MULTI-PHASE FLOWS USING DISCONTINUOUS GALERKIN METHODS

A Thesis
Presented to
The Academic Faculty

by

Leandro D. Gryngarten

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
December 2012

MULTI-PHASE FLOWS USING DISCONTINUOUS GALERKIN METHODS

Approved by:

Prof. Suresh Menon, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Prof. Yingjie Liu
School of Mathematics
Georgia Institute of Technology

Prof. Stephen Ruffin
School of Aerospace Engineering
Georgia Institute of Technology

Prof. Lakshmi Sankar
School of Aerospace Engineering
Georgia Institute of Technology

Prof. Marilyn Smith
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: 24 August 2012

To Isaias(z'l), Berta(z'l), Bernardo, Berta,

Gabriel, Silvia, Giselle,

and Bela.

ACKNOWLEDGEMENTS

This study has been sponsored by GE Aviation and also in part by Air Force Office of Scientific Research. Computations were mostly conducted at the Computational Combustion Laboratory computing facility in Georgia Tech. The Department of Defense (DoD) High Performance Computing Modernization Program (HPCMP) at the Engineer Research and Development Center (ERDC) and the Air Force Research Laboratory are also gratefully acknowledged for providing resources to conduct some of the code development effort.

I would like to express my gratitude to those who directly or indirectly made this work possible. Everybody in the Aerospace Engineering deserves a recognition, since without them none of this would have been possible. I would like to thank Prof. Suresh Menon, for guiding me and giving me the opportunity to explore aspects of my own interest. Also, I would like to thank professors Y. Liu, S. Ruffin, L. Sankar, and M. Smith for taking the time to be in my committee.

Nowadays, it is not possible to conduct cutting-edge research without a team. Thus, I deeply thank to all my classmates and lab-mates I worked with since I started my graduate studies, particularly to A. Smith, O. Kozaka, B. Colcord, M. Masquelet, T. Gallagher, J. Choi, M. Sanchez Rocha, D. Donzis and E. Kurth. Without their discussions and contributions, no research and study would have been possible.

My deepest gratitude goes to my family – Gabriel, Silvia, and Gigi – for supporting and encouraging me all along this long path. I also include those who became part of my family – Martin, Ed, Yoyi, Evin, Noah, Moshe, and Leah – and also encouraged and supported me. Finally, even though she deserves to be mentioned first, I know she is one of the very few (or only one) who is going to read down to the end of every

single paragraph. I thank my wife, Bela, for laughing with me during the good times, supporting me in the hard times, and making every day better.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS OR ABBREVIATIONS

α_ξ	Volume fraction of phase ξ.
AMR	Adaptive mesh refinement.
AP-TVD	Accuracy Preserving Total Variation Diminishing.
ASL	Absorbing Sponge Layer.
$\bar{\triangle}$	Characteristic filter size.
BC	Boundary Condition.
C_ν	Dynamic parameter for LDKM.
C_ϵ	Dynamic parameter for dissipation term in LDKM.
CDG	Compact Discontinuous Galerkin.
CFD	Computational Fluid Dynamics.
C_s	Smagorinsky constant.
CSF	Continuum Surface Force.
CSS	Continuous Surface Stress.
c_{Vk}	Specific heat at constant volume of phase k.
DG	Discontinuous Galerkin.
DIM	Diffused Interface Method.
DNS	Direct Numerical Simulation.
EOS	Equation of State.
FE	Finite Element.
FV	Finite Volume.
G	Filter kernel.
GFM	Ghost Fluid Method.
IG	Ideal gas.
κ	Interface curvature.
LDG	Local Discontinuous Galerkin.

LDKM	Localized Dynamic k-equation Model.
LES	Large-Eddy Simulations.
LODI	Local one-dimensional inviscid.
LS	Level Set.
l_s	Smagorinsky lengthscale.
$\mathbf{n}$	Interface normal.
ML	Moment limiter.
N-S	Navier-Stokes.
ν_t	Turbulent kinematic viscosity.
Ω	Domain.
P	Pressure.
PML	Perfectly Matched Layer.
PR	Peng-Robinson.
RK	Runge-Kutta.
R_u	Universal gas constant per moles.
SD	Spectral Difference.
SDC	Spectral Deferred Correction.
SEM	Spectral Element Method.
SG	Stiffened Gas.
sgs	Sub-grid scale.
σ	Surface tension coefficient.
SSP	Strong Stability Preserving.
T	Temperature, or mixture temperature in DIM.
T_k	Temperature of phase k.
TVD	Total Variation Diminishing.
U-RANS	Unsteady Reynolds Averaged Navier-Stokes.
VDW	van der Waals.

VMS	Variational Multi-Scale.
VOF	Volume of Fluid.
WENO	Weighted essentially non-oscillatory.
$\widehat{\triangle}$	Characteristic test-filter size.
W_k	Molecular weight of species/phase k.

SUMMARY

This thesis is concerned with the development of numerical techniques to simulate compressible multi-phase flows, in particular a high-accuracy numerical approach with mesh adaptivity.

The Discontinuous Galerkin (DG) method was chosen as the framework for this work for being characterized for its high-order of accuracy –thus low numerical diffusion– and being compatible with mesh adaptivity due to its locality. A DG solver named DiGGIT (Discontinuous Galerkin at the Georgia Institute of Technology) has been developed and several aspects of the method have been studied. The Local Discontinuous Galerkin (LDG) method –an extension of DG for equations with high-order derivatives– was extended to solve multiphase flows using Diffused Interface Methods (DIM). This multi-phase model includes the convection of the volume fraction, which is treated as a Hamilton-Jacobi equation. This is the first study, to the author's knowledge, in which the volume fraction of a DIM is solved using the DG and the LDG methods. The formulation is independent of the Equation of State (EOS) and it can differ for each phase. This allows for a more accurate representation of the different fluids by using cubic EOSs, like the Peng-Robinson and the van der Waals models. Surface tension is modeled with a new numerical technique appropriate for LDG. Spurious oscillations due to surface tension are common to all the capturing schemes, and this new approach presents oscillations comparable in magnitude to the most common schemes. The moment limiter (ML) was generalized for non-uniform grids with hanging nodes that result from adaptive mesh refinement (AMR). The effect of characteristic, primitive, or conservative decomposition in the limiting stage was studied. The characteristic option cannot be used with the ML in multi-dimensions.

In general, primitive variable decomposition is a better option than with conservative variables, particularly for multiphase flows, since the former type of decomposition reduces the numerical oscillations at material discontinuities. An additional limiting technique was introduced for DIM to preserve positivity while minimizing the numerical diffusion, which is especially important at the interface. The accuracy-preserving total variation diminishing (AP-TVD) marker for "troubled-cell" detection, which uses an averaged-derivative basis, was modified to use the Legendre polynomial basis. Given that the latest basis is generally used for DG, the new approach avoids transforming to the averaged-derivative basis, what results in a more efficient technique. Furthermore, a new error estimator was proposed to determine where to refine or coarsen the grid. This estimator was compared against other estimator used in the literature and it showed an improved performance. In order to provide equal order of accuracy in time as in space, the commonly used 3^{rd}-order TVD Runge-Kutta (RK) scheme in the DG method was replaced in some cases by the Spectral Deferred Correction (SDC) technique. High orders in time were shown to only be required when the error in time is significant. For instance, convection-dominated compressible flows require for stability a time step much smaller than is required for accuracy, so in such cases 3^{rd}-order TVD RK resulted to be more efficient than SDC with higher orders.

All these new capabilities were included in DiGGIT and have provided a generalized approach capable of solving sub- and super-critical flows at sub- and super-sonic speeds, using a high-order scheme in space and time, and with AMR. Canonical test cases are presented to verify and validate the formulation in one, two, and three dimensions. Finally, the solver is applied to practical applications. Shock-bubble interaction is studied and the effect of the different thermodynamic closures is assessed. Interaction between single-drops and a wall is simulated. Sticking and the onset of splashing are observed. In addition, the solver is used to simulate turbulent flows, where the high-order of accuracy clearly shows its benefits. Finally, the methodology

is challenged with the simulation of a liquid jet in cross flow.

CHAPTER I

INTRODUCTION AND MOTIVATION

1.1 Background

Design of state-of-the-art turbine engines requires a good understanding of the mixing of fuel and air at intricate conditions. These may include the fluids at sub- or super-critical states, in gas phase or liquid phase, and even transitioning from one state to the other. In addition, these types of flows include a wide range of scales and complex geometries.

The same physics and challenges can be found in other multi-phase flow applications, from rocket engines to painting and from ship design to metallurgy. In particular, the interaction of a shock or pressure wave with a material discontinuity is a problem observed in a wide range of circumstances, from shock-wave lithotripsy [27] to explosives [17].

So far, most of the knowledge obtained is experimental, but this approach tends to be very complex and is subject to technical limitations when obtaining some measurements. Experimental data tends to include uncertainties that do not allow correct validation of the theory. As a consequence, numerical simulations are a promising method for gaining further insight into the physics, especially as computational-power limitations have decreased. Nonetheless, simulations require accurate validation data from theoretical approaches or experimentation.

Accurate numerical studies can provide important information, especially at complicated conditions where it can be difficult to extract measurements experimentally. Obviously, these flows are also challenging for simulations, what makes compressible multi-phase flows an area of active research. For decades scientists and engineers

working in the field of computational fluid dynamics (CFD) have been trying to improve numerical schemes in order to simulate increasingly complex flows with greater accuracy and efficiency. Simulations of multi-phase flows present an additional complication than one-phase flows, which is the existence of a change in properties among a length scale several orders of magnitude smaller than the scale of the problem. That area with a high gradient of properties is the interface between both fluids. Resolving this gradient turns out to be impractical because of the large range of length-scales present. One solution has been to treat the interface as an area with a sharp change of properties or even a complete discontinuity.

Fortunately, the advancement of the computational power has eased some challenges. However, even with the current computational power, this kind of simulations cannot be conducted resolving all the scales, namely, from the size of the system to the size of the droplets in the spray, including the broad range of turbulent scales. In order to circumvent this limitation, mathematical techniques have been developed to ease the computational needs. For instance, for highly turbulent flow Large Eddy Simulations (LES) can be conducted, where the large scales of the turbulence is resolved, while the small scales are modeled. At the same time, the droplets in a spray tend to be smaller than the grid resolution allowed by the computational resources, thus the sprays are usually modeled assuming that the droplets are point-particles and a Lagrangian formulation is used for each particle or group of particles. This formulation requires the interaction of the flow with the droplets to be modeled and this is where the study of individual droplets and groups of droplets come into the scene.

Given the increased numerical challenges, the decrease of computational constraints, and the limitations of traditional numerical methods, the discontinuous Galerkin (DG) method has widely grown in popularity. It has shown promise as a modern simulation technique, which combines the strengths of several other classes

of schemes.

1.2 Literature review

1.2.1 Multi-phase models

The terms "multi-phase" and "multi-fluid" are used almost interchangeably, so their definitions should be discussed before we continue using them carelessly. The following definitions are given in [137]:

- Multi-phase: "same fluid, different phases".
- Multi-fluid: "different fluids".

A phase is a region of space where the physical properties of a material are essentially uniform. Thus, a multi-fluid flow is a multi-phase flow. In addition, the following can be found in [117] in a numerical context:

- "In multiphase flows, a control volume contains a large number or individual particles (or bubbles, etc.) with many interfaces".
- "in multifluid flows, nearly all control volumes contain pure phases, except for the computational cells around the interface."

In this thesis the term multi-phase is used for the general case of more than one phase (or fluid) present, while the term multi-fluid is reserved for the specific case of two different fluids where the interface is being resolved.

Many numerical multi-phase approaches exist in the literature and can be split into two groups. In the first one the interface is modeled as a discontinuity and is composed by the following techniques. The more traditional approaches consist of some form of interface tracking, where a marker at the interface is convected [138, 42]. Even though this method can be accurate, an important disadvantage is that it is difficult to simulate large deformations, even more when the interface disconnects or

reconnects. Other techniques capture the interface instead of tracking it. A common approach of this type is the volume of fluid (VOF) method [47, 38]. Unfortunately, geometric properties of the interface (e.g., curvature) are difficult to compute with this method, especially in three dimensions, and are mostly limited to incompressible flows [142]. The Level-Set (LS) method initially presented by Osher and Sethian [96] solves these difficulties by using a smoothly varying function to identify each fluid. This approach is usually complemented by the ghost-fluid method (GFM) to avoid pressure oscillations at the interface [36]. The LS method has mostly been applied to incompressible fluids and has the scheme-dependent problem of not conserving mass [142, 90, 89]. The LS and VOF have been combined, usually called CLSVOF, to reduce mass loss, however, it is still limited by the complexity of VOF and mostly to incompressible flows [92, 142].

The second group is usually called Diffused Interface Method (DIM). In this approach the interface is diffused and an artificial mixture zone is created. One unique set of governing equations are used in the whole domain. Also, the thermodynamic closure is valid at each pure fluid and at the interface. These properties make the DIM easier to implement than the methods mentioned above. In addition, they usually conserve mass, allow for topological changes, and do not require moving grids. Nonetheless, the cost to pay is the high resolution needed to keep the effects of the artificial thickness of the interface small in contrast to the flow features. The DIM is inspired by the fact that real interfaces have a finite thickness that vary depending on the thermodynamic properties and are the cause of surface tension effects [54]. In many applications this thickness is several orders of magnitude smaller than the flow features. The phase-field theory [60, 61] is a microscopic way to capture surface tension effects by physically resolving the interface. The Second Gradient Method can be used in the context of the phase-field theory, especially to simulate flows close to the critical point. Jamet et al. [63] used the Second Gradient Method with an artificially

thickened interface to reduce the range of scales. Even though phase change is naturally represented by this approach, so far this method is only useful for fundamental research of small scale problems.

Most multi-phase implementations are incompressible. This has the advantage of allowing larger time steps and reducing the computational cost. However, they cannot include acoustic effects and simulate cases where the density changes significantly within one fluid. This is particularly important close to the critical point or the trans-critical region, where small displacements in the state diagram produce large changes in the material properties and density.

Saurel & Abgrall [117] present a compressible model for two-phase flows by formulating the Navier-Stokes (N-S) equations for each constituent and using an averaging procedure, as suggested by Baer & Nunziato [6]. Subsequently, viscosity and mass transfer were included in [118]. The result is a system of 7 equations similar to the N-S, but with some specific terms that do not allow a conservative formulation. If mass and/or heat transfer are present, they have to be closed depending on the specific problem: evaporation, condensation, combustion, etc. The 7-equation model includes relaxation between the pressure and velocity fields of different phases. If relaxation characteristic times are relatively much smaller than the characteristic times of the flow, these non-equilibrium effects can be neglected. As a result of this model, when only two phases are present, following an asymptotic analysis to find the equilibrium conditions and assuming some simplifications that are only valid for interface problems, the system can be reduced. If the velocity non-equilibrium is relaxed, the 6-eq. model is obtained [116]. If the pressure non-equilibrium is also relaxed, the systems results in 5 equations [100]. This approach conserves the total energy, the momentum and the mass of each individual phase.

Further simplifications of the governing equations can be done. For instance, the reader can find examples of the 4-equation model in [125, 1] and an equivalent

formulation in [66, 160]. Although the 4-eq. model is easier to implement than the 5-equation model, it has some disadvantages: it only conserves the total mass, instead of the mass of each phase, and the individual thermodynamic properties cannot be recovered [1], which are needed for some Equations of State (EOS).

Surface tension is a considerable effect in several problems, including when breakup or coalescence is involved. Brackbill et al. [18] pioneered the field of incorporating surface tension in diffused interfaces. They proposed the Continuum Surface Force (CSF) method for modeling capillarity in the form of a volumetric force, reformulating a discontinuous interface effect in a continuum framework. In this case the volumetric force is non-zero in the transition region of finite thickness representing the diffused interface. The CSF method has been applied to VOF, LS, and DIM [137, 100]. Another method is the Continuous Surface Stress (CSS) method, which allows for variable surface tension. Every method that captures the surface tension presents spurious oscillations. In some cases, these instabilities are so large that they can break a drop apart. The CSF and CSS were shown to present large spurious oscillations at the interface unless some smoothing is applied. Instead of explicitly smoothing the interface, the Proper Representation Of Surface Tension (PROST) method can be used, which uses a least-squared approximation at each point before applying the CSF [114]. The disadvantage of this approach is the additional computational cost. Originally, it was shown for VOF, but Marchandise et al. [88] adopted the idea of the least-squared approximation to smooth out the interface for LS. The CSF is usually implemented as a source term. However, surface tension effects have been treated in a conservative fashion in [100, 60]. It would seem that the later option leads to lower spurious oscillations [100].

As many other multi-fluid models, a fluid marker has to be convected and in the case of 4- and 5-eq. model it is the volume fraction. Even though this equation is not in conservative form, the mass, momentum, and total energy equations are. Thus,

these two models conserves the mass, the momentum, and the total energy. Finally, DIM is efficient and it is the easiest method to implement [137] since it does not require special treatment of the interface.

Compressible two-phase models, like the 5-eq. model, usually use the Stiffened Gas (SG) Equation of State (EOS) to model pure liquids and gases. Its popularity is due to its simplicity and flexibility to be adjusted to match certain properties. Allaire et al. [1] implemented the 5-eq. model with each phase modeled by a different EOS. Even though their approach is robust, it puts constraints on the fluid properties that make general EOSs difficult to apply, especially for real gases.

Even though DIMs may present the difficulty of allowing the interface to diffuse too much, especially when high shear is located at the interface, great efforts can be found in the literature to prevent it [71, 123, 127].

1.2.2 Interface sharpening

Most numerical schemes present some numerical dissipation, partially, to provide stability or prevent unphysical oscillations. This is the case of shock-capturing schemes.In the case of shock waves modeled by the Euler equations as discontinuities, the physical phenomena makes the shock become sharper balancing the numerical diffusion. However, in contact discontinuities there is no natural behavior that counteracts the effect of the numerical dissipation of the scheme, justifying the need of a numerical artifact to control the thickness of the interface.

A lot of effort has been invested trying to overcome this problem. The most helpful approach is a low-diffusive scheme, although any diffusion will smooth out the solution for very long simulations. Harten [51] proposed the artificial compression method, extended by Yang [153]. Later, Harten [53] presented the sub-cell resolution method. Shu & Osher [122] compared these two methods in the ENO framework. They observed that although the sub-cell resolution method could not be properly

generalized to two dimensions, it presented slightly better results in 1-D than the artificial compression method. At the same time, Yang's approach counts with a problem-dependent parameter that needs to be adjusted and the result in 2-D is not as satisfying as for 1-D. Deprés & Lagoutière [31] presented a new approach that had the problematic of producing stepping results. This was fixed later by Bouchut [15] avoiding entropy violations, however it was only shown for a monotone scalar conservation law in 1-D. Xu & Shu [150] applied the approach in [31] for a fifth-order WENO scheme. They showed that first-order Euler forward time discretization is unstable –needing at least a second-order Runge-Kutta time discretization– and that shapes may get slightly distorted in 2-D. Bokanowski [14] presented a new limiter, the N-bee, and compared it to the Ultra-bee for the advection and the Hamilton-Jacobi-Bellman equations, however it also presented some over-compressibility.

A completely different approach was presented by Kokh [70] for the 4-eq. two-phase model, where a source term compensates the numerical diffusion. This term has a problem-dependent parameter that needs to be adjusted. A more sophisticated approach was shown by Shukla et al. [123]. The previously mentioned problem-dependent parameter is avoided by applying a relaxation scheme at the end of every time step. In addition, they presented a correction term for the density, which was ignored in [70]. Even though correcting every equation with source terms bring conservation issues, affecting only the volume fraction brings instabilities at the interface. Nonetheless, the conservation property can be preserved if the correction is applied as a flux instead of a source term. Recently, So et al. [127] presented an approach for FV, also using the relaxation idea, but replacing the source term by a viscous term with negative diffusion coefficient. Although the implementation is more complex, the conservation property is preserved. A conservative approach was also proposed by [128] using FD and applied to waves. In order to avoid the ad-hoc complexity of those techniques or at least to reduce their need, a scheme with low numerical

dissipation should be used.

1.2.3 Discontinuous Galerkin

The discontinuous Galerkin (DG) method was first derived in 1973 by Reed and Hill [111] to study neutron transport. It was not used for fluid mechanics until after Cockburn et al. generalized the method for hyperbolic equations in conservative form with first order derivatives [23, 22, 21, 25].

The DG method belongs to the finite element (FE) family and uses a piece-wise discontinuous space for the test function and the numerical solution [26]. The use of the same function space for the test function and solution defines all Galerkin methods. Usually, the basis to form the space is composed of Legendre [26] or Lagrange [41] polynomials, although other options have been studied in the literature [158]. The discontinuity is localized at the boundary of each element and the coupling between elements is done by computing fluxes as in finite volume (FV) schemes, e.g., using an approximate Riemann solver. This kind of coupling allows DG to formulate each element locally, making the implementation highly parallelizable, h-p adaptivity friendly, compatible with complex geometries, and is capable of achieving high-orders of accuracy even with unstructured grids and hanging nodes (e.g., see [112]). Given that DG is a result of FE and FV, the terms element and cell are generally used indistinctly in this context.

DG can only solve governing equations with first-order derivatives in space. Several modifications have been suggested for high-order spatial derivatives, including elliptic and parabolic partial differential equations, since is needed for several common applications like with the Navier-Stokes equations due to the viscous terms. Most options for elliptic equations were discussed in a unified way in [2], including the interior penalty (IP) method and the BR2 method [9]. The most popular technique has been the Local Discontinuous Galerkin (LDG) method [24] for its stability and simplicity.

In this approach, the governing equation is rewritten as a system of equations with only first-order derivatives. This way, it can handle higher-order derivatives [152]. The Compact Discontinuous Galerkin (CDG) method [99] provides a smaller stencil that LDG, only beneficial for schemes that require assembling a global matrix. The recovery DG method was suggested in [139], which consist in using a least-square approximation of the solution at the face to avoid the discontinuity. Unfortunately, this approach is more computationally expensive than the previous options.

The time integration scheme most widely used has been the ubiquitous 3^{rd}-order TVD Runge-Kutta (RK) method, leading to what is known as the RKDG method [23, 22, 21, 25]. Given that DG has the ability to easily achieve high-order spatial accuracy, some effort to maintain comparable time accuracy has been reported [148]. Under some conditions, especially with higher order derivatives, the time step required for stability of the RKDG method can be very limiting. Recently, Xu & Shu [148] suggested that the Spectral Deferred Correction (SDC) method, derived by Dutt et al. [35], may be an alternative time stepping scheme. It has been shown that SDC can be used in an explicit, semi-implicit, or fully implicit form, and it is easy to extend to high-order accuracy in time [93]. Xia et al. [145] studied a semi-implicit SDC method, in addition to other alternative techniques, to use with the Local Discontinuous Galerkin (LDG) method. SDC combined with DG (SDC-DG) has not yet been used extensively for practical applications. Grooss and Hesthaven [46] used a semi-implicit SDC to solve the incompressible Navier-Stokes with free-surface flows. Even though Gottlieb et al. [45] presented RK methods of order higher than 3, these schemes are very difficult to derive, while the extension of SDC to any order is straightforward. In addition, TVD-RK methods of 4^{th}-order or greater require the governing equation to be invariant to time reversal [44, 45]. The Euler equations are invariant to this transformation, but the Navier-Stokes (NS) equations are not. The possibility of an SDC method with the strong stability preserving (SSP) property

was studied by Gottlieb et al. [43] and more extensively by Liu et al. [84]. Note that TVD schemes are SSP schemes that were originally derived using the total variation norm [45], instead of a generic norm. Therefore, in practice the TVD and the SSP properties are equivalent, but TVD could be considered a particular case of SSP. Liu et al. [84] showed that SSP-SDC algorithms can be obtained, but the derivation gets very complicated as the order increases and the CFL coefficient is smaller than for the SSP-RK.

The current method also combines the SDC-DG approach with adaptive mesh refinement (AMR) to dynamically and locally refine or coarsen the grid based on an estimation of the numerical error. Issues with the implementation such as hanging nodes, particularly in quadrilateral or hexahedral grids are addressed. The DG formulation works well with AMR because of its local nature [112] and its performance is demonstrated in this paper.

As with other numerical approaches, it is well known that DG methods may cause non-physical oscillations close to discontinuities due to the Gibbs phenomenon, especially when higher order schemes are used because of lower numerical dissipation. Therefore, some approach to "limit" this effect is needed. One common technique consists of applying limiters inherited from FV techniques, several of which have been developed in the last two decades. Cockburn & Shu [23] demonstrated a modified *minmod* limiter for the DG method, but it has the disadvantages of dropping the order of accuracy when it is activated and relies on a user-defined parameter to make it total variation bounded (TVB) instead of total variation diminishing (TVD). Qiu & Shu [108] showed that the weighted essentially non-oscillatory (WENO) approach, borrowed from FV, can smoothen the non-desired oscillations but increases the size of the stencil and loses the sub-cell information that DG provides. In a later study, Qiu & Shu [107] used a modified WENO scheme based on Hermite polynomials to reduce the stencil.

Other limiters, such as the moment limiter (ML), originally proposed by Biswas et al. [11] for uniform grids and further improved, e.g., by Krivodonova [72], has also been proposed for DG applications. The ML is generally applied to a Legendre polynomial basis limiting the conservative or the characteristic variables. Yang & Wang [154] modified the ML for unstructured grids for a spectral difference (SD) method, applying it to a polynomial basis based on the averaged derivatives along the cell, instead of estimating the derivatives at the cell center as in [72]. The hierarchical reconstruction (HR) method, introduced by Liu et al. [86], was applied to DG with a WENO-type reconstruction at each hierarchical level [149]. In this approach characteristic decomposition is not used, but rather small overshoots/undershoots appear especially as the order of accuracy is increased [85]. For DG schemes with very high order elements, artificial dissipation to smooth out discontinuities has also been proposed [62, 101, 8]. In this paper the ML as presented in [72] is modified for non-uniform grids with hanging nodes. The ML is usually applied to characteristic variables, which is only consistent in a one-dimensional sense. Therefore, the consequences of limiting the conservative, primitive, or characteristic variables are studied to later apply it to multi-dimensional cases.

Even though good limiters tend to keep the original order of accuracy in smooth regions, they may increase the error slightly [106, 72]. Hence, the application of such limiters within the domain needs to be minimized. This task is carried out by what is usually called a "troubled-cell" detector, which identifies the cells that may be becoming oscillatory or unstable, and thus require a limiter. Moreover, if the detector is computationally faster than the limiter, the speed of the solver can be increased by reducing the number of cells where the limiter is applied. In the past, several limiters were adapted as detectors [106], and the ones with best success are the minmod-based TVB limiter [23], the shock-detector by Krivodonova et al. [74] (KXRCF), and the indicator based on Harten's sub-cell resolution [52]. In [154], the accuracy-preserving

TVD (AP-TVD) detector is suggested in an SD frame and compared against the other detectors just mentioned above and was shown to produce better agreement. Therefore, the AP-TVD detector is adapted to the DG method with some additional modifications, as reported below.

In order to achieve the best balance between accuracy and efficiency, computational effort should be focused on regions of the domain where "interesting" flow phenomena occur, e.g., at the complex structure of the interface between two phases. This is made possible with adaptive mesh refinement (AMR), also known as h adaptivity, where the grid is dynamically and locally refined or coarsened based on some criteria. Because of its local nature, DG works well with this technique since most of the information needed from a neighboring cell is located on the face, thus refinement or coarsening of neighboring cells does not alter the formulation. AMR requires an indicator to determine where to refine or coarsen the grid based on an estimated numerical error. The numerical error well depends on the scheme, thus accurate error estimators used in Finite Volumes (FV) or Finite Differences (FD) are not valid here. Considerable research has been invested in estimating the numerical error for the DG method for conservative hyperbolic equations (e.g., see [37]), but usually these approaches are computationally expensive and therefore inefficient. Faster –but perhaps less accurate– methods have also been derived for DG. Remacle et al. [112] used a simple error estimator based on the jump between elements, which is the same principle as used in the shock-detector KXRCF. Trouble-cell detectors have also been used as error estimators [113, 162]. Zhu et al. [162] compared a few of them and found that KXRCF provided very good results for typical one-dimensional shock problems. In addition, Leicht and Hartmann [78] used the jump between elements to determine the direction for anisotropic refinement. A similar approach to AMR is the multi-resolution technique, which is more complex but it has a more thorough mathematical proof. Refer to [121] for the application of multi-resolution to DG.

After time integration (sub-steps included) cells may have non-physical values at some of the quadrature or nodal points. In the case of the Euler equation (and Navier-Stokes equation) this usually means negative density, pressure, or temperature. In the case of some DIM this also includes a volume fraction lower than zero or greater than one. Zhang and Shu [159] presented a limiting technique to help preserve the positivity of density and pressure, while minimizing the numerical diffusion, for the Euler equations with the ideal gas assumption. This equation of state makes the relationship between energy, density, and pressure simple, thus, the positivity of pressure is relatively easy to control. However, this is not the case for the two-phase models with generic EOSs (e.g., the Peng-Robinson EOS). Wang et al. [141] suggested a modification to increase the efficiency valid when the pressure is a concave function of the conservative state vector. Unfortunately, this is also not valid for generic EOSs. In [149] a brute-force iterative procedure is applied to enforce pressure positivity. The inconvenient of this strategy is that after every iteration the pressure has to be checked at every quadrature point, what has some computational cost.

The non-linear convection equation for the volume fraction can be expressed as a particular case of the Hamilton-Jacobi equation. Its discretization is not trivial and requires a special approach. In particular for DG, Shu and others showed an approach to solve the Hamilton-Jacobi equation using the RKDG in a series of papers [55, 79, 82, 20]. A caveat of this technique is that under certain circumstances it may violate the entropy criteria, in which case an ad hoc approach has to be done locally. Wang and Shu [140] applied the RKDG to a compressible multi-fluid model using the GFM. Yan and Osher [151] solved the Hamilton-Jacobi equation using the Local Discontinuous Galerkin (LDG) method. The advantage of this approach is that it does not violate the entropy criteria. Having to use LDG instead of DG is not an inconvenient, since LDG is already required to include viscous terms. Dirichlet conditions are usually strongly imposed in FD and FV schemes. However, the natural way to apply Dirichlet

conditions in DG is weakly. In the context of CFD, this is mostly relevant for wall conditions. Nordstrom et al. [95] studied the convergence to a steady-state solution of the Navier-Stokes equations for weak and strong wall conditions using a FV scheme. For the cases analyzed, the weak approach converged faster and both techniques converged at the same rate when reducing the grid size. Collis [29] compared the weak and strong techniques for DG and concluded that the first one produced more stable results, especially at low resolutions. This is particularly interesting in the context of wall models for LES, where, in general, it is difficult to properly resolve the flow close to the wall.

Given the nature of the DG formulation there are two similar approaches to weakly impose the boundary condition. The boundary condition is specified through the numerical flux at the element face. Thus, one option is to replace the numerical flux used in the interior of the domain by a specific flux based on the properties of the boundary [3]. The other option is to use the same flux as in the interior, but replace the outer state needed in the numerical flux by the state vector at the boundary [9].

The limited domain of analysis leads to the requirement of artificial inflow and outflow boundaries conditions (BC). At the same time, unsteady simulations require minimum numerical dissipation schemes, to avoid non-physical waves originated by the initial value, artificial boundaries and other inaccuracies. Therefore, it is necessary to have a way to get rid of them and one way is with non-reflective boundaries, where ideally any wave goes through them without returning.

The development of non-reflective boundary conditions (BCs) has been a field of active research for years and a common exploited method to formulate them is making use of the characteristics of the flow. In 1990, Thomson [134] presented a classical paper on the treatment of time-dependent BCs for hyperbolic equations where characteristic analysis was used in the frame of finite differences. Subsequently, in 1992 Poinsot and Lele [103] extended this formulation to the Navier-Stokes equations for

calorically-perfect gases, where a local one-dimensional inviscid (LODI) problem was assumed. In 1994, Baum et al. [10] extended the formulation to multicomponent reactive flows for thermally perfect gases. In 2003, Sutherland et al. [129] made evident the need for including the effect of the source terms, previously neglected. In 2005, Yoo et al. [157] presented a treatment that solved some of the problems resulting from the assumptions of one-dimensionality and non-viscosity. In 2007, Yoo and Im [156] improved this formulation even further. Later, Lodato [87] applied it to three dimensional flows with a special edge/corner treatment. Most of the literature express the governing equations in terms of x, y and z coordinates and it is generally assumed that the inflow and outflow boundaries are perpendicular to the x direction. All this can be generalized writing the equations in local orthonormal coordinates, where one direction is normal to the boundary and the other two are tangent to it, as followed by Moureau [94].

Different approaches had been sought to improve the absorption of the boundaries, what resulted in the family known as Absorbing Layers, originally introduced by Israeli & Orszag [59]. In recent years, they have been used to complement the characteristic boundary conditions. The Absorbing Layers have one small disadvantage, which is that an additional computational layer is needed next to the artificial boundary. Absorbing Layers can be reflective themselves, so it has been shown that this effect can be reduced if their strength increases from null inside the domain to a maximum at the domain boundary [30].

Two popular techniques exist in this family. The simplest one is the Absorbing Sponge Layer (ASL), where linear friction is added as source term. This technique is very simple to implement, but its performance is limited. The other technique is the Perfectly Matched Layer (PML), which adds complex terms to the eigenvalues to make the governing equation dissipative. Although its implementation is more difficult, its performance is greater. A broad review on artificial boundary conditions

with non-reflective properties are presented by Colonius [30].

Due to the late evolution of the DG scheme and its weak formulation, the characteristic boundary condition was not applied until 2011 by Toulopoulos & Ekaterinaris [136]. The Absorbing Layers have been analyzed in the context of DG mostly with the linearized Euler equation [3, 56]. The development of non-reflective boundary conditions are particularly of interest when turbulence is involved, because it can generate significant fluctuations near the boundaries.

1.2.4 Large-eddy Simulations

Several applied flows are in a turbulent regime. Turbulence is characterized by a large range of scales, what requires a very large grid to be able to include the extremes, what renders the simulation impractical for some applications. Cases where all the scales are captured are called Direct Numerical Simulation (DNS). When the Reynolds number that characterizes the specific problem is large enough, the range of scales is wide and three ranges can be identified: the large scales, the inertial range, and the small scales [133]. The small scales tend to be isotropic and universal, meaning that their behavior can be predicted for any problem of this kind. This concept gave birth to what is known as Large Eddy Simulations (LES), where the large scales are resolved, while the behavior of the small universal scales are modeled [104].

In order to apply the idea described for LES, the governing equations are filtered with a low-pass spatial filter. This filter creates unknown terms corresponding to the unresolved scales and need to be modeled. Sometimes the unresolved scales are called sub-grid scales (sgs), referring to the fact that the unresolved scales are usually smaller than the scales the grid is capable of resolving. This idea is linked to the discretization and the specific methodology. Therefore, the term *unresolved* is preferred over the *sgs* concept, as mentioned in [104].

The small scales are responsible of removing energy from larger scales, corresponding to the energy cascade theory. However, not always is the energy transfered to smaller scales, but in some instants the energy is transfered in the opposite direction, what is called backscattering. The most popular model to close the unresolved scales is the Smagorinsky model. Although it is attractive for its simplicities, it count with several inaccuracies, including the lack of backscattering. More sophisticated models like the Dynamic Smagorinsky may produce backscattering when the viscosity term becomes negative, but their behavior is not based on physical models.

An inconvenient that is found in the discretization of the filtered governing equation is the commutativity required between the filter and the derivatives in space and time. This problem arises in several applications with complex wall-bounded geometries, especially if high-order schemes are used. Given the severity of this issue, it is an area of active research. Note that highly accurate simulations require of highly accurate models and they must be applied together with highly accurate schemes. Otherwise, if the scheme is not accurate enough, the numerical error may be larger than the effect of the model.

Several approaches based on multi-scale methods have been under development in order to overcome the difficulties mentioned. Some examples are the Variational Multi-Scale (VMS) method by Hughes [57, 58], the Dynamic Multi-Level (DML) method by Dubois et al. [33], the rapid distortion theory (RDT) by Laval et al. [75, 76], and the Two Level Simulation (TLS) method by Kemenov and Menon [68].

VMS makes use of local spectral decomposition to split the large and small scales. This decomposition makes this method an ideal approach to be implemented with DG. DG/VMS counts with the following advantages over traditional LES methods as mentioned by Collis [28]: a priori scale separation avoids filtering problems like commutation errors; the method converges to DNS as the small scales become resolved; the method is high-order accurate; the method is independent of the grid quality; the

simulation can go from a U-RANS regime to a DNS condition passing through LES in between; the framework eases the requirements over the small-scale model, making the solution less sensitive to it; the method has a solid mathematical formulation; and the technique is highly local, like the DG method.

By making use of the spectral decomposition in VMS, the modeling of the small scales only affects the resolved small scales. For the modeling any traditional approach can be used. Even the Smagorinsky model with constant coefficient has shown very good results [13]. Nonetheless, in order to avoid setting a parameter that somewhat depends on the problem, a dynamic model, like the Dynamic Smagorinsky model could be used. In summary, the VMS framework still needs of the traditional models, but VMS improves the performance of the model in many ways.

The Dynamic Smagorinsky model with the traditional LES approach has been applied to DG solvers in [120]. By slightly rewriting the model equations carefully, the formulation becomes independent of the filter size. This is specially interesting for DG, since it avoids defining the size of the filter. The test filter needed is copied from applications of the Spectral Element Method (SEM) [81, 12, 119], since DG and SEM belong to the same FE family. The filter is applied directly to the coefficients that multiply the (hierarchical) basis. A comparison between different bases was done in [12], but no significant differences were observed, except for the fact that the Legendre basis maintains the conservation property of the scheme. Different types of filters have been used: sharp cut-off[120, 119], exponential[120], and Boyd-Vandeven[81, 12]. The sharp cut-off and exponential were compared in [120], and the former showed better results for a channel flow respect to its DNS solution. In addition, the sharp cut-off is computationally cheaper to implement.

Another dynamic model is the Localized Dynamic k-equation Model (LDKM). The main distinction between Dynamic Smagorinsky and LDKM is that the later does not neglect the unresolved kinetic energy. For this purpose, it includes an additional

equation to track the amount of kinetic energy that is not being resolved. Up to the author's knowledge, LDKM has never been used for DG.

1.3 Motivation

The subjects covered in the literature review are summarized and put in context to show what is motivating the current research. All the ideas described below are pictured in Fig. 1.

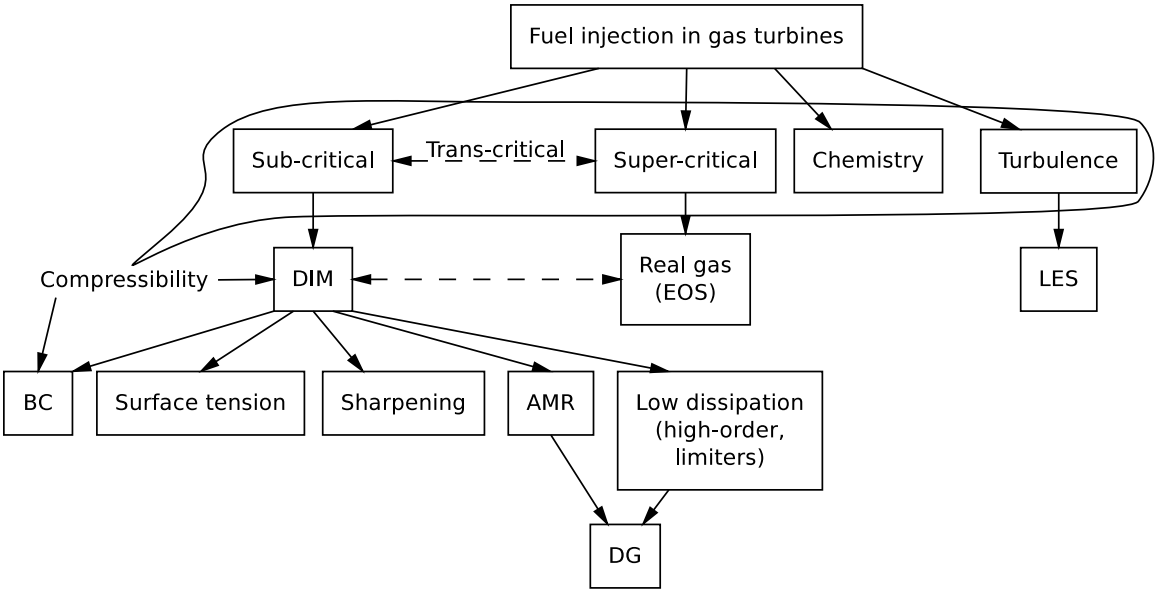


Figure 1: Diagram of the physics and numerical aspects involved in the simulation of fuel injection for gas turbines.

Modern fundamental and applied research in fields related to fluid mechanics require advanced numerical techniques that maximize the efficiency, i.e. that provide a smart balance between maximizing the accuracy and minimizing the computational time required. Fuel injection into combustors of gas turbines is an example of one of such fields. Experimental studies provide limited information and numerical techniques could complement them. However, these methods need to be improved to achieve the level of modern challenges.

At the physics level, simulation of fuel injection into combustors includes different

thermodynamic states –sub-, super-critical, a mixture of them, and even transitions–, chemistry, and turbulence. Fluctuating thermodynamic states and chemistry demand a compressible framework, since the assumption of constant density would not work in this conditions. Furthermore, acoustics are key elements in combustors and if one is interested in studying their effect, compressibility must be considered. For sub-critical simulations a compressible model is needed and, based on the literature review, a compressible DIM like the 5-eq. model is a promising option. At the same time, the framework has to be compatible with real gases (i.e. work with advanced EOSs) to be able to include the super-critical range, particularly close to the critical point. The turbulence in common combustors cover a wide range of scales impossible to resolve with the computational power available, so LES is a required approach. Limited domains, compressibility, and sophisticated governing equations tend to raise the expectations of the boundary conditions. Multi-phase simulation, particularly DIM, commonly present other 4 numerical requirements: formulation of surface tension, control of the thickness of the interface, provide an efficient way of capturing the wide range of scales in addition to LES, e.g., using AMR, and present low numerical diffusion. The last two aspects make DG a valuable framework for these types of flows. As mentioned in the literature review, DG also provides promising qualities regarding LES and, in some extent, the BCs.

