_{j=1}^{N_c} h_{K_j} \|\nabla f\|_{L^2(K_j)}. \quad (3.80)$$

In order to prevent extremely small or huge cells, the Eq. 3.80 can be reformulated as

$$h_{K_i}^* = \frac{C}{\max\left(\min\left(\|\nabla f\|_{L^2(K_i)}, \frac{C}{h_{\min}}\right), \frac{C}{h_{\max}}\right)}, \quad (3.81)$$

where h_{\min} and h_{\max} represent the maximum and minimum allowable cell-size values, respectively. Beside, in the practice huge number of cells should be prevent, it is found through the parameter Λ_r , which can be estimated as

$$\Lambda_r = \max\left(\left(\frac{N_c}{N_{c,\max}}\right)^{\frac{1}{d}}, 1\right), \quad (3.82)$$

where $N_{c,\max}$ is a maximum number of cells prescribed.

In the practice, for the scalar field f , quantities as density, pressure, Mach number, or any other meaningful value can be used. Besides, in other cases the vorticity function $\nabla \times \mathbf{v}$ can be employed, instead of ∇f , to improve the resolution of the approximation in the boundary layers [BZ09]. Sometimes only one criterion is not enough to get a good mesh, hence different criteria can be employed and then the several sets of cells to be refined/coarsened can be combined.

Algorithm 1: Adaptive grid strategy employed.

Data: $\mathcal{T}_h, \mathbf{w}_h(\cdot, t_{n-1}), \mathbf{w}_h(\cdot, t_n), N_{c,\max}, r_{L,\max}, r_{L,\min}$
Result: \mathcal{T}_h^*

```

1  $C = 0$ ;
2  $\tilde{\mathbf{w}}_h(\cdot) = 2\mathbf{w}_h(\cdot, t_n) - \mathbf{w}_h(\cdot, t_{n-1})$ ;
3 for  $i = 1 : N_c$  do
4    $C = C + h_{K_i} \|f(\tilde{\mathbf{w}}_h)\|_{L^2(K_i)}$ ;
5 end
6  $\Lambda_r = \max\left(\left(\frac{N_c}{N_{c,\max}}\right)^{\frac{1}{d}}, 1\right)$ ;
7  $C = \Lambda_r \frac{C}{N_c}$ ;
8 for  $j = 1 : r_{L,\max}$  do
9   for  $i = 1 : N_c$  do
10     $h_{K_i}^* = \frac{C}{\|f(\tilde{\mathbf{w}}_h)\|_{L^2(K_i)}}$ ;
11    if  $(h_{K_i} > h_{K_i}^*) \wedge (r_L < r_{L,\max})$  then
12       $K_i \rightarrow$  mark cell to be refined
13    else if  $(h_{K_i} < h_{K_i}^*) \wedge (r_L > r_{L,\min})$  then
14       $K_i \rightarrow$  mark cell to be coarsened
15    end
16  end
17  execute coarsening and refinement to get  $\mathcal{T}_h^*$ ;
18  project  $\tilde{\mathbf{w}}_h(\cdot)$  onto  $\mathcal{T}_h^*$ ;
19 end

```

For the implementation of this strategy, the direct control of the size of the each element is not the easiest way. Instead of that, the preferable size in Eq. 3.80 can be used as a refinement indicator. Then, if the actual size of K_i is larger than $h_{K_i}^*$ then the element should be refined, conversely if $h_{K_i} < h_{K_i}^*$ then the cell should be coarsened. In the context of this work the library `deal.ii` [BHK07] is used to handle the triangulation, hence to refine a cell should be understood as to split it in four new elements one refinement level upper. Likewise, to coarsen cells here mean to merge four adjacent cells to create a new element one refinement level lower. By default all cells into the initial triangulation have a refinement level $r_L = 0$.

In this form, eventually the value of $h_{K_i}^*$ can be too much smaller than the current cell-size, and then a single refinement is not enough to find the preferable size. A similar effect can happen with too small cells that should be coarsened. Hence, a iterative procedure of coarsening/refinement can be used to close the current cell-size to $h_{K_i}^*$.

In order to prevent very small or large cells, the idea in the Eq. 3.81 can be reformulated restricting the refinement level of each cell (r_L) according to a maximum ($r_{L,\max}$) and minimum ($r_{L,\min}$) levels prescribed for the triangulation. Further, if something about the behavior of the solution is known a priori, then different values of $r_{L,\max}$ and $r_{L,\min}$ can be defined along the domain.

Finally, in order to improve this strategy, the quantity (or quantities) used as criterion for the adaptation is computed using an estimation of the solution in the next time step ($t_{n+1} = t_n + \tau$). In this work the simple predictor

$$\begin{aligned} \mathbf{w}_h(\mathbf{x}, t_{n+1}) &\approx \mathbf{w}_h(\mathbf{x}, t_n) + \tau \frac{\partial \mathbf{w}_h}{\partial t} \\ &\approx \mathbf{w}_h(\mathbf{x}, t_n) + \tau \frac{\mathbf{w}_h(\mathbf{x}, t_n) - \mathbf{w}_h(\mathbf{x}, t_{n-1})}{\tau} \\ &= 2\mathbf{w}_h(\mathbf{x}, t_n) - \mathbf{w}_h(\mathbf{x}, t_{n-1}) \end{aligned} \quad (3.83)$$

is used.

The Algorithm 1 summarizes the adaptive grid strategy implemented here.

3.6. Inviscid boundary conditions

As was mentioned in Sec. 2.3, for the inviscid problem the boundary condition should represent the physical behavior of the fluid in the borders, but in addition should be in accordance with the hyperbolic character of the system in Eq. 2.40.

As was reported by Bazilevs and Hughes ([BH07]) that strongly enforced Dirichlet boundary conditions can produce spurious oscillations in the vicinity of inlet/outlet borders. Conversely, a weak imposition of the boundary conditions avoids this non-physical oscillations and is even able to improve the accuracy in the case of boundary layers in Navier–Stokes problems. A consequence of using *DGFEM* is that the weak imposition of Dirichlet boundary condition can be done in a natural way.

Thus, using the definitions of Sec. 3.1 and the discontinuous approximation *DGFEM*, the inviscid boundary conditions can be imposed weakly via the numerical flux, i.e. using the flux integral in Eq. 3.12

$$\int_{\Gamma_{ij}} \mathbf{H}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}) \cdot \boldsymbol{\varphi} dS, \quad (3.84)$$

with $\mathbf{w}^- = \mathbf{w}|_{\Gamma_{ji}}$ and $\Gamma_{ji} \subset \partial\Omega_h$, i.e., for all $j \in \kappa(i)$ and $i \in I$.

In order to consider different kinds of physical boundary conditions, $\partial\Omega_h$ will be represented as

$$\partial\Omega_h = \Gamma_{\text{IN}}^{\text{sup}} \cup \Gamma_{\text{IN}}^{\text{sub}} \cup \Gamma_{\text{O}}^{\text{sup}} \cup \Gamma_{\text{O}}^{\text{sub}} \cup \Gamma_{\text{W}}, \quad (3.85)$$

where $\Gamma_{\text{IN/O}}^{\text{sup}}$ represent the supersonic inlet/outlet boundaries, $\Gamma_{\text{IN/O}}^{\text{sub}}$ are the subsonic inlet/outlet borders and Γ_{W} represents the impermeable walls.

3.6.1. Inlet-outlet boundary condition

At the inlet/outlet borders it is necessary to use non-reflecting boundary conditions in order to avoid that acoustic waves propagate back from these boundaries into the domain. A way to avoid this effect is to use a formulation for boundary conditions based on the local characteristics of the system ([FKP10, FK07]).

3.6.1.1. Subsonic inlet/outlet boundary conditions

Defining a coordinate system $\bar{\mathbf{x}}$, with origin in $\mathbf{x} \in \partial\Omega_h$ and \bar{x}_1 -direction parallel to the unit normal-boundary vector \mathbf{n} , a rotation is applied to the conservative variable vector \mathbf{w} in order to express it according to the local coordinate system $\bar{\mathbf{x}}$, i.e.

$$\mathbf{q} = \mathbf{Q}(\mathbf{n}) \mathbf{w}, \quad (3.86)$$

where $\mathbf{Q}(\mathbf{n})$ is the rotation matrix defined in Eq. 2.80, and

$$\mathbf{n} = \begin{pmatrix} \cos \alpha \cos \beta & \sin \alpha \cos \beta & \sin \beta \end{pmatrix}^T. \quad (3.87)$$

From using the rotational invariance property of the Euler equations (Sec. 2.2.4), it follows that it is possible to write the hyperbolic system as

$$\frac{\partial \mathbf{q}}{\partial t} + \mathbb{A}_1(\mathbf{q}_{ij}) \frac{\partial \mathbf{q}}{\partial \bar{x}_1} = 0 \quad \text{in } -\infty < \bar{x}_1 < \infty, \quad (3.88)$$

equipped with the boundary condition

$$\mathbf{q}(0, t) = \mathbf{q}_{ji} \quad \text{in } t > 0 \quad (3.89)$$

and the initial condition

$$\mathbf{q}(\bar{x}_1, 0) = \mathbf{q}_{ij} \quad \text{on } \bar{x}_1 < 0, \quad (3.90)$$

where $\mathbf{q}_{ij} = \mathbf{Q}(\mathbf{n}) \mathbf{w}^+$. Thus, to impose a numerically suitable boundary condition is equivalent to find an adequate value for \mathbf{q}_{ji} such that the problem Eq. 3.88–Eq. 3.90 has a unique solution.

If λ_s and \mathbf{r}_s represent the eigenvalues and eigenvector of the matrix $\mathbb{A}_1(\mathbf{q}_{ij})$ (see Lemma 2.2.6) and $\mathbf{q}_{ji}^* = \mathbf{Q}(\mathbf{n}) \mathbf{w}^D$, where \mathbf{w}^D is a prescribed boundary state in the inlet/outlet border, the vectors \mathbf{q}_{ij} and \mathbf{q}_{ji}^* can be written using spectral decomposition as follows

$$\mathbf{q}_{ij} = \sum_{s=1}^m \alpha_s \mathbf{r}_s, \quad (3.91)$$

$$\mathbf{q}_{ji}^* = \sum_{s=1}^m \beta_s \mathbf{r}_s, \quad (3.92)$$

and the vector state \mathbf{q}_{ji} can be expressed as a combination of \mathbf{q}_{ij} and \mathbf{q}_{ji}^* based on the characteristic speed λ_s

$$\mathbf{q}_{ji} = \sum_{s=1}^m \gamma_s \mathbf{r}_s, \quad \text{with } \gamma_s = \begin{cases} \alpha_s, & \lambda_s \geq 0, \\ \beta_s, & \lambda_s < 0. \end{cases} \quad (3.93)$$

Finally, the vector state \mathbf{w}^- is computed as

$$\mathbf{w}^- = \mathbf{w}|_{\Gamma_{ji}} = \mathbf{Q}^{-1}(\mathbf{n}) \mathbf{q}_{ji} \quad \text{on } \Gamma_{\text{IN/O}}^{\text{sub}}. \quad (3.94)$$

3.6.1.2. Supersonic inlet/outlet boundary conditions

According to the theory of characteristics in supersonic inlet boundaries all the eigenvalues of $\mathbf{P}(\mathbf{w}, \mathbf{n})$ are negative (see Eq. 2.54), this means that all the information is carried from the boundary toward the interior, and then the complete vector state $\mathbf{w}^- = \mathbf{w}|_{\Gamma_{ji}}$ should correspond to the prescribed boundary values, i.e.,

$$\mathbf{w}^- = \mathbf{w}^D \quad \text{on } \Gamma_{\text{IN}}^{\text{sup}}. \quad (3.95)$$

On the other hand, at supersonic outlet boundaries all the eigenvalues of $\mathbf{P}(\mathbf{w}, \mathbf{n})$ are positive, and therefore the boundary conditions should be extrapolated from the flowfield, i.e.,

$$\mathbf{w}^- = \mathbf{w}^+ \quad \text{on } \Gamma_{\text{O}}^{\text{sup}}. \quad (3.96)$$

3.6.2. Impermeable wall boundary condition

The wall condition for the inviscid case corresponds to a *slipping boundary condition*, also named *reflecting boundary condition*. This condition is built using

$$\mathbf{v}|_{\Gamma_{ji}} = \mathbf{v}|_{\Gamma_{ij}} - 2 \left(\mathbf{v}|_{\Gamma_{ij}} \cdot \mathbf{n} \right) \mathbf{n},$$

or equivalently

$$\mathbf{w}^- = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 - 2n_1^2 & -2n_1n_2 & -2n_1n_3 & 0 \\ 0 & -2n_1n_2 & 1 - 2n_2^2 & -2n_2n_3 & 0 \\ 0 & -2n_1n_3 & -2n_2n_3 & 1 - 2n_3^2 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{w}^+ \quad \text{on } \Gamma_w. \quad (3.97)$$

3.7. Numerical treatment of viscous terms

3.7.1. Scalar convection-diffusion problem

Considering the scalar convection-diffusion problem

$$\frac{\partial w}{\partial t} + \sum_{k=1}^d \frac{\partial \mathbf{f}_k^s}{\partial x_k}(w) = \nu \Delta w \quad \text{in } \mathcal{M} = \Omega \times (0, T), \quad (3.98)$$

subject to the boundary and initial conditions

$$\begin{aligned} w &= w^D && \text{on } \Gamma^D, \\ \nu \nabla w \cdot \mathbf{n} &= q_n && \text{on } \Gamma^N, \end{aligned} \quad (3.99)$$

and

$$w(x, 0) = w^0(x), \quad \text{with } x \in \Omega, \quad (3.100)$$

where $\nu \in \mathbb{R}_{>0}$, $\mathbf{f}^s = (\mathbf{f}_k^s)_k : w \rightarrow \mathbb{R}^d$, $\partial\Omega = \Gamma^D \cup \Gamma^N$, $w^D : \Gamma^D \times (0, T) \rightarrow \mathbb{R}$ and $q_n : \Gamma^N \times (0, T) \rightarrow \mathbb{R}$.

In order to solve the convection-diffusion problem using a *DGFEM* spatial discretization it is possible to split the Eq. 3.98 into two coupled equations ([Coc99])

$$\left. \begin{aligned} \frac{\partial w}{\partial t} + \sum_{k=1}^d \frac{\partial \mathbf{f}_k^s(w)}{\partial x_k} &= \sum_{k=1}^d \frac{\partial q_k(w)}{\partial x_k} \\ \mathbf{q} + \nu \nabla w &= 0, \end{aligned} \right\} \quad \text{in } \mathcal{M} = \Omega \times (0, T), \quad (3.101)$$

where $\mathbf{q} = (q_k)_k \in \mathbb{R}^d$. Nevertheless this change converts the scalar differential equation Eq. 3.98 into a system of differential equations, and therefore new degrees of freedom have to be introduced for the numerical solution.

The *interior penalty method (IPM)*, originally proposed by Arnold ([Arn82]), can be used to formulate a discrete solution of the convection-diffusion problem Eq. 3.98 using discontinuous elements.

Assuming $w(\cdot, t) \in H^2(\Omega)$, multiplying the equation Eq. 3.98 by $\varphi \in H^2(\Omega, \mathcal{T}_h)$, integrating by parts and using the identity

$$\llbracket (\nabla w) \varphi \rrbracket = \{\nabla w\} \llbracket \varphi \rrbracket + \llbracket \nabla w \rrbracket \{\varphi\},$$

it is possible to obtain the weak formulation

$$\frac{d}{dt} (w, \varphi)_h + \mathbf{b}_h(w, \varphi) + \mathbf{a}_h(w, \varphi) = 0, \quad (3.102)$$

with

$$(w, \varphi)_h = \sum_{i \in I} \int_{K_i} w \varphi \, dx, \quad (3.103)$$

$$\mathbf{b}_h(w, \varphi) = - \sum_{i \in I} \int_{K_i} \sum_{k=1}^d f_k^s(w) \frac{\partial \varphi}{\partial x_k} \, dx + \sum_{i \in I} \sum_{j \in S(i)} \int_{\Gamma_{ij}} \sum_{k=1}^d f_k^s(w) n_k \varphi \, dS, \quad (3.104)$$

and

$$\mathbf{a}_h(w, \varphi) = \sum_{i \in I} \int_{K_i} \nu \nabla w \cdot \nabla \varphi \, dx - \sum_{i \in I} \sum_{j \in S(i)} \int_{\Gamma_{ij}} \nu \{\nabla w\} \cdot \mathbf{n} \llbracket \varphi \rrbracket \, dS. \quad (3.105)$$

Thus, in order to preserve the symmetry property in the viscous bilinear operator $\mathbf{a}_h(w, \varphi)$, the term

$$- \sum_{i \in I} \sum_{j \in S(i)} \int_{\Gamma_{ij}} \nu \{\nabla \varphi\} \cdot \mathbf{n} \llbracket w \rrbracket \, dS \quad (3.106)$$

can be added, without losing consistency, to get

$$\begin{aligned} \mathbf{a}_h^{\text{sym}}(w, \varphi) &= \sum_{i \in I} \int_{K_i} \nu \nabla w \cdot \nabla \varphi \, dx \\ &\quad - \sum_{i \in I} \sum_{j \in S(i)} \int_{\Gamma_{ij}} \nu (\{\nabla w\} \cdot \mathbf{n} \llbracket \varphi \rrbracket + \{\nabla \varphi\} \cdot \mathbf{n} \llbracket w \rrbracket) \, dS. \end{aligned} \quad (3.107)$$

Now, considering that ([PE11])

$$\mathbf{a}_h^{\text{sym}}(w_h, w_h) = \|\nabla w_h\|_{L^2(\Omega)}^2 - 2 \sum_{i \in I} \sum_{j \in S(i)} \int_{\Gamma_{ij}} \{\nabla w_h\} \cdot \mathbf{n} \llbracket w_h \rrbracket \, dS \quad (3.108)$$

for all $w_h \in H^2(\Omega, \mathcal{T}_h)$, the discrete coercivity in the operator $\mathbf{a}_h^{\text{sym}}$ can be recovered with the addition of a sufficient larger term. Thus, the interface penalty term

$$\sum_{i \in I} \sum_{j \in S(i)} \int_{\Gamma_{ij}} \frac{\sigma}{|\Gamma_{ij}|} \llbracket w \rrbracket \llbracket \varphi \rrbracket \, dS \quad \text{with } \sigma \in \mathbb{R}_{>0}, \quad (3.109)$$

is introduced into $\mathbf{a}_h^{\text{sym}}$ to establish the coercivity property of the operator. This numerical formulation is named *symmetric interior penalty Galerkin method (SIPG)*.

A non-symmetric formulation can be found changing the sign of the term in Eq. 3.106, recovering thus the coercivity in the viscous operator for any size of the coefficient σ . This version is usually named *non-symmetric interior penalty Galerkin method (NIPG)*.

If the integrals on boundary faces are written explicitly, the complete weak formulation is

$$\frac{d}{dt} (w, \varphi)_h + \mathbf{b}_h(w, \varphi) + \mathbf{a}_h^{\text{IPM}}(w, \varphi) + \mathbf{J}_h(w, \varphi) = \mathbf{l}_h(w, \varphi), \quad (3.110)$$

where

$$\begin{aligned} \mathbf{a}_h^{\text{IPM}}(w, \varphi) &= \sum_{i \in I} \int_{K_i} \nu \nabla w \cdot \nabla \varphi \, dx \\ &\quad - \sum_{i \in I} \sum_{j \in S(i)} \int_{\Gamma_{ij}} \nu \{\nabla w\} \cdot \mathbf{n} \llbracket \varphi \rrbracket \, dS \pm \sum_{i \in I} \sum_{j \in S(i)} \int_{\Gamma_{ij}} \nu \{\nabla \varphi\} \cdot \mathbf{n} \llbracket w \rrbracket \, dS \\ &\quad - \sum_{i \in I} \sum_{j \in \kappa^D(i)} \int_{\Gamma_{ij}} \nu \nabla w \cdot \mathbf{n} \varphi \, dS \pm \sum_{i \in I} \sum_{j \in \kappa^D(i)} \int_{\Gamma_{ij}} \nu \nabla \varphi \cdot \mathbf{n} w \, dS, \end{aligned} \quad (3.111)$$

$$\mathbf{J}_h(w, \varphi) = \sum_{i \in I} \sum_{j \in s(i)} \int_{\Gamma_{ij}} \frac{\sigma}{|\Gamma_{ij}|} \llbracket w \rrbracket \llbracket \varphi \rrbracket dS + \sum_{i \in I} \sum_{j \in \kappa^D(i)} \int_{\Gamma_{ij}} \frac{\sigma}{|\Gamma_{ij}|} w \varphi dS, \quad (3.112)$$

$$\begin{aligned} \mathbf{l}_h(\varphi) = & \sum_{i \in I} \sum_{j \in \kappa^N(i)} \int_{\Gamma_{ij}} q_n \varphi dS \pm \sum_{i \in I} \sum_{j \in \kappa^D(i)} \int_{\Gamma_{ij}} \nu \nabla \varphi \cdot \mathbf{n} w^D dS \\ & + \sum_{i \in I} \sum_{j \in \kappa^D(i)} \int_{\Gamma_{ij}} \frac{\sigma}{|\Gamma_{ij}|} w^D \varphi dS \end{aligned} \quad (3.113)$$

and $\kappa(i) = \kappa^D(i) \cup \kappa^N(i)$. In Eq. 3.111 and Eq. 3.113 select the $-$ sign for *SIPG* and $+$ sign for non-symmetric formulation (*NIPG*).

3.7.2. Interior penalty method applied to Navier–Stokes problem

The *IPM* have been implemented by different authors in the context of the numerical solution of Navier–Stokes equations via *DGFEM* ([PE11, HH06a, HH06b, FFS03]). The extension from the *IPM*, presented in Sec. 3.7.1 for linear convection–diffusion, to Navier–Stokes equations is straightforward.

Recalling the Navier–Stokes system in Eq. 2.89, multiplying by $\varphi \in [H^2(\Omega, \mathcal{T}_h)]^m$, integrating by parts and operating in a similar fashion to Sec. 3.7.1, it is possible to write the discrete weak formulation for this problem in the form

$$\frac{d}{dt} (\mathbf{w}_h(t), \varphi)_h = \beta_h(\mathbf{w}_h(t), \varphi) - \mathbf{b}_h(\mathbf{w}_h(t), \varphi) - \mathbf{a}_h(\mathbf{w}_h(t), \varphi) - \mathbf{J}_h(\mathbf{w}_h(t), \varphi), \quad (3.114)$$

with $\mathbf{w}_h(\mathbf{x}, t) \in \mathbf{S}_h$. Here,

$$(\mathbf{w}_h(t), \varphi)_h = \sum_{i \in I} \int_{K_i} \mathbf{w}_h \cdot \varphi d\mathbf{x}, \quad (3.115)$$

$$\begin{aligned} \mathbf{b}_h(\mathbf{w}_h(t), \varphi) = & - \sum_{i \in I} \int_{K_i} \sum_{k=1}^d \mathbb{F}_k^s(\mathbf{w}_h) \cdot \frac{\partial \varphi}{\partial x_k} d\mathbf{x} \\ & + \sum_{i \in I} \sum_{j \in s(i)} \int_{\Gamma_{ij}} \mathbf{H}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n}) \cdot \varphi dS, \end{aligned} \quad (3.116)$$

$$\begin{aligned} \mathbf{a}_h(\mathbf{w}_h(t), \varphi) = & \sum_{i \in I} \int_{K_i} \sum_{k=1}^d \mathbb{F}_k^v(\mathbf{w}_h, \nabla \mathbf{w}_h) \cdot \frac{\partial \varphi}{\partial x_k} d\mathbf{x} \\ & - \sum_{i \in I} \sum_{j \in s(i)} \int_{\Gamma_{ij}} \sum_{k=1}^d \{ \mathbb{F}_k^v(\mathbf{w}_h, \nabla \mathbf{w}_h) \} n_k \cdot \llbracket \varphi \rrbracket dS \\ & \pm \sum_{i \in I} \sum_{j \in s(i)} \int_{\Gamma_{ij}} \sum_{k=1}^d \{ \mathbb{F}_k^v(\mathbf{w}_h, \nabla \varphi) \} n_k \cdot \llbracket \mathbf{w} \rrbracket dS \\ & - \sum_{i \in I} \sum_{j \in \kappa^D(i)} \int_{\Gamma_{ij}} \sum_{k=1}^d \mathbb{F}_k^v(\mathbf{w}_h, \nabla \mathbf{w}_h) n_k \cdot \varphi dS \\ & \pm \sum_{i \in I} \sum_{j \in \kappa^D(i)} \int_{\Gamma_{ij}} \sum_{k=1}^d \mathbb{F}_k^v(\mathbf{w}_h, \nabla \varphi) n_k \cdot \mathbf{w} dS, \end{aligned} \quad (3.117)$$

$$\mathbf{J}_h(\mathbf{w}_h(t), \varphi) = \sum_{i \in I} \sum_{j \in s(i)} \int_{\Gamma_{ij}} \frac{\sigma}{|\Gamma_{ij}|} [\mathbf{w}_h] \cdot [\boldsymbol{\varphi}] dS + \sum_{i \in I} \sum_{j \in \kappa^D(i)} \int_{\Gamma_{ij}} \frac{\sigma}{|\Gamma_{ij}|} \mathbf{w}_h \cdot \boldsymbol{\varphi} dS \quad (3.118)$$

and

$$\begin{aligned} \beta_h(\mathbf{w}_h(t), \varphi) &= \sum_{i \in I} \int_{K_i} \mathbf{F}(\mathbf{w}_h) \cdot \boldsymbol{\varphi} d\mathbf{x} \\ &\pm \sum_{i \in I} \sum_{j \in \kappa^D(i)} \int_{\Gamma_{ij}} \sum_{k=1}^d \mathbb{F}_k^v(\mathbf{w}_h, \nabla \boldsymbol{\varphi}) n_k \cdot \mathbf{w}^D dS \\ &+ \sum_{i \in I} \sum_{j \in \kappa^D(i)} \int_{\Gamma_{ij}} \frac{\sigma}{|\Gamma_{ij}|} \mathbf{w}^D \cdot \boldsymbol{\varphi} dS. \end{aligned} \quad (3.119)$$

3.7.3. Implementation of the viscous flux

In order to implement computationally the viscous terms $\mathbb{F}_k^v(\mathbf{w}, \nabla \mathbf{w})$, $k = 1, 2, 3$, the viscous flux expression in Eq. 2.16 should be written in terms of the conservative variables \mathbf{w} defined in Eq. 2.14. If the viscous stress $\boldsymbol{\tau}$ is written as

$$\tau_{ij} = \lambda \operatorname{div} \mathbf{v} \delta_{ij} + \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right), \quad \text{for } i, j = 1, \dots, d, \quad (3.120)$$

then introducing the relation

$$\frac{\partial v_i}{\partial x_j} = \frac{1}{w_1} \left(\frac{\partial w_{i+1}}{\partial x_j} - \frac{w_{i+1}}{w_1} \frac{\partial w_1}{\partial x_j} \right) \quad (3.121)$$

into Eq. 3.120, all τ_{ij} stress in Eq. 2.16 are expressed in terms of \mathbf{w} . In the same way, the derivative $\partial \theta / \partial x_i$ in Eq. 2.16 should also be expressed in terms of \mathbf{w} . However, when a real gas formulation is been used, θ normally cannot be written explicitly in terms of \mathbf{w} , then we use the relation

$$\frac{\partial \theta}{\partial x_i} = \frac{1}{\partial e / \partial \theta} \left(\frac{\partial e}{\partial x_i} - \frac{\partial e}{\partial \rho} \frac{\partial \rho}{\partial x_i} \right) \quad (3.122)$$

and the total energy definition in Eq. 2.9

$$e = \frac{E}{\rho} - \frac{1}{2} |\mathbf{v}|^2. \quad (3.123)$$

The internal energy can be written in terms of \mathbf{w} as

$$e = \frac{w_m}{w_1} - \frac{1}{2w_1^2} \sum_{j=1}^d w_{j+1}^2, \quad (3.124)$$

where, as previously, d is the number of spatial dimensions and $m = d + 2$ is the number of components in \mathbf{w} . Thus, the derivative $\frac{\partial e}{\partial x_i}$ can be written as

$$\frac{\partial e}{\partial x_i} = \frac{1}{w_1} \left(\frac{\partial w_m}{\partial x_i} - \frac{w_m}{w_1} \frac{\partial w_1}{\partial x_i} - \frac{1}{w_1} \sum_{j=1}^d w_{j+1} \frac{\partial w_{j+1}}{\partial x_i} + \frac{1}{w_1^2} \frac{\partial w_1}{\partial x_i} \sum_{j=1}^d w_{j+1}^2 \right). \quad (3.125)$$

Now, Eq. 3.125 can be replaced in Eq. 3.122 to obtain

$$\begin{aligned} \frac{\partial \theta}{\partial x_i} &= \frac{1}{e_\theta w_1} \left(\frac{\partial w_m}{\partial x_i} - \frac{w_m}{w_1} \frac{\partial w_1}{\partial x_i} - \frac{1}{w_1} \sum_{j=1}^d w_{j+1} \frac{\partial w_{j+1}}{\partial x_i} \right. \\ &\quad \left. + \frac{1}{w_1^2} \frac{\partial w_1}{\partial x_i} \sum_{j=1}^d w_{j+1}^2 \right) - \frac{\partial w_1}{\partial x_i} \frac{e_\rho}{e_\theta}. \end{aligned} \quad (3.126)$$

Then, for example with $d = 2$, the flux terms $\mathbb{F}_1^v(\mathbf{w}, \nabla \mathbf{w})$ and $\mathbb{F}_2^v(\mathbf{w}, \nabla \mathbf{w})$ are given by

$$\mathbb{F}_1^v(\mathbf{w}, \nabla \mathbf{w}) = \begin{pmatrix} 0 \\ \frac{2}{3} \frac{\mu}{w_1} \left(2 \left(\frac{\partial w_2}{\partial x_1} - \frac{w_2}{w_1} \frac{\partial w_1}{\partial x_1} \right) - \left(\frac{\partial w_3}{\partial x_2} - \frac{w_3}{w_1} \frac{\partial w_1}{\partial x_2} \right) \right) \\ \frac{\mu}{w_1} \left(\left(\frac{\partial w_3}{\partial x_1} - \frac{w_3}{w_1} \frac{\partial w_1}{\partial x_1} \right) + \left(\frac{\partial w_2}{\partial x_2} - \frac{w_2}{w_1} \frac{\partial w_1}{\partial x_2} \right) \right) \\ \frac{w_2}{w_1} \left(\mathbb{F}_1^v(\mathbf{w}, \nabla \mathbf{w}) \right)_2 + \frac{w_3}{w_1} \left(\mathbb{F}_1^v(\mathbf{w}, \nabla \mathbf{w}) \right)_3 + \frac{k}{e_\theta w_1} \left(\frac{\partial w_4}{\partial x_1} - \frac{w_4}{w_1} \frac{\partial w_1}{\partial x_1} \dots \right. \\ \left. \dots - \frac{1}{w_1} \left(w_2 \frac{\partial w_2}{\partial x_1} + w_3 \frac{\partial w_3}{\partial x_1} \right) + \frac{1}{w_1^2} (w_2^2 + w_3^2) \frac{\partial w_1}{\partial x_1} \right) - \frac{e_\rho}{e_\theta} \frac{\partial w_1}{\partial x_1} \end{pmatrix} \quad (3.127)$$

and

$$\mathbb{F}_2^v(\mathbf{w}, \nabla \mathbf{w}) = \begin{pmatrix} 0 \\ \frac{\mu}{w_1} \left(\left(\frac{\partial w_3}{\partial x_1} - \frac{w_3}{w_1} \frac{\partial w_1}{\partial x_1} \right) + \left(\frac{\partial w_2}{\partial x_2} - \frac{w_2}{w_1} \frac{\partial w_1}{\partial x_2} \right) \right) \\ \frac{2}{3} \frac{\mu}{w_1} \left(2 \left(\frac{\partial w_3}{\partial x_2} - \frac{w_3}{w_1} \frac{\partial w_1}{\partial x_2} \right) - \left(\frac{\partial w_2}{\partial x_1} - \frac{w_2}{w_1} \frac{\partial w_1}{\partial x_1} \right) \right) \\ \frac{w_2}{w_1} \left(\mathbb{F}_2^v(\mathbf{w}, \nabla \mathbf{w}) \right)_2 + \frac{w_3}{w_1} \left(\mathbb{F}_2^v(\mathbf{w}, \nabla \mathbf{w}) \right)_3 + \frac{k}{e_\theta w_1} \left(\frac{\partial w_4}{\partial x_2} - \frac{w_4}{w_1} \frac{\partial w_1}{\partial x_2} \dots \right. \\ \left. \dots - \frac{1}{w_1} \left(w_2 \frac{\partial w_2}{\partial x_2} + w_3 \frac{\partial w_3}{\partial x_2} \right) + \frac{1}{w_1^2} (w_2^2 + w_3^2) \frac{\partial w_1}{\partial x_2} \right) - \frac{e_\rho}{e_\theta} \frac{\partial w_1}{\partial x_2} \end{pmatrix}, \quad (3.128)$$

where $\left(\mathbb{F}_k^v(\mathbf{w}, \nabla \mathbf{w}) \right)_i$ represents the i -th component of $\mathbb{F}_k^v(\mathbf{w}, \nabla \mathbf{w})$.

Additionally, the derivatives $e_\theta = \partial e / \partial \theta$ and $e_\rho = \partial e / \partial \rho$ are calculated using the EOS. For the ideal gas assumption $e = C_v \theta$ ($e_\rho = 0$ and $e_\theta = C_v$) and the expressions for $\mathbb{F}_k^v(\mathbf{w}, \nabla \mathbf{w})$ used in [FFS03] are obtained.

In this work the thermodynamic behavior of CO₂ is modeled by the *egB-EOS* (Eq. 2.99) and the derivatives e_θ and e_ρ , used in Eq. 3.126–Eq. 3.128, are calculated analytically from Eq. 2.107. In Sec. A the expressions for these derivatives are included.

3.7.4. Viscous boundary conditions

In dependence on the physical nature of the boundary, different sets of values should be prescribed as boundary conditions in the vector \mathbf{w}^D . Here, the idea about imposition of boundary condition used by several authors, as Feistauer et al. [FFS03] and Hartmann et al. [HH02, Har06], is employed in the present work. As previous, $\mathbf{w}^+ = \mathbf{w}_{\Gamma_{ij}}$.

3.7.4.1. Subsonic inlet boundary condition

An option for the treatment of the subsonic inlet-boundaries, which is used in the present work, is to prescribe the boundary values for pressure p^D , temperature θ^D and velocity

\mathbf{v}^D . In this case the vector \mathbf{w}^D is defined by

$$\mathbf{w}^D = \begin{pmatrix} \bar{\rho} \\ \bar{\rho}v_1^D \\ \bar{\rho}v_2^D \\ \bar{\rho}v_3^D \\ \bar{e} + \frac{1}{2}\bar{\rho}|\mathbf{v}^D|^2 \end{pmatrix} \quad \text{on } \Gamma_{\text{IN}}^{\text{sub}}. \quad (3.129)$$

Here, the values of density ($\bar{\rho}$) and energy density (\bar{e}) correspond to the thermodynamic state defined by p^D - θ^D . For ideal gas these values are computed as

$$\bar{\rho} = \frac{p^D}{(\gamma - 1) C_v \theta^D} \quad (3.130)$$

and

$$\bar{e} = \frac{p^D}{\gamma - 1}. \quad (3.131)$$

For cases where CO₂ and real gas assumption are used, the values of \bar{e} and $\bar{\rho}$ are calculated employing the library PROPATH [PRO08], which offers a good accuracy in the computation of the thermodynamic properties of different substances, and particularly for the CO₂ at states above the triple point.

3.7.4.2. Subsonic outlet boundary condition

On the subsonic outlet boundary the vector \mathbf{w}^D , for $d = 3$, is written as

$$\mathbf{w}^D = \begin{pmatrix} w_1^+ \\ w_2^+ \\ w_3^+ \\ w_4^+ \\ \bar{e} + \frac{1}{2w_1^+} \sum_{i=1}^3 (w_{i+1}^+)^2 \end{pmatrix} \quad \text{on } \Gamma_{\text{O}}^{\text{sub}}, \quad (3.132)$$

where the energy density \bar{e} for ideal gas can be calculated using Eq. 3.131. For real gases \bar{e} is defined for a thermodynamic state, then in this case pressure p^D and temperature θ^D are prescribed on $\Gamma_{\text{O}}^{\text{sub}}$. In this work the values of \bar{e} for CO₂, at state defined by p^D and θ^D , are computed using the PROPATH library.

Additionally, this condition is complemented with a zero normal component for the stress tensor and for the heat flux, i.e.,

$$\left. \begin{array}{l} \sum_{k=1}^d \tau_{kj} n_k = 0 \quad j = 1, \dots, d, \\ \nabla\theta \cdot \mathbf{n} = 0 \end{array} \right\} \quad \text{on } \Gamma_{\text{O}}^{\text{sub}}. \quad (3.133)$$

3.7.4.3. Supersonic inlet boundary condition

According to the theory of characteristics, and in the same fashion as in the case of inviscid boundary conditions, on the supersonic inlet flow all the conservative variables should be prescribed. Thus, on this border we define values for velocity \mathbf{v}^D , temperature θ^D and pressure p^D , and for computing the vector \mathbf{w}^D we proceed as in Sec. 3.7.4.1.

3.7.4.4. Supersonic outlet boundary condition

On the supersonic outlet no value should be prescribed. Nevertheless, in practice the conditions in Eq. 3.133 are frequently incorporated into the viscous flux at this border.

3.7.4.5. No-slip wall boundary condition

Unlike the slip boundary condition, this condition imposes a zero velocity on the wall (not only in the normal component). In the case of an adiabatic wall, besides the null-velocity terms, the other components in the state vector \mathbf{w}^D are taken from \mathbf{w}^+

$$\mathbf{w}^D = \begin{pmatrix} w_1^+ \\ 0 \\ 0 \\ 0 \\ w_5^+ \end{pmatrix} \quad \text{on } \Gamma_{\text{W}}^{\text{adia}}. \quad (3.134)$$

Additionally, because the adiabatic condition on the wall, the heat conduction term $(\nabla\theta \cdot \mathbf{n})$ in \mathbb{F}^v should be set to zero.

In the case of an isothermal wall, the internal energy is computed using the prescribed temperature θ^D , i.e., for $d = 3$ the vector \mathbf{w}^D is defined by

$$\mathbf{w}^D = \begin{pmatrix} w_1^+ \\ 0 \\ 0 \\ 0 \\ w_1^+ \tilde{e} + \frac{1}{2} w_1^+ |\mathbf{v}^+|^2 \end{pmatrix} \quad \text{on } \Gamma_{\text{W}}^{\text{iso}}, \quad (3.135)$$

where, for ideal gas \tilde{e} is computed as

$$\tilde{e} = C_v \theta,$$

and in the real gas case, the expression for specific internal energy, derived from the EOS (see Eq. 2.107), is used. For the CO_2 the specific internal energy, in terms of density and temperature, is included in Sec. A, and \tilde{e} is calculated as

$$\tilde{e} = e(\mathbf{w}_1^+, \theta^D).$$

4. Time integration

Recalling the semidiscrete weak formulation for the Navier–Stokes equations written in Eq. 3.114, and considering that the approximation in one cell $\mathbf{w}_h \in C^1([0, T]; \mathbf{S}_h)$ can be expressed as the linear combination of the nodal functions $\mathbf{w}_h : [0, T] \rightarrow \mathbb{R}^n$

$$\mathbf{w}_h(\mathbf{x}, t) = \mathbf{N}_h(\mathbf{x}) \mathbf{w}_h(t),$$

where $\mathbf{N}_h : \Omega_h \rightarrow \mathbb{R}^{m,n}$ is the matrix function

$$\mathbf{N}_h(\mathbf{x}) = \text{diag}(\bar{\mathbf{N}}, \bar{\mathbf{N}}, \dots, \bar{\mathbf{N}}),$$

with

$$\bar{\mathbf{N}} = \begin{pmatrix} N_1(\mathbf{x}) & N_2(\mathbf{x}) & \cdots & N_{n_d}(\mathbf{x}) \end{pmatrix},$$

and being $N_j : \Omega_h \rightarrow \mathbb{R}$, $j = 1, \dots, n_d$, the nodal basis, m the number of system components ($m = d + 2$), n_d the number of nodes per cell and n the number of unknowns per cell ($n = n_d \times m$). The system in Eq. 3.114 can be written as a set of ordinary differential equations (*ODEs*)

$$\mathbf{M} \frac{d\mathbf{w}(t)}{dt} = \mathfrak{B}(\mathbf{w}(t)), \quad (4.1)$$

where $\mathbf{M} \in \mathbb{R}^{N_u, N_u}$ is the global mass matrix, the operator $\mathfrak{B} : \mathbb{R}^{N_u} \rightarrow \mathbb{R}^{N_u}$ represents the right hand side of Eq. 3.114, $\mathbf{w}(t)$ represents the vector of nodal functions and N_u is the total number of unknowns. Here, the global matrix \mathbf{M} is a block diagonal matrix, with as many symmetric matrices on the diagonal as elements K in the discretization. Hence, the inverse of \mathbf{M} can be computed easily, or even, for a fixed grid, can be precomputed during preprocessing.

Premultiplying at both sides of Eq. 4.1 by \mathbf{M}^{-1} , the system of *ODEs* can be rewritten in the form

$$\frac{d\mathbf{w}(t)}{dt} = \mathfrak{F}(\mathbf{w}(t)), \quad (4.2)$$

subject to the initial condition

$$\mathbf{w}(t^0) = \mathbf{w}^0. \quad (4.3)$$

Various numerical techniques to solve this temporal systems of *ODEs* have been developed. In general, the implicit methods are prohibitively expensive for large-scale systems, since the computational demand is dominated by the numerical linear algebra involved into the

nonlinear solver employed. On the other hand, explicit strategies exhibit strong time step restrictions, especially in case of stiff systems.

For Navier–Stokes problems there exist different sources of stiffness. The inviscid terms of Navier–Stokes equations have an acoustic time scale, defined by the characteristic speed (eigenvalue) $|\mathbf{v}| + c$, and a convective time scale defined by the eigenvalue $|\mathbf{v}|$. Thus, for problems into low Mach number regime, i.e. $|\mathbf{v}|/c \rightarrow 0$, the time step for explicit schemes will be strongly restricted by the faster acoustic phenomenon. There, a large time scale, defined by slow convective terms, should be discretized using extremely small time steps. Another example is found when special behaviors of the fluid should be captured, e.g. boundary layers or shock waves. For these problems fine, or extremely fine, grids are normally employed in specific parts of the domain, reducing the allowable time step (in case of explicit methods). This small cell-size requirement can also derive from large aspect ratios in the lengths of the domain, this condition is sometimes called *geometric stiffness*. A third example of stiff problems in fluid dynamic is the introduction of stiff source terms to the Navier–Stokes equations, as in the case of modeling of the gas dynamic of combustion processes.

Considering the simulation of the rapid expansion of a jet into a quiescent atmosphere, which is the central problem of this work, two of these stiffness sources can be expected. First, although high speed regions will be found which corresponds to the supersonic (or even hypersonic) regime in the core of the jet, other subsonic regions with very slow convective terms are also expected inside and outside of the jet. On the other hand, the large aspect ratio between the diameter of the nozzle and the characteristic dimension of the expansion chamber demands the use of spatial discretization with small cells (comparatively with the size of the domain). Additionally, the capturing of shocks and the boundary layers presented between the jet boundary and the stagnant fluid will require of an extra local refinement. Therefore, under this analysis, the selection of an efficient and stable time marching scheme is crucial aspect in the solution of this problem.

4.1. Runge–Kutta methods

Considering the system Eq. 4.2, the exact solution at time $t_{n+1} = t_n + \tau$ can be written in the form

$$\mathbf{w}(t_{n+1}) = \mathbf{w}(t_n) + \int_0^\tau \mathfrak{F}(\mathbf{w}(t_n + \sigma)) d\sigma. \quad (4.4)$$

The named Runge–Kutta methods are based on the approximation of the integral term in Eq. 4.4 using a quadrature formula with nodes $0 \leq c_1 < \dots < c_s \leq 1$, and weights b_1, \dots, b_s . Then, defining the approximation of \mathbf{w} on the i -th node as

$$\mathbf{W}_{n,i} \approx \mathbf{w}(t_n + c_i\tau), \quad i = 1, \dots, s, \quad (4.5)$$

an approximation to Eq. 4.4 can be expressed as

$$\mathbf{w}(t_{n+1}) \approx \mathbf{w}_n + \tau \sum_{i=1}^s b_i \mathfrak{F}(\mathbf{W}_{n,i}). \quad (4.6)$$

From now on $\mathbf{w}_n \approx \mathbf{w}(t_n)$. Further, $\mathbf{W}_{n,i}$ can be computed using another quadrature formula, i.e.,

$$\begin{aligned} \mathbf{w}(t_n + c_i\tau) &= \mathbf{w}(t_n) + \int_{t_n}^{t_n + c_i\tau} \mathfrak{F}(\boldsymbol{\sigma}) d\boldsymbol{\sigma}, \\ \mathbf{W}_{n,i} &= \mathbf{w}_n + \tau \sum_{j=1}^s a_{ij} \mathfrak{F}(\mathbf{W}_{n,j}), \quad i = 1, \dots, s. \end{aligned} \quad (4.7)$$

A complete scheme to approximate Eq. 4.4 can be formulated as

$$\begin{aligned}\mathbf{W}_{n,i} &= \mathbf{w}_n + \tau \sum_{j=1}^s a_{ij} \mathfrak{F}(\mathbf{W}_{n,j}), \quad i = 1, \dots, s, \\ \mathbf{w}_{n+1} &= \mathbf{w}_n + \tau \sum_{i=1}^s b_i \mathfrak{F}(\mathbf{W}_{n,i})\end{aligned}\tag{4.8}$$

Particularly, schemes with $a_{ij} = 0$ for $i \leq j$ correspond to explicit Runge–Kutta methods.

4.2. Strong stability preserving Runge–Kutta methods

A lot of efforts in this field focused on developing schemes to improve the stability properties. Particularly, the named strong stability preserving Runge–Kutta methods (*SSPRK*) [SR02, Got05] were developed to solve semidiscrete formulations of hyperbolic conservation laws, where discontinuous solutions can be present. These schemes preserve the stability properties satisfied by the forward Euler method, incorporating relaxed time-step restrictions. Although several optimal *SSPRK* schemes, in which the number of stages s equals the order of the scheme p , have been built and tested, more recent works propose new versions of *SSPRK*(s, p) with stage-exceeding-order ($s > p$) [SR02, KWD07, Ruu06, SR03], thereby relaxing the linear stability *CFL* constraints, allowing hence the use of larger time steps. Here, the optimal concept should be understood in the sense of allowing the maximum time-step size possible among all schemes with the same order and equal number of stages. Low-storage implementations of *SSPRK* are presented in [Ket08, Ruu06], optimal implicit versions of *SSPRK* are presented in [KMG09].

4.3. Exponential integrators

A different kind of methods are based on the idea of using exponential functions (or related functions) of the Jacobian matrix \mathcal{A}

$$\mathcal{A}(\mathbf{w}) = -\frac{d\mathfrak{F}}{d\mathbf{w}}(\mathbf{w}).\tag{4.9}$$

From [HLS98], this family of methods is named *exponential integrators*. The idea about to use the exponential function $e^{\tau\mathcal{A}}\mathbf{w}_n$ to predict the function $\mathbf{w}(t_{n+1})$, with $t_{n+1} = t_n + \tau$, is not new, but its use in the practice of numerical schemes was restricted until mid of eighties, when the first applications of the Krylov subspace theory in the efficient computation of matrix exponential functions were developed. The first works on this way were done in the field of the quantum dynamics [NW83, PL86] and later some applications in heat-conduction and general parabolic problems [CMV69, CZ11, HO05], chemical kinetics in combustion process [OY99, Bis12, Pus11], advection-diffusion [CVB04, MBCV09], diffusion-reaction [CZ11], shallow water equations [GP16] and fluid dynamics [SS91, ETFS94, SSS09, CP13], have been developed.

Here, only two family of methods will be briefly presented: the exponential Runge–Kutta methods and the exponential Rosenbrock methods. A extended explanation of these methods can be found in [HO05, HO10, HOS09, CO09].

4.3.1. Exponential Runge–Kutta methods

This kind of methods are based on a fixed linearization of $\mathfrak{F}(t, \mathbf{w}(t))$, i.e.

$$\mathfrak{F}(t, \mathbf{w}(t)) = -\mathcal{A}\mathbf{w}(t) + \mathfrak{G}(t, \mathbf{w}(t)),\tag{4.10}$$

where the system matrix is computed as

$$\mathcal{A} \approx -\frac{d\mathfrak{F}}{d\mathbf{w}}(\mathbf{w}^0),$$

and $\mathfrak{G}(t, \mathbf{w}(t))$ is the nonlinear remainder. Then, the system Eq. 4.2 is now written in the semilinear form

$$\frac{d\mathbf{w}(t)}{dt} + \mathcal{A}\mathbf{w}(t) = \mathfrak{G}(t, \mathbf{w}(t)), \quad (4.11)$$

and, using the Volterra integral equation (variation-of-constants formula), it is possible to write its exact solution as

$$\mathbf{w}(t + \tau) = e^{-\tau\mathcal{A}}\mathbf{w}(t) + \int_0^\tau e^{-(\tau-\sigma)\mathcal{A}}\mathfrak{G}(t + \sigma, \mathbf{w}(t + \sigma)) d\sigma. \quad (4.12)$$

Thus, selecting nodes $0 \leq c_1 < \dots < c_s \leq 1$ to approximate the integral term in Eq. 4.12 using a quadrature formula, the expression

$$\mathbf{w}(t + \tau) \approx \mathbf{w}_{n+1} = \chi(-\tau\mathcal{A})\mathbf{w}_n + \tau \sum_{i=1}^s b_i(-\tau\mathcal{A})\mathfrak{G}(t_n + c_i\tau, \mathbf{W}_{n,i}) \quad (4.13)$$

is obtained. Further, as in Eq. 4.5,

$$\mathbf{W}_{n,i} \approx \mathbf{w}(t_n + c_i\tau), \quad i = 1, \dots, s,$$

and, in the same spirit of the standard Runge–Kutta schemes, terms $\mathbf{W}_{n,i}$ will be approximated using a second quadrature rule

$$\mathbf{W}_{n,i} = \chi_i(-\tau\mathcal{A})\mathbf{w}_n + \tau \sum_{j=1}^s a_{ij}(-\tau\mathcal{A})\mathfrak{G}(t_n + c_j\tau, \mathbf{W}_{n,j}). \quad (4.14)$$

In Eq. 4.13 and Eq. 4.14, the functions $\chi(\mathbf{z})$ and $\chi_i(\mathbf{z})$, as well as the weights $a_{ij}(\mathbf{z})$ and $b_i(\mathbf{z})$, for $\mathbf{z} \in \mathbb{R}^{N_u, N_u}$, are constructed from exponential operators or related functions. Here, in order to get consistency in these formulas the conditions

$$\chi(0) = \chi_i(0) = 1 \quad (4.15)$$

will be assumed.

The scheme

$$\begin{aligned} \mathbf{W}_{n,i} &= \chi_i(-\tau\mathcal{A})\mathbf{w}_n + \tau \sum_{j=1}^s a_{ij}(-\tau\mathcal{A})\mathfrak{G}(t_n + c_j\tau, \mathbf{W}_{n,j}), \\ \mathbf{w}_{n+1} &= \chi(-\tau\mathcal{A})\mathbf{w}_n + \tau \sum_{i=1}^s b_i(-\tau\mathcal{A})\mathfrak{G}(t_n + c_i\tau, \mathbf{W}_{n,i}), \end{aligned} \quad (4.16)$$

is termed *exponential Runge–Kutta method*.