DG is a new technique with very promising characteristics. However, given its short age, its growth has been mostly in theoretical contexts, while the range of applications and the experience so far is limited. Hence, it requires all the physics and numerical techniques to be studied or at least verified. For instance, limiting techniques and refinement approaches need to be computationally efficient; DG for multi-phase flows have limited literature, particularly DIM and surface tension which do not have any coverage up to the author's knowledge.

1.4 Thesis outline

This thesis is organized as follows. Chapter 2 establishes the research objectives. The physical models are covered in Chapter 3. This is followed by the description of the numerical methods used and developed in Chapter 4. A new technique to control the interface thickness is verified in Chapter 5. This is followed by the verification and validation of the Discontinuous Galerkin solver with improved techniques for AMR and limiting with emphasis on flows with discontinuities in Chapter 6. The innovative application of a Discontinuous Galerkin method to Diffused Interface Methods is shown in Chapter 7. This is followed by a new approach using the Local Discontinuous Galerkin method for sub- and super-critical flows in Chapter 8 with applications to underwater bubble collapse. This approach is extended to two-phase viscous flows with surface tension in Chapter 9 with applications to drop-wall interaction. In addition, Chapter 10 introduces the developed solver to applied aspects, like turbulence, LES, and boundary conditions, and applies the solver to more complex flows. Finally, Chapter 11 concludes the current study and it is followed by recommendations of future work in Chapter 12.

CHAPTER II

OBJECTIVES

The final goal is to develop the numerical capability required to provide insight into the nature of multiphase flows in reacting environments at different thermodynamic conditions with complex geometries. In particular, the following steps are specific objectives for the current study toward the main goal:

1. Develop a technique to control the thickness of the interface for a compressible DIM.

2. Develop an efficient DG scheme for flows dominated by discontinuities and with a large range of scales that can run on hundreds or thousands of CPUs. This includes AMR capabilities, a framework parallelized for a distributed memory computer, and limiters. The limiter is compatible with AMR. An efficient refinement criteria is determined. The efficiency of the solver tends to increase with the order of accuracy, but it is assessed if that is the case when discontinuities dominate. All the algorithms are as compact as DG in order to exploit its simple parallelization characteristics.

3. Develop a DG scheme for multi-phase flows that:

   - is compatible with complex EOSs; and
   - can model surface tension effects.

   Thus, the following new approaches are derived and implemented:

   - a technique to work with multiple EOSs simultaneously compatible with complex EOSs;

- an approach to solve a DIM with DG; and
- a method with a compact stencil to compute surface tension effects with DG.

4. Demonstrate the application of DG to turbulent flows with DNS and LES. Decaying isotropic turbulence is used for this purpose because it is a well known fundamental problem.

5. Solve a liquid jet in cross flow. One of the cases studied in the Combustion Lab for GE is used as reference.

6. Introduce and discuss a methodology to capture evaporation.

CHAPTER III

MATHEMATICAL FORMULATION

3.1 Governing equations

Below a series of related governing equations are presented. The first one is the most generic and includes the capability of resolving multiple phases with multiple components. As certain effects are neglected the equations get simplified almost like in a hierarchical way. Figure 2 shows a diagram representing the relation between them and how the simplifications evolve.

3.1.1 The multi-component multi-phase model

The model presented here resembles the Navier-Stokes equation with multi-species in the sense that there is conservation of mass of each species, momentum, and energy. However, in addition the current approach tracks each phase by convecting the volume fraction of the different phases. This is a new formulation that results from combining the the Navier-Stokes equation with multi-species and the 5-eq. model shown below. Nonetheless, the surface tension is neglected to reduce the complexity.

Let α_ξ represent the volume fraction of each phase ξ, Np the number of phases, and Ns_ξ the number of species in phase ξ. In the notation adopted in this section, ξ refers to the phase and η to species, unless explicitly stated otherwise. Also, when a variable has two indexes, the first one indicates the phase and the second one the species. The volume fraction follows this constraint:

$$\sum_{\xi=1}^{Np} \alpha_\xi = 1 \tag{1}$$

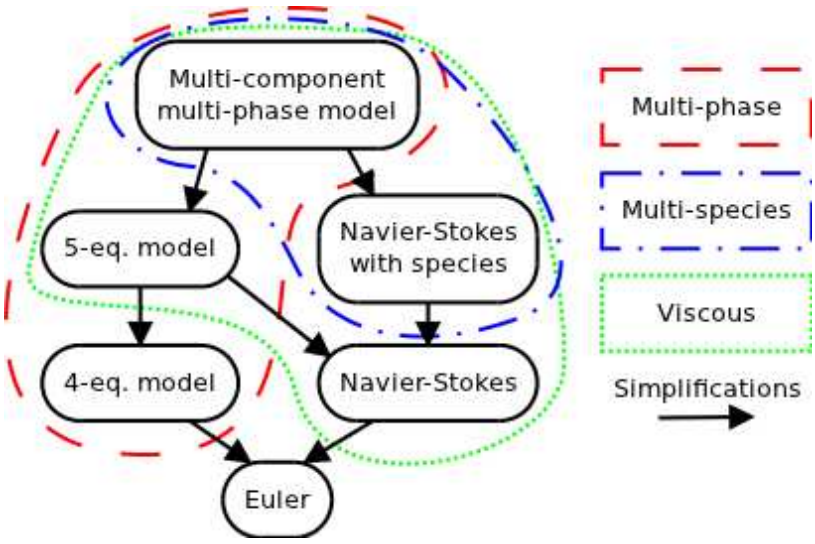


Figure 2: Diagram of the governing equations included in this study.

Also, for each phase ξ the following applies to the species of phase ξ:

$$\sum_{\eta=1}^{Ns_\xi} Y_{\xi,\eta} = 1$$

where $Y_{\xi,\eta}$ is the mass fraction of species η taking into account only the species in phase ξ. This means that:

$$Y_{\xi,\eta} = \frac{\rho_{\xi,\eta}}{\sum_{k=1}^{Ns_\xi} \rho_{\xi,k}}$$

The molar fraction, X, follows a similar rule:

$$\sum_{\eta=1}^{Ns_\xi} X_{\xi,\eta} = 1$$

and it relates to the mass fraction:

$$X_{\xi,\eta} = \frac{W_\xi}{W_\eta} Y_{\xi,\eta}$$

where W_η is the molecular weight of species η and W_ξ is the mean molecular weight of phase ξ:

$$\frac{1}{W_\xi} = \sum_{\eta=1}^{Ns_\xi} \frac{Y_{\xi,\eta}}{W_\eta}$$

or

$$W_\xi = \sum_{\eta=1}^{Ns_\xi} W_\eta X_{\xi,\eta}$$

The mixture density is defined as:

$$\rho = \sum_{\xi=1}^{Np} \alpha_\xi \rho_\xi$$

where the density of phase ξ is defined as:

$$\rho_\xi = \sum_{\eta=1}^{Ns_\xi} \rho_{\xi,\eta}$$

The term "mixture" is added to clarify that it refers to the result of all the phases present in a given unit volume.

The governing equations in 3D are given as:

$$\begin{cases} \frac{\partial \alpha_\xi}{\partial t} + v_j \frac{\partial \alpha_\xi}{\partial x_j} = \dot{m}_{\alpha_\xi} \\ \frac{\partial \rho_{\xi,\eta}\alpha_\xi}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho_{\xi,\eta}\alpha_\xi v_j + f_{\xi,\eta,j}\right] = \dot{m}_{\rho_{\xi,\eta}\alpha_\xi} \\ \frac{\partial \rho v_i}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho v_i v_j + P\delta_{ij} - \tau_{ij}\right] = \rho g_i \\ \frac{\partial \rho E}{\partial t} + \frac{\partial}{\partial x_j}\left[\left(\rho E + P\right) v_j - v_i \tau_{ij} - \lambda \frac{\partial T}{\partial x_j} + \sum_{\xi=1}^{Np}\left(\alpha_\xi \sum_{\eta=1}^{Ns_\xi} h_{\xi,\eta}\rho_{\xi,\eta}V_{j,\eta}\right)\right] = \rho v_i g_i \end{cases} \tag{2}$$

$$\text{for } i,j = 1,2,3, \quad \xi = 1,...,Np, \quad \text{and} \quad \eta = 1,...,Ns_\xi$$

where v_j is the velocity component in the x_j direction, P is the pressure, δ_{ij} is the Kronecker delta tensor, $\dot{m}$ is a reaction rate, λ is the thermal conductivity, T is the temperature, E_T is the total energy defined as $E_T = 1/2\ v_j v_j + e$, where e is the mixture specific internal energy defined as:

$$\rho e = \sum_{\xi=1}^{Np} \alpha_\xi \rho_\xi e_\xi$$

Also, g_j is the gravity component in the j^{th} direction, $V_{j,\eta}$ is a diffusion speed (defined below), and τ_{ij} is the stress tensor:

$$\tau_{ij} = \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) - \frac{2}{3}\mu\ \delta_{ij}\frac{\partial v_k}{\partial x_k}$$

where the viscosity of the mixture is computed as $\mu = \sum_{\xi=0}^{Np} \alpha_\xi \mu_\xi$. Likewise, the thermal conductivity is computed as $\lambda = \sum_{\xi=0}^{Np} \alpha_\xi \lambda_\xi$ (see [91] for the derivation).

The species diffusion is according to Fick's law:

$$X_{\xi,\eta} V_{i,\eta} = -D_\eta \frac{\partial X_{\xi,\eta}}{\partial x_i}$$

where X_η is the molar fraction of species η within phase ξ, $V_{i,\eta}$ is the diffusion velocity of η in the i^{th} direction, and D_η is the diffusion coefficient estimated as:

$$D_\eta = \frac{\mu}{\rho Sc_\eta}$$

where Sc_η is the Schmidt number of species η. In order to make sure that there is conservation of mass a correction velocity V_ξ^C needs to be added (see [94] for an explanation). Then, the diffusion flux is defined as:

$$f_{\xi,\eta,i} = \alpha_\xi \rho_{\xi,\eta} (V_{i,\eta} + V_{i,\xi}^C)$$

where

$$V_{i,\xi}^C = \sum_{\eta}^{Ns_\xi} D_\eta \frac{W_\eta}{W_\xi} \frac{\partial X_{\xi,\eta}}{\partial x_i}$$

Hence,

$$f_{\xi,\eta,i} = \alpha_\xi \left(-\rho_\xi \frac{W_\eta}{W_\xi} D_\eta \frac{\partial X_{\xi,\eta}}{\partial x_i} + \rho_{\xi,\eta} \sum_{k}^{Ns_\xi} D_\eta \frac{W_k}{W_\xi} \frac{\partial X_{\xi,\eta}}{\partial x_i} \right)$$

Molecular diffusion due to temperature gradients and heat flux due to mass fraction gradients, known as Soret and Dufour effects, respectively, are neglected here.

The mixture speed of sound is computed from:

$$\rho c^2 = \sum_{\xi=1}^{Np} \beta_\xi \rho_\xi c_\xi^2, \qquad \beta_\xi = \frac{\frac{\alpha_\xi}{\gamma_\xi - 1}}{\sum_{k=1}^{Np} \frac{\alpha_k}{\gamma_k - 1}}$$

where γ_ξ is the specific heat ratio of phase ξ.

Given that each phase is assumed a homogeneous mixture, the properties in each

phase are computed as in any multi-species formulation. Therefore, for phase ξ:

$$\begin{aligned} e_\xi &= \sum_{\eta=1}^{Ns_\xi} Y_{\xi,\eta} e_\eta \\ \lambda_\xi &= \sum_{\eta=1}^{Ns_\xi} Y_{\xi,\eta} \lambda_\eta \\ \mu_\xi &= \sum_{\eta=1}^{Ns_\xi} Y_{\xi,\eta} \mu_\eta \end{aligned} \tag{3}$$

Note that because of the constraint expressed in Eq.1, the first row of Eq.2 only needs to be solved for $Np-1$ phases. For example, the volume fraction α_{Np} can be computed as $\alpha_{Np} = 1 - \sum_{\xi=1}^{Np-1} \alpha_\xi$.

In order to render the governing equations solvable, proper boundary and initial conditions need to be defined.

The governing equation presented above applies to non-miscible phases, meaning that species diffuse within the same phase, but not into a different phase. For example, a system with water and air could be modeled as a liquid phase with H_2O, and a gas phase with N_2 and O_2, where N_2 and O_2 can diffuse into each other, but they do not diffuse into H_2O. In real systems, there could be a small concentration of N_2 or O_2 in gas phase inside the liquid phase of H_2O, but this effect is neglected in the current approach. For solubility of one phase into another in the context of absorption/desorption of a surfactant refer to [132].

A system composed of $H_2O_{(gas)}$ and $H_2O_{(liquid)}$ has two phases and one species. However, the H_2O in gas phase and the one in liquid phase are treated as two different components and do not diffuse into each other. Nonetheless, $H_2O_{(gas)}$ could become $H_2O_{(liquid)}$, and vice versa, if modeled with the reaction terms.

3.1.2 The 5-equation model

The multi-component multi-phase formulation can be reduced to the 5-eq. model [100] when only two phases are present with one component per phase. In addition, for this case it is easier to take the surface tension into account, which is a non-negligible effect in problems where breakup or coalescence is involved.

The resulted governing system converges to the formulation in [100] extended by heat conduction as presented in [91]:

$$\begin{cases} \frac{\partial \alpha_1}{\partial t} + v_j \frac{\partial \alpha_1}{\partial x_j} = 0 \\ \frac{\partial \rho_1 \alpha_1}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho_1 \alpha_1 v_j\right] = 0 \\ \frac{\partial \rho v_i}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho v_i v_j + P\delta_{ij} - \tau_{ij}\right] = -\sigma\kappa\frac{\partial \alpha_1}{\partial x_i} + \rho g_i \\ \frac{\partial \rho E}{\partial t} + \frac{\partial}{\partial x_j}\left[\left(\rho E + P\right) v_j - v_i \tau_{ij} - \lambda \frac{\partial T}{\partial x_j}\right] = -\sigma\kappa v_j \frac{\partial \alpha_1}{\partial x_j} + \rho v_i g_i \end{cases} \tag{4}$$

where σ is the constant surface tension coefficient, which is determined by the two phases in contact, and κ is the curvature defined as:

$$\kappa = -\nabla \cdot \mathbf{n} \tag{5}$$

where $\mathbf{n}$ is the normal to the interface: $\mathbf{n} = \frac{\nabla \alpha_1}{|\nabla \alpha_1|}$.

3.1.3 The 4-equation model

In the 5-eq. model the first two equations are the conservation of mass for each phase, and the addition of them leads to the total conservation of mass as in the conventional Navier-Stokes equation. In order to reduce the computational cost, these two equations can be combined under certain approximations. This combination leads to the 4-equation model [1] containing three equations for the conservation of mass, momentum, and energy, and are equivalent to the Navier-Stokes equations; the fourth equation is for the convection of the properties of the fluids. Given that the temperature cannot be recovered properly, the viscous stress and the heat diffusion are neglected.

The 4-eq model is:

$$\begin{cases} \frac{\partial \alpha_1}{\partial t} + v_j \frac{\partial \alpha_1}{\partial x_j} = 0 \\ \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho v_j\right] = 0 \\ \frac{\partial \rho v_i}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho v_i v_j + P\delta_{ij}\right] = -\sigma\kappa\frac{\partial \alpha_1}{\partial x_i} + \rho g_i \\ \frac{\partial \rho E_T}{\partial t} + \frac{\partial}{\partial x_j}\left[\left(\rho E_T + P\right) v_j\right] = -\sigma\kappa v_j \frac{\partial \alpha_1}{\partial x_j} + \rho v_i g_i \end{cases} \tag{6}$$

Even though this model is easier to implement and computationally faster than the 5-eq. model, its limitations render it only usable for simple inviscid problems.

3.1.4 Navier-Stokes

3.1.4.1 Compressible Navier-Stokes with species

The Navier-Stokes equations with species conservation is:

$$\begin{cases} \frac{\partial \rho_\eta}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho_\eta v_j + f_{\eta j}\right] = \dot{m}_\eta \\ \frac{\partial \rho v_i}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho v_i v_j + P\delta_{ij} - \tau_{ij}\right] = \rho g_i \\ \frac{\partial \rho E_T}{\partial t} + \frac{\partial}{\partial x_j}\left[\left(\rho E_T + P\right) v_j + Q_j - v_i\tau_{ij} + \sum_{\eta=1}^{Ns} h_\eta \rho_\eta V_{j\eta}\right] = \rho v_i g_i \end{cases} \tag{7}$$

The source term $\dot{m}_\eta$ represent the chemical reaction, which is defined by a reaction mechanism specific to the conditions being solved.

3.1.4.2 Compressible Navier-Stokes equations

Equation 7 can further be simplified to the traditional compressible Navier-Stokes equations when a substance with constant composition can be assumed:

$$\begin{cases} \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho v_j\right] = 0 \\ \frac{\partial \rho v_i}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho v_i v_j + P\delta_{ij} - \tau_{ij}\right] = \rho g_i \\ \frac{\partial \rho E_T}{\partial t} + \frac{\partial}{\partial x_j}\left[\left(\rho E_T + P\right) v_j - \lambda\frac{\partial T}{\partial x_j} - v_i\tau_{ij}\right] = \rho v_i g_i \end{cases} \tag{8}$$

3.1.4.3 Euler equation

Equation 8 can further be simplified to the Euler equation when the viscous terms can be neglected.

$$\begin{cases} \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho v_j\right] = 0 \\ \frac{\partial \rho v_i}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho v_i v_j + P\delta_{ij}\right] = \rho g_i \\ \frac{\partial \rho E_T}{\partial t} + \frac{\partial}{\partial x_j}\left[\left(\rho E_T + P\right) v_j\right] = \rho v_i g_i \end{cases} \tag{9}$$

3.2 Large-eddy Simulations

3.2.1 Filtered Navier-Stokes

The LES approach resolves the large-scales of the flow and models the small, unresolved, scales, such that an exact field f can be expressed as $f = \bar{f} + f''$, where $\bar{f}$ is resolved and f'' is the unresolved part. For this purpose, a spatial low-pass (in the frequency domain) filter is applied to the governing equation, Eq. 8. This filter is expressed as the convolution product between a field to be filtered f and the filter kernel G:

$$\bar{f}(\mathbf{x}, t) = \int_{\Omega} f(\mathbf{x}', t) G(\mathbf{x} - \mathbf{x}') \mathbf{dx}' \tag{10}$$

where Ω is the entire spatial domain and $\mathbf{x}'$ is a dummy variable. The filter kernel satisfies the normalization condition:

$$\int_{\Omega} G(\mathbf{x}) \mathbf{dx}' = 1 \tag{11}$$

For compressible flows it is convenient to filter conservative fields and divide them by the filtered density to obtain filtered non-conservative fields. This is called Favre filtering:

$$\tilde{f} = \frac{\overline{\rho f}}{\bar{\rho}} \tag{12}$$

Assuming that spatial and temporal differentiation commute with the filtering,

the filtered (resolved) governing equations are obtained:

$$\begin{cases} \frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_j}\left[\overline{\rho v_j}\right] = 0 \\ \frac{\partial \overline{\rho v_i}}{\partial t} + \frac{\partial}{\partial x_j}\left[\overline{\rho}\tilde{v}_i\tilde{v}_j + \bar{P}\delta_{ij} - \tilde{\tau}_{ij} + (\tilde{\tau}_{ij} - \overline{\tau}_{ij}) + \tau_{ij}^{sgs}\right] = \overline{\rho} g_i \\ \frac{\partial \overline{\rho E_T}}{\partial t} + \frac{\partial}{\partial x_j}\left[\left(\overline{\rho E_T} + \bar{P}\right)\tilde{v_j} + \tilde{Q}_j + \left(\bar{Q}_j - \tilde{Q}_j\right) - \tilde{\tau}_{ij}\tilde{v}_i + H_j^{sgs} + \sigma_j^{sgs}\right] = \overline{\rho v_i} g_i \end{cases} \tag{13}$$

where:

$$\tilde{\tau}_{ij} = \tilde{\mu}\left(\frac{\partial \tilde{v}_i}{\partial x_j} + \frac{\partial \tilde{v}_j}{\partial x_i}\right) - \frac{2}{3}\tilde{\mu}\ \delta_{ij}\frac{\partial \tilde{v}_k}{\partial x_k} \tag{14}$$

$$\tilde{Q}_j = -\bar{\lambda}\frac{\partial \tilde{T}}{\partial x_j} \tag{15}$$

where $\tilde{\mu}$ and $\tilde{\lambda}$ are estimated based on the resolved temperature $\tilde{T}$. The term $(\tilde{\tau}_{ij} - \overline{\tau}_{ij})$ originates due to the filtering of the (non-linear) viscous term τ_{ij}, and is neglected. The same applies to $\left(\bar{Q}_j - \tilde{Q}_j\right)$. The filtered total energy can be formulated as:

$$\overline{\rho E_t} = \overline{\rho e} + \frac{1}{2}\bar{\rho}\tilde{v}_i\tilde{v}_i + \bar{\rho} e_k^{sgs} \tag{16}$$

The filtered pressure is computed from the filtered equation of state:

$$\bar{P} = R\bar{\rho}\tilde{T} \tag{17}$$

and the filtered temperature from:

$$\overline{\rho e} = \bar{\rho} c_P\left(\tilde{T} - T_0\right) + \bar{\rho}\Delta h_f^0 - \bar{P} \tag{18}$$

The sgs terms are sub-grid scale terms that require closure:

$$\tau_{ij}^{sgs} = \bar{\rho}\left(\widetilde{v_i v_j} - \tilde{v}_i\tilde{v}_j\right) \tag{19}$$

$$\sigma_j^{sgs} = \tilde{v}_i\tilde{\tau}_{ij} - \overline{v_i\tau_{ij}} \tag{20}$$

$$H_j^{sgs} = \overline{\rho E_T v_j} - \overline{\rho E_T}\tilde{v}_j + \overline{P v_j} - \bar{P}\tilde{v}_j \tag{21}$$

$$e_k^{sgs} = \frac{1}{2}\left(\widetilde{v_i v_i} - \tilde{v}_i\tilde{v}_i\right) \tag{22}$$

σ_j^{sgs} is usually neglected. After all the assumptions mentioned, Eq. 13 can be expressed as:

$$\begin{cases} \frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_j}\left[\overline{\rho v_j}\right] = 0 \\ \frac{\partial \overline{\rho v_i}}{\partial t} + \frac{\partial}{\partial x_j}\left[\overline{\rho}\tilde{v}_i\tilde{v}_j + \bar{P}\delta_{ij} - \tilde{\tau}_{ij} + \tau_{ij}^{sgs}\right] = \overline{\rho} g_i \\ \frac{\partial \overline{\rho E_T}}{\partial t} + \frac{\partial}{\partial x_j}\left[\left(\overline{\rho E_T} + \bar{P}\right)\tilde{v}_j + \tilde{Q}_j - \tilde{\tau}_{ij}\tilde{v}_i + H_j^{sgs}\right] = \overline{\rho v_i} g_i \end{cases} \tag{23}$$

3.2.2 Smagorinsky model

Equation 23 has the unknown terms τ_{ij}^{sgs} and H_j^{sgs}, which need to be modeled to close the governing equations. The Smagorinsky model provides one of the simplest closures. Using an eddy viscosity concept [104], τ_{ij}^{sgs} is modeled as:

$$\tau_{ij}^{sgs} = -2\bar{\rho}\nu_t\tilde{S}_{ij} \tag{24}$$

where the resolved strain-rate is $\tilde{S}_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{v}_i}{\partial x_j} + \frac{\partial \tilde{v}_j}{\partial x_i}\right)$, and the sub-grid eddy viscosity is modeled as $\nu_t = l_s^2\bar{\mathcal{S}}$, where l_s is the Smagorinsky lengthscale and is traditionally defined as $l_s = C_s\bar{\triangle}$, $\bar{\mathcal{S}} = \sqrt{2\tilde{S}_{ij}\tilde{S}_{ij}}$ is the characteristic filtered rate of strain, C_s is a constant and $\bar{\triangle}$ is the filter size.

Assuming that the filter size is in the inertial range, the value of C_s depends on the type of filter. For instance, for a sharp spectral filter $C_s = 0.17$ [104].

Assuming that C_s is constant is very constraining for many practical cases. For example, when the model is applied in a region where the flow is laminar, the constant should be zero [104]. This kind of problem usually makes the Smagorinsky model highly diffusive. Thus, more sophisticated dynamic models were created.

3.2.3 LDKM

In the LDKM model, using an eddy viscosity concept, τ_{ij}^{sgs} is modeled as:

$$\tau_{ij}^{sgs} = -2\bar{\rho}\nu_t\left(\tilde{S}_{ij} - \frac{1}{3}\tilde{S}_{kk}\delta_{ij}\right) + \frac{2}{3}\bar{\rho}k^{sgs}\delta_{ij} \tag{25}$$

where the resolved strain-rate is $\tilde{S}_{ij} = \frac{1}{2}\left(\frac{\partial \tilde{v}_i}{\partial x_j} + \frac{\partial \tilde{v}_j}{\partial x_i}\right)$, and the sub-grid eddy viscosity ν_t is modeled as $\nu_t = (C_\nu\bar{\triangle})\sqrt{k^{sgs}}$, where $\bar{\triangle}$ is the filter size.

LDKM makes use off an additional equation to model the transport of sub-grid kinetic energy:

$$\frac{\partial \bar{\rho} k^{sgs}}{\partial t} + \frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{v}_j k^{sgs}\right) = P^{sgs} - D^{sgs} + \frac{\partial}{\partial x_j}\left(\bar{\rho}\frac{\nu_t}{\sigma_k}\frac{\partial k^{sgs}}{\partial x_j}\right) \tag{26}$$

where the right hand side terms represent the production, dissipation, and diffusion of sub-grid kinetic energy, respectively. The first two are defined as:

$$P^{sgs} = -\tau_{ij}^{sgs}\frac{\partial \tilde{v}_i}{\partial x_j} \tag{27}$$

$$D^{sgs} = (C_\epsilon/\bar{\triangle})\bar{\rho}\left(k^{sgs}\right)^{3/2} \tag{28}$$

σ_k is usually considered equal to 1. Like in other dynamic models, C_ν and C_ϵ are computed locally and dynamically. One significant advantage of LDKM respect to the previous model, is that the coefficients are dynamic, allowing for better accuracy and including effects like backscattering. Another advantage is that it LDKM does not neglect the unresolved kinetic energy, which can be significant in some applications.

Dynamic models require a second filter of size $\widehat{\triangle}$ called the test filter, which is usually defined has $\widehat{\triangle} = 2\bar{\triangle}$ and this value is adopted for the following derivations. Resolved variables do not need to be explicitly filtered, however, the test filter does need the be applied. Variables at this level are indicated as $\widehat{u}$.

We define the sub-grid kinetic energy at the test filter level as

$$k^{test} = \frac{1}{2}\left(\frac{\widehat{\bar{\rho}\tilde{v}_k\tilde{v}_k}}{\widehat{\bar{\rho}}} - \frac{\widehat{\bar{\rho}\tilde{v}_k}\widehat{\bar{\rho}\tilde{v}_k}}{\widehat{\bar{\rho}}^2}\right) \tag{29}$$

Then, the term $C_\nu\bar{\triangle}$ for ν_t is computed as:

$$C_\nu\bar{\triangle} = -\frac{L'_{ij}M'_{ij}}{4M'_{kl}M'_{kl}} \tag{30}$$

where

$$L'_{ij} = L_{ij} - \frac{2}{3}\widehat{\bar{\rho}}k^{test}\delta_{ij} \tag{31}$$

$$L_{ij} = \widehat{\bar{\rho}\tilde{v}_i\tilde{v}_j} - \frac{\widehat{\bar{\rho}\tilde{v}_i}\widehat{\bar{\rho}\tilde{v}_j}}{\widehat{\bar{\rho}}} \tag{32}$$

$$M'_{ij} = \widehat{\bar{\rho}}\sqrt{k^{test}}\left(\widehat{\tilde{S}_{ij}} - \frac{1}{3}\widehat{\tilde{S}_{kk}}\delta_{ij}\right) \tag{33}$$

Similarly, for $C_\epsilon/\bar{\triangle}$ we have:

$$C_\epsilon/\bar{\triangle} = \frac{2\left(\bar{\mu}+\mu_t\right)}{\widehat{\bar{\rho}}(k^{test})^{3/2}}\left(\widehat{\tilde{T}_{ij}\frac{\partial\tilde{v}_j}{\partial x_i}} - \widehat{\tilde{T}_{ij}}\widehat{\frac{\partial\tilde{v}_j}{\partial x_i}}\right) \tag{34}$$

where $\mu_t = \bar{\rho}\nu_t$, $\tilde{T}_{ij} = \left(\frac{\partial\tilde{v}_i}{\partial x_j} + \frac{\partial\tilde{v}_j}{\partial x_i} - \frac{2}{3}\frac{\partial\tilde{v}_k}{\partial x_k}\delta_{ij}\right)$, and $\widehat{\tilde{T}_{ij}}$ is the same tensor but applied at the test filter level. It is important to note that the filter sizes, $\bar{\triangle}$ and $\widehat{\triangle}$, are not required to be specified individually, only their ratio.

Finally, the sub-grid term in the energy equation is modeled as:

$$H_j^{sgs} = -\bar{\rho}\frac{\nu_t}{Pr_t}\frac{\partial\tilde{H}_T}{\partial x_j} \tag{35}$$

where $\tilde{H}_T = \tilde{E}_T + \bar{P}/\bar{\rho} + \tilde{v}_i\tilde{v}_i/2 + k^{sgs}$, and assuming constant c_P:

$$\frac{\partial\tilde{H}_T}{\partial x_j} = \tilde{c_P}\frac{\partial\tilde{T}}{\partial x} + \tilde{v}_i\frac{\partial\tilde{v}_i}{\partial x_j} + \frac{\partial k^{sgs}}{\partial x} \tag{36}$$

3.3 Thermodynamic closure

For single phase models, any of the Equations of State (EOS) shown below can be used. When more than one species is present in one phase mixture parameters have to be computed. This step depends on the EOS.

For diffused interface methods, the interface is diffused into a finite thickness zone, where both phases coexist, known as the artificial mixture zone. Therefore, the construction of an EOS is needed, so that it can be applied in both the pure fluid and in the mixture zone. For applications involving liquid and gas, the Stiffened gas EOS (SG EOS) [49] is generally employed to describe the thermodynamics of those pure substances. Note that the Ideal Gas EOS (IG EOS) is a special case of the SG EOS. In cases where the accuracy of the SG EOS to compute thermodynamic properties is not enough, more sophisticated formulations can be used, such as the van der Waals EOS (VDW EOS) [143] or the Peng-Robinson EOS (PR EOS) [98].

In the numerically diffused interface it can be assumed that both phases are at the same pressure (isobaric closure) or same temperature (isothermal closure), leading to two different formulations at the interface. The isobaric closure is more commonly used and consistent with the derivation of the 5-eq model [100].

The mixture variables are computed from each phase properties independently of the EOS. Hence, the mixture variables are given by:

$$\rho e = \sum_k \alpha_k \rho_k e_k(\rho_k, P) \tag{37}$$

and

$$T = \frac{\sum_k \alpha_k \rho_k c_{vk} T_k}{\sum_k \alpha_k \rho_k c_{vk}}$$

where T is the mixture temperature, T_k is the temperature of phase k and c_{Vk} is the specific heat at constant volume of phase k.

When different EOSs are used for each phase an iterative solver is needed to find the pressure resulting from the artificial mixture zone. Allaire et al. [1] presented an efficient method that relies on the following three assumptions for each phase k: $P_k > 0$, $P_k = 0$ if $\rho_k e_k = 0$, and $\frac{\partial \rho_k e_k}{\partial P_k}|_{\rho_k} > 0$. Under certain circumstances these requirements may present some limitations. For instance, water violates the third rule between 0 and 4^oC, and the second rule may not always be satisfied by real gases or for common enthalpies of formation. Therefore, given the isobaric closure ($P_1 = P_2$), we suggest to solve Eq. 37 where the pressure P is the variable we need to solve for. For this purpose we use a Newton-Raphson method. The numerical function to solve and its derivative are:

$$g(P) = \alpha_1 \rho_1 e_1(\rho_1, P) + \alpha_2 \rho_2 e_2(\rho_2, P) - \rho e$$

$$\frac{\partial g(P)}{\partial P} = \alpha_1 \rho_1 c_{v,1} \frac{\partial T_1}{\partial P}|_\rho + \alpha_2 \rho_2 c_{v,2} \frac{\partial T_2}{\partial P}|_\rho$$

Given that $\frac{\partial g(P)}{\partial P} > 0$, $g(P) = 0$ has one and only one solution.

3.3.1 Ideal gas

The ideal gas assumption leads to the widely used equation $P/\rho_k = R_k T_k$, where $R_k = R_u/W_k$ is the gas constant for phase k, R_u is the universal gas constant per mole ($R_u = 8.314\ J/(mol\ K)$), and W_k is its molecular weight. The caloric closure is:

$$h_k = \int_{T_0}^{T} c^0_{P,k} dT + \Delta h^0_{f,k} \tag{38}$$

where $c^0_{p,k}$ is the specific heat at constant pressure and Δh^0_f is the enthalpy of formation at standard conditions ($P_0 = 101325\ Pa$ and $T_0 = 298.15\ K$). The super index 0 refers to a low pressure limit (where pressure effects are negligible). The energy can be computed from the enthalpy: $e_k = h_k - P/\rho_k$. The specific heat could be assumed constant, but a better approximation, especially for a wide range of temperatures, is to extract it from some common database based on curve-fitted data. For variable c^0_P we use the JANAF tables [19].

3.3.2 Stiffened gas

Massoni et al. [91] presented a thorough derivation of the required closures making use of the Stiffened Gas (SG) and the Mie-Gruneisen formulation. The SG EOS can be adjusted to specific conditions, so its accuracy is reduced to a narrow range. Nonetheless, it is widely used to model compressible liquids due to its simplicity and robustness.

The energy of phase k can be computed as:

$$e_k = \frac{P + \gamma_k P_{\infty k}}{\rho_k (\gamma_k - 1)} + c_{0k} \tag{39}$$

where γ_k and $P_{\infty k}$ are constant parameters that depend on each fluid, and

$$c_{0k} = \Delta h^0_{f,k} - \gamma_k \frac{P_0 + P_{\infty k}}{\rho_{0k} (\gamma_k - 1)} \tag{40}$$

where ρ_{0k} is the density of phase k at standard conditions. Note that if $P_{\infty k}$ is equal to zero, Eq. 39 reduces to the ideal gas EOS.

The temperature of each phase is given by:

$$T_k = \frac{P + P_{\infty k}}{c_{vk}\rho_k\left(\gamma_k - 1\right)} \tag{41}$$

where c_{vk} is the specific heat at constant volume for phase k and is assumed to be constant.

3.3.3 Van der Waals

The VDW EOS can be expressed as:

$$\left(P + a_k/v_{m,k}^2\right)\left(v_{m,k} - b_k\right) = R_u T_k \tag{42}$$

where $v_{m,k}$ is the molar volume ($v_{m,k} = W_k/\rho_k$), and

$$a_k = \frac{27(R_u T_{c,k})^2}{64 P_{c,k}} \qquad\qquad b_k = \frac{R_u T_{c,k}}{8 P_{c,k}}$$

$T_{c,k}$ and $P_{c,k}$ are the critical temperature and pressure of phase k.

The enthalpy is defined as:

$$h_k = \int_{T_0}^{T} c_{p,k}^0 dT + \Delta h_{f,k}^0 + \Delta h_k \tag{43}$$

where c_p^0 is the specific heat at constant pressure for a low pressure limit, and Δh is the departure function that takes into account high-density effects:

$$\Delta h_k = \left(P\ v_{m,k} - R_u\ T - a/v_{m,k}\right)/W_k \tag{44}$$

Finally, the internal energy is $e_k = h_k - P/\rho_k$.

3.3.4 Peng-Robinson

The traditional Peng-Robinson EOS (PR EOS) has been shown to be a good representations of fluid behavior at large values of the "reduced" conditions (subscript r) defined by $P_r = P/P_c$ and $T_r = T/T_c$. The PR EOS is:

$$P = \frac{R_u T_k}{v_{m,k} - B_k} - \frac{A_k\ \Theta(T)}{v_{m,k}^2 + 2 v_{m,k} B_k + B_k^2} \tag{45}$$

where

$$A_k = 0.457236\frac{R_u^2 T_{c,k}^2}{P_{c,k}} \qquad B_k = 0.077796\frac{R_u T_{c,k}}{P_{c,k}}$$

$$\Theta(T) = \left[1 + f(\omega_k)\left(1 - \sqrt{T/T_{c,k}}\right)\right]^2$$

$$f(\omega_k) = 0.379642 + 1.48503\omega_k - 0.164423\omega_k + 0.016666\omega_k$$

where ω_k is the acentric factor and depends on the species in phase k (we assume one species per phase). Eq. 43 still applies, but the departure function is computed as [50]:

$$\Delta h_k = \left[P\ v_{m,k} - R_u\ T_k + K_1\ A_k\left(\Theta(T_k) - T_k\frac{\partial\Theta(T_k)}{\partial T_k}\right)\right]/W_k \tag{46}$$

CHAPTER IV

NUMERICAL METHOD

4.1 Interface Sharpening

4.1.1 The source terms

The sharpening method treated here is applied through source terms in every equation. These terms should not be considered part of the governing equations because the sharpening technique is a numerical artifact to compensate the numerical diffusion of the scheme. The same way as limiters in TVD schemes affect the solution and are numerical artifacts not taking part in the governing equations.

Nonetheless, the source terms do not affect the analytical ideal solution where contact discontinuities are present. In this case the volume fraction would be zero or one and the added source term is always zero under either condition.

The core of the sharpening terms is a function $f(\alpha)$, defined below, which is applied to the numerical scheme as part of a source term in each differential equation:

$$\begin{aligned}
\frac{\partial \alpha_1}{\partial t} + u_j \frac{\partial \alpha_1}{\partial x_j} &= f(\alpha_1) \\
\frac{\partial \rho_1 \alpha_1}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho_1 \alpha_1 u_j\right] &= f(\alpha_1)\rho_1 \\
\frac{\partial \rho_2 \alpha_2}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho_2 \alpha_2 u_j\right] &= f(\alpha_1)\rho_2 \\
\frac{\partial \rho u_i}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho u_i u_j + P\delta_{ij} - \tau_{ij}\right] &= -\sigma\kappa\frac{\partial \alpha_1}{\partial x_i} + \rho g_i + f(\alpha_1) u_i \left(\rho_1 - \rho_2\right) \\
\frac{\partial \rho E}{\partial t} + \frac{\partial}{\partial x_j}\left[\left(\rho E + P\right) u_j - u_i \tau_{ij} - \lambda \frac{\partial T}{\partial x_j}\right] &= -\sigma\kappa u_j \frac{\partial \alpha_1}{\partial x_j} + \rho u_i g_i + f(\alpha_1)\left(\rho_1 E_1 - \rho_2 E_2\right)
\end{aligned} \tag{47}$$

These additional terms were derived neglecting the change of ρ_1 and ρ_2 at the interface. Note that making use of this sharpening approach the conservation property is lost.

4.1.2 The sharpening function

Similarly to what Kokh[70] suggested for the 4-equation model[1], the core of the sharpening terms is defined as:

$$f(\alpha_1) = \eta\alpha_1(1-\alpha_1)(\alpha_1 - 0.5) \tag{48}$$

In Kokh's paper[70] constant values of η are used among the domain, but it varies for different problems and schemes. Below a defined function for η is suggested to avoid problem-dependent calibrations.

The effect of $f(\alpha)$ can easily be seen in the convection of the volume fraction equation and Fig.3. Assuming that η is positive, the source term is greater than zero where the volume fraction is between 0.5 and 1, making the volume fraction tend to 1. The opposite effect happens where α_1 is between 0 and 0.5. Given this it can be concluded that the sharpening strength is dictated by the value of η.

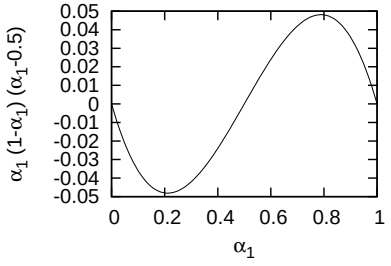


Figure 3: $f(\alpha)/\eta$ vs. α.

4.1.3 One-dimensional analysis

As it was said above, the purpose of the sharpening method is to compensate the diffusivity of the numerical scheme. Hence, the behavior of the scheme has to be understood to properly adjust the strength of the sharpening source terms.

The diffusivity of a scheme can be analyzed deriving the Modified Difference Equation (MDE). The convection of the volume fraction is governed by the linear wave

equation with variable coefficients. A possible scheme to solve it is the First-Order Upwind scheme (first order in time and space), which in 1-D leads to the following MDE:

$$\frac{\partial \alpha_1}{\partial t} + u\frac{\partial \alpha_1}{\partial x} = \beta \alpha_{1,xx} + \dots \tag{49}$$

where $\beta = \frac{u\Delta x}{2}(1-\nu)$ and $\nu = \frac{u\Delta t}{\Delta x}$. The numerical diffusivity is governed by the term with the lowest even order derivative on the right hand side of the MDE, it is scheme-dependent and unfortunately it can not easily be derived for every scheme.

To simplify the analysis, let's assume that the scheme used follows Eq. 49. If the sharpening term is included in the differential equation, the MDE becomes:

$$\frac{\partial \alpha_1}{\partial t} + u\frac{\partial \alpha_1}{\partial x} = f(\alpha_1) + \beta \alpha_{1,xx} + \dots \tag{50}$$

In order to cancel or control the diffusivity the source terms should balance the second term on the right hand side, what means that, ignoring higher order terms, $f(\alpha_1) + \beta\ \alpha_{1,xx}$ should be as close as possible to zero. An analytical approach could require finding the value of η that minimizes the L^2 norm of $f(\alpha_1) + \beta\ \alpha_{1,xx}$. Let's approximate the volume fraction at a given time as an hyperbolic tangent:

$$\alpha_1 = \frac{1}{2}\left[1 + \tanh\left(K(x - x_0)\right)\right] \tag{51}$$

Its first derivative respect to x is:

$$\alpha_{1,x} = 2K\alpha_1(1-\alpha_1) \tag{52}$$

and its maximum is equal to:

$$\alpha_{1,x\ MAX} = K/2 \tag{53}$$

The second derivative respect to x is:

$$\alpha_{1,xx} = 4K^2\alpha_1(1-\alpha_1)(1-2\alpha_1) \tag{54}$$

The L^2 norm is defined as:

$$L^2 = \int_{-\infty}^{\infty} \left[f(\alpha_1) + \beta\alpha_{1,xx}\right]^2 dx \tag{55}$$

The minimum of L^2 is where $\frac{\partial L^2}{\partial \eta} = 0$ and $\frac{\partial^2 L^2}{\partial \eta^2} > 0$. Given that the integration boundaries in 55 are independent of η:

$$\frac{\partial L^2}{\partial \eta} = \int_{-\infty}^{\infty} 2\left[f(\alpha_1) + \beta\alpha_{1,xx}\right] \frac{\partial f(\alpha_1)}{\partial \eta} dx \tag{56}$$

Applying a change of variables the integration becomes:

$$\frac{\partial L^2}{\partial \eta} = \int_{\alpha_1(-\infty)}^{\alpha_1(\infty)} 2\left[f(\alpha_1) + \beta\alpha_{1,xx}\right] \frac{\partial f(\alpha_1)}{\partial \eta} \frac{dx}{d\alpha_1} d\alpha_1 \tag{57}$$

which has an analytical solution:

$$\frac{\partial L^2}{\partial \eta} = \frac{\eta}{120K} - \frac{\beta K}{15} \tag{58}$$

The minimum is where $\frac{\partial L^2}{\partial \eta} = 0$, thus:

$$\eta_{opt} = 8\beta K^2 \tag{59}$$

The second derivative is $\frac{\partial^2 L^2}{\partial \eta^2} = 1/\left(120K\right)$, which is always greater than zero, showing that the solution corresponds to a minimum and not a maximum.

The value of K, as shown in Eq. 51, specifies the thickness of the interface. The volume fraction tends asymptotically to 0 and 1, so some criteria has to be specified in order to define the interface thickness. One option is to set two limits: $\alpha_a = \phi$ and $\alpha_b = 1 - \phi$, where ϕ is a positive number close to zero. Let's define the thickness as $d = x_b - x_a$, where $\alpha_1(x_a) = \alpha_a$ and $\alpha_1(x_b) = \alpha_b$. This means that K can be estimated as:

$$K = \frac{ln\left(1/\phi - 1\right)}{d} \tag{60}$$

The thickness should be specified by a number of cells n. If it is assumed that the grid is uniform, the following possible definition can be used:

$$d = n \cdot \Delta x \tag{61}$$

Inserting this into Eq. 60:

$$K = \frac{ln\left(1/\phi - 1\right)}{n \cdot \Delta x} \tag{62}$$

Given that the thickness is something that one wants to specify in advance, n and ϕ are constant parameters. For example, n could be in the order of 8 and ϕ approximately 0.01. Let's rewrite Eq. 62 as:

$$K = \frac{\varphi}{\Delta x} \tag{63}$$

where φ could be a constant of approximately 0.6. Using Eq. 53 we find that $\frac{\varphi}{\Delta x} = 2\alpha_{1,x\ MAX}$. Inserting Eq. 63 into Eq. 59:

$$\eta_{opt} = 8\varphi^2 \frac{\beta}{\Delta x^2} \tag{64}$$

Finally, for the Upwind scheme this could be written as:

$$\eta_{opt,upwind} = 4\varphi^2 \frac{u}{\Delta x}(1-\nu) \tag{65}$$

4.1.4 Extension to two dimensions

For a proper formulation of a multidimensional scheme, the MDE has to be formulated accordingly. A new factor present here and not in the one dimensional case is the curvature. The derivation is not included for the sake of brevity. In two dimensions, Eq. 59 is replaced by:

$$\eta_{opt} = 8\beta_1 K^2 - 5\beta_2 \kappa^2 \tag{66}$$

where:

$$\beta_1 = 1/2\ \left[\left(u\ \Delta x - \Delta t\ u^2\right) n_x^2 + \left(v\ \Delta y - \Delta t\ v^2\right) n_y^2 - 2\ \Delta t\ u\ v\ n_x\ n_y\right] \tag{67}$$

$$\beta_2 = 1/2\ \left[\left(u\ \Delta x - \Delta t\ u^2\right) t_x^2 + \left(v\ \Delta y - \Delta t\ v^2\right) t_y^2 - 2\ \Delta t\ u\ v\ t_x\ t_y\right] \tag{68}$$

n_i is the component of the normal vector to the interface on the i^{th} axis and the same applies to the tangent vector components t_x and t_y. They can be computed as: $t_x = (1 - n_x^2)^{1/2}$, if t_x is zero then $t_y = \pm 1$, otherwise, $t_y = t_x\ n_x/n_y$.

Moreover, Δx in Eq. 61 can be replaced by $(n_x\ \Delta x^2 + n_y\ \Delta y^2)^{1/2}$, such that Eq. 66 becomes:

$$\eta_{opt} = 8\frac{\beta_1}{n_x\ \Delta x^2 + n_y\ \Delta y^2} - 5\beta_2 \kappa^2 \tag{69}$$

4.2 The Local Discontinuous Galerkin

4.2.1 Discretization

The Local Discontinuous Galerkin (LDG) method is used to solve the conservation law governing equations with viscous and source terms. For this purpose the governing equation can only contain first order derivatives. Therefore, the governing equation is written as:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot [\mathbf{F}(\mathbf{u}) - \mathbf{F_V}(\mathbf{u}, \mathbf{q})] = \mathbf{S}, \quad \text{for} \quad t > 0 \\ \mathbf{q} = \nabla \mathbf{f}(\mathbf{u}), \quad \text{for} \quad t \geq 0 \\ \mathbf{u} = \mathbf{u}_0, \quad \text{for} \quad t = 0 \end{cases} \tag{70}$$

where $\mathbf{u}$ is the state vector, $\mathbf{F}$ is the inviscid flux, $\mathbf{F_V}$ is the viscous flux, $\mathbf{S}$ is the source term, $\mathbf{q}$ is the auxiliary vector, $\mathbf{f}(\mathbf{u})$ is some function of the state vector (e.g. temperature, velocity, etc.), and $\mathbf{u}_0$ is the initial value. The addition of $\mathbf{q}$ to the governing equation is to reduce the order of the derivatives in the equations to be discretized, as required by DG methods.

These equations are discretized in a domain Ω divided into N non-overlapping elements:

$$\Omega = \bigcup_{j=1}^{N} \Omega_j \tag{71}$$

An inner product is done between the governing equation and a test function $\mathbf{w}$. The solution vector $\mathbf{u}$ is approximated per element by $\mathbf{U}_j$ defined by the basis ϕ:

$$\mathbf{U}_j = \sum_{i=0}^{(p+1)^d - 1} \phi_i \mathbf{c}_{i,j} \tag{72}$$

where p indicates the order of the approximation (not the order of accuracy). Likewise, the derivatives $\mathbf{q}$ are approximated by $\mathbf{q}_j$:

$$\mathbf{q}_j = \sum_{i=0}^{(p+1)^d - 1} \phi_i \mathbf{d}_{i,j} \tag{73}$$

The same basis ϕ is used for the test function $\mathbf{w}$, what leads to a Galerkin method. Therefore, after applying the Divergence Theorem, the approach consists of finding

$\mathbf{U}_j$ such that for every element j:

$$\begin{cases} \int_{\Omega_j} \phi \, \frac{\partial \mathbf{U}_j}{\partial t} \, dV - \int_{\Omega_j} \nabla\phi \cdot [\mathbf{F}(\mathbf{U}_j) - \mathbf{F_V}(\mathbf{U}_j)] \, dV \\ \qquad + \int_{\partial\Omega_j} \phi \, \left[\widehat{F}(\mathbf{U}^-, \mathbf{U}^+) - \widehat{F_V}(\mathbf{U}^-, \mathbf{U}^+, \mathbf{q}^-, \mathbf{q}^+) \right] dS \\ \qquad = \int_{\Omega_j} \phi \mathbf{S}(\mathbf{U}_j) \, dV, \quad \text{for} \quad t > 0 \\ \int_{\Omega_j} \phi \, \mathbf{q}_j \, dV + \int_{\Omega_j} \nabla\phi \cdot \mathbf{f}(\mathbf{U}_j) dV - \int_{\partial\Omega_j} \phi \, \widehat{U}(\mathbf{U}^-, \mathbf{U}^+) dS = 0, \quad \text{for} \quad t \geq 0 \\ \int_{\Omega_j} \phi \, \mathbf{U}_j \, dV = \int_{\Omega_j} \phi \, \mathbf{u}_0 \, dV, \quad \text{for} \quad t = 0 \end{cases} \tag{74}$$

$\widehat{F}$ is a numerical flux normal to the boundary of the element that needs to be properly defined given that is computed at the face of the elements, which may be discontinuous. $\mathbf{U}^-$ is the value of $\mathbf{U}_j$ according to the current element j at the face and $\mathbf{U}^+$ is the value of $\mathbf{U}_k$ at the face based on the neighboring element k. The same applies to $\mathbf{q}^-$ and $\mathbf{q}^+$. $\widehat{U}$ and $\widehat{F_V}$ are numerical fluxes defined according to the LDG method [24]. In this study the spatial integration is done with a quadrature rule using Lobatto points. Figure 4 depicts a 2-D, $p = 4$ element. For more details refer to [23, 22, 21, 25].

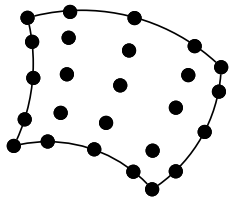


Figure 4: Diagram of a 2-D, $p = 4$ element showing its nodes.

The numerical fluxes used in this report for the conservation of mass, momentum, and energy are the Local Lax-Friedrich flux and the HLLC approximate Riemann solver [135].

4.2.2 The Discontinuous Galerkin method for multi-fluid models

Diffuse interface methods, similar to many other multi-phase techniques, requires to solve the convection of the volume fraction, which mathematical representation is a

particular case of the Hamilton-Jacobi equation. Two successful approaches can be found in the literature and are described below.

4.2.2.1 A Roe-type flux for the volume fraction equation

[20] showed that the convection of volume fraction can be solved in the pseudo-conservative form:

$$\frac{\partial \alpha}{\partial t} + \frac{\partial \alpha u}{\partial x} = \alpha \frac{\partial u}{\partial x} \tag{75}$$

The right hand side is a source term and the second term on the left hand side is the inviscid term. This last term requires a numerical flux as explained above and [20] suggested using:

$$\widehat{F_\alpha}(\mathbf{U}^-, \mathbf{U}^+) = v_n^- \alpha^- + \frac{1}{2}\left[\min(v_n^-, v_n^+) - |\min(v_n^-, v_n^+)|\right](\alpha^+ - \alpha^-) \tag{76}$$

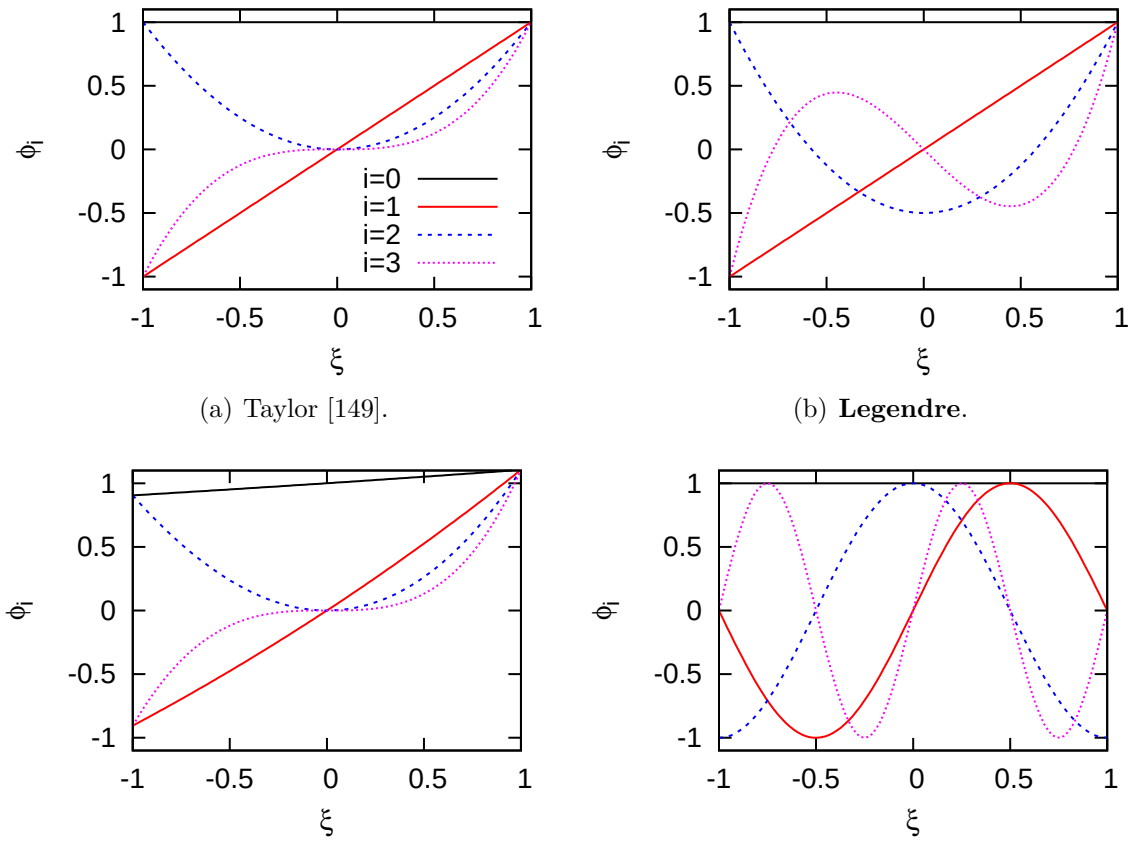
where v_n is the normal velocity, $v_n = \mathbf{v} \cdot \mathbf{n}$. This Roe-type flux is appropriate as long as there is no entropy violation. In case there was one, a special treatment would be needed. In the cases treated in this report, no entropy violation was observed, so no additional procedure was applied.

For more detail on how to apply this method to the generic Hamilton-Jacobi equation see [20].

4.2.2.2 The Local Discontinuous Galerkin method for the volume fraction equation

Another method to solve the Hamilton-Jacobi equation was presented by [151]. Even though this method requires an auxiliary variable, it has the advantage that it cannot violate any entropy constrain. In this case, the gradient of α has two numerical definitions: $\mathbf{q}_{\alpha m} = \nabla\alpha$ and $\mathbf{q}_{\alpha p} = \nabla\alpha$. For $\mathbf{q}_{\alpha m}$ the element on the left hand side of the face is used when computing $\widehat{f}$, whereas for $\mathbf{q}_{\alpha p}$ the element on the right hand side is used. Then $\mathbf{S}_\alpha$ is approximated by $\widehat{\mathbf{S}}_\alpha$ and it is computed based on a combination of $\mathbf{q}_{\alpha m}$ and $\mathbf{q}_{\alpha p}$:

$$\widehat{\mathbf{S}}_\alpha = -\frac{1}{2}\left[(\mathbf{v} - \mathbf{v_M})\,\mathbf{q}_{\alpha p} + (\mathbf{v} + \mathbf{v_M})\,\mathbf{q}_{\alpha m}\right] \tag{77}$$



(a) Taylor [149].

(b) **Legendre**.

(c) Exponential [158].

(d) Trigonometric.

Figure 5: Example of 1-D bases.

where $\mathbf{v_M}$ is the maximum absolute velocity (in each direction) within the element.

4.2.3 DG bases

Several options can be used to form the finite element space. Figure 5 show examples of hierarchical bases in 1D up to order 3. The most common hierarchical basis consists of Legendre polynomials. The basis that better represents the solution tends to be more accurate, as shown in [158], where exponential terms were added. Monomials have also been used given that they are very easy to construct [149]. However, they are not orthogonal. All these bases are usually referred to as modal bases.

Lagrange polynomials have also been widely used, which constitute a non-hierarchical

basis. When these polynomials are used, the scheme is usually referred to as nodal-DG, since the nodal values are equal to the coefficients that form the solution. The main advantage of Lagrange polynomials is obtained when computing the state vector at a face, because the result is independent of the nodal values not on that face. Thus, the computation can be faster. Nonetheless, most limiters and filters are done based on hierarchical bases, especially with Legendre polynomials, so the transformation between nodal and modal spaces is required, what has some computational cost.

4.2.3.1 Legendre polynomials

Legendre polynomials P_i are a common choice because it is a polynomial (instead of other computationally more expensive functions like trigonometric or exponentials), orthogonal and hierarchical. This leads to several numerical advantages, like a lower condition number of the Vandermond matrix. For instance, if the Jacobian is constant inside the element the mass matrix is diagonal. In addition, if the basis is correctly normalized and the Jacobian is constant, the mass matrix is equal to the identity matrix. Finally, when the computational speed is important, one should keep in mind that polynomial expressions are faster to compute than other kind of functions like trigonometric or exponentials.

Here the normalized Legendre polynomials are used:

$$\phi_i = P_i \sqrt{\frac{2i+1}{2}} \quad \text{for} \quad i = 0, ..., p \tag{78}$$

Thus the basis in 1-D up to 6^{th} order is:

$$\phi_0(\xi) = \frac{1}{\sqrt{2}}$$
$$\phi_1(\xi) = \sqrt{\frac{3}{2}}\xi$$
$$\phi_2(\xi) = \sqrt{\frac{5}{2}}\frac{3\xi^2 - 1}{2}$$
$$\phi_3(\xi) = \sqrt{\frac{7}{2}}\frac{5\xi^3 - 3\xi}{2}$$
$$\phi_4(\xi) = \frac{3(35\xi^4 - 30\xi^2 + 3)}{8\sqrt{2}}$$
$$\phi_5(\xi) = \sqrt{\frac{11}{2}}\frac{(63\xi^5 - 70\xi^3 + 15\xi)}{8}$$
$$\phi_6(\xi) = \sqrt{\frac{13}{2}}\frac{(231\xi^6 - 315\xi^4 + 105\xi^2 - 5)}{16}$$

4.2.3.2 Averaged-derivatives

A different polynomial basis can be derived such that the coefficients become the element-averaged value, the element-averaged first derivative, the element-averaged second derivative, and so on. Having these values explicitly can be convenient, however the resulting mass matrix is not diagonal given that the basis is not orthogonal.

The basis in 1-D up to 6^{th} order is:

$$\psi_0(\xi) = 1$$
$$\psi_1(\xi) = \xi$$
$$\psi_2(\xi) = \frac{3\xi^2 - 1}{6}$$
$$\psi_3(\xi) = \frac{(\xi^2 - 1)\xi}{6}$$
$$\psi_4(\xi) = \frac{(15\xi^4 - 30\xi^2 + 7)}{360}$$
$$\psi_5(\xi) = \frac{(3\xi^4 - 10\xi^2 + 7)\xi}{360}$$
$$\psi_6(\xi) = \frac{(21\xi^6 - 105\xi^4 + 147\xi^2 - 31)}{15120}$$

It is important to remark that the element-averaged derivatives are respect to ξ,

the element-internal coordinate, where $\xi \in [-1, 1]$. Thus, to estimate the i^{th} derivative respect to the physical coordinate x, the coefficient has to be multiplied by $(2/h_j)^i$, where h_j is the length of the element:

$$\mathbf{U}_j = \sum_{i=0}^{p} \psi_i \mathbf{d}_{i,j} \tag{79}$$

and

$$\bar{\mathbf{U}}_j^{(i)} = \mathbf{d}_{i,j} \left(\frac{2}{h_j} \right)^i \tag{80}$$

4.2.3.3 2 and 3 dimensions

For quadrilateral and hexahedral elements the basis is generated from the 1-D basis applying a tensor product. In 2-D:

$$\phi_{ij}(\xi, \eta) = \phi_i(\xi)\ \phi_j(\eta)$$
$$\psi_{ij}(\xi, \eta) = \psi_i(\xi)\ \psi_j(\eta)$$

and in 3-D:

$$\phi_{ijk}(\xi, \eta, \zeta) = \phi_i(\xi)\ \phi_j(\eta)\ \phi_k(\zeta)$$
$$\psi_{ijk}(\xi, \eta, \zeta) = \psi_i(\xi)\ \psi_j(\eta)\ \psi_k(\zeta)$$

4.2.4 Time integration

The time integration is conducted explicitly using the Runge-Kutta (RK) method or the Spectral Deferred Correction (SDC) method. Both approaches treat the governing equations as a system of ordinary differential equations (ODE):

$$\frac{d\mathbf{u}}{dt} = G(t, \mathbf{u}) \tag{81}$$

where $G(t, \mathbf{u})$ includes all the terms in the governing equations except for the time derivative. In this study, unless specified otherwise, for elements of polynomial order

p a time integration of order $p+1$ is used. Unless specified otherwise, the time-step size $\triangle t$ is computed at every time step as:

$$\triangle t = C\left(1/\triangle t_c + 1/\triangle t_v\right)^{-1} \tag{82}$$

where C is a constant to be on the safe size of the stability criteria (here, $C = 1.0$ is used unless mentioned otherwise), and $\triangle t_c$ and $\triangle t_v$ are the convective and viscous time steps, respectively:

$$\triangle t_{c1} = \frac{1}{\||\mathbf{v}| + c\|_\infty (2p+1)\sum_j 1/\Delta x_j} \tag{83}$$

$$\triangle t_v = \min_j \left(\frac{\triangle x^2}{4p^4 \upsilon/\rho}\right)_j \tag{84}$$

where $\triangle x_j$ is the length of the element in the j direction and $\upsilon = \max(\mu, \lambda/c_p)/\rho$. Even though, Eq. 83 is required for stability, for positivity-preserving conditions, a more stringent rule is needed:

$$\triangle t_{c2} = \frac{w_1/2}{\||\mathbf{v}| + c\|_\infty \sum_j 1/\Delta x_j} \tag{85}$$

where w_1 is the weight corresponding to the first Lobatto point in the quadrature rule on $[-1, 1]$. $\triangle t_{c1}$ is used by default. Based on experience, $\triangle t_{c2}$ is only required for multi-phase flows with surface tension. The ratio $\triangle t_{c1}/\triangle t_{c2}$ depends on the number of quadrature points used, which determine w_1. If The number of quadrature points is $2(p+1)$, then $\triangle t_{c1}/\triangle t_{c2} = 2(p+1)$.

The stability criteria for the convective time step for RKDG has been widely studied in the literature [26]. However, neither the stability criteria for the viscous time step, nor how to combine the viscous and convective limits have been thoroughly researched. The cause for this is that the conditions change from one numerical scheme to the other and the limits found are usually an upper limit, so in practice there is always some uncertainty. We analyze this stability criteria for a convection-diffusion problem in Section 10.4.1. The same stability limits as for RKDG are used

for SDC-DG and resulted to provide a stable condition for the tests presented in this study.

4.2.4.1 The Runge-Kutta method

The Runge-Kutta method is a well known family of schemes. In the current study the Total Variation Diminishing RK (TVD-RK) of second and third order are used [122], and can be summarized in the following three steps:

Step 1:

$$u^0 = u_n \tag{86}$$

Step 2:

$$u^i = \sum_{l=0}^{i-1} \alpha_{il} w^{il}, \quad w^{il} = u^l + \frac{\beta_{il}}{\alpha_{il}} \Delta t G\left(u^l, t + \Delta t \cdot d_l\right), \qquad \text{for } i = 1, ..., K \tag{87}$$

Step 3:

$$u_{n+1} = u^K \tag{88}$$

where the super-indexes of u determine a intermediate steps between u_n and u_{n+1}. The parameters are given in Table 1. A good property for these 2^{nd}- and 3^{rd}-order TVD schemes is that if for some semi-norm $|\cdot|$, we have that $|w^{il}| \leq |u^l|$, then $|u_{n+1}| \leq |u_n|$.

Table 1: Parameters for TVD-RK of order 2 and 3

Order	α_{il}	β_{il}	d_l
2	1	1	0
	1/2 1/2	0 1/2	1
3	1	1	0
	3/4 1/4	0 1/4	1
	1/3 0 2/3	0 0 2/3	1/2

4.2.4.2 The spectral deferred correction method

Although details of this method are given elsewhere [35, 93, 84, 148], the main algorithm is included for completeness. The scheme is based on first-order explicit integration of sub-steps and iterative correction [35]. For stiff problems, the scheme can be varied with a more implicit character, but only the explicit method is addressed here. Each time step $[t_n, t_{n+1}]$ is divided into J sub-steps: $t_n = t_{n,0} < t_{n,1} < \ldots < t_{n,m} < t_{n,m+1} < \ldots < t_{n,J} = t_{n+1}$. These points are chosen as quadrature points (Lobatto points in the current study). This approach makes the scheme more stable because it avoids a uniform distribution and leads to the spectral characteristic of the scheme [35]. This property is important to stabilize higher orders. Initially, the governing equations are integrated with a first-order explicit integration from t_n to t_{n+1} using $t_{n,m}$ points:

$$u^1_{n,m+1} = u^1_{n,m} + \Delta t_{n,m} G\left(t_{n,m}, u^1_{n,m}\right) \quad \text{for } m = 0, ..., J-1 \tag{89}$$

where $u^1_{n,0} = u_n$ and $\Delta t_{n,m} = t_{n,m+1} - t_{n,m}$.

Now K iterations are computed for $k = 1, \ldots, K$ and $m = 0, ..., J-1$ (m is the inner loop):

$$u^{k+1}_{n,m+1} = u^{k+1}_{n,m} + \theta \Delta t_{n,m} \left(G\left(t_{n,m}, u^{k+1}_{n,m}\right) - G\left(t_{n,m}, u^{k}_{n,m}\right)\right) + I^{m+1}_m \left(G\left(t_{n,m}, u_{n,m}\right)\right) \tag{90}$$

where $0 \leq \theta \leq 1$ and $I^{m+1}_m \left(G\left(t_{n,m}, u_{n,m}\right)\right)$ is the integral of the interpolating polynomial along the quadrature points:

$$I^{m+1}_m \left(G\left(t_{n,m}, u_{n,m}\right)\right) = \int_{t_{n,m}}^{t_{n,m+1}} G(\tau, u(\tau))\, d\tau \tag{91}$$

Finally, $u_{n+1} = u^{K+1}_{n,J}$. Here, $\theta = 1$ is used as in the original study [35] and $K = J - 1$. For the cases studied in this report, I observed that neglecting the second term on the right hand side, i.e. using $\theta = 0$, provides similar results but with greater numerical error.

4.2.5 Adaptive mesh refinement

The solver relies on a tree to handle the hierarchical structure of the grid adaptations. The initial grid is composed by *root* cells, corresponding to the lowest level. Each cell can have children. A cell that does not have children is called a *leaf* cell. If a root cell does not have children it is also tagged as a leaf cell. The root cells correspond to $level = 1$ and the maximum level is given by $l_{\max}$, which may depend on the problem. Each face of every cell has to be connected to a neighbor or to a boundary element. A cell can connect to a neighbor at the same level or at a lower level, but never at a higher level. In addition, there is a ghost tree to handle the ghost cells for inter-processor communications. Details about the tree structures are given elsewhere [69, 64].

When a cell is marked for refinement it is split in two, four, or eight for 1, 2, or 3 dimensions, respectively. The variables from the parent are projected onto the children with an identity projection. On the other hand, when all the children of a cell are marked for coarsening, the variables from the children are projected onto the parent cell with a least square projection. It is important to note that if the order p is kept constant, no data is lost when refinement is done, however data is lost when coarsening is done.

Cells are marked for adaptation based on an error estimator. The error ϵ_l of cell l is then normalized by the maximum error $\epsilon_{\max}$ found in the whole domain. Then, a logarithmic scale is applied as in [37]. The current hierarchical level in the tree for cell l is $level_l$. A target level l_t is estimated as:

$$l_t = \min\left(l_{\max}, \max\left(1, l_{\max} - \mathrm{INT}\left(\log\left(\epsilon_{\max}/\epsilon_j\right)/log(d)\right)\right)\right) \tag{92}$$

where d is a parameter that determines the sensibility of the refinement, the larger its value, the more refinement will be done. Even though the accuracy is expected to increase as d is raised, the computational cost will be higher too. The default value

adopted here is $d = 10$ as in [37], which is a good balance between computational cost and accuracy. If l_t is greater than $level_l$ then cell l is marked for refinement. If l_t is lower than $level_l$ then cell l is merged for coarsening.

In order to reduce oscillations in the adaptation, $\epsilon_{\max}$ is filtered in time:

$$\epsilon_{\max}^{[n+1]} = \varpi \ \epsilon_{\max} + (1 - \varpi) \ \epsilon_{\max}^{[n]}$$

where the super-index indicates the value used at the corresponding time step, and ϖ is a parameter. $\varpi = 0.63212$ is used here, which produces a first-order filter with a characteristic time of Δt.

Note that for coarsening to actually be feasible, all the children have to be marked for coarsening. The level difference between neighboring cells is not allowed to be larger than 1. For example, if cells 1 and 2 are neighbors, and $level_1 = 3$ and $level_2 = 4$; let's say that cell 2 is marked for refinement, then cell 1 will be marked for refinement also.

For the sake of simplicity and computational speed, a simple error estimator is used here. More accurate approaches are slower and may increase the overhead, making the adaptivity too costly.

Zhu et al. [162] compared a few different shock-detectors as estimators for refinement, and concluded that the most efficient based on their 1-D discontinuous problems was the KXRCF [74]:

$$\epsilon_{A,l} = \frac{|\int_{\delta\Omega_l^-} (U^- - U^+) dS|}{h_l^{\frac{p+1}{2}} \int_{\delta\Omega_j^-} dS ||U_l||} \tag{93}$$

where U is some relevant variable, $\delta\Omega_l^-$ is the element-boundary where the velocity is going into the element, h_l is the radius of the circumscribed circle in the element l, and the norm is based on an element average. In [112] the following error estimator was used for element l:

$$\epsilon_{B,l} = \int_{\Omega_l} |U^- - U^+| dS \tag{94}$$

Here, the best of both are combined, Eqs. 93 and 94, to obtain:

$$\epsilon_{C,l} = \frac{\int_{\delta\Omega_l} |U^- - U^+| dS}{\int_{\delta\Omega_l} dS} \tag{95}$$

In the current implementation the error is already normalized by the maximum error in the domain (Eq. 92) and so the additional normalization needed in Eq. 93 is not required. For the Euler equation two error estimators based on the density and the total energy are used. The estimators in Eqs. 93–95 are referred as KXRCF, JUMP1, and JUMP2, respectively. The difference between JUMP1 and JUMP2 can only be observed in 2 and 3 dimensions, so for the 1-D cases JUMP1 is not used. Unless specified otherwise, the estimator used below in the test cases is JUMP2.

4.2.6 Moment-Limiter for non-uniform grids

The limiting strategy of the ML is shown below first for 1-D. However, for completeness, the 2-D and 3-D extensions are summarized below. In section 4.2.6.2, I extend the original ML to non-uniform grids for 1-D, but its extension to higher dimensions is trivial, except for when a neighbor is split due to refinement, in which case the average of the two is used, and when the neighbor is coarser, in which case a virtual refinement of the neighbor is done. This last step has no analytical complexity, but its implementation may not be trivial. Figure 6 shows the stencil used for limiting purposes when coarser, finer, or equal level neighbors are present.

4.2.6.1 The Moment-Limiter concept

The idea is to limit the i^{th} derivative in x of U_l in the following way:

$$\frac{\partial^i \tilde{U}_l}{\partial x^i} = \text{minmod}\left(\frac{\partial^i U_l}{\partial x^i}, \beta_i D_i^+, \beta_i D_i^-\right) \tag{96}$$

where the minmod function is given by:

$$\text{minmod}(a,b,c) = \begin{cases} \text{sgn}(a)\ \min(|a|,|b|,|c|) & \text{if } \text{sgn}(a) = \text{sgn}(b) = \text{sgn}(c) \\ 0 & \text{otherwise} \end{cases} \tag{97}$$

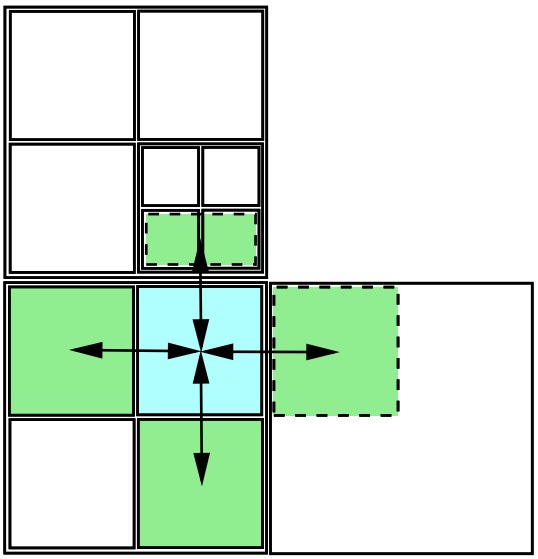


Figure 6: Example of a 2-D stencil used for limiting when coarser, finer, or equal-level neighbors are present.

and $\tilde{U}_l$ is the solution, U_l, after the limiter is applied. $D_i^{+/-}$ is an estimation of the i^{th} derivative based on one-sided differences:

$$D_i^+ = \frac{\overline{\frac{\partial^{i-1} U_{l+1}}{\partial x^{i-1}}} - \overline{\frac{\partial^{i-1} U_l}{\partial x^{i-1}}}}{\bar{x}_{l+1} - \bar{x}_l} \tag{98}$$

$$D_i^- = \frac{\overline{\frac{\partial^{i-1} U_l}{\partial x^{i-1}}} - \overline{\frac{\partial^{i-1} U_{l-1}}{\partial x^{i-1}}}}{\bar{x}_l - \bar{x}_{l-1}} \tag{99}$$

where $\bar{x}_l$ is the location of the centroid of element l, and β_i is a parameter to control the sensibility of the limiter. If there is a boundary condition against one of the faces of the element, then that side is neglected in Eq. 96.

In the literature, e.g. see [72], it is recommended to limit in characteristic variables when a system of equations is being solved. This means to replace Eq. 96 by:

$$\left(\mathbf{L}\frac{\partial^i \tilde{\mathbf{U}}_l}{\partial x^i}\right)_k = \text{minmod}\left(\left(\mathbf{L}\frac{\partial^i \mathbf{U}_l}{\partial x^i}\right)_k, \beta_i \left(\mathbf{L}\mathbf{D}_i^+\right)_k, \beta_i \left(\mathbf{L}\mathbf{D}_i^-\right)_k\right) \tag{100}$$

where $\mathbf{L}$ is a matrix composed by the left eigenvectors of the Jacobian, $\partial\mathbf{F}/\partial\mathbf{u}$, and

the subindex k refers the the k^{th} characteristic variable. Each characteristic variable is limited individually and this means that if a variable in a given element is not limited the others can still be limited. If limiting is applied in an element the resulted characteristic variables have to be converted back to the conservative variables, multiplying the characteristic variables by the inverse of $\mathbf{L}$, which is composed by the right eigenvectors of $\partial\mathbf{F}/\partial\mathbf{u}$. In the other hand, if one wants to use primitive variables for this stage, $\mathbf{L}$ should be replaced by the Jacobian, $\partial\mathbf{u}_p/\partial\mathbf{u}$, where $\mathbf{u}_p$ is the state vector in primitive variables. For instance, for the 5-eq. model de primitive variables are $(\alpha_1, \rho_1, \rho_2, v_i, P)^T$, while for single-phase models (e.g. Navier-Stokes) they are $(\rho, v_i, P)^T$. In the latest case, the temperature could have been used instead of the density. However, for multi-phase models the density is a better choice than the temperature. Thus, for consistency, density is used in both cases.

The algorithm to apply the limiter is the following:

1. Apply Eq. 96 for $i = p$ to every element. If $\dfrac{\partial^i \tilde{U}_l}{\partial x^i} = \dfrac{\partial^i U_l}{\partial x^i}$, then mark the element as if it does not need limiting anymore.

2. Apply Eq. 96 for $i = p - 1$ to every element that still needs to be limited.

3. Continue for $i = p - 2, ..., 1$ or until no element requires limiting.

Note that only the derivatives are modified, not the mean value, thus the limiter does not violate the conservation property.

4.2.6.2 The ML using a Legendre basis

The $(i-1)^{th}$ derivative respect to x of $\mathbf{U}_l$ (Eq. 72) can be expressed as:

$$\frac{\partial^{i-1}\mathbf{U}_l}{\partial x^{i-1}} = \left(\frac{2}{\Delta x_l}\right)^{i-1} \left[\sqrt{\frac{2i-1}{2}}\,(2i-3)!!\mathbf{c}_{l,i-1} + \frac{\partial^{i-1}}{\partial \xi^{i-1}} \sum_{k=i}^{p} \mathbf{c}_{l,k}\phi_k(\xi)\right] \tag{101}$$

and the i^{th} derivative in x of Eq. 72 can be expressed as:

$$\frac{\partial^{i}\mathbf{U}_l}{\partial x^{i}} = \left(\frac{2}{\Delta x_l}\right)^{i} \left[\sqrt{\frac{2i+1}{2}}\,(2i-1)!!\mathbf{c}_{l,i} + \frac{\partial^{i}}{\partial \xi^{i}} \sum_{k=i+1}^{p} \mathbf{c}_{l,k}\phi_k(\xi)\right] \tag{102}$$

where Δx_l is the length of element l.

At the same time, the i^{th} derivative could be estimated from the forward or backward differences of $\dfrac{\partial^{i-1}\mathbf{U}_l}{\partial x^{i-1}}$:

$$\frac{\partial^i \mathbf{U}_l}{\partial x^i} = \left(\frac{\partial^{i-1} U_{l+1}}{\partial x^{i-1}} - \frac{\partial^{i-1} U_l}{\partial x^{i-1}}\right) \frac{2}{\Delta x_{l+1} + \Delta x_l} \tag{103}$$

$$\frac{\partial^i \mathbf{U}_l}{\partial x^i} = \left(\frac{\partial^{i-1} U_l}{\partial x^{i-1}} - \frac{\partial^{i-1} U_{l-1}}{\partial x^{i-1}}\right) \frac{2}{\Delta x_l + \Delta x_{l-1}} \tag{104}$$

Therefore, ignoring higher order derivatives:

$$\mathbf{c}_{l,i} = \frac{2\vartheta_+}{1+\vartheta_+}\sqrt{\frac{2i-1}{2i+1}}\frac{1}{2(2i-1)}\left(\vartheta_+^{i-1}\mathbf{c}_{l+1,i-1} - \mathbf{c}_{l,i-1}\right) \tag{105}$$

$$\mathbf{c}_{l,i} = \frac{2\vartheta_-}{1+\vartheta_-}\sqrt{\frac{2i-1}{2i+1}}\frac{1}{2(2i-1)}\left(\mathbf{c}_{l,i-1} - \vartheta_-^{i-1}\mathbf{c}_{l-1,i-1}\right) \tag{106}$$

where $\vartheta_- = \Delta x_l/\Delta x_{l-1}$ and $\vartheta_+ = \Delta x_l/\Delta x_{l+1}$. Note that if the grid is uniform $\vartheta_- = 1$, $\vartheta_+ = 1$, and the derived equations converge to the solution in [72]. Thus, the difference between the current derivation and the one in [72] starts in Eq. 103 and 104, where constant Δx is not assumed.

One could apply the limiter as:

$$\tilde{\mathbf{c}}_{l,i} = \text{minmod}\left(\mathbf{c}_{l,i}, \boldsymbol{\Delta}_i^+, \boldsymbol{\Delta}_i^-\right) \tag{107}$$

where $\boldsymbol{\Delta}^+$ and $\boldsymbol{\Delta}^-$ are the right hand sides of Eq. 105 and Eq. 106. However, to make the limiter less numerically diffusive, [72] uses an expression equivalent to:

$$\tilde{\mathbf{c}}_{l,i} = \text{minmod}\left[\mathbf{c}_{l,i}, 2\left(2i-1\right)\boldsymbol{\Delta}_i^+, 2\left(2i-1\right)\boldsymbol{\Delta}_i^-\right] \tag{108}$$

This equation should be the actual implementation of what Eq. 96 represents. The same procedure is easily extended to 2-D and 3-D.

4.2.6.3 Limiting with an averaged-derivatives basis

With this basis the derivatives $D_i^{+/-}$ in Eqs. 98 and 99 can easily be estimated in an element-averaged sense. Using Eq. 80:

$$D_i^+ = \frac{\left(\frac{2}{\Delta xj+1}\right)^{i-1}\mathbf{d}_{j+1,i-1} - \left(\frac{2}{\Delta xj}\right)^{i-1}\mathbf{d}_{j,i-1}}{\bar{x}_{j+1} - \bar{x}_j} \tag{109}$$

$$D_i^- = \frac{\left(\frac{2}{\Delta xj}\right)^{i-1} \mathbf{d}_{j,i-1} - \left(\frac{2}{\Delta xj-1}\right)^{i-1} \mathbf{d}_{j-1,i-1}}{\bar{x}_j - \bar{x}_{j-1}} \tag{110}$$

Thus the limiter becomes:

$$\tilde{\mathbf{d}}_{j,i} = \text{minmod}\left[\mathbf{d}_{j,i}, \frac{\beta_i \vartheta_+}{1+\vartheta_+}\left(\vartheta_+^{i-1}\mathbf{d}_{j+1,i-1} - \mathbf{d}_{j,i-1}\right), \frac{\beta_i \vartheta_-}{1+\vartheta_-}\left(\mathbf{d}_{j,i-1} - \vartheta_-^{i-1}\mathbf{d}_{j-1,i-1}\right)\right] \tag{111}$$

where Eqs. 109 and 110 are used to replace D^+ and D^-, respectively. [154] recommend using $\beta_i = 2$. The same authors do not mention the need of characteristic variables.