Remark 4.3.1. If $\mathcal{A} = 0$ in the semilinear form Eq. 4.11, i.e. $\mathfrak{F}(t, \mathbf{w}(t)) = \mathfrak{G}(t, \mathbf{w}(t))$, the scheme Eq. 4.16 reduces to the so called underlying Runge–Kutta method with coefficients $b_j = b_i(0)$ and $a_{ij} = a_{ij}(0)$. Besides, in order to keep this scheme invariant under transformation of Eq. 4.11 into autonomous form, the conditions

$$\sum_{j=1}^s b_j(0) = 1, \quad \sum_{j=1}^s a_{ij}(0) = c_i, \quad \text{for } i = 1, \dots, s, \quad (4.17)$$

should be satisfied.

If \mathbf{w}^* represents an equilibrium state of the autonomous system

$$\frac{d\mathbf{w}(t)}{dt} + \mathcal{A}\mathbf{w}(t) = \mathfrak{G}(\mathbf{w}(t)), \quad (4.18)$$

then it is desirable that the numerical scheme preserves the equilibrium, i.e. if $\mathbf{w}^0 = \mathbf{w}^*$, then $\mathbf{w}_n = \mathbf{W}_{n,i} = \mathbf{w}^*$ for all $n \geq 1$. This requirement gives rise to the additional conditions

$$\begin{aligned} \sum_{j=1}^s b_j(\mathbf{z}) &= \frac{\chi(\mathbf{z}) - 1}{\mathbf{z}}, \\ \sum_{j=1}^s a_{ij}(\mathbf{z}) &= \frac{\chi_i(\mathbf{z}) - 1}{\mathbf{z}}, \text{ for } i = 1, \dots, s. \end{aligned} \quad (4.19)$$

Definition 4.3.2. Let be \mathbb{X} a Banach space with seminorm $\|\cdot\|$. Then, defining the family of functions $\phi_k : \mathbb{R}^{N_u \cdot N_u} \rightarrow \mathbb{R}^{N_u \cdot N_u}$, as

$$\phi_k(\mathbf{z}) = \begin{cases} e^{\mathbf{z}}, & \text{for } k = 0, \\ \int_0^1 e^{(1-\sigma)\mathbf{z}} \frac{\sigma^{k-1}}{(k-1)!} d\sigma, & \text{for } k \geq 1, \end{cases} \quad (4.20)$$

it is possible to show that these operators are bounded on \mathbb{X} . Additionally, these functions satisfy $\phi_k(0) = 1/k!$, as well as the recurrence relation

$$\phi_{k+1}(\mathbf{z}) = \frac{\phi_k(\mathbf{z}) - \phi_k(0)}{\mathbf{z}}. \quad (4.21)$$

Restricting the methods presented in Eq. 4.16 to χ -functions with the form

$$\chi(\mathbf{z}) = e^{\mathbf{z}}, \quad \chi_i(\mathbf{z}) = e^{c_i \mathbf{z}}, \text{ for } i = 1, \dots, s, \quad (4.22)$$

the conditions Eq. 4.19 can now be written as

$$\begin{aligned} \sum_{j=1}^s b_j(\mathbf{z}) &= \phi_1(\mathbf{z}), \\ \sum_{j=1}^s a_{ij}(\mathbf{z}) &= c_i \phi_1(c_i \mathbf{z}), \text{ for } i = 1, \dots, s. \end{aligned} \quad (4.23)$$

Remark 4.3.3. The set of conditions Eq. 4.23 reduces to Eq. 4.17 for $\mathbf{z} = 0$.

From Eq. 4.22 and restricting this analysis to explicit schemes, i.e. $a_{ij} = 0$ for $i \leq j$, it follows that the exponential Runge–Kutta scheme Eq. 4.16 can be expressed in the form

$$\begin{aligned} \mathbf{W}_{n,i} &= e^{-\tau c_i \mathcal{A}} \mathbf{w}_n + \tau \sum_{j=1}^{i-1} a_{ij}(-\tau \mathcal{A}) \mathfrak{G}(t_n + c_j \tau, \mathbf{W}_{n,j}), \\ \mathbf{w}_{n+1} &= e^{-\tau \mathcal{A}} \mathbf{w}_n + \tau \sum_{i=1}^s b_i(-\tau \mathcal{A}) \mathfrak{G}(t_n + c_i \tau, \mathbf{W}_{n,i}). \end{aligned} \quad (4.24)$$

Further, considering that Eq. 4.24 involves calculation of matrix functions applied to vectors, then an efficient implementation of these operations is decisive in the general computational performance of the time integrator. Thus, using Eq. 4.23, a preferable form to

write Eq. 4.24 is

$$\begin{aligned}\mathbf{W}_{n,i} &= \mathbf{w}_n + c_i \tau \phi_1(-c_i \tau \mathcal{A}) \mathfrak{F}(\mathbf{w}_n) + \tau \sum_{j=2}^{i-1} a_{ij}(-\tau \mathcal{A}) \mathbf{D}_{n,j}, \\ \mathbf{w}_{n+1} &= \mathbf{w}_n + \tau \phi_1(-\tau \mathcal{A}) \mathfrak{F}(\mathbf{w}_n) + \tau \sum_{i=2}^s b_i(-\tau \mathcal{A}) \mathbf{D}_{n,i}.\end{aligned}\tag{4.25}$$

where $\mathbf{D}_{n,i} = \mathfrak{G}(t_n + c_i \tau, \mathbf{W}_{n,i}) - \mathfrak{G}(t_n, \mathbf{w}_n)$. Here, s computations of matrix functions applied to vectors are involved: $s - 1$ products using vectors $\mathbf{D}_{n,j}$ with $\|\mathbf{D}_{n,j}\| = O(\tau)$, and only one product using the vector $\mathfrak{F}(\mathbf{w}_n)$ with $\|\mathfrak{F}(\mathbf{w}_n)\| = O(1)$. Therefore, it is to be expected that if Krylov subspaces are used to compute the matrix exponentials, only one expensive operation has to be computed, and its cost should dominate the global computational cost of the scheme.

4.3.1.1. Exponential Euler method

The simplest exponential Runge–Kutta scheme is built using $s = 1$ and is termed *exponential Euler method*. Thus, the Eq. 4.23, for this case yields

$$b_1(-\tau \mathcal{A}) = \phi_1(\tau \mathcal{A})$$

and from Eq. 4.24 we get

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \tau \phi_1(-\tau \mathcal{A}) \mathfrak{F}(t_n, \mathbf{w}_n).\tag{4.26}$$

Hence, the comparative low cost associated to this scheme comes from the computation of a single product matrix-exponential $(\phi_1(-\tau \mathcal{A}))$ times vector $(\mathfrak{F}(t_n, \mathbf{w}_n))$.

Assumption 4.3.4. *Let be \mathbb{X} a Banach space with seminorm $\|\cdot\|_{\mathbb{X}}$. In the context of this text it will be assumed that \mathcal{A} is a linear operator on \mathbb{X} and that $-\mathcal{A}$ is an infinitesimal generator of the strongly continuous semigroup $e^{-\tau \mathcal{A}}$ on \mathbb{X} .*

Theorem 4.3.5. *Considering that the system Eq. 4.11, subject to the initial condition $\mathbf{w}(t^0) = \mathbf{w}^0$, satisfies the Assumption 4.3.4, and assuming that $\mathfrak{G} : [0, T] \rightarrow \mathbb{X}$ is differentiable with $\mathfrak{G}' \in L^\infty([0, T]; \mathbb{X})$, then the error generated by the exponential Euler scheme Eq. 4.26 is uniformly bounded by*

$$\|\mathbf{w}_n - \mathbf{w}(t_n)\|_{\mathbb{X}} = C\tau \sup_{0 \leq t \leq t_n} \|\mathfrak{G}'(t)\|_{\mathbb{X}},\tag{4.27}$$

where C is a constant independent of n and τ , but dependent on T .

Proof. The proof of this theorem can be found in [HO10]. □

4.3.1.2. Convergence of higher order exponential Runge–Kuta methods

Assumption 4.3.6. *Let be \mathbb{X} a Banach space with seminorm $\|\cdot\|_{\mathbb{X}}$. We will assume that \mathbf{w} is a sufficiently smooth solution of Eq. 4.11, and that $\mathfrak{G} : [0, T] \times \mathbb{X} \rightarrow \mathbb{X}$ is a sufficiently often differentiable function in the sense of Fréchet in a neighborhood of the exact solution. Thus, all derivatives involved can be assumed uniformly bounded.*

Theorem 4.3.7. *Let the system Eq. 4.11, endowed with the initial condition $\mathbf{w}(t^0) = \mathbf{w}^0$, satisfies the Assumption 4.3.4 and the Assumption 4.3.6. Consider that this initial value problem is solved using the explicit exponential Runge–Kutta method presented in Eq. 4.24, complying the conditions in Eq. 4.23. Then, assuming that for $2 \leq p \leq 4$, the order*

Table 4.1.: Stiff order conditions for explicit exponential Runge–Kutta methods.

No.	Order	Order condition
1	1	$\psi_1(-\tau\mathcal{A}) = 0$
2	2	$\psi_2(-\tau\mathcal{A}) = 0$
3	2	$\psi_{1,i}(-\tau\mathcal{A}) = 0$
4	3	$\psi_3(-\tau\mathcal{A}) = 0$
5	3	$\sum_{i=1}^s b_i(-\tau\mathcal{A}) J \psi_{2,i}(-\tau\mathcal{A}) = 0$
6	4	$\psi_4(-\tau\mathcal{A}) = 0$
7	4	$\sum_{i=1}^s b_i(-\tau\mathcal{A}) J \psi_{3,i}(-\tau\mathcal{A}) = 0$
8	4	$\sum_{i=1}^s b_i(-\tau\mathcal{A}) J \sum_{j=2}^{i-1} a_{ij}(-\tau\mathcal{A}) J \psi_{2,j}(-\tau\mathcal{A}) = 0$
9	4	$\sum_{i=1}^s b_i(-\tau\mathcal{A}) c_i K \psi_{2,i}(-\tau\mathcal{A}) = 0$

conditions in Table 4.1 hold up to order $p - 1$, with $\psi_p(0) = 0$, and that the remaining conditions of order p hold in a weaker form, with $b_i(0)$ instead of $b_i(-\tau\mathcal{A})$, for $i = 2, \dots, s$, the error involved in the numerical solution \mathbf{w}_n is uniformly bounded as

$$\|\mathbf{w}_n - \mathbf{w}(t_n)\| \leq C\tau^p, \quad (4.28)$$

where C is a constant independent of τ and n , but dependent on T .

Proof. All the details about the proof of this theorem can be found in [HO10]. \square

Note that in Table 4.1, J and K denote arbitrary bounded operators on \mathbb{X} , and that the functions ψ_j and $\psi_{j,i}$ are defined in Eq. 4.29 and Eq. 4.30, respectively.

$$\psi_j(-\tau\mathcal{A}) = \phi_j(-\tau\mathcal{A}) - \sum_{i=1}^s b_i(-\tau\mathcal{A}) \frac{c_i^{j-1}}{(j-1)!} \quad (4.29)$$

$$\psi_{j,i}(-\tau\mathcal{A}) = \phi_j(-c_i\tau\mathcal{A}) c_i^j - \sum_{k=1}^{i-1} a_{ik}(-\tau\mathcal{A}) \frac{c_k^{j-1}}{(j-1)!}. \quad (4.30)$$

4.3.1.3. Second order exponential Runge–Kutta method

For a second order exponential Runge–Kutta method, two internal stages $s = 2$ are required, thus from Theorem 4.3.7 the conditions

$$\begin{aligned} \phi_1(-\tau\mathcal{A}) &= b_1(-\tau\mathcal{A}) + b_2(-\tau\mathcal{A}), \\ \phi_2(-\tau\mathcal{A}) &= c_1 b_1(-\tau\mathcal{A}) + c_2 b_2(-\tau\mathcal{A}), \\ \phi_{1,1}(-\tau\mathcal{A}) &= \phi_1(-c_1\tau\mathcal{A}) c_1 = 0, \\ \phi_{1,2}(-\tau\mathcal{A}) &= \phi_1(-c_2\tau\mathcal{A}) c_2 - a_{21}(-\tau\mathcal{A}) = 0, \end{aligned} \quad (4.31)$$

from Table 4.1 should be satisfied. From these conditions it follows that a c_2 -parameter family of methods can be formulated as

$$\begin{array}{c|cc} c_1 & & 0 \\ c_2 & a_{21} & = c_2 \frac{c_2 \phi_1(-c_2\tau\mathcal{A})}{\phi_1(-\tau\mathcal{A}) - \frac{1}{c_2} \phi_2(-\tau\mathcal{A}) - \frac{1}{c_2} \phi_2(-\tau\mathcal{A})} \end{array}. \quad (4.32)$$

Besides, using the weak form

$$\phi_2(0) = c_2 b_2(0) = \frac{1}{2},$$

instead of the second condition in Eq. 4.31, the second order scheme

$$\begin{array}{c|c} 0 & \\ \hline c_2 & \frac{c_2 \phi_1(-c_2 \tau \mathcal{A})}{\phi_1(-\tau \mathcal{A}) \left(1 - \frac{1}{2c_2}\right) - \frac{1}{c_2} \phi_2(-\tau \mathcal{A})} \end{array}. \quad (4.33)$$

is achieved. Here, an attractive version is found with $c_2 = 1/2$, which simplifies the scheme because then $b_1 = 0$.

4.3.2. Rosenbrock-type methods

In contrast to the exponential Euler methods, the Rosenbrock-type methods [CO09, HOS09, HO10] are based on the continuous linearization of Eq. 4.2. Thus, using the linearization

$$\mathfrak{F}(\mathbf{w}(t_n)) = -\mathcal{A}_n \mathbf{w}(t_n) + \mathfrak{G}(\mathbf{w}(t_n)), \quad (4.34)$$

with

$$\mathcal{A}_n \approx -\frac{\partial \mathfrak{F}}{\partial \mathbf{w}}(\mathbf{w}_n), \quad (4.35)$$

the autonomous system Eq. 4.2 can be written as

$$\frac{d\mathbf{w}(t)}{dt} + \mathcal{A}_n \mathbf{w}(t_n) = \mathfrak{G}(\mathbf{w}(t_n)). \quad (4.36)$$

Proceeding as in Sec. 4.3.1, a general exponential Rosenbrock scheme is given by

$$\begin{aligned} \mathbf{W}_{n,i} &= \mathbf{w}_n + c_i \tau \phi_1(-c_i \tau \mathcal{A}_n) \mathfrak{F}(\mathbf{w}_n) + \tau \sum_{j=2}^{i-1} a_{ij} (-\tau \mathcal{A}_n) \mathbf{D}_{n,j}, \quad \text{for } 1 \leq i \leq s, \\ \mathbf{w}_{n+1} &= \mathbf{w}_n + \tau \phi_1(-\tau \mathcal{A}_n) \mathfrak{F}(\mathbf{w}_n) + \tau \sum_{i=2}^s b_i (-\tau \mathcal{A}_n) \mathbf{D}_{n,i}, \end{aligned} \quad (4.37)$$

where

$$\mathbf{D}_{n,i} = \mathfrak{G}(\mathbf{W}_{n,i}) - \mathfrak{G}(\mathbf{w}_n).$$

4.3.2.1. Convergence order of the exponential Rosenbrock methods

Theorem 4.3.8. *Let the system Eq. 4.11, endowed with the initial condition $\mathbf{w}(t^0) = \mathbf{w}^0$, satisfies the Assumption 4.3.4 and the Assumption 4.3.6. Consider that this initial value problem is solved using the explicit exponential Rosenbrock method presented in Eq. 4.37, complying the conditions in Eq. 4.23. Thus, assuming that for $2 \leq p \leq 4$ the order conditions in Table 4.2 hold up to order p , and that the step size sequence satisfy the condition*

$$\sum_{k=1}^{n-1} \sum_{j=0}^{k-1} \tau_j^{p+1} \leq C_H, \quad (4.38)$$

with C_H being a constant uniform for $t_0 \leq t \leq T$, then for sufficiently small values of C_H the numerical error of the scheme is uniformly bounded by

$$\|\mathbf{w}_n - \mathbf{w}(t_n)\| \leq C \sum_{j=0}^{n-1} \tau_j^{p+1}, \quad (4.39)$$

where C is a constant independent of the time step sequence.

Table 4.2.: Stiff order conditions for exponential Rosenbrock methods.

No.	Order	Order condition
1	1	$\sum_{i=1}^s b_i(-\tau\mathcal{A}) = \phi_1(-\tau\mathcal{A})$
2	2	$\sum_{j=1}^{i-1} a_{ij}(-\tau\mathcal{A}) = c_i\phi_1(-c_i\tau\mathcal{A}), \text{ for } 2 \leq i \leq s$
3	3	$\sum_{i=2}^s b_i(-\tau\mathcal{A}) c_i^2 = 2\phi_3(-\tau\mathcal{A})$
4	4	$\sum_{i=2}^s b_i(-\tau\mathcal{A}) c_i^3 = 6\phi_4(-\tau\mathcal{A})$

Note that for constant time-steps, the condition Eq. 4.38 holds with

$$C_H = \frac{1}{2}\tau^{p-1}(t_n - t^0)^2, \quad (4.40)$$

and since $p \geq 2$, C_H tends to zero if $\tau \rightarrow 0$. Additional details about this condition for non-constant time-steps can be found in [HOS09].

Proof. The proof of this theorem can be found in [HOS09, HO10]. \square

4.3.2.2. Exponential Rosenbrock–Euler method

The simplest exponential Rosenbrock scheme, called *exponential Rosenbrock–Euler* method, is given by

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \tau\phi_1(-\tau\mathcal{A}_n) \mathfrak{F}(\mathbf{w}_n). \quad (4.41)$$

Thus, according with Theorem 4.3.8, it is a second order convergent scheme with one stage and with only one operation matrix-function times vector involved.

4.3.2.3. Higher order Rosenbrock-type methods

In order to allow higher order schemes with variable time-steps, an embedded approximation should be included into the basic Rosenbrock-type stencil in Eq. 4.37

$$\widehat{\mathbf{w}}_{n+1} = e^{-\tau_n\mathcal{A}_n} + \tau_n \sum_{i=1}^s \widehat{b}_i(-\tau_n\mathcal{A}_n) \mathfrak{G}(\mathbf{W}_{n,i}).$$

Third order exponential Rosenbrock method (*exprb32*)

The *exprb32* scheme [HO10, HOS09] is a third-order Rosenbrock scheme with a second-order error estimator (exponential Rosenbrock-Euler scheme). Thus, the coefficients for this methods are given by

$$\begin{array}{c|cc} c_1 & & 0 \\ c_2 & a_{21} & 1 \\ \hline & b_1 & b_2 \\ & \widehat{b}_1 & \end{array} = \frac{1}{\begin{array}{c|cc} & \phi_1 & \\ \hline & \phi_1 - 2\phi_3 & 2\phi_3 \\ & \phi_1 & \end{array}}. \quad (4.42)$$

Fourth order exponential Rosenbrock method (*exprb43*)

This scheme [HO10, HOS09, CO09] consists of a fourth-order Rosenbrock scheme along with a third-order error estimator. Thus, the coefficients for this methods are given by

$$\begin{array}{c|cc|c|cc}
 c_1 & & & 0 & & \\
 c_2 & a_{21} & & \frac{1}{2} & \frac{1}{2}\phi_1\left(\frac{1}{2}\cdot\right) & \\
 c_3 & a_{31} & a_{32} & = 1 & 0 & \phi_1 \\
 \hline
 & b_1 & b_2 & b_3 & \phi_1 - 14\phi_3 + 36\phi_4 & 16\phi_3 - 48\phi_4 & -2\phi_3 + 12\phi_4 \\
 & \widehat{b}_1 & \widehat{b}_2 & \widehat{b}_3 & \phi_1 - 14\phi_3 & 16\phi_3 & -2\phi_3
 \end{array} . \quad (4.43)$$

exp4 method

The *exp4* method, presented initially in [HLS98], is one of the most known exponential integrators. In its efficient-implementation form, this scheme consists of seven stages, which demand only three function evaluations and involve only computations of the ϕ_1 -operator. In this form, the scheme can be written as:

$$\mathbf{w}(t_n + \tau) = \mathbf{w}(t_n) + \tau \left(k_3 + k_4 - \frac{4}{3}k_5 + k_6 + \frac{1}{6}k_7 \right), \quad (4.44)$$

where the first three stages are given by

$$\begin{aligned}
 k_1 &= \phi_1 \left(-\frac{1}{3}\tau\mathcal{A}_n \right) \mathfrak{F}(\mathbf{w}(t_n)), \\
 k_2 &= \phi_1 \left(-\frac{2}{3}\tau\mathcal{A}_n \right) \mathfrak{F}(\mathbf{w}(t_n)), \\
 k_3 &= \phi_1(-\tau\mathcal{A}_n) \mathfrak{F}(\mathbf{w}(t_n)),
 \end{aligned} \quad (4.45)$$

the fourth, fifth and sixth stages are computed using

$$\begin{aligned}
 w_4 &= -\frac{7}{300}k_1 + \frac{97}{150}k_2 - \frac{37}{300}k_3, \\
 u_4 &= \mathbf{w}(t_n) + \tau w_4, \\
 r_4 &= \mathfrak{F}(u_4) - \mathfrak{F}(\mathbf{w}(t_n)) + \tau\mathcal{A}_n w_4, \\
 k_4 &= \phi_1 \left(-\frac{1}{3}\tau\mathcal{A}_n \right) r_4, \\
 k_5 &= \phi_1 \left(-\frac{2}{3}\tau\mathcal{A}_n \right) r_4, \\
 k_6 &= \phi_1(-\tau\mathcal{A}_n) r_4,
 \end{aligned} \quad (4.46)$$

while the last stage is computed as

$$\begin{aligned}
 w_7 &= \frac{59}{300}k_1 - \frac{7}{75}k_2 + \frac{269}{300}k_3 + \frac{2}{3}(k_4 + k_5 + k_6), \\
 u_7 &= \mathbf{w}(t_n) + \tau w_7, \\
 r_7 &= \mathfrak{F}(u_7) - \mathfrak{F}(\mathbf{w}(t_n)) + \tau\mathcal{A}_n w_7, \\
 k_7 &= \phi_1 \left(-\frac{1}{3}\tau\mathcal{A}_n \right) r_7.
 \end{aligned} \quad (4.47)$$

Although this scheme can be interpreted as a three-stages exponential Rosenbrock method, which is fourth-order convergent, the use of only ϕ_1 -operators restricts it, according to the conditions in Table 4.2, to be second-order convergent. A good comparative analysis about the performance of this scheme is included in [TL10].

4.4. Approximation of the matrix exponential operators

All different schemes presented in the previous section involve computations of matrix exponentials or related functions. Thus, an efficient implementation of these operations is crucial in the global performance of the time-stepping scheme.

Of course the first idea to compute a matrix exponential applied to a vector is using its representation in a convergent power series

$$e^{\mathbf{z}\mathbf{v}} = \sum_{n=0}^{\infty} \frac{1}{n!} \mathbf{z}^n \mathbf{v}, \quad (4.48)$$

for $\mathbf{z} \in \mathbb{R}^{N_u, N_u}$ and $\mathbf{v} \in \mathbb{R}^{N_u}$. Nevertheless, the implementation of a truncated series is one of the lowest performance options to select [ML03]. Even, other techniques as Padé approximations [ACF96, LZG11, ML03] or rational Chebyshev polynomials [GS92, ML03] are not applicable for large scale problems, as is the case of *ODE*-systems derived from spatial semidiscretization of partial differential equations. For large-scale problems special strategies, as approximations using Leja points [CVB04] or methods based on Krylov subspaces [HL97, GS89, GS92, Saa92], should be implemented.

Particularly, using projections onto Krylov subspaces, the approximation of $e^{-\tau\mathcal{A}}\mathbf{v}$ or $\phi_1(-\tau\mathcal{A})\mathbf{v}$ operations, involved into the exponential integrators, converges considerably faster than the Krylov projection of $(\mathbb{I} - \tau\mathcal{A})^{-1}\mathbf{v}$ for implicit Euler scheme, as long as a good preconditioner is not available [HL97].

4.4.1. Approximation of matrix exponential operators using Krylov subspaces

The idea here is to approximate the matrix-vector product at the left hand side of Eq. 4.48 using a polynomial expression, i.e.

$$e^{\mathbf{z}\mathbf{v}} \approx P_{m-1}(\mathbf{z})\mathbf{v}, \quad (4.49)$$

where P_{m-1} represents a $(m-1)$ -degree polynomial. As this approximation is an element of

$$\mathcal{K}_m = \text{span}\{\mathbf{v}, \mathbf{z}\mathbf{v}, \mathbf{z}^2\mathbf{v}, \dots, \mathbf{z}^{m-1}\mathbf{v}\}, \quad (4.50)$$

the problem can be reformulated. The idea now consists in identifying the element in the m -dimensional Krylov subspace \mathcal{K}_m that best approximates the term $e^{\mathbf{z}\mathbf{v}}$. In order to build the Krylov subspace, Arnoldi's algorithm (see Algorithm 2) can be used. This method constructs an orthonormal base

$$\mathbf{V} = [V_1, V_2, \dots, V_m]$$

of \mathcal{K}_m , with $V_i \in \mathbb{R}^{N_u}$, $i = 1, 2, \dots, m$, which satisfies the relation

$$\mathbf{z}\mathbf{V} = \mathbf{V}\mathbf{H} + H_{m+1,m}V_{m+1}\mathbf{e}_m^T, \quad (4.51)$$

and

$$\mathbf{V}^T\mathbf{V} = \mathbb{I}, \quad (4.52)$$

where $\mathbf{e}_m \in \mathbb{R}^m$ denotes the m -th unit vector and $H_{i,j}$ denotes an element of the upper Hessenberg matrix $\mathbf{H} \in \mathbb{R}^{m,m}$.

Algorithm 2: Arnoldi's decomposition algorithm.

Data: \mathbf{z}, \mathbf{v}
Result: \mathbf{V}, \mathbf{H}

```

1  $V_1 = \mathbf{v} / \|\mathbf{v}\|$ ;
2 for  $j = 1 : m$  do
3    $\mathbf{p} = \mathbf{z} V_j$ ;
4   for  $(i = 1 : j)$  do
5      $H_{i,j} = V_i^T \mathbf{p}$ ;
6      $\mathbf{p} = \mathbf{p} - H_{i,j} V_i$ ;
7   end
8    $H_{j+1,j} = \|\mathbf{p}\|$ ;
9    $V_{j+1} = \mathbf{p} / H_{j+1,j}$ ;
10 end
```

Theorem 4.4.1. *Let the function $\phi(\mathbf{z}) : \mathbb{R}^{N_u, N_u} \rightarrow \mathbb{R}^{N_u, N_u}$ be analytic in a neighborhood of its numerical range*

$$\mathcal{F}(\mathbf{z}) = \{\mathbf{y}^T \mathbf{z} \mathbf{y} : \mathbf{y} \in \mathbb{R}^{N_u}, \mathbf{y}^T \mathbf{y} = 1\}$$

and let $\mathbf{v} \in \mathbb{R}^{N_u}$. Then, using the Hessenberg matrix \mathbf{H} derived from Arnoldi's algorithm, it is possible to approximate the product $\phi(\mathbf{z}) \mathbf{v}$ by the expression

$$\phi(\mathbf{z}) \mathbf{v} \approx \beta \mathbf{V} \phi(\mathbf{H}) \mathbf{e}_1, \quad (4.53)$$

where $\beta = |\mathbf{v}|$.

Proof. Writing the Cauchy integral

$$\phi(\mathbf{z}) \mathbf{v} = \frac{1}{2\pi i} \int_{\Gamma} \phi(\lambda) (\lambda \mathbb{I} - \mathbf{z})^{-1} \mathbf{v} d\lambda = \frac{1}{2\pi i} \int_{\Gamma} \phi(\lambda) x(\lambda) d\lambda, \quad (4.54)$$

where Γ is a suitable contour that surrounds the field of values $\mathcal{F}(\mathbf{z})$. This relation contains, for each $\lambda \in \Gamma$, the solution of the linear system

$$(\lambda \mathbb{I} - \mathbf{z}) x(\lambda) = \mathbf{v}. \quad (4.55)$$

Now, considering that by definition this linear system can be solved approximately using the relation [HL97]

$$x_m(\lambda) = \beta \mathbf{V} (\lambda \mathbb{I} - \mathbf{H})^{-1} \mathbf{e}_1, \quad (4.56)$$

and that $\mathcal{F}(\mathbf{H}) \subset \mathcal{F}(\mathbf{z})$, then the Eq. 4.54 can be written now as

$$\phi(\mathbf{z}) \mathbf{v} \approx \frac{1}{2\pi i} \int_{\Gamma} \phi(\lambda) x_m(\lambda) d\lambda = \frac{1}{2\pi i} \int_{\Gamma} \phi(\lambda) \beta \mathbf{V} (\lambda \mathbb{I} - \mathbf{H})^{-1} \mathbf{e}_1 d\lambda, \quad (4.57)$$

and from this relation, the expression

$$\phi(\mathbf{z}) \mathbf{v} \approx \beta \mathbf{V} \phi(\mathbf{H}) \mathbf{e}_1$$

holds. □

Using Eq. 4.51, it is possible to write

$$\mathbf{H} \approx \mathbf{V}^T \mathbf{V} \mathbf{z}, \quad (4.58)$$

and then, the Hessenberg matrix can be understood as the projection of the large system matrix \mathbf{z} onto the Krylov subspace \mathcal{K}_m . Likewise, from Theorem 4.4.1, it is possible to approximate the matrix-vector products $e^{\mathbf{z}}\mathbf{v}$ and $\phi_k(\mathbf{z})\mathbf{v}$ as

$$e^{\mathbf{z}}\mathbf{v} \approx \beta \mathbf{V} e^{\mathbf{H}} \mathbf{e}_1 \quad (4.59)$$

and

$$\phi_k(\mathbf{z})\mathbf{v} \approx \beta \mathbf{V} \phi_k(\mathbf{H}) \mathbf{e}_1, \quad \text{for } k = 1, 2, \dots, \quad (4.60)$$

respectively. Here, $\beta = |\mathbf{v}|$ and ϕ_k -functions are defined according to Def. 4.3.2.

It is important to note that, in Eq. 4.59 and Eq. 4.60, the exponential functions of large-scale matrices $\mathbf{z} \in \mathbb{R}^{N_u, N_u}$ was replaced by exponential operations of a Hessenberg matrix $\mathbf{H} \in \mathbb{R}^m$, with $m \ll N_u$.

4.4.2. Computation of the exponential of \mathbf{H}

In previous section the computation of large-scale matrix exponentials was replaced by exponential operators that involve Hessenberg matrices considerably smaller than the original system matrix. In practical applications the size of the Krylov subspaces built via Arnoldi decomposition use to be less than 50 vectors [Sid98].

Nevertheless, the cost of computing the exponential of Hessenberg matrices is nonnegligible [GS92]. Thus, a stable, precise and low cost method to approximate these operations should be implemented. One common option is to use rational approximation and computation through partial fraction expansions, i.e.

$$e^{\mathbf{z}} \approx R_{\nu_1, \nu_2}(\mathbf{z}) = [Q_{\nu_2}(\mathbf{z})]^{-1} P_{\nu_1}(\mathbf{z}), \quad \text{for } \mathbf{z} \in \mathbb{R}^{m, m} \quad (4.61)$$

and where P_{ν_1} and Q_{ν_2} denote polynomials of degrees ν_1 and ν_2 , respectively. Two methods derived from Eq. 4.61 are the rational Chebyshev approximation [GS92, Saa92, Sid98] and the Padé approximation [Sid98]. A good collection of methods to compute matrix exponential can be found in [ML03]. An comparative analysis, in terms of accuracy and efficiency, between Padé methods and rational Chebyshev approximations is presented in [OY99].

4.4.2.1. Rational Chebyshev approximations

The idea with this method is to seek the rational Chebyshev approximation that minimizes the maximum error in $z \in [0, +\infty)$, i.e.

$$\|R_{\nu, \nu}(z) - e^{-z}\|_{L^\infty[0, +\infty)} = \min_{r_{\nu, \nu} \in \mathcal{C}_{\nu, \nu}} \max_{z \in [0, +\infty)} |r_{\nu, \nu}(z) - e^{-z}|, \quad (4.62)$$

where $\mathcal{C}_{\nu, \nu}$ represents the full set of rational Chebyshev functions of degree (ν, ν) . Further more, the rational approximation can be expressed as a expansion in partial fractions

$$e^{-z} \approx R_{\nu, \nu}(z) = \alpha_0 + \sum_{i=1}^{\nu} \frac{\alpha_i}{z - \lambda_i}, \quad (4.63)$$

where both coefficients α_i and poles λ_i , that solve the problem configured in Eq. 4.62 can be precomputed. In [GS92] a complete set of coefficients and poles for $\nu = 10$ and $\nu = 14$ is presented. A more accurate set of values for α_i and λ_i were computed and presented in [Pus11] and are used in the present work.

For cases where z is real, the poles form conjugate pairs and then the computational cost of Eq. 4.63 can be reduced to half using

$$e^{-z} \approx R_{\nu,\nu}(z) = \alpha_0 + 2\operatorname{Re} \sum_{i=1}^{\frac{\nu}{2}} \frac{\alpha_i}{z - \lambda_i}. \quad (4.64)$$

Additionally, for $\mathbf{z} \in \mathbb{R}^{m,m}$ and $\mathbf{v} \in \mathbb{R}^m$, a more efficient implementation of Eq. 4.64, avoiding inversion of matrices, can be achieved using a matrix-vector product formulation

$$e^{-\mathbf{z}} \mathbf{v} \approx \alpha_0 \mathbf{v} + 2\operatorname{Re} \sum_{i=1}^{\frac{\nu}{2}} \alpha_i (\mathbf{z} - \lambda_i \mathbb{I})^{-1} \mathbf{v}. \quad (4.65)$$

According to the problem definition in Eq. 4.62, the Chebyshev rational approximation is more suited to use for symmetric and definite positive matrices. In this case, the error in the approximation is bounded by [Sid98]

$$\|R_{\nu,\nu}(\mathbf{z}) - e^{-\mathbf{z}}\| \leq \Lambda_{\nu,\nu}, \quad (4.66)$$

where $\Lambda_{\nu,\nu} = 10^\nu$. However, [GS92] reports accurate approximations are also achieved in case of matrices with eigenvalues near to the positive real axis. In [Sid98] an interesting analysis about the behavior of the error in case of matrices with complex spectrum is presented.

4.4.2.2. Diagonal Padé approximations

An alternative approximation to the matrix exponential $e^{\mathbf{z}}$ consists in to use a (ν_1, ν_2) -type Padé approximation, i.e.

$$e^{\mathbf{z}} \approx R_{\nu_1,\nu_2}(\mathbf{z}) = [Q_{\nu_1,\nu_2}(\mathbf{z})]^{-1} P_{\nu_1,\nu_2}(\mathbf{z}), \quad (4.67)$$

where

$$P_{\nu_1,\nu_2}(\mathbf{z}) = \sum_{i=0}^{\nu_1} \frac{(\nu_1 + \nu_2 - i)! \nu_1!}{(\nu_1 + \nu_2)! i! (\nu_1 - i)!} \mathbf{z}^i \quad (4.68)$$

and

$$Q_{\nu_1,\nu_2}(\mathbf{z}) = \sum_{i=0}^{\nu_2} \frac{(\nu_1 + \nu_2 - i)! \nu_2!}{(\nu_1 + \nu_2)! i! (\nu_2 - i)!} (-\mathbf{z})^i. \quad (4.69)$$

It is possible to show that this approximation match the Taylor series expansion up to order $\nu_1 + \nu_2$ [Sid98]. A stable and economy version of Eq. 4.67 is found using $\nu_1 = \nu_2$, configuring the termed diagonal Padé rational approximation. In an efficient form this approximation can be computed as

$$e^{\mathbf{z}} \approx R_{\nu,\nu}(\mathbf{z}) = \frac{P_{\nu,\nu}(\mathbf{z})}{P_{\nu,\nu}(-\mathbf{z})} = \begin{cases} 1 + 2 \frac{\mathbf{z} \sum_{i=0}^{\nu/2-1} C_{2i+1} \mathbf{z}^{2i}}{\sum_{i=0}^{\nu/2} C_{2i} \mathbf{z}^{2i} - \mathbf{z} \sum_{i=0}^{\nu/2-1} C_{2i+1} \mathbf{z}^{2i}}, & \text{if } p \text{ is even,} \\ -1 - 2 \frac{\sum_{i=0}^{(\nu-1)/2} C_{2i} \mathbf{z}^{2i}}{\mathbf{z} \sum_{i=0}^{(\nu-1)/2} C_{2i+1} \mathbf{z}^{2i} - \sum_{i=0}^{(\nu-1)/2} C_{2i} \mathbf{z}^{2i}}, & \text{if } p \text{ is odd,} \end{cases} \quad (4.70)$$

where $C_0 = 1$ and

$$C_i = C_{i-1} \frac{\nu + 1 - i}{(2\nu + 1 - i) i}.$$

Because the Padé method can only produce accurate approximations near to the origin, i.e. for small values of $\|\mathbf{z}\|$, then to incorporate strategies as the scaling-squaring method [ML03, Hig09, Sid98] is recommended in case of matrices with large norms.

The scaling and squaring method

The idea of the scaling-squaring method is initially to reduce the norm of \mathbf{z} in such a way that $2^{-s}\|\mathbf{z}\| \approx 1$, allowing the computation of the exponential via of Padé approximation. Then, the computation of the exponential of the original matrix is completed applying s successive squaring operations, i.e.

$$e^{\mathbf{z}} \approx \left(R_{\nu,\nu} \left(\frac{\mathbf{z}}{2^s} \right) \right)^{2^s}. \quad (4.71)$$

However, for matrices with too large norms $s \gg 1$ is required. Then the cost increases significantly with the number of squaring employed, as well as the rounding errors introduced into the approximation. Hence, choosing of an adequate pair (ν, s) is crucial in order to conserve a good balance between accuracy and performance.

In [ML03] it is shown that if $\|\mathbf{z}\| \leq 2^{s-1}$, then

$$\left(R_{\nu,\nu} \left(\frac{\mathbf{z}}{2^s} \right) \right)^{2^s} = e^{\mathbf{z} + \mathbf{E}}, \quad (4.72)$$

where

$$\frac{\|\mathbf{E}\|}{\|\mathbf{z}\|} \leq \frac{(\nu!)^2}{(2\nu)!(2\nu+1)!} \left(\frac{1}{2} \right)^{2\nu-3} \approx \begin{cases} 0.34 \times 10^{-15} & \text{with } \nu = 6, \\ 0.11 \times 10^{-18} & \text{with } \nu = 7, \\ 0.27 \times 10^{-22} & \text{with } \nu = 8 \end{cases}. \quad (4.73)$$

In [ML03] this result is used for computing optimal pairs (ν, s) for some combinations of matrix norms $\|\mathbf{z}\|$ and desirable error tolerances ϵ such that

$$\frac{\|\mathbf{E}\|}{\|\mathbf{z}\|} \leq \epsilon.$$

Likewise, from Eq. 4.73 it follows that an error bound for the diagonal Padé approximation with scaling-squaring method has the form [ML03]

$$\frac{\| (R_{\nu,\nu} \left(\frac{\mathbf{z}}{2^s} \right))^{2^s} - e^{\mathbf{z}} \|}{\|\mathbf{z}\|} \leq \epsilon \|\mathbf{z}\| e^{\epsilon \|\mathbf{z}\|}. \quad (4.74)$$

4.4.3. Computation of $\phi_k(\mathbf{H})$ -functions

The numerical schemes discussed in Sec. 4.3 involve not only the computation of the matrix exponential $e^{\mathbf{z}}$, but also the computation of $\phi_k(\mathbf{z})$ -functions (see Def. 4.3.2). There exist different ways to compute these functions, here only two widely used techniques are presented.

Theorem 4.4.2. *Let $\mathbf{z} \in \mathbb{R}^{m,m}$, $\mathbf{v} \in \mathbb{R}^m$ and*

$$\tilde{\mathbf{z}} = \begin{pmatrix} \mathbf{z} & \mathbf{v} & 0 & \cdots & 0 \\ & 0 & 1 & \ddots & \vdots \\ & & 0 & \ddots & 0 \\ & & & \ddots & 1 \\ 0 & & & & 0 \end{pmatrix}, \quad (4.75)$$

then the exponential of the extended matrix $\tilde{\mathbf{z}} \in \mathbb{R}^{m+p, m+p}$ is given by

$$e^{\tilde{\mathbf{z}}} = \begin{pmatrix} e^{\mathbf{z}} & \phi_1(\mathbf{z}) \mathbf{v} & \phi_2(\mathbf{z}) \mathbf{v} & \cdots & \phi_p(\mathbf{z}) \mathbf{v} \\ & 1 & \frac{1}{1!} & \cdots & \frac{1}{(p-1)!} \\ & & 1 & \ddots & \vdots \\ & & & \ddots & \frac{1}{1!} \\ 0 & & & & 1 \end{pmatrix}, \quad (4.76)$$

Proof. The proof of this theorem can be found in [Sid98]. \square

A first method raises from Theorem 4.4.2. Thus, for example, the matrix function $\phi_1(\mathbf{z})$ can be compute using [Saa92]

$$\exp \begin{pmatrix} \mathbf{z} & \mathbf{v} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} e^{\mathbf{z}} & \phi_1(\mathbf{z}) \mathbf{v} \\ 0 & 1 \end{pmatrix}. \quad (4.77)$$

Although Theorem 4.4.2 is clearly useful because allows to compute simultaneously the matrix exponential $e^{\mathbf{z}}$ and the first p ϕ -functions, or more precisely the product of this functions times the vector \mathbf{v} , numerically it can exhibit problems because it involves the computation of the exponential of a nonsymmetric matrix.

A second method, presented in [Saa92], is restricted to the computation of $\phi_1(\mathbf{z})$ -function. This method proposes an approximation to $\phi_1(\mathbf{z})$ based on rational Chebyshev polynomials

$$\phi_1(\mathbf{z}) \approx R_{\nu, \nu}(\mathbf{z}) = \sum_{i=1}^{\nu} \frac{\alpha_i}{\lambda_i} (\mathbf{z} - \lambda_i \mathbb{I})^{-1}, \quad (4.78)$$

where the coefficients α_i and the poles λ_i used for Eq. 4.63 can be employed also here.

4.4.4. Reorthogonalization process

The Arnoldi's algorithm previously presented contains a modified Gram-Schmidt sequence (see lines 4–7 in Algorithm 2) in order to produce an orthonormal base of \mathcal{K}_m . It is well known that in the context of exact arithmetic this procedure ensures Eq. 4.52, i.e.

$$\mathbf{V}^T \mathbf{V} - \mathbb{I} = 0,$$

however, using finite arithmetic precision this algorithm can produce severe loss of orthogonality in the calculated vectors [GLR05]. In fact, if $\mathbf{Q} = (Q_1, \dots, Q_M) \in \mathbb{R}^{N_u \times M}$ represents the original subspace, the orthogonality error introduced by the modified Gram-Schmidt algorithm at the iteration j is bounded by [BP92]

$$\|\mathbf{V}^T \mathbf{V} - \mathbb{I}\| \leq \zeta(N_u, j) \varepsilon \kappa(\mathbf{Q}_j), \quad (4.79)$$

where ε is the machine precision, $\kappa(\mathbf{Q}_j)$ is the condition number of the matrix $\mathbf{Q}_j = (Q_1, \dots, Q_j)$ and $\zeta(N_u, j)$ is a polynomial of low degree.

An option to reduce this error in Arnoldi's algorithm, getting orthogonal vectors to full working precision, is including additional reorthogonalization stages. In [GS89] reorthogonalization steps in Arnoldi's algorithm are recommended in case of non-symmetric matrices. Normally one additional reorthogonalization step is enough [GLR05], but sometimes rounding errors persist in this correction and some other reorthogonalization steps could

be necessary [HRT07]. The Arnoldi algorithm with selective reorthogonalization proposed in [HRT07], and included in Algorithm 3, is used in the present work.

Algorithm 3: Arnoldi's algorithm with reorthogonalization.

Data: z, \mathbf{v}
Result: \mathbf{V}, \mathbf{H}

```

1  $V_1 = \mathbf{v} / \|\mathbf{v}\|$ ;
2 for  $j = 1 : m$  do
3    $\mathbf{p} = zV_j$ ;
4    $\rho = \|\mathbf{p}\|$ ;
5   for  $(i = 1 : j)$  do
6      $H_{i,j} = V_i^T \mathbf{p}$ ;
7      $\mathbf{p} = \mathbf{p} - H_{i,j}V_i$ ;
8   end
9    $H_{j+1,j} = \|\mathbf{p}\|$ ;
10  while  $H_{j+1,j} < \eta\rho$  do
11    for  $(i = 1 : j)$  do
12       $c = V_i^T \mathbf{p}$ ;
13       $H_{i,j} = H_{i,j} + c$ ;
14       $\mathbf{p} = \mathbf{p} - cV_i$ ;
15    end
16     $H_{j+1,j} = \|\mathbf{p}\|$ ;
17  end
18   $V_{j+1} = \mathbf{p} / H_{j+1,j}$ ;
19 end

```

The lines 10–17 in Algorithm 3 represent the additional reorthogonalization steps, which are developed if the condition

$$H_{j+1,j} < \eta\rho$$

is satisfied. Here, according to [DGKS76], the parameter η will be taken equal to $1/\sqrt{2}$.

4.4.5. Error control and size of the Krylov subspace

Theorem 4.4.3. *Let \mathcal{A} be a matrix with numerical range contained in the disk $|z + \varrho| \leq \varrho$, then the error in the Arnoldi approximation of $e^{\tau\mathcal{A}}\mathbf{v}$ is bounded by*

$$\|e^{\tau\mathcal{A}}\mathbf{v} - \mathbf{V}e^{\tau\mathbf{H}}\mathbf{e}_1\| \leq 12e^{-\varrho\tau} \left(\frac{e^{\varrho\tau m}}{m} \right), \quad m \geq 2\varrho\tau. \quad (4.80)$$

Proof. The proof of this theorem can be consulted in [HL97]. □

As was discussed previously, the idea with the Arnoldi algorithm is to build a relatively small Krylov subspace, i.e. with $m \ll N_u$, to approximate the exponential operations employing a reduced Hessenberg matrix \mathbf{H} instead of a large-scale system matrix \mathcal{A} . Here, the size of the Krylov subspace built should be large enough to ensure a rounding error, stemmed from the matrix exponential approximation, much less than the error introduced by the time marching scheme.

From Theorem 4.4.3 it follows that, in the rough sense, the error in the approximation of the matrix exponential using Arnoldi's decomposition is bounded by $C(m)\tau^m$, where $C(m)$ is a function of m . This means that the error in the projection can be reduced limiting the time-step size or increasing the number of vectors in the Krylov subspace. In the present work the time-step size will be considered constant for each problem, therefore the error control will be based on the size of \mathcal{K}_m .

In order to limit adequately the number of vectors in \mathcal{K}_m , a stopping-criterion was introduced in Arnoldi's algorithm. Hochbruck et al. in [HLS98] suggested as stopping-criterion the condition

$$\text{err} = \tau \|\mathbf{v}\| \tau H_{m+1,m} \left| [(\mathbb{I} - \tau \mathbf{H})^{-1} \phi_1(\tau \mathbf{H})]_{m,1} \right| \|V_{m+1}\|_{\text{tol}} < 0.1. \quad (4.81)$$

Here,

$$\|d\|_{\text{tol}} = \left(\frac{1}{n} \sum_{i=1}^n \left(\frac{d_i}{w_i} \right)^2 \right)^{\frac{1}{2}}, \quad (4.82)$$

$$w_i = a_{\text{tol}} + \max(|\mathbf{w}_{n,i}|, |\mathbf{w}_{n-1,i}|) r_{\text{tol}}, \quad (4.83)$$

and, a_{tol} and r_{tol} are the given absolute and relative error tolerances, respectively.

On the other hand Sidje [Sid98] proposed to stop the iterations in Arnoldi's algorithm if

$$\text{err} \leq a_{\text{tol}}, \quad (4.84)$$

where the error is computed in accordance with Algorithm 4.

Because in Algorithm 4 no inverse matrix should be computed, and besides the matrix-

Algorithm 4: Estimation of the error for the Arnoldi algorithm.

Data: $\beta, \mathcal{A}, \mathbf{H}, \mathbf{V}, \tau$

Result: err

```

1 err1 =  $\beta |H_{m+1,m} \tau \mathbf{e}_m^T \phi_1(\tau \mathbf{H}) \mathbf{e}_1|$ ;
2 err2 =  $\beta |H_{m+1,m} \tau^2 \mathbf{e}_m^T \phi_2(\tau \mathbf{H}) \mathbf{e}_1| \|AV_{m+1}\|_2$ ;
3 if err1  $\gg$  err2 then
4   | err = err2;
5 else if err1 > err2 then
6   | err = err2 / (1 - (err2/err1));
7 else
8   | err = err1;
9 end

```

vector products involved ($\phi_1(\tau \mathbf{H}) \mathbf{e}_1$ and $\phi_2(\tau \mathbf{H}) \mathbf{e}_1$) can be computed using Theorem 4.4.2 in only one operation, in the present work the stopping-criterion in Eq. 4.84 was implemented. Additionally, if the condition Eq. 4.84 is satisfied the term $\phi_1(\tau \mathbf{H}) \mathbf{e}_1$ used to evaluate the error can be used latter to compute Eq. 4.60. Here, the tolerance value a_{tol} used by default is $1\text{E} - 7$.

Thus, Algorithm 3 can be modified to get Algorithm 5 used in all examples in this work. Note that in line 22 of Algorithm 5 the error is not computed in each iteration of the algorithm. Here, as in [HL97], to save computational time the stopping criteria is only computed for some steps in the iteration process.

Algorithm 5: Arnoldi's algorithm with reorthogonalization and stop condition.

Data: z, \mathbf{v}
Result: \mathbf{V}, \mathbf{H}

```

1  $V_1 = \mathbf{v} / \|\mathbf{v}\|$ ;
2  $err = 1$ ;
3  $j = 1$ ;
4 while  $err \geq a_{tol}$  do
5    $\mathbf{p} = zV_j$ ;
6    $\rho = \|\mathbf{p}\|$ ;
7   for  $(i = 1 : j)$  do
8      $H_{i,j} = V_i^T \mathbf{p}$ ;
9      $\mathbf{p} = \mathbf{p} - H_{i,j}V_i$ ;
10  end
11   $H_{j+1,j} = \|\mathbf{p}\|$ ;
12  while  $H_{j+1,j} < \eta\rho$  do
13    for  $(i = 1 : j)$  do
14       $c = V_i^T \mathbf{p}$ ;
15       $H_{i,j} = H_{i,j} + c$ ;
16       $\mathbf{p} = \mathbf{p} - cV_i$ ;
17    end
18     $H_{j+1,j} = \|\mathbf{p}\|$ ;
19  end
20   $V_{j+1} = \mathbf{p} / H_{j+1,j}$ ;
21  if  $j \in \{1, 2, 3, 4, 6, 8, 11, 15, 20, 27, 34, 42, 53, 67, 84, 106, 133, 167, 211\}$  then
22    Compute the error ( $err$ ) from Algorithm 4;
23  end
24   $j = j + 1$ ;
25 end

```

5. Compressible Navier-Stokes solver

In order to simulate the rapid expansion of a supercritical CO₂-flow a generic Navier–Stokes compressible flow solver was implemented and tested. The solver was written in C++ language and it uses several libraries as `deal.ii` [BHK07], `Trilinos` [HBH⁺05] and Intel[®] TBB [Int16].