4.2.6.4 Two dimensional Moment Limiter

In this case, cross derivatives should be taken into account. Hence, for element l, m:

$$\frac{\partial^{i+j}\tilde{U}_{l,m}}{\partial X_1^i \partial X_2^j} = \text{minmod}\left(\frac{\partial^{i+j}U_{l,m}}{\partial X_1^i \partial X_2^j}, \beta_{ij} D_{ij}^{X_1+}, \beta_{ij} D_{ij}^{X_1-}, \beta_{ij} D_{ij}^{X_2+}, \beta_{ij} D_{ij}^{X_2-}\right) \tag{112}$$

where the frame (X_1, X_2) is a rotation of (x_1, x_2) aligned to the computational coordinates (ξ, η) of the current element.

The limiting starts from orders (p, p), and continuous with the pair $(p, p-1)$ and $(p-1, p)$, then with the pair $(p, p-2)$ and $(p-2, p)$, and so on until $(p, 0)$ and $(0, p)$. Then the loop starts again from $(p-1, p-1)$, and continuous with $(p-1, p-2)$ and $(p-2, p-1)$, and so on. Whenever a pair is not changed, the limiting procedure is stopped.

If a neighboring cell is split because of refinement, the average between the two neighboring children is used. If a neighboring cell is coarser because the current cell is more refined, the modes of the neighbor have to be computed as it were refined too. Note that the characteristic decomposition is only consistent in a 1-D sense. Given that the ML can be multi-dimensional, the characteristic decomposition would have to be done in an arbitrary direction. Therefore, for multi-dimensional cases a primitive-variable decomposition may be more appropriate.

4.2.6.5 Three dimensional Moment Limiter

For element l, m, n:

$$\frac{\partial^{i+j+k}\tilde{U}_{l,m,n}}{\partial X_1^i \partial X_2^j \partial X_3^k} = \text{minmod}\left(\frac{\partial^{i+j+k}U_{l,m,n}}{\partial X_1^i \partial X_2^j \partial X_3^j}, \beta_{ijk} D_{ijk}^{X_1+}, \beta_{ijk} D_{ijk}^{X_1-}, \right.$$
$$\left. \beta_{ijk} D_{ijk}^{X_2+}, \beta_{ijk} D_{ijk}^{X_2-}, \beta_{ijk} D_{ijk}^{X_3+}, \beta_{ijk} D_{ijk}^{X_3-}\right) \tag{113}$$

where the frame (X_1, X_2, X_3) is a rotation of (x_1, x_2, x_3) aligned to the computational coordinates (ξ, η, ζ) of the current element.

In this case, the limiting starts from orders (p, p, p), and continuous for the triad $(p, p, p-1)$, $(p, p-1, p)$ and $(p-1, p, p)$, then for the triad $(p, p, p-2)$, $(p, p-2, p)$ and $(p-2, p, p)$, and so on until $(p, p, 0)$, $(p, 0, p)$ and $(0, p, p)$. Then the loop starts again from $(p-1, p-1, p-1)$, and continuous for $(p-1, p-1, p-2)$, $(p-1, p-2, p-1)$ and $(p-2, p-1, p-1)$, and so on. Whenever a triad is not changed, the limiting procedure is stopped.

If a neighboring cell is split because of refinement, the average between the four neighboring children is used. Like in the 2-D case, a primitive-variable decomposition may be the optimal approach.

4.2.7 Positivity check

After limiting, a cell may still have a non-valid state, especially close to an interface due to the sharp change of α. In order to solve this problem, Krivodonova [72] recommends cropping the order down to $p = 1$, and if it is still not valid, then cropping it again and leave the cell-average value ($p = 0$). I refer to this technique as LIM1. Even though this approach works, it can be very diffusive, in particular for the 5-eq. model at the interface.

Therefore, instead of the approach in [72], I apply a similar method to [159], which I call LIM2, where a function f_l is replaced by $\widehat{f}_l(x)$:

$$\widehat{f}_l(x) = \theta\left[f_l(x) - \overline{f}_l\right] + \overline{f}_l \tag{114}$$

If f is defined for $[0, 1]$ then:

$$\theta = \min\left(\frac{\overline{f}_l - \epsilon}{\overline{f}_l - f_{\min}}, \frac{\overline{f}_l - 1 + \epsilon}{\overline{f}_l - f_{\max}}, 1\right)$$

If f is defined for $[0, \infty)$ then:

$$\theta = \min\left(\frac{\overline{f}_l - \epsilon}{\overline{f}_l - f_{\min}}, 1\right)$$

where $f_{\min} = \min_\beta f_l(\mathbf{x}_{l,\beta})$, $f_{\max} = \max_\beta f_l(\mathbf{x}_{l,\beta})$, and $\mathbf{x}_{l,\beta}$ are the quadrature points of element l. The behavior is exemplified in Fig. 7, where the solution is expected to be greater than zero, and given that the original solution crosses is below the valid limit, it is limited. Note that the average of the original and corrected solutions is the same within the element.

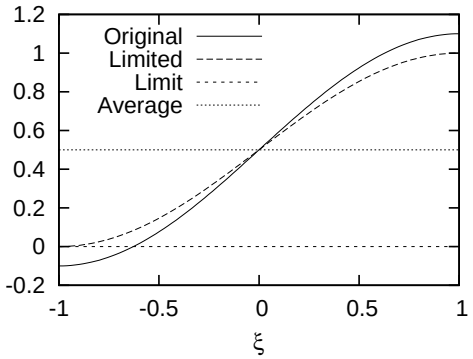


Figure 7: Diagram of the positivity correction algorithm.

Given that α_1, $\alpha_1\rho_1$, and $\alpha_2\rho_2$ need to be checked, θ is computed for each one of them. For clarity, the subindex for the phase is ignored here, so α refers to α_1. The volume fraction should be bounded between ϵ and $1 - \epsilon$ (here I use $\epsilon = 10^{-6}$):

$$\theta_\alpha = \min\left(\frac{\overline{\alpha}_l - \epsilon}{\overline{\alpha}_l - \alpha_{\min}}, \frac{\overline{\alpha}_l - 1 + \epsilon}{\overline{\alpha}_l - \alpha_{\max}}, 1\right)$$

Similarly, for $\alpha_1\rho_1$ and $\alpha_2\rho_2$:

$$\theta_{\alpha\rho} = \min\left(\frac{\overline{\alpha\rho}_l - \epsilon}{\overline{\alpha\rho}_l - \alpha\rho_{\min}}, 1\right)$$

We compute a common coefficient: $\theta = \min\left(\theta_{\alpha}, \theta_{\alpha_1\rho_1}, \theta_{\alpha_2\rho_2}\right)$. Now, θ is applied to all the conservative variables based on Eq. 114.

When only one phase is present, this limiter is directly applied to the density ρ, which should be greater than ϵ, so $\rho_l(x)$ is replaced by $\widehat{\rho}_l(x)$:

$$\widehat{\rho}_l(x) = \theta_\rho\left(\rho_l(x) - \overline{\rho}_l\right) + \overline{\rho}_l$$

$$\theta_\rho = \min\left(\frac{\overline{\rho}_l - \epsilon}{\overline{\rho}_l - \rho_{\min}}, 1\right)$$

$$\rho_{\min} = \min_{\beta} \rho_l(\mathbf{x}_{l,\beta})$$

Finally, the pressure is checked at every quadrature point. If it is lower than the threshold value, the modes are cropped in the element only leaving its average. Using the ideal-gas EOS, the authors in [159] suggested a more sophisticated check for the pressure. However, for generic equations of state the relationship between pressure and the conservative variables is more complicated, so a precise limiter would be prohibitively expensive. In [141] a faster option was proposed, but only valid if p is a concave function of $\mathbf{u}$ and this is not the case for generic EOSs, so it is not used here. LIM2 is used by default in all the test cases unless specified otherwise. This approach was shown to preserve the order of accuracy in [159] under a given CFL restriction.

4.2.8 Troubled-cell detector

The detector presented in this study is a modification of the AP-TVD detector presented in [154] for a Spectral Difference (SD) scheme. The adapted technique consists of the following two steps:

1. For each cell l compute:

$$\bar{\mathbf{U}}_{\max,l} = \max\left(\bar{\mathbf{U}}_{l-1}, \bar{\mathbf{U}}_l, \bar{\mathbf{U}}_{l+1}\right) \tag{115}$$

$$\bar{\mathbf{U}}_{\min,l} = \min\left(\bar{\mathbf{U}}_{l-1}, \bar{\mathbf{U}}_l, \bar{\mathbf{U}}_{l+1}\right) \tag{116}$$

Where the $\bar{\mathbf{U}}_l$ indicated the average of $\mathbf{U}$ in cell l. If for any node i in element l $\mathbf{U}_{i,l} > 1.001\ \bar{\mathbf{U}}_{\max,l}$ or $\mathbf{U}_{i,l} < 0.999\ \bar{\mathbf{U}}_{\min,l}$, then proceed to step 2, else the element is not marked.

2. For each dimension j the idea is to compute:

$$\frac{\partial^2 \tilde{\mathbf{U}}_l}{\partial x_j^2} = minmod\left(\frac{\partial^2 \mathbf{U}_l}{\partial x_j^2}, \beta\frac{\dfrac{\partial \mathbf{U}_{l+1}}{\partial x_j} - \dfrac{\partial \mathbf{U}_l}{\partial x_j}}{x_{l+1} - x_j}, \beta\frac{\dfrac{\partial \mathbf{U}_l}{\partial x_j} - \dfrac{\partial \mathbf{U}_{l-1}}{\partial x_j}}{x_l - x_{l-1}}\right) \tag{117}$$

The derivatives are estimated from the Legendre polynomials as in Section 4.2.6.2, so the implementation of Eq. 117 is:

$$\tilde{\mathbf{c}}_{l,2} = minmod\left(\mathbf{c}_{l,2}, \varrho\vartheta_+\frac{\vartheta_+\mathbf{c}_{l+1,1} - \mathbf{c}_{l,1}}{1+\vartheta_+}, \varrho\vartheta_-\frac{\mathbf{c}_{l,1} - \mathbf{c}_{l-1,1}\vartheta_-}{1+\vartheta_-}\right) \tag{118}$$

where $\varrho = 2\sqrt{3/5}$. If $\tilde{\mathbf{c}}_{2,l} \neq \mathbf{c}_{2,l}$ then the cell is marked for limiting. According to [154], β is a parameter between 1 and 2, the higher its value the less dissipative the scheme will be. $\beta = 2$ is used to make it consistent with the ML used here.

There are two main differences in the current limiter with respect to the AP-TVD developed earlier [154]. The first one is in step 1 where every node in the element is tested, while in [154] only the nodes at element boundaries are checked. The second difference is in the way the derivatives are estimated in step 2, averaged-derivatives were used in [154], while here I suggest using estimations of the derivatives, as done in the ML based on Legendre polynomials to avoid computing the averaged-derivatives. Thus, I call the current detector the moment-based AP-TVD or MB-AP-TVD.

Usually, the detection is done based on the conservative variables, instead of transforming to characteristic or primitive variables, in order to keep this stage as computationally cheap as possible.

4.2.9 Surface tension

The force due to the surface tension is modeled as a volumetric force as shown in Eq. 4: $-\sigma\kappa\frac{\partial\alpha_1}{\partial x_i}$, where σ is a parameter based on the two phases present, and $\frac{\partial\alpha_1}{\partial x_i}$ is computed as $\frac{\partial\alpha_1}{\partial\mathbf{x}} = (\mathbf{q}_{\alpha m} + \mathbf{q}_{\alpha p})/2$, see Section 4.2.1. Note that although flux terms usually require one sided numerical fluxes, the source term can use a central formulation.

The rapid change in α at the interface produces oscillations, especially in high-order derivatives. This is relevant when calculating κ. If κ is computed using high order derivatives of the solution within the element, as is natural in DG, the result is extremely noisy. This was shown by [88], where they tried to solve:

$$\int_{\Omega_j} \phi\ \kappa\ dV - \int_{\Omega_j} \nabla\phi \cdot \mathbf{n}(\mathbf{q}_j) dV + \int_{\partial\Omega_j} \phi\ \widehat{n}(\mathbf{q}^-, \mathbf{q}^+) dS = 0$$

Note that in [88] the level-set function was used, which is smooth at the interface. In our current formulation α is not so smooth, so the fluctuations should be even larger here.

In order to reduce the oscillations, we first apply a spectral filter to the derivative $(\mathbf{q}_{\alpha m} + \mathbf{q}_{\alpha p})/2$, where only the first mode (the average) is kept and the rest of the modes are ignored. This produces a constant value of the gradient within each element and the result is used to compute the normal $\mathbf{n}$. This procedure limits the order of accuracy, but prevents oscillations. The least-squared approximation by [88] prevents oscillations also with the cost of limiting the order of accuracy, but with a more demanding procedure.

Finally, the divergence is computed based on a Finite Difference formulation using the cell-centered value of the current element and its immediate neighbors. The result is assumed constant within the element. The cost of the simplicity and stencil-compactness of the method is the maximum expected order accuracy of 1.

AMR may produce non-matching faces. When a neighbor is split due to refinement, the average of the two children is used. When the neighbor is coarser, a virtual refinement of the neighbor is done. Figure 6 shows an example of the stencil when coarser, finer, and equal-level neighbors are present.

The line where two fluids and a wall come into contact is called *contact line*. The angle formed by the interface between the fluids at the contact line is called *contact angle*, ϑ. When the flow is steady, the contact angle is equal to the equilibrium contact angle, ϑ_{eq}, see Fig. 8. A simple model to represent the behavior of the contact angle,

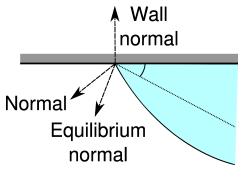


Figure 8: Diagram of the contact angle.

used by [100] and reproduced here for completeness, consists of manipulating the direction of the normal at the contact line, such that the angle between the new normal and the wall is $90^o - \vartheta_{eq}$. This way, the interface will move trying to adjust itself such that the contact angle tends to ϑ_{eq}. The modified normal $\tilde{n}_{eq}$ is given as $\tilde{n}_{eq} = b_1\tilde{n} + b_2\tilde{n}_w$, where $\tilde{n}$ is the normal to the interface ($\tilde{n} = -\nabla\alpha_1/|\nabla\alpha_1|$), $\tilde{n}_w$ is the normal into the wall, and

$$b_1 = \frac{\sin(\vartheta_{eq})}{\sqrt{1-(\tilde{n}\cdot\tilde{n}_w)}} \qquad\qquad b_2 = -b_1(\tilde{n}\cdot\tilde{n}_w) - \cos(\vartheta_{eq})$$

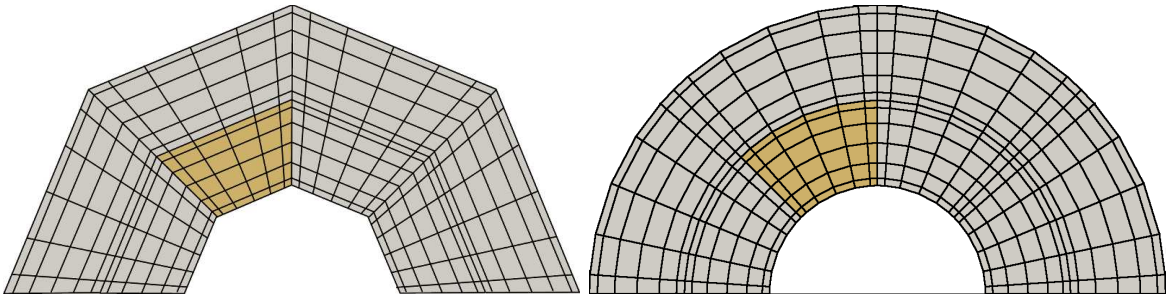
4.2.10 Boundary conditions

Looking at Eq. 74 one can see that the boundary conditions may affect the element-boundary integral ($\partial\Omega$). There are two common approaches to enforce boundary conditions. One is changing the numerical fluxes $\tilde{F}$, $\tilde{F}_V$, and $\tilde{U}$, based on the type of condition [3]. The other option is to modify the state vector on the outside based on the type of condition [9]. Unless mentioned otherwise, the second method is used in

this study. This second method may cause loss of conserved properties particularly at curved walls with low resolution, but this is an issue that requires further research.

4.2.10.1 Walls

As Krivodonova and Berger [73] point out, difficulties can arise when the numerical boundary does not coincide with the physical boundary. For instance, if the faces of an element on a wall boundary are kept straight, artificial entropy is generated at the kinks formed at the joint of two elements. This error does not decrease fast enough when reducing the grid size. Thus, it is very important for high-order elements to copy the the shape of the boundary accurately. Figure 9 shows both cases: elements straight and curved edges. The later one is the right approach for a curved wall. Note that the shape of the face corresponds to a polynomial of order p.



(a) Straight edges (wrong). (b) Curved edges (right).

Figure 9: Grid in between two coaxial cylinders.

For inviscid walls, the momentum of at the outside is equal to the inside but mirrored respect to the wall, while the rest of the variables are copied. For viscous walls two options are available: adiabatic or isothermal. In both cases the velocity is fixed at the boundary (usually zero). In the adiabatic case, the gradient of the temperature is equal to zero, while the energy and density are copied from the inside. For the isothermal case, the derivatives in the outside are copied from the inside, the temperature is fixed and pressure is copied from the inside.

4.2.10.2 Inflow

The velocity, temperature, and volume fraction are fixed, while the pressure is copied from the inside. The $\mathbf{q}^+$ is copied from the inside ($\mathbf{q}^-$). Note that this is a reflective configuration.

4.2.10.3 Outflow

The pressure is fixed, while the velocity, volume fraction, and temperature are copied from the inside. Note that this is a reflective configuration.

4.2.10.4 Farfield

The whole state vector in the farfield $\mathbf{U}_{BC}$ needs to be known The state vector outside, $\mathbf{U}^+$, is computed based on a characteristic decomposition as follows. The fluctuations in characteristic variables is computed multiplying the left eigenvector matrix by $\mathbf{U}^- - \mathbf{U}_{BC}$. The sign of eigenvalues determine the direction of the characteristic waves. The fluctuations corresponding to incoming waves are made equal to zero. The resulted vector is multiplied on the left by the right eigenvector matrix. The product is equal to $\mathbf{U}^+ - \mathbf{U}_{BC}$. Finally, the gradients are made equal to zero ($\mathbf{q}^+ = 0$). This approach is non-reflective for small fluctuations.

4.2.11 Absorbing Sponge Zone

The Absorbing Sponge Zone is a type of absorbing boundary condition that intends to reduce fluctuations respect to a reference state vector. This is accomplished by adding a source term to the governing equations. This source term should be applied on a layer inside the domain of a predetermined thickness and touching the artificial boundary. Two of the advantages of this approach are that it is independent of the numerical scheme used and that it is simple to implement. The source term added is defined as:

$$\mathbf{S} = \beta(\mathbf{x}) \left[\mathbf{u}(\mathbf{x}, t) - \tilde{\mathbf{u}}(\mathbf{x})\right]$$

Table 2: Face index for a Cartesian block

Face	Index
$x_{1,\min}$	1
$x_{1,\max}$	2
$x_{2,\min}$	3
$x_{2,\max}$	4
$x_{3,\min}$	5
$x_{3,\max}$	6

where $\beta(\mathbf{x})$ is the absorbing coefficient, $\mathbf{u}(\mathbf{x}, t)$ is the local state vector, and $\tilde{\mathbf{u}}(\mathbf{x})$ is a reference state vector. β goes from 0 inside the domain to β_0 at the boundary, i.e. at the beginning of the layer and at the end of the layer, respectively. The layer is more easily defined for a Cartesian zone, and in this study we restrict ourselves to this case. For 3-D, there are 6 possible faces where layers can be defined and each one is numbered as in Table 2. Each Cartesian zone i is defined in $[x_{i,\text{start}}, x_{i,\text{end}}]$, where $x_{i,\text{start}} < x_{i,\text{end}}$, for $i = 1 \ldots 6$. β_0 is defined by:

$$\beta(\mathbf{x}) = \beta_0 \ \{1 + \cos[\pi A(\mathbf{x})]\}$$

where $A(\mathbf{x}) \in [0, 1]$ and its computation is shown in pseudo-code in Algorithm 1.

Algorithm 1 Calculate A for the absorbing coefficient.

$A \leftarrow 1$
for $i = 1 \ldots 3$ **do**
    $B \leftarrow 1$
    if layer $2i - 1$ enabled **then**
        $B \leftarrow B - \max[1 - (x_i - x_{2i-1,\text{start}})/(x_{2i-1,\text{end}} - x_{2i-1,start}), 0]$
    end if
    if layer $2i$ enabled **then**
        $B \leftarrow B - \max[1 - (x_{2i,\text{start}} - x_i)/(x_{2i,\text{end}} - x_{2i,start}), 0]$
    end if
    $A \leftarrow A \cdot B$
end for

4.2.12 Fourier transform

4.2.12.1 Introduction

The spectrum of some variables can reveal very useful information of a flow like its turbulence characteristics. The spectrum of some variable is obtained applying a Fourier Transform to it, thus, the data is converted from a spatial space into a spectral space. This space can be in terms of the frequency (ω) or the wavelength (κ). We'll focus on the latest one, but the analysis applies to either one. This approach can be used for any number of dimensions, here it is only used in one dimension, but it can be trivially extended.

The Fourier transform $\mathfrak{F}_\kappa$ of a periodic variable u with period L is:

$$\widehat{u}(\kappa_n) = \mathfrak{F}_\kappa(u(x)) = \frac{1}{L}\int_0^L u(x) e^{-i\ \kappa_n\ x} dx \tag{119}$$

where $\kappa_n = \kappa_0\ n$ for $n = -\infty, \ldots, \infty$, and $\kappa_0 = 2\pi/L$. This periodic variable u can now be expressed as a Fourier Series:

$$u(x) = \sum_{n=-\infty}^{\infty} \widehat{u}(\kappa_n) e^{i\ \kappa_n x} \tag{120}$$

where the coefficients of the series were obtained in Eq. 119. Note that the coefficients are complex. In the case of a real function $\widehat{u}(-\kappa) = \widehat{u}(\kappa)^*$, where * is the conjugate. In the case of fluid dynamics, researchers are usually interested in the kinetic energy spectrum:

$$E(\kappa) = \frac{1}{2}\widehat{u}(\kappa)\widehat{u}(\kappa)^* \tag{121}$$

More specifically, the interest is in the spectrum in terms of $|\kappa|$. Given that $u(x)$ is real, $E(\kappa)$ is symmetric respect to $\kappa = 0$, thus

$$E(|\kappa|) = \widehat{u}(\kappa)\widehat{u}(\kappa)^* \tag{122}$$

4.2.12.2 Discrete Fourier Transform

When dealing with numerical methods, discrete functions are usually used. If one is interested in the Fourier transform of a discrete function, the analysis above has

to be adapted. For discrete function $f(x) \rightarrow f(x_j)$ defined over N points such that $f_j = f(x_j)$, where $x_j = j\ \Delta x$, with $j = 0, 1, \ldots, N-1$, the discrete Fourier transform is:

$$\widehat{f} = \sum_{j=0}^{N-1} f_j e^{-i2\pi nj/N} \tag{123}$$

Although this discrete operation can be convenient, it is not as accurate as operating with the original Fourier transform.

Evaluating the exponential function is a relatively expensive computation. A Fourier transform of a function defined over N points has a complexity $O(N^2)$. Fast Fourier Transforms (FFT) are algorithms that reduce this complexity to $O(Nlog_2N)$.

4.2.12.3 Fourier transform of DG solutions - DG-FT

In the case of the DG method, the solution is represented by some piecewise-continuous basis. If one needs the spectrum of the solution, one could interpolate the solution onto a uniform grid and apply an FFT, or one could apply the original Fourier transform since the variable is piecewise-continuous. In a DG space $u(x)$ is approximated by $U(x)$ such that for element l

$$U_l(x) = \sum_{j=0}^{p} \mathbf{c}_{j,l}\phi_j(\xi) \tag{124}$$

where $\phi(\xi)$ is the basis and $\mathbf{c}_{j,l}$ are the coefficients. Then, the Fourier transform in Eq. 119 could be approximated as

$$\widehat{u}(\kappa_n) \simeq \widehat{U}(\kappa_n) = \frac{1}{L}\int_0^L U(x)e^{-i\ \kappa_n\ x}dx \tag{125}$$

$$= \frac{1}{L}\sum_{l=0}^{N_e}\int_{x_{1_l}}^{x_{2_l}} U_l(x)e^{-i\ \kappa_n\ x}dx \tag{126}$$

where x_{1_l} and x_{2_l} are the beginning and end of element l. Assuming a uniform grid with spacing Δx and element centroids at x_{0_l}, it can be simplified:

$$\widehat{U}(\kappa_n) = \frac{1}{L}\sum_{l=1}^{N_e}\int_{-1}^{1} U_l(\xi)e^{-i\ \kappa_n(x_{0_l}+\Delta x\ \xi/2)}\frac{\Delta x}{2}d\xi \tag{127}$$

$$= \frac{\Delta x}{2L}\sum_{l=1}^{N_e}\int_{-1}^{1}\left[\sum_{j=0}^{p}\mathbf{c}_{j,l}\phi_j(\xi)\right]e^{-i\ \kappa_n(\Delta x(l-\frac{1}{2})+\Delta x\ \xi/2)}d\xi \tag{128}$$

$$= \frac{\Delta x}{2L}\sum_{l=1}^{N_e}e^{-i\ \kappa_n\ \Delta x(l-\frac{1}{2})}\left[\sum_{j=0}^{p}\mathbf{c}_{j,l}\int_{-1}^{1}\phi_j(\xi)e^{-i\ \kappa_n\ \Delta x\ \xi/2}d\xi\right] \tag{129}$$

$$= \frac{\Delta x}{2L}\sum_{l=1}^{N_e}e^{-i\ \kappa_n\ \Delta x(l-\frac{1}{2})}\sum_{j=0}^{p}\mathbf{c}_{j,l}\mathfrak{K}_j \tag{130}$$

where $\mathfrak{K}$ is the kernel, which is independent of the element. Given κ_n and Δx it can be computed analytically:

$$\mathfrak{K}_j = \int_{-1}^{1}\phi_j(\xi)e^{-i\ \kappa_n\ \Delta x\ \xi/2}d\xi \tag{131}$$

where $j = 0, \ldots, p$. These expressions are shown up to 9^{th} order in Table 3 for $\kappa_n \neq 0$. If $\kappa_n = 0$, then $\mathcal{K}_0 = \sqrt{2}$, and $\mathcal{K}_j = 0$ for $j > 0$.

4.2.13 Spectral filter for LES

An explicit filter is required for dynamic models in LES. The filtering approach used here is based on the techniques presented for the Spectral Element Method (SEM) in [81, 12, 119] and DG in [120].

The filter is explained for the 1-D case, but its extension to multi-dimensions is straightforward given that the multi-dimensional filter is a result of the tensor-product of the 1-D filter. The coefficients of the filter are ℓ_i for $i = 0, \ldots, p$, and they are applied to the coefficients of the solution as: $\mathbf{c}'_{i,l} = \ell_i \mathbf{c}_{i,l}$. This is equivalent to multiplying the solution vector by a diagonal matrix which has the ℓ_i coefficients on its diagonal.

Given that the exact characteristic length $\bar{\triangle}$ of the filter does not have to be specified (see Section 3.2.3), we can simply say that it is proportional to the size of

Table 3: Kernel to compute the Fourier transform of a DG solution for $\kappa_n \neq 0$. Notation: $\mathcal{K} = \kappa_n \Delta x/2$.

j	$\mathfrak{K}_j$ for $\kappa_n \neq 0$
0	$\frac{\sqrt{2}\sin(\mathcal{K})}{\mathcal{K}}$
1	$\frac{i\sqrt{6}(\mathcal{K}\cos(\mathcal{K})-\sin(\mathcal{K}))}{\mathcal{K}^2}$
2	$\frac{\sqrt{10}\left(3\mathcal{K}\cos(\mathcal{K})+\left(-3+\mathcal{K}^2\right)\sin(\mathcal{K})\right)}{\mathcal{K}^3}$
3	$\frac{i\sqrt{14}\left(\mathcal{K}\left(-15+\mathcal{K}^2\right)\cos(\mathcal{K})+3\left(5-2\mathcal{K}^2\right)\sin(\mathcal{K})\right)}{\mathcal{K}^4}$
4	$\frac{3\sqrt{2}\left(5\mathcal{K}\left(-21+2\mathcal{K}^2\right)\cos(\mathcal{K})+\left(105-45\mathcal{K}^2+\mathcal{K}^4\right)\sin(\mathcal{K})\right)}{\mathcal{K}^5}$
5	$\frac{i\sqrt{22}\left(\mathcal{K}\left(945-105\mathcal{K}^2+\mathcal{K}^4\right)\cos(\mathcal{K})-15\left(63-28\mathcal{K}^2+\mathcal{K}^4\right)\sin(\mathcal{K})\right)}{\mathcal{K}^6}$
6	$\frac{\sqrt{26}\left(21\mathcal{K}\left(495-60\mathcal{K}^2+\mathcal{K}^4\right)\cos(\mathcal{K})+\left(-10395+4725\mathcal{K}^2-210\mathcal{K}^4+\mathcal{K}^6\right)\sin(\mathcal{K})\right)}{\mathcal{K}^7}$
7	$\frac{i\sqrt{30}\left(\mathcal{K}\left(-135135+17325\mathcal{K}^2-378\mathcal{K}^4+\mathcal{K}^6\right)\cos(\mathcal{K})-7\left(-19305+8910\mathcal{K}^2-450\mathcal{K}^4+4\mathcal{K}^6\right)\sin(\mathcal{K})\right)}{\mathcal{K}^8}$
8	$\frac{\sqrt{34}\left(9\mathcal{K}\left(-225225+30030\mathcal{K}^2-770\mathcal{K}^4+4\mathcal{K}^6\right)\cos(\mathcal{K})+\left(2027025-945945\mathcal{K}^2+51975\mathcal{K}^4-630\mathcal{K}^6+\mathcal{K}^8\right)\sin(\mathcal{K})\right)}{\mathcal{K}^9}$
9	$\frac{i\sqrt{38}\left(\mathcal{K}\left(34459425-4729725\mathcal{K}^2+135135\mathcal{K}^4-990\mathcal{K}^6+\mathcal{K}^8\right)\cos(\mathcal{K})-45\left(765765-360360\mathcal{K}^2+21021\mathcal{K}^4-308\mathcal{K}^6+\mathcal{K}^8\right)\sin(\mathcal{K})\right)}{\mathcal{K}^{10}}$

the element, which has $p+1$ degrees of freedom in each direction. The test filter level $\widehat{\triangle}$ was defined to be $2\bar{\triangle}$. Therefore, its solution can be represented with the first $(p+1)/2$ degrees of freedom. Accordingly, the filter is defined for odd values of p as:

$$\ell_i = \begin{cases} 1 & \text{if } i < (p+1)/2 \\ 0 & \text{otherwise} \end{cases} \tag{132}$$

for $i = 0, \ldots, p$. Whereas, for even values of p:

$$\ell_i = \begin{cases} 1 & \text{if } i < p/2 \\ 0.5 & \text{if } i = p/2 \\ 0 & \text{otherwise} \end{cases} \tag{133}$$

CHAPTER V

INTERFACE SHARPENING

5.1 Introduction

In spite of the several advantages of DIM respect to other multi-fluid models, it present one main caveat, which is the thickening of the interface as the simulation progresses. The governing equations do not present a natural way to control the thickness, so an artificial algorithm has to be used. Some methods developed for this purpose have been presented in the literature, most of them without adequate success. In this chapter, the method presented in Section 4.1 is evaluated in one and two dimensions.

The test cases included in this chapter were simulated with a Finite Volume solver with MUSCL and a predictor-corrector for time integration. The sharpening method is independent of the scheme except for the strength, which is a scalar that can be calibrated for each solver.

5.2 Verifications

The details of the test cases and their rationals to verify the sharpening scheme are summarized in Table 4.

5.2.1 Interface convected in 1-D

A domain of 2 mm-long with 100 cells is initialized with water in the left 0.5 mm and air in the rest. The velocity is 10 m/s. The boundary conditions are inflow on the left and non-reflective outflow on the right. The value of η_{opt} used is the one derived for the Upwind scheme. The properties of each material is specified in table 5.

Every simulation presented here is initialized with a sharp interface and smoothed

Table 4: Summary of cases to verify and validate the sharpening scheme.

Test case	Purpose	Section
1-D convection	Show the capability to control the gradient.	5.2.1
Planar interface in 2-D	Verify the convection of an interface not aligned with the grid.	5.2.2
Cylinder in 2-D	Verify the scheme for case with curvature not equal to zero.	5.2.3

Table 5: Material properties

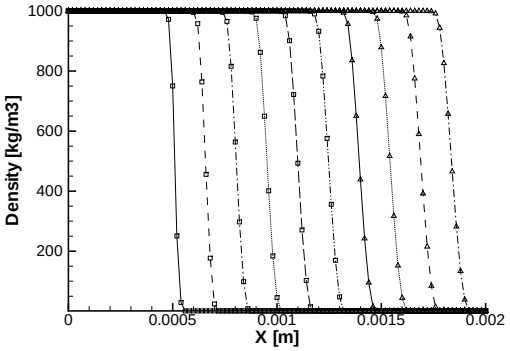
Substance	$P_\infty\ [Pa]$	γ	$\mu\ [Pa\ s]$	$\lambda\ [W/(m\ K)]$
Air	0.0	1.4	$19.3 \cdot 10^{-6}$	0.0
Water	$4.4 \cdot 10^8$	4.1	$1.14 \cdot 10^8$	0.0

out by applying twice a filter among the domain. This filter is computed at each point as the average between its previous value and the average of the 27 cells in the neighborhood.

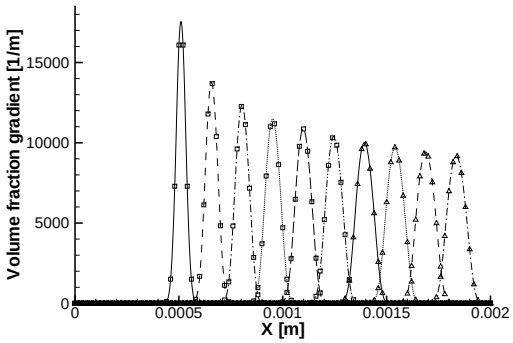
Figure 10 shows the density and $\frac{\partial \alpha_1}{\partial x}$ computed with a second-order central difference scheme at different time-steps with φ^2 equal to 0, 0.3 and 1. As expected, according to the derivation presented above, the interface keeps its gradient for $\varphi^2 \cong 0.3$.

Figure 11 shows the volume fraction gradient at different time steps for $u = 1.0\ m/s$ and $u = 100.0\ m/s$.

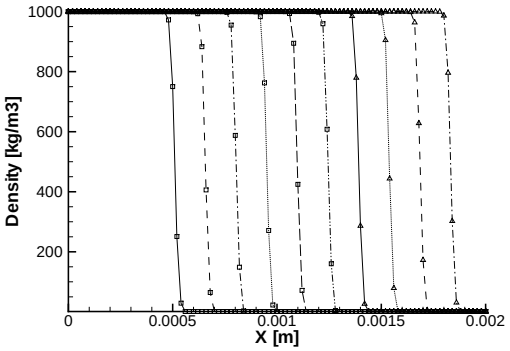
Figure 12 shows the density and volume fraction gradient for a grid with 1000 cells and same physical length. Having increased the number of cells 10 times respect to the previous cases raised the gradient 10 times (see Figs. 10(d) and 12(b)). This means that the number of cells used to represent the interface did not change, what is one of the purposes of the sharpening scheme.



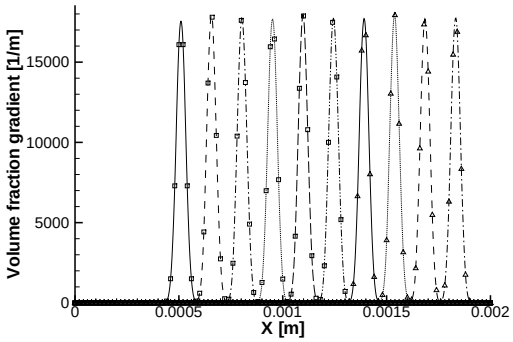
(a) Density with $\varphi^2 = 0.0$.



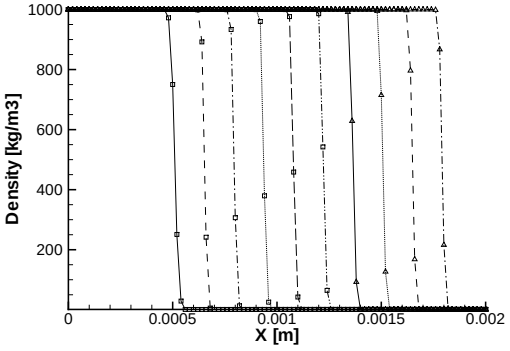
(b) $\frac{\partial \alpha_1}{\partial x}$ with $\varphi^2 = 0.0$, interface speed= 10.0 m/s.



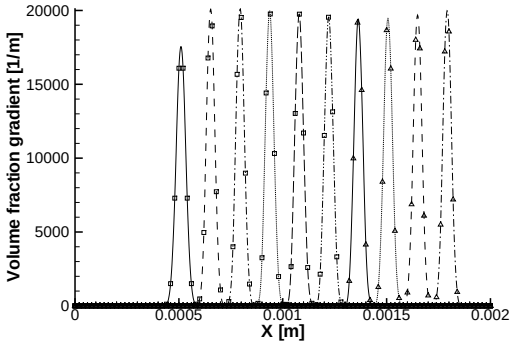
(c) Density with $\varphi^2 = 0.3$.



(d) $\frac{\partial \alpha_1}{\partial x}$ with $\varphi^2 = 0.3$, interface speed= 9.96 m/s.

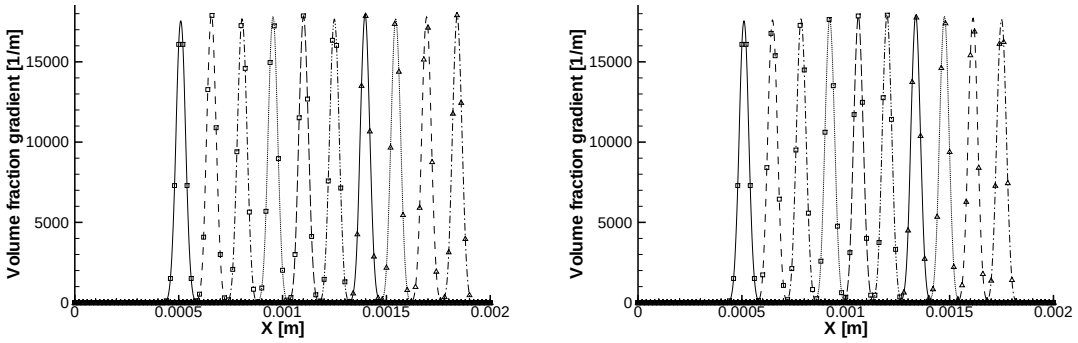


(e) Density with $\varphi^2 = 1.0$.



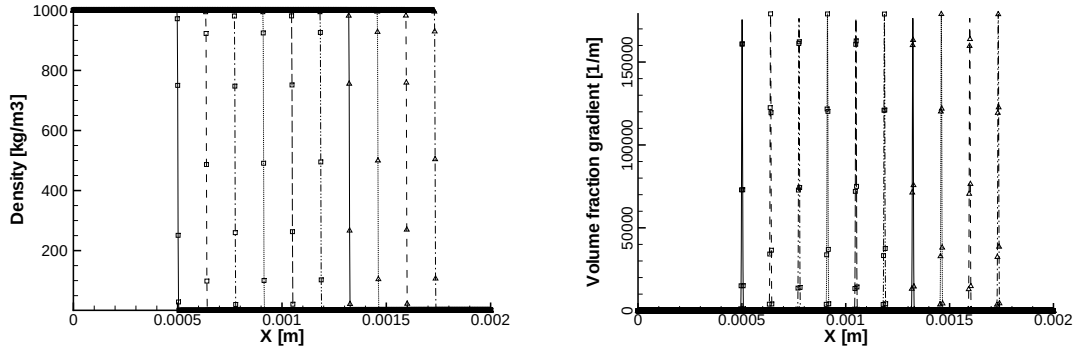
(f) $\frac{\partial \alpha_1}{\partial x}$ with $\varphi^2 = 1.0$, interface speed= 9.66 m/s.

Figure 10: Density and volume fraction gradient for different time-steps with $u = 10\ m/s$ and different values of φ.



(a) $u = 1.0\ m/s$, interface speed= $0.997\ m/s$. (b) $u = 100.0\ m/s$, interface speed= $99.7\ m/s$.

Figure 11: Volume fraction gradient for different time-steps with different velocities, $\varphi^2 = 0.3$ and 100 cells.

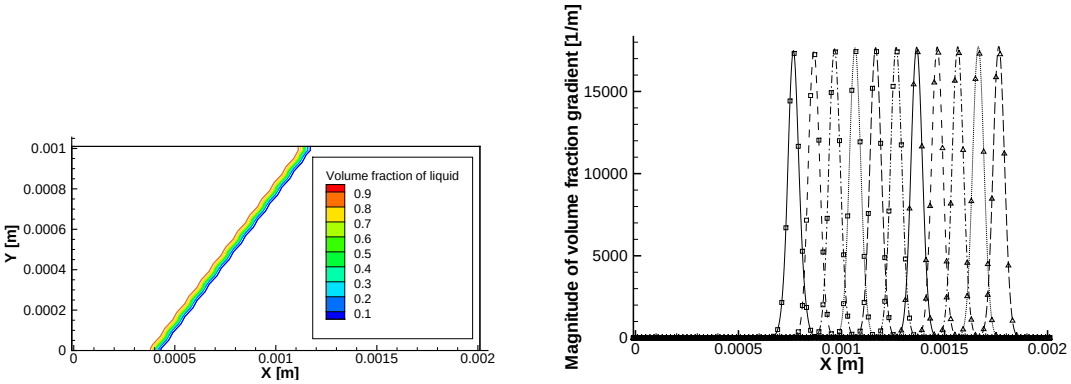


(a) Density with $\varphi^2 = 0.3$ and 1000 cells. (b) $\frac{\partial \alpha_1}{\partial x}$ with $\varphi^2 = 0.3$ and 1000 cells.