5.1. Characteristics of the compressible Navier-Stokes solver

In a brief description, the solver implemented has the following characteristics:

- Spatial discretization using *DG*-elements for 2D/3D or axisymmetric models. Dimension independent structure of `deal.ii` library (based on templates) allowed to develop a generic solver for 2D/3D spatial discretizations.
- Use of both structure and unstructured quadrilateral (or hexahedral for 3D cases) meshes is possible.
- Generic programming incorporated in `deal.ii` library allowed to use both bilinear \mathbb{P}^1 or higher order polynomial basis. Likewise, it is possible to select linear or higher order mapping for curvilinear boundaries. Thus, eventually it is possible to work with subparametric, isoparametric or superparametric formulations.
- Global or adaptive mesh refinement are available. The used adaptive strategy is presented in Sec. 3.5. Use of the gradient of density, velocity, Mach number or pressure as refinement indicator is possible.
- Multiple time marching scheme can be used. Very efficient exponential time integrators as the exponential Rosenbrock–Euler method (see Sec. 4.3.2.2) or the *exp4* method (see Sec. 4.3.2.3) can be selected. Explicit schemes as the strong stability preserving Runge–Kutta methods (*SSPRK*) or the forward Euler method are also available. The backward Euler method is the only implicit scheme included.
- The Jacobian matrix, required in the exponential time integration, is computed through automatic differentiation tools from `Sacado-Trilinos` are used for these computations [HBH⁺05]. This technique allows to compute exact Jacobians with a reduced computational cost [LT13].
- Computation of thermophysical properties of the fluid can be done using the ideal gas assumption. Real gas formulation is available only for CO₂. In this last case,

the extended generalized Bender EOS (*egB-EOS*) (see Sec. 2.99) is used to model the gas behavior. A computationally less efficient version for real gases using the *PROPATH* library [PRO08] is also available.

- Compressible flow problems can exhibit discontinuous solutions. Therefore, the shock capturing strategy described in Sec. 3.4 was implemented in the code. Discontinuity indicators based on the jumps of density, Mach number, pressure, or velocity can be chosen. Multiple indicators can also be used.
- Multiple numerical flux formulations are included: Lax–Friedrichs (see Sec. 3.3.1.1), Lax–Wendroff, Vijayasundaram (see Sec. 3.3.2.2), Steger–Warming and *AUSM⁺-up* (see Sec. 3.3.3). A *FCT* formulation (see Sec. 3.3.4) for the numerical flux can be used, too.
- A parallelization strategy based on shared memory is used in the code. Intel[®] TBB - library is employing for scheduling tasks to available threads.

A summary of these characteristics is included in Fig. 5.1.

5.2. Algorithm description

In this section a summary of the algorithms and equations used in the implemented solver is presented. In general, known the vector \mathbf{w}_n , the computation of the next time step follows these steps:

- Compute the discontinuity indicator. In this work the value of the discontinuity indicator g for each cell of the current triangulation is calculated using Eq. 3.71.
- Define the G -switch value for each cell. A threshold value for the discontinuity indicator g_{lim} is defined (as in Eq. 3.72) and the value of G for the i -th cell is computed according to

$$G(i) = \begin{cases} 0, & \text{if } g(i) < g_{\text{lim}}, \\ 1, & \text{otherwise.} \end{cases}$$

- Compute the term $\mathbf{b}_h(\mathbf{w}_h(t_n), \boldsymbol{\varphi})$ in Eq. 3.116. If a hybrid numerical flux is used (see Sec. 3.3.4), then the G variable replaces Θ in Eq. 3.70.

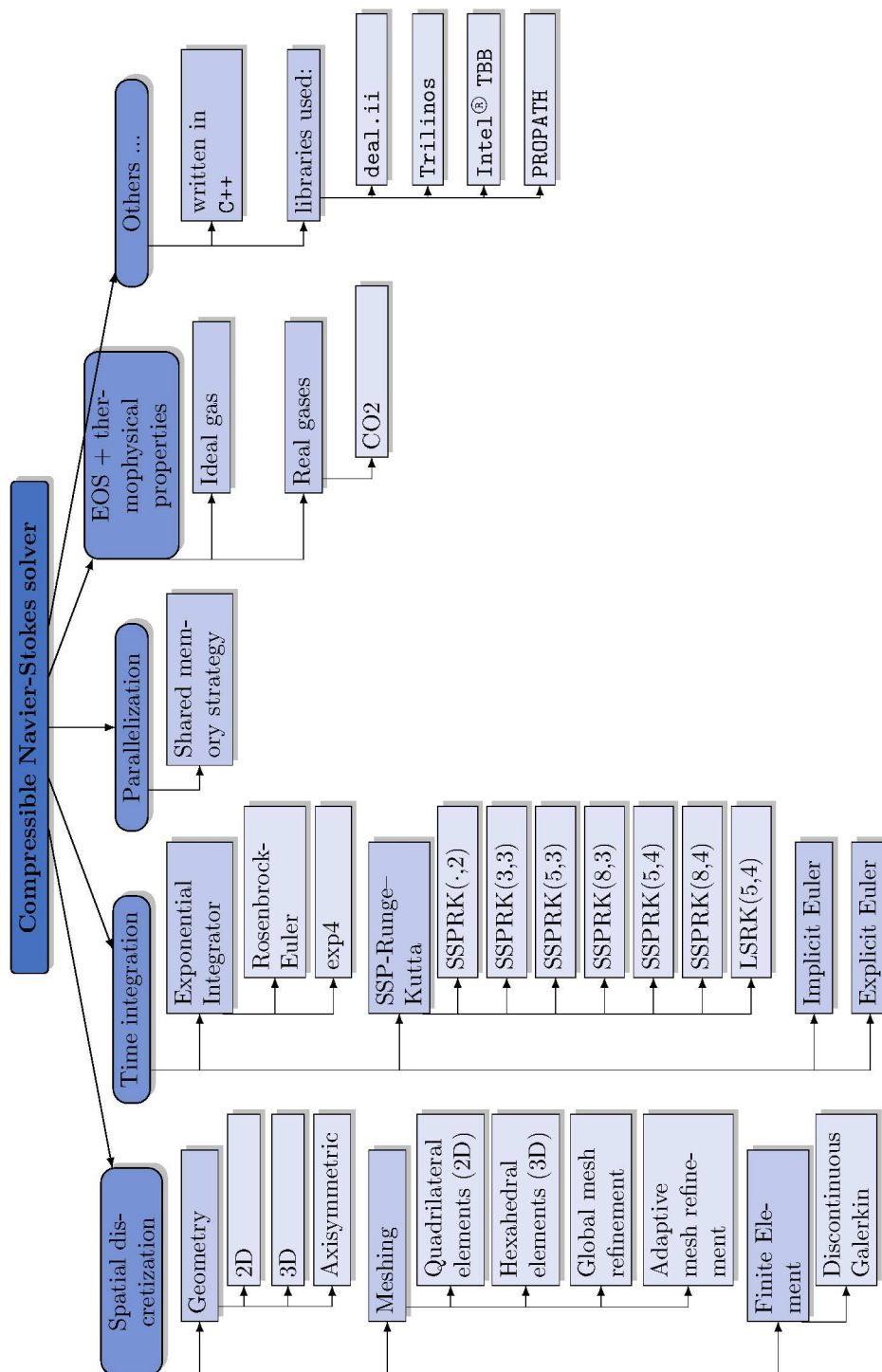
For a non-interior element K_i with face Γ_{ij} such that $j \in \kappa(i)$, the term \mathbf{w}^- in the numerical flux $\mathbf{H}(\mathbf{w}^+, \mathbf{w}^-, \mathbf{n})$ is defined following the considerations presented in Sec. 3.6. Here, the eigenvectors and eigenvalues of the matrix \mathbb{A}_1 are computed with the expressions in Lemma 2.2.6. Likewise, the derivative $\frac{\partial p}{\partial \rho}$ is taken from the analytic expression in Sec. B and the derivative $\frac{\partial p}{\partial e}$ is calculated from Eq. 2.55, i.e.,

$$\frac{\partial p}{\partial e} = \left(c^2 - \frac{\partial p}{\partial \rho} \right) \frac{\rho^2}{p}.$$

For the special case of an ideal gas, the simplifications included in Remark 2.2.7 should be considered.

- Compute the term $\mathbf{a}_h(\mathbf{w}_h(t_n), \boldsymbol{\varphi})$ in Eq. 3.117. Here, the flux terms $\mathbb{F}_k^v(\mathbf{w}_h, \nabla \mathbf{w}_h)$ are calculated as in Sec. 3.7.3. In the formulation of this term the non-symmetric interior penalty method (*NIPG*) was employed (see Sec. 3.7.1 and Sec. 3.7.2).
- Compute the term $\mathbf{J}_h(\mathbf{w}_h(t_n), \boldsymbol{\varphi})$ defined in Eq. 3.118 plus the stabilization term in Eq. 3.76.

Figure 5.1.: Characteristics of the solver developed.



- Compute the term $\beta_h(\mathbf{w}_h(t_n), \boldsymbol{\varphi})$ in Eq. 3.119 along with the stabilization term in Eq. 3.75. For a non-interior element K_i with face Γ_{ij} such that $j \in \kappa^D(i)$, the vector \mathbf{w}^D is estimated in accordance with Sec. 3.7.4.
- Compute the operation $\mathfrak{B}(\mathbf{w}_n)$ in Eq. 4.1.
- Compute the mass matrix \mathbf{M} in Eq. 4.1.
- Compute the operation $\mathfrak{F}(\mathbf{w}_n) = \mathbf{M}^{-1}\mathfrak{B}(\mathbf{w}_n)$ in Eq. 4.2.
- The Jacobian matrix

$$\mathcal{A}(\mathbf{w}_n) = -\frac{d\mathfrak{F}}{d\mathbf{w}}(\mathbf{w}_n)$$

is computed employing automatic differentiation via the `Sacado-Trilinos` library.

- Given $\mathcal{A}(\mathbf{w}_n)$ and τ , the corresponding Krylov subspace (\mathbf{V}) and the Hessenberg matrix (\mathbf{H}) are obtained using Algorithm 5. The stopping criterion for the Arnoldi algorithm is given in Eq. 4.84 and Algorithm 4. By default, the error tolerance a_{tol} is set to $1\text{E} - 7$.
- The term

$$\phi_1(-\tau\mathcal{A}_n)\mathfrak{F}(\mathbf{w}_n)$$

is computed using Eq. 4.53, i.e.,

$$\phi_1(-\tau\mathcal{A}_n)\mathfrak{F}(\mathbf{w}_n) \approx \|\mathfrak{F}(\mathbf{w}_n)\|\mathbf{V}\phi_1(\tau\mathbf{H})\mathbf{e}_1,$$

where the matrix-vector product $\phi_1(\tau\mathbf{H})\mathbf{e}_1$ is calculated using Eq. 4.76 in Theorem 4.4.2. Here the matrix exponential is approximated by the rational Chebyshev method in Eq. 4.65.

- Finally, the state vector \mathbf{w}_{n+1} is computed using the exponential Rosenbrock–Euler method in Eq. 4.41, i.e.,

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \tau\phi_1(-\tau\mathcal{A}_n)\mathfrak{F}(\mathbf{w}_n).$$

- As general note, when the temperature value is required, e.g. to compute the viscosity in Eq. 2.109, we proceed using the values of density ρ and specific internal energy e , which are known from the conservative vector \mathbf{w} , and with the analytic expressions for $e(\rho, \theta)$ and $\partial e/\partial\theta$ in Sec. A, we compute the temperature value through the Newton–Raphson root-finding method.

5.3. Numerical tests

In order to validate the implemented code, multiple benchmark problems were simulated. The problems presented in the following involve flows in a wide range of Mach regimes: subsonic, transonic and supersonic cases are shown. Finally, in Sec. 5.4 the achieved results in the simulation of the rapid expansion of supercritical CO_2 are reported.

5.3.1. Two dimensional rotating Gaussian pulse

Rotating functions are problems commonly used in order to evaluate the performance of numerical schemes, e.g. stability, numerical diffusion, spurious oscillations or phase problems. Here a two dimensional rotating Gaussian pulse problem is introduced to validate the numerical scheme presented in Sec. 4 and Sec. 5. This problem, also used in [WDE⁺99], is defined through the equation

$$\frac{\partial w}{\partial t} - 0.0001\Delta w + \text{div}(\mathbf{v}w) = 0, \quad \text{in } \mathcal{M} = \Omega \times (0, T), \quad (5.1)$$

with $w(\mathbf{x}, t) \in \mathbb{R}$, $\Omega = \{\mathbf{x} \in \mathbb{R}^2 : x_i \in (-0.5, 0.5), i = 1, 2\}$, $\mathbf{v}(\mathbf{x}) = [-4x_2 \ 4x_1]^T$, $T = \pi/2$, and endowed with the initial condition

$$w(\mathbf{x}, 0) = \exp\left(-\frac{(x_1 + 0.25)^2 + x_2^2}{0.004}\right) \quad \text{for } \mathbf{x} \in \Omega. \quad (5.2)$$

The analytical solution for this problem is given by the function

$$u(\mathbf{x}, t) = \frac{1.0}{1.0 + 0.1t} \exp\left(-\frac{(\bar{x}_1 + 0.25)^2 + \bar{x}_2^2}{0.004(1 + 0.1t)}\right), \quad (5.3)$$

where

$$\bar{x}_1 = x_1 \cos(4t) + x_2 \sin(4t) \quad \text{and} \quad \bar{x}_2 = -x_1 \sin(4t) + x_2 \cos(4t). \quad (5.4)$$

Here, Dirichlet boundary conditions, defined by Eq. 5.3, are employed.

This problem was solved using four different quadrilateral meshes with 20×20 elements, 40×40 elements, 80×80 elements and 160×160 elements, respectively. For the time integration procedure, twenty time stepping configurations were used: nine configurations using implicit backward Euler and time steps given by $\{\frac{T}{20000}, \frac{T}{15000}, \frac{T}{10000}, \frac{T}{6000}, \frac{T}{4000}, \frac{T}{2000}, \frac{T}{1000}, \frac{T}{500}, \frac{T}{200}\}$, as well as eleven configurations using the Rosenbrock–Euler method with time steps given by $\{\frac{T}{2000}, \frac{T}{1000}, \frac{T}{500}, \frac{T}{200}, \frac{T}{100}, \frac{T}{50}, \frac{T}{25}, \frac{T}{10}, \frac{T}{5}, \frac{T}{2}, T\}$. In all cases the Vijayasundaram numerical flux (see Sec. 3.3.2.2) was used for the advection term, as well as the *NIPG* formulation for the diffusion term with $\sigma = 0.1$ (see Sec. 3.7.1).

In Fig. 5.2 eight depictive cases are contrasted: two grids with 20×20 and 160×160 \mathbb{P}^1 *DG*-elements, as well as four time marching procedures, two of these employing the implicit Euler method and other two cases employing the Rosenbrock–Euler method. In all these cases the approximations are plotted in $t = T$. In Fig. 5.3 a similar comparisons are presented for \mathbb{P}^3 *DG*-elements. Further, in Fig. C.1 results using \mathbb{P}^2 *DG*-elements are shown. As is suggested in these figures, the analysis of the results found will focus on the error introduced by the spatial semidiscretization, as well as the error produced by the time marching schemes.

For a qualitative analysis, comparing the contour plots found in Fig. 5.2 and Fig. 5.3, using the Rosenbrock–Euler scheme with $\Delta t = \frac{T}{2000}$ and $\Delta t = T$, for a given mesh, no difference can be noted. Even more, no difference is found comparing the maximum and minimum values for w . Now, comparing the approximations built using both coarsest and the finest mesh, using Rosenbrock–Euler scheme for a given Δt , is clear that increasing the number of unknowns raise the resolution of the approximation and bring the peak values closer to the analytical value of about 0.8642. These observations allow to suspect that the error in these approximations, computed employing *DG*-elements in space and Rosenbrock–Euler as time integrator, is dominated by the error component introduced by the spatial discretization. This conclusion can be verified checking the convergence graphic presented in Fig. 5.4. Here, in fact the error behavior does not depend on the size of the time step, but it is only related to the mesh size used.

For scalar hyperbolic problems discretized with arbitrary triangulations using discontinuous Galerkin, formal proofs [JP86, Pet91] provide an error estimate of the form

$$\|w_h - w\|_{L^2(\Omega)} \leq Ch^{q+\frac{1}{2}} \|w\|_{H^{q+1}(\Omega)}, \quad (5.5)$$

where h is the mesh size, q is the polynomial degree of the basis employed, and C is a positive constant independent of w and h . Although, according to this estimate, solutions with *DG* discretizations only find suboptimal rates of convergence, but optimal rates $O(h^{q+1})$

Figure 5.2.: Contours of w for the rotating Gaussian-pulse problem at $t = T$ using \mathbb{P}^1 DG-elements. Coarse mesh conformed by 20×20 elements and fine mesh formed by 160×160 elements.

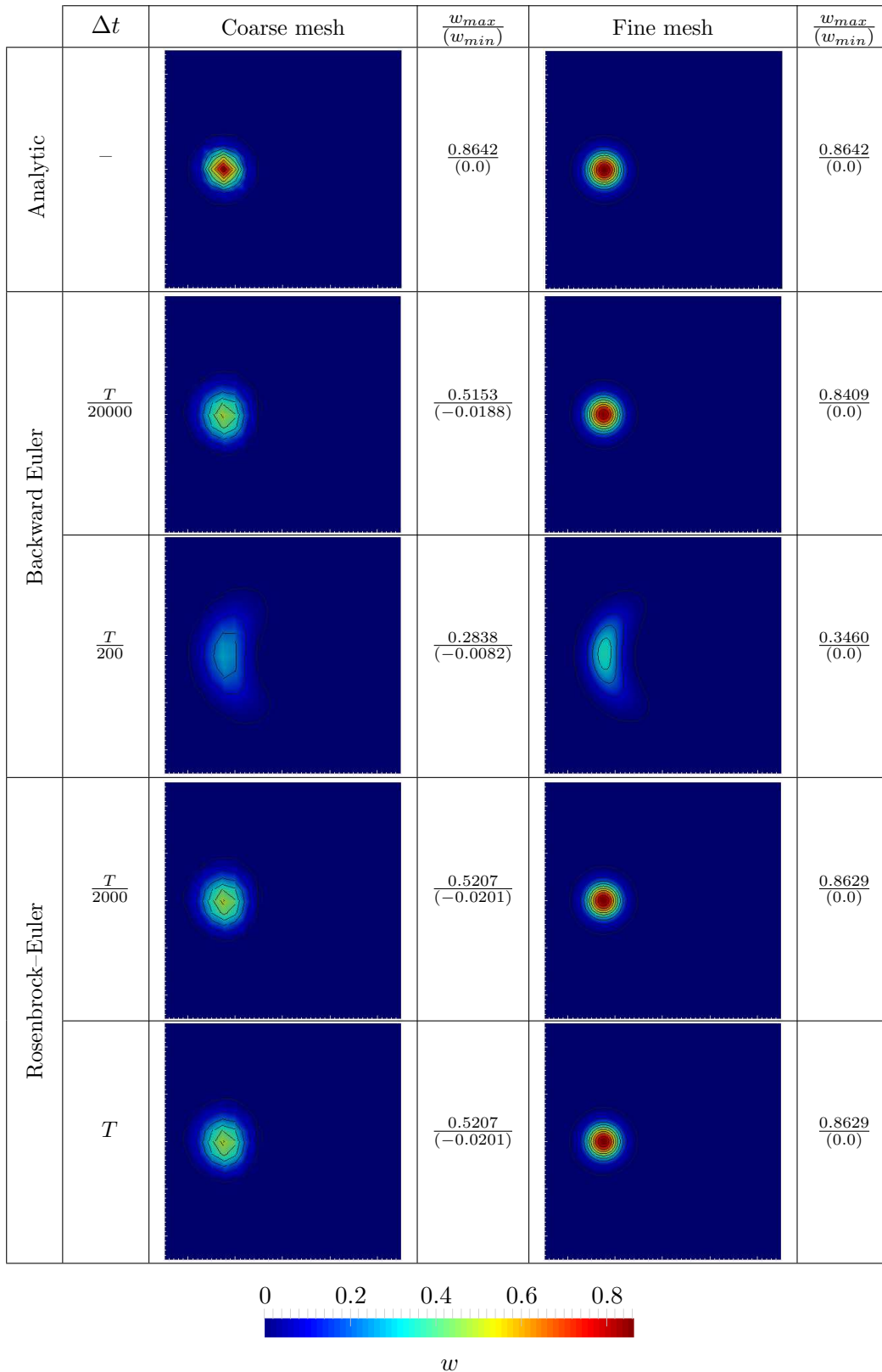


Figure 5.3.: Contours of w for the rotating Gaussian-pulse problem at $t = T$ using \mathbb{P}^3 DG-elements. Coarse mesh conformed by 20×20 elements and fine mesh formed by 160×160 elements.

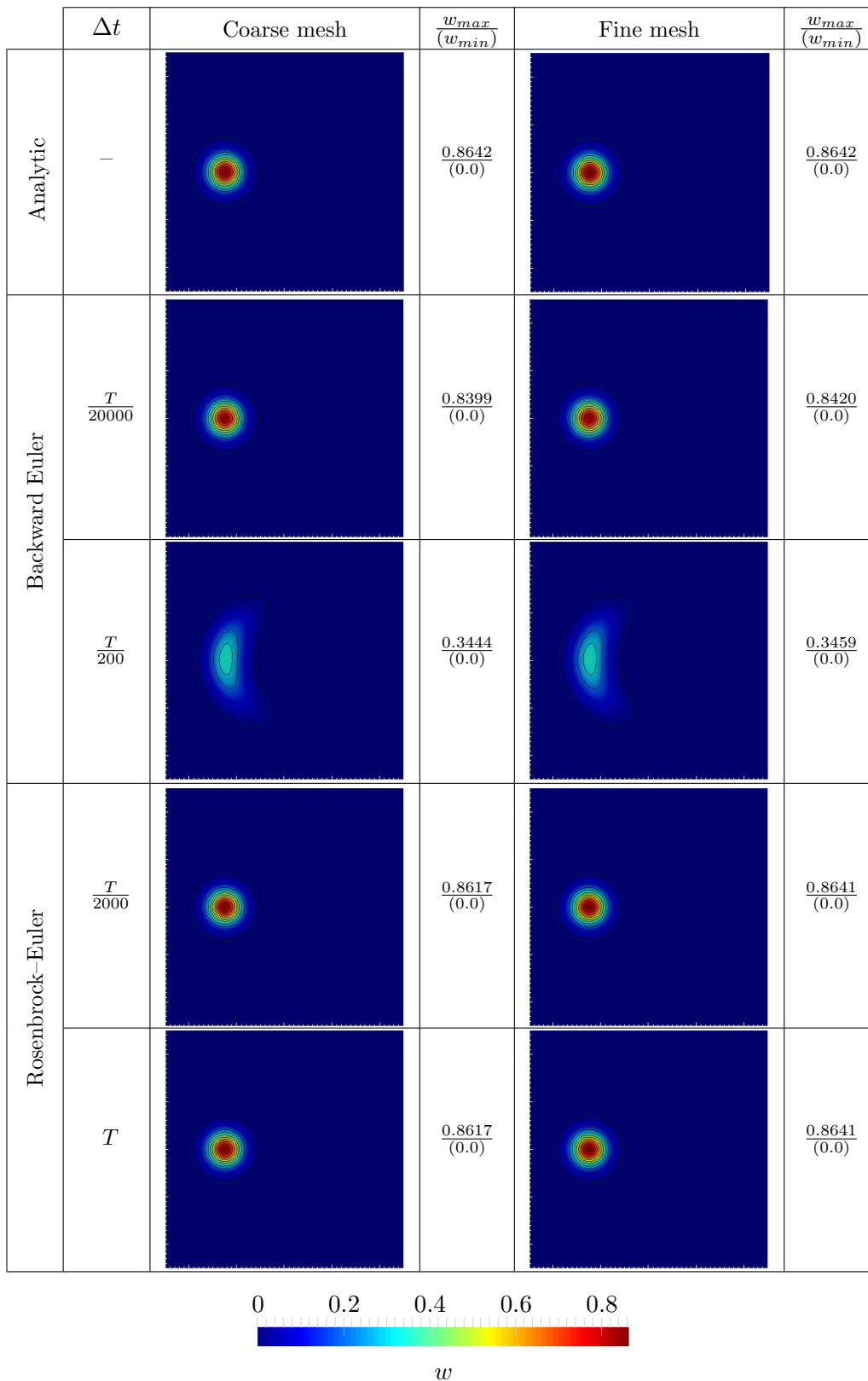
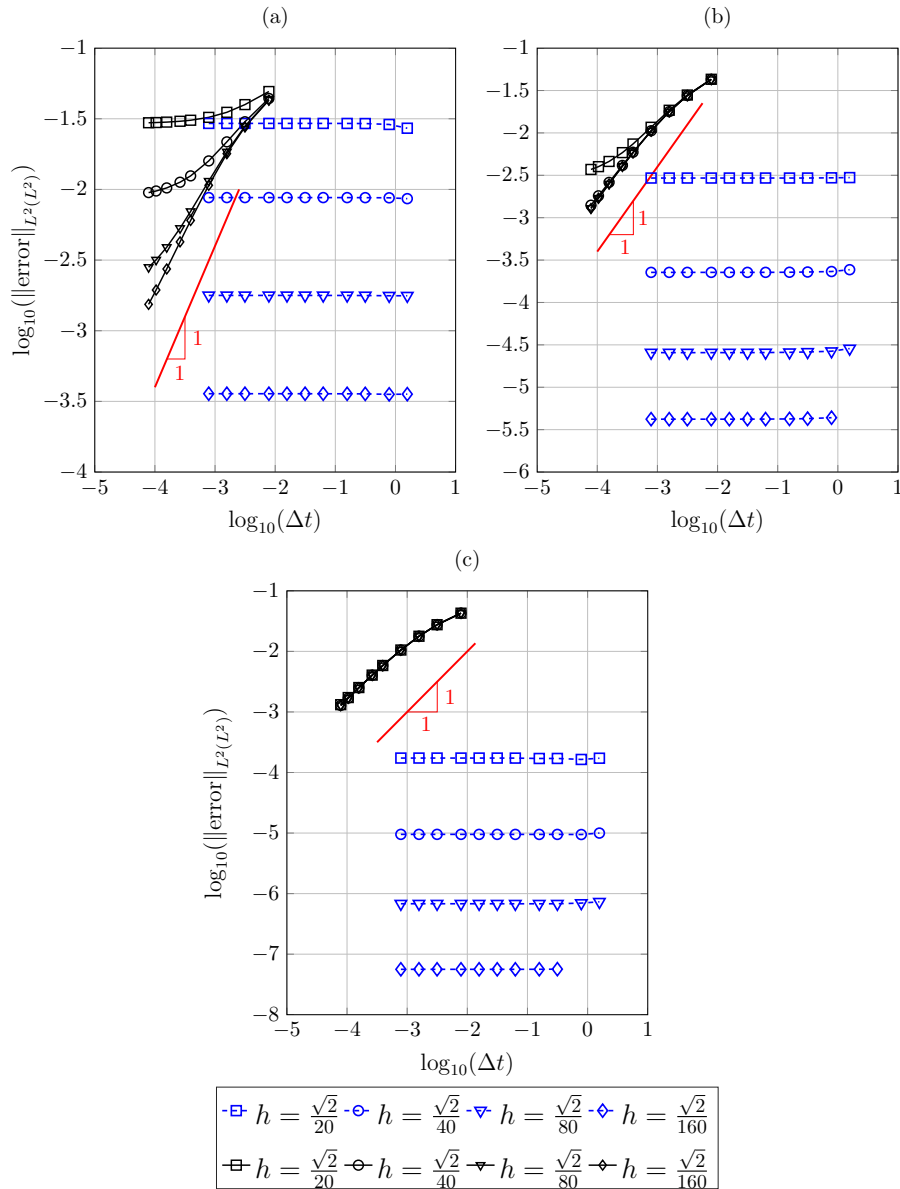


Figure 5.4.: Experimental order of convergence in time for the rotating Gaussian pulse problem using (a) \mathbb{P}^1 , (b) \mathbb{P}^2 and (c) \mathbb{P}^3 DG -elements. Continuous lines represent solutions with the backward Euler method, whereas dashed lines represent solutions that employ the Rosenbrock–Euler method.



are commonly found in practice [JJS95]. For example some works report optimal convergence rates using special grids [Ric88, CDG08, Ric08], or even superconvergent solutions of order $2q + 1$ or higher [AH09].

For diffusion-advection problems Cookburn and Shu [CS98a] reported, for local discontinuous Galerkin (*LDG*), an error estimate with a suboptimal rate of convergence $O(h^q)$ in energy norm. This result was corroborated in [CD99] for problems with variable diffusion coefficient and velocity field, in a bounded spatial domain. Despite the theoretical estimation in [CS98a], we present in this work several numerical experiments where, with the suitable choice of a numerical flux, optimal convergence rates can be found. Later, in [Cas00] it was mathematically proved that for the linear diffusion-advection problems with constant coefficients, the *LDG* method can converge with an optimal rate $O(h^{q+1})$ using an appropriate numerical flux.

In order to determine the convergence rate of the implemented *DG* formulation, the space-time error in the L^2 -norm was computed for different grids. The error graphics for \mathbb{P}^1 , \mathbb{P}^2 and \mathbb{P}^3 *DG*-elements are presented in Fig. 5.5, and the corresponding convergence rates computed are reported in Table 5.1. These results show that, for approximations computed using the Rosenbrock–Euler scheme, the behavior of the error does not depend on Δt , and it is dominated by the error generated by the spatial discretization. Additionally, it is possible to observe that this spatial error has, in all cases analyzed, an order of convergence near to the optimal value $O(h^{q+1})$. As was discussed previously, although some theoretical error estimates predict an suboptimal convergence rate for diffusion-advection problems discretized with *DG* elements, optimal experimental order of convergence are frequently reported in the literature.

	\mathbb{P}^1		\mathbb{P}^2		\mathbb{P}^3	
h	$\ \text{error}\ $	EOC	$\ \text{error}\ $	EOC	$\ \text{error}\ $	EOC
7.07 ₋₂	2.93 ₋₂		2.94 ₋₃		1.73 ₋₄	
3.54 ₋₂	8.76 ₋₃	1.74	2.26 ₋₄	3.70	9.52 ₋₆	4.19
1.77 ₋₂	1.78 ₋₃	2.30	2.56 ₋₅	3.14	6.80 ₋₇	3.81
8.84 ₋₃	3.58 ₋₄	2.31	4.20 ₋₆	2.61	5.62 ₋₈	3.60

Table 5.1.: Experimental order of convergence (*EOC*) for rotating Gaussian pulse problem using *DG*-elements and Rosenbrock–Euler method. Consider

$$\|\text{error}\| = \|w_h - w\|_{L^2(L^2)}.$$

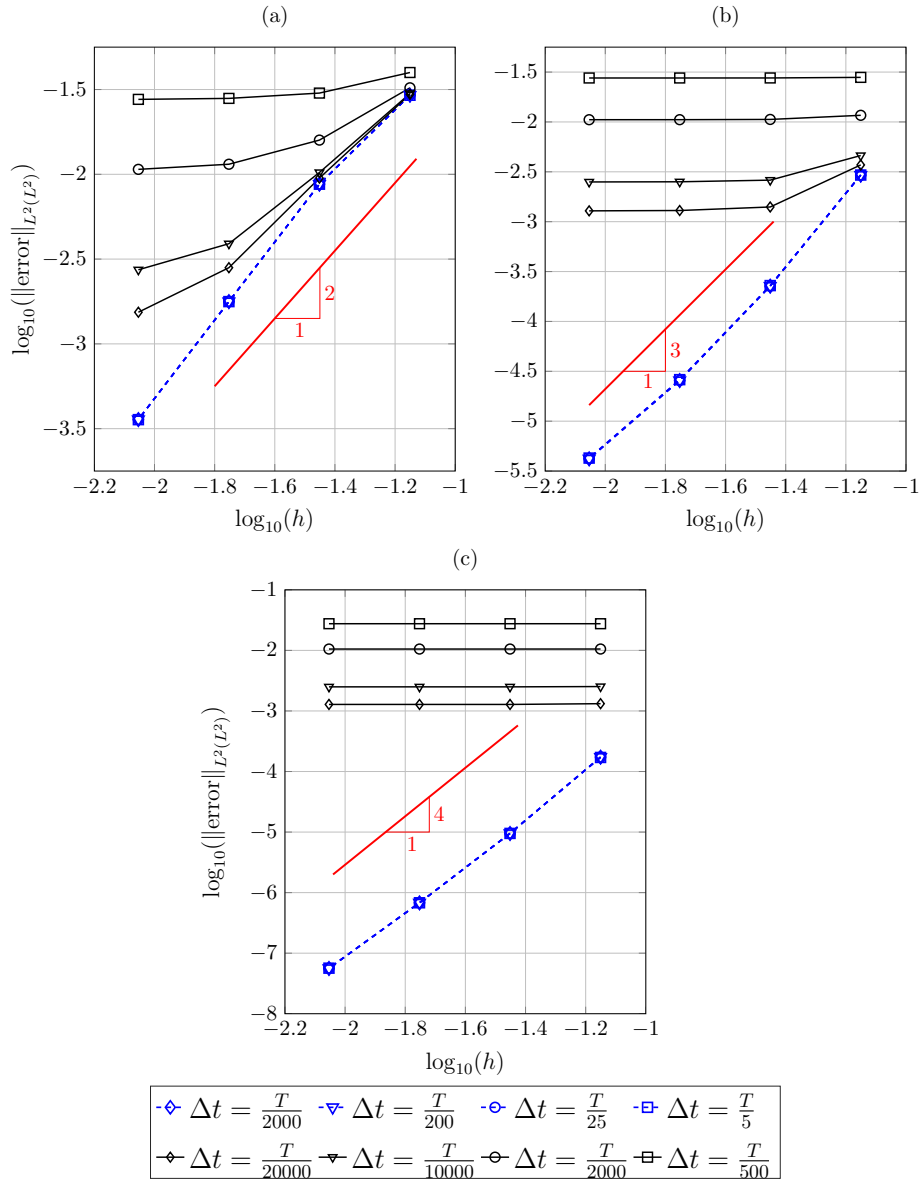
From the last results, it is possible to conclude that, using numerical scheme conformed by *DGFEM* in space and Rosenbrock–Euler for time marching, for this linear case it is possible to employ a time step as long as $\Delta t = T$ without detriment of the precision or restrictions in the CFL number. For this case, the CFL number is computed as

$$\text{CFL} = \frac{|\mathbf{v}|\Delta t}{h}. \quad (5.6)$$

In Table 5.2 the maximum values for the CFL numbers, computed using the maximum velocity $|\mathbf{v}| = 2\sqrt{2}$, are listed for the cases of Fig. 5.2. Here, e.g. a relatively high *Courant* number ($\text{CFL} \approx 500$) could be used, conserving a high accuracy in time integration and without loss of stability in the numerical scheme.

Although, from the theoretical point of view (see Sec. 4.3.2.2), the Rosenbrock–Euler

Figure 5.5.: Experimental order of convergence in h for the rotating Gaussian pulse problem using (a) \mathbb{P}^1 , (b) \mathbb{P}^2 and (c) \mathbb{P}^3 DG -elements. Continuous lines represent solutions with the backward Euler method, whereas dashed lines represent solutions that employ the Rosenbrock–Euler method.



		CFL number	
		Δt	
		Coarse mesh	Fine mesh
Backward Euler	$\frac{T}{20000}$	$\frac{\pi}{1000}$	$\frac{\pi}{125}$
	$\frac{T}{200}$	$\frac{\pi}{10}$	$\frac{4\pi}{5}$
Rosenbrock–Euler	$\frac{T}{2000}$	$\frac{\pi}{100}$	$\frac{2\pi}{25}$
	T	20π	160π

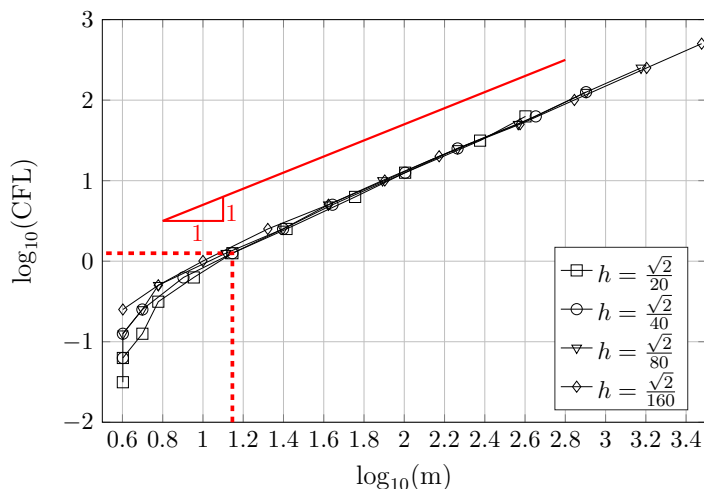
Table 5.2.: Maximum CFL values found the cases included in Fig. 5.2.

method is exact for integration of linear systems of *ODE*'s, there exist an error introduced by the approximation of the exponential matrix via the Krylov subspaces. In the numerical scheme presented, an error control strategy was used (see Sec. 4.4.5) in order to keep this error below a prescribed threshold through adapting the size of the Krylov subspace.

In Fig. 5.6, for every mesh and time steps used, the *Courant* number vs. the size of the employed Krylov subspace are plotted. Here, the CFL number was again calculated using $|\boldsymbol{v}| = 2\sqrt{2}$, which is the maximum velocity on \mathcal{M} . From this graphic it is possible to observe how the number of vectors required in the Krylov subspace is linearly proportional to the CFL number used. As expected, for a fixed mesh the amount of vectors required in the Krylov subspace is proportional to the size of the time step, and therefore proportional to the computational effort demanded. Note that, at the left of this graphic an irregular behavior is observed, but it is because an artificial limit was imposed in the error control algorithm in order to have at least 4 vectors in the Krylov subspace.

Additionally, in Fig. 5.7 is plotted, for every mesh and for the different time steps employed, the total computational cost vs. the corresponding number of vectors used in the Krylov subspace. From this graphic, it is possible to identify an optimal point at $m \approx 15$, which corresponds to $CFL \approx 1.25$. Thus, for a given mesh, CFL numbers smaller than the critical value, produce simulations with a lot of time steps, each one computationally cheaper in the tasks related to the computation of the exponential matrix (because small Krylov subspaces are enough), but with high fix costs derived mainly from the assembly process. On the other hand, CFL numbers higher than the critical one, correspond to simulation with few, but extremely expensive, time steps (because m is linear proportional to the Courant number). In this situation the fix costs in each time step are not representative compared to the cost of computing the exponential matrix function.

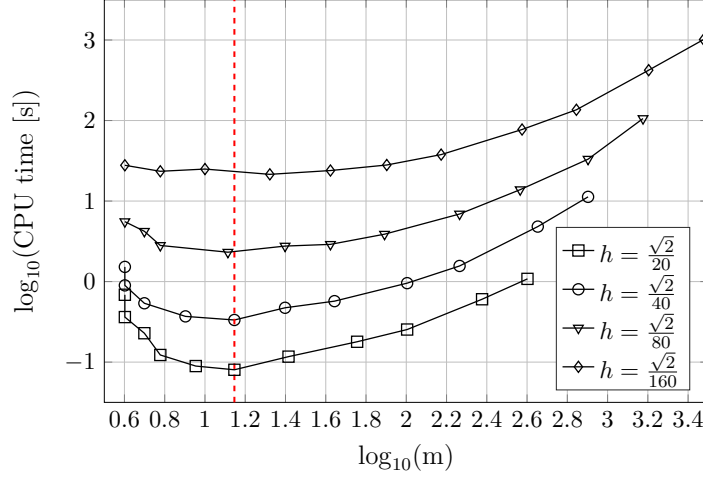
Figure 5.6.: CFL number vs. size of the Krylov subspace used for the rotating Gaussian pulse. All meshes conformed by \mathbb{P}^1 *DG*-elements.



5.3.2. Sod's shock tube problem

This test case, originally presented in [Sod78], consists of a long tube filled with an ideal gas and endowed with a thin diaphragm, located in $x_1 = 0$, that separates a high pressure-density region (ρ_L, p_L) at $x_1 < 0$, from a low pressure-density zone (ρ_R, p_R) at $x_1 > 0$. At initial time $t = 0$ the fluid in both sides is at rest, then at $t > 0$ the membrane is broken and a high speed flow is produced.

Figure 5.7.: Computational cost demanded vs. size of Krylov subspace used for the rotating Gaussian pulse. All meshes conformed by \mathbb{P}^1 DG -elements. The CPU times reported correspond to computations using 8 cores in an *Intel Core i7-3770@3.4GHz* processor.



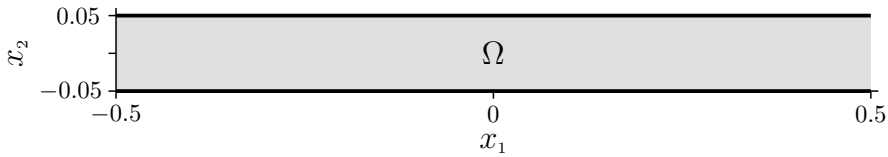
This problem is modeled using the set of Euler equations Eq. 2.40, complemented with the ideal gas equation of state. This system of equation was solved for the rectangular domain shown in Fig. 5.8, using the boundary conditions

$$\partial\Omega = \Gamma_{IN} \cup \Gamma_O \cup \Gamma_W,$$

where

- $\Gamma_{IN} = \{\mathbf{x} \in \mathbb{R}^2 : x_1 = -0.5, x_2 \in (0, 0.1)\}$ represents the inlet boundary,
- $\Gamma_O = \{\mathbf{x} \in \mathbb{R}^2 : x_1 = 0.5, x_2 \in (0, 0.1)\}$ is outlet boundary,
- $\Gamma_W = \{\mathbf{x} \in \mathbb{R}^2 : x_1 \in [-0.5, 0.5], x_2 = 0\} \cup \{\mathbf{x} \in \mathbb{R}^2 : x_1 \in [-0.5, 0.5], x_2 = 0.1\}$ defines the slip walls.

Figure 5.8.: Spatial domain defined for Sod's shock tube problem.



Likewise, the initial conditions that define the thermodynamic condition at both sides of the diaphragm, are given by

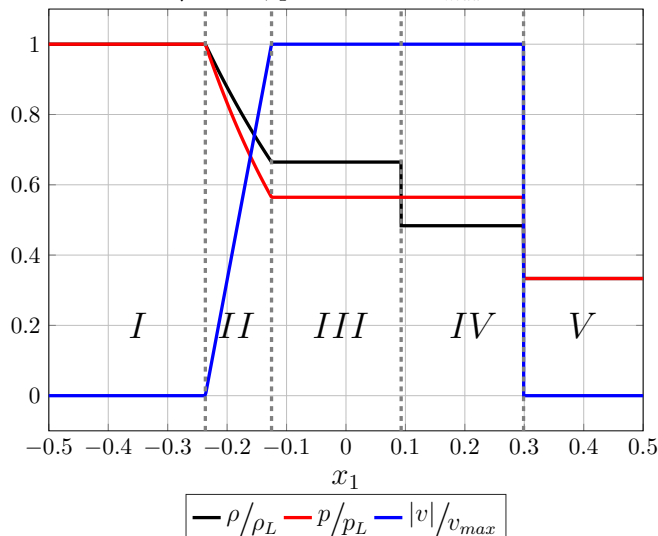
$$\rho(\mathbf{x}, 0) = \begin{cases} \rho_L, & x_1 < 0, \\ \rho_R, & x_1 > 0 \end{cases}, \quad p(\mathbf{x}, 0) = \begin{cases} p_L, & x_1 < 0, \\ p_R, & x_1 > 0 \end{cases}, \quad \mathbf{v}(\mathbf{x}, 0) = 0, \quad (5.7)$$

with $\rho_L = 3$ density units, $\rho_R = 1$ density-units, $p_L = 3$ pressure-units and $p_R = 1$ pressure-units.

The initial discontinuity in density and pressure produces, once the membrane is broken, a supersonic flow with a shock pattern conformed by: an expansion fan (region *II*

in Fig. 5.9), a rarefaction (between regions *III* and *IV* in Fig. 5.9) and a genuine shock wave (between regions *IV* and *V* in Fig. 5.9).

Figure 5.9.: Analytic solution for the 1D shock tube problem at $t = 0.2$. For this graphic the reference values $\rho_L = 3$, $p_L = 3$ and $v_{max} = 0.46411$ were taken.



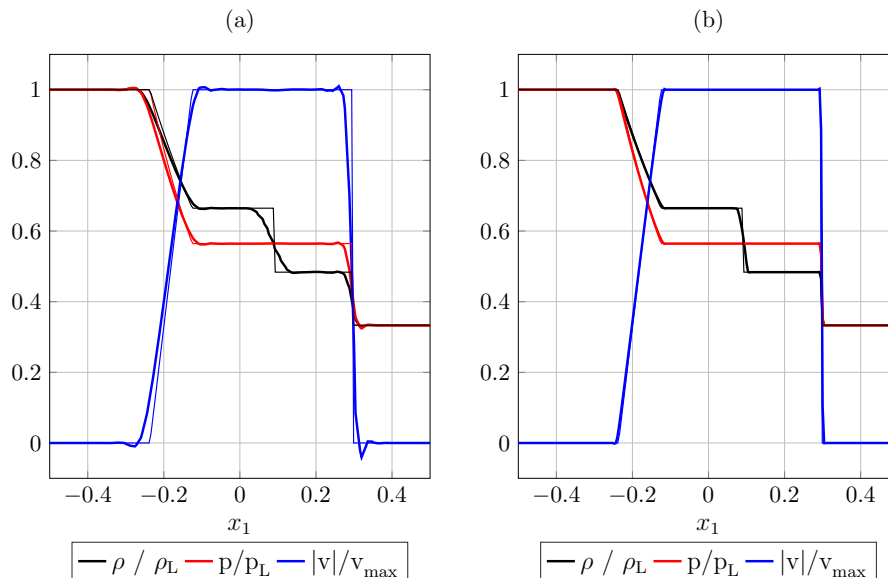
The rectangular domain in Fig. 5.8 was discretized using four different quadrilateral structured meshes: a base grid conformed by 60×6 elements ($h = \sqrt{2}/60$), and three grids formed by successive global refinements of the base mesh, i.e., meshes with $h = \sqrt{2}/120$, $h = \sqrt{2}/240$ and $h = \sqrt{2}/480$, respectively. The numerical formulation for spatial semidiscretization employed both \mathbb{P}^1 and \mathbb{P}^2 -*DG* elements, and incorporated a *FCT* strategy (see Sec. 3.3.4) to estimate the inviscid flux in the borders of the elements. Here, the upwind Vijayasundaram flux in Eq. 3.30 was used as low order term and the *AUSM*-flux in Eq. 3.56 as high order term. Additionally, in order to avoid the Gibbs phenomenon near to the discontinuities, the stabilization technique based on the introduction of artificial viscosity presented in Sec. 3.4.2 was employed. For this case, the computation of the discontinuity indicator was done using the jump in the value of density, according to Eq. 3.72. Thus, for the cells identified by the shock indicator, the stabilizations terms Eq. 3.75 and Eq. 3.76 were added to the numeric formulation (with $\nu_1 = \nu_2 = 0.5$) and the Θ parameter in Eq. 3.70 was set to 1. For all other cells no stabilization terms are added and $\Theta = 0$ is taking by default.

For the time marching scheme, the Rosenbrock–Euler method, presented in Sec. 4.3.2.2, was implemented. In order to evaluate the experimental order of convergence in time, 11 time step sizes were used: $\{T/2000, T/1000, T/500, T/250, T/100, T/70, T/40, T/20, T/10, T/5, T\}$, where $T = 0.2$ time-units.

In Fig. 5.10 two different approximations, found using a coarse grid with 60×6 elements and a finer mesh with 480×48 elements, are plotted. These profiles are presented at $t = T$ and using time steps $\Delta t = T/2000$. In the solution with the coarsest grid small oscillations are detected near to the shock wave ($x_1 \approx 0.3$), nevertheless this spurious behavior are not present in the finest solution. This effect is associated with the ability of the shock capturing strategy to detect steep gradients, while on the coarse mesh the discontinuity is approximated as a smooth step function and then the discontinuity indicator is not able to detect the elements adjacent to the jump. On the finest grid the shock is captured with a higher resolution, allowing to detect precisely where the artificial viscosity should be introduced. On the other hand, from Fig. 5.10 it is possible to note that how the error

sources are located in the shock wave, as well as at the begin and end of the rarefaction, but mainly on the contact discontinuity. This behavior is in accordance with the results reported in previous works [BHS09, GHS02].

Figure 5.10.: Numerical profiles for density, pressure and velocity along the center line $x_2 = 0$ at $t = T$ for the shock tube problem. Approximation built using (a) a grid with 60×6 \mathbb{P}^1 -elements, and (b) a finer grid with 480×48 \mathbb{P}^1 -elements. In both cases a time step $\Delta t = T/2000$ was used. Thick lines represent the numerical approximations and thin lines the analytic solutions.



A convergence analysis over space and time was developed for this problem. The analysis involved four meshes with element sizes of $h\sqrt{2}/60$, $h\sqrt{2}/120$, $h\sqrt{2}/240$ and $h\sqrt{2}/480$, two types of DG -elements (\mathbb{P}^1 and \mathbb{P}^2), and eleven time step sizes $\{T/2000, T/1000, T/500, T/250, T/100, T/70, T/40, T/20, T/10, T/5, T\}$. Fig. 5.11 and Table 5.3 show the results for the convergence analysis in space. Here, the error was computed for any of the conservative variables (ρ , ρv_1 , ρv_2 , E) and then the resulting error was calculated.

The experimental order of convergence found conforms to the theoretical error estimation proposed by [JJS95], which predicts a suboptimal convergence rate $O(h^{1/2})$, solving hyperbolic conservation equations with discontinuous Galerkin in case of non-regular solutions. Likewise, multiple numerical experiments solving Euler equations, such as those presented in [GHS02, HLR99], report convergence orders slightly above 0.5 for solutions with discontinuities.

Fig. 5.12 shows the behavior of the error in time. This plot shows that for every used mesh the error stays constant for time steps $\Delta t < T/100$, i.e. in these cases even using the finest mesh and a higher polynomial basis \mathbb{P}^2 , the global error is dominated by the spatial component. We also observe high accuracy using the exponential Rosenbrock–Euler method. Further, for larger time steps ($\Delta t > T/100$), the approximations computed using the finest grid ($h = \sqrt{2}/480$) exhibits oscillations and eventually loss of stability. This conditional-stability behavior can be observed in all the meshes analyzed. The set of approximations that have spurious oscillations are shown in Fig. 5.12 with solid markers.

In Fig. 5.13 is plotted the relation between the CFL number and the size of the Krylov

h	\mathbb{P}^1		\mathbb{P}^2	
	$\ \text{error}\ $	EOC	$\ \text{error}\ $	EOC
2.36_{-2}	3.47_{-2}		2.90_{-2}	
1.18_{-2}	2.53_{-2}	0.46	2.09_{-2}	0.47
5.89_{-3}	1.81_{-2}	0.49	1.47_{-2}	0.51
2.95_{-3}	1.31_{-2}	0.46	1.06_{-2}	0.47

Table 5.3.: Experimental order of convergence (EOC) for Sod’s shock tube problem using DG-elements and Rosenbrock–Euler method. Consider $\|\text{error}\| = \|w_h - w\|_{L^2(L^2)}$.

Figure 5.11.: Experimental order of convergence in h for Sod’s shock tube problem using \mathbb{P}^1 (black lines) and \mathbb{P}^2 (blue lines) DG-elements.

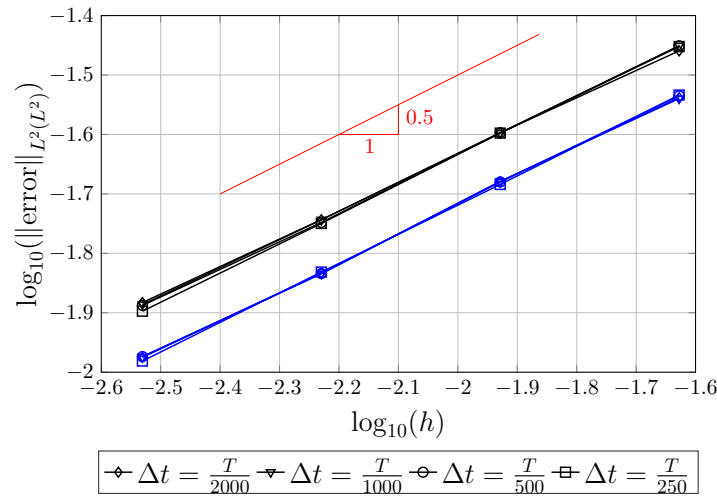
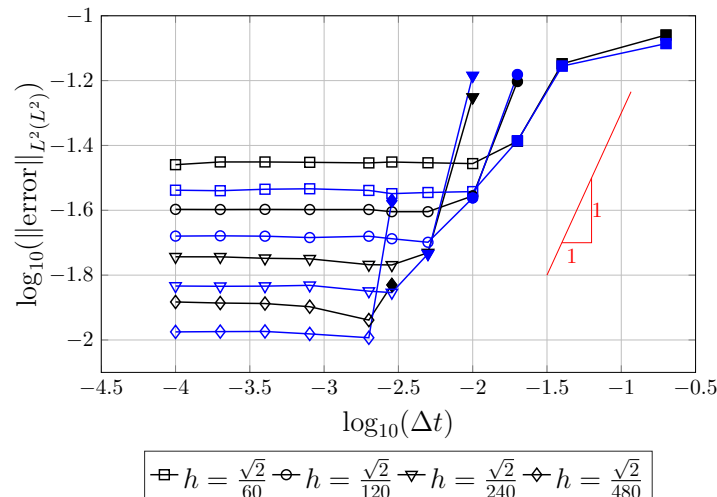


Figure 5.12.: Experimental order of convergence in time for Sod’s shock tube problem using \mathbb{P}^1 (black lines) and \mathbb{P}^2 (blue lines) DG-elements. Solid markers correspond to solutions with spurious oscillations.

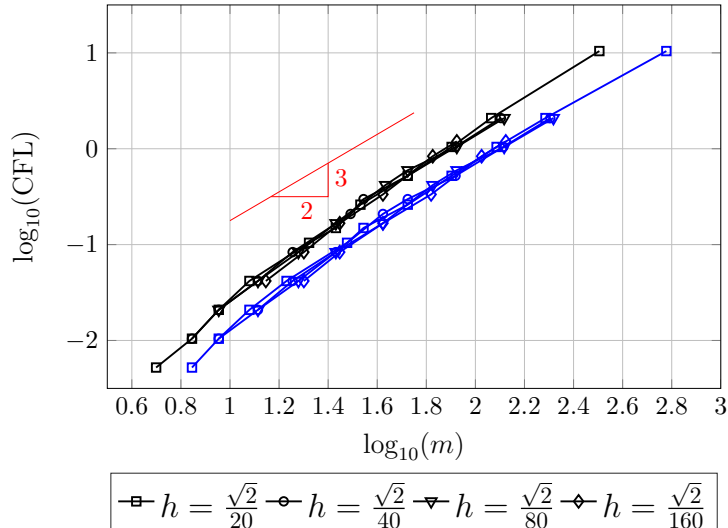


subspace computed for the shock tube problem. For this problem, and onwards for all compressible flow problem, the CFL number was computed as

$$\text{CFL} = \frac{(|\mathbf{v}| + c) \Delta t}{h}. \quad (5.8)$$

As expected, for a given mesh, increasing the time step (or equivalently the CFL number) requires a higher number of vectors m . These numerical experiments indicate a proportionality relation $m \propto \text{CFL}^{2/3}$. On the other hand, in Fig. 5.14 the time consumption is evaluated in terms of the size of Krylov subspace employed. As in previous cases, a high cost is observed for small time steps, like with explicit time marching methods, as well as a reduction of the computational cost when larger time steps are employed. Here, in similar fashion to Fig. 5.7, there will be a critical value for m (and then for CFL and Δt), after which the excessive cost for computing the exponential of a large Hessenberg matrix dominates the total cost.

Figure 5.13.: CFL number vs. size of the Krylov subspace used for the Sod's shock tube problem. Black lines correspond to solutions with \mathbb{P}^1 elements and blue lines represent approximations built with \mathbb{P}^2 elements.



5.3.3. GAMM channel problem

The flow along a channel with a circular arc bump on the lower wall is commonly named *GAMM*-channel problem (see Fig. 5.15). This case was originally proposed as test problem in the context of a workshop of the Society for Applied Mathematics and Mechanics (*GAMM* – german abbreviation of *Gesellschaft für Angewandte Mathematik und Mechanik*) [RV81], and it is commonly used to evaluate the performance of numerical schemes [FFS03, FP02, Ni82]. Here, as in [FP02], three different cases are presented: subsonic, transonic and supersonic problems are analyzed respectively. This problem was modelled using inviscid Euler equations along with the ideal gas EOS.

5.3.3.1. Subsonic case

For this case the domain represented in Fig. 5.15 was used and the thickness-to-cord ratio of the bump was set to 10%. The boundary $\partial\Omega$ is defined as

$$\partial\Omega = \Gamma_{IN} \cup \Gamma_O \cup \Gamma_W^u \cup \Gamma_W^l,$$

where

Figure 5.14.: Computational cost demanded vs. size of Krylov subspace used for the Sod's shock tube problem. Black lines correspond to solutions with \mathbb{P}^1 elements and blue lines represent approximations built with \mathbb{P}^2 elements. The CPU time reported correspond to computations with 8 cores in an *Intel Core i7-3770@3.4GHz* processor.

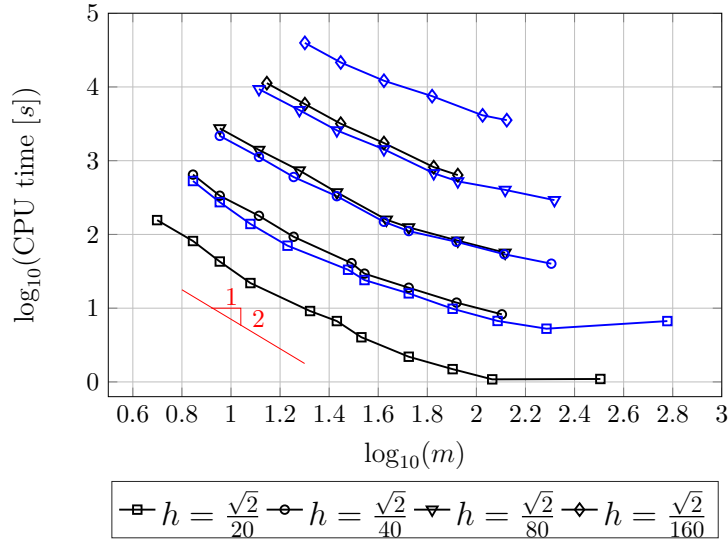
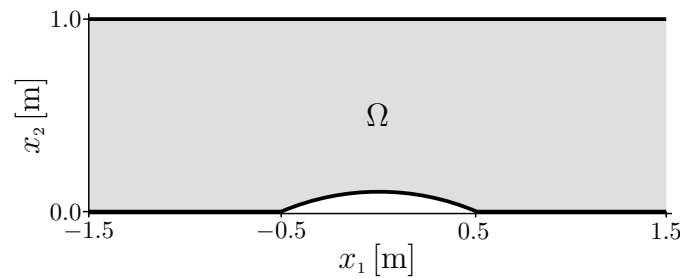


Figure 5.15.: *GAMM*-channel domain.

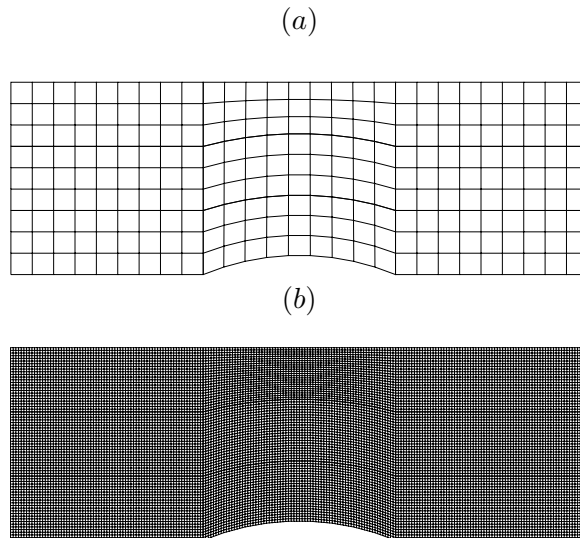


- $\Gamma_{IN} = \{\mathbf{x} \in \mathbb{R}^2 : x_1 = -1.5, x_2 \in (0, 1.0)\}$ represents the inlet boundary,
- $\Gamma_O = \{\mathbf{x} \in \mathbb{R}^2 : x_1 = 1.5, x_2 \in (0, 1.0)\}$ is outlet boundary,
- $\Gamma_W^u = \{\mathbf{x} \in \mathbb{R}^2 : x_1 \in [-1.5, 1.5], x_2 = 1.0\}$ is upper wall, and
- $\Gamma_W^l = \{\mathbf{x} \in \mathbb{R}^2 : x_1 \in [-1.5, -0.5], x_2 = 1.0\}$
 $\cup \{\mathbf{x} \in \mathbb{R}^2 : x_1 \in [-0.5, 0.5], x_2 = \sqrt{1.69 - x_1^2} - 1.2\}$
 $\cup \{\mathbf{x} \in \mathbb{R}^2 : x_1 \in (0.5, 1.5], x_2 = 1.0\}$
 defines the lower wall.

Here, a subsonic inflow condition with $M = 0.5$ was prescribed on Γ_{IN} , slip wall boundary condition was adopted on Γ_W^u and Γ_W^l , and an atmospheric pressure value, $p = 0.11\text{MPa}$, was set on the outlet boundary Γ_O .

For the spatial semidiscretization of this problem, four different structured grids were generated as follows: a base grid consisting of 27×9 quadrilateral elements was built, then it was refined three times by element splitting, producing meshes with 54×18 elements, 108×36 elements and 216×72 elements, respectively. In Fig. 5.16 the coarsest base-grid and the finest grid employed for this problem are shown. The *AUSM⁺-up* numerical flux

Figure 5.16.: Initial grids used for solving the subsonic *GAMM*-channel problem. (a) coarsest grid with 27×9 elements and (b) finest grid with 216×72 elements.



(see Sec. 3.3.3) was employed for all the tests carried out.

According to [BR97], in the context of Euler equations solved via *DG* elements, in order to get good quality in the numerical solution a sufficiently accurate approximation of the geometry is required. Thus, e.g. if a linear mapping \mathbb{Q}^1 is used to represent a curved boundary, every vertex of the polygonal approximation of this boundary gives rise to a spurious entropy production [FFS03], which normally is convected downstream, polluting other regions in the domain. To avoid this effect, a finer mesh can be introduced for the cells next to curved walls. Here, in all simulations of this test case, a second order mapping \mathbb{Q}^2 was implemented in order to ensure a precise geometric approximation of the bump.

The temporal domain was set to $(0, 0.5\text{s})$ and the Rosenbrock–Euler scheme was used as time marching scheme. In all cases a time step size $\Delta t = 0.1\text{ms}$ was used. The initial condition is given by

$$\mathbf{w}(\mathbf{x}, 0) = \left(\rho^0 \quad \rho^0 v_1^0 \quad \rho^0 v_2^0 \quad E^0 \right)^T,$$

where $\rho^0 = 1.2785\text{kg/m}^3$, $v_1^0 = 173.5312\text{m s}^{-1}$ and $v_2^0 = 0$. Here, the specific energy E^0 is computed from Eq. 2.22

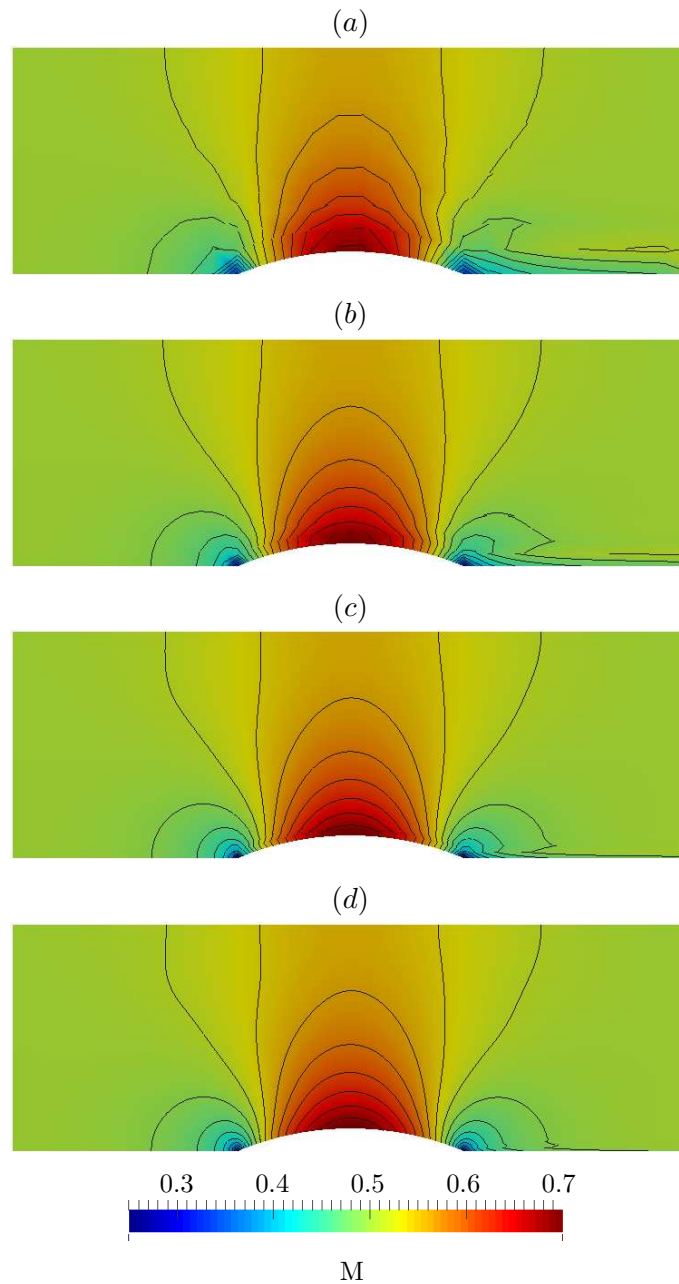
$$E^0 = \frac{p^0}{\gamma - 1} + \rho^0 \frac{|v_1^0|^2}{2},$$

with $p^0 = 0.11\text{MPa}$ and $\gamma = 1.4$.

In Fig. 5.17 the Mach number contours are plotted at steady state for the different grids used. In Fig. 5.17(a) can be noted that a layer is formed on the lower wall, downstream from the position $x_1 = -0.5$. This boundary layer is also formed in the test of Fig. 5.17(b)–(d), although the thickness of this decreases when finer grids are used. In Fig. 5.28 the Mach number profiles along the lower and upper walls are plotted. These graphics show that the Mach values computed along the upper wall agree with the profile reported in [FP02], even for the coarsest mesh. Nevertheless, on the lower wall the profile exhibit an error in the bump region, and specially along the straight wall after the bump. The plots in Fig. 5.28 also show that the error decreases slowly as finer meshes are employed,

although the difference remains even in the test with the finest mesh. This spurious layer is also present in solutions with high order polynomial basis, as is shown in Fig. 5.19.

Figure 5.17.: Mach number contours for the subsonic *GAMM*-channel problem at steady state. Results found employing \mathbb{P}^1 *DG*-elements along with grids conformed by (a) 27×9 elements, (b) 54×18 elements, (c) 108×36 elements and (d) 216×72 elements.



Using as entropy production indicator the term $\kappa = \frac{p}{\rho^\gamma}$, in Fig. 5.20 is displayed the entropy production for every grid used. Considering that the solution reported in [FP02] does not exhibit shock waves, and thus cannot appear any entropy production in the domain, the entropy layer at lower wall shown in Fig. 5.20 can be considered a numerical artifact. This spurious layer is formed when the entropy production on the bottom wall, produced at

Figure 5.18.: Mach number profiles along lower and upper walls in subsonic *GAMM*-channel using different grids. All solutions were computed using \mathbb{P}^1 *DG*-elements.

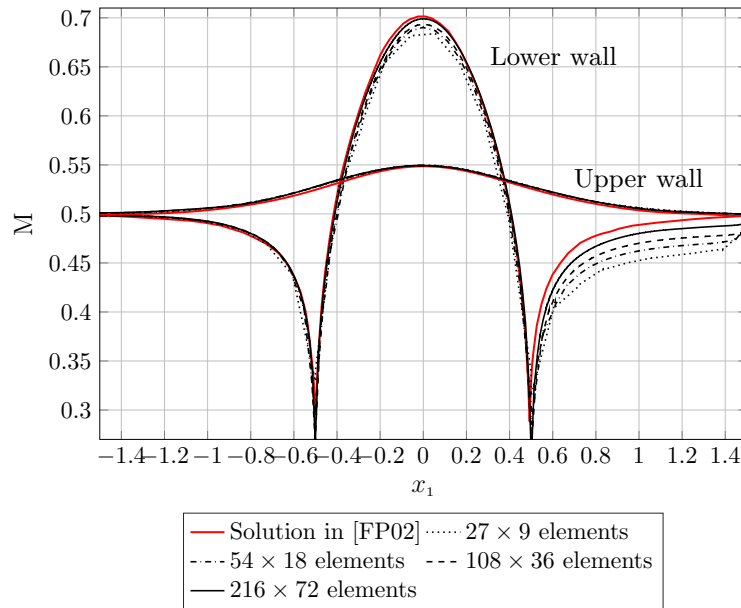
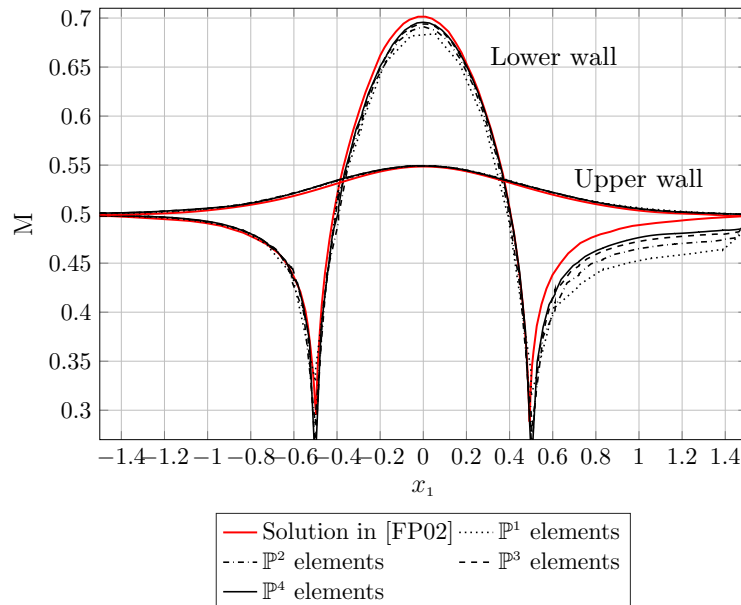


Figure 5.19.: Mach number profiles along lower and upper walls in subsonic *GAMM*-channel using different polynomial *DG*-basis. All solutions were computed using the base grid (27×9 elements).



$x_1 = -0.5m$ and $x_1 = -0.5m$ where the slope of the wall changes suddenly, is advected downstream, polluting the approximation in the elements next to the wall. In order to reduce the entropy production in the extreme points of the bump, a locally refined grid, as in Fig. 5.21, can be implemented.

Figure 5.20.: κ -indicator for entropy production for the subsonic *GAMM*-channel problem. The contours correspond to the steady state. Approximations computed using \mathbb{P}^1 *DG*-elements along with grids conformed by (a) 27×9 elements, (b) 54×18 elements, (c) 108×36 elements and (d) 216×72 elements.

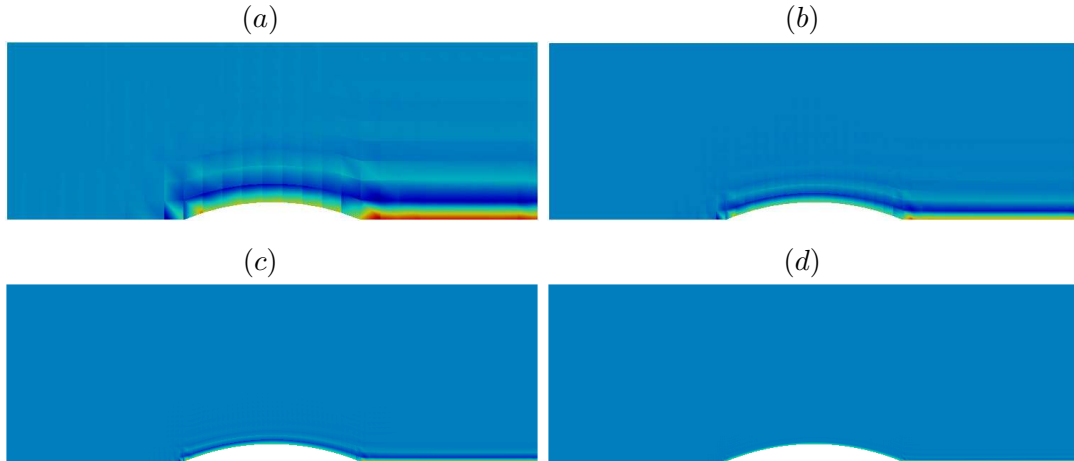
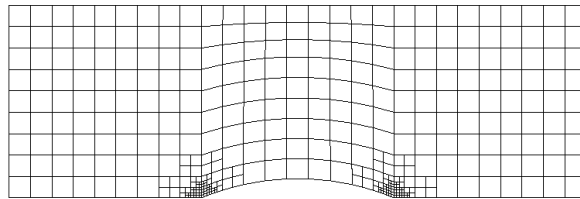


Figure 5.21.: Base grid locally refined in the extreme points of the bump. Total of 387 elements in the grid.



Increasing the mesh resolution at the extreme points of the bump reduces the entropy production, as can be noted in Fig. 5.22, suppresses the boundary layer at the downstream bottom wall and improves dramatically the approximation. This effect on the quality of the approximation can be appreciated in Fig. 5.23, where a comparison between the solution with the base mesh and the approximation using the locally refined mesh in Fig. 5.21 is shown. Finally, in Fig. 5.24 can be noted as the reduction in the spurious entropy production allows to fit better the Mach profile along both walls, even using a coarse mesh.

For the time step used ($\Delta t = 0.1\text{ms}$) and considering the four uniform grids employed, maximum values for the CFL number equal to 0.5, 1.0, 2.0 and 4.0 were found. In Fig. 5.25 a sublinear relation between the CFL number and the number of vectors employed in the Krylov subspace is shown, as well as a quadratic relation between the CFL number and the total *CPU*-time demanded in every case. Here, the CFL number was computed according to Eq. 5.8.

5.3.3.2. Transonic case

This case is similar to subsonic *GAMM*-channel problem but the velocity in the inlet boundary is increased up to $M = 0.675$. Under this inflow condition the solution (see [FP02]) exhibit a shock wave on the bump, approximately at 72% of chord [Ni82], and immediately downstream a minimum value for the Mach number, named *Zierep* singular-

Figure 5.22.: Entropy production for the subsonic *GAMM*-channel problem at steady state using an (a) uniform grid with $27 \times 9 \mathbb{P}^1$ elements, and (b) a locally refined grid on the extreme points of the bump.

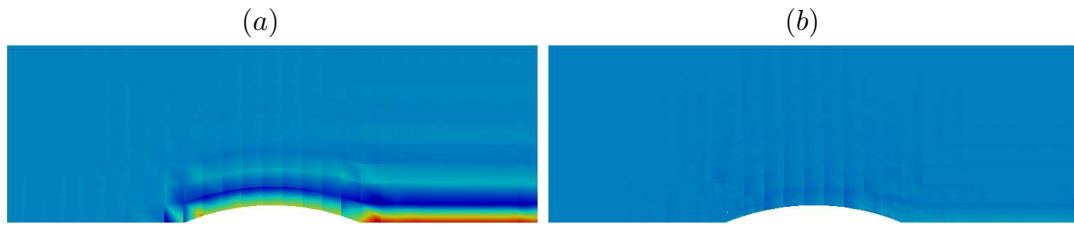
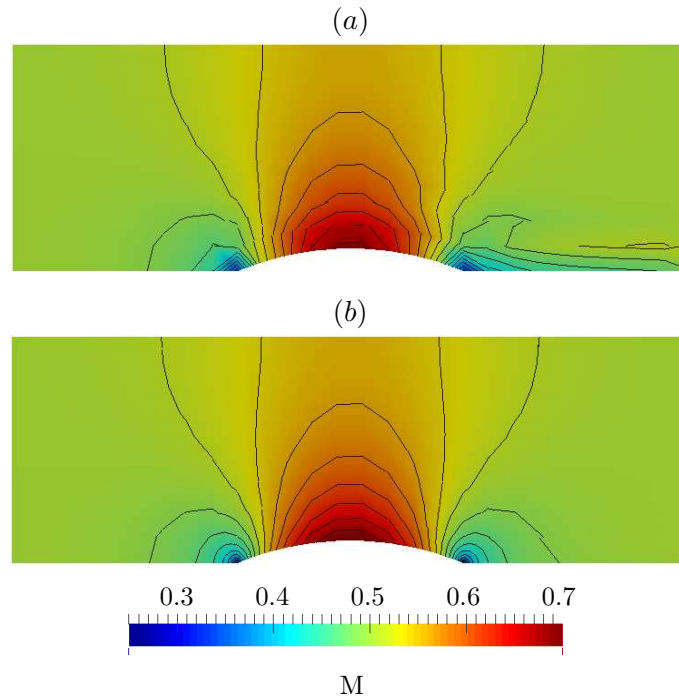


Figure 5.23.: Mach number contours for the subsonic *GAMM*-channel problem at steady state using an (a) uniform grid with $27 \times 9 \mathbb{P}^1$ elements, and (b) a locally refined grid on the extreme points of the bump.



ity, is localized. The challenge with this test case is to capture precisely the shock wave, avoiding the *Gibbs* phenomenon in the neighborhood of the discontinuity and allowing a good approximation of the *Zierep* singularity. This problem is extensively described and solved in [FFS03].

Initially, to solve this test case two quadrangular uniform grids with 27×9 and $216 \times 72 \mathbb{P}^1$ *DG*-elements were used. The *AUSM⁺-up* numerical flux was incorporated for all the transonic tests developed. Further, in order to avoid a spurious entropy production at the bump corners, the coarsest grid was locally refined at these points as in Fig. 5.21. For the time marching strategy the Rosenbrock–Euler method with $\Delta t = 0.1\text{ms}$ was implemented.

Taking in account the presence of discontinuities in the solution, a shock capturing strategy

Figure 5.24.: Mach number profiles along lower and upper walls in subsonic *GAMM*-channel. Solutions computed using a uniform grid with 27×9 \mathbb{P}^1 elements and a locally refined grid.

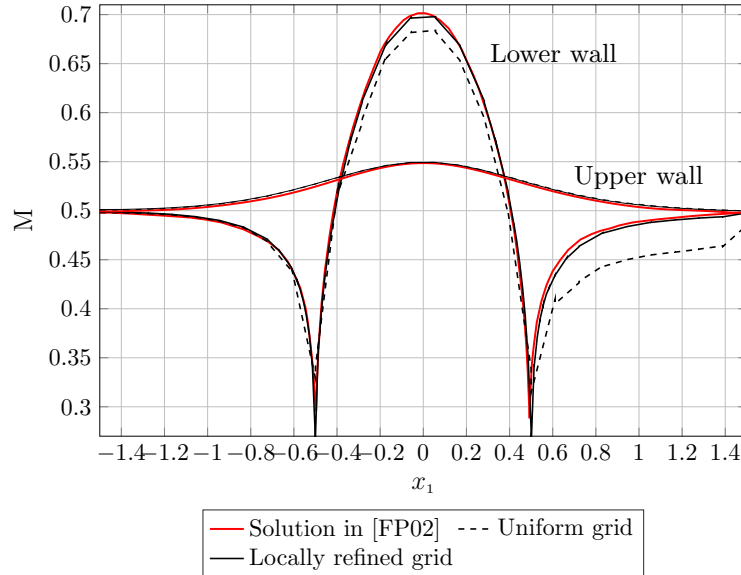
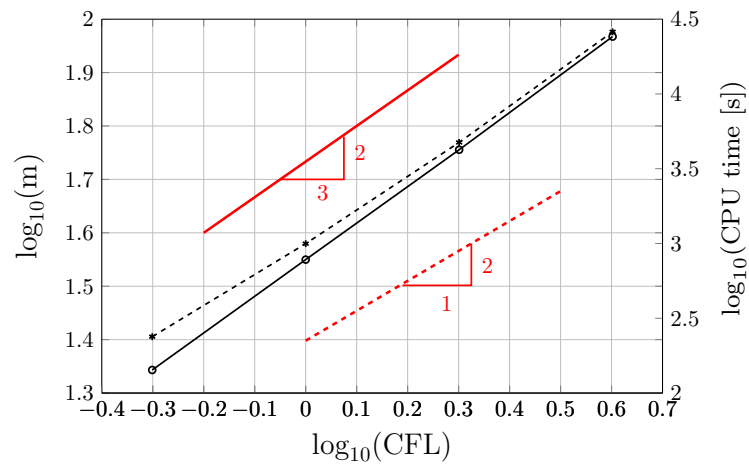


Figure 5.25.: Size of the Krylov subspace m (solid line) and computational cost (dashed line) vs. CFL number used to solve the subsonic *GAMM*-channel problem. The CPU times reported correspond to computations using 8 cores in an *Intel Core i7-3770@3.4GHz* processor.



was included for this case. The discontinuity indicator procedure described in Sec. 3.4.1, using the density jumps as indicator variable, was used to detect the elements in the neighborhood of the discontinuity. Besides, the artificial viscosity technique presented in Sec. 3.4.2, using $\nu_1 = \nu_2 = 1.0$, was introduced in order to avoid spurious oscillations on these elements.

Fig. 5.26(a) shows the Mach number contours found at steady state using the coarsest grid (≈ 3900 unknowns). Further, in Fig. 5.27 the corresponding Mach number profiles along upper and lower walls are plotted. As in the previous test case this coarse grid is enough to find a good approximation in the regions where the solution is smooth, however, it shows a poor resolution to capture properly both shock waves and the *Zierep* singularity. Looking for increasing the accuracy in the approximation, the coarsest mesh with cubic polynomial *DG*-basis (≈ 15500 unknowns) were used in the solution shown in Fig. 5.26(b) and Fig. 5.27(solid line). Here, this higher order approximation allows to capture the *Zierep* singularity, although the improvement in the resolution of the shock wave is not remarkable. Further, as is shown in Fig. 5.26(c) and Fig. 5.28, using a finer grid, a high resolution and very good location of the shock wave is found. Nevertheless, a high computational cost, derived from the 250 000 unknowns employed, has to be payed.

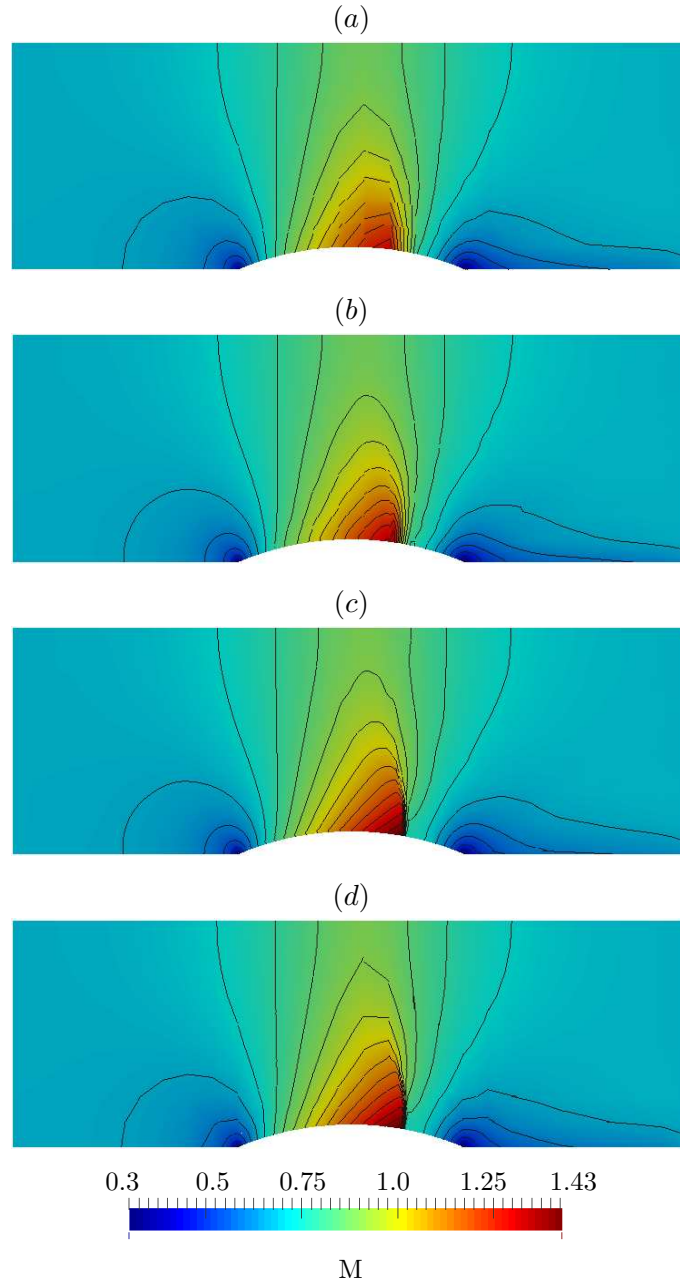
A way to ensure a good resolution in both discontinuity and *Zierep* singularity is employing an adaptive meshing strategy. Here, the adaptive mesh refinement technique proposed in Sec. 3.5 was implemented. Then, using the gradient of the Mach number as the refinement indicator, a dynamic adaptive grid, able to capture and to follow the shock waves through the domain, was obtained. Particularly, based on the coarsest mesh and limiting the maximum refinement level of each element to 4, the grid shown in Fig. 5.29 was found. Thus, the higher element density in the critical non-smooth regions increases the resolution without a notable raising of the total unknowns (579 \mathbb{P}^1 elements and ≈ 9300 unknowns). This effect can be appreciated in Mach number contours of Fig. 5.26(d), but specially in the Mach profile along the lower wall plotted in Fig. 5.28 (solid line) where the steepness of the shock wave is even better than the one presented in [FP02].

In Fig. 5.30 the κ -indicator for entropy production is plotted for the different discretizations used. In these graphics, specially in Fig. 5.30(c) and (d), the parallel isentropic lines show that the entropy produced along the shock wave is convected downstream, and no spurious entropy sources are detected throughout the domain. Additionally, as expected, the magnitude of the entropy produced in the discontinuity reflects how precise the shock wave was approximated.

On the other hand, in the right column in Fig. 5.30 the discontinuity indicator used to stabilize the solution in the neighborhood of the shock wave is plotted. There, it is noted that a better grid resolution in the region of the discontinuity allows to define precisely where to introduce the stabilizations terms. The precise shock capturing registered, using both finest mesh and the adaptive grid, as well as the absence of oscillations near to the Mach jump in the lower-wall profiles of Fig. 5.28, show that the stabilization terms, used into marked elements, introduced enough artificial viscosity to reduce the *Gibbs* phenomenon in the solution without detriment of its resolution.

Finally, Fig. 5.31 shows the CFL numbers found with both the finest uniform grid and with the adaptive mesh. Here, the CFL number was computed according to Eq. 5.8. It is necessary to highlight that the wide stability region of the Rosenbrock–Euler method allows e.g. to implement an adaptive grid strategy, where local small elements can produce high CFL numbers, without loss of stability.

Figure 5.26.: Mach number contours for the transonic *GAMM*-channel problem at steady state. Results found employing grids with (a) $27 \times 9 \mathbb{P}^1$ *DG*-elements, (b) $27 \times 9 \mathbb{P}^3$ *DG*-elements, (c) $216 \times 72 \mathbb{P}^1$ *DG*-elements and (d) an adaptive grid with 579 \mathbb{P}^1 *DG*-elements.



5.3.3.3. Supersonic case

In this case a 4% bump is employed along with a supersonic inflow boundary condition ($M = 1.65$). The solution presented in [FP02] shows a shock wave generated just when the flow reaches the bump. Then, this shock crosses the channel until it reaches upper wall, where it is reflected downward to cross a second shock wave, generated at the end of the bump, and to finally arrive to the outflow boundary. In contrast to the previous tests, here supersonic speeds are registered on all domain and the shock waves are extended along the channel. The challenge here is to achieve stability of the solver at supersonic regime,

Figure 5.27.: Mach number profiles along lower and upper walls in transonic *GAMM*-channel using different polynomial *DG*-basis. All solutions were computed using the base grid (27×9 elements).

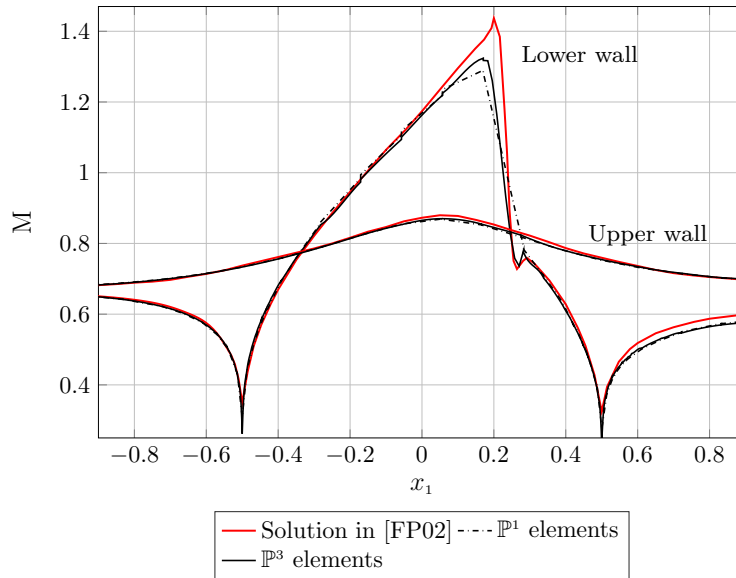
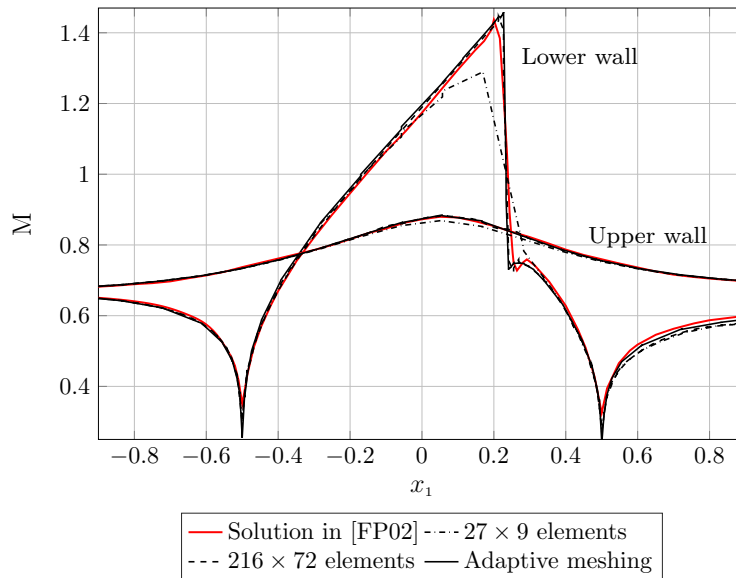


Figure 5.28.: Mach number profiles along lower and upper walls in transonic *GAMM*-channel using different grids with \mathbb{P}^1 *DG*-elements.



as well as the accuracy of the shock capturing strategy implemented.

For this test case two different grids were employed: an uniform grid with 216×72 \mathbb{P}^1 *DG*-elements and a dynamically adaptive grid with 4596 \mathbb{P}^1 *DG*-elements at steady state (see Fig. 5.32). The *AUSM⁺-up* numerical flux (see Sec. 3.3.3) was employed for all the tests with supersonic inlet condition. As in the transonic *GAMM*-channel case, the shock indicator in Eq. 3.72 was used to identify the cells where the stabilization terms should be included, although in this case the parameters ν_1 and ν_2 were set equal to 1.5. For the

Figure 5.29.: Adaptive grid for the shock capturing in the transonic *GAMM*-channel problem. Total of 579 elements in the grid.

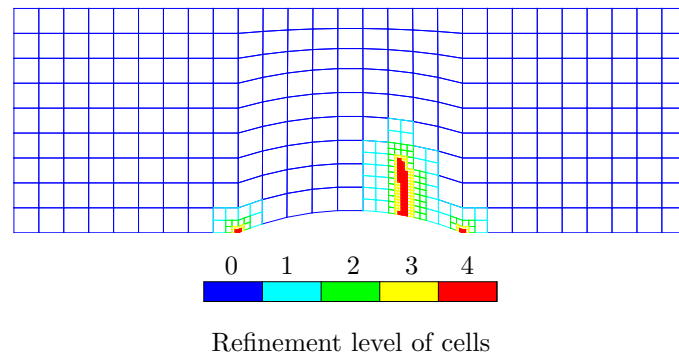
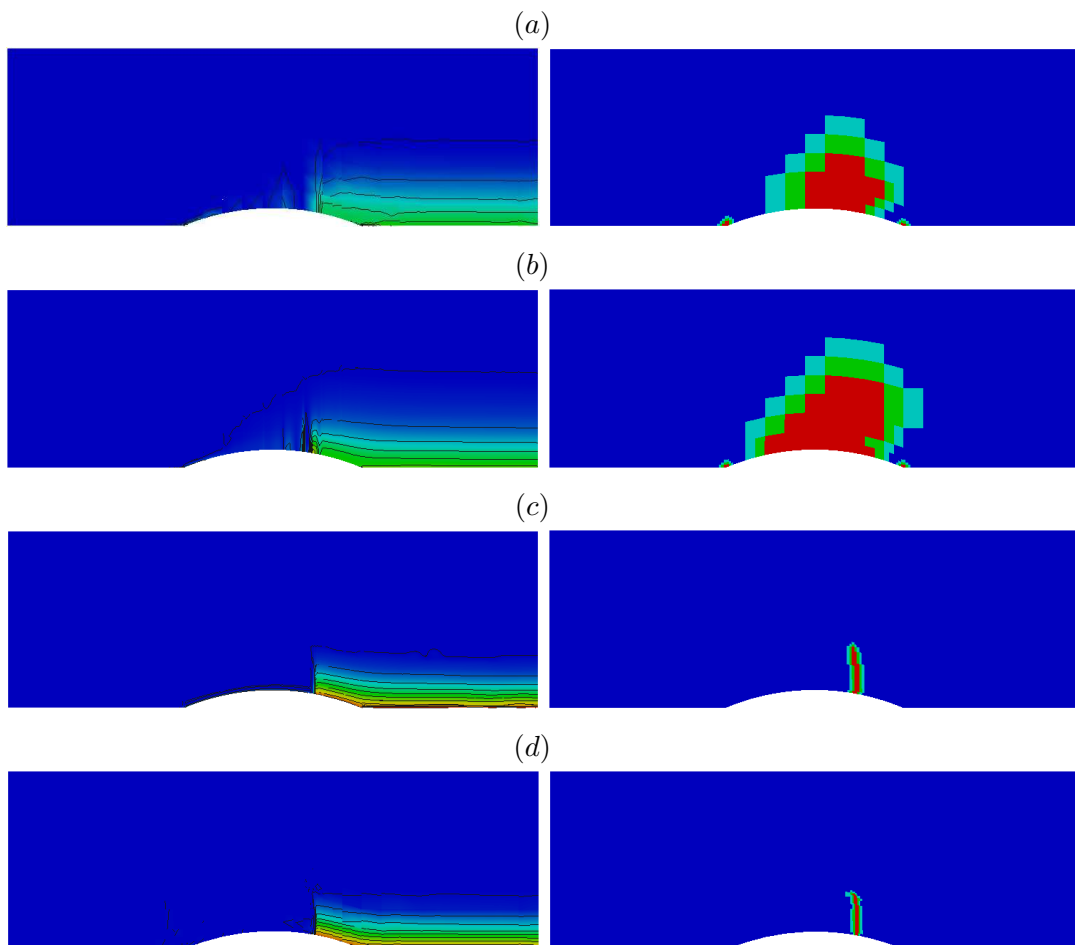
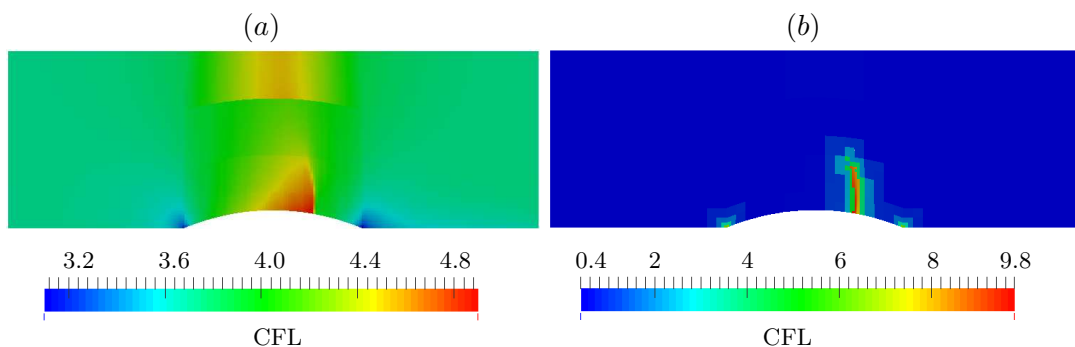


Figure 5.30.: κ -entropy production indicator (first column) and G -switch indicator (second column) for the transonic *GAMM*-channel problem. Contours correspond to the steady state. Values computed using (a) 27×9 \mathbb{P}^1 *DG*-elements, (b) 27×9 \mathbb{P}^3 *DG*-elements, (c) 216×72 \mathbb{P}^1 *DG*-elements and (d) an adaptive grid with 579 \mathbb{P}^1 *DG*-elements.



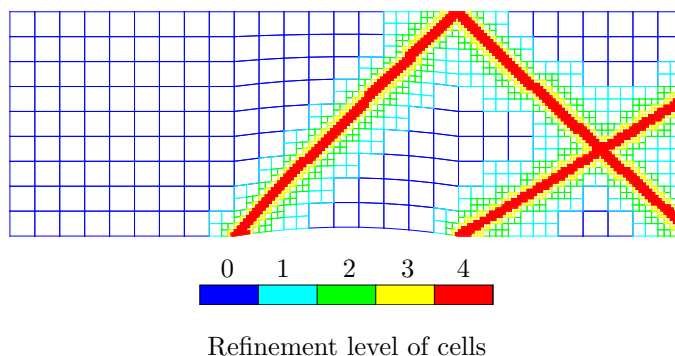
local mesh refinement the strategy in Algorithm 1, employing the gradient of the Mach

Figure 5.31.: *CFL* numbers achieved for the transonic *GAMM*-channel problem. Values correspond to the steady state using (a) an uniform grid with 216×72 elements and (b) an adaptive grid with 579 elements.



number as indicator, was incorporated. In this case the maximum refinement level was set to 4. For the time marching strategy the Rosenbrock–Euler method with $\Delta t = 0.1\text{ms}$ was implemented.

Figure 5.32.: Adaptive grid for the shock capturing in the supersonic *GAMM*-channel problem. Total of 4596 elements in the grid.



In Fig. 5.33 the predicted Mach number contours at steady state are plotted for both uniform grid (a) and adaptive mesh (b). On the other hand, in Fig. 5.34 the Mach number profiles along upper and lower walls are traced. Here, it is noted that in both grids the shock waves are localized with high precision, although the steepness of the shock captured using an adaptive meshing is slightly better. This better behavior in the second case is produced not only for the better mesh resolution along the discontinuities, but also because of the stabilization terms introduced in a narrow region, as is illustrated in Fig. 5.35. It allows to control more precisely the amount of artificial viscosity incorporated into the solution.

Finally, in Fig. 5.36 is reported the CFL number reached at steady state using each mesh. As in the previous examples, the *DG* discretization along with the exponential time integrator allow to reach high local CFL numbers, and then it is possible to use locally refined grids without loss of the stability or a strong penalization in the size of the time step.

Figure 5.33.: Mach number contours for the supersonic *GAMM*-channel problem at steady state. Results found employing grids with (a) $216 \times 72 \mathbb{P}^1$ *DG*-elements and (b) an adaptive grid with 4596 \mathbb{P}^1 *DG*-elements.

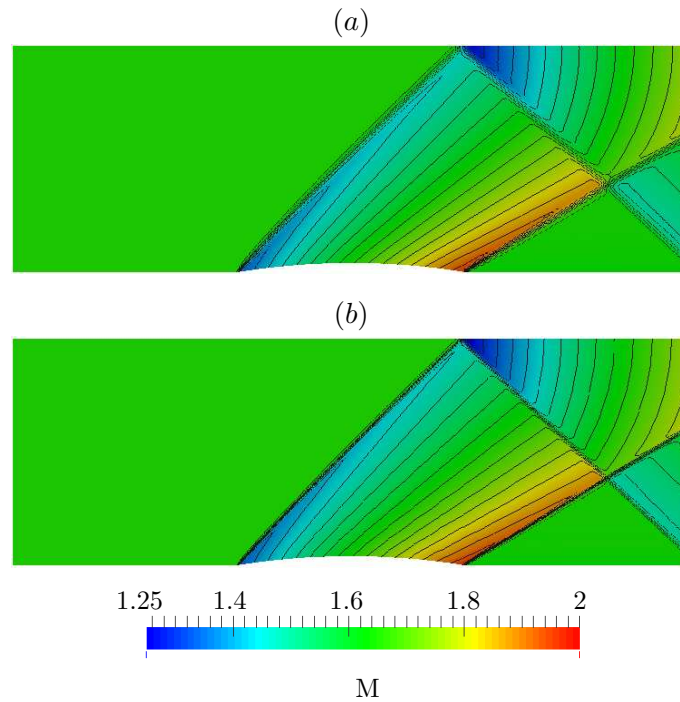
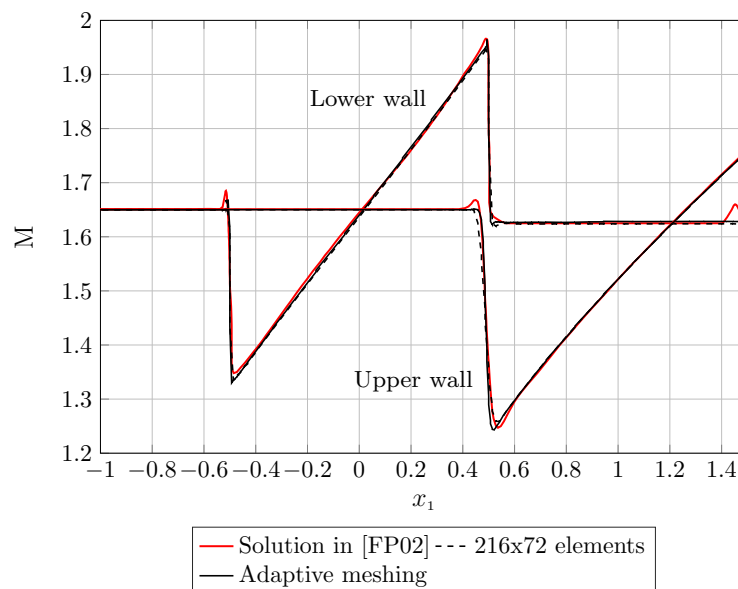


Figure 5.34.: Mach number profiles along lower and upper walls in supersonic *GAMM*-channel using different grids with \mathbb{P}^1 *DG*-elements.