Figure 12: Volume fraction gradient for different time-steps with different velocities, $\varphi^2 = 0.3$ and 1000 cells.

5.2.2 Convection of a planar interface in 2-D

In comparison to 1-D cases, in two dimensions the orientation of the interface respect to the grid and the flow is not trivial. Therefore, now the scheme is tested in a $2\ mm \times 1\ mm$ domain, with 100×50 cells. The boundary conditions are outflow on the right, inflow on the left and slip-walls on top and bottom. The domain is initialized as shown in Fig. 13(a) with a velocity of 100 m/s to the right. Figure 13(b) shows the magnitude of volume fraction gradient at the centerline. This magnitude is constant and measured to be convected at 100.1 m/s, showing proper behavior of the scheme in 2-D for planar interfaces.



(a) Initial volume fraction of the liquid.

(b) Volume fraction gradient.

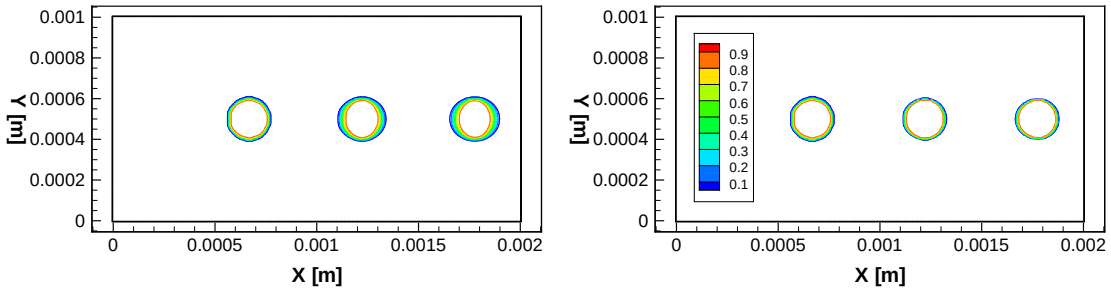
Figure 13: Planar interface being convected in a 2-D domain.

5.2.3 Convection of a liquid cylinder in 2-D

In order to test the scheme in two dimensions when curvature is present the convection of a liquid cylinder is analyzed. The domain is a rectangle of $2\ mm \times 1\ mm$ with 100×50 cells. The left boundary is an inflow with fixed velocity and temperature, the right limit is a subsonic outflow and the other two borders are symmetric boundaries. The cylinder has a radius $R_0 = 0.1\ mm$, its center is initially located at $x = 0.5\ mm$ and is filled with water, while the surrounding is air.

Three cases are conducted with and without sharpening. Figure 14(a) shows the

volume fraction contours at three different time steps without sharpening, presenting how the interface becomes more diffused in areas where the velocity is normal to the interface. Figure 14(b) shows the volume fraction contours at three different time steps with sharpening. With sharpening the size of the drop is constant but its interface gets diffused mostly where the normal to the interface is aligned with the velocity. Whereas, with sharpening, the thickness is kept constant, but the size of the drop reduces, representing loss of mass.



(a) No sharpening.

(b) With sharpening.

Figure 14: Volume fraction contours of a cylinder being convected in a Cartesian grid at $t = 0\ s$, $t = 10^{-5}\ s$ and $t = 2 \cdot 10^{-5}\ s$.

5.3 Discussion

The mathematical model does not provide a natural way to control the thickness of the interface. Thus, the numerical diffusion spreads out the interface significantly. A sharpening technique was suggested as an artifact to control thickness. The method only relies on a constant that is scheme dependent, while the rest of the technique is independent of the scheme. Even though it presents excellent properties in 1-D, in 2-D conservation properties are significantly violated. The methods suggested in the literature up to the time of the current research are not better than the method proposed here. Hence, sharpening techniques is a subject of current research. Probably, the best way to handle this issue so far is using a numerical scheme with minimal numerical diffusion and with localized adaptive mesh refinement, since the interface

thickness is proportional to the cell size.

CHAPTER VI

HIGH-ORDER DG IN SPACE AND TIME WITH AMR

6.1 Introduction

In this chapter the benefit of using high-order in space and time is evaluated. Knowing that the purpose of this new solver is mostly for applications with material interfaces, the cases in this chapter are focused on problems with discontinuities. Given that this type of problems are first-order accurate in spite of the order of the scheme, it is verified if the efficiency of the solver can be improved using AMR.

Three different rules to control the adaptivity were formulated above (see Section 4.2.5). Two are found in the literature and one is new. A series of test cases are used to compare them and determine which one is the most efficient, at least for the cases chosen.

In addition, discontinuous problems require limiters to prevent oscillations. Hence, the modified moment limiter with the MB-AP-TVD detector presented above are evaluated and compared with other methods to verify their improved efficiency. The variables used for the limiting stage affect the accuracy and computational speed, thus, they are included in the comparison.

6.2 Verification and validation

Various test cases used in past studies are used to establish the capability of this new numerical algorithm. In addition, some 2D and 3D cases are used to demonstrate the potential of the method for more complex problems. The details of the test cases and the rationals for them are summarized in Table 6.

Table 6: Summary of inviscid single phase cases for verification.

Test case	Purpose	Section
Order of convergence (linear)	Order of accuracy in space with a smooth linear problem.	6.2.1
Order of convergence (non-linear)	Order of accuracy in space with a smooth non-linear problem.	6.2.2
Accuracy in time	Order of accuracy in time with a smooth solution using RK and SDC.	6.2.3
Advection of mixed pulses	Order of accuracy with a non-smooth solution with and without detector.	6.2.4
Order for a smooth and non-smooth solution	Order of accuracy with a localized discontinuity.	6.2.5
Sod's problem	Limiting variables, with and without detector, and with and without AMR.	6.2.6
Lax's problem	Limiting variables, with and without detector, and with and without AMR.	6.2.7
Blast waves	Limiting variables, with and without detector, and with and without AMR.	6.2.8
Shock-entropy waves interaction	Limiting variables, with and without detector, and with and without AMR.	6.2.9
2D convection	Detector and AMR in 2D.	6.2.10
Double-Mach reflection	Example in 2D.	6.2.11
Vortex convection	Smooth example in 2D.	6.2.12
Shock-vortex interaction	Example in 2D.	6.2.13
Spherical shock test	Multi-dimensional symmetry (3D).	6.2.14

6.2.1 Order of convergence - linear

The order of convergence is studied with the one-dimensional convection equation because the exact solution is known:

$$\frac{\partial u(x,t)}{\partial t} + c\frac{\partial u(x,t)}{\partial x} = 0 \tag{134}$$

$$u(x, t = 0) = \sin(x)$$

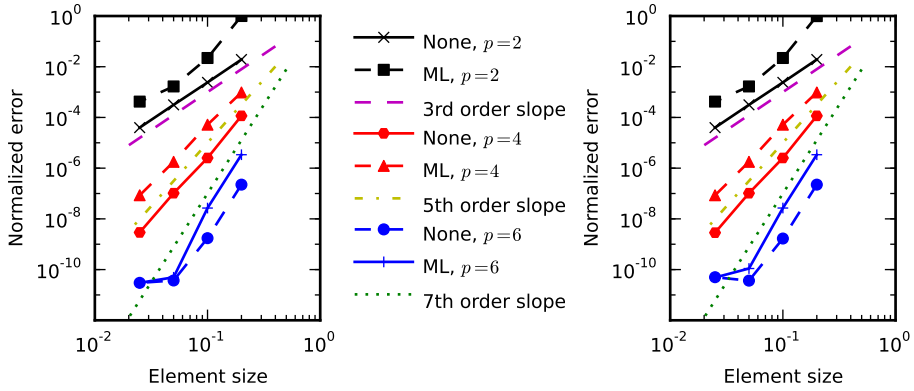
where u is a passive scalar and c is the constant convection velocity equal to 1. The exact solution is $u(x,t) = \sin(x - ct)$. The domain has a size of 2π long and periodic boundary conditions. The number of cells N and the polynomial order p are varied in this study.

This case is run without detector to show the effect of the limiters on smooth solutions. Moreover, given that the governing equation is not a system of equations, no characteristic decomposition is needed.

The time integration schemes used here are the SDC and the TVD-RK of 3^{rd} order with a time step of 10^{-5}. This gives an error in time of the order of 10^{-15}, which is negligible respect to the spatial error and of the order of the round-off error. The L_∞ error, e_{L_∞}, at $t = 2$ is computed at the centroid of the element and with respect to the exact solution, i.e.:

$$e_{L_\infty} = \max_{l=1,\ldots,N} |U_l(\bar{x}_l, t) - u(\bar{x}_l, t)| \tag{135}$$

where $\bar{x}_l$ is the centroid of element l. The L_∞ error in the plots is normalized by the case with the largest error. As shown in Fig. 15, elements of order p lead to an order of accuracy of $p + 1$, as the literature predicts [26]. Even though the solutions with limiter have the same order of accuracy, they have a greater error. Therefore, the limiter should not be used if it is not really needed. The curves in Fig. 15 get flattened out for very low errors (around $O(10^{-11})$) due to accumulated round-off error.



(a) 3^{rd}-order TVD-RK (b) 3^{rd}-order SDC

Figure 15: Grid convergence for different orders when a smooth solution is convected.

6.2.2 Order of convergence – non-linear

Now the order of convergence in space is studied with the one-dimensional burger equation as in [26]:

$$\frac{\partial u(x,t)}{\partial t} + \frac{\partial (u(x,t)^2/2)}{\partial x} = 0 \tag{136}$$

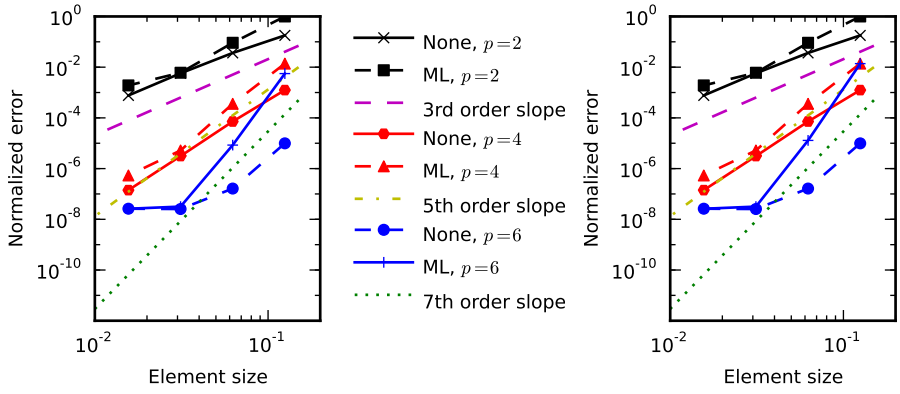
$$u(x, t=0) = 1/4 + 1/2 \ \sin(\pi(2x-1))$$

where u is the velocity. The domain has a unit length and periodic boundary conditions. The number of cells N and the polynomial order p are varied in this study. The exact solution is estimated with $N = 2048$, $p = 2$, 3^{rd}-order TVD-RK, and without limiter. The problem is solved until $t = 0.05$, when the solution is still smooth.

This case is run without detector to show the effect of the limiters on smooth solutions.

The time integration schemes used here are the SDC and the TVD-RK of 3^{rd} order with a time step of 10^{-5}. This gives an error in time of the order of 10^{-15}, which is negligible respect to the spatial error and of the order of the round-off error. The L_∞ error, e_{L_∞}, at $t = 0.05$ is computed at the centroid of the element and with respect to the estimated exact solution as in the previous case.As shown in Fig. 16,

the order of accuracy in space matches closely with what the theory predicted even for a non-linear problem. Even though the solutions with limiter have the same order



(a) 3^{rd}-order TVD-RK (b) 3^{rd}-order SDC

Figure 16: Grid convergence for different orders with a smooth non-linear problem.

of accuracy, they have a greater error. Therefore, the limiter should not be used if it is not really needed. The curves in Fig. 16 get flattened out for very low errors due to accumulated round-off error. In conclusion, the observations for the non-linear case are very similar to the linear case above.

6.2.3 Accuracy in time

The time integration is studied with the following equation:

$$\begin{aligned} \frac{\partial u(x,t)}{\partial t} + 0\frac{\partial u(x,t)}{\partial x} &= u(x,t) \\ u(x,0) &= 1 \end{aligned} \tag{137}$$

which has the exact solution $u(x,t) = e^t$. The convection velocity is 0 so that the truncation error in space is zero and the truncation error in time can be studied by itself. A 1-D domain of unit length, periodic boundaries and 100 elements is used. The time integration is performed with the TVD-RK and SDC methods for different order. The L_∞ error is computed at $t = 6.28$ for different number of time steps and shown in Tables 7 – 9 along with the order of accuracy. The results are also shown in

Fig. 17 for a more clear appreciation. The order of accuracy for N_i elements (knowing that $N_i = 2N_{i-1}$) is computed as:

$$\frac{\log(e_i/e_{i-1})}{\log(0.5)}. \tag{138}$$

Table 7: e_{L_∞} for TVD-RK.

Number of time-steps	2^{nd} order e_{L_∞}	Order	3^{nd} order e_{L_∞}	Order
8	$1.6537 \cdot 10^{2}$	-	$3.5302 \cdot 10^{1}$	-
16	$6.0804 \cdot 10^{1}$	1.4435	$6.1493 \cdot 10^{0}$	2.5213
32	$1.8276 \cdot 10^{1}$	1.7342	$9.0205 \cdot 10^{-1}$	2.7691
64	$4.9757 \cdot 10^{0}$	1.8770	$1.2200 \cdot 10^{-1}$	2.8863
128	$1.2948 \cdot 10^{0}$	1.9422	$1.5861 \cdot 10^{-2}$	2.9434
256	$3.2999 \cdot 10^{-1}$	1.9722	$2.0219 \cdot 10^{-3}$	2.9717

Table 8: e_{L_∞} for 2^{nd} and 3^{rd} order SDC.

Number of time-steps	2^{nd} order e_{L_∞}	Order	3^{nd} order e_{L_∞}	Order
8	$1.6537 \cdot 10^{2}$	-	$1.9510 \cdot 10^{1}$	-
16	$6.0804 \cdot 10^{1}$	1.4435	$2.8648 \cdot 10^{0}$	2.7677
32	$1.8276 \cdot 10^{1}$	1.7342	$3.7984 \cdot 10^{-1}$	2.9150
64	$4.9757 \cdot 10^{0}$	1.8770	$4.8588 \cdot 10^{-2}$	2.9667
128	$1.2948 \cdot 10^{0}$	1.9422	$6.1332 \cdot 10^{-3}$	2.9859
256	$3.2999 \cdot 10^{-1}$	1.9722	$7.7005 \cdot 10^{-4}$	2.9936

The fact that the computed order approaches the order of the scheme verifies the proper implementation of the temporal integration. Also, note that at equal theoretical order, SDC results to be more accurate while they have very similar order of accuracy.

Figure 18 shows the CPU time against the e_{L_∞} obtained for different orders and schemes. The CPU time is normalized by the fastest case. The curves closer to the

Table 9: e_{L_∞} error for 4^{th} and 5^{th} order SDC.

Number of time-steps	4^{nd} order e_{L_∞}	Order	5^{nd} order e_{L_∞}	Order
8	$1.2840 \cdot 10^{0}$	-	$7.2084 \cdot 10^{-2}$	-
16	$9.2132 \cdot 10^{-2}$	3.8007	$2.4229 \cdot 10^{-3}$	4.8949
32	$6.0812 \cdot 10^{-3}$	3.9213	$7.7303 \cdot 10^{-5}$	4.9700
64	$3.8890 \cdot 10^{-4}$	3.9669	$2.4288 \cdot 10^{-6}$	4.9922
128	$2.4556 \cdot 10^{-5}$	3.9852	$7.5987 \cdot 10^{-8}$	4.9984
256	$1.5422 \cdot 10^{-6}$	3.9931	$2.3588 \cdot 10^{-9}$	5.0096

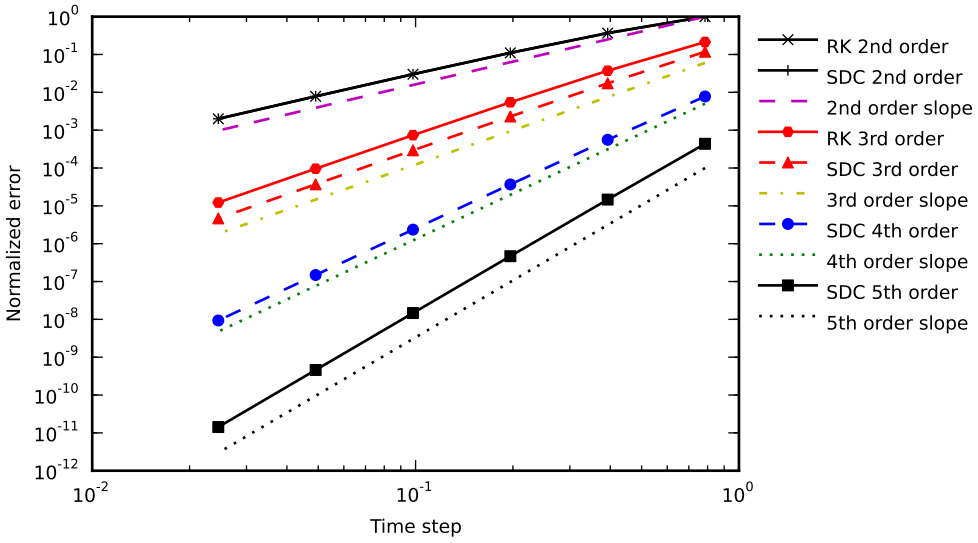


Figure 17: Normalized error for TVD-RK and SDC.

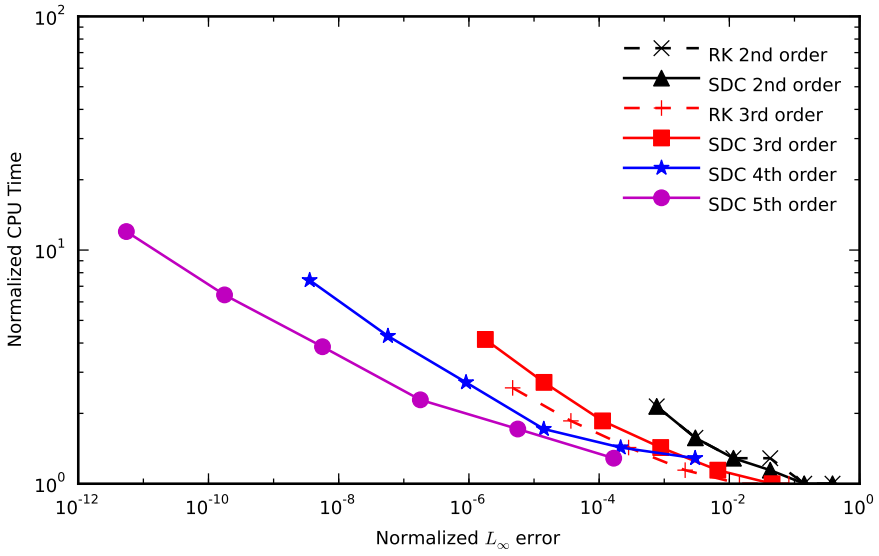


Figure 18: CPU time for different time integration schemes and orders.

bottom left corner represent a more efficient scheme. For the same order, TVD-RK is more efficient than SDC. At the same time, the efficiency is increased with the order. For instance for this case 5^{th} order SDC is more efficient than 3^{rd} order TVD-RK.

In conclusion, the advantage of SDC respect to RK is that the extension to higher orders is trivial. One could argue that SDC does not have the TVD property of TVD-RK of 2^{nd} and 3^{rd} order, however, RK schemes of 4^{th} order or greater are not TVD either. For certain problems where the error in time is important higher orders may be more suitable. Thus, explicit SDC seems to be a possible approach for high-order time integration of DG schemes.

The test cases below tend to have a dominant spatial error, thus very high orders in time are not required.

6.2.4 Advection of mixed pulses

The convection equation (Eq. 134) is used with the initial value given by:

$$u(x,t=0)=\begin{cases}\frac{1}{6}\left(G(x,\beta,z-\delta)+G(x,\beta,z+\delta)+4G(x,\beta,z)\right) & -0.8\leq x\leq -0.6\\ 1 & -0.4\leq x\leq -0.2\\ 1-|10(x-0.1)| & 0\leq x\leq 0.2\\ \frac{1}{6}\left(F(x,\alpha,a-\delta)+F(x,\alpha,a+\delta)+4F(x,\alpha,z)\right) & 0.4\leq x\leq 0.6\\ 0 & \text{otherwise}\end{cases}$$

$$G(x,\beta,z)=e^{-\beta(x-z)^2}$$

$$F(x,\alpha,a)=\sqrt{\max(1-\alpha^2(x-a)^2,0)}$$

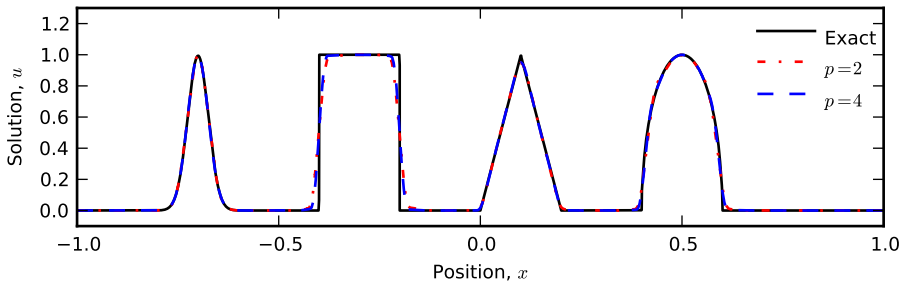
where $a=0.5$, $z=-0.7$, $\delta=0.005$, $\alpha=10$, and $\beta=\log 2/\left(36\delta^2\right)$. The domain is a uniform grid from $x=-1$ to $x=1$ with periodic boundary conditions.

The SDC method of 3^{rd} order is used. The ML is applied on every element or on the ones flagged by the MB-AP-TVD detector. The result at $t=8.0$ is shown in Fig. 19 for $p=2,4$ and for 200 cells.

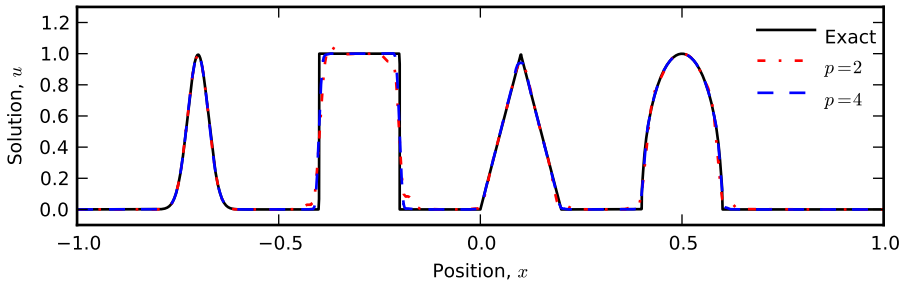
Figure 20 shows the L_1 error computed at the center of the elements for different number of cells and polynomial orders:

$$e_{L_1}=\sum_{l=1}^{N}\int_{\Omega_l}|U_l(x,t)-u(x,t)|dx \tag{139}$$

where U is the numerical result and u is the exact solution (or its estimation).In the previous test case L_∞ was used, which is an adequate parameter to analyze smooth solution, however, for discontinuous solutions L_1 is more appropriate. A few observations can be made from this figure. The error is reduced as the number of elements n or the polynomial order p increases. Also, using the detector improves the accuracy. Note that the order of accuracy is approximately 1 because of the presence of discontinuities in the solution. Therefore, increasing the order p when discontinuities are present reduces the error, but not the order of accuracy.



(a) ML applied to all cells



(b) ML applied only to "troubled" cells flagged by the detector

Figure 19: Convection of mixed pulses at $t = 8$, with 200 cells and $p = 2, 4$.

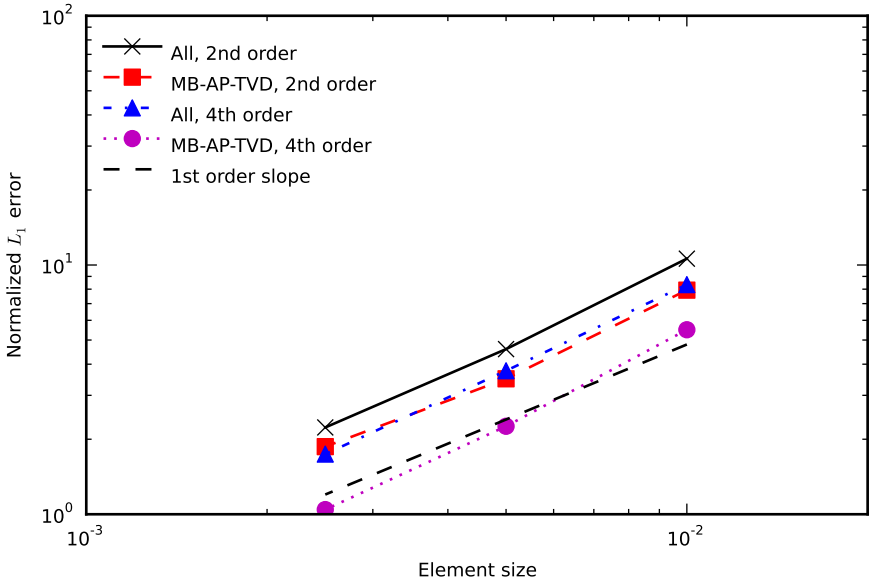


Figure 20: L_1 error for different orders and number of elements for the convection of mixed pulses.

The same case is run using the original AP-TVD detector, and the efficiency of the AP-TVD and MB-AP-TVD detectors are compared in Figure 21. Curves closer to the bottom left corner represent more accurate schemes. Note that for the same

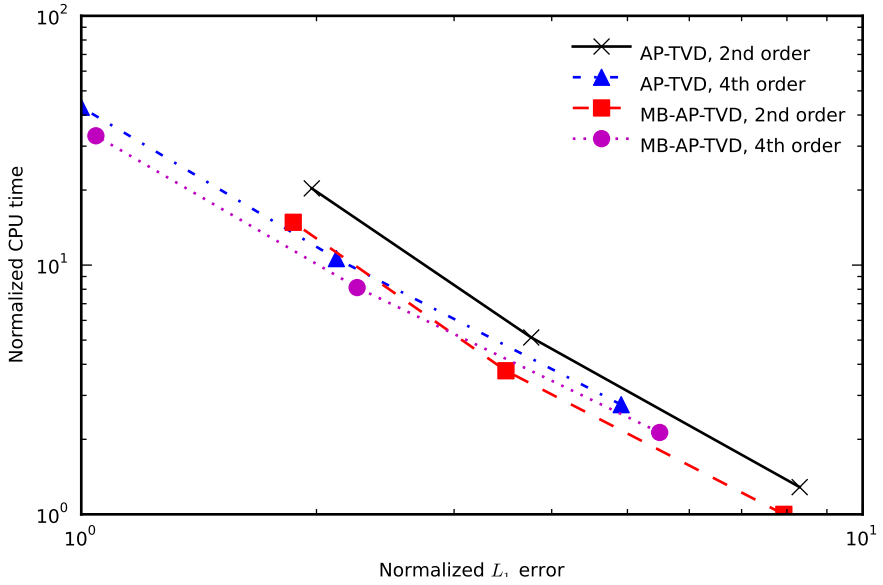


Figure 21: CPU time vs. L_1 error for different orders for the convection of mixed pulses.

number of elements AP-TVD tends to be slower, while the error is similar. Thus, MB-AP-TVD ends up being a better choice than AP-TVD when the default basis of the element is formed by Legendre polynomials. Given this, we only use the MB-AP-TVD for the cases below. In addition, figure 21 shows that for this case, which contains discontinuities, increasing the order of the scheme does not improve its efficiency.

6.2.5 High order for a smooth and non-smooth solution

The same case as in Section 6.2.2 is observed here for a longer period of time. At $t = 0.4$ a discontinuity is found at approximately $x = 0.1$, while the rest of the solution is smooth. The error is usually computed taking into account the whole domain. However, in order to analyze only the region with a smooth solution, it can

be computed for part of the domain. For this purpose we define $\tilde{e}_{L_1}$:

$$\tilde{e}_{L_1} = \sum_{l=1,\ 0.3\leq x\leq 0.9}^{N} \int_{\Omega_l} |U_l(x,t) - u(x,t)| dx \tag{140}$$

This similar to Eq. 139, but the integration is done away from the discontinuity, i.e. for $0.3 \leq x \leq 0.9$. We estimate the exact solution with 512 elements with $p = 6$, and using the SDC of 7^{th} order. The problem is studied using $p = 2, 4, 6$, $N = 10, 20, 30, 40, 80, 160$, SDC of order $p + 1$, and limiting on all the elements or as flagged by the MB-AP-TVD detector. The errors e_{L_1} and $\tilde{e}_{L_1}$ are shown in Tables 10-15. Note that the the order of accuracy based on $\tilde{e}_{L_1}$ is close to $p + 1$, while for e_{L_1} it is close 1. At very low errors the order drops due to accumulated round-off error. For $p = 4$ and $p = 6$ using 20 elements the order is much greater then $p + 1$ since the error for $N = 10$ is relatively large. This is due to the propagation to smooth regions of instabilities generated at the discontinuity. For large-enough number of elements, $N \geq 20$, the instabilities do not affect the smooth area being considered in Eq. 140 for $\tilde{e}_{L_1}$.

Table 10: e_{L_1} and $\tilde{e}_{L_1}$ for $p = 2$ and limiting on all the elements.

Number of elements	Whole domain e_{L_1}	Order	Smooth region $\tilde{e}_{L_1}$	Order
10	$3.0333 \cdot 10^{-2}$	-	$4.4679 \cdot 10^{-4}$	-
20	$1.2243 \cdot 10^{-2}$	1.3089	$2.8444 \cdot 10^{-5}$	3.9734
30	$7.6765 \cdot 10^{-3}$	1.1513	$6.2605 \cdot 10^{-6}$	3.7332
40	$5.6299 \cdot 10^{-3}$	1.0778	$2.2165 \cdot 10^{-6}$	3.6093
80	$2.7534 \cdot 10^{-3}$	1.0319	$1.9190 \cdot 10^{-7}$	3.5298
160	$1.3740 \cdot 10^{-3}$	1.0029	$1.7365 \cdot 10^{-8}$	3.4661

Figure 22 shows the efficiency of the scheme for different orders, with and without the MB-AP-TVD detector. The error and CPU time are normalized based on the case with the largest error. As observed for previous cases, the troubled-cell detector

Table 11: e_{L_1} and $\tilde{e}_{L_1}$ for $p = 2$ and limiting based on the MB-AP-TVD detector.

Number of elements	Whole domain e_{L_1}	Order	Smooth region $\tilde{e}_{L_1}$	Order
10	$2.7784 \cdot 10^{-2}$	-	$1.5107 \cdot 10^{-4}$	-
20	$9.7931 \cdot 10^{-3}$	1.5044	$9.2714 \cdot 10^{-6}$	4.0263
30	$4.4951 \cdot 10^{-3}$	1.9205	$2.2911 \cdot 10^{-6}$	3.4477
40	$4.5546 \cdot 10^{-3}$	$-4.5661 \cdot 10^{-2}$	$8.4867 \cdot 10^{-7}$	3.4521
80	$1.5292 \cdot 10^{-3}$	1.5745	$8.4769 \cdot 10^{-8}$	3.3236
160	$6.5156 \cdot 10^{-4}$	1.2308	$9.0416 \cdot 10^{-9}$	3.2289

Table 12: e_{L_1} and $\tilde{e}_{L_1}$ for $p = 4$ and limiting on all the elements.

Number of elements	Whole domain e_{L_1}	Order	Smooth region $\tilde{e}_{L_1}$	Order
10	$2.6758 \cdot 10^{-2}$	-	$5.0581 \cdot 10^{-5}$	-
20	$1.1347 \cdot 10^{-2}$	1.2376	$1.3111 \cdot 10^{-8}$	11.914
30	$7.2367 \cdot 10^{-3}$	1.1093	$1.2282 \cdot 10^{-9}$	5.8399
40	$5.3411 \cdot 10^{-3}$	1.0558	$2.3009 \cdot 10^{-10}$	5.8220
80	$2.6630 \cdot 10^{-3}$	1.0041	$5.1326 \cdot 10^{-12}$	5.4864
160	$9.1757 \cdot 10^{-4}$	1.5372	$1.7473 \cdot 10^{-13}$	4.8765

Table 13: e_{L_1} and $\tilde{e}_{L_1}$ for $p = 4$ and limiting based on the MB-AP-TVD detector.

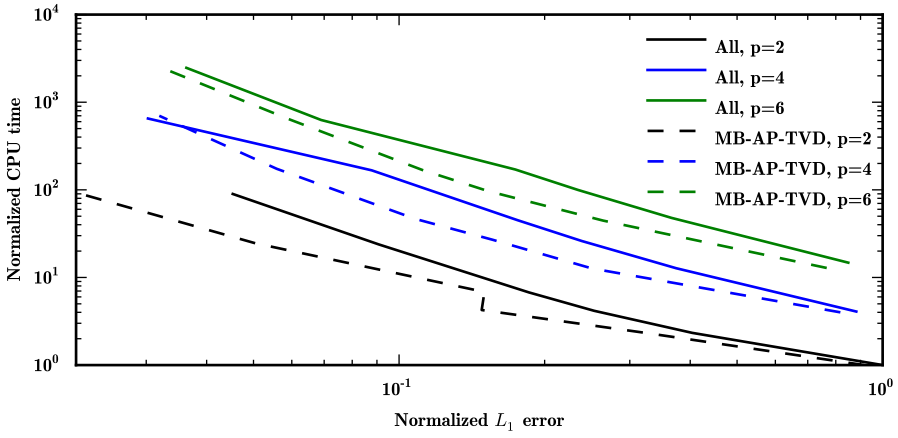
Number of elements	Whole domain e_{L_1}	Order	Smooth region $\tilde{e}_{L_1}$	Order
10	$2.4803 \cdot 10^{-2}$	-	$4.2981 \cdot 10^{-5}$	-
20	$7.6011 \cdot 10^{-3}$	1.7063	$6.3106 \cdot 10^{-9}$	12.7336
30	$4.7370 \cdot 10^{-3}$	1.1663	$6.1355 \cdot 10^{-10}$	5.7483
40	$3.2698 \cdot 10^{-3}$	1.2885	$1.2682 \cdot 10^{-10}$	5.4799
80	$1.6939 \cdot 10^{-3}$	0.9489	$3.3972 \cdot 10^{-12}$	5.2223
160	$9.6828 \cdot 10^{-4}$	0.8068	$1.4398 \cdot 10^{-13}$	4.5604

Table 14: e_{L_1} and $\tilde{e}_{L_1}$ for $p = 6$ and limiting on all the elements.

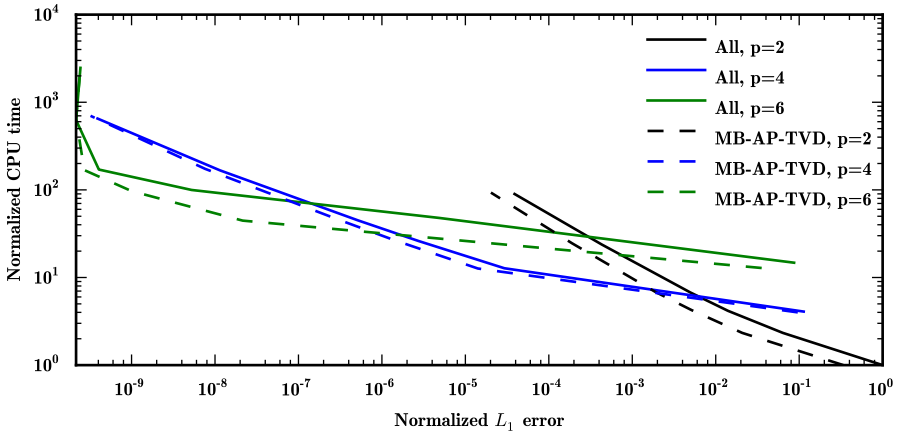
Number of elements	Whole domain e_{L_1}	Order	Smooth region $\tilde{e}_{L_1}$	Order
10	$2.5755 \cdot 10^{-2}$	-	$3.8820 \cdot 10^{-5}$	-
20	$1.1102 \cdot 10^{-2}$	1.2140	$2.1739 \cdot 10^{-9}$	14.1242
30	$7.1097 \cdot 10^{-3}$	1.0991	$2.3454 \cdot 10^{-12}$	16.8493
40	$5.2764 \cdot 10^{-3}$	1.0366	$1.8132 \cdot 10^{-13}$	8.8985
80	$2.0933 \cdot 10^{-3}$	1.3338	$9.6648 \cdot 10^{-14}$	0.9077
160	$1.1018 \cdot 10^{-3}$	0.9260	$1.0902 \cdot 10^{-13}$	-0.1738

Table 15: e_{L_1} and $\tilde{e}_{L_1}$ for $p = 6$ and limiting based on the MB-AP-TVD detector.

Number of elements	Whole domain e_{L_1}	Order	Smooth region $\tilde{e}_{L_1}$	Order
10	$2.3243 \cdot 10^{-2}$	-	$1.5876 \cdot 10^{-5}$	-
20	$8.0117 \cdot 10^{-3}$	1.5366	$9.5822 \cdot 10^{-12}$	20.6600
30	$4.6348 \cdot 10^{-3}$	1.3498	$4.4873 \cdot 10^{-13}$	7.5500
40	$3.3924 \cdot 10^{-3}$	1.0848	$1.2121 \cdot 10^{-13}$	4.5497
80	$1.7780 \cdot 10^{-3}$	0.9321	$9.6667 \cdot 10^{-14}$	0.3265
160	$9.5344 \cdot 10^{-4}$	0.8990	$1.0951 \cdot 10^{-13}$	-0.1799



(a) For the whole domain, e_{L_1}.



(b) For the smooth region, $\tilde{e}_{L_1}$.

Figure 22: Efficiency of the scheme for different orders for a solution with one discontinuity. The limiter is applied to all the element or based on the MB-AP-TVD detector.

helps improve the accuracy and efficiency of the solver. For lower L_1 error high-order schemes become more efficient. The limiter reduces numerical oscillations at discontinuities, but with a minimal numerical diffusion, so small instabilities still exist. As the number of elements is increased the numerical error originated at the discontinuity is localized in a smaller region. Thus, probably, p-adaptivity could improve the efficiency by dropping the order at the discontinuity and keeping high order in the smooth region.

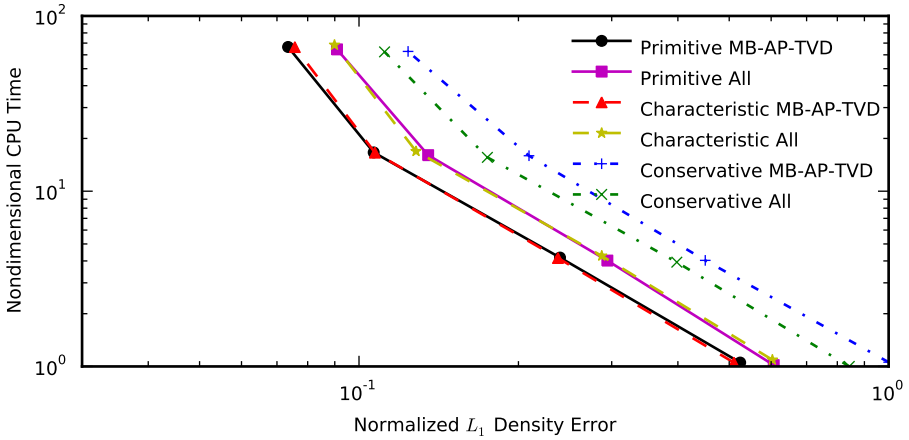
6.2.6 Sod's problem

The initial conditions are:

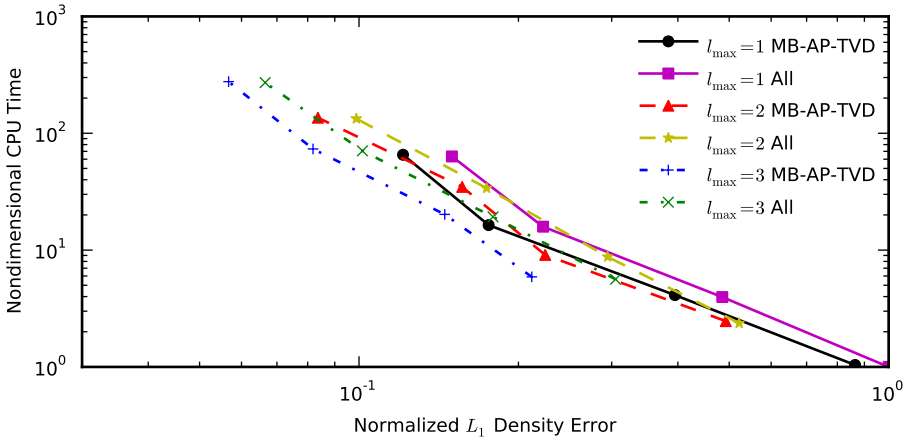
$$(\rho, v, p) = \begin{cases} (1.0, 0.0, 1.0)\,, & x \leq 0.5 \\ (0.125, 0.0, 0.1)\,, & x > 0.5 \end{cases} \tag{141}$$

A 1-D domain is used and it extends from $x = 0$ to $x = 1$. The case is run with different number of elements and the limiting is based on conservative, primitive, or characteristic variables. In addition, two options are tested, one applies the ML with the MB-AP-TVD detector, and the second option applies the ML to all cells. The grid is uniform with $p = 2$, and the time integration is performed using the 3^{rd}-order SDC. The simulation is run until $t = 0.2$. The normalized CPU time vs. the L_1 error of the final density is shown for the three cases in Fig. 23(a). Limiting with primitive or characteristic variables requires computing the respective Jacobians for each element, so it is computationally slightly more expensive than using conservative variables, but the error is lower. For primitive and characteristic variables, using the MB-AP-TVD detector to apply the ML to only "troubled" cells increases the efficiency and lowers the error since the solution is smooth in a large portion of the domain.