5.3.4. Supersonic forward-facing step problem

This case test, originally presented in [Eme68] and widely discussed in [WC84, WC81], consists of a wind tunnel with 3 length-units long and 1 length-units of height. As is

Figure 5.35.: κ -entropy production indicator (first row) and G -switch indicator (second row) for the supersonic *GAMM*-channel problem. Contours correspond to the steady state. Values computed using (a) an uniform grid with 216×82 \mathbb{P}^1 *DG*-elements and (b) an adaptive grid with 4596 \mathbb{P}^1 *DG*-elements.

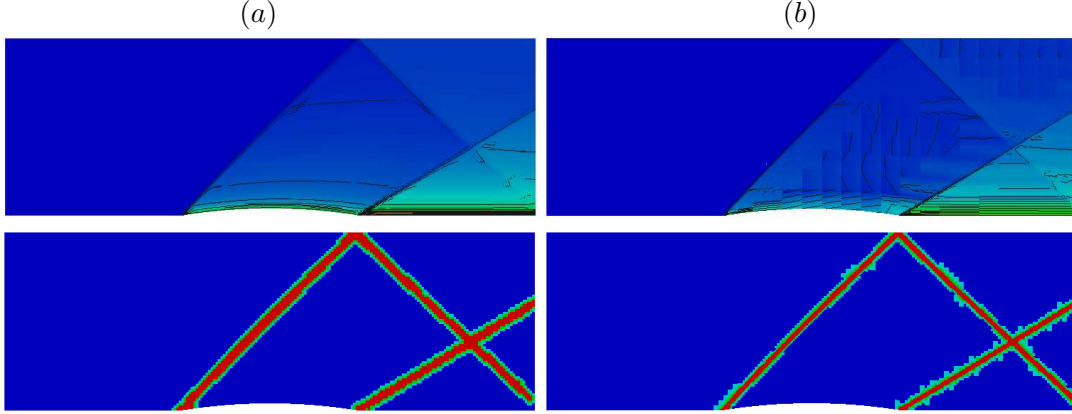
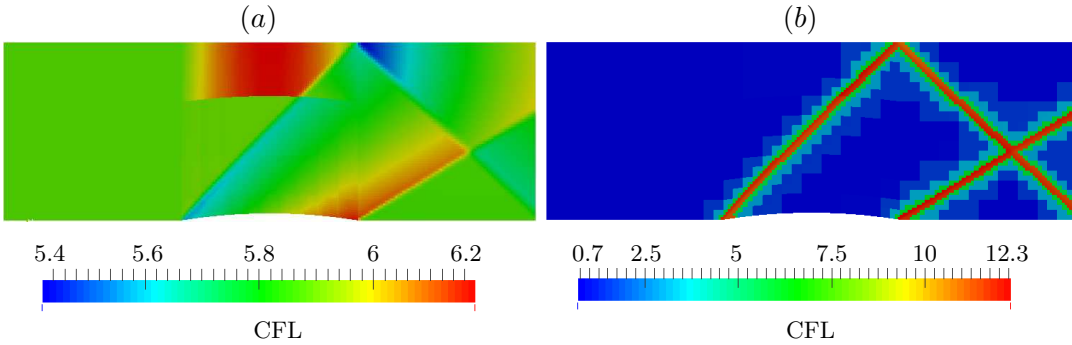


Figure 5.36.: *CFL* numbers achieved for the supersonic *GAMM*-channel problem. Values correspond to the steady state using (a) an uniform grid with 216×72 elements and (b) an adaptive grid with 4596 elements.



showed in Fig. 5.37, at 0.6 length-units from the inlet border a step with 0.2 length-units of height is localized. Thus, the boundary $\partial\Omega$ is defined as

$$\partial\Omega = \Gamma_{IN} \cup \Gamma_O \cup \Gamma_W^u \cup \Gamma_W^l,$$

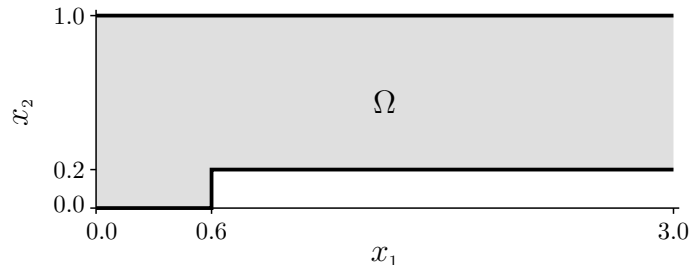
where

- $\Gamma_{IN} = \{\mathbf{x} \in \mathbb{R}^2 : x_1 = 0, x_2 \in (0, 1.0)\}$ represents the inlet boundary,
- $\Gamma_O = \{\mathbf{x} \in \mathbb{R}^2 : x_1 = 3.0, x_2 \in (0.2, 1.0)\}$ is outlet boundary,
- $\Gamma_W^u = \{\mathbf{x} \in \mathbb{R}^2 : x_1 \in [0, 3.0], x_2 = 1.0\}$ is upper wall, and
- $\Gamma_W^l = \{\mathbf{x} \in \mathbb{R}^2 : x_1 \in [0, 0.6), x_2 = 0\}$
 $\cup \{\mathbf{x} \in \mathbb{R}^2 : x_1 \in [0.6, 3.0], x_2 = 0.2\}$
 defines the lower wall.

On the inflow boundary Γ_{IN} a supersonic condition with $M = 3$ was imposed. Additionally, a reflective slip boundary condition is assumed on $\Gamma_W^u \cup \Gamma_W^l$ and a supersonic outflow

condition is taken on Γ_O . For the initial condition the values $\rho^0 = 1.4$ density-units, $\mathbf{v}^0 = (3 \ 0)^T$ velocity-units and $p^0 = 1$ pressure-units, were taken for the entire domain.

Figure 5.37.: Forward-facing step domain.



In this problem, widely used to test multiple numerical techniques for compressible flows, an initial shock emanating from the step expands through the channel and reflects on the upper and lower walls, creating a complicated effect of shock interactions. Although the steady state for this problem is found after $t = 12$ time-units, multiple works analyze the unsteady state at $t = 4$ time-units, because of the complex interactions of shocks waves, rarefactions and contact discontinuities at this time [WC81]. In Fig. 5.38 density contours reported by [ZQSD08] at $t = 4$ time-units are plotted. This figure shows, on the upper side, a λ -type shock with its Mach stem aligned with the step, and a contact discontinuity arises from its triple point. Along this contact discontinuity a Kelvin-Helmholtz instability can be appreciated. Despite this is a physical instability, according with [WC84] it is numerically triggered by very small entropy oscillations produced in the Mach shock. On the lower wall, the Fig. 5.38 shows the rarefaction fan generated in the step corner, as well as a weak contact discontinuity arising from the step. Additionally, on the upper surface of the step appears a shock reflection, along with a artificial Mach stem (small in the case of Fig. 5.38), and spurious boundary layer.

In addition to the proper capturing of the multiple shocks, one of the most challenging problems in the numerical solution of this case is the numerical treatment for the neighborhood around the corner of the step. This point is the origin of a rarefaction, thus a singular point. A poor resolution in the corner singularity produces an error source that introduce spurious entropy, which is advected downstream along the top of the step, forming an erroneous entropy layer on this surface. Then, the shock wave reflected on the lower wall will be altered by this boundary layer, creating a Mach stem (see item e in Fig. 5.38) whose length will depend of the layer thickness. In fact, a bad treatment of this problem can degenerate drastically the whole solution. In order to solve this problem some works propose to implement a special boundary condition near to the step corner [WC84], and others improve the resolution in this region either employing local refinement [CS98b] or using high order polynomial basis [HCP12].

This test case was modeled using inviscid Euler equations for ideal gas. For the spatial discretization an initial grid with 696 \mathbb{P}^1 DG -elements was employed (see Fig. 5.39), along with the adaptive grid technique described in Sec. 3.5, using the $\nabla\rho$ to compute the refinement indicator and a maximum refinement level $r_{L,\max} = 4$. In this test we employ a numerical flux based on the FCT strategy which uses $AUSM$ -flux as high order term and the Vijayasundaram-flux as low order component.

Here, in spite of other works [WC84], no special boundary condition for the regions near to the singular point was used, but looking for limiting the entropy production in the step corner, a local refinement around this point was done. Thus, taking the initial mesh, a small neighborhood around the corner, with radius $r = 0.02$, was defined and the elements inside were refined by splitting six times. The refinement level in these cells was forced into the adaptive strategy to conserve this local high resolution in the dynamic grid.

For time marching scheme a Rosenbrock–Euler method, with $\Delta t = 0.005$ time-units, was implemented. In order to avoid the Gibbs-phenomenon in the neighborhood of the shocks, the shock capturing strategy described in Sec. 3.4.1 was engaged, using a discontinuity indicator based on the jumps in density, as in Eq. 3.72. The parameters ν_1 and ν_2 , included in the stabilization terms in Eq. 3.75 and Eq. 3.76, were set to 0.2 for this test.

Figure 5.38.: Density contours for the forward-facing step problem at $t = 4$ time-units. Thirty contourlines uniformly spaced from 0.32 to 6.15. Here (a) points to the upper Mach stem, (b) shows the triple point, (c) points to the shock wave, (d) shows the contact discontinuity that arises from the step, (e) point to the lower Mach stem, (f) shows the spurious entropy boundary layer, (g) shows to the contact discontinuity that emanate from the triple point, and (h) shows the Kelvin–Helmholtz instability developed along the upper contact discontinuity. Results reported by [ZQSD08].

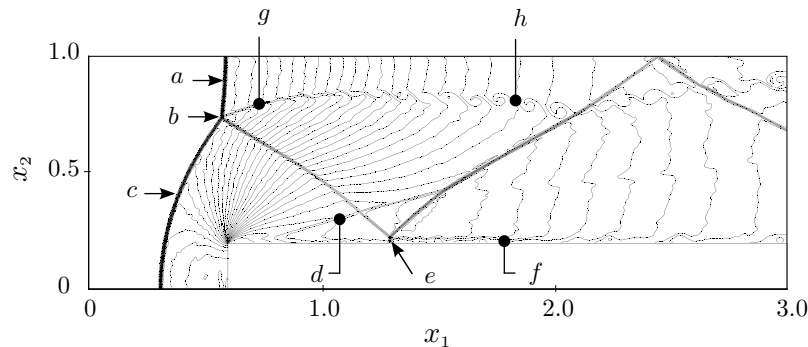
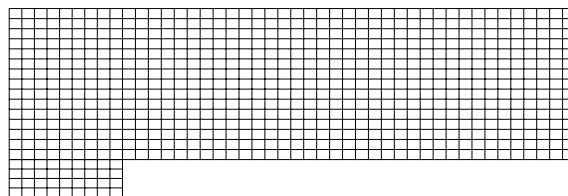


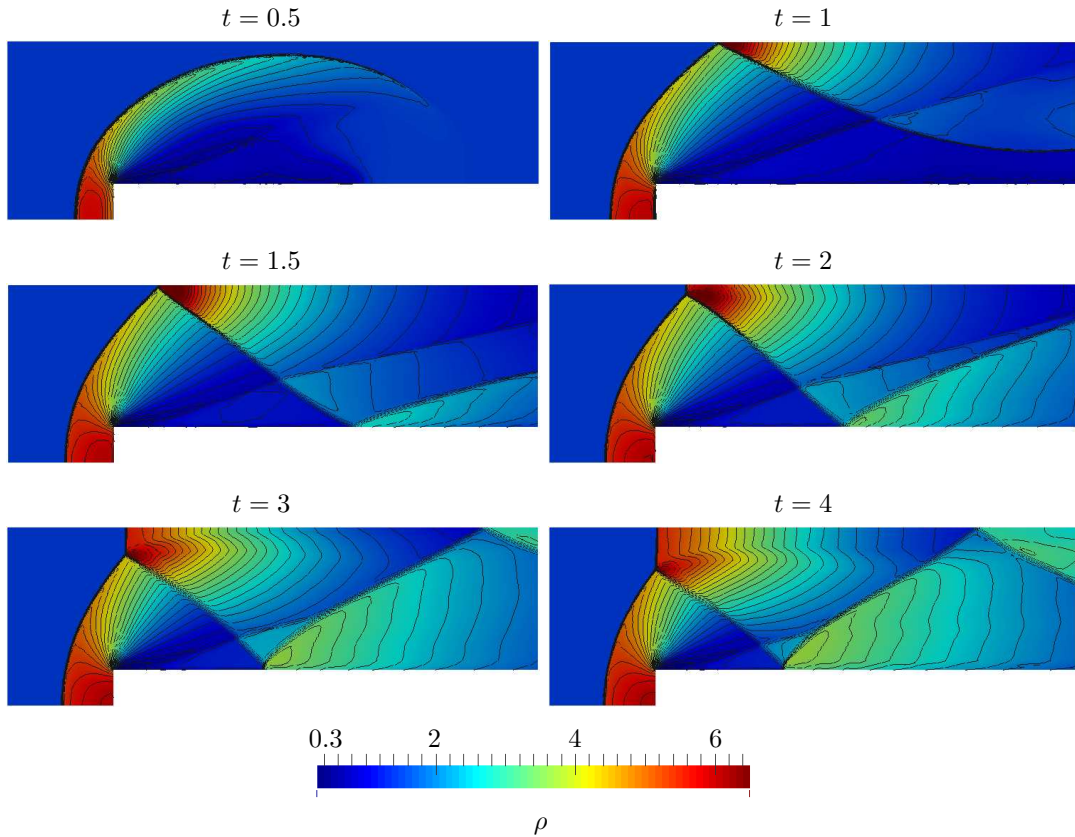
Figure 5.39.: Coarse grid used for solving the forward-facing step problem.



In Fig. 5.40 and Fig. 5.41 the predicted values for density and Mach number at different times are plotted. Here, at $t = 4$ it is possible to identify most of the relevant flow-characteristics previously described. Thus, the λ -type wave on the upper wall of the channel was accurately captured. Its location, with the Mach stem aligned with the step, as well as the length of the stem $l_\lambda \approx 0.24$, are in accordance with the results reported by multiple authors, e.g. [WC84, CS98b]. On the other hand, arising from the triple point of

this λ -shock, it is possible to identify a contact discontinuity, which extends downstream and weakly crosses the shock wave reflected on the upper wall in $x_1 \approx 2.4$. Citing Woodward and Colella [WC81], this effect is one of the most difficult to capture in this test case. Here, unfortunately, no Kelvin–Helmholtz instability could be reproduced, most probably because the lack of the resolution in the contact discontinuity captured.

Figure 5.40.: Density contours at different times for the forward-facing step problem using adaptive grid. Thirty contourlines uniformly spaced from 0.32 to 6.15.

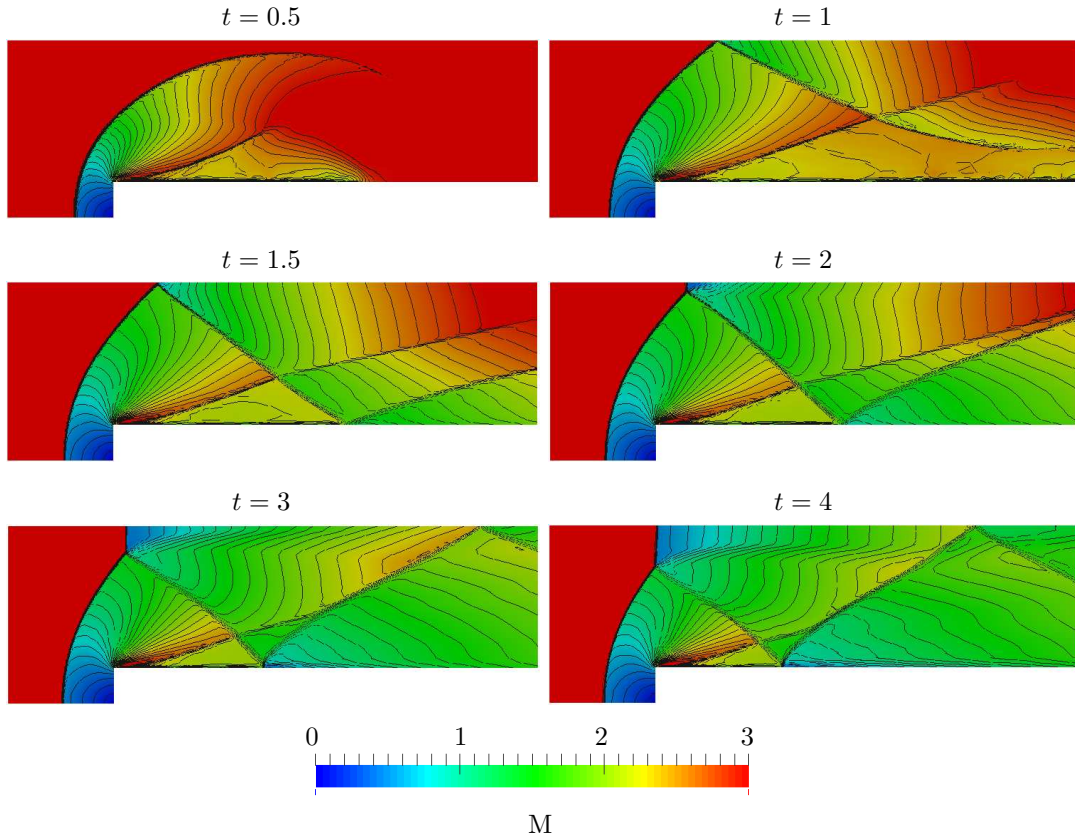


On the upper wall of the step ($x_2 = 0.2$), the proper location of the lower shock reflection (at $x_1 \approx 1.3$) and the lack of a Mach stem in it, show a reduced amount of entropy generated in the corner. This effect can be verified in the graphic of the entropy production indicator of Fig. 5.42. However, the isentropic lines display a thin entropy layer on the upper wall of the step, although it is not enough to perturb neither the shock reflection and the shock sliding on the wall.

On the other hand, the rarefaction fan with origin in the step corner was high-accurately reproduced, while the lower contact discontinuity that arises from the step was detected early and reproduced along the entire simulation.

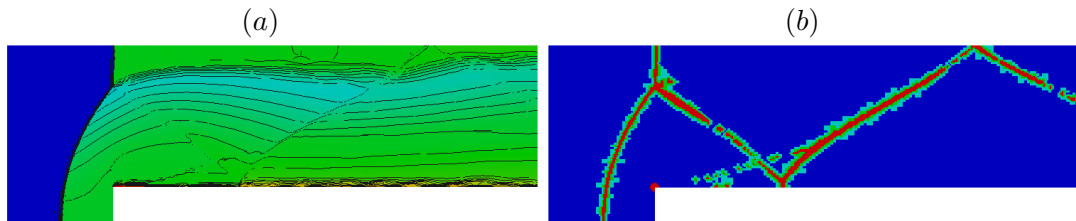
Along the simulation, the steepness of the shock waves, as well as the lack of spurious oscillations in the solution, show the good performance of the shock capturing and stabilization techniques used here. In Fig. 5.42 the values for the G -switch variable, which is employed to identify the elements where the stabilization terms need to be introduced, are plotted.

Figure 5.41.: Mach number contours at different times for the forward-facing step problem using adaptive grid. Thirty contourlines uniformly spaced from 0 to 3.



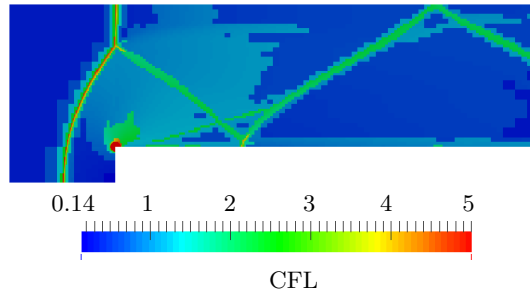
In Fig. 5.43 the values of the CFL number achieved at $t = 4$ time-units are plotted. Here, the CFL number was computed according to Eq. 5.8. It is important to highlight that the maximum CFL value found is approximately 18 and it is located in the region near to the step corner, where the mesh was refined to avoid the spurious entropy. Nevertheless, in Fig. 5.43 the maximum value in the colormap was limited to 5 in order to appreciate the lower CFL values distributed along the domain.

Figure 5.42.: (a) κ -entropy production indicator and (b) G -switch indicator for the forward-facing step problem.



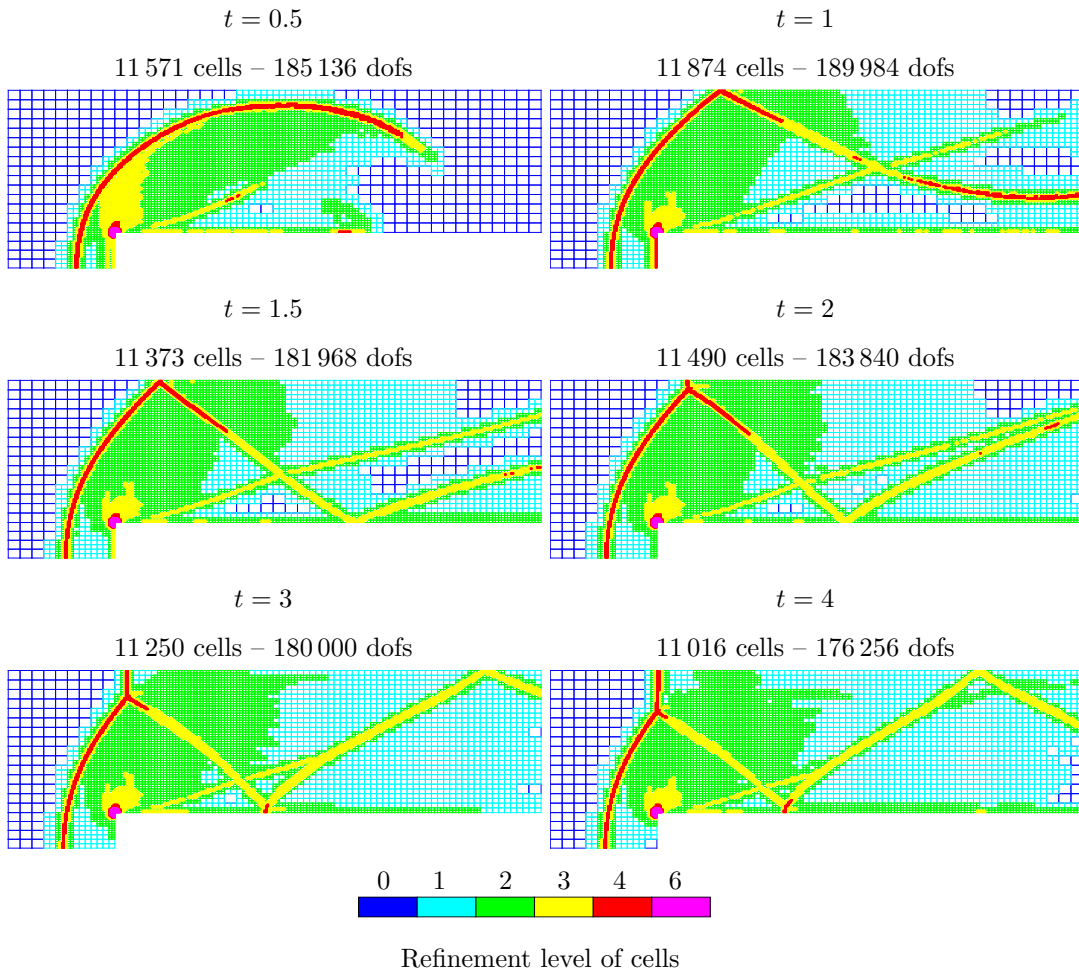
Finally, Fig. 5.44 illustrates the changes in the dynamic grid over time, following the discontinuities and allowing a high resolution of these.

Figure 5.43.: CFL numbers achieved for the forward-facing step problem at $t = 4$.



Further, for comparative purposes, time evolution of the density contours for this benchmark problem can be found in [WC84, AMJ14]. In Sec. D a similar set of results, using a

Figure 5.44.: Time evolution of the adaptive grid implemented for the forward-facing step problem.



uniform grid with $h \approx 1/150$, is presented.

5.3.5. Axisymmetric free underexpanded air-jet

In this section the releasing of a fluid from a round nozzle (or hole) and their free expansion into a quiescent medium is modeled. Depending of the value of the ratio between the static pressure at the exit of the nozzle (p_e) to the ambient pressure (p_∞), named nozzle pressure ratio NPR , the jet obtained can be: subsonic ($NPR < 1$), moderately underexpanded ($1.1 \lesssim NPR \lesssim 3.0$), highly underexpanded ($2.0 \lesssim NPR \lesssim 4.0$) or very highly underexpanded ($NPR \gtrsim 4.0$) [FPGB15].

The structure formed by the underexpanded circular-jets have been studied widely theoretically, experimentally and numerically. A good description of the supersonic jet structures can be found in [CSG66, AN59, DS71]. A actual and complete review about experimental and computational researches of underexpanded jets is found in [FPGB15].

In underexpanded case, as soon as the fluid come up from the nozzle exit, it is accelerated through a Prandtl–Meyer expansion located at the lip of the nozzle, until a supersonic speed and away from the axis line. These expansion waves cross the core of the jet until reach the constant-ambient pressure streamline, where these are reflected as compression waves (see Fig. 5.45). These acoustic waves coalesce to conform a oblique shock, commonly named *intercepting shock*, *incident shock* or *barrel shock*. For moderately underexpanded jets the intercepting shock reaches the axis line, and then it is reflected producing another oblique shock (usually called *reflected shock*).

For moderately underexpanded jets, along the axis line a high expansion, followed by a recompression of the fluid, is produced. At $NPR \approx 2$ the limit value of recompression for conical shocks is reached, and then the recompression is given now through a normal shock [DS71]. Thus, for $NPR > 2$ the barrel shock is not longer regular and a normal shock, termed *Mach disk*, is formed. The Mach disk is a slightly curved shock, with form of a lens, and normal to the axis line. Under this condition, both the position from the nozzle exit and the diameter of this disk increase as the NPR grows. For very highly underexpanded jets, this shock cannot be considered as normal shock and its curvature should be taken in account [DS71]. Here, the meeting point of the barrel shock, the reflected shock and the Mach disk is named *triple point*.

Further, in the point where the reflected shock reaches the jet boundary, a new expansion fan is developed, and so the previous cell structure is repeated until the viscous effects become dominant and where the jet is perfectly expanded. As higher NPR values are taken, the lower the number of repetitions of this shock-structure, and finally the potential core of the jet will be conformed by a single cell [FPGB15].

Thus, as soon as the fluid emerges from the nozzle exit, it suffers an isentropic expansion, accelerating until it reaches supersonic speeds (or in some cases hypersonic speeds), as well as very low values of density, pressure and temperature at the core of the jet (see Fig. 5.46). Further, when the fluid crosses the Mach disk, it is suddenly recompressed to match the ambient pressure, and it decelerates into a subsonic regime. On the other hand, the flow between the intercepting shock and the jet boundary is also supersonic, but with a lower Mach number than the core of the jet, and it is still supersonic after crossing the reflected shock. Thus, in the downstream of the jet, an inner subsonic region, as well as an outer supersonic region, appears. These two regions are separated by a slip line along which a shear layer is developed.

For this test case the very highly underexpanded jet problem presented in [OKM⁺08] is modeled. In this problem the exhausting of an ideal gas through a round nozzle, as well as

Figure 5.45.: Highly underexpanded jet structure. Here, (a) shows the Prandtl–Meyer expansion, (b) denotes the expansion waves, (c) is the compression waves, (d) shows the jet boundary, (e) indicates the intercepting shock (barrel shock), (f) is the reflected shock, (g) marks the triple point, (h) points out the Mach disk, (i) shows the expansion fan, (j) represents the outer shear layer, (k) denotes the inner jet core and (l) marks the outer jet region.

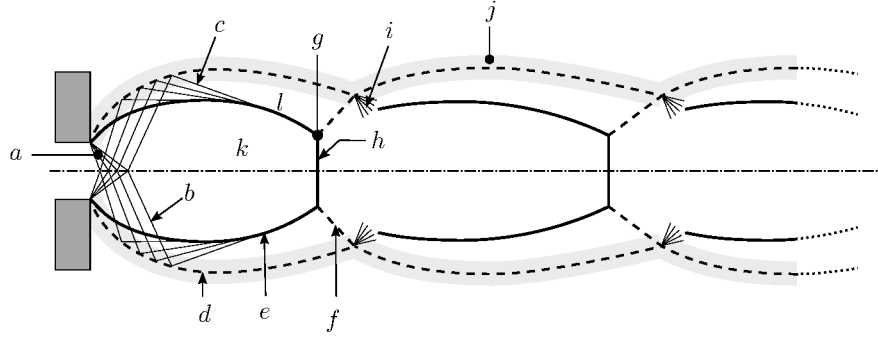
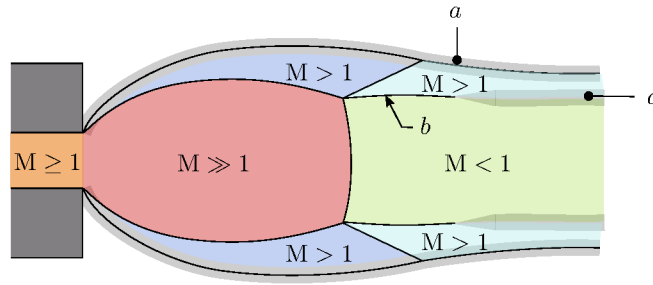


Figure 5.46.: Extremely underexpanded jet structure. Here, (a) shows the outer shear layer, (b) points out the slip line, and (c) marks the inner shear layer.



its expansion into a quiescent ambient, is simulated. For this case a high nozzle pressure ratio $NPR = 10.0$ is used.

For the modeling of this axisymmetric problem a 2D domain was employed. The domain illustrated in Fig. 5.47 represents the generator plane (by rotational sweeping) of the expansion chamber. For this problem a round nozzle with diameter $D_e = 1\text{mm}$ is used. Here, the boundary $\partial\Omega$ is defined as

$$\partial\Omega = \Gamma_{IN} \cup \Gamma_O \cup \Gamma_W \cup \Gamma_A,$$

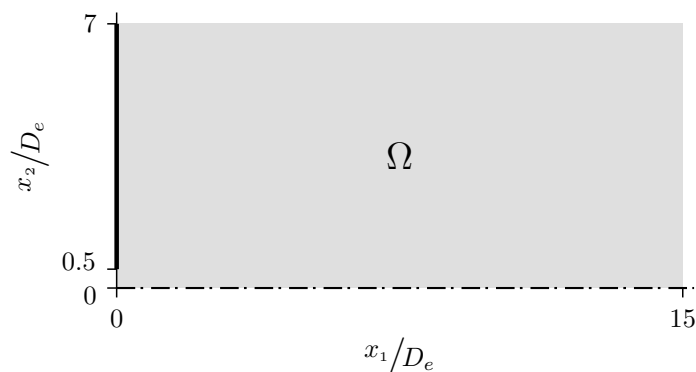
where

- $\Gamma_{IN} = \{\mathbf{x} \in \mathbb{R}^2 : x_1 = 0, x_2 \in (0, D_e/2)\}$ represents the inlet boundary,
- $\Gamma_O = \{\mathbf{x} \in \mathbb{R}^2 : x_1 \in (0, 15D_e], x_2 = 7D_e\} \cup \{\mathbf{x} \in \mathbb{R}^2 : x_1 = 15D_e, x_2 \in (0, 7D_e]\}$ is outlet boundary,
- $\Gamma_W = \{\mathbf{x} \in \mathbb{R}^2 : x_1 = 0, x_2 \in [D_e/2, 7D_e]\}$ is the wall at left of the domain, and
- $\Gamma_A = \{\mathbf{x} \in \mathbb{R}^2 : x_1 \in [0, 15D_e], x_2 = 0\}$ is the axis line.

On the inflow boundary Γ_{IN} , i.e. the nozzle exit, a sonic condition ($M_e = 1$) was imposed. Thus, on the inlet boundary values for pressure $p_e = 1.1\text{MPa}$, temperature $\theta_e = 300\text{K}$

and velocity $\mathbf{v}_e = (347.0 \ 0)^T$ m/s were prescribed. On the outlet boundary Γ_O an atmospheric pressure value $p_\infty = 0.11$ MPa was taken. Additionally, a reflective slip boundary condition is used to simulate the behavior on the axis line Γ_A . For the initial condition values for density $\rho^0 = 1.28$ kg/m³, velocity $\mathbf{v}^0 = 0$ and pressure $p^0 = 0.11$ MPa were taken homogeneous for the entire domain.

Figure 5.47.: Domain used for the underexpanded air-jet problem.



The problem was modeled using the axisymmetric version of the Navier–Stokes equations (see Sec. 2.1.8) for an ideal gas with $\gamma = 1.4$.

For the spatial semi-discretization a formulation with \mathbb{P}^1 *DG*-elements was implemented. In this formulation a numerical flux based on the *FCT* strategy presented in Eq. 3.70, using *AUSM*-flux (see Sec. 3.3.3) as high order term and the Vijayasundaram-flux (see Sec. 3.3.2.2) as low order term, was incorporated. Additionally, in order to improve the shock-capturing performance of this discretization, an adaptive grid strategy, based on the technique presented in Sec. 3.5 and using the gradient of the Mach number as indicator, was used here. In this test the stabilization parameters ν_1 and ν_2 were set to 1.5 and the non-symmetric interior penalty method *NIPG*, with $\sigma = 0.1$, was used for the viscous terms. The Fig. 5.48 shows a grid endowed with 792 quadrilateral elements, which it was employed as initial mesh.

On the other hand, the Rosenbrock–Euler method with $\Delta t = 0.1\mu$ s was employed for time integration. The simulation was computed until $t = 300\mu$ s, when the jet was considered totally developed and stable in time.

Figure 5.48.: Initial grid used for solving the underexpanded air-jet problem.

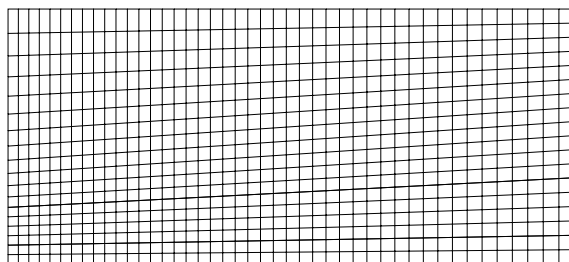


Fig. 5.49 shows the evolution over time for Mach number contours. The graphic at quasi-steady state ($t = 300\mu\text{s}$) shows a very fast, regular and isentropic expansion of the fluid, as soon as it exits from the nozzle. This expansion accelerates the fluid in the core of the jet until $M = 4.7$, while the pressure drops until $p \approx 4\text{kPa}$ and temperature decrease dramatically from 300K to a minimum of 56K (see Fig. 5.50). Then the fluid is suddenly recompressed throughout the Mach disk, up to a pressure even higher than the pressure chamber (p_∞). Subsequently slight expansion-compression cycles are repeated along the core of the jet. As for the extremely high underexpanded jet, a unique shock-cell structure is present.

From the third frame in Fig. 5.49, at $t = 20\mu\text{s}$, it is possible to identify the formation of a shock-cell with all elements cited previously, i.e. a jet boundary, a Mach disk, a barrel shock and a reflected shock (along with the triple point where these shocks are connected).

A zoom in to the near-field at $t = 300\mu\text{s}$ can be found in Fig. 5.51. This graphic shows that the shock-cell structure of the predicted jet presents a good agreement with the numerical results reported in [OKM⁺08, DKBT11], specially in terms of the Mach disk, but also in the position and form of the barrel shock and the reflected shock. The detail at the left of Fig. 5.51 shows the good resolution reached in the approximation of the expansion fan at the nozzle lip.

Here, one of the most relevant characteristics to be analyzed is the position of the Mach disk. A lot of experimental studies point out that the position of the Mach disk is mainly governed by the pressure ratio NPR , although it also can be influenced by the Mach number at nozzle exit, or even by the geometry of the nozzle [FPGB15]. Further, different empirical formulas have been proposed to calculate the L_{MD} , which is the position of the Mach disk measured from nozzle exit. According to [FPGB15], one of the most precise expressions to estimate L_{MD} is the equation presented in [CSG66]

$$\frac{L_{MD}}{D_e} = \left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{\frac{\gamma}{2(\gamma-1)}} \sqrt{\frac{p_e}{2.4p_\infty}}, \quad (5.9)$$

which returns the value $L_{MD} \approx 2.8D_e$ for the simulated problem.

On the other hand, from [FPGB15] the size of the Mach disk, for the actual pressure ratio, can be predicted using the empirical equation

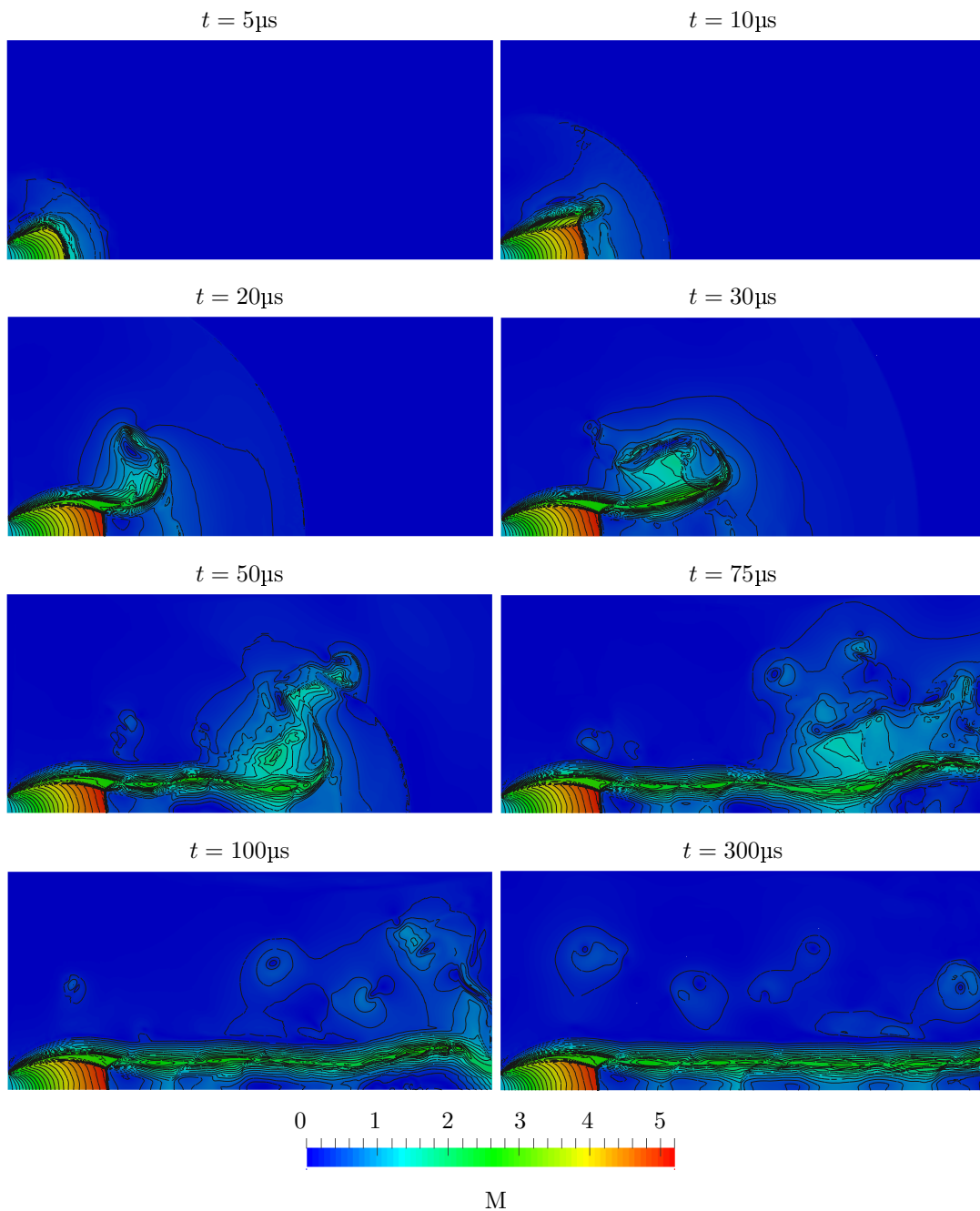
$$\frac{D_{MD}}{D_e} = 0.06 \left(1 + \frac{\gamma - 1}{2} M_e^2\right)^{\frac{\gamma}{\gamma-1}} \frac{p_e}{p_\infty}, \quad (5.10)$$

which returns the value $D_{MD} \approx 1.1D_e$ for this case.

Thus, comparing the empirical predictions calculated via Eq. 5.9 and Eq. 5.10, with the numerical results shown in Fig. 5.51 and Fig. 5.52, which presents a Mach disk, with diameter $D \approx 1.3D_e$, located at $x_1 \approx 3.0D_e$, it is possible to note the good accuracy reached with our numerical implementation.

Fig. 5.52 shows the predicted profiles for the Mach number, temperature and pressure along the axis line. Here, in the rapid expansion region (near field) these profiles coincide well with the results reported in [DKBT11]. Away from the Mach disk, the profiles follow the same oscillatory behavior that represents cyclic expansion-compression processes, although the position and magnitude of this events are slightly different.

Figure 5.49.: Mach number contours at different times for the underexpanded air-jet problem using an adaptive grid. Thirty contourlines uniformly spaced from 0 to 5.2.



In Fig. 5.53 the evolution over time of the adaptive grid is shown. Here, it is possible to identify that the finest elements are located mainly near to the shock waves, although also in the region of the outer and inner shear layer (see markers *(a)* and *(c)* in Fig. 5.46). The formation of the shear layers is especially clear in the schlieren plots of Fig. 5.54. Additional results about evolution over time of temperature, pressure, density and velocity for this problem can be found in Sec. E.

Figure 5.50.: Contours of temperature (a), pressure (b) and density (c) predicted at $t = 300\mu\text{s}$ for the underexpanded air-jet problem.

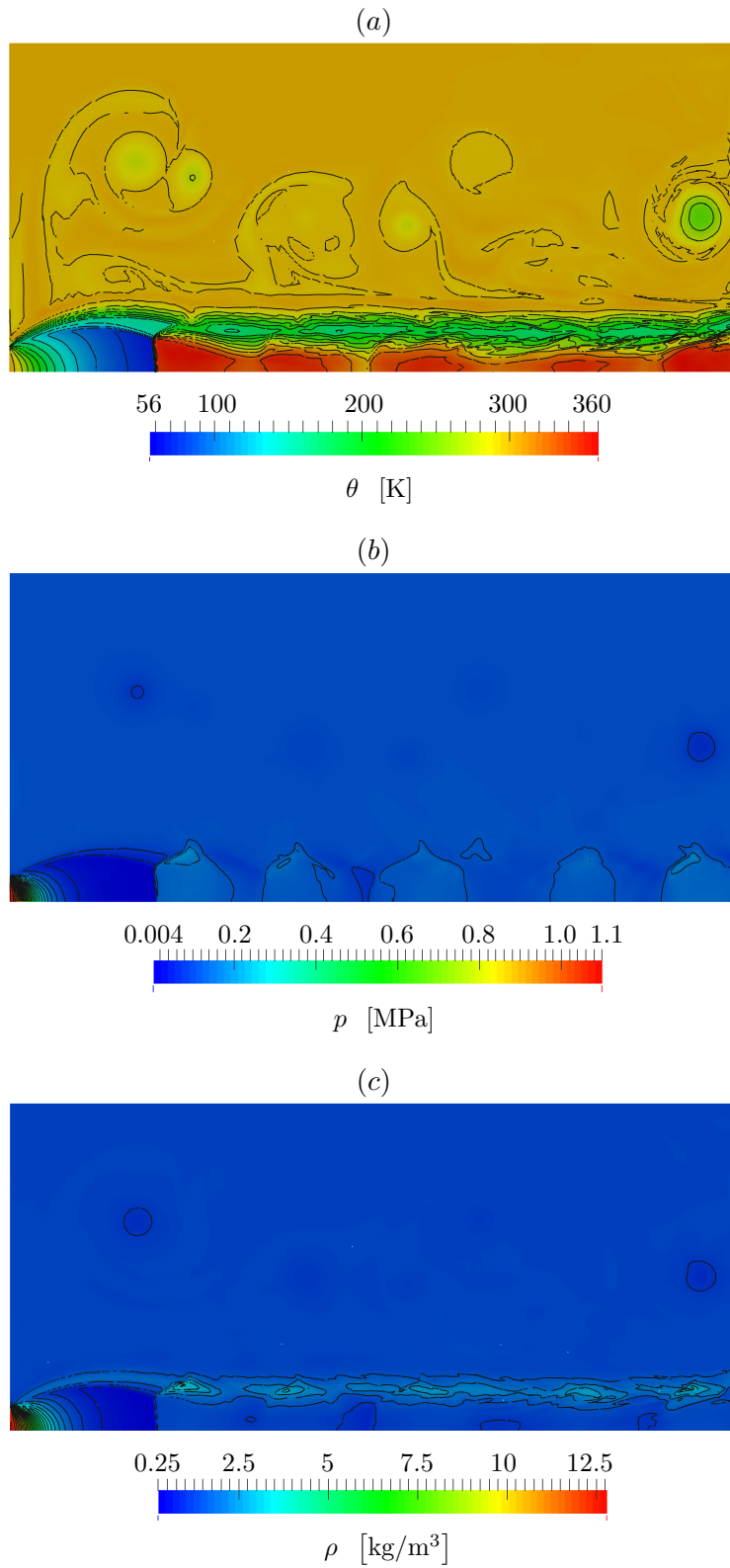


Figure 5.51.: Predicted Mach disk position for the underexpanded air-jet problem at $t = 300\mu\text{s}$ using an adaptive grid. White curves show the results reported in [OKM⁺08].

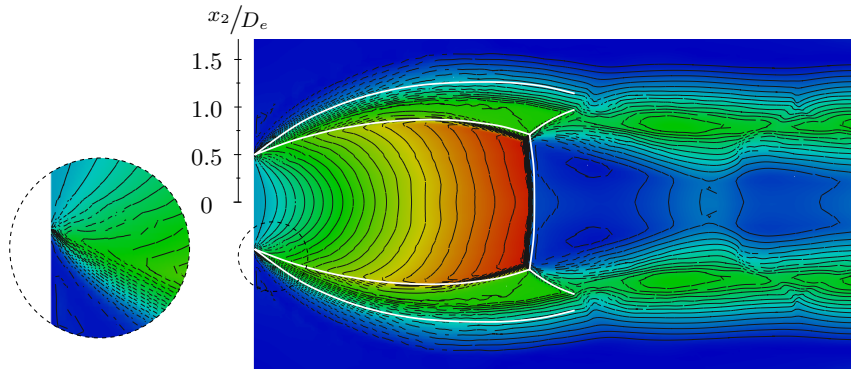
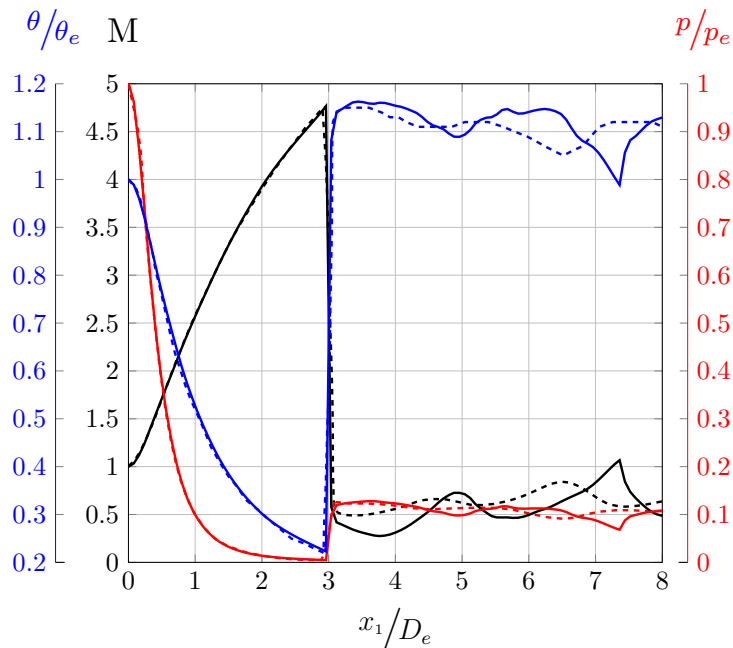


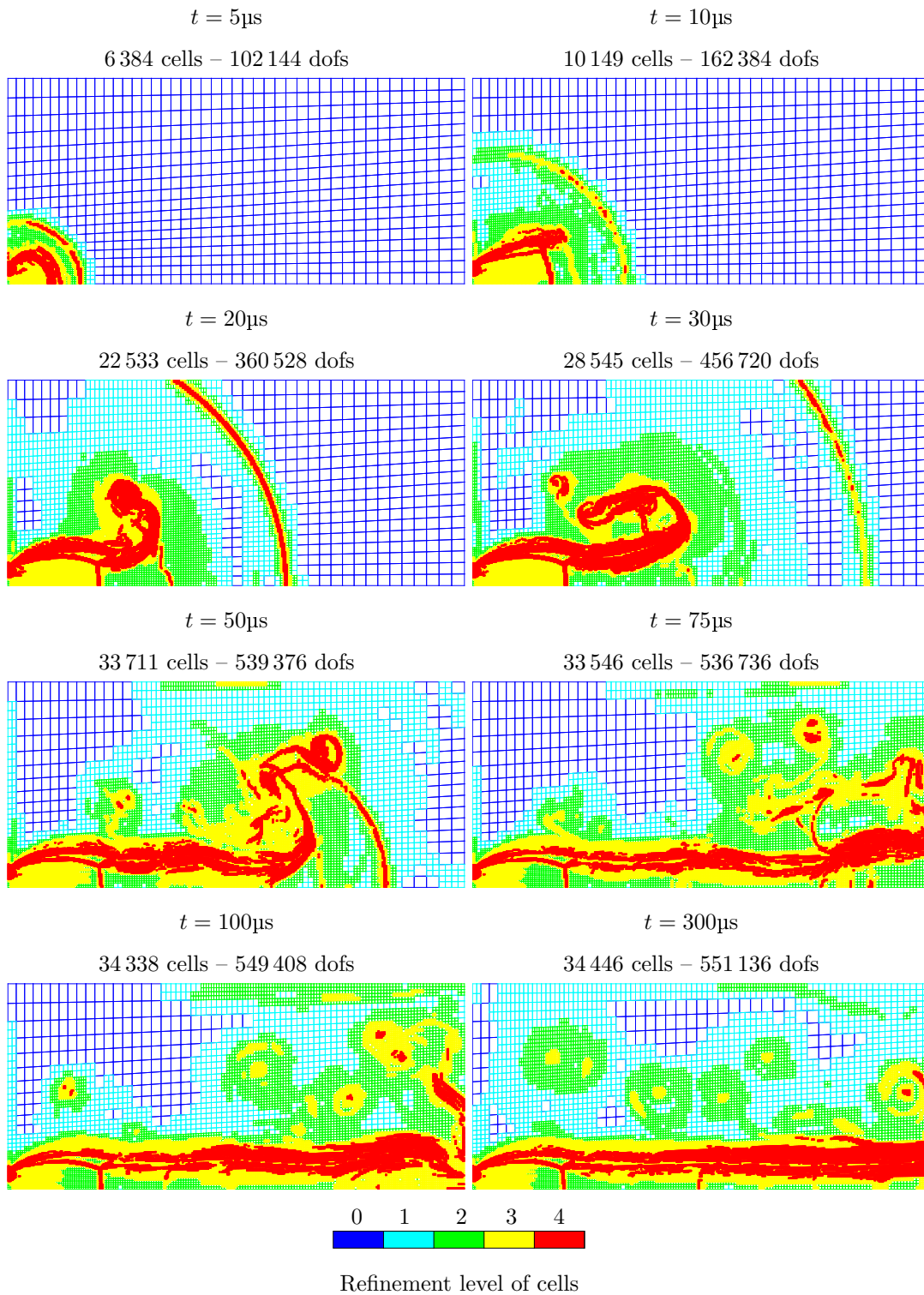
Figure 5.52.: Predicted profiles of Mach number (solid black), dimensionless temperature (solid blue) and dimensionless pressure (solid red) along the center line for the underexpanded air-jet problem. Dashed lines represent the profiles reported in [DKBT11].