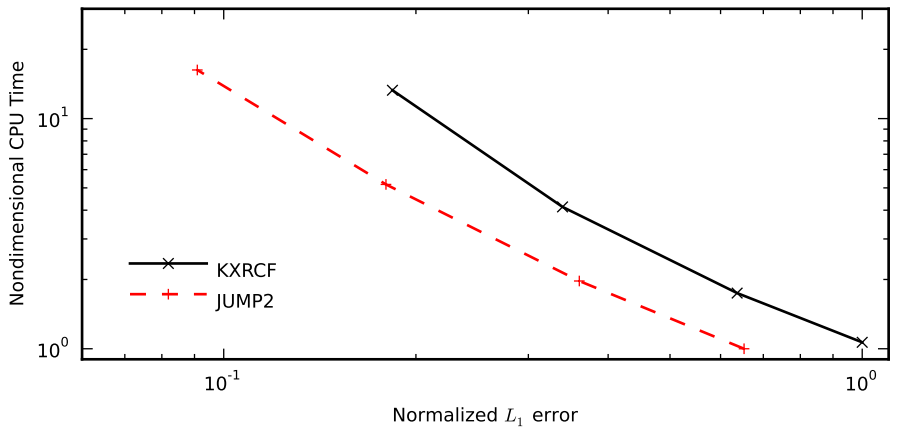
The same case is run enabling the adaptive mesh refinement for $l_{max} = 1, 2, 3$. The CPU time vs. the L_1 error of the density is shown in Fig. 23(b) for limiting with primitive variables. Note that both variables are normalized by the fastest simulation.



(a) Comparison for different limiting variables with and without detector.



(b) AMR comparison, limiting using primitive variables.

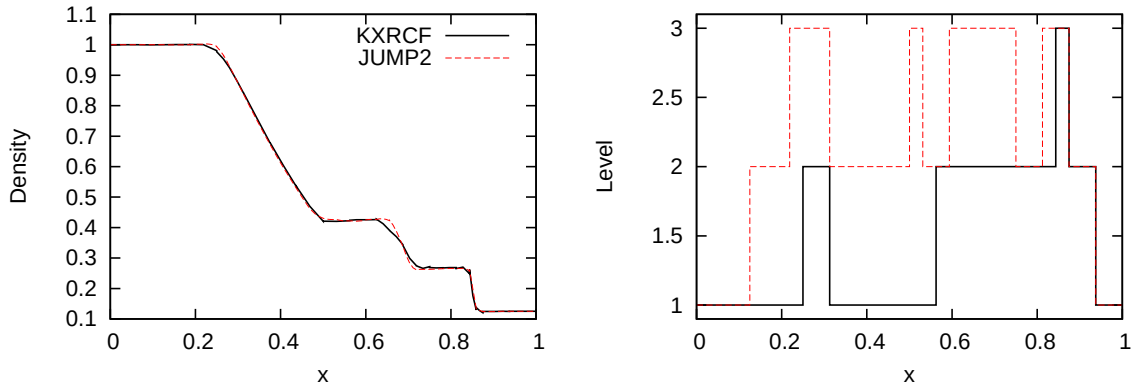


(c) Estimator comparison, limiting using primitive variables, the MB-AP-TVD detector and $l_{\max} = 3$.

Figure 23: Sod's problem for different solver options. The curves closer to the bottom left corner represent a more efficient set of options.

As l_{max} is increased the curves get slightly closer to the origin. This means that for this test case enabling the adaptivity produces some increase in the efficiency of the solver. Here, the MB-AP-TVD also shows to improve the efficiency.

The estimators are compared using $l_{max} = 3$, the MB-AP-TVD detector, and limiting with primitive variables. The efficiency is represented in Fig. 23(c). Clearly, JUMP2 is more efficient than KXRCF for this case. Figure 24 shows the density and number of levels for 16 elements. The main issue with KXRCF is that it does not refine properly at the contact discontinuity.



(a) Density. (b) Levels.

Figure 24: Sod's problem for different solver options. Contrast between the error estimators.

6.2.7 Lax's problem

The initial solution is:

$$(\rho, v, p) = \begin{cases} (0.445, 0.698, 3.528)\,, & x \leq 0 \\ (0.5, 0, 0.571)\,, & x > 0 \end{cases} \tag{142}$$

The problem is solved in the 1-D domain $[-0.5, 0.5]$ until $t = 0.13$.

Initially, the effect of the variables used for limiting is studied. A uniform grid is used with $N = 64, 128, 256, 512$ and $p = 2$ with the limiter applied to either all cells or those flagged by the MB-AP-TVD detector. The integration in time is done with

the 3^{rd}-order SDC method. The CPU time vs. the L_1 error of the density is shown in Fig. 25(a). These results show that for this particular test problem the MB-AP-TVD detector increases the efficiency for conservative and characteristic variables, while for primitive variables it did not affect significantly. The CPU time is very similar independent of the set of variables used. Even though conservative variables do not require to compute the Jacobian to transform between the variables, they may require more steps of the limiter.

Now the effect of the adaptation is studied, limiting with primitive variables. The same grid is used, but the adaptation is enable with $l_{max} = 1, 2, 3$. The result is shown in Fig. 25(b). When l_{max} is raised, the efficiency of the solver increases and it improves more using the MB-AP-TVD detector. The estimators are compared using $l_{max} = 3$, the MB-AP-TVD detector, and limiting with primitive variables. The efficiency is represented in Fig. 25(c). Clearly, JUMP2 is more efficient than KXRCF for this case.

6.2.8 Blast waves

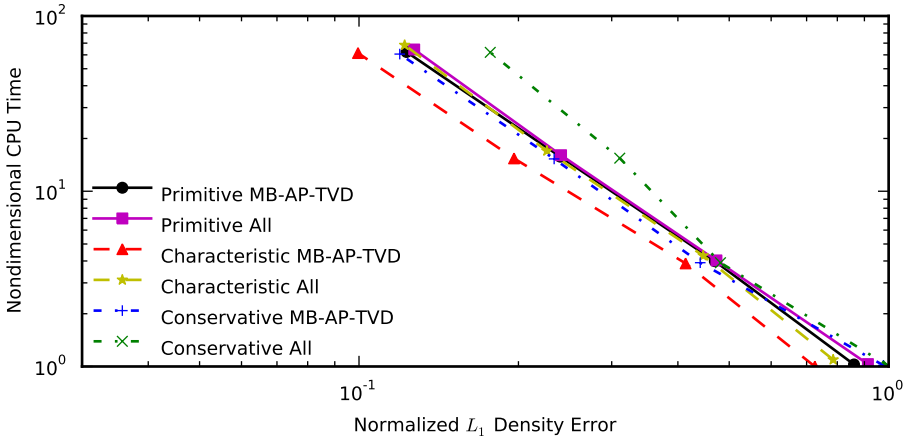
Consider the initial data $\rho = 1.0$, $v = 0.0$, and

$$P = \begin{cases} 1000, & 0 \leq x < 0.1, \\ 0.01, & 0.1 \leq x < 0.9, \\ 100, & 0.9 \leq x \leq 1.0, \end{cases} \tag{143}$$

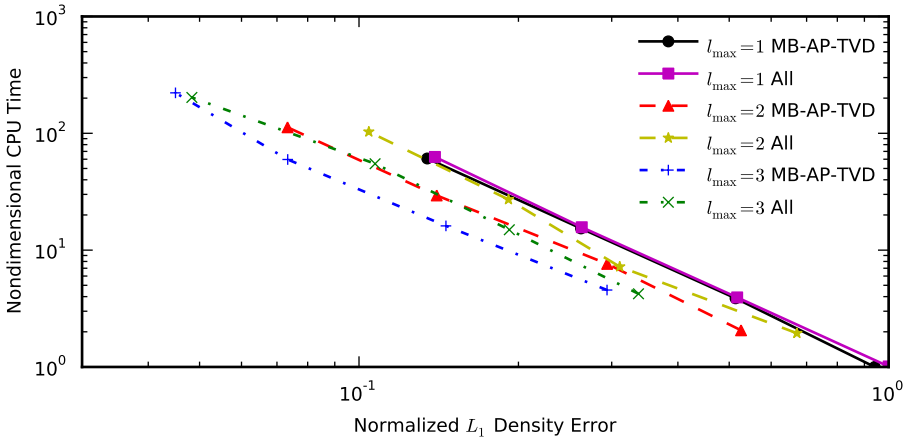
This problem is a common test case first presented in [144]. Walls are located at $x = 0$ and $x = 1$.

The problem is run until $t = 0.038$ s for $p = 2$, $l_{max} = 1, 2, 3$, different number of root cells and the 3^{rd}-order SDC. This test case does not have an exact solution, so it is approximated using a uniform mesh with $N = 4096$, $p = 2$, $l_{max} = 1$, the ML without detector and with characteristic decomposition, similar to [74].

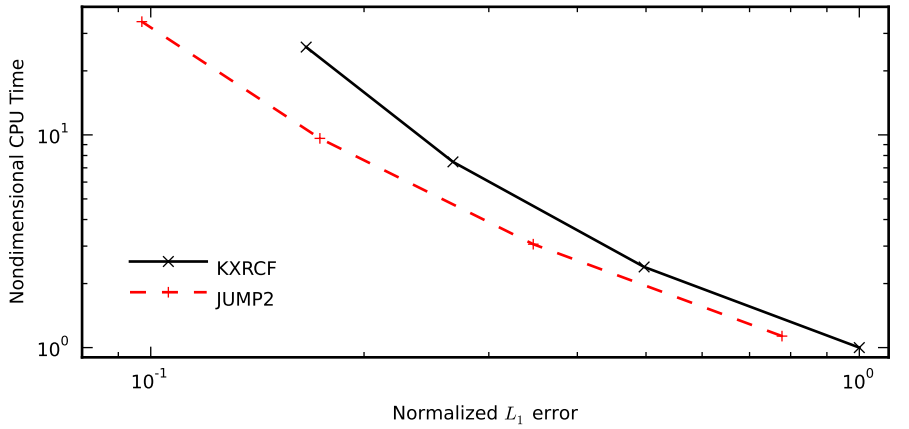
Figure 26 shows the CPU time vs. the L_1 error in density, both variables are



(a) Limiting comparison.



(b) AMR comparison, limiting using primitive variables.



(c) Estimator comparison, limiting using primitive variables, the MB-AP-TVD detector and $l_{\max} = 3$.

Figure 25: Lax problem for different solver options.

normalized by the fastest run. Figure 26(a) shows the effect of the detector and the limiting variables. Figure 26(b) represents the efficiency of the AMR approach using primitive variables. The efficiency of the solver clearly improves increasing l_{max}. In this case the MB-AP-TVD detector does not produce any significant difference when studying the refinement aspects. The estimators are compared using $l_{max} = 3$, the MB-AP-TVD detector, and limiting with primitive variables and the results are in Fig. 26(c). JUMP2 tends to be more efficient than KXRCF for this case.

6.2.9 Shock-entropy wave interaction

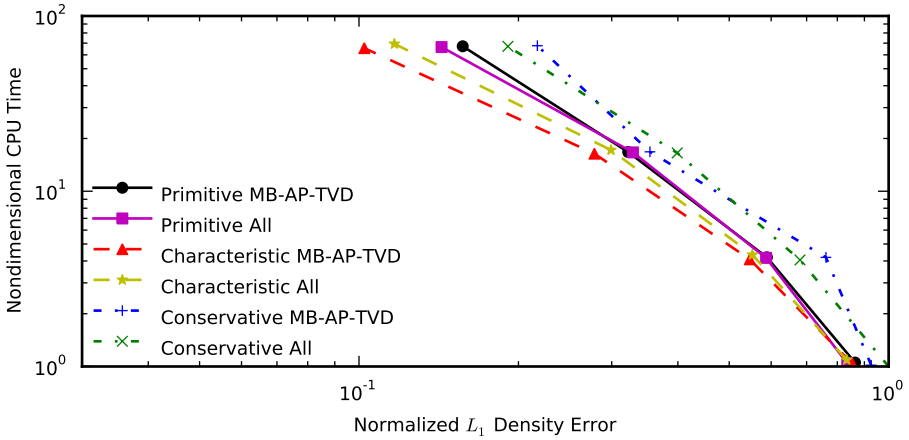
Consider the Euler equation with the following initial values:

$$(\rho, v, p) = \begin{cases} (3.857143, 2.629369, 10.333333), & x < -4 \\ (1.0 + 0.2\sin(5x), 0.0, 1.0), & x \geq -4 \end{cases} \tag{144}$$

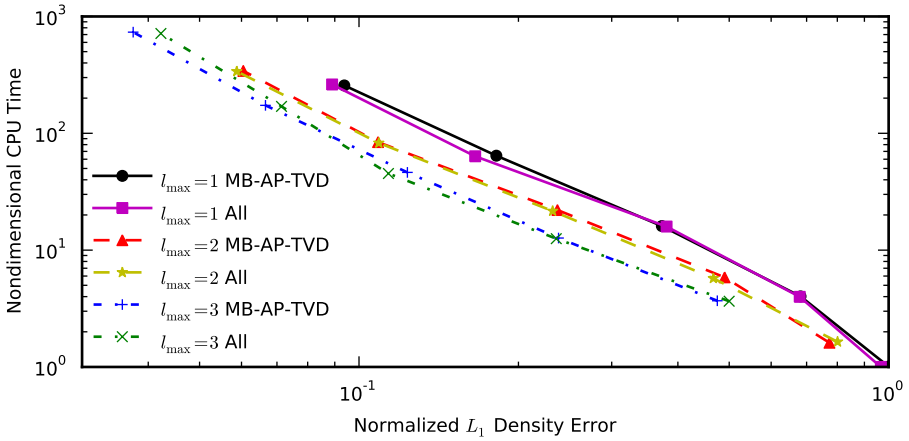
The problem is solved in the 1-D domain $[-5, 5]$ until $t = 1.8$.

As before, the effect of the variables used for limiting are studied on a uniform grid with $N = 64, 128, 256, 512$ and $p = 2$. The integration in time is done with the 3^{rd}-order SDC method. The CPU time vs. the L_1 error of the density is shown in Fig. 27. For this problem limiting using primitive variable is advantageous compared with conservative variables. Figure 28 shows the solution at $t = 1.8$ and it clearly presents that limiting using primitive variables captures the smooth oscillations much more accurately than with conservative variables. Characteristic limiting provides a even more efficient solution than with primitive variables. Also, the MB-AP-TVD detector clearly improves the efficiency. Using AMR for this test case gives no efficiency gains in the low element count (larger normalized error) regime, but AMR is more justified at lower errors where the number of elements increases.

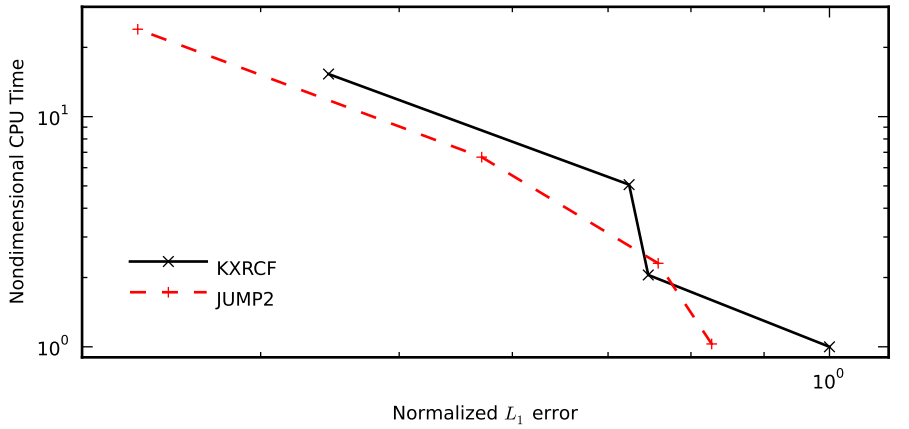
The estimators are compared in Fig. 27(c) using $l_{max} = 3$, the MB-AP-TVD detector, and limiting with primitive variables. JUMP2 tends to be more efficient than KXRCF for this case.



(a) Limiting comparison.

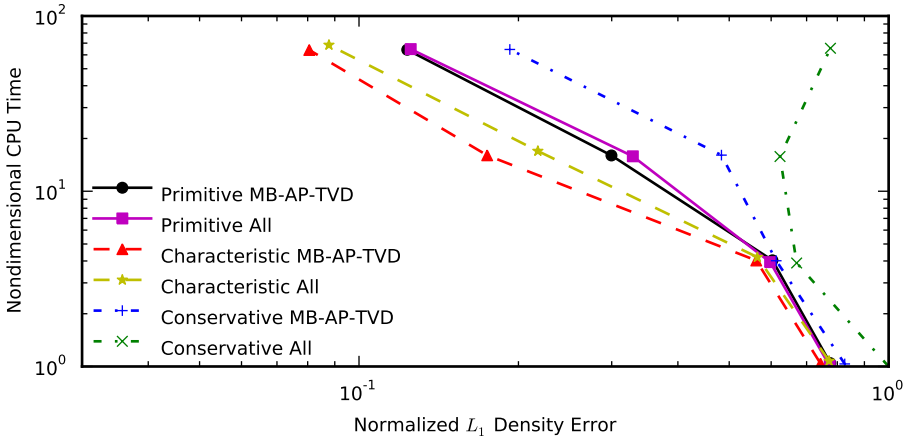


(b) AMR comparison, limiting using primitive variables.

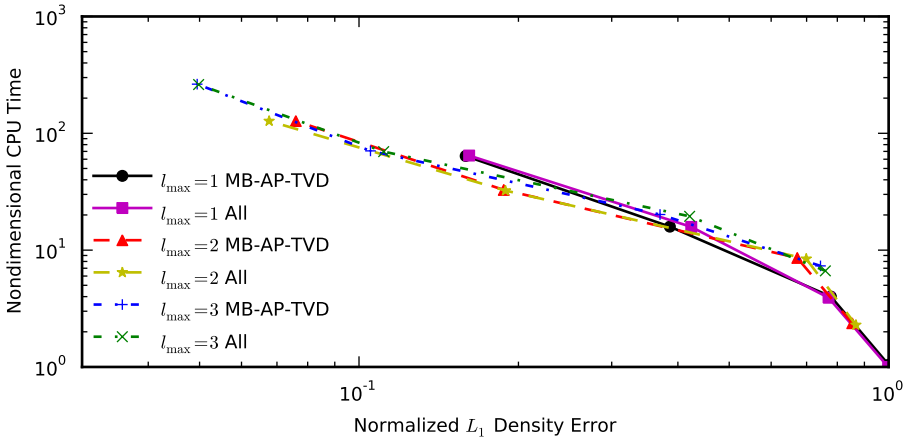


(c) Estimator comparison, limiting using primitive variables, the MB-AP-TVD detector and $l_{\max} = 3$.

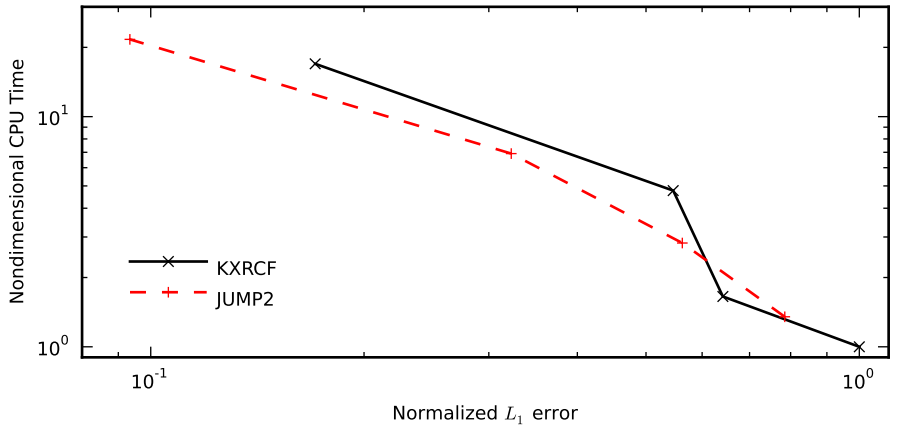
Figure 26: Interacting blast waves for different solver options.



(a) Limiting comparison.

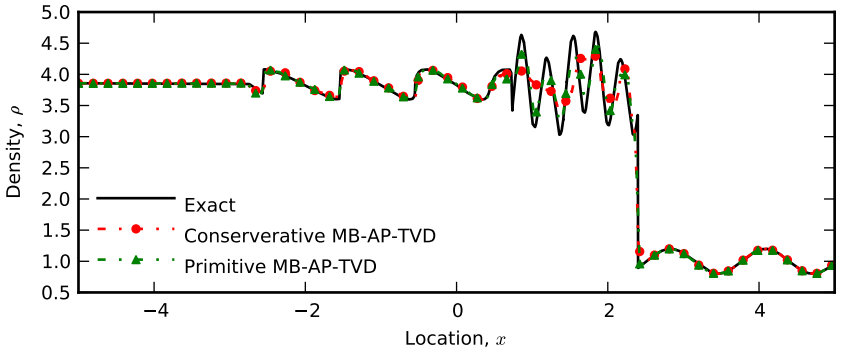


(b) AMR comparison, limiting using the primitive variables.

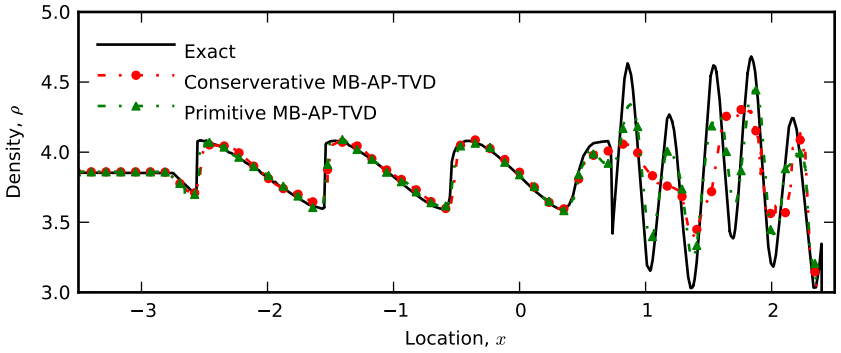


(c) Estimator comparison, limiting using primitive variables, the MB-AP-TVD detector and $l_{\max} = 3$.

Figure 27: Shock-entropy wave interaction problem for different solver options.



(a) Density



(b) Density detail

Figure 28: Shock-entropy wave interaction at $t = 1.8$ for $N = 256$, $p = 2$.

6.2.10 Convection in 2-D

The limiter and adaptivity approach is studied in 2D using the two-dimensional convection equation:

$$\frac{\partial u(\mathbf{x},t)}{\partial t} + c_1 \frac{\partial u(\mathbf{x},t)}{\partial x} + c_2 \frac{\partial u(\mathbf{x},t)}{\partial y} = 0 \tag{145}$$

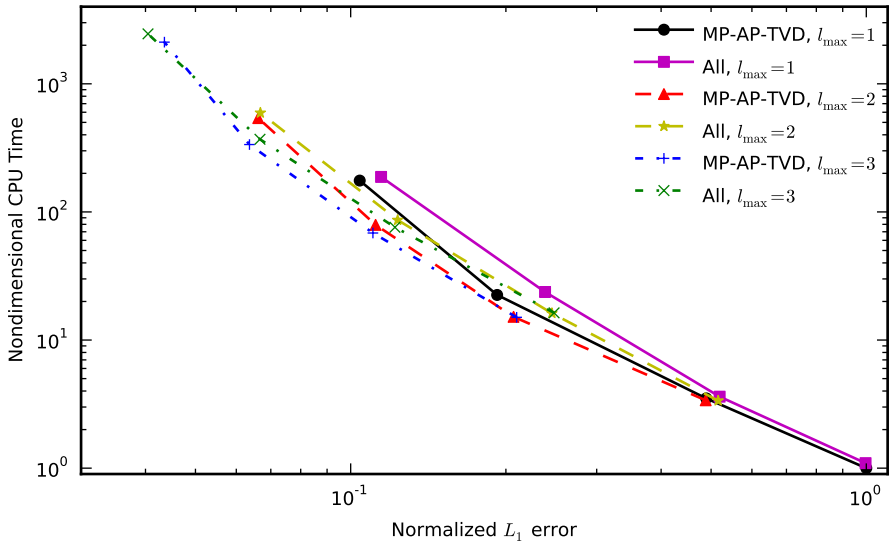
$$u(\mathbf{x}, t=0) = \begin{cases} 1 & \text{for} \quad (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.25^2 \\ 0 & \text{otherwise} \end{cases}$$

where u is a passive scalar, and c_1 and c_2 are the constant convection velocities equal to 1. The domain is the unit square $[0,1] \times [0,1]$ with periodic boundary conditions. The time integration used is the 3^{rd}-order SDC. Figure 29 shows the CPU time vs. the L_1 error at $t = 1$ for different solver options varying the number of element. In Fig. 29(a) $l_{\max}$ is varied together with the detector. Increasing the $l_{\max}$ improves the efficiency, and using the MB-AP-TVD helps too. In Fig. 29(b) the error estimator for AMR is varied. For this 2-D case JUMP1 and JUMP2 produce slightly different results, and JUMP2 is the most efficient of the three estimators. The mesh with the solution for KXRCF and and JUMP2 are shown in Fig. 30. KXRCF was originally designed for shocks, so it perform poorly here.

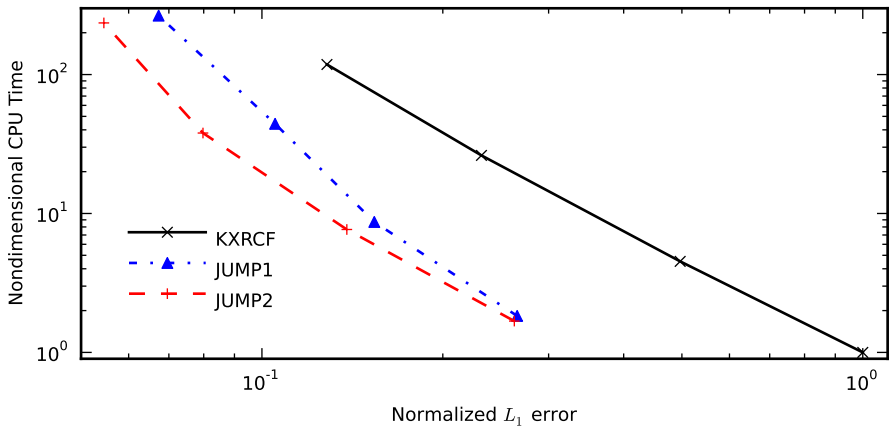
6.2.11 Double Mach reflection

This is a very common test case for the Euler equation first used by Woodward and Colella [144]. It was also solved by Krivodonova [72] using the ML with a uniform grid and without trouble-cell detector. This case consists of a strong shock impacting a wedge with a semi-angle of 30^o, thus it is usually simulated by a rectangular domain with a frame rotated 30^o over the original horizontal axis.

The rectangular domain has a size of $[0,4] \times [0,1]$. A right-moving Mach 10 shock is initially located forming an angle of 60^o with the x-axis passing by the coordinate $x = 1/6, y = 0$. The undisturbed air on the right of the shock has a density of 1.4 and a pressure of 1. The specific heat ratio is $\gamma = 1.4$. A slip-wall boundary is located

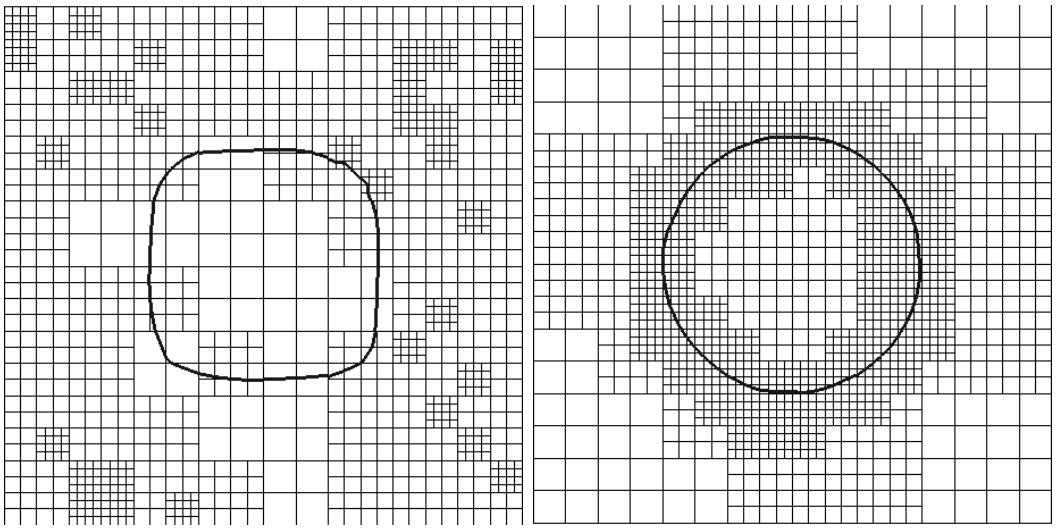


(a) AMR comparison, with and without detector.



(b) Estimator comparison, using the MB-AP-TVD detector and $l_{\max} = 3$.

Figure 29: Two-dimensional convection for different solver options.



(a) KXRCF. (b) JUMP2.

Figure 30: Mesh and isocontour for $u = 0.5$ for a 2-D convection problem.

at the lower boundary from $x = 1/6$ to $x = 4$. The right boundary is a supersonic outflow. The left boundary and bottom boundary for $x < 1/6$ are supersonic inflow. The reason for applying supersonic inflow at the bottom boundary is to mimic the effect of the wedge. The top boundary mimics the exact motion of the moving shock.

The grid has 48×12 cells with $l_{max} = 5$. The ML is used with the MB-AP-TVD detector. Second order polynomial elements are used with the 3^{rd}-order SDC method.

The results are shown for $t = 0.2$. Figure 31 shows 60 equally-spaced density contours. Figure 32(a) shows in black the cells flagged by the MB-AP-TVD detector as troubled cells. Note that the ML is not applied here where the flow is uniform, as intended. Figure 32(b) shows the level of refinement l. It can be noted that the level is increased where the features of the flow are smaller, as can be expected from Fig. 31.

6.2.12 Vortex convection

As we have seen with previous tests, the global efficiency does not improve significantly for problems dominated by discontinuities when the order is increased. We

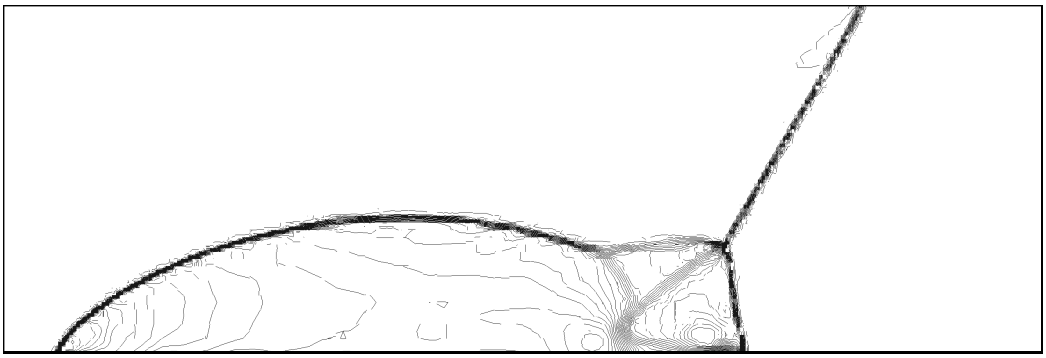


Figure 31: Density at $t = 0.2$ for the double Mach reflection.

(a) Troubled-cell.

(b) Refinement level.

Figure 32: Double Mach reflection.

studied simple smooth cases in 1D, but here we extend the study to a slightly more applicable case in 2D. An isentropic vortex is centered at the center of the domain $(x_c, y_c) = (0.5, 0.5)$. The flow is described by:

$$v_1 = M\sqrt{\gamma} + \epsilon\ \tau\ e^{\alpha\left(1-\tau^2\right)} \sin(\theta)$$

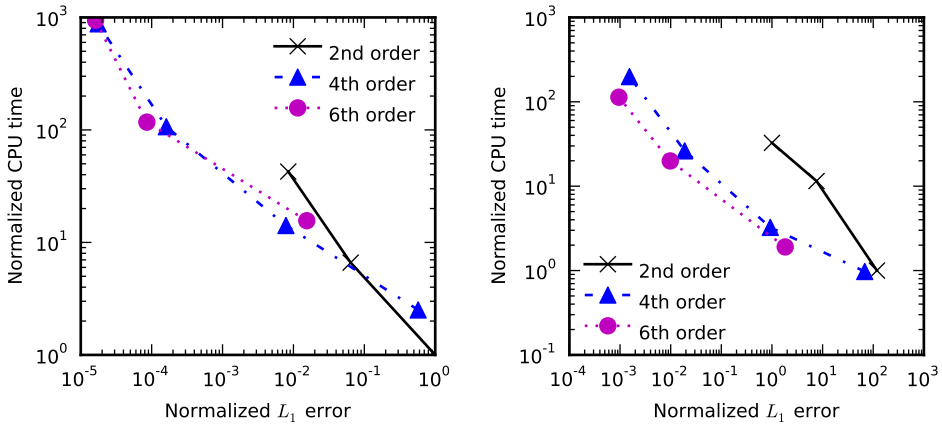
$$v_2 = -\epsilon\ \tau\ e^{\alpha\left(1-\tau^2\right)} \cos(\theta)$$

$$\rho = \left[1 - \frac{(\gamma-1)\ \epsilon^2\ e^{2\alpha\left(1-\tau^2\right)}}{4\alpha\gamma}\right]^{1/(\gamma-1)}$$

$$p = \rho^\gamma$$

where $M = 0.3$, $\tau = r/r_c$, $r = \sqrt{(x-x_c)^2 + (y-y_c)^2}$ and $\theta = \arctan\left((y-y_c)/(x-x_c)\right)$. Three parameters describe the vortex: the strength of the vortex ϵ, the decay rate of the vortex α, and the critical radius r_c. For this test the following values are used: $\epsilon = 0.3$, $\alpha = 0.204$, and $r_c = 0.05$. The domain is a unit square with periodic boundaries in every direction. Different number of elements and spatial orders, p, are used. Even though this is a smooth problem, the ML limiter with the MB-AP-TVD detector are used. The range of length scales is very narrow, so AMR is not needed.

Figure 33(a) presents the CPU time vs. the L_1 error after one period using the SDC of order $p + 1$. The same pattern as for previous 1D cases is observed here. The efficiency increases with the order at in the high accuracy range since at equal CPU time the numerical error is smaller. However, in the low accuracy range low order schemes perform more efficiently. Even though this problem is dominated by convection, the time-step size is limited by the acoustic time, so one could assume that we are over-resolving in time. Thus, we rerun the same cases with the 3^{rd}-order SDC for every p. Note that in this case the CFL has to be adjusted for $p > 2$. We use $C = 0.5$, $C = 0.45$, and $C = 0.4$ for $p = 2$, $p = 4$, and $p = 6$, respectively. Now higher orders in space have a greater advantage. In cases where the error in time is more significant, increasing the order of the scheme in time would make improvements. In this case, however, higher orders in time only add more computational cost.



(a) SDC of order $p+1$.

(b) SDC of order 3.

Figure 33: Efficiency for the convection of an isentropic vortex.

It can be concluded that the limiting procedure can be freely applied in the whole domain even where smooth features are present. This aspect is important for large-scale applied problems where several types of features can be present at the same time, so a generic and robust scheme is wanted. The shock-vortex interaction case shown below elaborates more on this.

6.2.13 Shock-vortex interaction

This problem consists of a vortex going through a shock and helps to test how the solver behaves when smooth features interact with discontinuities. For more information on this kind of problems see [110]. The initial conditions are the same as in [154, 65]. The size of the domain is $[0, 2] \times [0, 1]$. Reflective boundary conditions are used on top and bottom. The left boundary is a supersonic inflow, while the right boundary is an outflow. A stationary shock is located at $x = 0.5$, its pre-shock Mach number is $M_s = 1.1$, and the left side state is defined by $\rho = 1$, $u = M_s\sqrt{\gamma}$, $v = 0$ and $p = 1$. The right state can be determined from the left state by using the stationary shock relations. An isentropic vortex is centered at $(x_c, y_c) = (0.25, 0.5)$. Therefore,

on the left hand side of the shock the flow is described by:

$$v_1 = M_s\sqrt{\gamma} + \epsilon\ \tau\ e^{\alpha\left(1-\tau^2\right)}\sin(\theta)$$

$$v_2 = -\epsilon\ \tau\ e^{\alpha\left(1-\tau^2\right)}\cos(\theta)$$

$$\rho = \left[1 - \frac{(\gamma-1)\ \epsilon^2\ e^{2\alpha\left(1-\tau^2\right)}}{4\alpha\gamma}\right]^{1/(\gamma-1)}$$

$$p = \rho^\gamma$$

where $\tau = r/r_c$, $r = \sqrt{(x-x_c)^2 + (y-y_c)^2}$ and $\theta = \arctan\left((y-y_c)/(x-x_c)\right)$. For this test the following values are used: $\epsilon = 0.3$, $\alpha = 0.204$, and $r_c = 0.05$.

A uniform grid with 32×16 cells and $l_{max} = 4$ is used with $p = 2$. The time integration is done with the 3^{rd}-order SDC method. The ML is used with the MB-AP-TVD detector. The pressure at $t = 0.8$ are shown in Fig. 34 with 60 equally-spaced contours. Figures 35(a) and 35(b) indicate how the solver adapt to the solution to avoid instabilities and waste computational resources. The vortex successfully goes through the shock and features with different length scales are properly be resolved. Similar result were observed in [154, 65] using other numerical schemes.

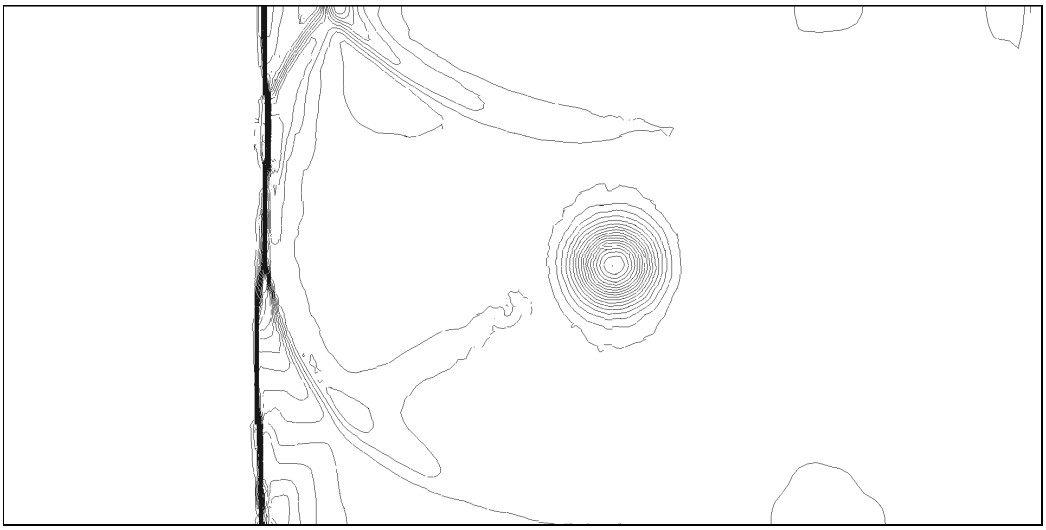
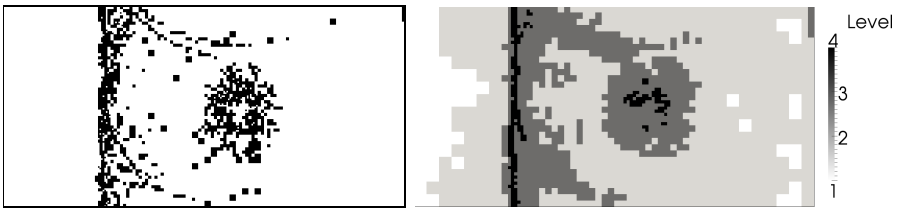


Figure 34: Pressure iso-contours for a shock-vortex interaction.



(a) Troubled-cell. (b) Refinement level.

Figure 35: Shock-vortex interaction.

6.2.14 Spherical shock test

The final test is a spherical shock case in a cube defined in $[0,1] \times [0,1] \times [0,1]$. The initial conditions are similar to the typical Sod's problem, but in this case a spherical symmetry is used:

$$(\rho, v_1, v_2, v_3, p) = \begin{cases} (1.0, 0.0, 0.0, 0.0, 1.0), & r \leq 0.5 \\ (0.125, 0.0, 0.0, 0.0, 0.1), & r > 0.5 \end{cases} \tag{146}$$

where r is the distance from $(0,0,0)$. The initial grid has 32^3 $p = 2$ elements, each allowed to refine to a level $l_{max} = 3$. The integration in time is done with the 3^{rd}-order SDC method.

An "exact" solution is estimated solving the Euler equation in spherical coordinates assuming spherical symmetry. Thus, the equation being solved in the domain $[0,1]$ is:

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{u})}{\partial x} = \mathbf{S} \tag{147}$$

where $\mathbf{S} = -2/x \left(\rho v_1, \rho v_1^2, (\rho E_T + p) v_1\right)^T$. This 1D problem is solved on a grid with 1024 cells with $p = 2$ and integrated in time with the 3^{rd}-order SDC method.

A very similar test case to this one was studied in 2D in [146, 135].

The density and pressure at $t = 0.2$ over four different vectors are shown in Fig. 36. Each of these four vectors are: $(1,0,0)$, $(0,1,0)$, $(0,0,1)$, and $(1,1,1)$. Given that the density on the different trajectories match, the scheme successfully respects the spherical symmetry of the problem. Note that the results shown do not match exactly

the classical one-dimensional Sod shock-tube problem due to 3D effects. Figure 37 demonstrates the ability of AMR to track the shock and rarefaction waves as required.