5.4. Highly underexpanded CO₂-jet

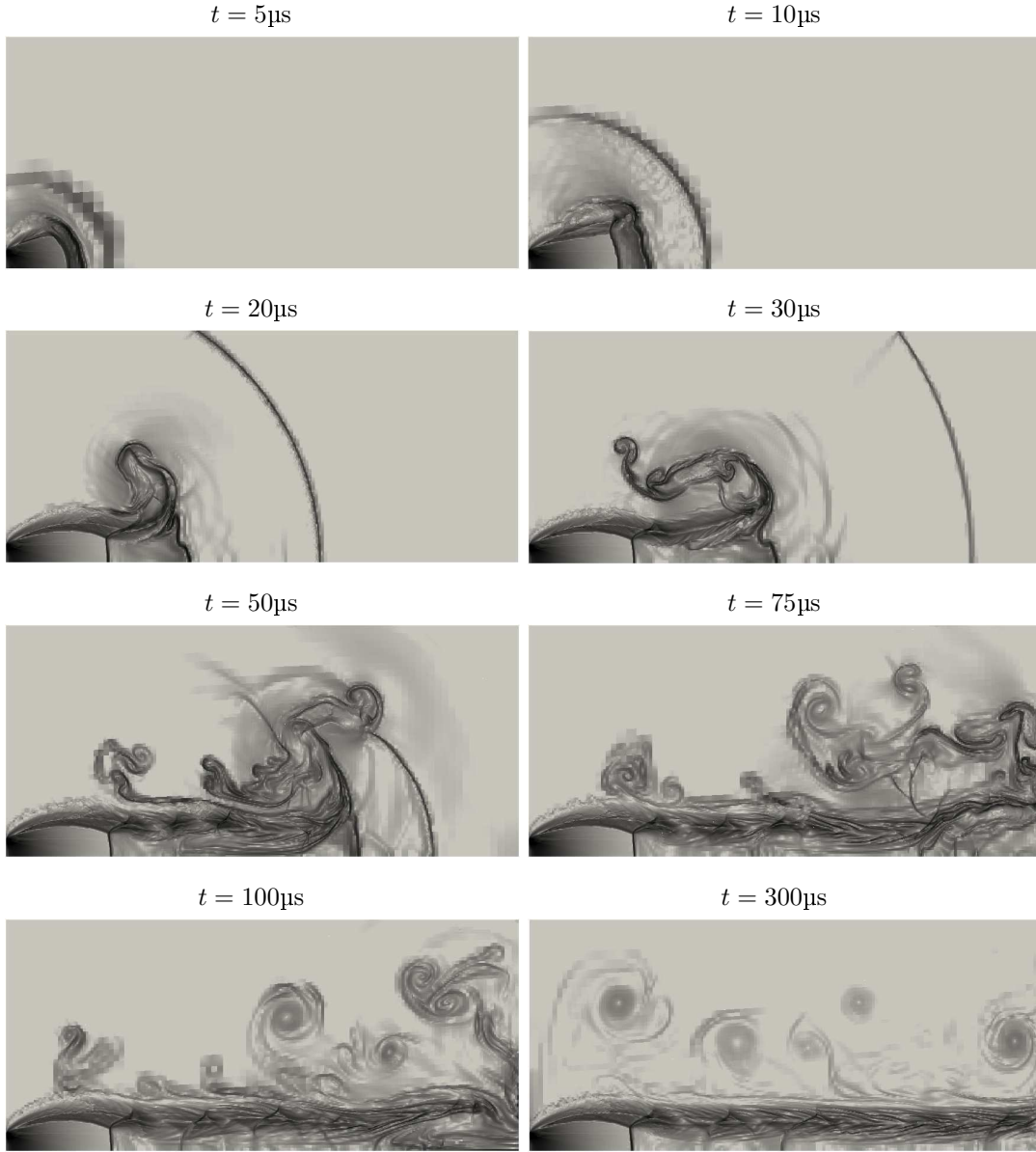
In this section the results found in the simulation of the rapid expansion of a supercritical carbon dioxide flow are presented. In this problem, originally formulated in [HTS03], supercritical CO₂ flows from a high-pressure line ($p_o = 20\text{MPa}$, $\theta_o = 393\text{K}$), throughout a capillary nozzle with diameter $D_e = 50\mu\text{m}$ and length $50\mu\text{m}$, to a quiescent and low-pressure medium. The conditions at this expansion chamber are $p_\infty = 0.1\text{MPa}$ and $\theta_\infty = 298\text{K}$. In Fig. 5.55 the domain used for the simulation is shown. The domain consists of an axisymmetric 2D model with three zones: a stretch of the high-pressure line ($x_1/D_e < 2.2$), the capillary nozzle ($2.2 < x_1/D_e < 3.2$) and the expansion chamber ($x_1/D_e > 3.2$). Here,

Figure 5.53.: Time evolution of the adaptive grid implemented for the underexpanded air-jet problem.



a region with diameter of $40D_e$ and length equal to $45D_e$ is taken to represent the expansion chamber.

Figure 5.54.: Schlieren-plots computed at different times for the underexpanded air-jet problem using an adaptive grid.

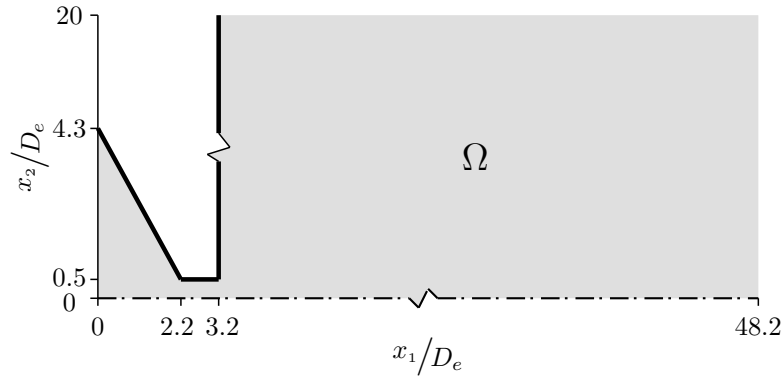


For this problem the boundary $\partial\Omega$ is defined as

$$\partial\Omega = \Gamma_{IN} \cup \Gamma_O \cup \Gamma_W \cup \Gamma_A,$$

where

- $\Gamma_{IN} = \{\mathbf{x} \in \mathbb{R}^2 : x_1 = 0\}$ represents the inlet boundary,
- $\Gamma_O = \{\mathbf{x} \in \mathbb{R}^2 : x_1 \in [3.2D_e, 48.2D_e], x_2 = 20D_e\} \cup \{\mathbf{x} \in \mathbb{R}^2 : x_1 = 48.2D_e\}$ is outlet boundary,
- $\Gamma_W = \{\mathbf{x} \in \mathbb{R}^2 : x_1 \in (0, 2.2D_e], x_2 = -\frac{3.8}{2.2}x_1 + 4.3\} \cup \{\mathbf{x} \in \mathbb{R}^2 : x_1 \in (2.2D_e, 3.2D_e], x_2 = 0.5D_e\} \cup \{\mathbf{x} \in \mathbb{R}^2 : x_1 = 3.2D_e, x_2 \in (0.5D_e, 20D_e)\}$ is an adiabatic wall, and

Figure 5.55.: Domain used for simulating the underexpanded CO₂-jet problem.

- $\Gamma_A = \{\mathbf{x} \in \mathbb{R}^2 : x_1 \in (0, 48.2D_e), x_2 = 0\}$ is the axis line.

The problem was modeled using the set of axisymmetric Navier–Stokes equations presented in Sec. 2.1.8. In order to close these set of equations, the *egB* equation for CO₂, presented in Sec. 2.4.3, was used as equation of state. Likewise, Eq. 2.109 was employed to compute the dynamic viscosity of CO₂, and Eq. 2.111 was implemented to calculate the heat conductivity of the fluid.

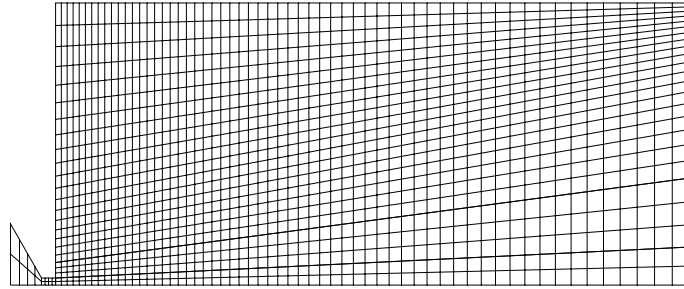
On the inflow boundary Γ_{IN} , values for pressure $p_o = 20\text{MPa}$, temperature $\theta_o = 393\text{K}$ and velocity $\mathbf{v}_o = (0 \ 0)^T \text{ m/s}$ were prescribed. On the outlet boundary Γ_O atmospheric values for pressure $p_\infty = 0.1\text{MPa}$ and temperature $\theta_\infty = 298\text{K}$ were taken. A reflective slip boundary condition is used to simulate the behavior on the axis line Γ_A , while conditions for adiabatic walls were applied on Γ_W .

For the initial conditions, a supercritical state with $p^0 = 20\text{MPa}$ and $\theta^0 = 393\text{K}$ was taken in $\{\mathbf{x} \in \mathbb{R}^2 : x_1 < 2.7D_e\}$, while a subcritical state given by $p^0 = 0.1\text{MPa}$ and $\theta^0 = 298\text{K}$ was assumed in the rest of the domain. Besides, in the complete domain we set the states initially to be at rest, i.e., $\mathbf{v}^0 = 0$.

The domain in space was discretized with a structured mesh using 1432 \mathbb{P}^1 *DG*-elements (see Fig. 5.56). Onwards it will be named *base grid*. The time integration was computed using a Rosenbrock–Euler method with $\Delta t = 5\text{ns}$, until a final time $T = 54\mu\text{s}$, where the jet can be considered as totally developed.

Initially four simulations were done, two of them used Lax–Friedrichs numerical flux, and the others using the *AUSM⁺-up* flux (see Sec. 3.3.3). Likewise, in these simulations two grids were employed, one with 5728 \mathbb{P}^1 elements, generated by the global refinement of the base grid, and the second with 22912 \mathbb{P}^1 elements, obtained with a new global refinement of the previous grid. In all cases a discontinuity indicator based on Eq. 3.71 and the jump in the magnitude of the velocity was employed. Here, the preliminary tests show that this quantity in the discontinuity indicator allows to identify the elements next to the shock structure in the core of the jet. Further, in these tests we found that a limit value $g_{\text{lim}} = 0.15$ for the discontinuity indicator is adequate for the precise identification of the critical cells. In these cells the stabilization terms in Eq. 3.75 and Eq. 3.76 were introduced using the parameters $\nu_1 = \nu_2 = 1.3$. For all simulations in this section the penalization parameter σ in the *NIPG* method was taken equal to 0.1.

Figure 5.56.: Grid, forming by 1432 \mathbb{P}^1 cells, used as initial mesh for the highly underexpanded CO_2 -jet problem.



As was discussed in Sec. 5.3.5, this type of jets present a single shock-cell structure formed by a jet boundary, a barrel shock, a reflective shock and a Mach disk. Fig. 5.57 shows the Mach number contour lines for the four cases previously defined. In these plots can be observed that the shock structure in the core of the jet was captured in all the cases. However, corresponding to problems with discontinuous solutions the resolution of the shocks, and in this case specially in the shock barrel, improves rapidly when a finer grid is used. Here, the solutions obtained with the *AUSM* flux look slightly more diffusive than the equivalent ones computed with Lax–Friedrichs flux. Besides, in the graphics can be appreciated that the position and diameter of the Mach disk are essentially constant independently of the grid resolution.

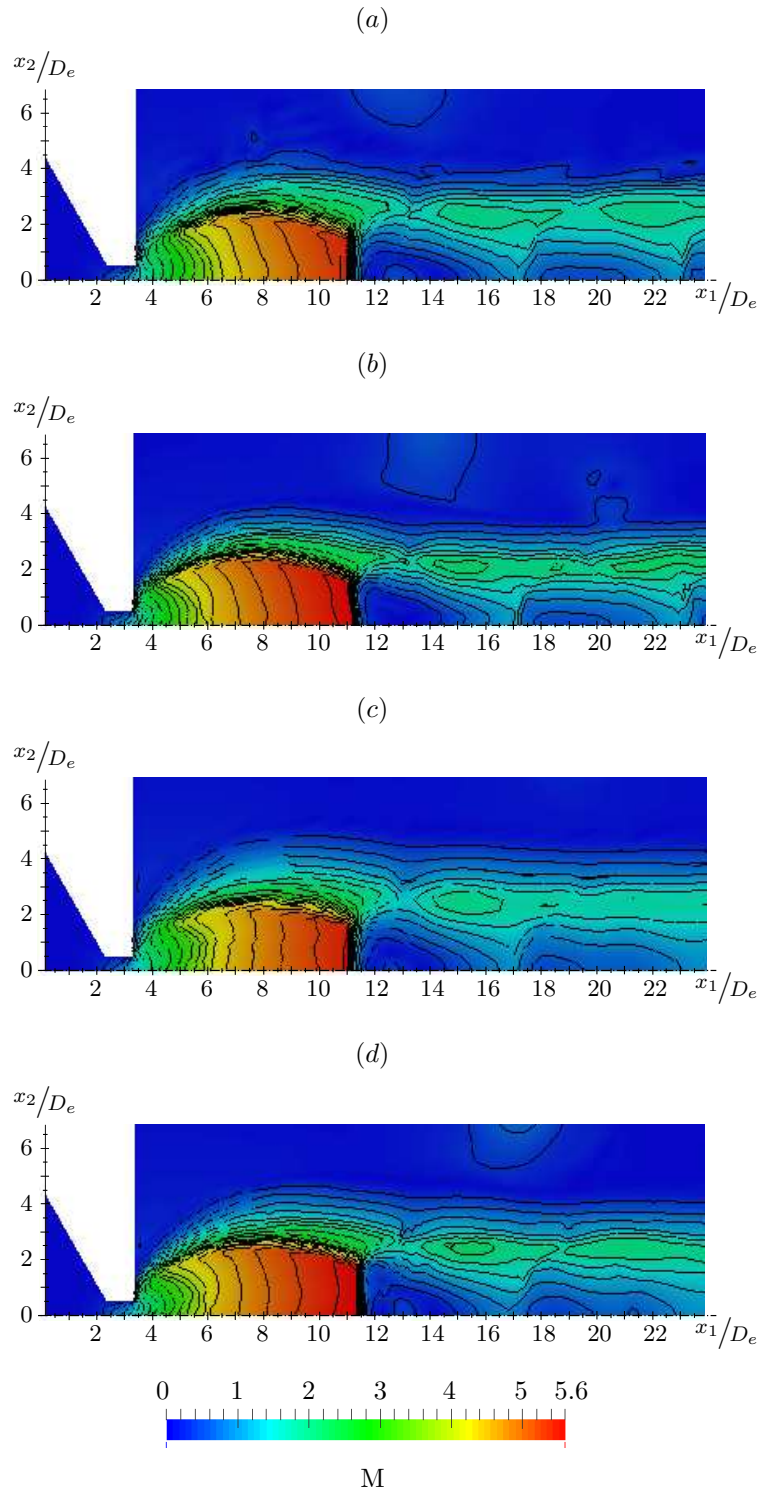
As in all previous test case, the shock capturing strategy, based on the idea of introducing artificial viscosity terms only in the elements where the *G*-switch variable is activated, allows to avoid the spurious oscillations without a considerable reduction in the resolution of the discontinuities. In Fig. 5.58 the behavior of the *G* variable in the near field of the jet can be observed.

In Table 5.4 the CPU time demanded in the computation of each of these simulations is reported. In this table the time required for the assembly process, which includes the computation of the vector $\mathfrak{F}(\mathbf{w}_n)$ and its Jacobian, is included as percentage of the total CPU time, as well as the time required for the Arnoldi decomposition algorithm. The results in the table show that the most relevant computational cost in each time step is the assembly process. Further, the total time consumed using the *AUSM* flux is approximately 14% larger with respect to the time demanded using the Lax–Friedrichs flux in the case of the coarsest grid, and approximately 10% higher in the case of the finest grid.

As in the previous test cases, an adaptive grid strategy can be useful to improve the shock capturing capability. Thus, in the next simulation an adaptive strategy based on the Mach number gradient was implemented. Starting from the base grid in Fig. 5.56, i.e., considering in this condition $r_L = 0$, the parameter $r_{L,\max}$ was set to 3 for this simulation (see Algorithm 1). Besides, in order to increase the resolution of the expansion fan presents in the lip of the nozzle, a local refinement was applied in a small area around this point (see Fig. 5.65).

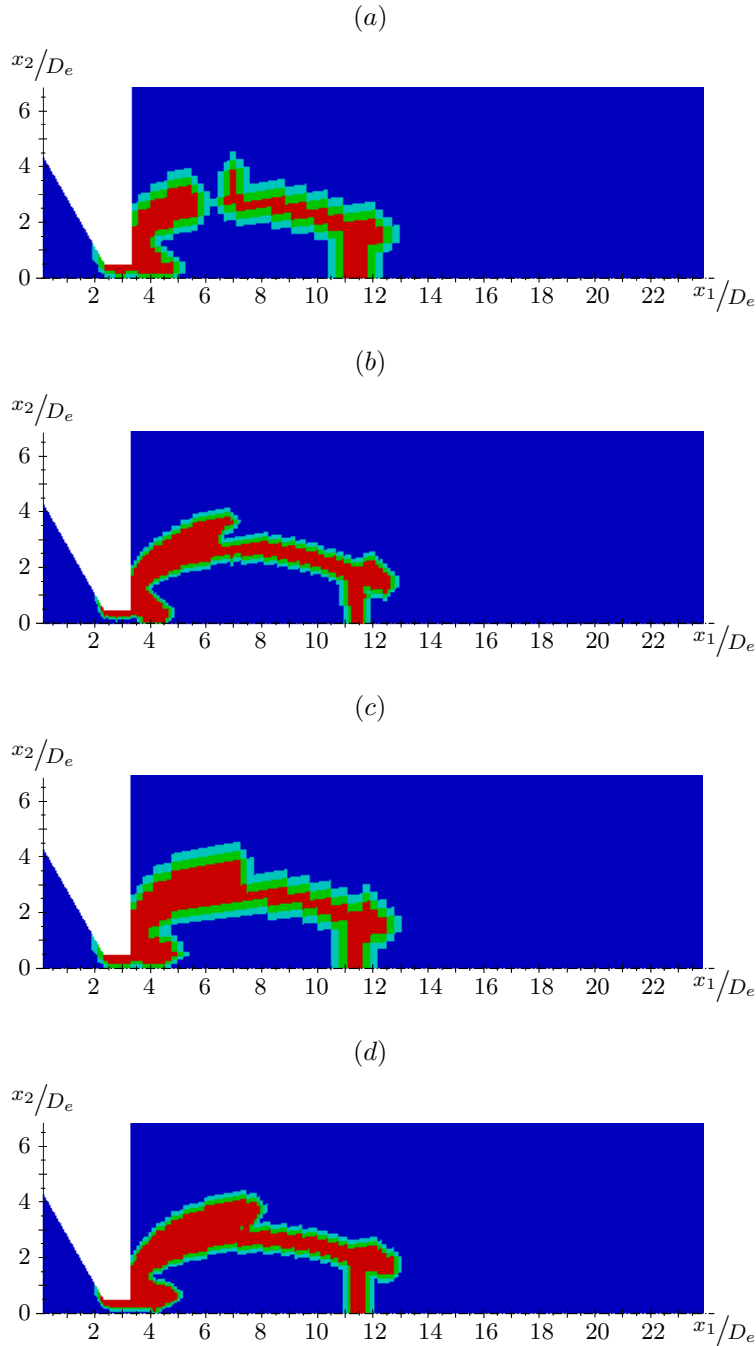
In this case we used a numerical flux based on the *FCT* strategy (see Eq. 3.70), with the *AUSM*-flux as high order term and the Lax–Friedrichs flux as low order component. The parameters for the shock capturing are conserved from the previous simulations, as well as the time marching scheme and the time step size.

Figure 5.57.: Mach number contours computed using (a) Lax–Friedrichs numerical flux and 5728 \mathbb{P}^1 elements, (b) Lax–Friedrichs numerical flux and 22912 \mathbb{P}^1 elements, (c) the *AUSM*-flux with 5728 \mathbb{P}^1 elements, and (d) the *AUSM*-flux with 22912 \mathbb{P}^1 elements.



The simulation, that consumed a total of 40.4h of computation time using 8 cores in an *Intel Core i7-3770@3.4GHz* processor, employed 58.5% of this CPU time for assem-

Figure 5.58.: G -variable computed using (a) Lax–Friedrichs numerical flux and 5728 \mathbb{P}^1 elements, (b) Lax–Friedrichs numerical flux and 22912 \mathbb{P}^1 elements, (c) the $AUSM$ -flux with 5728 \mathbb{P}^1 elements, and (d) $AUSM$ -flux with 22912 \mathbb{P}^1 elements.



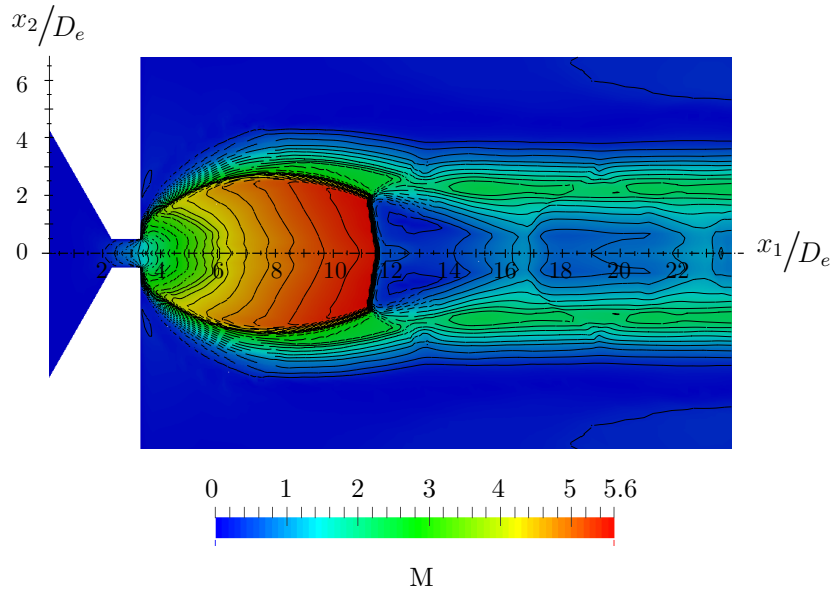
bling tasks, a 38.7% computing the Krylov projections and the remaining time in other tasks.

Fig. 5.59 shows a close up of the shock-cell structure formed in this case. In this figure it is possible to identify all shocks formed in the near field of this type of jets, and especially the Mach disk, which was accurately captured .

	Lax–Friedrichs		AUSM ⁺ -up	
	Grid A	Grid B	Grid A	Grid B
Assembly time [%]	85.7	74.3	82.5	81.9
Time for Krylov projections [%]	13.4	23.3	14.9	16.4
Time in other processes [%]	0.9	2.4	2.6	1.7
Total time [h]	6.5	31.5	7.4	34.8
Number of vectors in \mathcal{K}_m	34	67	42	53

Table 5.4.: CPU times employed in the solution of the underexpanded CO₂-jet using different grids and numerical fluxes. Here, Grid A is conformed by 5 728 cells – 91 648 dofs, whereas the Grid B is composed 22 912 cells – 366 592 dofs. The CPU times reported correspond to computations using 8 cores in an *Intel Core i7-3770@3.4GHz* processor.

Figure 5.59.: Predicted Mach disk position for the underexpanded CO₂-jet problem using an adaptive grid.



Using the analytical expression given in [CSG66], and recommended by [FPGB15], to estimate L_{MD}

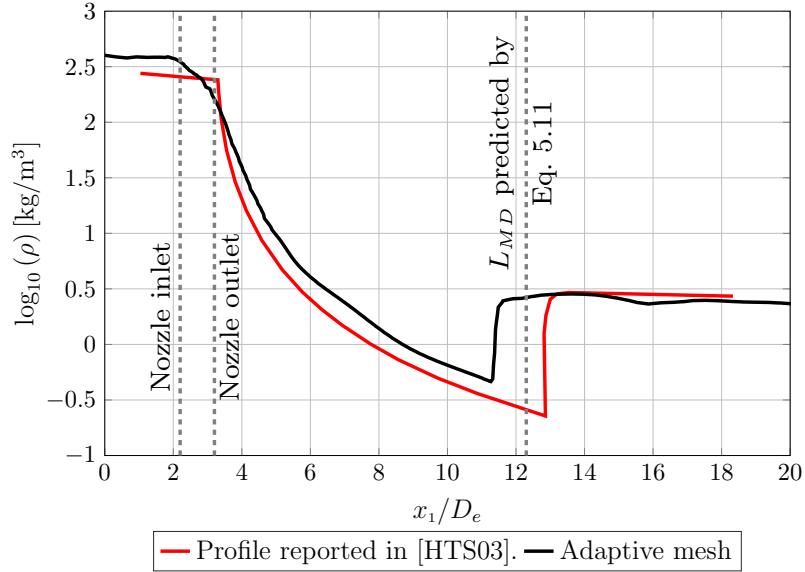
$$\frac{L_{MD}}{D_e} = \sqrt{\frac{p_o}{2.4p_\infty}} \approx 9.1, \quad (5.11)$$

as well as the empirical equation proposed by [FPGB15] to compute D_{MD}

$$\frac{D_{MD}}{D_e} = 0.28 \left(\frac{p_o}{p_\infty} \right)^{0.53} \approx 4.6, \quad (5.12)$$

and comparing these values with the corresponding quantities for the Mach disk position $x_1 \approx 8.3D_e$ and Mach disk diameter $D_{MD} \approx 4.0D_e$, taken from Fig. 5.59 (see also Fig. 5.61), it is possible to note the good precision achieved in the capturing of this shock

Figure 5.60.: Predicted density profile along the center line for the highly underexpanded CO₂-jet problem.



wave.

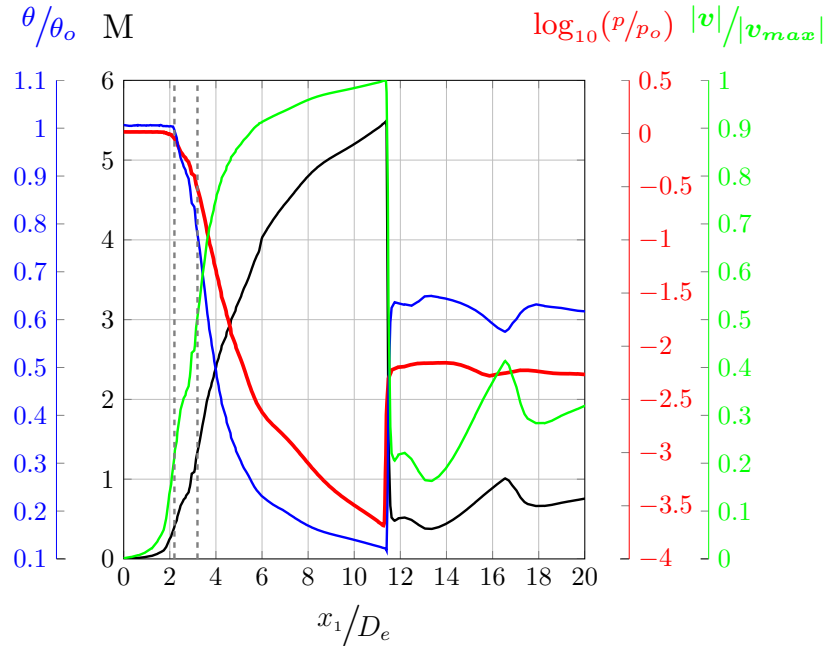
In Fig. 5.61 the density profile, along the center line, is plotted. Here, it is possible to identify an early expansion along the nozzle, as is reported in many works ([HHH⁺01, HTS03]), followed by a strong expansion after the nozzle until to reach the Mach disk. In this figure the profile found is compared with results reported in [HTS03], where a $1D$ model was implemented to simulate this problem. As is shown in Fig. 5.59 the rapid and isentropic expansion after the nozzle accelerates the CO₂-jet until to find a maximum speed of 585m/s, approximately $M = 5.6$ (see Fig. 5.61 and Fig. F.12). At the same time, CO₂ in the core of the jet suffers a strong drop of pressure and temperature, until to find a supercooled state at $\theta \approx 39\text{K}$ and $p \approx 3.0\text{kPa}$, just before the Mach disk. Behind the Mach disk, smoother expansion-compression cycles repeat downstream until the viscous component dominates the jet behavior. Because of the high pressure ratio NPR prescribed in this case, only one proper shock-cell structure is formed [FPGB15].

In Fig. 5.62 the thermodynamic behavior of CO₂, along the axis line at $t = 54\mu\text{s}$, is plotted into phase diagrams $p - \theta$ and $p - V$. There, the point X_I represents the nozzle inlet, behind this point, and along the nozzle, the flow suffers an early expansion, crossing the critical pressure and changing the phase from supercritical to gas. At the exit of the nozzle (point X_{II}), the CO₂ in gas phase begins a rapid expansion, and soon behind this point, condensed liquid phase is produced.

In Fig. 5.62 (b) it is possible to observe where the expansion process (solid black-line) crosses the saturated vapor line and then CO₂ in liquid phase starts to be formed. Thus, during part of this isentropic expansion, which is represented by the piece of solid black-line inside of saturation dome, two phases gas and liquid coexist in equilibrium.

Later, as is noted in Fig. 5.62(a), the free expansion curve crosses the melting line and then starts the solidification of the liquid phase. Further, pressure and temperature still decrease, even below of the triple point, until a minimum value for pressure ($p = 3\text{kPa}$) and temperature ($\theta = 39\text{K}$) is reached (point X_{III}). After this point, a recompression pro-

Figure 5.61.: Predicted profiles of Mach number, temperature, pressure and velocity along the center line for the underexpanded CO₂-jet problem. Here $\theta_o = 393\text{K}$, $p_o = 20\text{MPa}$ and $|\mathbf{v}_{max}| = 591.0\text{m/s}$.



duced on the Mach disk increases the pressure and temperature, crossing the sublimation line, until the point X_{IV} in the vapor region. Thus, the sudden compression on the Mach disk dissolves the solid phase, bringing it to gaseous phase again.

The blue lines in Fig. 5.62 (a) and (b) represent the gas behavior downstream of the Mach disk, i.e. after the X_{IV} point. Here, it is possible to observe some smooth and successive expansion-compression cycles, which further do not change the phase of the fluid.

In Fig. 5.63 a general view of the Mach number evolution along the time is plotted. Likewise, contours of temperature, pressure and density, at $t = 54\mu\text{s}$, are shown in Fig. 5.64. In Sec. F illustrative graphics for time evolution of temperature, pressure, density and velocity, are included. Finally in Fig. 5.65 the evolution along the time of the adaptive grid computed is illustrated.

Figure 5.62.: Thermodynamic behavior of CO₂-flow along the axis line at $t = 54\mu\text{s}$. In (a) the expansion process is represented into the pressure-temperature diagram, while in (b) this process is presented in the pressure-specific volume diagram. Here $X_I = 2.2D_e$ is the nozzle inlet, $X_{II} = 3.2D_e$ is the nozzle outlet and $X_{III} = X_{IV} = 11.5D_e$ represent the beginning and the end of the recompression process (Mach disc), respectively. Physical phases are defined by \mathcal{SC} (supercritical state), \mathcal{G} (gas phase), \mathcal{L} (liquid phase) and \mathcal{S} (solid phase).

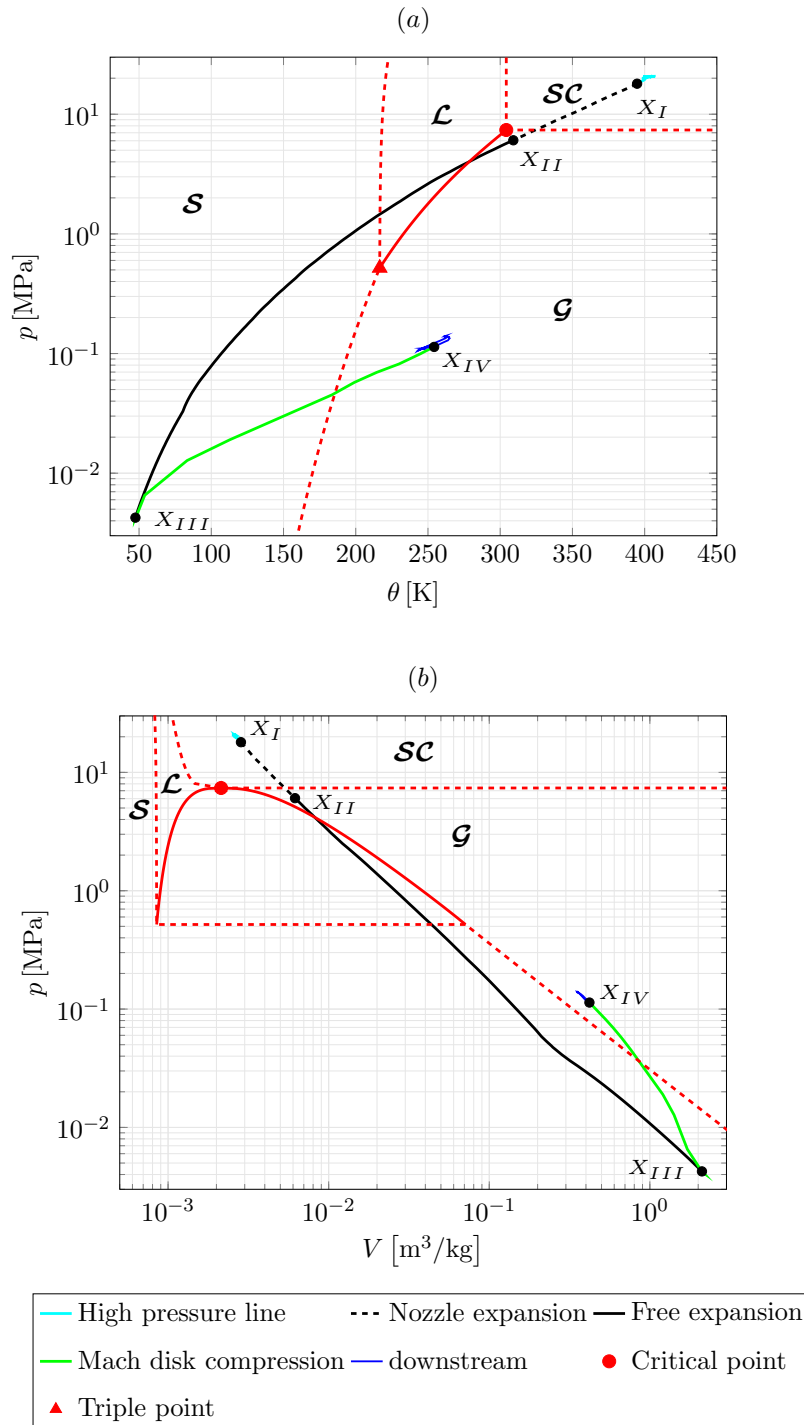


Figure 5.63.: Mach number predicted at different times for the highly underexpanded CO_2 -jet problem using an adaptive grid. Twenty five contourlines uniformly spaced from 0 to 5.6.

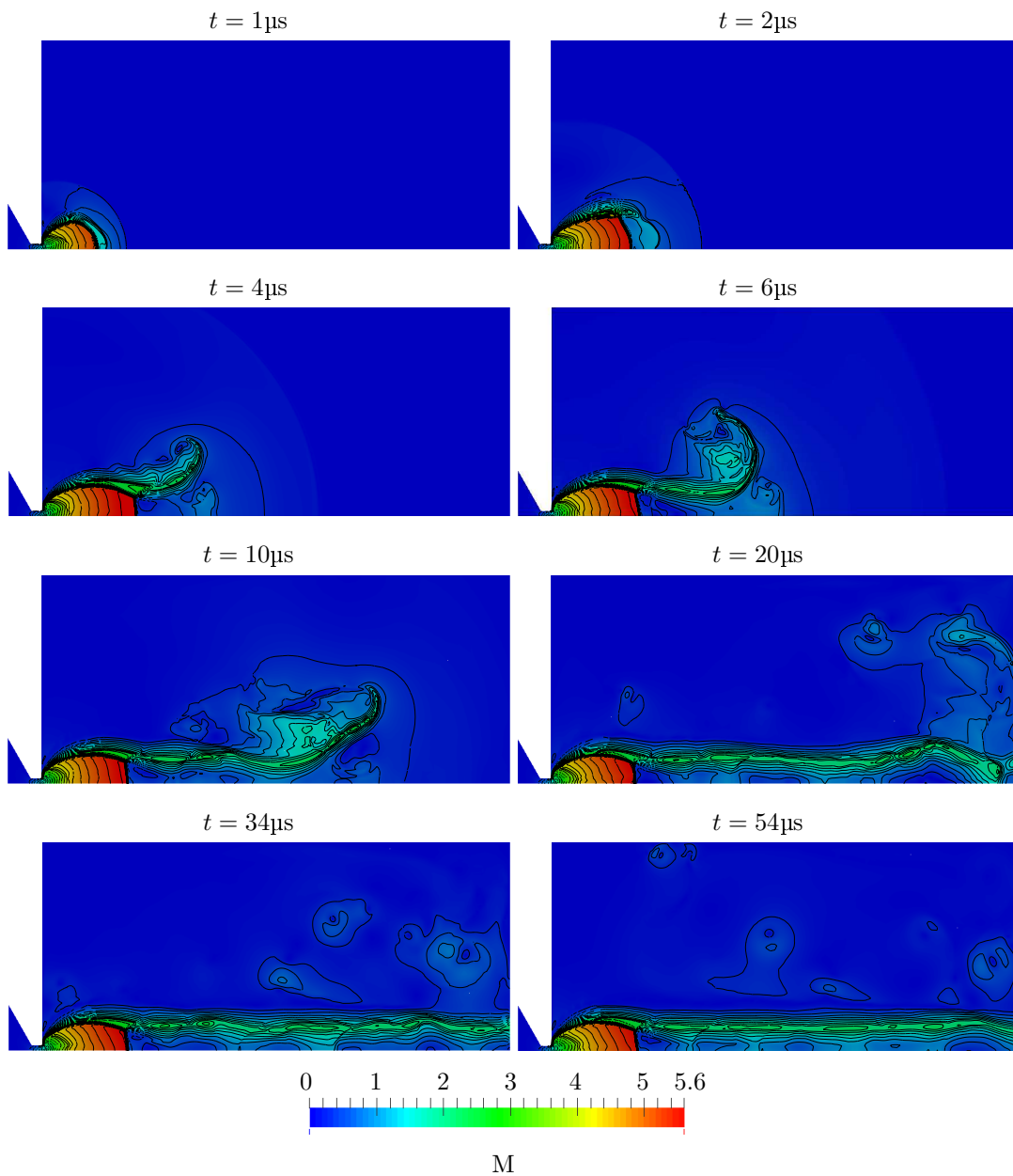


Figure 5.64.: Contourlines of temperature (a), pressure (b) and density (c) predicted at $t = 54\mu\text{s}$ for the highly underexpanded CO_2 -jet problem.

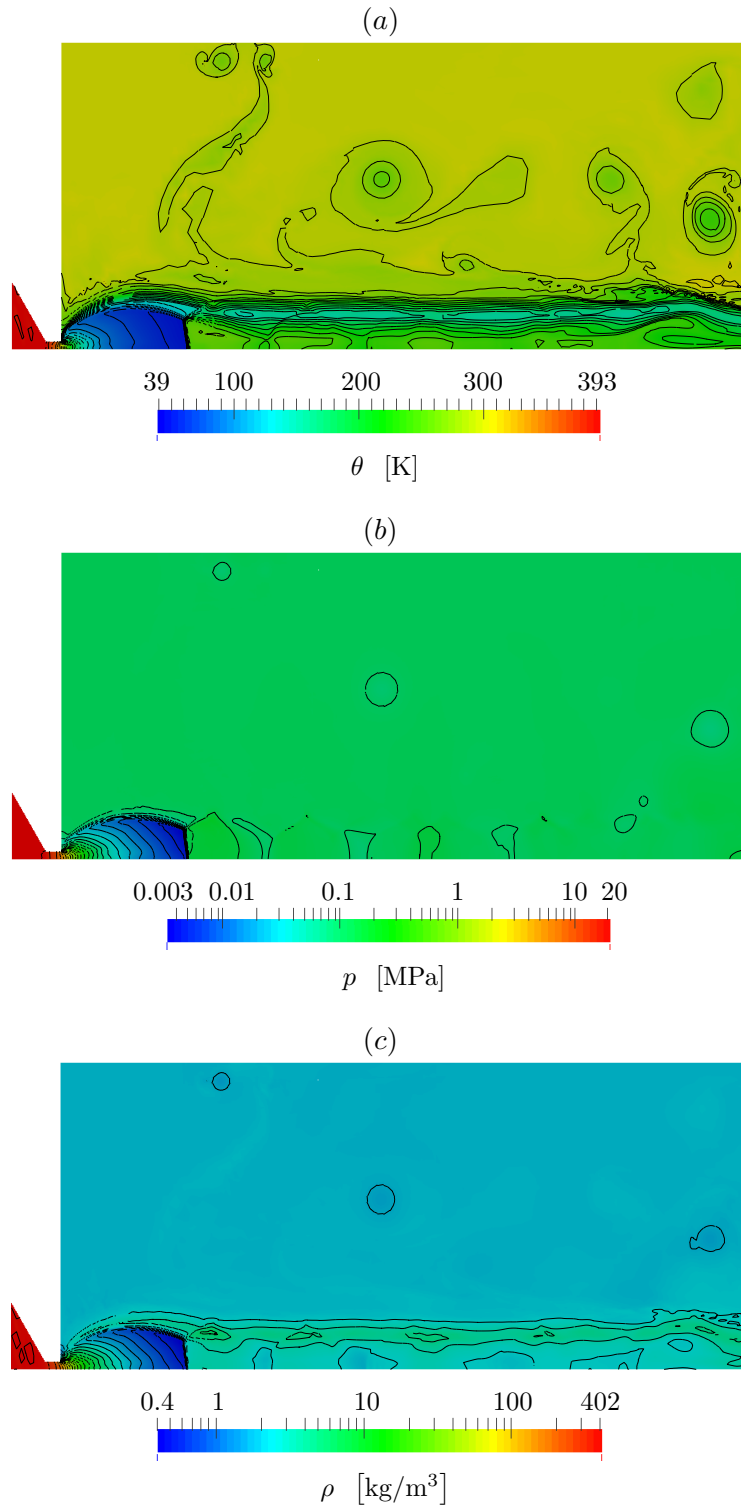
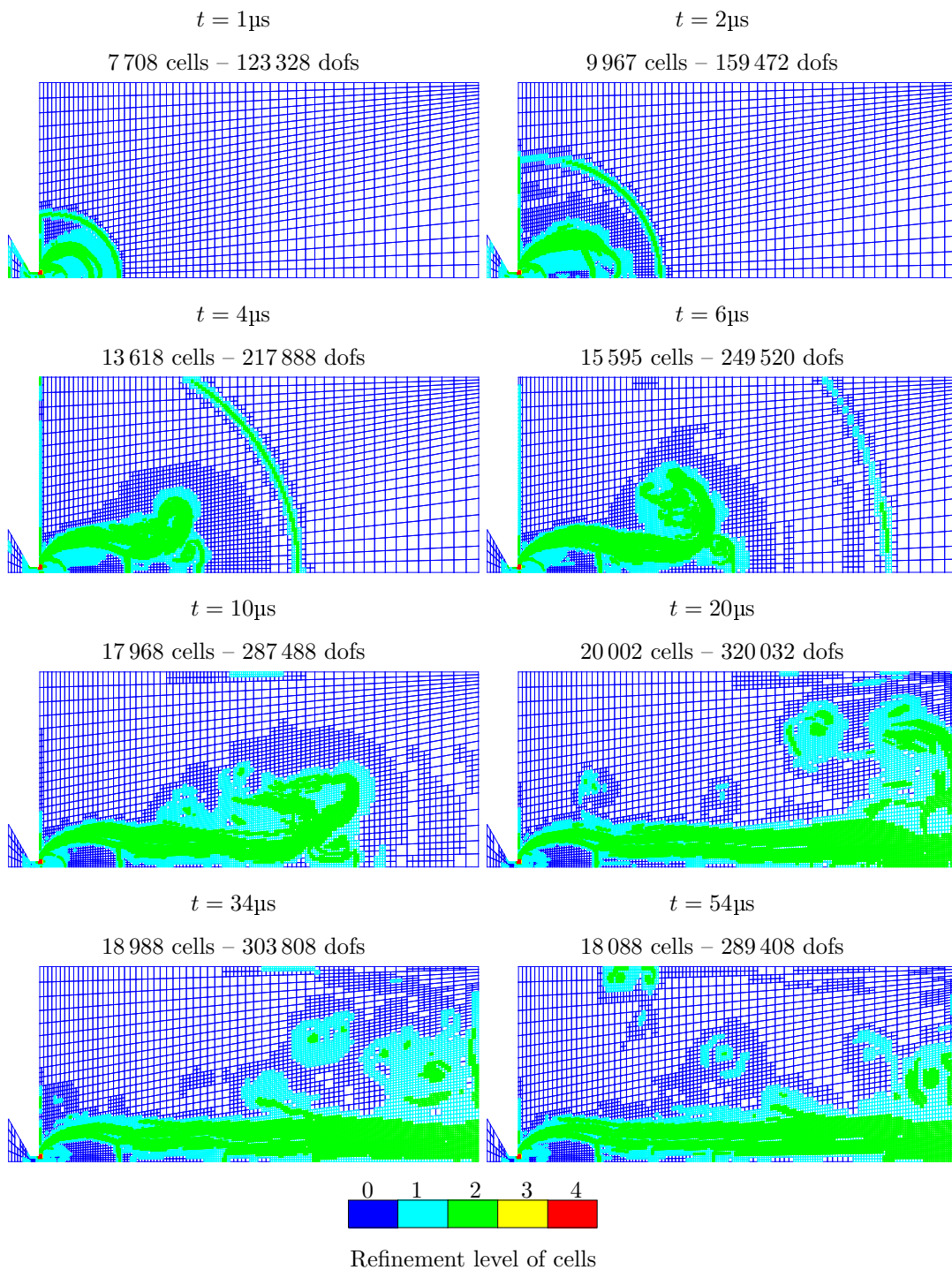


Figure 5.65.: Time evolution of the adaptive grid implemented for the highly underexpanded CO₂-jet problem.

6. Conclusions

6.1. Summary

A numerical model to simulate the formation of a very highly underexpanded jet was developed. The jet formation was analyzed in the context of the process called rapid expansion of supercritical solutions (*RESS*), using carbon dioxide as solvent. The mathematical model consisted of the Navier–Stokes equations along with the extended generalized Bender equation of state *egB-EOS* [Ben71]. This equation of state was used because it covers a wide range of states (supercritical, vapor, liquid and vapor-liquid mixtures), allows to compute precisely caloric properties (as the specific heats and speed of sound) and has relatively low complexity.

For the numerical model, an axisymmetric spatial-domain, conformed by a section of the high pressure line, a capillary nozzle and an expansion chamber, was employed. The domain was discretized using discontinuous Galerkin finite elements. A very precise shock capturing strategy was employed to identify the cells in the neighborhood of the discontinuities, preserving the stability of the method in this regions and preventing spurious entropy production. Additionally, in order to improve the resolution of the shocks captured, a simple mesh adaptive procedure was incorporated.

Considering that very high speeds, which corresponds to supersonic or even hypersonic regimens, and locally refined meshes are present in this case, especially in the core of the jet, a stable and efficient time marching scheme was implemented. Here, an time integration method, named exponential Rosenbrock–Euler, was used. Also all the matrix exponential terms involved in this method are approximated via projection onto Krylov subspaces, employing the Arnoldi decomposition algorithm.

The compressible flow solver was implemented in C++ language using several libraries as `deal.ii` [BHK07], `Trilinos` [HBH⁺05] and Intel[®] TBB [Int16]. Diverse benchmark problems were simulated using this solver in order to evaluate not only the accuracy of the approximation but also the computational efficiency at subsonic, supersonic and subsonic-supersonic conditions.

The solution achieved for the very highly underexpanded CO₂-jet problem, generated by a pre-expansion pressure equal to 20 MPa and a pressure at expansion chamber of 0.1 MPa, shows that the complex shock-structure present in this kind of rapid expansions

was accurately captured. All shocks, and specially the Mach disk, do not exhibit neither spurious oscillations or overly artificial viscosity, which shows the good performance in the shock capturing strategy employed. The position and size of the Mach disk captured fit well with the empirical expressions used to predict this values. The density profile along the axis line agrees with the results showed in [HTS03], obtained from a one-dimensional model. Likewise, the profiles of pressure, temperature and Mach number plotted along the axis line, show an early expansion within the nozzle, where the fluid cross the critical point toward the the gas phase, as well as a stronger rapid-expansion after the nozzle exit. Product of this last expansion, the temperature and pressure of CO₂ drop until 39 K and 3 kPa, respectively, whereas the Mach number finds a maximum value $M = 5.6$, just before to find the Mach disk.

6.2. Contributions

The main contributions of the present work can be summarized as follows:

- Although the underexpanded jets have been studied both experimental and numerically, there are few research works where these type of jets, under real operative condition of the *RESS* process, are simulated. Besides, the numerical results presented in these works exhibit either spurious oscillations or smearing discontinuities. Here, an accurate numerical model for a very highly underexpanded CO₂-jet, using real gas considerations, was developed. All shocks, produced in this kind of rapid expansion, were captured with high resolution thanks to an precise shock capturing strategy along with a simple and useful adaptive grid technique.
- A robust exponential time integrator scheme was incorporated into the compressible flow solver. The use of this kind of methods in the different test cases simulated showed that they are very useful in problems with strong time-step restrictions derived from either high speed characteristics or very fine meshes (as in case of local refinement). In fact, although there exists some few publications where exponential time integrators are used in the context of fluid dynamic problems, mainly focused on the incompressible flows, in knowledge of the author the use of exponential time integrators for the compressible high-speed flow problems has not been consider before. Further, the current implementation allows easily to incorporate other different exponential time-integrator schemes.
- A flexible and robust computational implementation for solving compressible flows was developed and tested. The current implementation incorporates spatial discretization, for 2D, 3D and axisymmetric geometries, using discontinuous Galerkin method and diverse numerical fluxes (central scheme fluxes, upwind fluxes and *AUSM*-type fluxes). As previously was mentioned, for the time-marching procedure the exponential Rosenbrock–Euler method was included, as well as the classical forward/backward Euler methods. Additionally, some schemes of the family called strong stability preserving Runge–Kutta (*SSPRK*) are available to use. Further, a shock capturing strategy and a adaptive grid refinement technique were also included in order to ensure a high resolution in regions with discontinuous solutions. The flexible computational implementation allows to use directly or to adapt the solver to simulate different kind of problems with compressible flows.

6.3. Future research

The work developed in this thesis could be extended in two main directions, on one hand, improving the mathematical model for simulating the *RESS* process, and on the other hand enhancing the numerical implementation.

6.3.1. Mathematical model

- After an exact prediction of the hydrodynamic field, and in order to complete the modeling of the *RESS* process, submodels for the nucleation and the particle growth should be included. Therefore the nucleation-growth models developed by Türk and Helfgen in [Tür99, HTS03, HHH⁺01, HTS00], as well as the experimental results presented in these works, are a very important starting point.
- Some authors report that the condensed CO₂-particles into the nozzle and in the later rapid expansion can modify the size-particle distribution in the *RESS* process [YF15]. Thus, a multiphase model to analyze the influence of the solvent condensation could be adequate to improve the current model. Here, multiple researches in condensing-steam flows (e.g. within divergent-convergent nozzle or in a turbine cascade) can be a good guideline. In this topic the works published by Halama et al. [HBF10, HDF⁺10, HF14] are recommended.
- Although the *egB-EOS* employed in this work allows to cover a wide range of pressure and temperature, even below the triple point, the adiabatic expansion, generated by very high nozzle pressure ratios (*NPR*), produces extremely low values of temperatures and pressure in the core of the jet. At this conditions a sublimation process can occur and some solid crystals can be formed. Hence, an better model to predict the carbon dioxide behavior under this low pressure-temperature conditions is required. A good candidate to model this behavior is the modified Peng–Robinson equation presented in [MBM13].

6.3.2. Numerical implementation

The compressible flow solver implemented, because of its versatility, can be used to simulate a wide variety of problems. Nevertheless, the implementation can be improved in aspects as:

- The parallelization strategy, based on shared memory, showed to be efficient for small to medium size problems, i.e., when the number of unknowns allowed the computation in a desktop computer (or a workstation). However, large scale problems demand to use computer clusters, and a distributed-memory strategy should be used. The finite element library `deal.ii`, used in this work, incorporates classes for this kind of parallelization, but the implementation of the exponential time integrator in this case is not straightforward.
- The complexity of the equation of state, as well as its efficient implementation into the solver, are two very important aspects in the global performance of the code. Therefore, it is important to explore and evaluate other alternatives to incorporate the real gas considerations to the compressible solver. An alternative way to introduce the EOS is to build look-up tables on a thermodynamic region of interest and pre-compute all thermophysical properties using an accurate EOS or specialized library [PSPR15, PRC12]. Then the values required on runtime can be obtained by simple interpolation during the simulation of the real gas flow. Another alternative is to use all expressions derived from ideal gas assumption but computing the heat capacity ratio from a real gas EOS [LLS90, GW89a].
- Finally, in some problems the turbulence is an important effect to be considered, as for example in the simulation of flows into the combustion chambers. Hence, incorporating a turbulence model into this compressible flow solver can offer new horizons of applications.

Bibliography

- [AAR76] S. Angus, B. Armstrong, and K. de Reuck, *International thermodynamic tables of the fluid State – 3 carbon dioxide*. Pergamon, 1976.
- [ACF96] M. Arioli, B. Codenotti, and C. Fassino, “The Padé method for computing the matrix exponential,” *Linear Algebra and its Applications*, vol. 240, pp. 111–130, 1996. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0024379594001901>
- [AH09] H. Atkins and B. Helenbrook, Eds., *Super-Convergence of Discontinuous Galerkin Method Applied to the Navier–Stokes Equations*, ser. 19th AIAA Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, 2009. [Online]. Available: <http://dx.doi.org/10.2514/6.2009-3787>
- [AMJ14] H. Abbassi, F. Mashayek, and G. B. Jacobs, “Shock capturing with entropy-based artificial viscosity for staggered grid discontinuous spectral element method,” *Computers & Fluids*, vol. 98, pp. 152–163, 2014, 12th USNCCM mini-symposium of High-Order Methods for Computational Fluid Dynamics – A special issue dedicated to the 80th birthday of Professor Antony Jameson. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045793014000292>
- [AN59] T. C. Adamson and J. Nicholls, “On the structure of jets from highly underexpanded nozzles into still air,” *Journal of the Aerospace Sciences*, vol. 26, no. 1, pp. 16–24, 1959, doi: 10.2514/8.7912.
- [Arn82] D. N. Arnold, “An interior penalty finite element method with discontinuous elements,” *SIAM Journal on Numerical Analysis*, vol. 19, no. 4, pp. 742–760, 1982. [Online]. Available: <http://dx.doi.org/10.1137/0719052>
- [AS72] V. Altunin and M. Sakhabetdinov, “Viscosity of liquid and gaseous carbon dioxide at temperatures 220–1300 K and pressure up to 1200 bar,” *Teploenergetika*, vol. 19, no. 8, pp. 85–89, 1972.
- [Bak97] T. J. Baker, “Mesh adaptation strategies for problems in fluid dynamics,” *Finite Elements in Analysis and Design*, vol. 25, no. 3, pp. 243–273, 1997. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168874X96000327>
- [Bat06] K. Bathe, *Finite Element Procedures*. Prentice Hall, 2006.
- [BB73] J. P. Boris and D. L. Book, “Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works,” *Journal of Computational Physics*, vol. 11, no. 1, pp. 38–69, 1973. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0021999173901472>

- [BD10] G. E. Barter and D. L. Darmofal, “Shock capturing with PDE-based artificial viscosity for DGFEM: Part I. Formulation,” *Journal of Computational Physics*, vol. 229, no. 5, pp. 1810–1827, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999109006299>
- [Ben71] E. Bender, “Die Berechnung von Phasengleichgewichten mit der termischen Zustandsgleichung dargestellt an den reinen Fluiden Argon, Stickstoff, Sauerstoff und an ihren Gemischen,” Habilitation, Ruhr-Universität Bochum, 1971.
- [BH07] Y. Bazilevs and T. Hughes, “Weak imposition of Dirichlet boundary conditions in fluid mechanics,” *Computers & Fluids*, vol. 36, no. 1, pp. 12–26, 2007, Challenges and Advances in Flow Simulation and Modeling. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045793005001258>
- [BHK07] W. Bangerth, R. Hartmann, and G. Kanschat, “deal.II - A general purpose object oriented finite element library,” *ACM Trans. Math. Softw.*, vol. 33, no. 4, pp. 24/1–24/27, 2007. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1268779>
- [BHS09] J. Banks, W. Henshaw, and J. Shadid, “An evaluation of the FCT method for high-speed flows on structured overlapping grids,” *Journal of Computational Physics*, vol. 228, no. 15, pp. 5349–5369, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999109002071>
- [Bis12] F. Bisetti, “Integration of large chemical kinetic mechanisms via exponential methods with Krylov approximations to jacobian matrix functions,” *Combustion Theory and Modelling*, vol. 16, no. 3, pp. 387–418, 2012. [Online]. Available: <http://dx.doi.org/10.1080/13647830.2011.631032>
- [BLR00] T. Braconnier, P. Langlois, and J. Rioual, “The influence of orthogonality on the Arnoldi method,” *Linear Algebra and its Applications*, vol. 309, no. 1, pp. 307–323, 2000. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0024379599001007>
- [BP92] Å. Björck and C. C. Paige, “Loss and recapture of orthogonality in the modified Gram–Schmidt algorithm,” *SIAM Journal on Matrix Analysis and Applications*, vol. 13, no. 1, pp. 176–190, 1992. [Online]. Available: <http://dx.doi.org/10.1137/0613015>
- [BR95] F. Bassi and S. Rebay, “Accurate 2D Euler computations by means of a high order discontinuous finite element method,” in *14th International Conference on Numerical Methods in Fluid Dynamics*. Springer Berlin, 1995, pp. 234–240.
- [BR97] —, “High-order accurate discontinuous finite element solution of the 2D Euler equations,” *Journal of Computational Physics*, vol. 138, no. 2, pp. 251–285, 1997. [Online]. Available: <http://dx.doi.org/10.1006/jcph.1997.5454>
- [BVM13] F. Bonelli, A. Viggiano, and V. Magi, “A numerical analysis of hydrogen underexpanded jets under real gas assumption,” *Journal of Fluids Engineering*, vol. 135, no. 12, pp. 121 101–1–121 101–11, 2013.
- [BZ09] K.-J. Bathe and H. Zhang, “A mesh adaptivity procedure for CFD and fluid-structure interactions,” *Computers & Structures*, vol. 87, no. 11–12, pp. 604–617, 2009, 5th MIT Conference on Computational Fluid and Solid Mechanics. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045794909000364>

- [Cas00] P. Castillo, “An optimal estimate for the local discontinuous Galerkin method,” in *Discontinuous Galerkin Methods: Theory, Computation and Applications*, B. Cockburn, G. E. Karniadakis, and C.-W. Shu, Eds. Springer Berlin, 2000, pp. 285–290. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-59721-3_23
- [CD99] B. Cockburn and C. Dawson, Eds., *Some Extensions Of The Local Discontinuous Galerkin Method For Convection-Diffusion Equations In Multidimensions*, ser. The Mathematics of Finite Elements and Applications X (MAFELAP 1999). Elsevier Science, 1999.
- [CDG08] B. Cockburn, B. Dong, and J. Guzmán, “Optimal convergence of the original DG method for the transport-reaction equation on special meshes,” *SIAM Journal on Numerical Analysis*, vol. 46, no. 3, pp. 1250–1265, 2008. [Online]. Available: <http://dx.doi.org/10.1137/060677215>
- [CG85] P. Colella and H. Glaz, “Efficient solution algorithms for the Riemann problem for real gases,” *Journal of Computational Physics*, vol. 59, pp. 264–289, 1985. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0021999185901469>
- [Cin06] P. Cinnella, “Roe-type schemes for dense gas flow computations,” *Computers & Fluids*, vol. 35, no. 10, pp. 1264–1281, 2006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045793005000800>
- [CL95] F. Coquel and M. Liou, “Hybrid upwind splitting (HUS) by a field-by-field decomposition,” National Aeronautics and Space Administration (NASA), Tech. Rep., 1995. [Online]. Available: <http://ntrs.nasa.gov/search.jsp?R=19950015540>
- [CL07] C.-H. Chang and M.-S. Liou, “A robust and accurate approach to computing compressible multiphase flow: Stratified flow model and AUSM+-up scheme,” *Journal of Computational Physics*, vol. 225, no. 1, pp. 840–873, 2007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999107000071>
- [CMV69] W. Cody, G. Meinardus, and R. Varga, “Chebyshev rational approximations to e^{-x} in $[0, +\infty)$ and applications to heat-conduction problems,” *Journal of Approximation Theory*, vol. 2, no. 1, pp. 50–65, 1969. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0021904569900306>
- [CO09] M. Caliarì and A. Ostermann, “Implementation of exponential Rosenbrock-type integrators,” *Applied Numerical Mathematics*, vol. 59, no. 3, pp. 568–581, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168927408000470>
- [Coc99] B. Cockburn, “Discontinuous Galerkin methods for convection-dominated problems,” in *High-Order Methods for Computational Physics*, T. J. Barth and H. Deconinck, Eds. Springer Berlin, 1999, pp. 69–224. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-03882-6_2
- [CP13] C. Clancy and J. Pudykiewicz, “On the use of exponential time integration methods in atmospheric models,” *Tellus A*, vol. 65, no. 0, pp. 1–16, 2013. [Online]. Available: <http://www.tellusa.net/index.php/tellusa/article/view/20898>
- [CS98a] B. Cockburn and C.-W. Shu, “The local discontinuous Galerkin method for time-dependent convection-diffusion systems,” *SIAM Journal on Numerical*

- Analysis*, vol. 35, no. 6, pp. 2440–2463, 1998. [Online]. Available: <http://dx.doi.org/10.1137/S0036142997316712>
- [CS98b] ———, “The Runge–Kutta discontinuous Galerkin method for conservation laws V,” *Journal of Computational Physics*, vol. 141, no. 2, pp. 199–224, 1998. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999198958922>
- [CSG66] S. Crist, P. M. Sherman, and D. R. Glass, “Study of the highly underexpanded sonic jet,” *AIAA Journal*, vol. 4, no. 1, pp. 68–71, 1966, doi: 10.2514/3.3386.
- [CVB04] M. Caliari, M. Vianello, and L. Bergamaschi, “Interpolating discrete advection-diffusion propagators at Leja sequences,” *Journal of Computational and Applied Mathematics*, vol. 172, no. 1, pp. 79–99, 2004. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0377042704000998>
- [CZ11] S. Chen and Y. Zhang, “Krylov implicit integration factor methods for spatial discretization on high dimensional unstructured meshes: Application to discontinuous Galerkin methods,” *Journal of Computational Physics*, vol. 230, no. 11, pp. 4336–4352, 2011, special issue High Order Methods for CFD Problems. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999111000283>
- [DCT98] A. Diefenbacher, M. Crone, and M. Türk, “Critical properties of CO₂, CHF₃, SF₆, (CO₂+ CHF₃), and (CHF₃+ SF₆),” *The Journal of Chemical Thermodynamics*, vol. 30, no. 4, pp. 481–496, 1998. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021961497903207>
- [DF03] V. Dolejší and M. Feistauer, “On the discontinuous Galerkin method for the numerical solution of compressible high-speed flow,” in *Numerical Mathematics and Advanced Applications: Proceedings of ENUMATH 2001 the 4th European Conference on Numerical Mathematics and Advanced Applications Ischia, July 2001*, F. Brezzi, A. Buffa, S. Corsaro, and A. Murli, Eds. Springer Milan, 2003, pp. 65–83. [Online]. Available: http://dx.doi.org/10.1007/978-88-470-2089-4_6
- [DFS03] V. Dolejší, M. Feistauer, and C. Schwab, “On some aspects of the discontinuous Galerkin finite element method for conservation laws,” *Mathematics and Computers in Simulation*, vol. 61, no. 3–6, pp. 333–346, 2003. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378475402000873>
- [DGKS76] J. W. Daniel, W. B. Gragg, L. Kaufman, and G. W. Stewart, “Reorthogonalization and stable algorithms for updating the Gram–Schmidt QR factorization,” *Mathematics of Computation*, vol. 30, no. 136, pp. 772–795, 1976. [Online]. Available: <http://www.jstor.org/stable/2005398>
- [DKBT11] P. Dubs, M. Khalij, R. Benelmir, and A. Tazibt, “Study on the dynamical characteristics of a supersonic high pressure ratio underexpanded impinging ideal gas jet through numerical simulations,” *Mechanics Research Communications*, vol. 38, no. 3, pp. 267–273, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0093641311000516>
- [DL16] M. Dumbser and R. Loubère, “A simple robust and accurate a posteriori sub-cell finite volume limiter for the discontinuous Galerkin method on unstructured meshes,” *Journal of Computational Physics*, vol. 319, no. 15, pp. 163–199, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999116301292>

- [DS71] C. Donaldson and R. Snedeker, “A study of free jet impingement. Part 1. Mean properties of free and impinging jets,” *Journal of Fluid Mechanics*, vol. 45, no. 2, pp. 281–319, 1971. [Online]. Available: <http://adsabs.harvard.edu/abs/1971JFM....45..281D>
- [DZLD14] M. Dumbser, O. Zanotti, R. Loubère, and S. Diot, “A posteriori subcell limiting of the discontinuous Galerkin finite element method for hyperbolic conservation laws,” *Journal of Computational Physics*, vol. 278, pp. 47–75, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999114005555>
- [EFL99] J. Edwards, R. Franklin, and M. Liou, “Low-diffusion flux-splitting methods for real fluid flows at all speeds,” *14th AIAA Computational Fluid Dynamics Conference*, vol. AIAA-99-3327, 1999.
- [EFL00] ———, “Low-diffusion flux-splitting methods for real fluid flows with phase transitions,” *AIAA Journal*, vol. 38, no. 9, pp. 1624–1633, 2000. [Online]. Available: <http://dx.doi.org/10.2514/2.1145>
- [Eme68] A. F. Emery, “An evaluation of several differencing methods for inviscid fluid flow problems,” *Journal of Computational Physics*, vol. 2, no. 3, pp. 306–331, 1968. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999168900600>
- [ETFS94] W. Edwards, L. Tuckerman, R. Friesner, and D. Sorensen, “Krylov methods for the incompressible Navier–Stokes equations,” *Journal of Computational Physics*, vol. 110, no. 1, pp. 82–102, 1994. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999184710072>
- [FFS03] M. Feistauer, J. Felcman, and I. Straškraba, *Mathematical and computational methods for compressible flow*. Oxford science publications, 2003.
- [FK07] M. Feistauer and V. Kučera, “On a robust discontinuous Galerkin technique for the solution of compressible flow,” *Journal of Computational Physics*, vol. 224, no. 1, pp. 208–221, 2007, special Issue Dedicated to Professor Piet Wesseling on the occasion of his retirement from Delft University of Technology. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999107000411>
- [FKP10] M. Feistauer, V. Kučera, and J. Prokopová, “Discontinuous Galerkin solution of compressible flow in time-dependent domains,” *Mathematics and Computers in Simulation*, vol. 80, no. 8, pp. 1612–1623, 2010, ESCO 2008 Conference. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S037847540900038X>
- [FP02] J. Ferziger and M. Perić, *Computational Methods for Fluid Dynamics*, third edition ed. Springer Berlin, 2002. [Online]. Available: <https://books.google.de/books?id=1D3EQgAACAAJ>
- [FPGB15] E. Franquet, V. Perrier, S. Gibout, and P. Bruel, “Free underexpanded jets in a quiescent medium: A review,” *Progress in Aerospace Sciences*, vol. 77, pp. 25–53, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0376042115000548>
- [FWV98] A. Feghhour, W. Wakeham, and V. Vesovic, “The viscosity of carbon dioxide,” *Journal of Physical and Chemical Reference Data*, vol. 27, no. 1, pp. 31–44, 1998. [Online]. Available: <http://scitation.aip.org/content/aip/journal/jpcrd/27/1/10.1063/1.556013>

- [GCP06] J. García-Cascales and H. Paillère, “Application of AUSM schemes to multi-dimensional compressible two-phase flow problems,” *Nuclear Engineering and Design*, vol. 236, no. 12, pp. 1225–1239, 2006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0029549305004280>
- [GGKS16] I. M. Gil’mutdinov, I. I. Gil’mutdinov, I. V. Kuznetsova, and A. N. Sabirzyanov, “Comminution of pharmaceutical substances by the adiabatic expansion of supercritical fluid solutions,” *Theoretical Foundations of Chemical Engineering*, vol. 50, no. 1, pp. 15–27, 2016. [Online]. Available: <http://dx.doi.org/10.1134/S0040579516010061>
- [GHS02] T. Gallouët, J.-M. Hérard, and N. Seguin, “Some recent finite volume schemes to compute Euler equations using real gas EOS,” *International Journal for Numerical Methods in Fluids*, vol. 39, no. 12, pp. 1073–1138, 2002. [Online]. Available: <http://dx.doi.org/10.1002/flf.346>
- [Gla88] P. Glaister, “An approximate linearised Riemann solver for the Euler equations for real gases,” *Journal of Computational Physics*, vol. 74, no. 2, pp. 382–408, 1988. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0021999188900848>
- [Gla89] —, “An efficient algorithm for compressible flows with real gases,” *International Journal for Numerical Methods in Fluids*, vol. 9, no. 10, pp. 1269–1283, 1989. [Online]. Available: <http://dx.doi.org/10.1002/flf.1650091007>
- [Gla90] —, “Flux-difference splitting for inviscid, real gases with non-equilibrium chemistry,” *Computers & Mathematics with Applications*, vol. 20, no. 3, pp. 45–51, 1990. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/089812219090029J>
- [GLR05] L. Giraud, J. Langou, and M. Rozloznik, “The loss of orthogonality in the Gram–Schmidt orthogonalization process,” *Computers & Mathematics with Applications*, vol. 50, no. 7, pp. 1069–1075, 2005. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0898122105003366>
- [God59] S. Godunov, “A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics,” *Mat. Sb. (N.S.)*, vol. 47(89), no. 3, pp. 271–306, 1959.
- [Got05] S. Gottlieb, “On high order strong stability preserving Runge–Kutta and multi step time discretizations,” *Journal of Scientific Computing*, vol. 25, no. 1, pp. 105–128, 2005. [Online]. Available: <http://dx.doi.org/10.1007/s10915-004-4635-5>
- [GP16] S. Gaudreault and J. A. Pudykiewicz, “An efficient exponential time integration method for the numerical solution of the shallow water equations on the sphere,” *Journal of Computational Physics*, vol. 322, pp. 827–848, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999116302911>
- [GPP11] J.-L. Guermond, R. Pasquetti, and B. Popov, “Entropy viscosity method for nonlinear conservation laws,” *Journal of Computational Physics*, vol. 230, no. 11, pp. 4248–4267, 2011, special issue High Order Methods for CFD Problems. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999110006583>
- [GS89] E. Gallopoulos and Y. Saad, “On the parallel solution of parabolic equations,” in *Proceedings of the 3rd International Conference on Supercomputing*, ser.

- ICS '89. New York, NY, USA: ACM, 1989, pp. 17–28. [Online]. Available: <http://doi.acm.org/10.1145/318789.318793>
- [GS92] —, “Efficient solution of parabolic equations by Krylov approximation methods,” *SIAM Journal on Scientific and Statistical Computing*, vol. 13, no. 5, pp. 1236–1264, 1992. [Online]. Available: <http://dx.doi.org/10.1137/0913071>
- [GV02] A. Guardone and L. Vigevano, “Roe linearization for the Van der Waals gas,” *Journal of Computational Physics*, vol. 175, no. 1, pp. 50–78, 2002. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999101969153>
- [GW89a] B. Grossman and R. Walters, “Analysis of flux-split algorithms for Euler’s equations with real gases,” *AIAA Journal*, vol. 27, no. 5, pp. 524–531, 1989.
- [GW89b] —, “Flux-split algorithms for the multi-dimensional Euler equations with real gases,” *Computers & Fluids*, vol. 17, no. 1, pp. 99–112, 1989. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0045793089900091>
- [Hab95] F. Habashi, *Computational Fluid Dynamics Techniques*. Taylor & Francis, 1995.
- [Har06] R. Hartmann, “Adaptive discontinuous Galerkin methods with shock-capturing for the compressible Navier–Stokes equations,” *International Journal for Numerical Methods in Fluids*, vol. 51, no. 9–10, pp. 1131–1156, 2006. [Online]. Available: <http://dx.doi.org/10.1002/fld.1134>
- [HBF10] J. Halama, F. Benkhaldoun, and J. Fořt, “Numerical modeling of two-phase transonic flow,” *Mathematics and Computers in Simulation*, vol. 80, no. 8, pp. 1624–1635, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378475409000421>
- [HBH⁺05] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley, “An overview of the Trilinos project,” *ACM Transactions on Mathematical Software*, vol. 31, no. 3, pp. 397–423, 2005. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1089021>
- [HCP12] A. Huerta, E. Casoni, and J. Peraire, “A simple shock-capturing technique for high-order discontinuous Galerkin methods,” *International Journal for Numerical Methods in Fluids*, vol. 69, no. 10, pp. 1614–1632, 8 2012. [Online]. Available: <http://dx.doi.org/10.1002/fld>
- [HDF⁺10] J. Halama, J. Dobeř, J. Fořt, J. Fürst, and K. Kozel, “Numerical modeling of unsteady flow in steam turbine stage,” *Journal of Computational and Applied Mathematics*, vol. 234, no. 7, pp. 2336–2341, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0377042709005718>
- [HF14] J. Halama and J. Fořt, “Homogeneous nucleation of steam in convergent-divergent nozzle,” *Engineering Mechanics*, vol. 21, no. 3, pp. 145–150, 2014. [Online]. Available: http://www.engineeringmechanics.cz/pdf/21_3_145.pdf
- [HH02] R. Hartmann and P. Houston, “Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations,” *Journal of Computational Physics*, vol. 183, no. 2, pp. 508–532, 2002. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999102972062>

- [HH06a] —, “Symmetric interior penalty DG methods for the compressible Navier–Stokes equations I: Method formulation,” *International Journal of Numerical Analysis and Modeling*, vol. 3, no. 1, pp. 1–20, 2006.
- [HH06b] —, “Symmetric interior penalty DG methods for the compressible Navier–Stokes equations II: Goal-oriented a posteriori error estimation,” *International Journal of Numerical Analysis and Modeling*, vol. 3, no. 2, pp. 141–162, 2006.
- [HHH⁺01] B. Helfgen, P. Hils, C. Holzknicht, M. Türk, and K. Schaber, “Simulation of particle formation during the rapid expansion of supercritical solutions,” *Journal of Aerosol Science*, vol. 32, no. 3, pp. 295–319, 2001. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S002185020000080X>
- [Hig09] N. J. Higham, “The scaling and squaring method for the matrix exponential revisited,” *SIAM Review*, vol. 51, no. 4, pp. 747–764, 2009. [Online]. Available: <http://dx.doi.org/10.1137/090768539>
- [HK07] H. Herwig and C. Kautz, *Technische Thermodynamik*, ser. Pearson Studium–Maschinenbau. Pearson Studium, 2007.
- [HL81] A. Harten and P. D. Lax, “A random choice finite difference scheme for hyperbolic conservation laws,” *SIAM Journal on Numerical Analysis*, vol. 18, no. 2, pp. 289–315, 1981. [Online]. Available: <http://dx.doi.org/10.1137/0718021>
- [HL97] M. Hochbruck and C. Lubich, “On Krylov subspace approximations to the matrix exponential operator,” *SIAM Journal on Numerical Analysis*, vol. 34, no. 5, pp. 1911–1925, 1997. [Online]. Available: <http://dx.doi.org/10.1137/S0036142995280572>
- [HLL83] A. Harten, P. D. Lax, and B. van Leer, “On upstream differencing and Godunov-type schemes for hyperbolic conservation laws,” *SIAM Review*, vol. 25, no. 1, pp. 35–61, 1983. [Online]. Available: <http://dx.doi.org/10.1137/1025002>
- [HLLM98] A. Harten, P. D. Lax, C. D. Levermore, and W. J. Morokoff, “Convex entropies and hyperbolicity for general Euler equations,” *SIAM Journal on Numerical Analysis*, vol. 35, no. 6, pp. 2117–2127, 1998. [Online]. Available: <http://dx.doi.org/10.1137/S0036142997316700>
- [HLR99] H. Holden, K.-A. Lie, and N. H. Risebro, “An unconditionally stable method for the Euler equations,” *Journal of Computational Physics*, vol. 150, no. 1, pp. 76–96, 1999. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999198961642>
- [HLS98] M. Hochbruck, C. Lubich, and H. Selhofer, “Exponential integrators for large systems of differential equations,” *SIAM Journal on Scientific Computing*, vol. 19, no. 5, pp. 1552–1574, 1998. [Online]. Available: <http://dx.doi.org/10.1137/S1064827595295337>
- [HO05] M. Hochbruck and A. Ostermann, “Exponential Runge–Kutta methods for parabolic problems,” *Applied Numerical Mathematics*, vol. 53, no. 2, pp. 323–339, 2005. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168927404001400>
- [HO10] —, “Exponential integrators,” *Acta Numerica*, vol. 19, pp. 209–286, 5 2010. [Online]. Available: http://journals.cambridge.org/article_S0962492910000048

- [HOS09] M. Hochbruck, A. Ostermann, and J. Schweitzer, “Exponential Rosenbrock-type methods,” *SIAM Journal on Numerical Analysis*, vol. 47, no. 1, pp. 786–803, 2009. [Online]. Available: <http://dx.doi.org/10.1137/080717717>
- [HRT07] V. Hernandez, J. Roman, and A. Tomas, “Parallel Arnoldi eigensolvers with enhanced scalability via global communications rearrangement,” *Parallel Computing*, vol. 33, no. 7–8, pp. 521–540, 2007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0167819107000622>
- [HTS00] B. Helfgen, M. Türk, and K. Schaber, “Theoretical and experimental investigations of the micronization of organic solids by rapid expansion of supercritical solutions,” *Powder Technology*, vol. 110, no. 1–2, pp. 22–28, 2000. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0032591099002648>
- [HTS03] —, “Hydrodynamic and aerosol modelling of the rapid expansion of supercritical solutions (RESS-process),” *The Journal of Supercritical Fluids*, vol. 26, no. 3, pp. 225–242, 2003. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0896844602001596>
- [Int16] Intel Corporation. (2016) Threading Building Blocks (Intel[®] TBB). [Online]. Available: <https://www.threadingbuildingblocks.org/>
- [JD96] J. Jones and R. Dugan, *Engineering Thermodynamics*. Prentice Hall, 1996.
- [JGT⁺04] M. Jugroot, C. P. T. Groth, B. A. Thomson, V. Baranov, and B. A. Collings, “Numerical investigation of interface region flows in mass spectrometers: neutral gas transport,” *Journal of Physics D: Applied Physics*, vol. 37, no. 8, pp. 1289–1300, 2004. [Online]. Available: <http://stacks.iop.org/0022-3727/37/i=8/a=019>
- [JJS95] J. Jaffre, C. Johnson, and A. Szepessy, “Convergence of the discontinuous Galerkin finite element method for hyperbolic conservation laws,” *Mathematical Models and Methods in Applied Sciences*, vol. 05, no. 03, pp. 367–386, 1995. [Online]. Available: <http://www.worldscientific.com/doi/abs/10.1142/S021820259500022X>
- [JP86] C. Johnson and J. Pitkäranta, “An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation,” *Mathematics of Computation*, vol. 46, no. 173, pp. 1–26, 1986. [Online]. Available: <http://dx.doi.org/10.2307/2008211>
- [KCG05] R. Kee, M. Coltrin, and P. Glarborg, *Chemically Reacting Flow: Theory and Practice*. John Wiley & Sons., 2005.
- [Ket08] D. I. Ketcheson, “Highly efficient strong stability-preserving Runge–Kutta methods with low-storage implementations,” *SIAM Journal on Scientific Computing*, vol. 30, no. 4, pp. 2113–2136, 2008. [Online]. Available: <http://dx.doi.org/10.1137/07070485X>
- [Kim03] N. Kim, “Remarks for the axisymmetric Navier–Stokes equations,” *Journal of Differential Equations*, vol. 187, no. 2, pp. 226–239, 2003. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0022039602000773>
- [KMG09] D. I. Ketcheson, C. B. Macdonald, and S. Gottlieb, “Optimal implicit strong stability preserving Runge–Kutta methods,” *Applied Numerical Mathematics*, vol. 59, no. 2, pp. 373–392, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168927408000688>

- [KTSG96] H. Ksibi, C. Tenaud, P. Subra, and Y. Garrabos, “Numerical simulation of rapid expansion of supercritical fluid,” *European Journal of Mechanics - B/Fluids*, vol. 15, no. 4, pp. 569–596, 1996.
- [KWD07] E. J. Kubatko, J. J. Westerink, and C. Dawson, “Semi discrete discontinuous Galerkin methods and stage-exceeding-order, strong-stability-preserving Runge–Kutta time discretizations,” *Journal of Computational Physics*, vol. 222, no. 2, pp. 832–848, 2007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999106003974>
- [KXR⁺04] L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeron, and J. Flaherty, “Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws,” *Applied Numerical Mathematics*, vol. 48, no. 3–4, pp. 323–338, 2004. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0168927403001831>
- [L95] R. Löhner, “Mesh adaptation in fluid mechanics,” *Engineering Fracture Mechanics*, vol. 50, no. 5, pp. 819–847, 1995. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0013794494E0062L>
- [LADQ14] J. Liu, G. Amberg, and M. Do-Quang, “Numerical simulation of particle formation in the rapid expansion of supercritical solution process,” *The Journal of Supercritical Fluids*, vol. 95, pp. 572–587, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0896844614002861>
- [Lax57] P. D. Lax, “Hyperbolic systems of conservation laws II,” *Communications on Pure and Applied Mathematics*, vol. 10, no. 4, pp. 537–566, 1957. [Online]. Available: <http://dx.doi.org/10.1002/cpa.3160100406>
- [LC64] C. H. Lewis and D. J. Carlson, “Normal shock location in underexpanded gas and gas-particle jets,” *AIAA Journal*, vol. 2, no. 4, pp. 776–777, 1964, doi: 10.2514/3.2409.
- [LDD14] R. Loubère, M. Dumbser, and S. Diot, “A new family of high order unstructured MOOD and ADER finite volume schemes for multidimensional systems of hyperbolic conservation laws,” *Communications in Computational Physics*, vol. 16, pp. 718–763, 9 2014. [Online]. Available: http://journals.cambridge.org/article_S1815240600005739
- [LDQA15] J. Liu, M. Do-Quang, and G. Amberg, “Numerical simulation of rapid expansion of supercritical carbon dioxide,” *AIChE Journal*, vol. 61, no. 1, pp. 317–332, 2015. [Online]. Available: <http://dx.doi.org/10.1002/aic.14603>
- [L82] B. van Leer, “Flux-vector splitting for the euler equation,” in *Eighth International Conference on Numerical Methods in Fluid Dynamics: Proceedings of the Conference, Rheinisch-Westfälische Technische Hochschule Aachen, Germany, June 28 – July 2, 1982*, E. Krause, Ed. Springer Berlin, 1982, pp. 507–512. [Online]. Available: http://dx.doi.org/10.1007/3-540-11948-5_66
- [LTR87] B. van Leer, J. Thomas, and P. Roe, “A comparison of numerical flux formulas for the Euler and Navier–Stokes equations,” *American Institute of Aeronautics and Astronautics*, pp. 36–41, 1987.
- [LeV02] R. J. LeVeque, *Finite volume methods for hyperbolic problems*. Cambridge University Press, 2002.

- [Lio96] M. Liou, “A sequel to AUSM: AUSM+,” *Journal of Computational Physics*, vol. 129, no. 2, pp. 364–382, 1996. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999196902569>
- [Lio01] —, “Ten years in the making: AUSM-family,” National Aeronautics and Space Administration, Tech. Rep., 2001. [Online]. Available: <http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20020009014.pdf>
- [Lio06] —, “A sequel to AUSM, part II: AUSM+-up for all speeds,” *Journal of Computational Physics*, vol. 214, no. 1, pp. 137–170, 2006. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999105004274>
- [Lio10] —, “Evolution of advection upstream splitting method schemes,” *Defence Science Journal*, vol. 60, no. 6, pp. 606–613, 2010.
- [LLS90] M. Liou, B. van Leer, and J. Shuen, “Splitting of inviscid fluxes for real gases,” *Journal of Computational Physics*, vol. 87, no. 1, pp. 1–24, 1990. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/002199919090222M>
- [LLK⁺13] Q. Lv, X. P. Long, Y. Kang, L. Z. Xiao, and W. Wu, “Numerical investigation on the expansion of supercritical carbon dioxide jet,” *IOP Conference Series: Materials Science and Engineering*, vol. 52, no. 7, pp. 1–6, 2013. [Online]. Available: <http://stacks.iop.org/1757-899X/52/i=7/a=072011>
- [LQ10a] G. Li and J. Qiu, “Hybrid weighted essentially non-oscillatory schemes with different indicators,” *Journal of Computational Physics*, vol. 229, no. 21, pp. 8105–8129, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999110003955>
- [LQ10b] —, “Hybrid weighted essentially non-oscillatory schemes with different indicators,” *Journal of Computational Physics*, vol. 229, no. 21, pp. 8105–8129, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999110003955>
- [LS93] M. Liou and C. Steffen, “A new flux splitting scheme,” *Journal of Computational Physics*, vol. 107, no. 1, pp. 23–39, 1993. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999183711228>
- [LT13] E. Laporte and P. Le Tallec, *Numerical Methods in Sensitivity Analysis and Shape Optimization*. Birkhäuser Basel, 2013. [Online]. Available: <http://www.springer.com/us/book/9780817643225>
- [LW60] P. Lax and B. Wendroff, “Systems of conservation laws,” *Communications on Pure and Applied Mathematics*, vol. 13, no. 2, pp. 217–237, 1960. [Online]. Available: <http://dx.doi.org/10.1002/cpa.3160130205>
- [LZG11] C. Li, X. Zhu, and C. Gu, “Matrix padé-type method for computing the matrix exponential,” *Applied Mathematics*, vol. 2, no. 2, pp. 247–253, 2011. [Online]. Available: <http://www.scirp.org/journal/PaperInformation.aspx?PaperID=3974>
- [MBCV09] A. Martínez, L. Bergamaschi, M. Caliari, and M. Vianello, “A massively parallel exponential integrator for advection-diffusion models,” *Journal of Computational and Applied Mathematics*, vol. 231, no. 1, pp. 82–91, 2009. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0377042709000429>

- [MBM13] S. Martynov, S. Brown, and H. Mahgerefteh, “An extended Peng–Robinson equation of state for carbon dioxide solid-vapor equilibrium,” *Greenhouse Gases: Science and Technology*, vol. 3, no. 2, pp. 136–147, 2013. [Online]. Available: <http://dx.doi.org/10.1002/ghg.1322>
- [MEKS03] D. Mao, J. R. Edwards, A. V. Kuznetsov, and R. K. Srivastava, “Development of low-diffusion flux-splitting methods for dense gas-solid flows,” *Journal of Computational Physics*, vol. 185, no. 1, pp. 100–119, 2003. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999102000499>
- [ML03] C. Moler and C. Van Loan, “Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later,” *SIAM Review*, vol. 45, no. 1, pp. 3–49, 2003. [Online]. Available: <http://dx.doi.org/10.1137/S00361445024180>
- [Mon13] M. Montagnac, “Variable normalization (nondimensionalization and scaling) for Navier–Stokes equations: a practical guide,” Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique, Tech. Rep. TRF-CFD-13-77, October 2013.
- [MKB08] A. Ben Moussa, H. Ksibi, and M. Baccar, “Simulation of particles transport and coagulation during the RESS process,” *The European Physical Journal–Applied Physics*, vol. 43, pp. 253–261, 2008. [Online]. Available: http://www.epjap.org/action/article_S1286004208001171
- [MKTB05] A. Ben Moussa, H. Ksibi, C. Tenaud, and M. Baccar, “Paramètres géométriques de contrôle de la détente d’un fluide supercritique,” *International Journal of Thermal Sciences*, vol. 44, no. 8, pp. 774–786, 2005. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1290072905000396>
- [MSC11] S. Moon, J. Shin, and J. Choi, “Speed of sound and real gas solution algorithm for supercritical flows,” *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, vol. 7, pp. 6305–6317, 2011. [Online]. Available: <http://dx.doi.org/10.2514/6.2011-471>
- [MY89] J. Montagné and H. Yee, “Comparative study of high-resolution shock-capturing schemes for a real gas,” *AIAA Journal*, vol. 27, no. 10, pp. 1332–1346, 1989.
- [NR50] J. von Neumann and R. D. Richtmyer, “A method for the numerical calculation of hydrodynamic shocks,” *Journal of Applied Physics*, vol. 21, no. 3, pp. 232–237, 1950. [Online]. Available: <http://scitation.aip.org/content/aip/journal/jap/21/3/10.1063/1.1699639>
- [Ni82] R.-H. Ni, “A multiple grid scheme for solving the Euler equations.” *AIAA Journal*, vol. 20, no. 11, pp. 1565–1571, 1982. [Online]. Available: <http://arc.aiaa.org/doi/pdf/10.2514/3.51220>
- [NW83] A. Nauts and R. E. Wyatt, “New approach to many–state quantum dynamics: The recursive–residue–generation method,” *Phys. Rev. Lett.*, vol. 51, pp. 2238–2241, Dec 1983. [Online]. Available: <http://link.aps.org/doi/10.1103/PhysRevLett.51.2238>
- [OKM⁺08] Y. Otobe, H. Kashimura, S. Matsuo, T. Setoguchi, and H. Kim, “Influence of nozzle geometry on the near-field structure of a highly underexpanded sonic jet,” *Journal of Fluids and Structures*, vol. 24, no. 2, pp. 281–293, 2008. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0889974607000618>

- [OS82] S. Osher and F. Solomon, “Upwind difference schemes for hyperbolic systems of conservation laws,” *Math. Comp.*, vol. 38, no. 158, pp. 339–374, 1982.
- [OY99] H. S. Oh and W. S. Yang, “Comparison of matrix exponential methods for fuel burnup calculations,” *Journal of the Korean Nuclear Society*, vol. 31, no. 2, pp. 172–181, 1999. [Online]. Available: <https://inis.iaea.org/search/searchsinglerecord.aspx?recordsFor=SingleRecord&RN=30059325>
- [Pet91] T. E. Peterson, “A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation,” *SIAM Journal on Numerical Analysis*, vol. 28, no. 1, pp. 133–140, 1991. [Online]. Available: <http://www.jstor.org/stable/2157936>
- [PE11] D. Di Pietro and A. Ern, *Mathematical Aspects of Discontinuous Galerkin Methods*, ser. Mathématiques et Applications. Springer Berlin, 2011. [Online]. Available: <http://www.springer.com/us/book/9783642229794>
- [PL86] T. J. Park and J. C. Light, “Unitary quantum time evolution by iterative Lanczos reduction,” *The Journal of Chemical Physics*, vol. 85, no. 10, pp. 5870–5876, 1986. [Online]. Available: <http://aip.scitation.org/doi/abs/10.1063/1.451548>
- [PM89] B. Platzler and G. Maurer, “A generalized equation of state for pure polar and nonpolar fluids,” *Fluid Phase Equilibria*, vol. 51, pp. 223–236, 1989. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0378381289803668>
- [PN98] C. Piao and M. Noguchi, “An international standard equation of state for the thermodynamic properties of HFC-125 (Pentafluoroethane),” *Journal of Physical and Chemical Reference Data*, vol. 27, no. 4, pp. 775–806, 1998. [Online]. Available: <http://scitation.aip.org/content/aip/journal/jpcrd/27/4/10.1063/1.556021>
- [PR76] D.-Y. Peng and D. B. Robinson, “A new two-constant equation of state,” *Industrial & Engineering Chemistry Fundamentals*, vol. 15, no. 1, pp. 59–64, 1976. [Online]. Available: <http://dx.doi.org/10.1021/i160057a011>
- [PRC12] R. Pecnik, E. Rinaldi, and P. Colonna, “Computational fluid dynamics of a radial compressor operating with supercritical CO₂,” *Journal of Engineering for Gas Turbines and Power*, vol. 134, no. 12, pp. 122 301–1–122 301–8, 2012. [Online]. Available: <http://gasturbinespower.asmedigitalcollection.asme.org/article.aspx?articleid=1660720>
- [PRO08] PROPATH Group, “Propath: A program package for thermophysical properties, version 13.1,” <http://www.mech.kyushu-u.ac.jp/heat/propath/>, 2008. [Online]. Available: <http://www.mech.kyushu-u.ac.jp/~heat/propath/>
- [PSPR15] M. Pini, A. Spinelli, G. Persico, and S. Rebay, “Consistent look-up table interpolation method for real-gas flow simulations,” *Computers & Fluids*, vol. 107, pp. 178–188, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S004579301400423X>
- [Pus11] M. Pusa, “Rational approximations to the matrix exponential in burnup calculations,” *Nuclear Science and Engineering*, vol. 169, no. 2, 2011. [Online]. Available: http://www.ans.org/pubs/journals/nse/a_12520
- [PVMZ87] J. Peraire, M. Vahdati, K. Morgan, and O. Zienkiewicz, “Adaptive remeshing for compressible flow computations,” *Journal of Computational*

- Physics*, vol. 72, no. 2, pp. 449–466, 1987. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0021999187900933>
- [QCGQ03] L. Quartapelle, L. Castelletti, A. Guardone, and G. Quaranta, “Solution of the Riemann problem of classical gasdynamics,” *Journal of Computational Physics*, vol. 190, no. 1, pp. 118–140, 2003. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999103002675>
- [QS05a] J. Qiu and C.-W. Shu, “A comparison of troubled-cell indicators for Runge–Kutta discontinuous Galerkin methods using weighted essentially nonoscillatory limiters,” *SIAM Journal on Scientific Computing*, vol. 27, no. 3, pp. 995–1013, 2005. [Online]. Available: <http://dx.doi.org/10.1137/04061372X>
- [QS05b] —, “Runge–Kutta discontinuous Galerkin method using WENO limiters,” *SIAM Journal on Scientific Computing*, vol. 26, no. 3, pp. 907–929, 2005. [Online]. Available: <http://dx.doi.org/10.1137/S1064827503425298>
- [RCG13] D. Robbins, R. Cant, and L. Gladden, “Development of accurate, robust liquid equations of state for multi-phase CFD simulations with a modified AUSM+-up scheme,” *Computers & Fluids*, vol. 77, pp. 166–180, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045793013000613>
- [Ric88] G. R. Richter, “An optimal-order error estimate for the discontinuous Galerkin method,” *Mathematics of Computation*, vol. 50, no. 181, pp. 75–88, 1988. [Online]. Available: <http://www.jstor.org/stable/2007915>
- [Ric08] —, “On the order of convergence of the discontinuous Galerkin method for hyperbolic equations,” *Mathematics of Computation*, vol. 77, no. 264, pp. 1871–1885, 2008. [Online]. Available: <http://www.jstor.org/stable/40234594>
- [Roe81] P. Roe, “Approximate Riemann solvers, parameter vectors, and difference schemes,” *Journal of Computational Physics*, vol. 43, no. 2, pp. 357–372, 1981. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0021999181901285>
- [Roh01] A. Rohde, “Eigenvalues and Eigenvectors of the Euler equations in general geometries,” *15th AIAA Computational Fluid Dynamics Conference*, 2001. [Online]. Available: <http://dx.doi.org/10.2514/6.2001-2609>
- [Ruu06] S. J. Ruuth, “Global optimization of explicit strong-stability-preserving Runge–Kutta methods,” *Mathematics of Computation*, vol. 75, pp. 183–207, 2006. [Online]. Available: <http://www.ams.org/journals/mcom/2006-75-253/S0025-5718-05-01772-2/S0025-5718-05-01772-2.pdf>
- [RV81] A. Rizzi and H. Viviand, *Numerical Methods for the Computation of Inviscid Transonic Flows with Shock Waves: A GAMM Workshop*, ser. Notes on Numerical Fluid Mechanics, GAMM, Ed. Informatica International, 1981. [Online]. Available: <http://www.springer.com/us/book/9783528080778>
- [Saa92] Y. Saad, “Analysis of some Krylov subspace approximations to the matrix exponential operator,” *SIAM Journal on Numerical Analysis*, vol. 29, no. 1, pp. 209–228, 1992. [Online]. Available: <http://dx.doi.org/10.1137/0729014>
- [SB10] G. Sutton and O. Biblarz, *Rocket Propulsion Elements*, 8th ed. John Wiley & Sons, 2010.
- [Sid98] R. B. Sidje, “Expokit: A software package for computing matrix exponentials,” *ACM Trans. Math. Softw.*, vol. 24, no. 1, pp. 130–156, 1998. [Online]. Available: <http://doi.acm.org/10.1145/285861.285868>

- [SLL90] J. Shuen, M. Liou, and B. V. Leer, “Inviscid flux-splitting algorithms for real gases with non-equilibrium chemistry,” *Journal of Computational Physics*, vol. 90, no. 2, pp. 371–395, 1990. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/002199919090172W>
- [SLL94] R. Saurel, M. Larini, and J. C. Loraud, “Exact and approximate Riemann solvers for real gases,” *Journal of Computational Physics*, vol. 112, no. 1, pp. 126–137, 1994. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999184710862>
- [Sod78] G. Sod, “A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws,” *Journal of Computational Physics*, vol. 27, no. 1, pp. 1–31, 1978. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0021999178900232>
- [SR02] R. J. Spiteri and S. J. Ruuth, “A new class of optimal high-order strong-stability-preserving time discretization methods,” *SIAM Journal on Numerical Analysis*, vol. 40, no. 2, pp. 469–491, 2002. [Online]. Available: <http://dx.doi.org/10.1137/S0036142901389025>
- [SR03] —, “Non-linear evolution using optimal fourth-order strong-stability-preserving Runge–Kutta methods,” *Mathematics and Computers in Simulation*, vol. 62, no. 1–2, pp. 125–135, 2003, Nonlinear Waves: Computation and Theory II. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378475402001799>
- [SS91] Y. Saad and D. Semeraro, “Application of Krylov exponential propagation to fluid dynamics equations,” Research Institute for Advanced Computer Science. National Aeronautics and Space Administration (NASA), Tech. Rep. NASA-CR-188887, NAS 1.26:188887, RIACS-TR-91-06, 1991. [Online]. Available: <https://ntrs.nasa.gov/search.jsp?R=19920002072>
- [SSS09] J. C. Schulze, P. J. Schmid, and J. L. Sesterhenn, “Exponential time integration using Krylov subspaces,” *International Journal for Numerical Methods in Fluids*, vol. 60, no. 6, pp. 591–609, 2009. [Online]. Available: <http://dx.doi.org/10.1002/fd.1902>
- [SW81] J. L. Steger and R. Warming, “Flux vector splitting of the inviscid gasdynamic equations with application to finite-difference methods,” *Journal of Computational Physics*, vol. 40, no. 2, pp. 263–293, 1981. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0021999181902102>
- [SW96] R. Span and W. Wagner, “A new equation of state for carbon dioxide covering the fluid region from the triple-point temperature to 1100 K at pressures up to 800 MPa,” *Journal of Physical and Chemical Reference Data*, vol. 25, no. 6, pp. 1509–1596, 1996. [Online]. Available: <http://scitation.aip.org/content/aip/journal/jpcrd/25/6/10.1063/1.555991>
- [THH⁺02] M. Türk, P. Hils, B. Helfgen, K. Schaber, H. Martin, and M. Wahl, “Micronization of pharmaceutical substances by the rapid expansion of supercritical solutions (RESS): a promising method to improve bioavailability of poorly soluble pharmaceutical agents,” *The Journal of Supercritical Fluids*, vol. 22, no. 1, pp. 75–84, 2002. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0896844601001097>
- [TL10] M. Tokman and J. Loffeld, “Efficient design of exponential-Krylov integrators for large scale computing,” *Procedia Computer Science*, vol. 1, no. 1, pp.