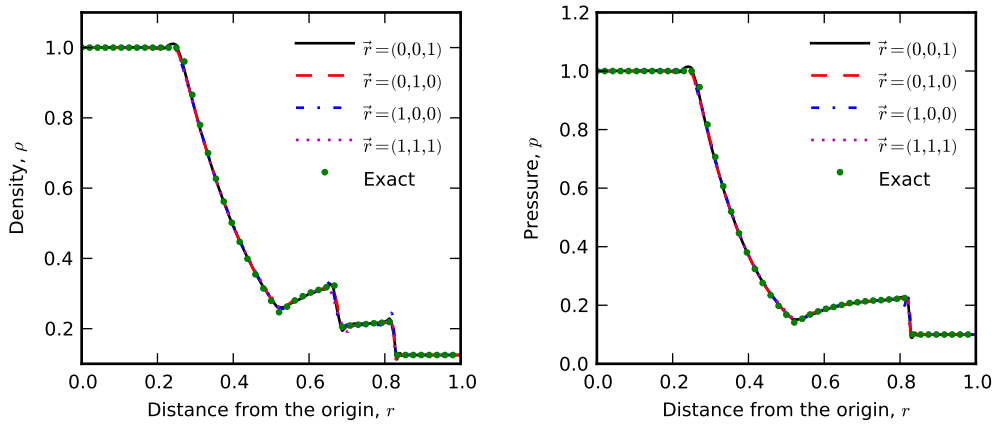


Figure 36: Spherical shock test at $t = 0.15$ over four different vectors.

6.3 Discussion

The Euler equations are solved using the discontinuous Galerkin method with adaptive mesh refinement and high-order of accuracy in space and time.

It was shown using high-order schemes that problems with discontinuities can present high order of convergence in the smooth regions, while the global order of accuracy is close to 1 in the L_1 and L_∞ norm. Most of the cases studied include discontinuities. Therefore, in such cases the order in space and time of the scheme used is 3, since, as it was shown, increasing the order of the solver does not improve its efficiency significantly when discontinuities dominate. Given that the time-step is limited by the acoustic time, convection dominated problems end up being over-resolved in time, so in such cases increasing the order in time produces an unnecessary computational cost.

A simple and effective error estimator for adaptivity based on the inter-element jump is suggested and it was shown to be more efficient than other estimator found

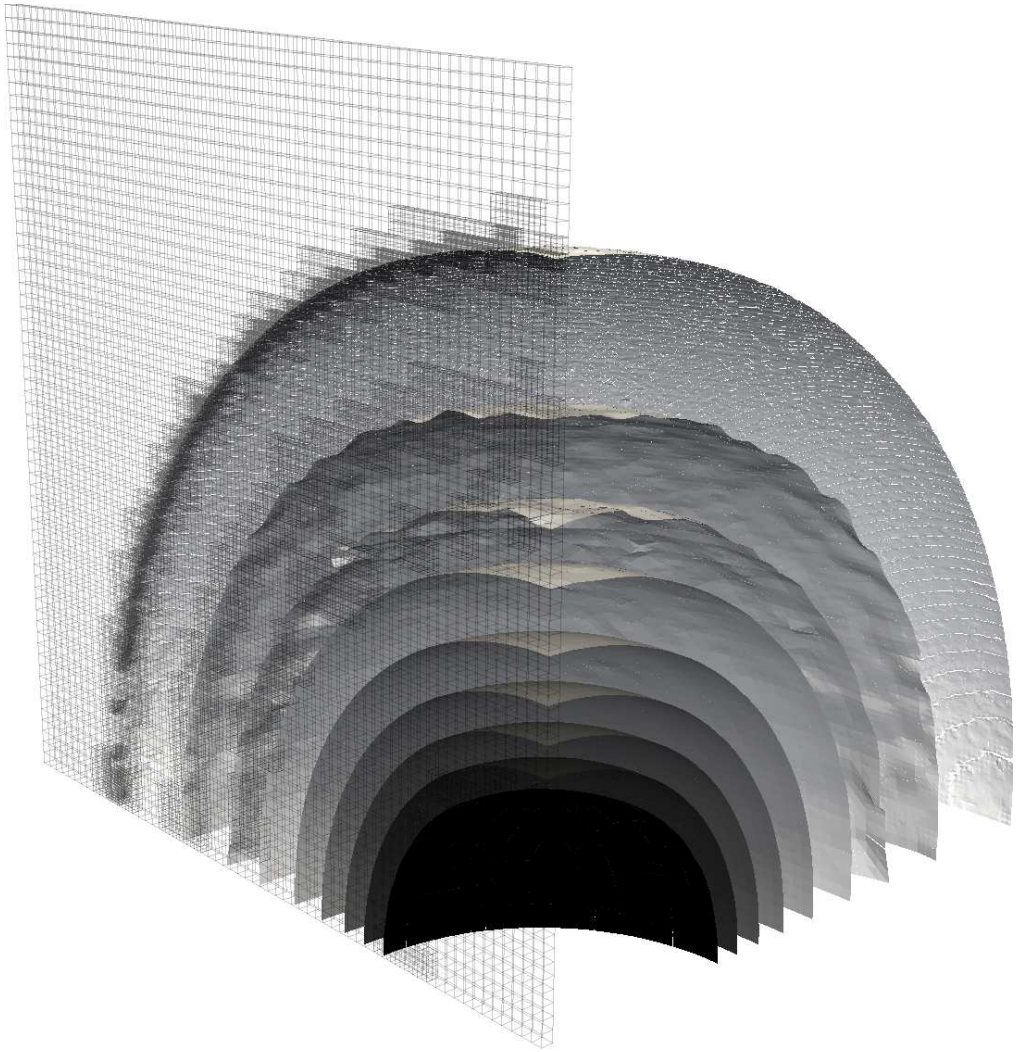


Figure 37: Spherical shock test. Density contours and grid refinement.

in the literature. From a computational-resources point of view, the most efficient combination of maximum levels of refinement and initial number of cells is problem-dependent. In a few of the tested problems the overhead caused by the adaptation made it unnecessary. However, in no case with a wide range of scales AMR caused a significant loss of efficiency.

The AP-TVD detector in [154] was modified replacing the averaged-derivative basis that it originally required by the Legendre polynomial basis, which is commonly used in DG. Therefore, the current approach avoids the transformation and a better efficiency of the scheme is observed. We named it the moment-based AP-TVD (MB-AP-TVD) since it uses the default moments of the solution – like the moment limiter (ML) does. Yang and Wang [154] showed that the AP-TVD detector gives better results than the more common detectors used in [106], so the MB-AP-TVD should be even more efficient than those.

The troubled cells were treated with a ML modified for non-uniform meshes with hanging nodes. The limiting stage is done using primitive, characteristic, and conservative variables and then appropriately evaluated. The optimal choice of limiting variables and where to apply the limiter is case-specific, but based on the results of the one-dimensional tests limiting using primitive variables and the MB-AP-TVD detector is the recommended starting point, especially for multi-dimensional problems since the ML is inherently multi-dimensional and the characteristic decomposition, slightly better than primitive variables in 1D, cannot be applied in 2 or 3 dimensions. The computational cost due to the conversion from conservative to the other variables seems to be negligible. This Jacobian (and its inverse) is computed each time an element is being limited, but the CPU advantage of conservative variables seems to be lost since worse limiting requires more correction steps of the limiter.

In addition, most test cases were studied with SDC method, what shows that it is an adequate time-integration scheme that could be considered as an alternative to the

Runge-Kutta methods for certain applications, especially as the order increases since it is easier to derive and implement. More research is still necessary to determine the numerical properties of SDC-DG, such as its maximum CFL number and its numerical dissipation at different frequencies.

For Cartesian, low-order cases DG may perform similarly to FD or FV [161]. However, it is important to note that when the conditions are more sophisticated (e.g. unstructured, non-Cartesian, high order), where other schemes cannot even be applied, DG still performs well.

The scheme, including our new developments, are relatively simple to implement, robust, with great numerical properties. Thus, it presents a technique that should be exploited for more generic applications. p-adaptivity could be useful especially for problems with discontinuities, which are better treated with low-order schemes.

For steady-state problems the proposed approach may not be highly efficient. Common modifications to improve the convergence to a steady state include some type of filter in time for the limiter and other discrete operations, since they create oscillations that do not let the residual decrease enough. However, the goal of this study is to investigate methods needed for time dependent problems.

Given the results observed in this chapter, for the rest of this thesis the following numerical techniques are used unless specified otherwise: the MB-AP-TVD for troubled-cell detector with the Moment Limit using primitive variables; 3^{rd}-order TVD-RK for time integration; $p = 2$ to balance efficiency in problems with smooth and discontinuous regions; and JUMP2 as error estimator.

CHAPTER VII

SHOCK-BUBBLE INTERACTION WITH THE 4-EQ. MODEL USING DG

7.1 Introduction

Two multi-fluid methods were described in detail in Chapter 3: the 4-eq. and the 5-eq. models. The former is simpler, thus, it is treated in this chapter to build experience, while the later is studied in the following chapters.

Also, two DG methods were described in Section 4.2.2: one based on the RKDG and the other as an extension to LDG. The former is used here also to build experience since it is slightly simpler, but it could result to be unstable unless an ad-hoc correction is used.

Inviscid shock-tube problems are used to verify the formulation and the increased efficiency when using AMR. In addition, the solver is applied to shock-bubble interactions and compared with the literature. These cases put the compressibility together with the material discontinuities to the test, two main aspects of the formulation developed.

7.2 Verification of the 4-eq. model with DG

Various test cases used in past studies are used to establish the capability of this new numerical algorithm. The details of the test cases and the rationals for them are summarized in Table 16.

7.2.1 Interface convection

The occurrence of pressure oscillations at interfaces is a common problem in numerical multiphase schemes that arise even in simple problems. We study here the

Table 16: Summary of two-phase inviscid cases for verification and validation.

Test case	Purpose	Section
1-D convection	Show the efficiency and noise of the scheme.	7.2.1
Two-phase Sod's problem	Verification for a material and pressure jump.	7.2.2
Two-phase Lax's problem	Verification for a material, pressure, and velocity jump.	7.2.3
Two-phase shock-entropy waves interaction	Mixture of sharp and smooth variations.	7.2.4
Shock-tube	Test for larger material differences.	7.2.5

pressure oscillations that originate when a material interface is convected. The initial conditions are:

$$(\rho, v_1, P, \alpha_1)^T = \begin{cases} (1.0, 1.0, 1.0, 0.9999)^T & \text{for } x < 0.0 \\ (10.0, 1.0, 1.0, 0.0001)^T & \text{for } x \geq 0.0 \end{cases}$$

The 1-D domain is $[-1, 1]$, has 25 elements, $l_{max} = 3$ and periodic boundaries. The material properties are $\gamma_1 = 1.4$, $P_{\infty,1} = 0.0$, $\gamma_2 = 1.2$, and $P_{\infty,2} = 0.0$. The result after one period, $t = 2.0$, is shown in figure 38. It can be seen that both pressure and velocity have negligible error. Note in figure 38(e) how refinement is applied only at the interface, while the grid is coarse where the solution is smooth. The same case run for different number of levels can give an estimation of the effectiveness of the AMR algorithm. Fig. 39 compares the L_1 error vs. the CPU time and was obtained using $l_{\max} = 1, 2, 3$ and 16, 32, and 64 $2^{nd} - order$ elements. The L_1 error is computed at $t = 2.0$ based on the volume fraction as:

$$e_{L_1} = \int_{-1}^{1} |\alpha_1(x) - \widehat{\alpha_1}(x)| dx \tag{148}$$

where $\widehat{\alpha_1}$ is the exact solution. The CPU time does not take I/O into account. Note that as the number of levels is increased the curves get closer to the origin.

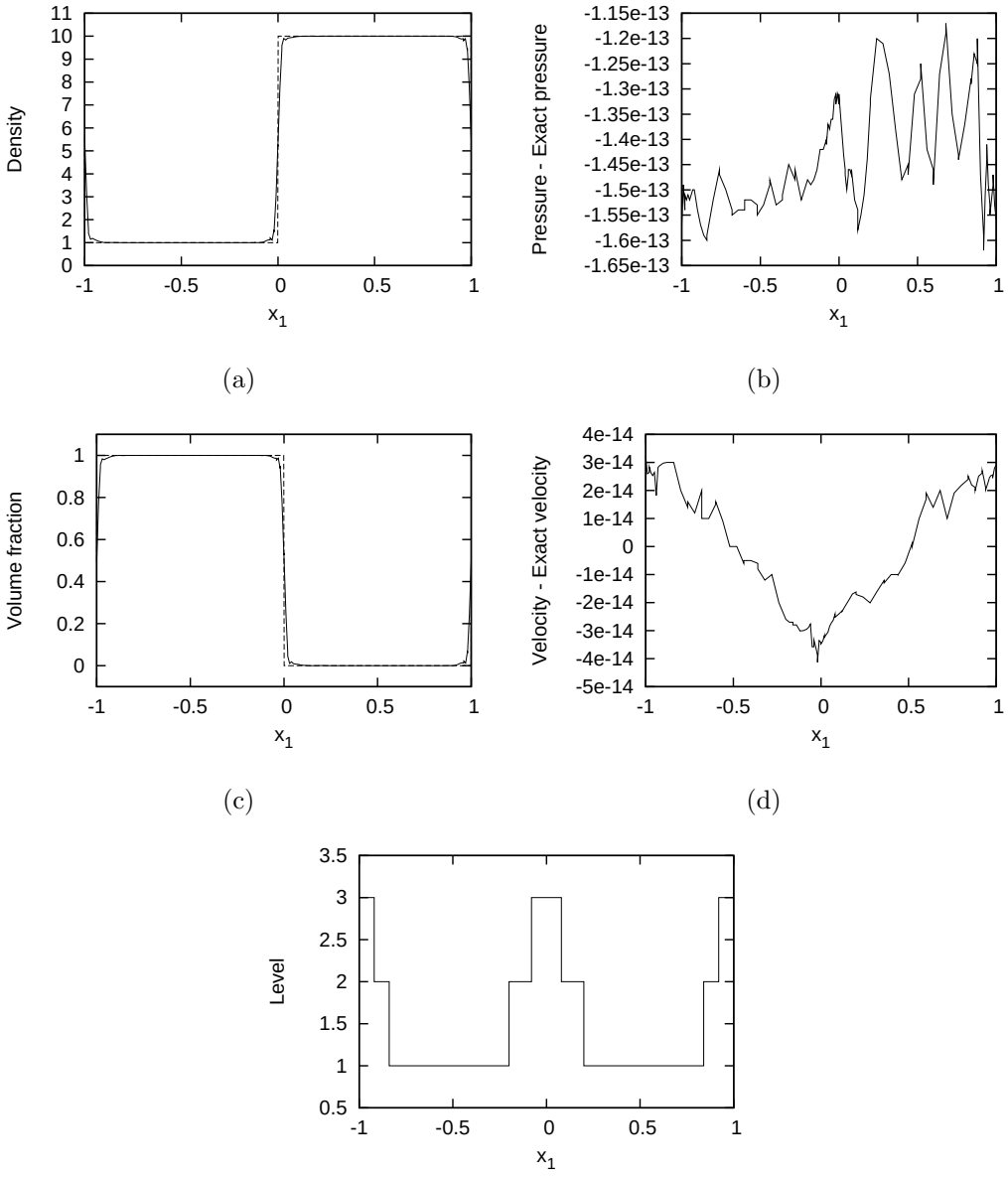


Figure 38: Convection of an interface at $t = 2.0$. Exact solution in dotted line, computed solution in solid line.

This represents that increasing the number of levels is more efficient than increasing the total number of elements. Cases with larger areas with smooth solution should provide even better efficiency, what is the case with the 2- and 3-D problems studied below.

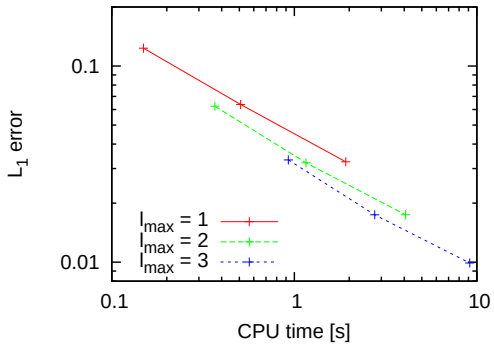


Figure 39: L_1 error in α_1 vs. CPU time for the convection of an interface at $t = 2.0$.

Now the same case is studied with more challenging materials: $\gamma_1 = 1.4$, $P_{\infty,1} = 0.0$, $\gamma_2 = 4.0$, and $P_{\infty,2} = 1.0$. The results are in figure 40. Even with greater material differences, the numerical error is close to machine precision.

7.2.2 2-phase Sod problem

This test is usually applied in a single-fluid context and it is known has the Sod problem [22]. The initial states and material properties are [140]:

$$(\rho, v_1, P, \alpha_1)^T = \begin{cases} (1, 0, 1, 0.9999)^T & \text{for } x < 0.0 \\ (0.125, 0, 0.1, 0.0001)^T & \text{for } x \geq 0.0 \end{cases}$$

The material properties are $\gamma_1 = 1.4$, $P_{\infty,1} = 0$, $\gamma_2 = 1.6$ and $P_{\infty,2} = 0$. The 1-D domain is $[-5, 5]$, has 25 elements and $l_{max} = 3$. An "exact" solution is estimated with 1000 elements and no adaptivity. The solution at $t = 2.0$ and the level of each cell are shown in Fig. 41. The grid adapted at discontinuities and sharp changes.

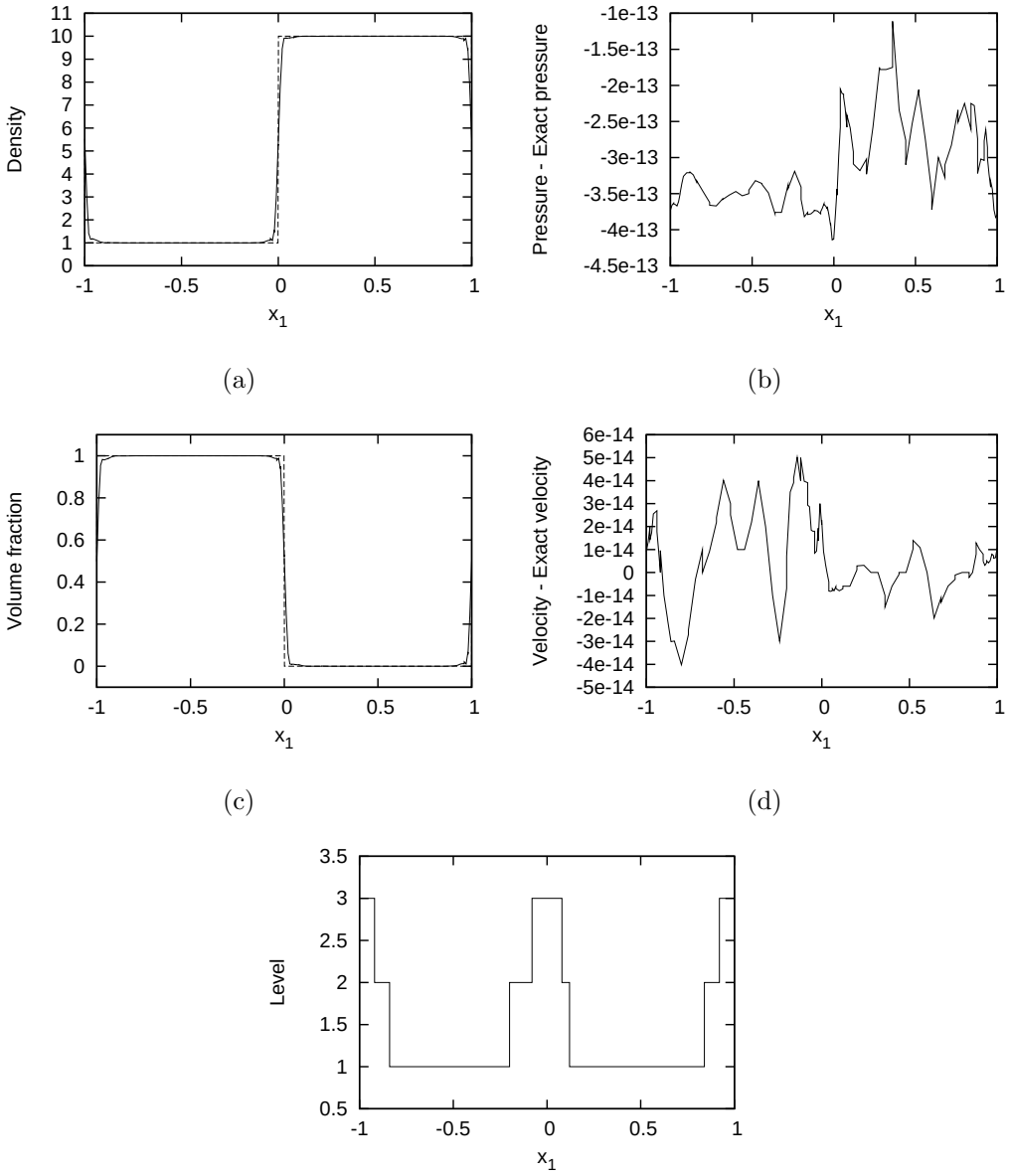


Figure 40: Convection of an interface at $t = 2.0$ with large material differences. Exact solution in dotted line, computed solution in solid line.

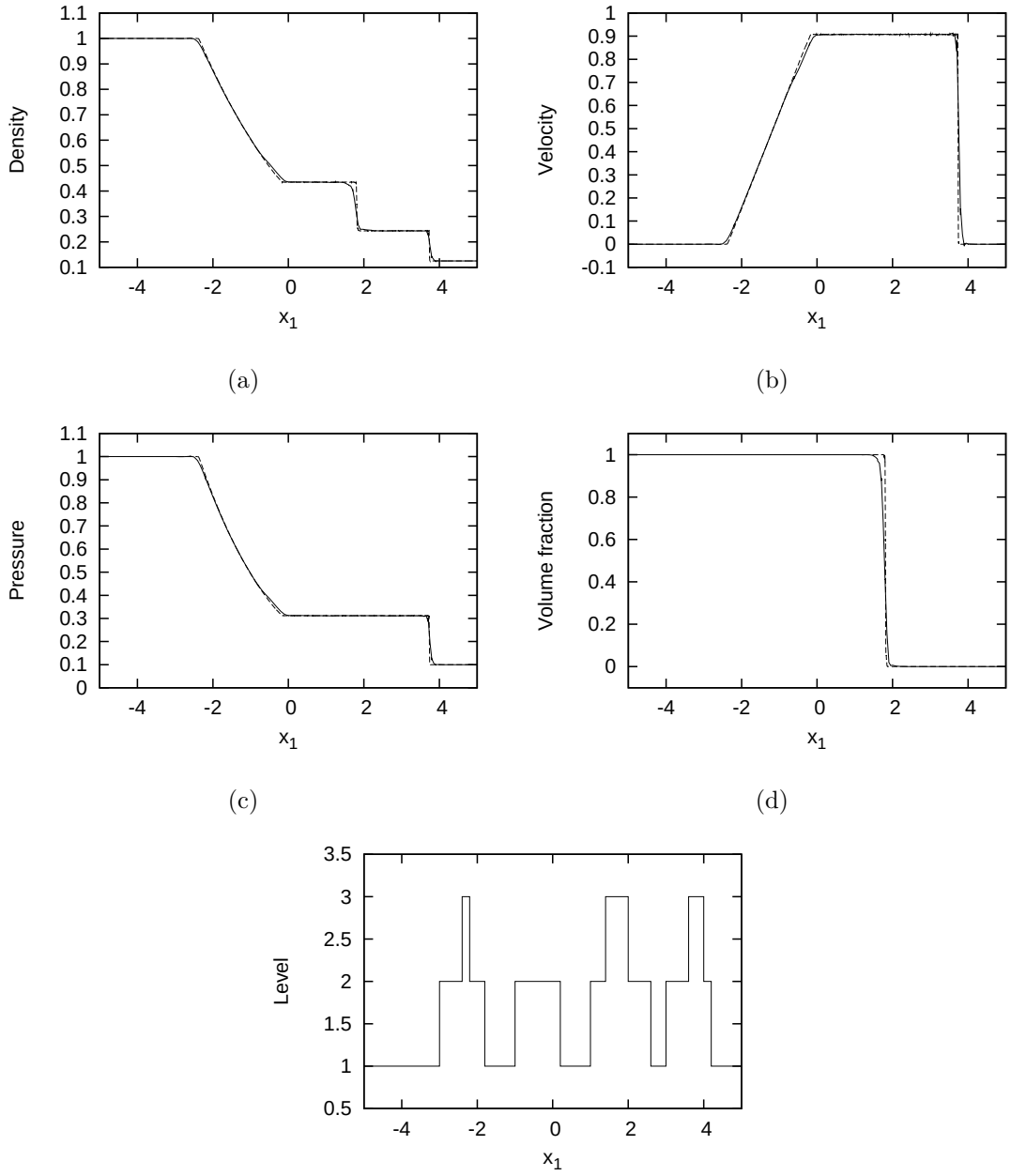


Figure 41: 2-phase Sod problem at $t = 2.0$. Adaptive solution in solid line and "exact" solution in dotted line.

7.2.3 2-phase Lax problem

This case is also a common test in a single-fluid context known has Lax problem [22]. The initial conditions are [140]:

$$(\rho, v_1, P, \alpha_1)^T = \begin{cases} (0.445, 0.699, 3.527, 0.9999)^T & \text{for } x < 0.0 \\ (0.5, 0, 0.8565, 0.0001)^T & \text{for } x \geq 0.0 \end{cases}$$

The material properties are $\gamma_1 = 1.4$, $P_{\infty,1} = 0$, $\gamma_2 = 1.6$ and $P_{\infty,2} = 0$. The 1-D domain is $[-5, 5]$, has 25 elements and $l_{max} = 3$. An "exact" solution is estimated with 1000 elements and no adaptivity. The solution at $t = 1.3$ and the level of each cell are shown in Fig. 42. The grid adapted at discontinuities and sharp changes.

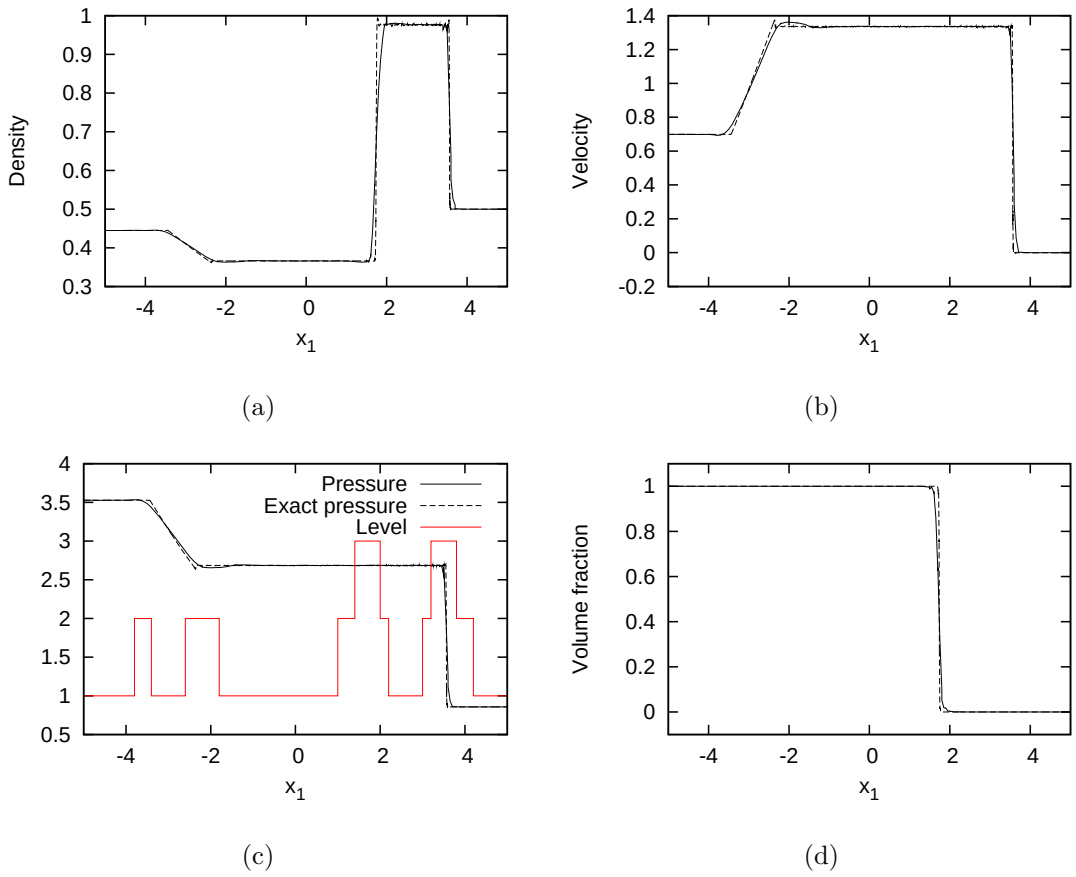


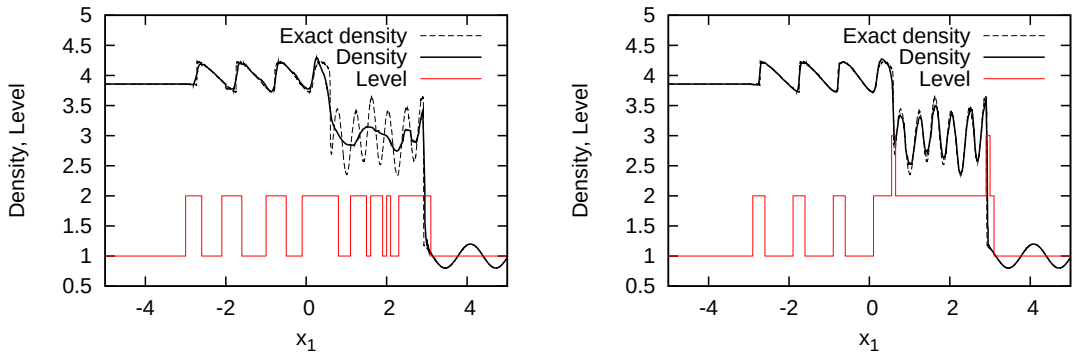
Figure 42: 2-phase Lax problem at $t = 1.3$. Adaptive solution in solid line and "exact" solution in dotted line.

7.2.4 Shock-entropy interaction

This test is common in a single-fluid context where a shock interacts with smoothly varying features [22] and here it is shown with an added material discontinuity. The initial states and material properties are [140]:

$$(\rho, v_1, P, \gamma, P_\infty, \alpha_1)^T = \begin{cases} (3.857143, 2.629369, 10.333333, 0.9999)^T & \text{for } x < -4.0 \\ (1 + \sin(5.0x), 0.0, 1.0, 0.0001)^T & \text{for } x \geq -4.0 \end{cases}$$

The material properties are $\gamma_1 = 1.4$, $P_{\infty,1} = 0.0$, $\gamma_2 = 1.666666$ and $P_{\infty,2} = 0.0$. The 1-D domain is $[-5, 5]$, has 100 elements and $l_{max} = 3$. An "exact" solution is estimated with 2000 elements and no adaptivity. The density at $t = 1.8$ and the level of each element are shown in figure 43. It is clear that when starting with 100 elements, $l_{max} = 2$ does not provide enough resolution to capture some fluctuations. Figure 43(b) shows that the adaptivity algorithm with $l_{max} = 3$ refines appropriately where the solution presents sharp changes and allows for an efficient capturing of all the features.



(a) $l_{max} = 2$.

(b) $l_{max} = 3$.

Figure 43: Density profile at $t = 1.8$ for the shock-entropy interaction case. Adaptive solution in solid line and "exact" solution in dotted line.

7.2.5 A two-phase shock-tube problem

Now the scheme is tested under higher pressure ratios with different material properties as in [66]. The initial conditions are:

$$(\rho, v_1, P, \alpha_1)^T = \begin{cases} (1.241, 0, 2.753, 0.9999)^T & \text{for } x < 0.0 \\ (0.991, 0, 3.059 \times 10^{-4}, 0.0001)^T & \text{for } x \geq 0.0 \end{cases}$$

with $\gamma_1 = 1.4$, $P_{\infty,1} = 0$, $\gamma_2 = 5.5$ and $P_{\infty,2} = 1.505$. The 1-D domain is $[-5, 5]$, has 25 elements and $l_{max} = 3$. An "exact" solution is estimated with 1000 elements and no adaptivity. The result at $t = 0.6$ is shown in Fig.44. Even though the two materials differ even in the value of P_∞, the oscillations at the discontinuities are minimum and are reduced as the resolution increases.

7.3 Shock-bubble interaction

7.3.1 Gas-gas, Bubble-shock interaction in 2-D

A Helium cylinder in air is impacted by a shock with a Mach number of $Ma = 1.22$, as studied experimentally and numerically in the literature[48, 109]. Given that a cylinder was used for the experiments, the resulted flow was mostly two-dimensional and symmetric respect to the horizontal line crossing the center of the bubble. Therefore, for this validation we use a 2-D domain as in figure 45. Fully reflective boundaries were used at the top and bottom, and non-reflective boundaries based on Riemann invariants were used at the left and right boundaries. A grid with 160×20 elements and one with 320×40 elements were used, both having $l_{max} = 4$. Initially, for the bubble:

$$(\rho, v_1, v_2, P, \gamma, P_\infty, \alpha_1)^T = \left(0.16347\ kg/m^3, 0.0\ m/s, 0.0\ m/s, 101325\ Pa, 1.67, 0.0\ Pa, 0.00001\right)^T$$

and for the air surrounding the bubble:

$$(\rho, v_1, v_2, P, \gamma, P_\infty, \alpha_1)^T = \left(1.1847\ kg/m^3, 0.0\ m/s, 0.0\ m/s, 101325\ Pa, 1.4, 0.0\ Pa, 0.99999\right)^T$$

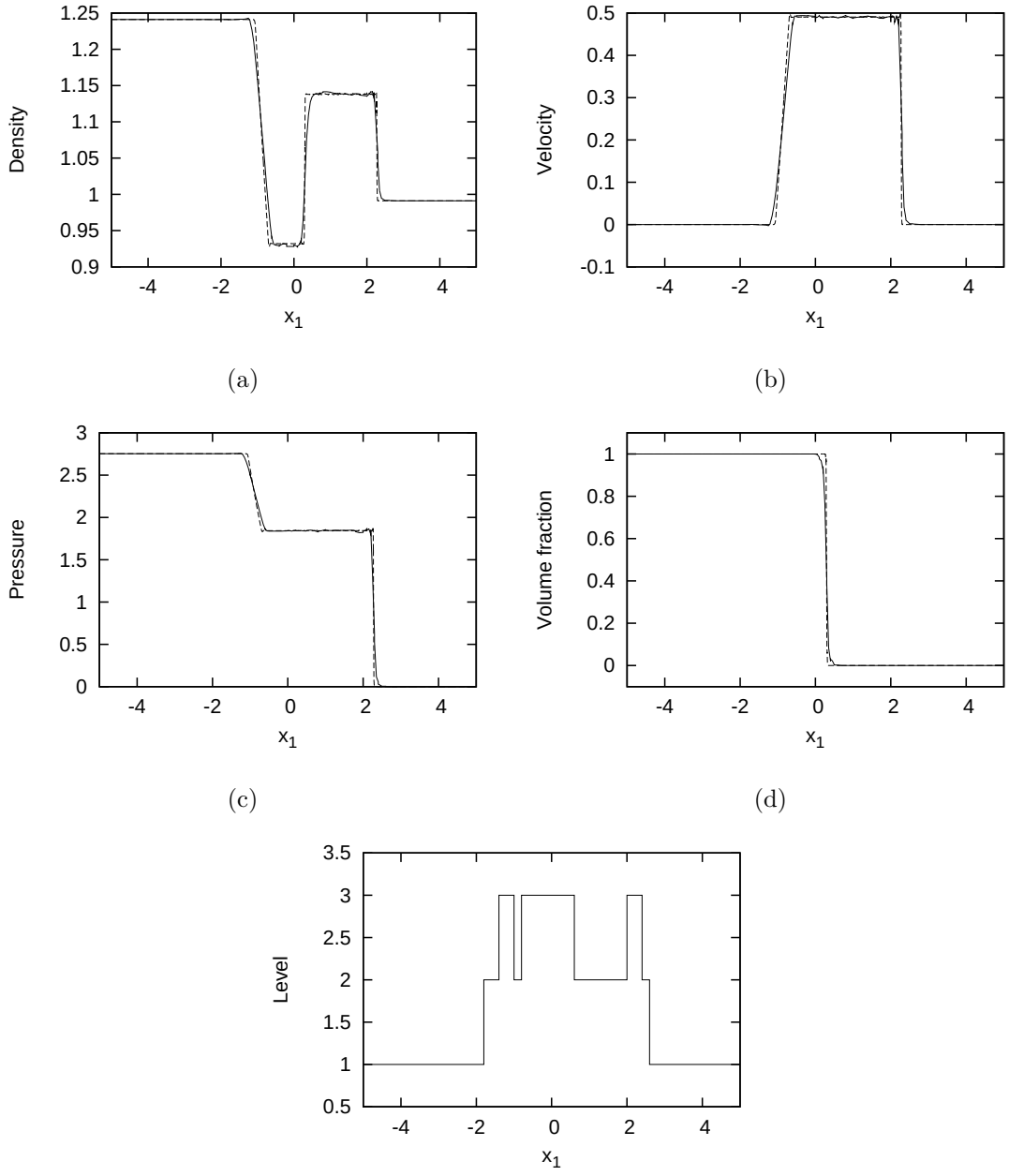


Figure 44: Shock-tube problem at time $t = 0.6\ s$. Adaptive solution in solid line and "exact" solution in dotted line.

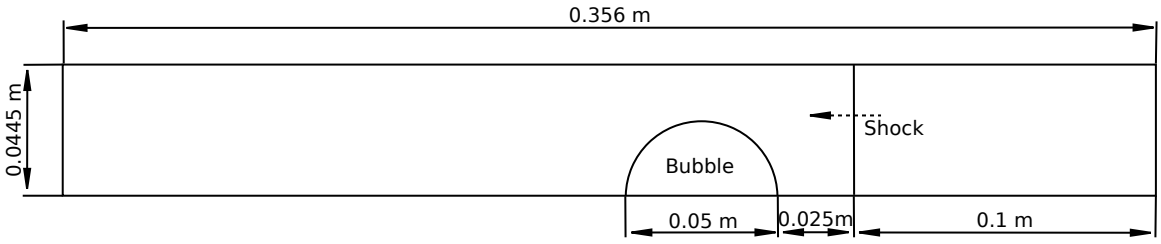
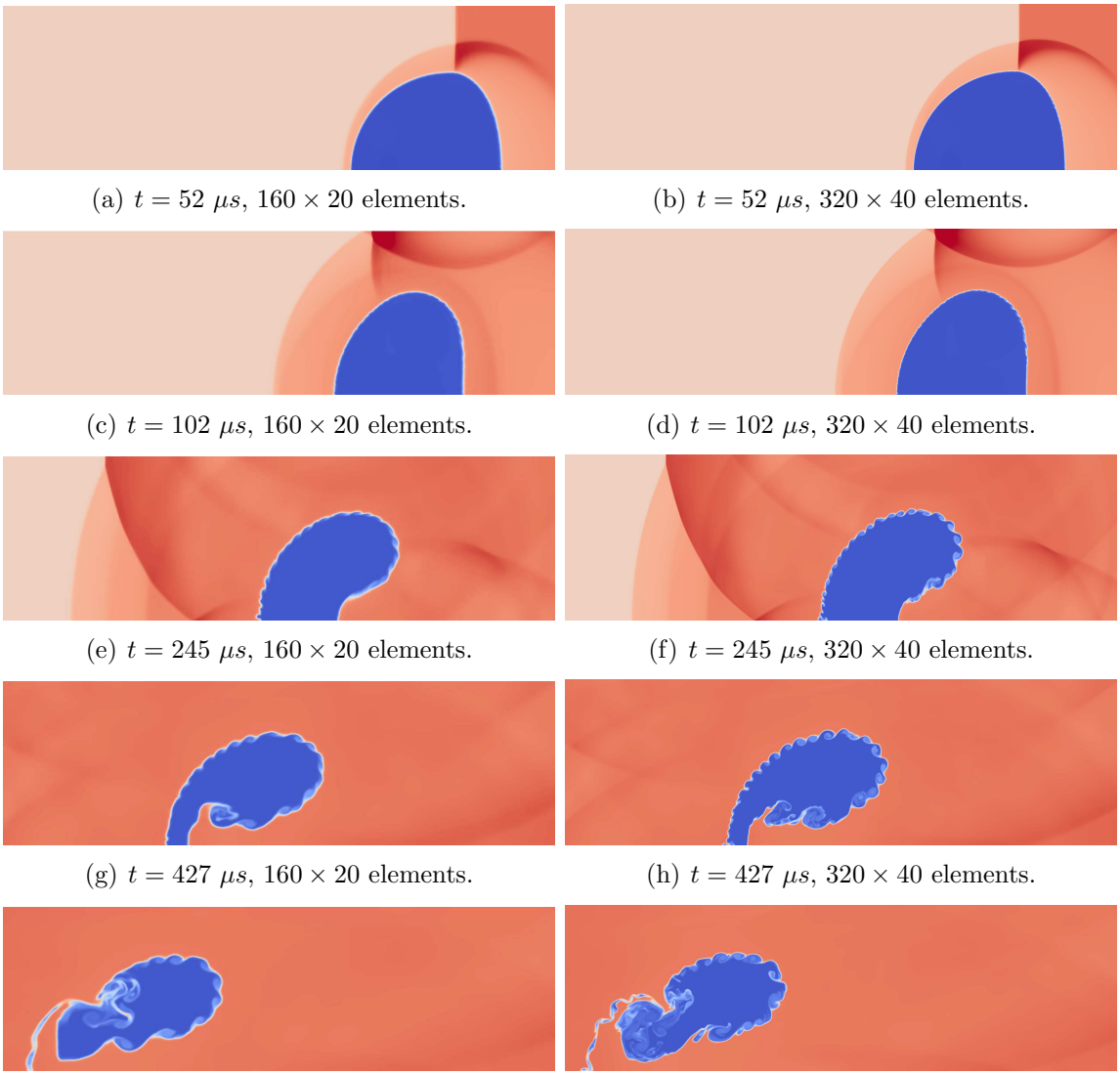


Figure 45: Geometry used for the 2-D interaction of a shock and a helium bubble in air.