- 229–237, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S187705091000027X>
- [Tor99] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer Berlin, 1999, pp. 265–291. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-03915-1_8
- [Tür99] M. Türk, “Formation of small organic particles by RESS: experimental and theoretical investigations,” *The Journal of Supercritical Fluids*, vol. 15, no. 1, pp. 79–89, 1999. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0896844698001314>
- [Tür00] —, “Influence of thermodynamic behaviour and solute properties on homogeneous nucleation in supercritical solutions,” *The Journal of Supercritical Fluids*, vol. 18, no. 3, pp. 169–184, 2000. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0896844600000802>
- [Vij86] G. Vijayasundaram, “Transonic flow simulations using an upstream centered scheme of Godunov in finite elements,” *Journal of Computational Physics*, vol. 63, no. 2, pp. 416–433, 1986. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0021999186902020>
- [VM90] M. Vinokur and J. Montagné, “Generalized flux-vector splitting and Roe average for an equilibrium real gas,” *Journal of Computational Physics*, vol. 89, no. 2, pp. 276–300, 1990. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/002199919090145Q>
- [V04] F. De Vuyst, “Stable and accurate hybrid finite volume methods based on pure convexity arguments for hyperbolic systems of conservation law,” *Journal of Computational Physics*, vol. 193, no. 2, pp. 426–468, 2004. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999103003747>
- [VWO⁺90] V. Vesovic, W. Wakeham, G. Olchoway, J. Sengers, J. Watson, and J. Millat, “The transport properties of carbon dioxide,” *Journal of Physical and Chemical Reference Data*, vol. 19, no. 3, pp. 763–808, 1990. [Online]. Available: <http://scitation.aip.org/content/aip/journal/jpcrd/19/3/10.1063/1.555875>
- [WC81] P. Woodward, P. and P. Colella, *High resolution difference schemes for compressible gas dynamics*. Springer Berlin, 1981, pp. 434–441. [Online]. Available: http://dx.doi.org/10.1007/3-540-10694-4_67
- [WC84] P. Woodward and P. Colella, “The numerical simulation of two-dimensional fluid flow with strong shocks,” *Journal of Computational Physics*, vol. 54, no. 1, pp. 115–173, 1984. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0021999184901426>
- [WDE⁺99] H. Wang, H. K. Dahle., R. E. Ewing, M. S. Espedal, R. C. Sharpley, and S. Man, “An ELLAM scheme for advection-diffusion equations in two dimensions,” *SIAM Journal on Scientific Computing*, vol. 20, no. 6, pp. 2160–2194, 1999. [Online]. Available: <http://dx.doi.org/10.1137/S1064827596309396>
- [Wes01] P. Wesseling, *Principles of Computational Fluid Dynamics*, ser. Lecture Notes in Computer Science. Springer, 2001. [Online]. Available: <http://link.springer.com/book/10.1007%2F978-3-642-05146-3>
- [Wil06] *Fluorescence Imaging of Underexpanded Jets and Comparison with CFD*, ser. Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics, 2006, doi:10.2514/6.2006-910.

- [WWFF13] C. J. Wareing, R. M. Woolley, M. Fairweather, and S. A. E. G. Falle, “A composite equation of state for the modeling of sonic carbon dioxide jets in carbon capture and storage scenarios,” *AIChE Journal*, vol. 59, no. 10, pp. 3928–3942, 2013. [Online]. Available: <http://dx.doi.org/10.1002/aic.14102>
- [YF15] S. Yamamoto and T. Furusawa, “Thermophysical flow simulations of rapid expansion of supercritical solutions (RESS),” *The Journal of Supercritical Fluids*, vol. 97, pp. 192–201, 2015. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0896844614003908>
- [YFM11] S. Yamamoto, T. Furusawa, and R. Matsuzawa, “Numerical simulation of supercritical carbon dioxide flows across critical point,” *International Journal of Heat and Mass Transfer*, vol. 54, no. 4, pp. 774–782, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0017931010006083>
- [YMF11] S. Yamamoto, R. Matsuzawa, and T. Furusawa, “Simulation of thermophysical flow in axisymmetric nozzle with expansion chamber,” *AIChE Journal*, vol. 57, no. 10, pp. 2629–2635, 2011. [Online]. Available: <http://dx.doi.org/10.1002/aic.12482>
- [Zal79] S. T. Zalesak, “Fully multidimensional flux-corrected transport algorithms for fluids,” *Journal of Computational Physics*, vol. 31, no. 3, pp. 335–362, 1979. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0021999179900512>
- [ZGMP13] V. Zingan, J.-L. Guermond, J. Morel, and B. Popov, “Implementation of the entropy viscosity method with the discontinuous Galerkin method,” *Computer Methods in Applied Mechanics and Engineering*, vol. 253, pp. 479–490, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0045782512002708>
- [ZQSD08] J. Zhu, J. Qiu, C.-W. Shu, and M. Dumbser, “Runge–Kutta discontinuous Galerkin method using WENO limiters II: Unstructured meshes,” *Journal of Computational Physics*, vol. 227, no. 9, pp. 4330–4353, 2008. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0021999108000120>

Appendix

A. Specific internal energy equations

In the present work the thermodynamic behavior of CO₂ is modeled by the *egB-EOS* described in Sec. 2.4.3. This equation is used to compute the caloric properties of the substance, as was explained in Sec. 2.5. In this way, the expression for the specific internal energy can be analytically derived from Eq. 2.107 to get

$$\begin{aligned}
e(\delta, \zeta) = & \left(-547973.8422 \delta^2 e^{-\delta^2} + 30172.1147 e^{-\delta^2} - 30172.1147 \right) \zeta^4 \\
& + \left(737799.3688 \delta^2 e^{-\delta^2} - 172793.3145 e^{-\delta^2} + 22383.9652 \delta + 172793.3145 \right) \zeta^3 \\
& - \left(220989.4477 \delta^2 e^{-\delta^2} - 123888.6843 e^{-\delta^2} - 177296.0833 \delta + 123888.6843 \right) \zeta^2 \\
& + (31547.0112 \delta^2 + 138505.8822 \delta) \zeta \\
& + \frac{93484.3862}{\zeta} + \frac{65741.7898}{\zeta^2} - \frac{8333.6339}{\zeta^3} + \frac{440.4745}{\zeta^4} \\
& + 1696.6026 \delta^5 - 9114.4995 \delta^4 + 4223.5262 \delta^3 + 7706.7840 \delta^2 \\
& - 158472.6882 \delta - 203712.1958,
\end{aligned} \tag{6.1}$$

where $\delta = \rho/\rho_c$ and $\zeta = \theta_c/\theta$. Likewise, the partial derivatives of e were calculated and we obtained

$$\begin{aligned}
\frac{\partial e}{\partial \zeta}(\delta, \zeta) = & \left(-2191895.3688 \delta^2 e^{-\delta^2} + 120688.4590 e^{-\delta^2} - 120688.4590 \right) \zeta^3 \\
& + \left(2213398.1064 \delta^2 e^{-\delta^2} - 518379.9436 e^{-\delta^2} + 67151.8956 \delta + 518379.9436 \right) \zeta^2 \\
& - \left(441978.8954 \delta^2 e^{-\delta^2} - 247777.3686 e^{-\delta^2} + 354592.1666 \delta + 247777.3686 \right) \zeta \\
& - \frac{93484.3862}{\zeta^2} - \frac{131483.5796}{\zeta^3} + \frac{25000.9016}{\zeta^4} - \frac{1761.898011}{\zeta^5} \\
& + 31547.0112 \delta^2 + 138505.8822 \delta
\end{aligned} \tag{6.2}$$

and

$$\begin{aligned}
\frac{\partial e}{\partial \delta}(\delta, \zeta) = & e^{-\delta^2} \left((1095947.6844 \delta^3 - 1156291.9139 \delta) \zeta^4 \right. \\
& \left. - (1475598.7376 \delta^3 - 1821185.3666 \delta) \zeta^3 + (441978.8954 \delta^3 - 689756.264 \delta) \zeta^2 \right) \\
& + 22383.9652 \zeta^3 - 177296.0833 \zeta^2 + 63094.0223 \delta \zeta + 138505.8822 \zeta \\
& + 8483.01321 \delta^4 - 36457.9978 \delta^3 + 12670.5786 \delta^2 + 15413.56808 \delta \\
& - 158472.6882.
\end{aligned} \tag{6.3}$$

B. Analytic expressions derived from egB-EOS

Computing the ϵ -coefficients in accordance with [Ben71], the equation of state Eq. 2.99 can be written as

$$\begin{aligned}
p(\delta, \zeta) = & e^{-\delta^2} \left((102493027.4439 \delta^5 - 108136419.7850 \delta^3) \zeta^4 \right. \\
& - (172497492.4217 \delta^5 - 212896569.3612 \delta^3) \zeta^3 \\
& \left. + (68889777.1613 \delta^5 - 107510009.6770 \delta^3) \zeta^2 \right) \\
& - (2616685.5303 \zeta^3 + 27634549.5184 \zeta^2 - 32382675.2499 \zeta) \delta^2 \\
& + 14751382.4240 \delta^3 \zeta + 3966656.9764 \delta^6 - 17047759.7847 \delta^5 \\
& + 5924762.5441 \delta^4 + 7207384.4347 \delta^3 - 7.4101828.9921 \delta^2 \\
& + \frac{1}{\zeta} \left(3717335.5572 \delta^5 + 4356125.1356 \delta^4 - 6884640.0889 \delta^3 \right. \\
& \left. + 32557847.9672 \delta^2 + 26866961.6892 \delta \right), \tag{6.4}
\end{aligned}$$

whereas its partial derivatives $\frac{\partial p}{\partial \delta}$ and $\frac{\partial p}{\partial \zeta}$ are given by

$$\begin{aligned}
\frac{\partial p}{\partial \delta}(\delta, \zeta) = & e^{-\delta^2} \left(- (204986054.8878 \delta^6 - 728737976.7894 \delta^4 + 324409259.3550 \delta^2) \zeta^4 \right. \\
& + (344994984.8434 \delta^6 - 1288280600.8309 \delta^4 + 638689708.0835 \delta^2) \zeta^3 \\
& \left. - (137779554.3226 \delta^6 - 559468905.1606 \delta^4 + 322530029.0310 \delta^2) \zeta^2 \right) \\
& + (5233371.0607 \zeta^3 - 55269099.0368 \zeta^2 + 64765350.4999 \zeta) \delta \\
& + 44254147.2720 \delta^2 \zeta + 23799941.8585 \delta^5 - 85238798.9237 \delta^4 \\
& + 23699050.1765 \delta^3 + 21622153.304 \delta^2 - 148203657.9841 \delta \\
& + \frac{1}{\zeta} \left(18586677.7859 \delta^4 + 17424500.5425 \delta^3 - 20653920.2666 \delta^2 \right. \\
& \left. + 65115695.9345 \delta + 26866961.6892 \right) \tag{6.5}
\end{aligned}$$

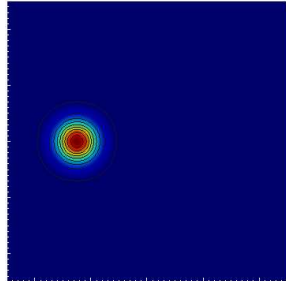
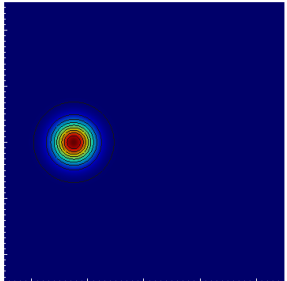
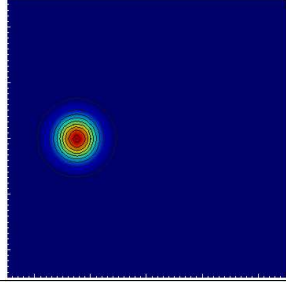
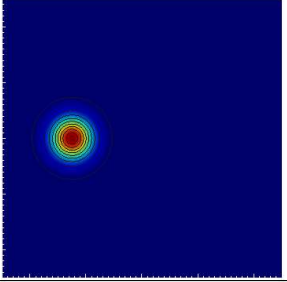
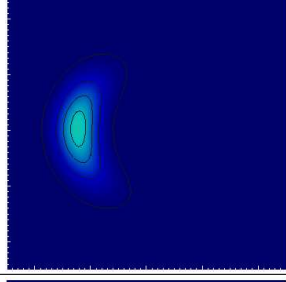
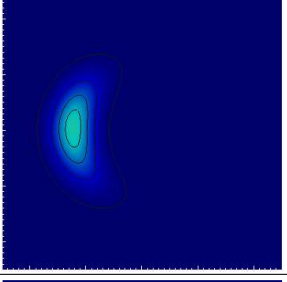
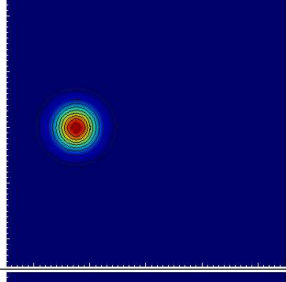
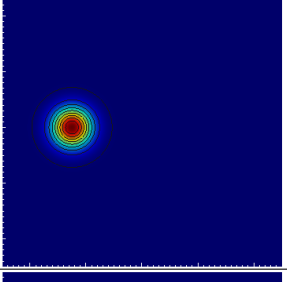
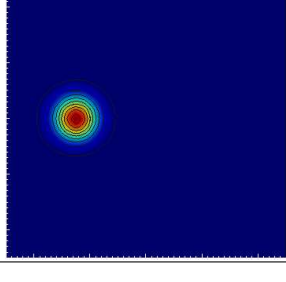
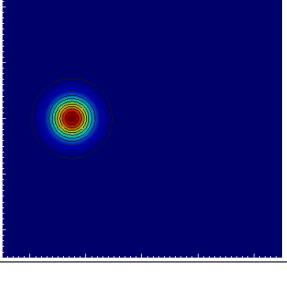
and

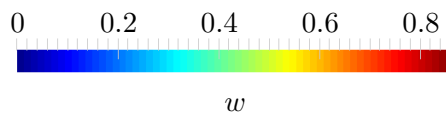
$$\begin{aligned}
\frac{\partial p}{\partial \zeta}(\delta, \zeta) = & e^{-\delta^2} \left((409972109.7755 \delta^5 - 432545679.1400 \delta^3) \zeta^3 \right. \\
& - (517492477.2651 \delta^5 - 638689708.0835 \delta^3) \zeta^2 \\
& \left. + (137779554.3226 \delta^5 - 2.1502001935401726 \cdot 10^8 \delta^3) \zeta \right) \\
& + (7850056.5910 \zeta^2 - 55269099.0368 \zeta + 32382675.2499) \delta^2 \\
& + 14751382.4240 \delta^3 - \frac{1}{\zeta^2} \left(3717335.5572 \delta^5 - 4356125.1356 \delta^4 \right. \\
& \left. + 6884640.0889 \delta^3 - 32557847.9672 \delta^2 - 26866961.6892 \delta \right). \tag{6.6}
\end{aligned}$$

C. Two dimensional rotating Gaussian pulse

In Sec. 5.3.1 the results achieved for the rotating Gaussian pulse, using \mathbb{P}^1 and \mathbb{P}^3 DG elements for the spatial discretization, were presented. Here, in order to completed the set of these results, the solution found with \mathbb{P}^2 DG -elements is shown.

Figure C.1.: Contours of w for the rotating Gaussian-pulse problem at $t = T$ using \mathbb{P}^2 DG -elements. Coarse mesh conformed by 20×20 elements and fine mesh formed by 160×160 elements.

	Δt	Coarse mesh	$\frac{w_{max}}{w_{min}}$	Fine mesh	$\frac{w_{max}}{w_{min}}$
Analytic	—		$\frac{0.8642}{(0.0)}$		$\frac{0.8642}{(0.0)}$
Backward Euler	$\frac{T}{20000}$		$\frac{0.8210}{(-0.0001)}$		$\frac{0.8419}{(0.0)}$
	$\frac{T}{200}$		$\frac{0.3459}{(0.0)}$		$\frac{0.3459}{(0.0)}$
Rosenbrock-Euler	$\frac{T}{2000}$		$\frac{0.8400}{(-0.0002)}$		$\frac{0.8641}{(-0.0002)}$
	T		$\frac{0.8400}{(-0.0002)}$		$\frac{0.8641}{(0.0)}$



D. Solution using an uniform grid with \mathbb{P}^1 DG-elements

In Sec. 5.3.4 the supersonic forward-facing step problem was introduced and a solution employing a adaptive grid was presented. Here, a similar set of results, computed using a uniform mesh with $h \approx 1/150$ (i.e. 46254 \mathbb{P}^1 cells and 740064 dofs), is shown.

Figure D.2.: Density contours at different times for the forward-facing step problem using an uniform grid with 46254 \mathbb{P}^1 DG-elements.

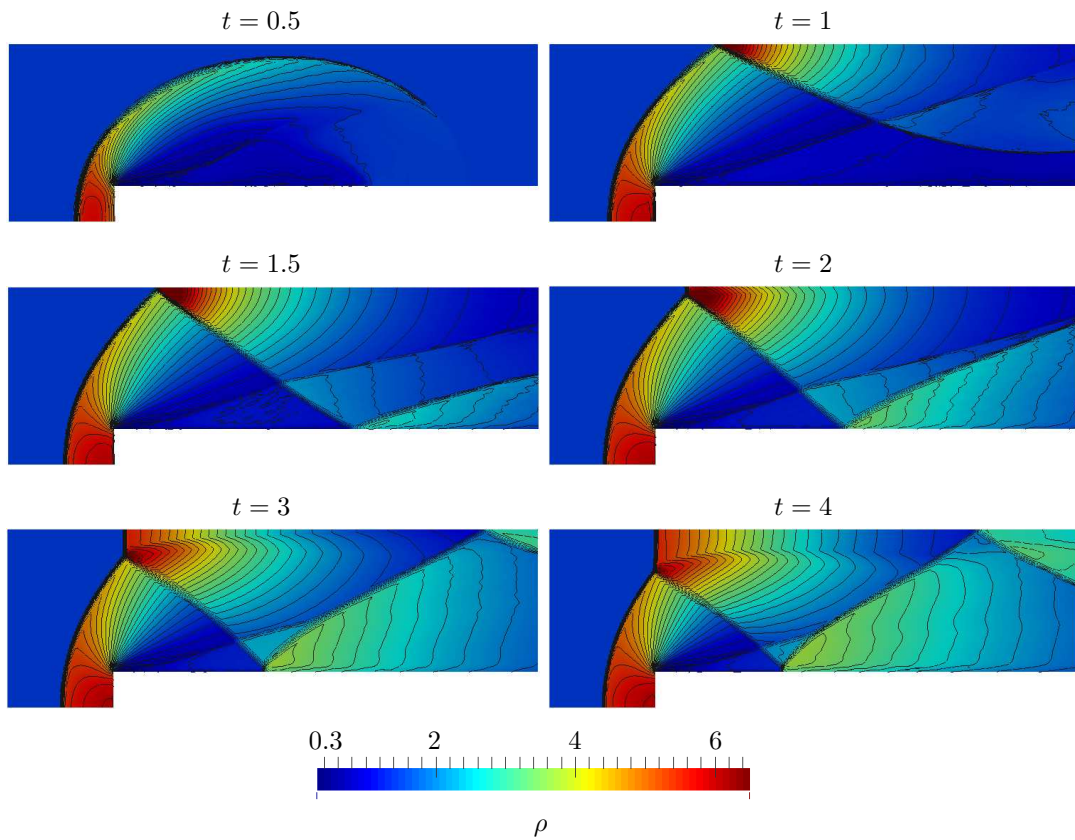


Figure D.3.: Mach number contours at different times for the forward-facing step problem using an uniform grid with 46254 \mathbb{P}^1 DG -elements.

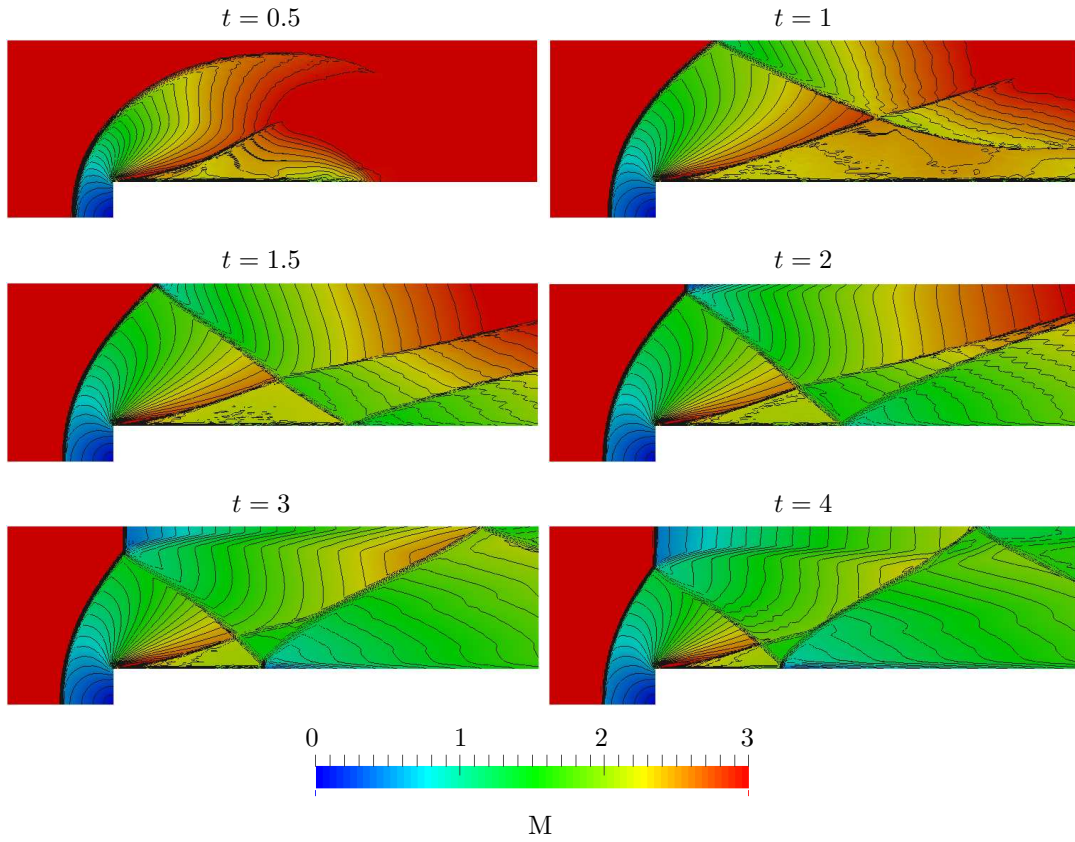
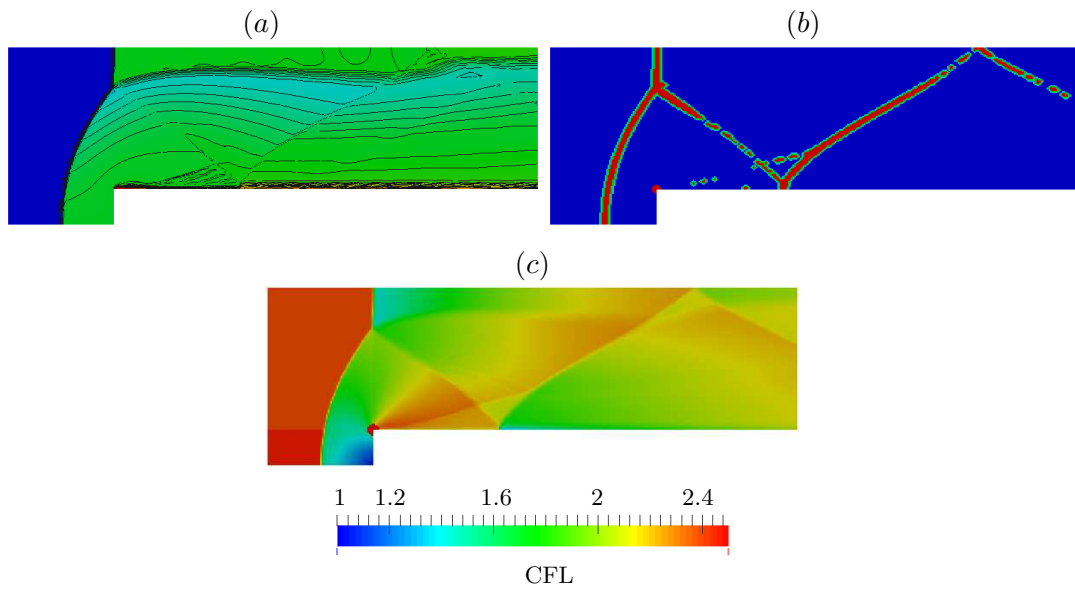


Figure D.4.: (a) κ -entropy production indicator, (b) G -switch indicator and (c) CFL number achieve for the forward-facing step problem at $t = 0.2$ using an uniform grid with 46254 \mathbb{P}^1 DG -elements.



E. Axisymmetric free underexpanded air-jet

In Sec. 5.3.5 the results found with the simulation of an underexpanded air-jet were shown. Here, some additional results are presented.

Figure E.5.: Temperature predicted at different times for the underexpanded air-jet problem using an adaptive grid.

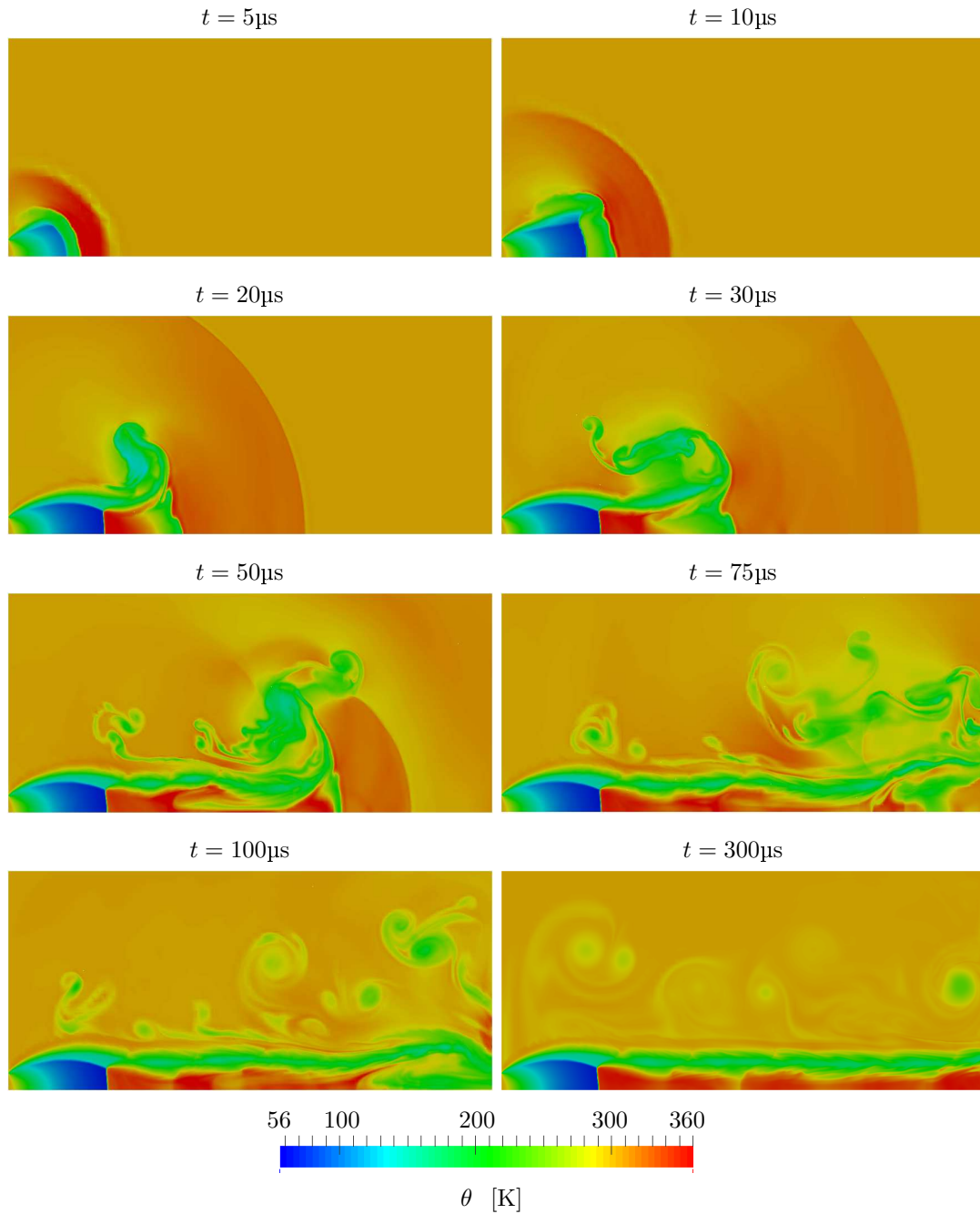


Figure E.6.: Pressure predicted at different times for the underexpanded air-jet problem using an adaptive grid. Thirty contourlines uniformly spaced from 0.004 to 1.1 MPa.

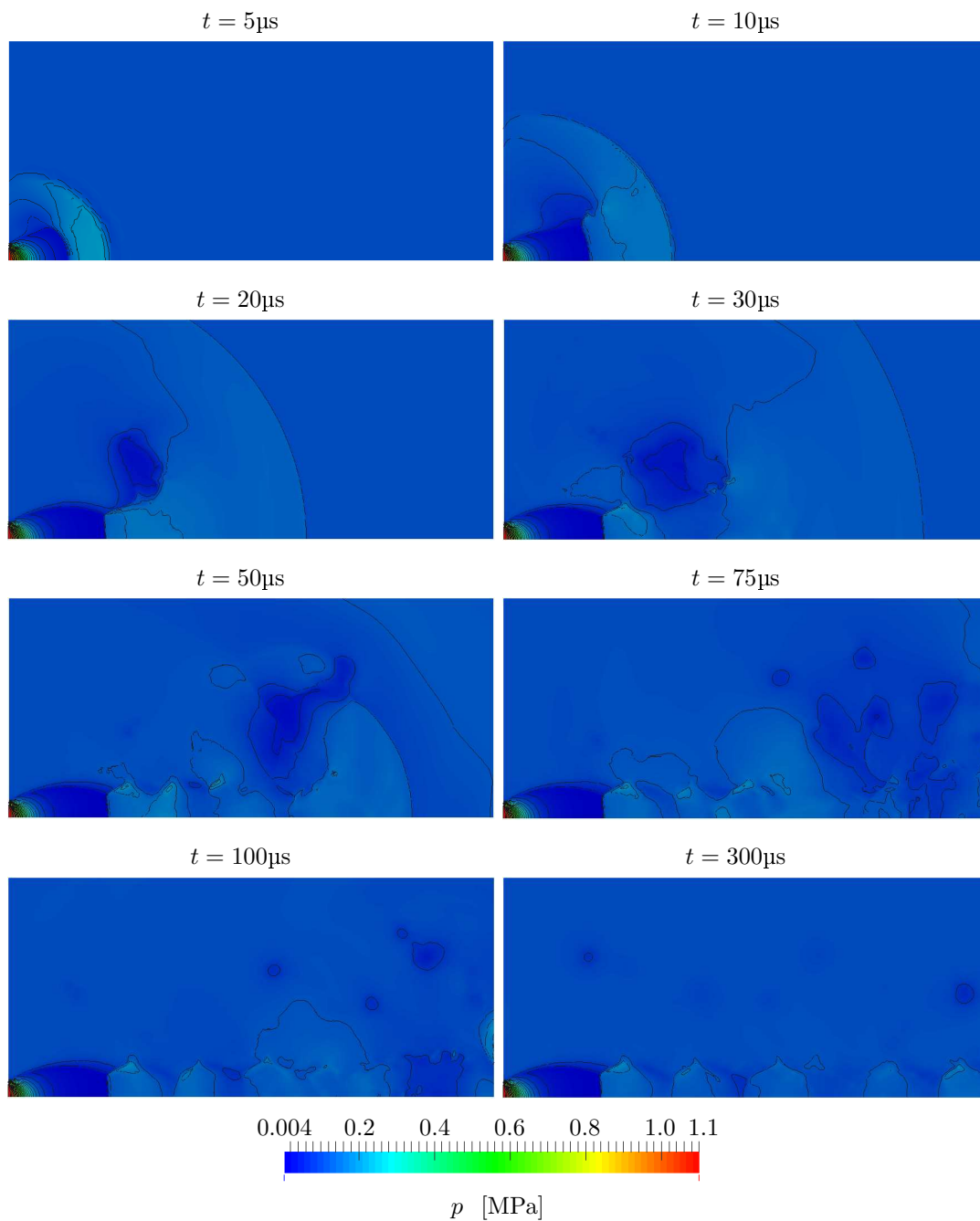


Figure E.7.: Density predicted at different times for the underexpanded air-jet problem using an adaptive grid. Thirty contourlines uniformly spaced from 0.25 to 12.8 kg/m³.

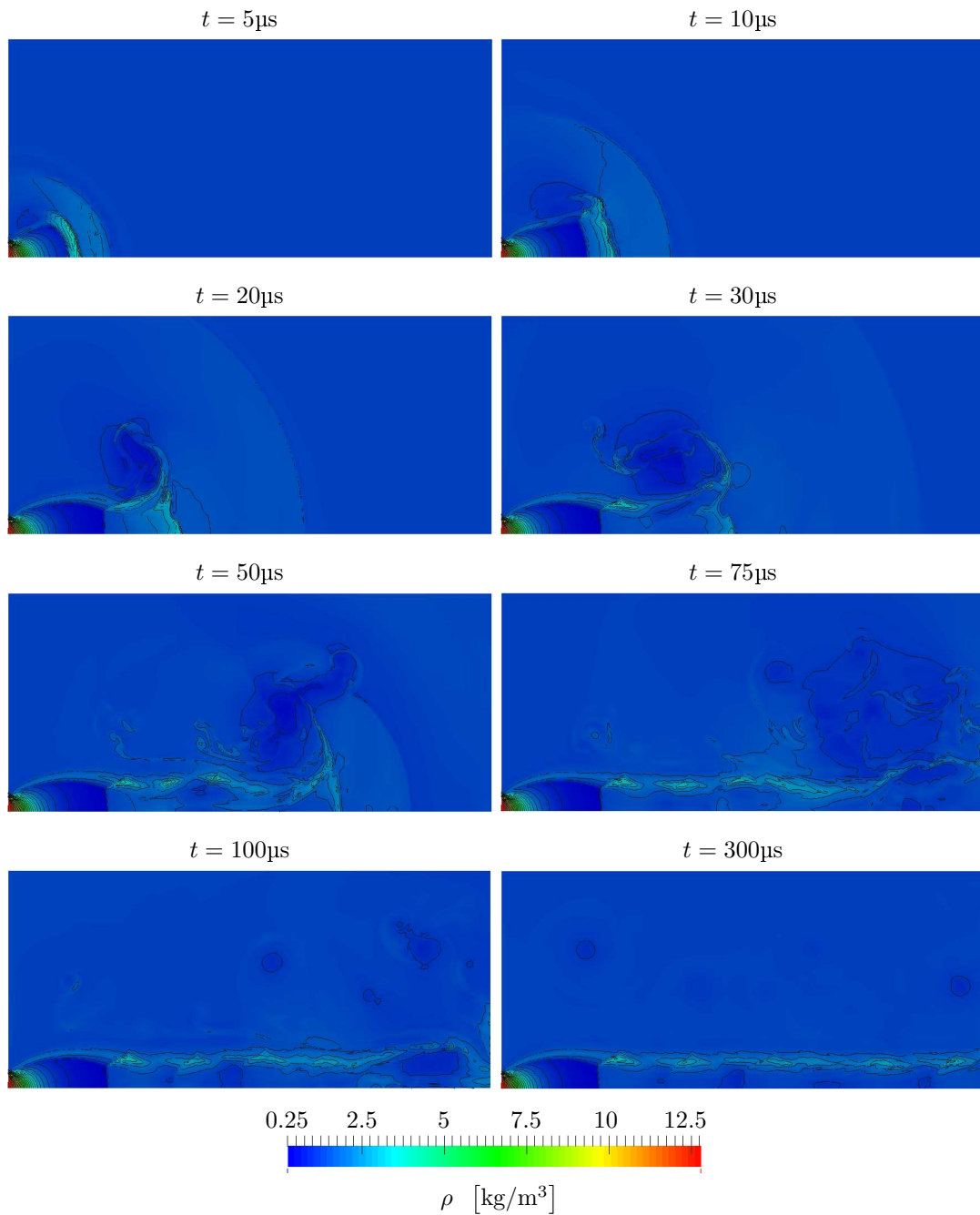
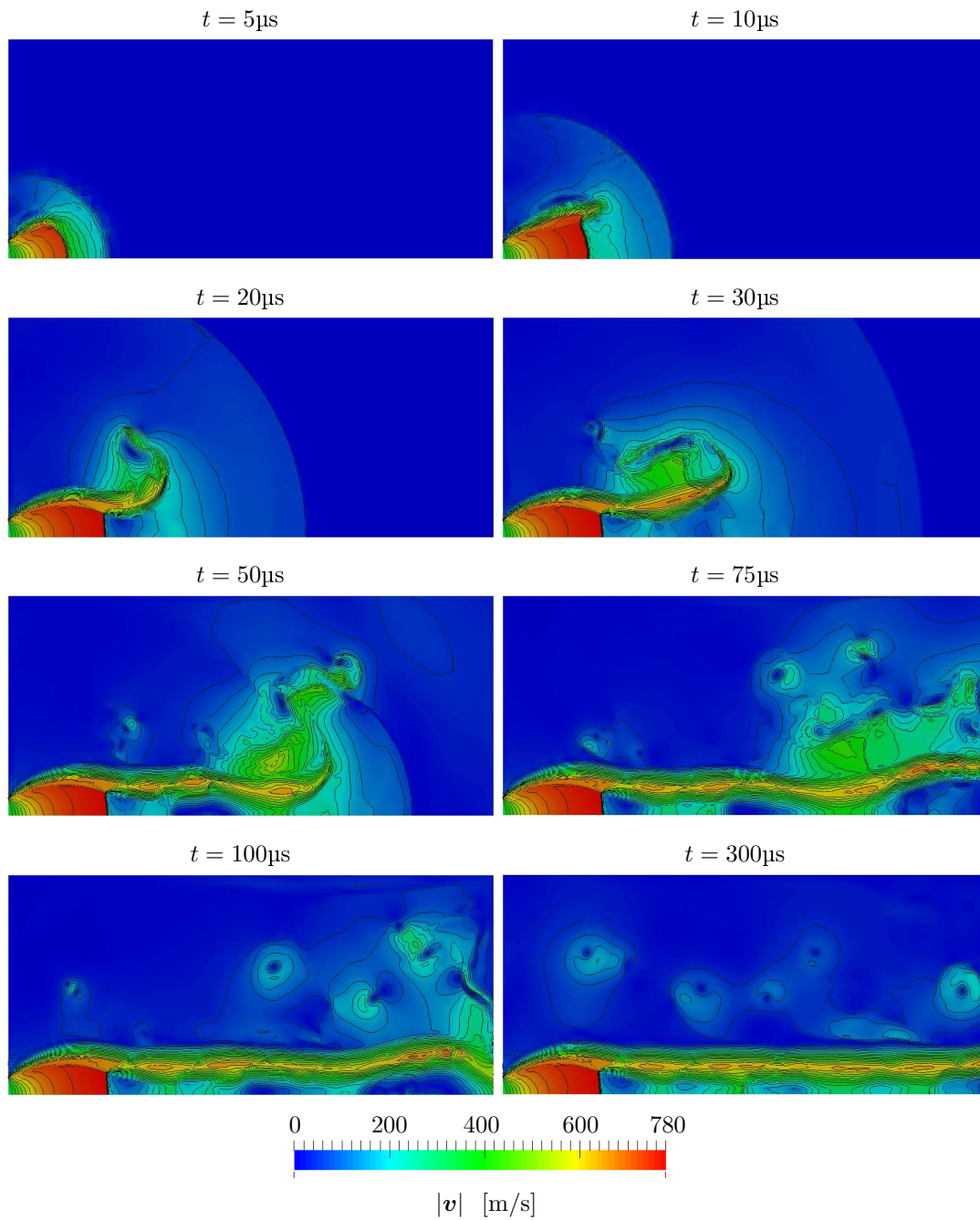


Figure E.8.: Magnitude of velocity predicted at different times for the underexpanded air-jet problem using an adaptive grid. Twenty contourlines uniformly spaced from 0 to 780 m/s.



F. Highly underexpanded CO₂-jet

In Sec. 5.4 the results achieved in the simulation of a highly underexpanded CO₂-jet were presented. Here some complementary results are shown.

Figure F.9.: Temperature predicted at different times for the highly underexpanded CO₂-jet problem using an adaptive grid. Twenty five contours uniformly spaced from 39 to 393 K.

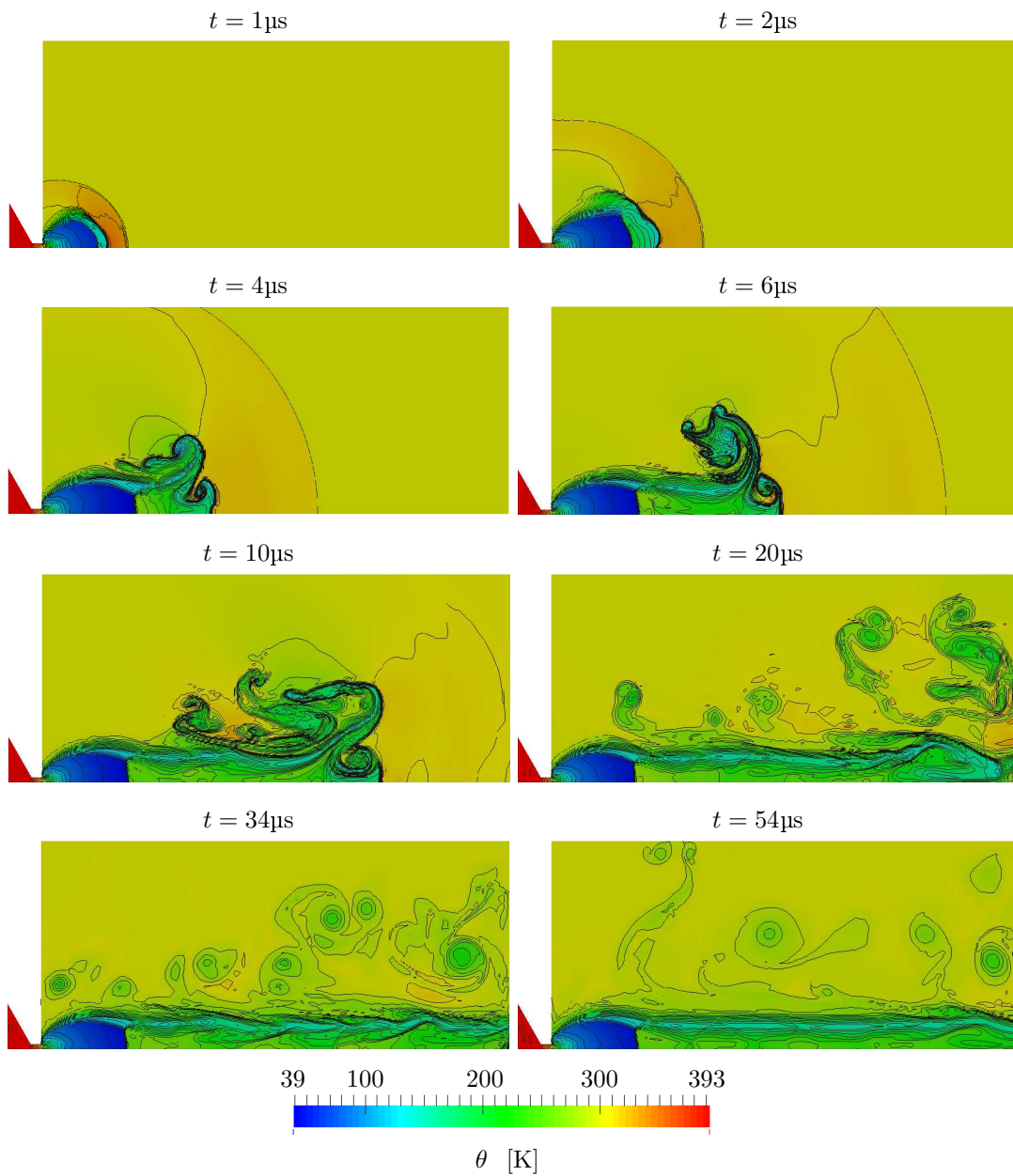


Figure F.10.: Pressure predicted at different times for the highly underexpanded CO₂-jet problem using an adaptive grid. Twenty five contours distributed in logarithmic scale from 0.003 to 20 MPa.

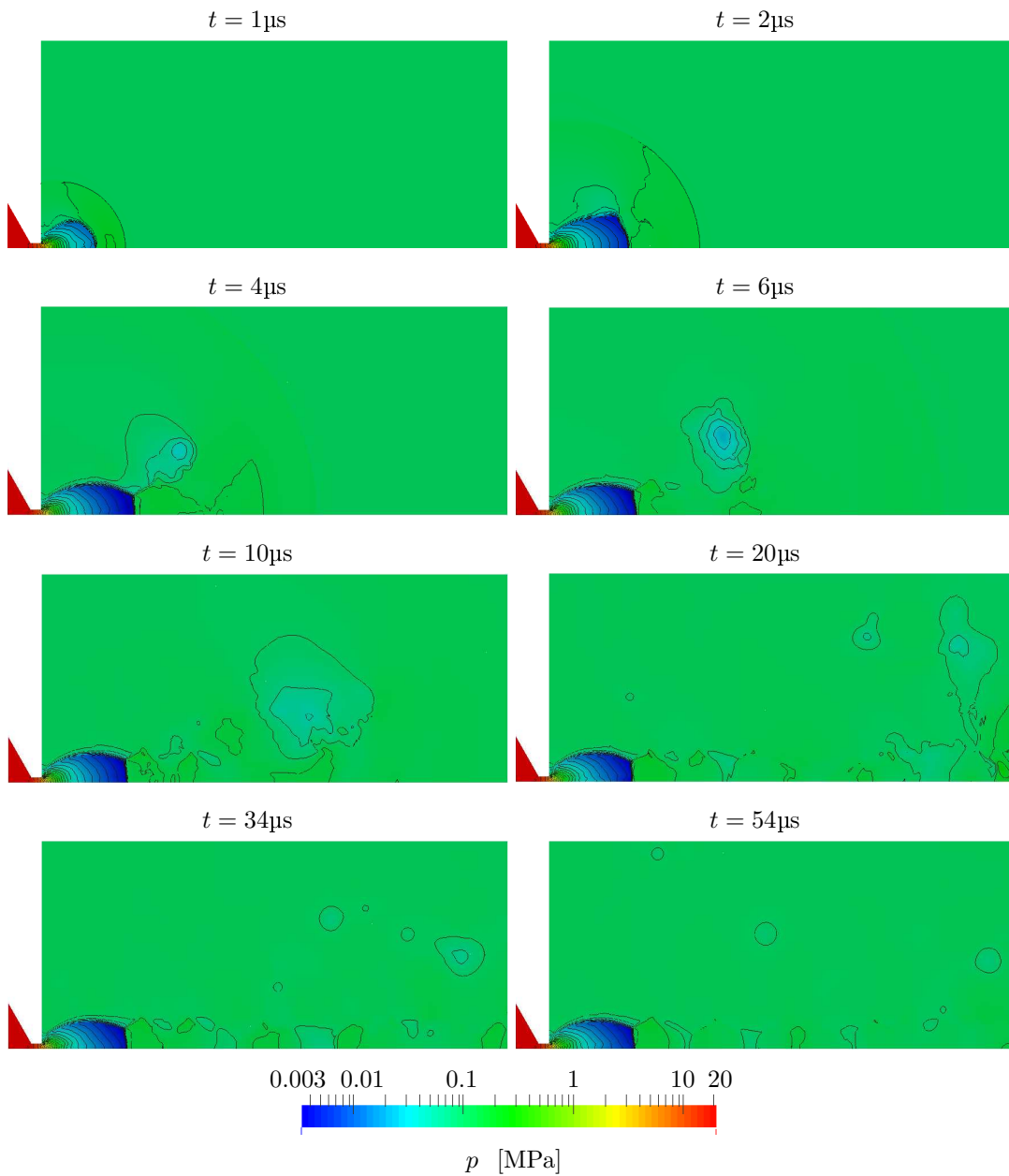


Figure F.11.: Density predicted at different times for the highly underexpanded CO₂-jet problem using an adaptive grid. Twenty five contours distributed in logarithmic scale from 0.4 to 400 kg/m³.

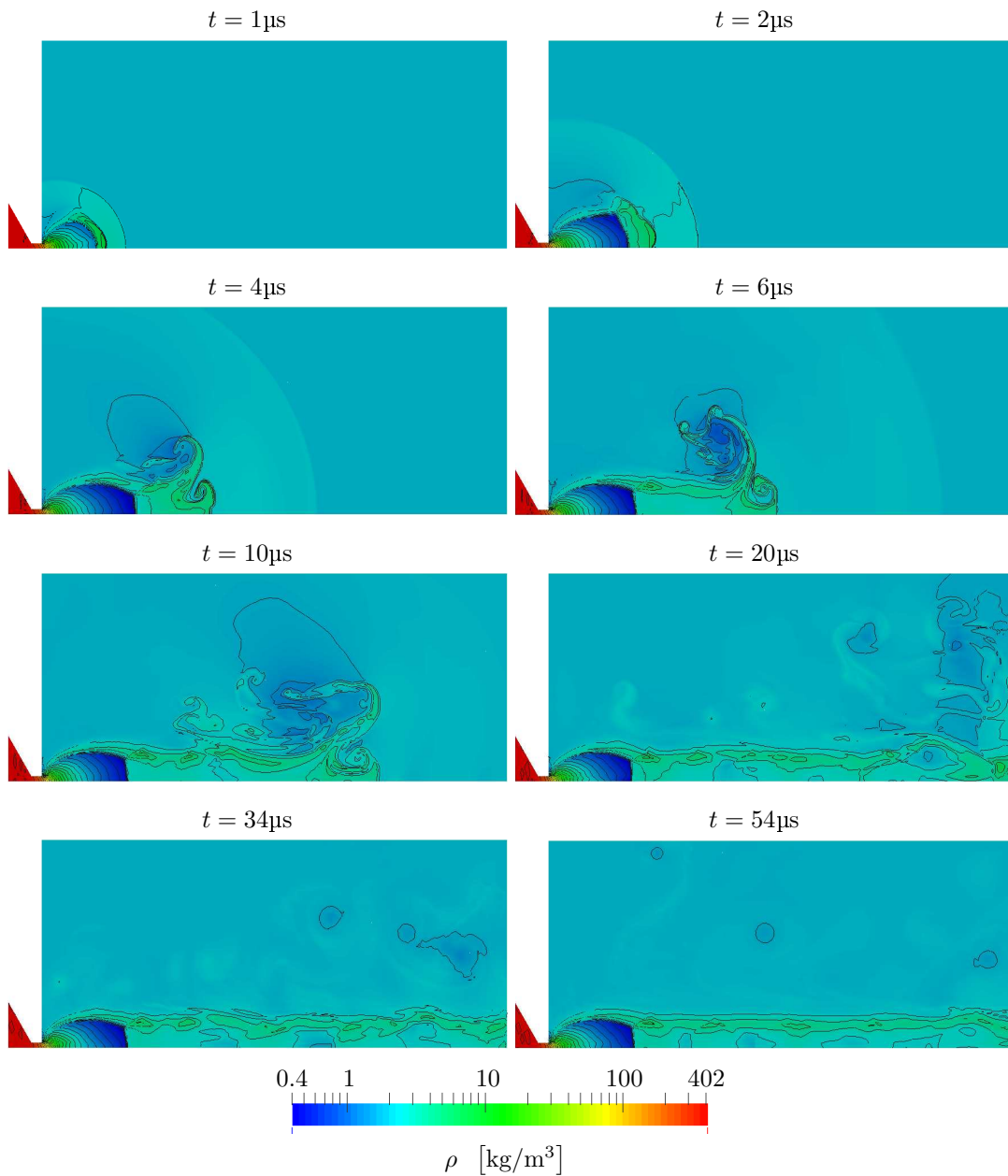


Figure F.12.: Magnitude of velocity predicted at different times for the highly underexpanded CO₂-jet problem using an adaptive grid. Twenty five contours uniformly spaced from 0 to 585 m/s.