The post-shock values are obtained by using the Rankine-Hugoniot condition.

The instability induced by the shock at the interface makes the problem ill-posed and grid-dependent for the Euler equations (more details below). Nonetheless, the inviscid phenomena are still useful to validate the scheme. A sequence is presented in figure 46. As studied by Haas and Sturtevant[48], depending on the speed of sound in the bubble the acoustic waves converge or diverge. For a Helium bubble, a lighter gas than air, the speed of sound is higher than in its surroundings and the waves diverge because the transmitted wave inside the bubble travels faster than around it. Figure 47 is the result at $t = 32\ \mu s$ zoomed in and presents the different waves created. When the incident shock hits the material discontinuity, a refracted shock is created inside the bubble and an expansion wave is created upstream. The refracted shock moves faster than the incident shock because the speed of sound in helium is faster than in air. Also, given that the refracted shock is ahead of the incident shock a side shock is created to compensate for this misalignment. The shock moving at different speeds inside and outside the bubble generates high vorticity concentrated at the interface, see figure 48. This shear layer evolves in time and generates the growing vortexes shown in figure 46, resembling a typical mixing layer problem. This growth is affected by viscous effect. Given that we ignored viscosity in the governing equation, the numerical diffusion is controlling these vortexes, thus this clarifies why the governing equation is ill-posed to study the interface behavior.



(a) $t = 52\ \mu s$, 160×20 elements.

(b) $t = 52\ \mu s$, 320×40 elements.

(c) $t = 102\ \mu s$, 160×20 elements.

(d) $t = 102\ \mu s$, 320×40 elements.

(e) $t = 245\ \mu s$, 160×20 elements.

(f) $t = 245\ \mu s$, 320×40 elements.

(g) $t = 427\ \mu s$, 160×20 elements.

(h) $t = 427\ \mu s$, 320×40 elements.

(i) $t = 674\ \mu s$, 160×20 elements.

(j) $t = 674\ \mu s$, 320×40 elements.

Figure 46: Density field for the interaction of a cylindrical bubble and a shock.

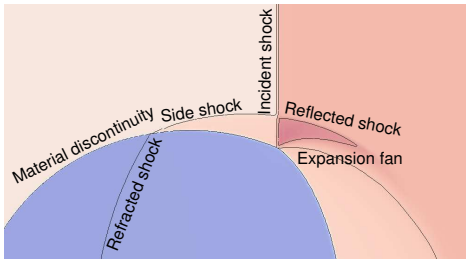


Figure 47: Result at $t = 32\ \mu s$ showing the different waves for a cylindrical Helium bubble. The lines are pressure and density iso-contours.

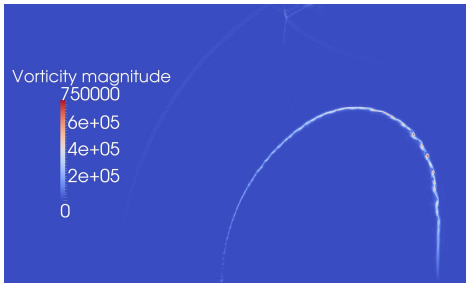


Figure 48: Vorticity magnitude at $t = 102\ \mu s$ with 320×40 elements for a cylindrical Helium bubble.

7.3.2 Gas-gas, Bubble-shock interaction in 3D

A similar test as before is conducted in 3D and based on experiments with a spherical Helium or Krypton bubble as presented by Layes and Le Métayer[77]. Two shock strengths are used, $Ma = 1.2$ and $Ma = 1.7$, which makes a combination of 4 different case. Due to the symmetry of the problem, one fourth of the geometry is actually simulated. A diagram of the whole geometry is shown in figure 49. The left and right boundaries are non-reflective as above, whereas the other 4 boundaries are fully reflective. The number of elements is $68 \times 8 \times 8$ with $l_{max} = 3$. Initially, for the bubble:

$$(\rho, v_1, v_2, P, \gamma, P_\infty, \alpha_1)^T = \big(0.167\ kg/m^3, 0.0\ m/s, 0.0\ m/s, 0.0\ m/s, 101325\ Pa, 1.67, 0.0\ Pa, 0.0000$$

and for the air surrounding the bubble:

$$(\rho, v_1, v_2, P, \gamma, P_\infty, \alpha_1)^T = \big(1.29\ kg/m^3, 0.0\ m/s, 0.0\ m/s, 0.0\ m/s, 101325\ Pa, 1.4, 0.0\ Pa, 0.99999\big)^T$$

The post-shock values are obtained by using the Rankine-Hugoniot condition.

Figures 50 and 51 show the X location of the down- and up-stream limits of the bubble respect to X_0, which is the initial up-stream limit. The same figure also shows the results published by Layes and Le Métayer[77], where they used slightly different points (named points 1 and 4) and one of them does not match the down-stream limit at the beginning of the simulation. Therefore, these plots show the good accuracy of the current solver once point 4 coincides with the down-stream border.

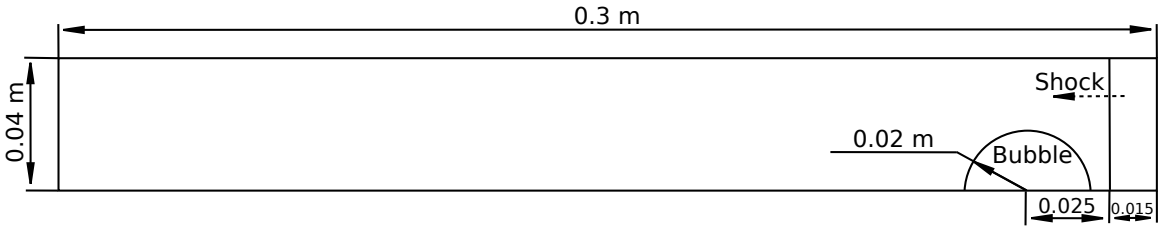
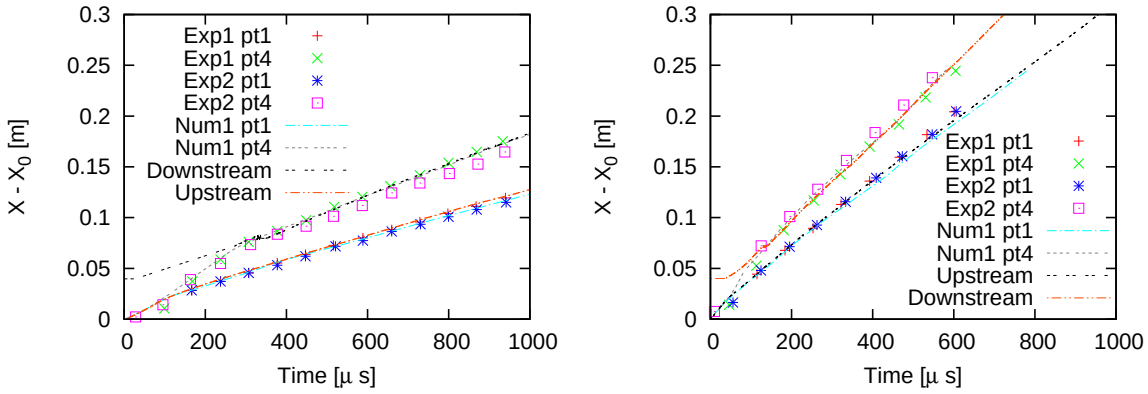
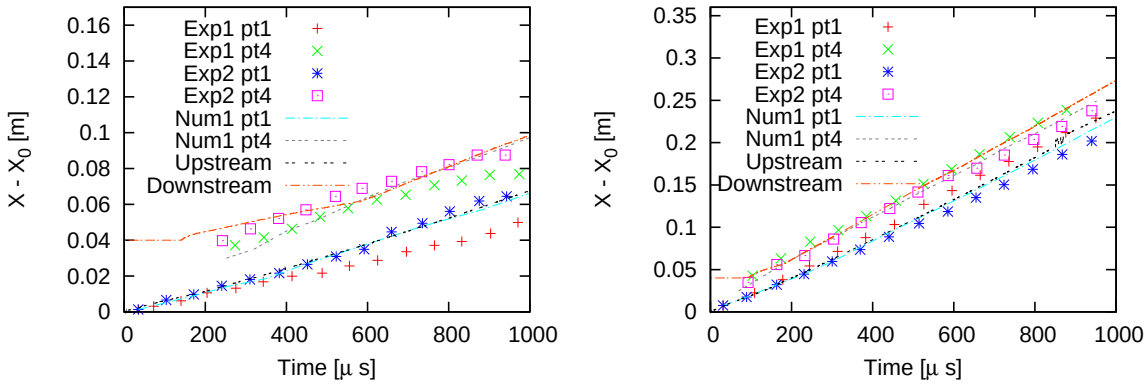


Figure 49: Geometry used for the 3-D interaction of a shock and a helium bubble in air.



(a) $Ma = 1.2$. (b) $Ma = 1.7$.

Figure 50: Trajectory of the Helium spherical bubble compared to Layes and Le Métayer[77]. Points 1 and 4 are from Layes and Le Métayer[77], whereas the down- and up-stream are from the current computations. Point 4 does not match the down-stream limit at the beginning of the test, but it does at a later stage.



(a) $Ma = 1.2$. (b) $Ma = 1.7$.

Figure 51: Trajectory of the Krypton spherical bubble compared to Layes and Le Métayer[77]. Points 1 and 4 are from Layes and Le Métayer[77], whereas the down- and up-stream are from the current computations.

7.3.3 Gas-liquid, Bubble-shock interaction in 2-D

Here a more challenging case is presented with greater material differences and very large pressure jumps. An air cylinder in water is impacted by a shock as studied experimentally and numerically by Bourne and Field[16] and Ball et al.[7], respectively. Given that a cylinder was tested, a 2D geometry should capture the main features properly. The geometry is described in figure 52. It was argued by Ball et al.[7] that the inviscid assumption for this simulation is appropriate since the Reynolds number is very large. Furthermore, surface tension can be neglected given that the dynamic effects are greater, i.e. the Weber number (We) is much greater than 1 ($We = \rho u^2 D/\sigma$, where ρ is the density of the environment, u is the characteristic velocity, D is the diameter of the bubble, and σ is the surface tension coefficient). The initial resolution is 50×20 and $l_{max} = 4$. Initially, for the air bubble:

$$(\rho, v_1, v_2, P, \gamma, P_\infty, \alpha_1)^T = \left(1.18\ kg/m^3, 0.0\ m/s, 0.0\ m/s, 101325\ Pa, 1.4, 0.0\ Pa, 0.99999\right)^T$$

For the liquid surrounding the bubble:

$$(\rho, v_1, v_2, P, \gamma, P_\infty, \alpha_1)^T = \left(1000\ kg/m^3, 0.0\ m/s, 0.0\ m/s, 101325\ Pa, 4.4, 6 \times 10^8\ Pa, 0.00001\right)^T$$

and for the post-shock:

$$(\rho, v_1, v_2, P, \gamma, P_\infty, \alpha_1)^T = \left(1323.65\ kg/m^3, -681.577\ m/s, 0.0\ m/s, 1.9 \times 10^9\ Pa, 4.4, 6 \times 10^8\ Pa, 0.\right.$$

The interface and pressure are shown in figure 53 for a shock wave with $1.9 \times 10^9\ Pa$. These results are very close to Ball et al.[7], where similar conditions were used. At large post- over pre-shock pressure ratios ($P_r > 5000$) a jet is formed and an asymmetric collapse occurs [32], like in the current case. The speed of the jet is greater than the post-shock particle velocity, and the difference increases with P_r [16]. Note that after the jet impacts the distal side, the pressure increases due to a "water-hammer" effect. In the current simulations we found a peak pressure of $4.6 \times 10^9\ Pa$,

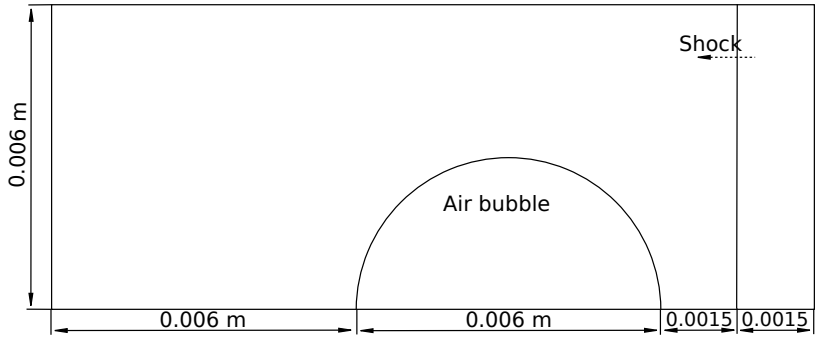
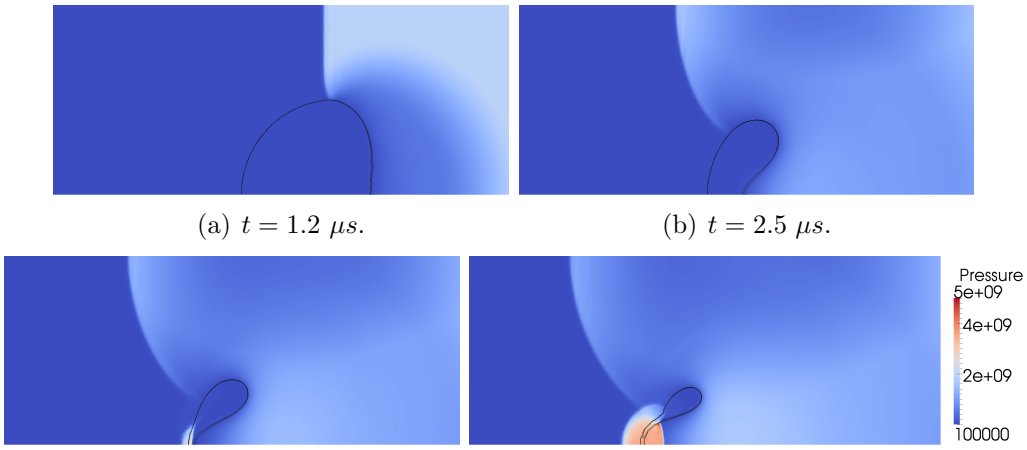


Figure 52: Geometry used for the 2-D interaction of a shock and an air bubble in water.



(a) $t = 1.2\ \mu s$. (b) $t = 2.5\ \mu s$.

(c) $t = 2.9\ \mu s$. (d) $t = 3.2\ \mu s$.

Figure 53: Pressure and interface for a gas bubble collapsing in liquid for a shock with $1.9 \times 10^9\ Pa$.

Table 17: Post-shock conditions and collapse time for a cylindrical air bubble in water.

$\rho\ [kg/m^3]$	$v_1\ [m/s]$	$P\ [Pa]$	$\tau\ [s]$	$\tau\ [s]$ (Exp.[16])	$\tau\ [s]$ (Num.[126])
1323.65	-681.577	1.9×10^9	$2.8\ \times 10^{-6}$	$(1.8 \pm 0.2)\ \times 10^{-6}$	–
1407.43	-1006.56	3.5×10^9	$2.0\ \times 10^{-6}$	$(1.0 \pm 0.5)\ \times 10^{-6}$	$2.072\ \times 10^{-6}$
1437.3	-1170.08	4.5×10^9	$1.7\ \times 10^{-6}$	–	–
1458.7	-1315.1	5.5×10^9	$1.6\ \times 10^{-6}$	–	–

very close to the results by Ball et al.[7]. It seems that some air is caught in between the two interface even after the collision. A similar phenomenon was observed by Ball et al.[7], but they could not confirm if that was a numerical issue or actually physical. According to Qian and Law[105], when two liquid interfaces collide under a certain regime, small gas bubbles should be retained in the liquid, however, it is very difficult to show experimentally.

The collapse time τ, defined as the time required for the jet to hit the distal side, was calculated for different post-shock pressures. Table 17 shows the results together with the corresponding post-shock density, velocity and pressure based on the Rankine-Hugoniot relations for each case. The results match the simulations by Shyue[126] and are in the same order as the experiments by Bourne and Field[16]. Figure 54 shows the collapse time versus the initial shock pressure ratio. Swantek and Austin[130] compared more numerical and experimental tests, and found that the simulations tend to have a collapse time twice as big as the experimental cases.

7.3.4 Gas-liquid, Bubble-shock interaction in 3D

The same case as above for a post-shock pressure of $1.9 \times 10^9\ Pa$ is now conducted in a 3-D domain with a spherical bubble. The grid has a size of $50 \times 20 \times 20$ elements and $l_{max} = 3$. Due to the symmetry of the problem, only one fourth of the geometry is simulated. The interface is shown in figure 55 colored by the pressure for different

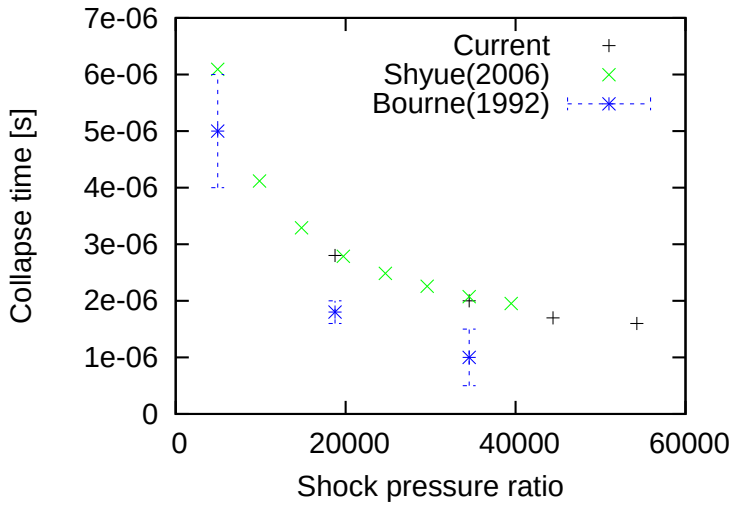
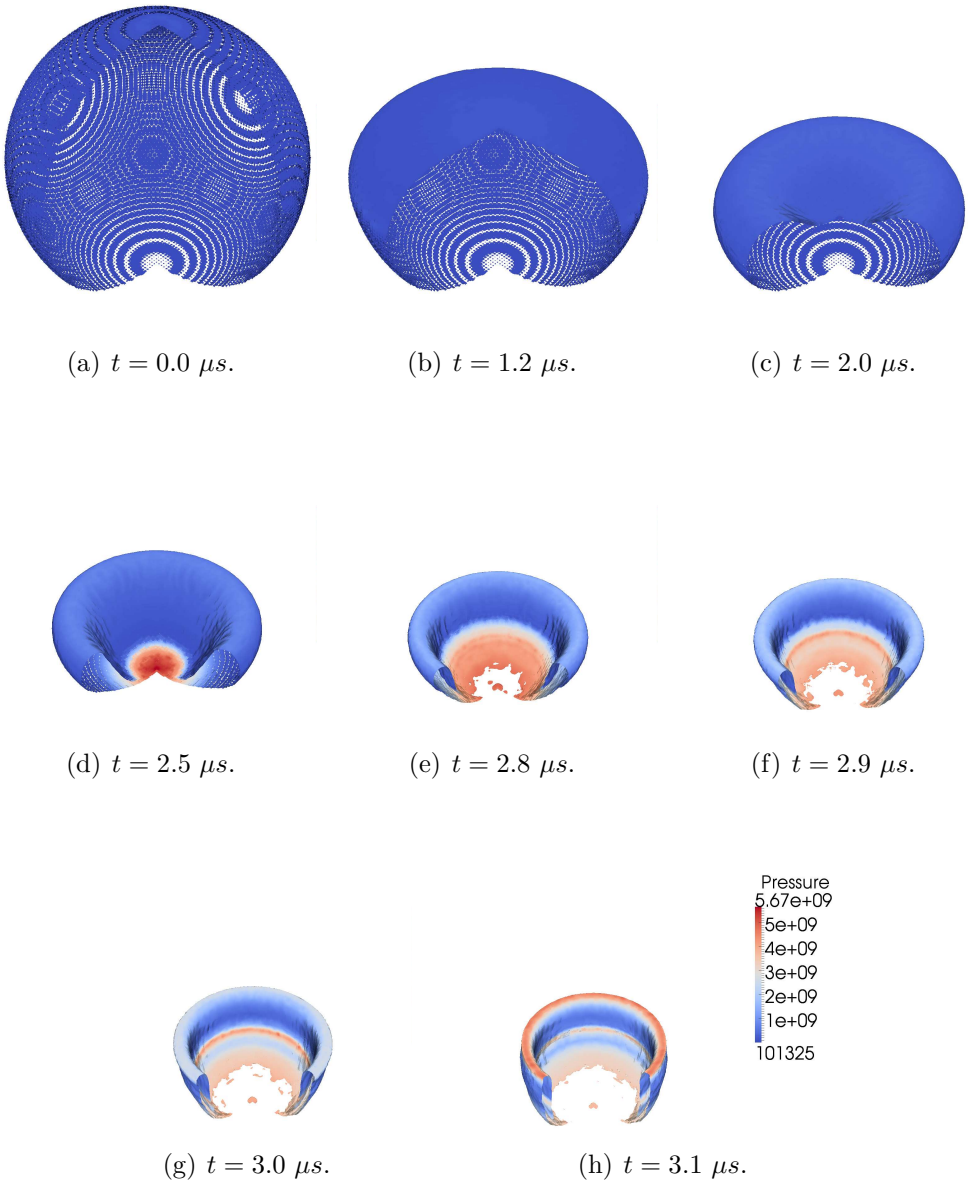


Figure 54: Collapse time for a cylindrical air bubble in water.

time steps, where $t = 0.0\ s$ is when the shock first hits the bubble. The result is reflected in the Y and Z plane for visual clarity. At $t = 1.2\ \mu s$ the up-stream side is approximately flat. At $t = 2.0\ \mu s$ the jet is starting to form. At $t = 2.5\ \mu s$ the jet impacts the down-stream face and produces a water-hammer effect with a pressure peak exceeding $5.6 \times 10^9\ Pa$. After the impact a gaseous film is formed. This film may break up and form small bubbles. In order to capture the film accurately, a much greater resolution would be needed together with viscous and surface tensions effects. Given that viscosity and surface tension were neglected, the thickness of the film is grid dependent. Therefore, we see at $t = 2.8\ \mu s$ that the interface looks broken up and the bubble adopts the shape of a ring. This means that the film got diffused and cannot be captured anymore.

Looking at these results and the 2D case above, we note that the pressure increase is greater and the deformation is faster for the sphere than for the cylinder due to a greater convergent effect.



(a) $t = 0.0\ \mu s$. (b) $t = 1.2\ \mu s$. (c) $t = 2.0\ \mu s$.

(d) $t = 2.5\ \mu s$. (e) $t = 2.8\ \mu s$. (f) $t = 2.9\ \mu s$.

(g) $t = 3.0\ \mu s$. (h) $t = 3.1\ \mu s$.

Figure 55: Pressure and volume-fraction iso-surface of a 3D bubble interacting with a shock.

7.4 Discussion

We presented a successful discontinuous Galerkin method for multifluid inviscid flows. The conservation of mass, momentum, and total energy was treated as in the Euler equation and the numerical flux used was the HLLC. Therefore, conservation of the three variables was obtained. The convection of the volume fraction was modeled as a Hamilton-Jacobi equation. Therefore, a truly unified formulation that does not require special treatment of the interface was obtained, which is not found with other common methods (e.g., ghost-fluid method and volume of fluid). At the same time, the potential excessive numerical diffusion that DIM may have did not present a serious problem. The moment limiter applied on primitive variables showed very small numerical oscillations close to the computational precision. Adaptive mesh refinement proved to be effective for the problems treated, since more resolution was provided around the interface while a coarse grid was kept in areas with less interest. Even though the viscous effects were neglected in the analysis of shock-bubble interactions, the comparison with experiments validated the inviscid aspects of the current solver in two and three dimensions. Therefore, we were able to gain some insight into the dynamics of the deformation of a bubble after it is impacted by a shock. The collapse time computed for an air bubble in water matched the numerical results compared to.

CHAPTER VIII

SUB- AND SUPER-CRITICAL FLOWS USING LDG

8.1 Introduction

The 5-eq. model is slightly more complex than the 4-eq. model, but permits to recover the individual thermodynamic properties of each fluid. These properties are required to use sophisticated equations of state, which are commonly needed to simulate fluids close to their critical point. Thus, in this chapter the 5-eq. model is used and the cases are extended to sub- and super-critical conditions. Note that in the later case, the 5-eq. model converges to the Navier-Stokes equations.

Even though the multi-fluid formulation based on the RKDG method was proven to be accurate in the previous chapter, it can also be unstable if no ad-hoc approach is used, which could be avoided in order to keep the formulation as unified as possible. Hence, in this chapter the multi-fluid formulation based on the LDG method is evaluated.

All the cases used above to test the limiter with one fluid are now used to challenge the limiter with a two-fluid formulation. Also, the efficiency and accuracy of the scheme with AMR and different orders are evaluated. The 5-eq. model together with the new method to work with multiple EOSs simultaneously is verified in 1D.

Some shock-bubble interaction cases studied above are used again to validate the new formulation with LDG and the 5-eq. model. The numerical results in the literature do not fully agree quantitatively with the experiments. Given that current methodology is compatible with more accurate EOSs, it is assessed if the thermodynamic closure can correct the discrepancy.

8.2 Verification and Validation of LDG

Various test cases used in past studies are used to establish the capability of this new numerical algorithm. The details of the test cases and the rationals for them are summarized in Table 18.

Table 18: Summary of two-phase inviscid cases for verification and validation.

Test case	Purpose	Section
Order of convergence for multi-phase flow	Show the order of accuracy and efficiency for multi-phase flows.	8.2.1
Two-phase Sod's problem	Test the limiter and AMR for inviscid two-phase flow.	8.2.2
Two-phase Lax's problem	Test the limiter and AMR for inviscid two-phase flow.	8.2.3
Two-phase shock-entropy waves interaction	Test the limiter and AMR for inviscid two-phase flow.	8.2.4
Real-gas shock-tubes	Test the real-gas formulations close to the critical point.	8.2.5
Shock-tube with mixed EOS.	Test the mixed-EOS formulation.	8.2.6
Underwater explosion.	Verify the symmetry of the scheme in 2D and 3D.	8.2.7

8.2.1 Spatial accuracy

In order to assess the accuracy of this new approach, the order of convergence in space is studied using a one-dimensional convection of change in volume fraction. The domain is $\Omega = [0, 1]\ m$ with periodic boundaries. The material properties are equal for both phases ($\gamma = 1.4$, $P_\infty = 0.0$). Three cases are studied all with the initial conditions of $P = 101325\ Pa$, $\rho_1 = 1\ kg/m^3$, $\rho_2 = 1\ kg/m^3$ and:

- Case A:

$$v_1 = 100\ m/s,$$

$$\alpha_1 = \begin{cases} 0.9999, & 0.25\ m \leq x_1 < 0.75\ m \\ 0.0001, & \text{otherwise} \end{cases}$$

- Case B:

$$v_1 = 100\ m/s, \quad \alpha_1 = 0.5 + 0.499\sin(2\pi x_1)$$

- Case C:

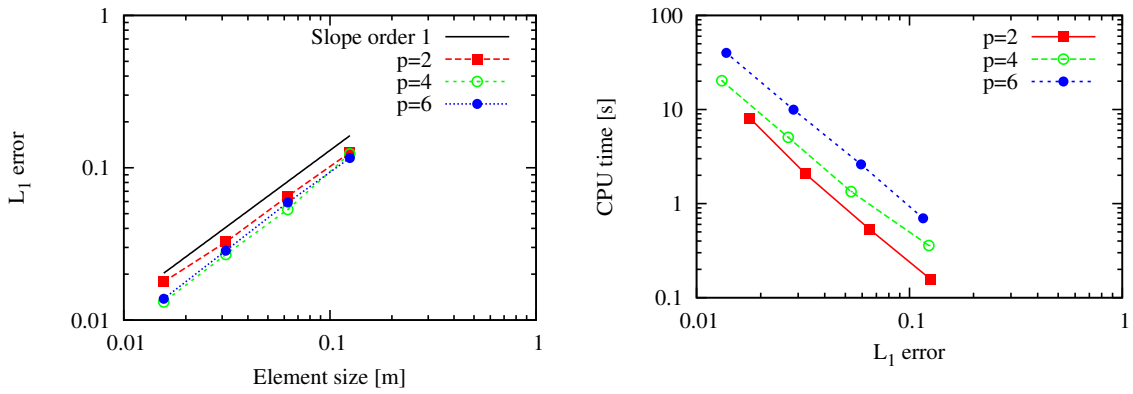
$$v_1 = -100\ m/s, \quad \alpha_1 = 0.5 + 0.499\sin(2\pi x_1)$$

Note that the only difference between cases B and C is the direction of the flow. The exact solution for these flows is known and therefore provides a clear approach to evaluate the accuracy. The number of cells N and the polynomial order p are varied in this study. We use $p > 1$ because $p = 1$ might be too diffusive for most applications, particularly when smooth and sharp discontinuities coexist. We only show results of even number of p, although other cases were run and showed consistent results as well. The MB-AP-TVD detector is used with the ML using primitive variables. The L_1-norm of the error, e_{L_1}, after one period ($t = 0.01\ s$) is computed with respect to the exact solution as

$$e_{L_1} = \sum_{l=1}^{N} \int_{\Omega_l} |\alpha_l - \alpha| dx \tag{149}$$

where α_l is the numerical solution and α is the exact solution. The same error is computed for cases B and C for the derivative ($\frac{\partial \alpha}{\partial x_1}$) based on the auxiliary functions $\mathbf{q}_{\alpha m}$ and $\mathbf{q}_{\alpha p}$. Figure 56(a) shows the error vs. the number of elements. The slopes indicate the order of convergence. Note that when a discontinuity is present, the actual order of accuracy cannot be greater than 1. The CPU time vs. the error is plot in Fig. 56(b). Curves closer to the bottom left corner represent more efficient schemes. This means that increasing the order of the scheme for a problem where discontinuities dominate is not beneficial.

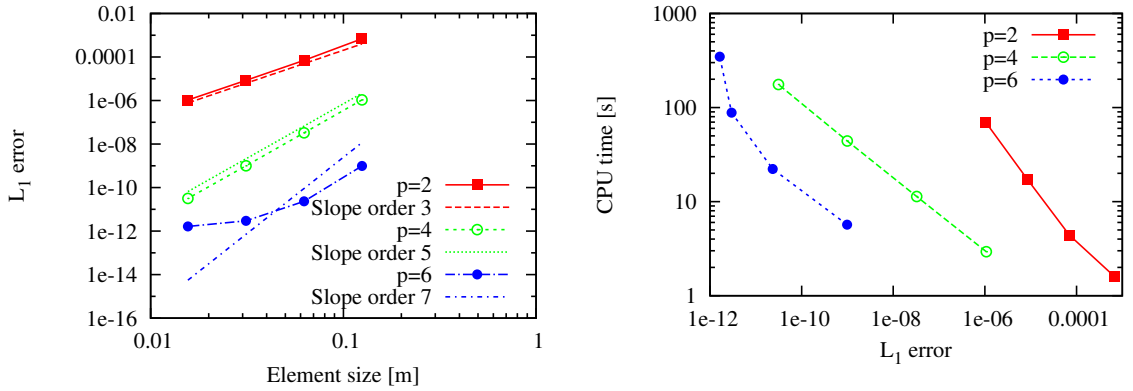
Similar results are shown in Fig. 57 for case B, which has a smooth solution. Note that the order of accuracy matches the theoretical order of $p + 1$ [26]. Also, in this



(a) Grid convergence. (b) Efficiency.

Figure 56: Accuracy study for different orders with a discontinuous solution (case A).

smooth case increasing the order of the scheme improves its efficiency.

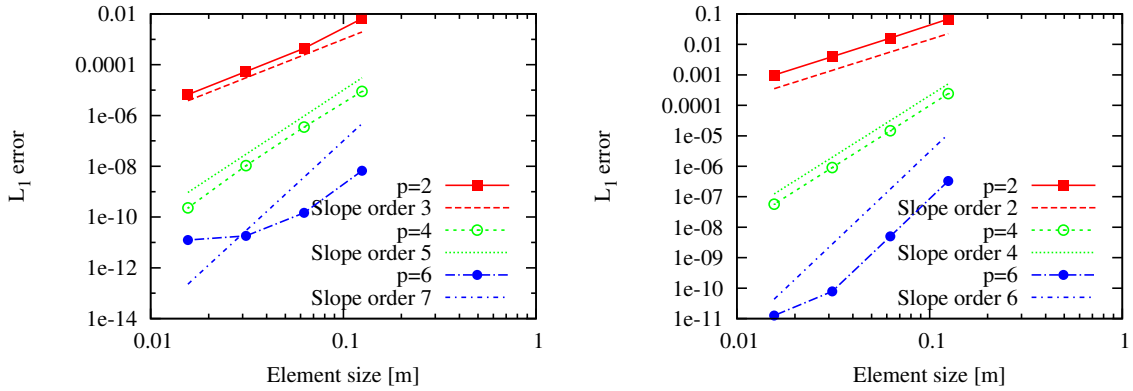


(a) Grid convergence. (b) Efficiency.

Figure 57: Accuracy study for different orders with a smooth solution (case B).

According to the LDG method, the auxiliary variable should also have an order of accuracy of $p+1$ for elements of order p [26]. The current scheme to solve the convection of the volume fraction makes use of two auxiliary variables, one doing up-winding and the other one doing down-winding. In case B, the flow is moving to the right so $\mathbf{q}_{\alpha m}$ is up-winding while $\mathbf{q}_{\alpha p}$ is down-winding. In case C, the opposite is true. The grid convergence for these auxiliary variables are presented in Fig. 58. Note that

the auxiliary variables that do up-winding achieve an optimal order of approximately $p+1$, while the other auxiliary variables only achieve an approximate order of p. It is important to remark that in spite of this drop in accuracy in some of the underlying auxiliary variables, the volume fraction has an optimal order of $p+1$ (see Fig. 57(a)). The plots in Fig. 57 are not needed for case C because the exact same results are obtained.



(a) Case B, $\mathbf{q}_{\alpha m}$. (b) Case B, $\mathbf{q}_{\alpha p}$.

(c) Case C, $\mathbf{q}_{\alpha m}$. (d) Case C, $\mathbf{q}_{\alpha p}$.

Figure 58: Accuracy study for the auxiliary variable for different orders (cases B and C).

Finally, LIM1 and LIM2 are compared for case A and the efficiencies are shown in Fig 59. LIM2 reduces the error for a negligible change in CPU time, so it is more efficient than LIM1 and, thus, it is used for all the other test cases by default. Given that discontinuities dominate in the following problems, the order of the elements

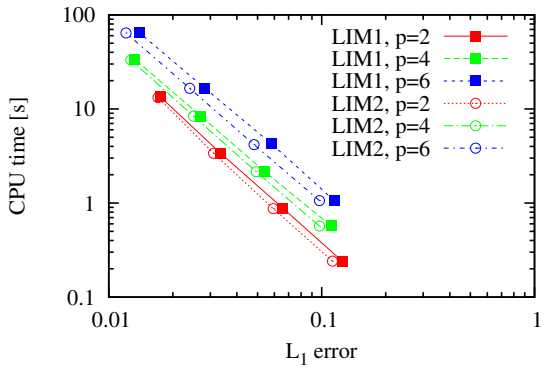


Figure 59: Comparison between LIM1 and LIM2 for case A.

used is $p = 2$. As remarked above for Fig. 56(b), it is more efficient to increase the number of elements than their order p when discontinuities dominate.

8.2.2 2-phase Sod problem

This test evaluates the behavior of the method for Riemann problem with a material discontinuity. The initial states and material properties are [140]:

$$(\rho, v_1, P, \alpha_1)^T = \begin{cases} (1, 0, 1, 0.9999)^T & \text{for } x < 0.0 \\ (0.125, 0, 0.1, 0.0001)^T & \text{for } x \geq 0.0 \end{cases}$$

The SG EOS is used with this material properties: $\gamma_1 = 1.4$, $P_{\infty,1} = 0$, $\gamma_2 = 1.6$ and $P_{\infty,2} = 0$. The 1-D domain is $[-0.5, 0.5]$, has 16 elements and $l_{\max} = 3$. An "exact" solution is estimated with 1024 elements with no adaptivity. The solution at $t = 0.2$ and the level of each cell are shown in Fig. 60 for primitive and conservative limiting. The grid is adapted at discontinuities and at sharp changes. Limiting with conservative variables presents large oscillations at the material discontinuity.

8.2.3 2-phase Lax problem

The initial conditions are [140]:

$$(\rho, v_1, P, \alpha_1)^T = \begin{cases} (0.445, 0.699, 3.527, 0.9999)^T & \text{for } x < 0.0 \\ (0.5, 0, 0.8565, 0.0001)^T & \text{for } x \geq 0.0 \end{cases}$$

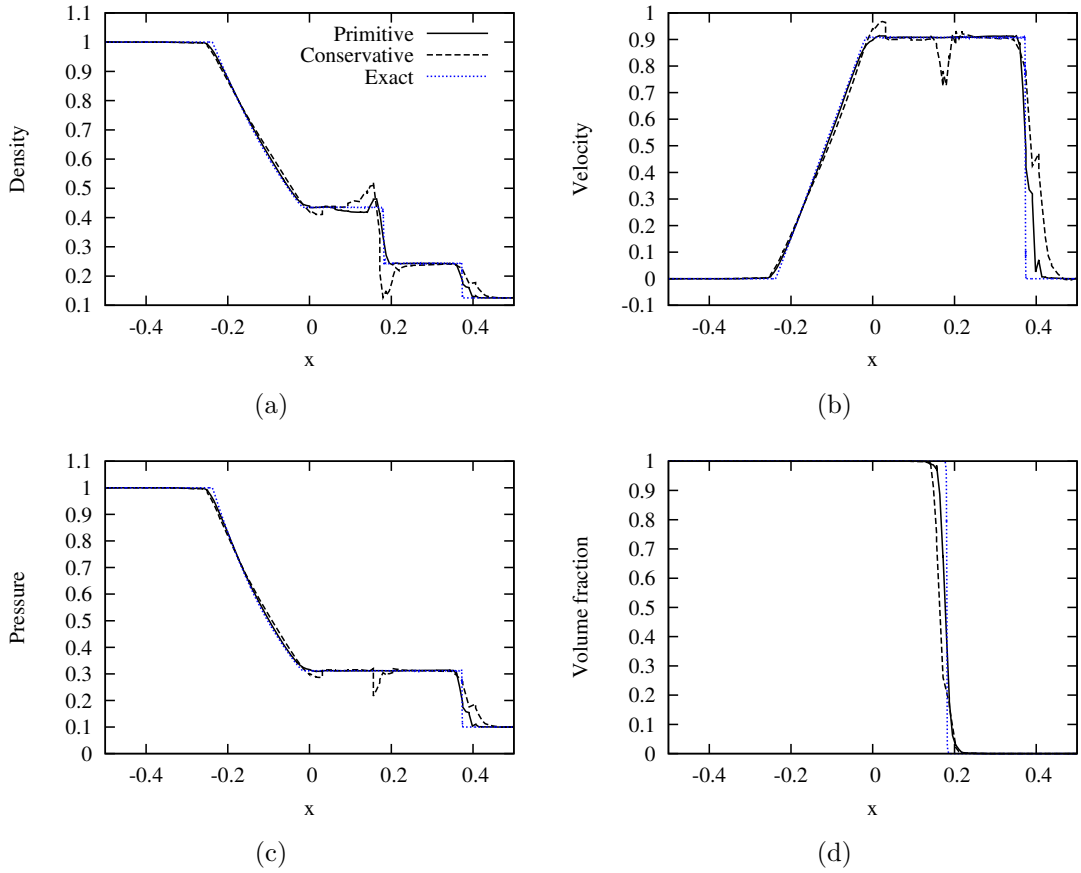


Figure 60: 2-phase Sod problem at $t = 0.2$.

The SG EOS is used with this material properties: $\gamma_1 = 1.4$, $P_{\infty,1} = 0$, $\gamma_2 = 1.6$ and $P_{\infty,2} = 0$. The 1-D domain is $[-0.5, 0.5]$, has 32 elements and $l_{\max} = 3$. An "exact" solution is estimated with 1024 elements and no adaptivity. The solution at $t = 0.13$ and the level of each cell are shown in Fig. 61 for primitive and conservative limiting. The grid adapted at discontinuities and sharp changes. Limiting with conservative variables presents oscillations at the material discontinuity.

8.2.4 2-phase shock-entropy interaction

This test evaluates the scheme for interactions of smooth and sharp variations with a material discontinuity. The initial states and material properties are [140]:

$$(\rho, v_1, P, \alpha_1)^T = \begin{cases} (3.857143, 2.629369, 10.333333, 0.9999)^T & \text{for } x < -4.0 \\ (1 + \sin(5.0x), 0.0, 1.0, 0.0001)^T & \text{for } x \geq -4.0 \end{cases}$$

The material properties are $\gamma_1 = 1.4$, $P_{\infty,1} = 0.0$, $\gamma_2 = 1.666666$ and $P_{\infty,2} = 0.0$. The 1-D domain is $[-5, 5]$ and $l_{\max} = 3$. An "exact" solution is estimated using 2000 elements with no adaptivity and primitive limiting. The density at $t = 1.8$ and the level of each element are shown in Fig. 62. It is clear that when starting with 100 elements and $l_{\max} = 3$, it does not have enough resolution to capture some fluctuations. However, all the features can be captured with 200 elements and $l_{\max} = 3$. Another strategy to provide better resolution could have been to increase $l_{\max}$. Note that the adaptivity algorithm refines appropriately where the solution presents sharp changes and allows for an efficient capturing of all the features. The limiting with conservative variables present an unacceptable spike on the left hand side of the material discontinuity for 100 elements. Given that, in general, limiting with primitive variables has performed better, limiting with conservative variables will not be used for the rest of the current study.

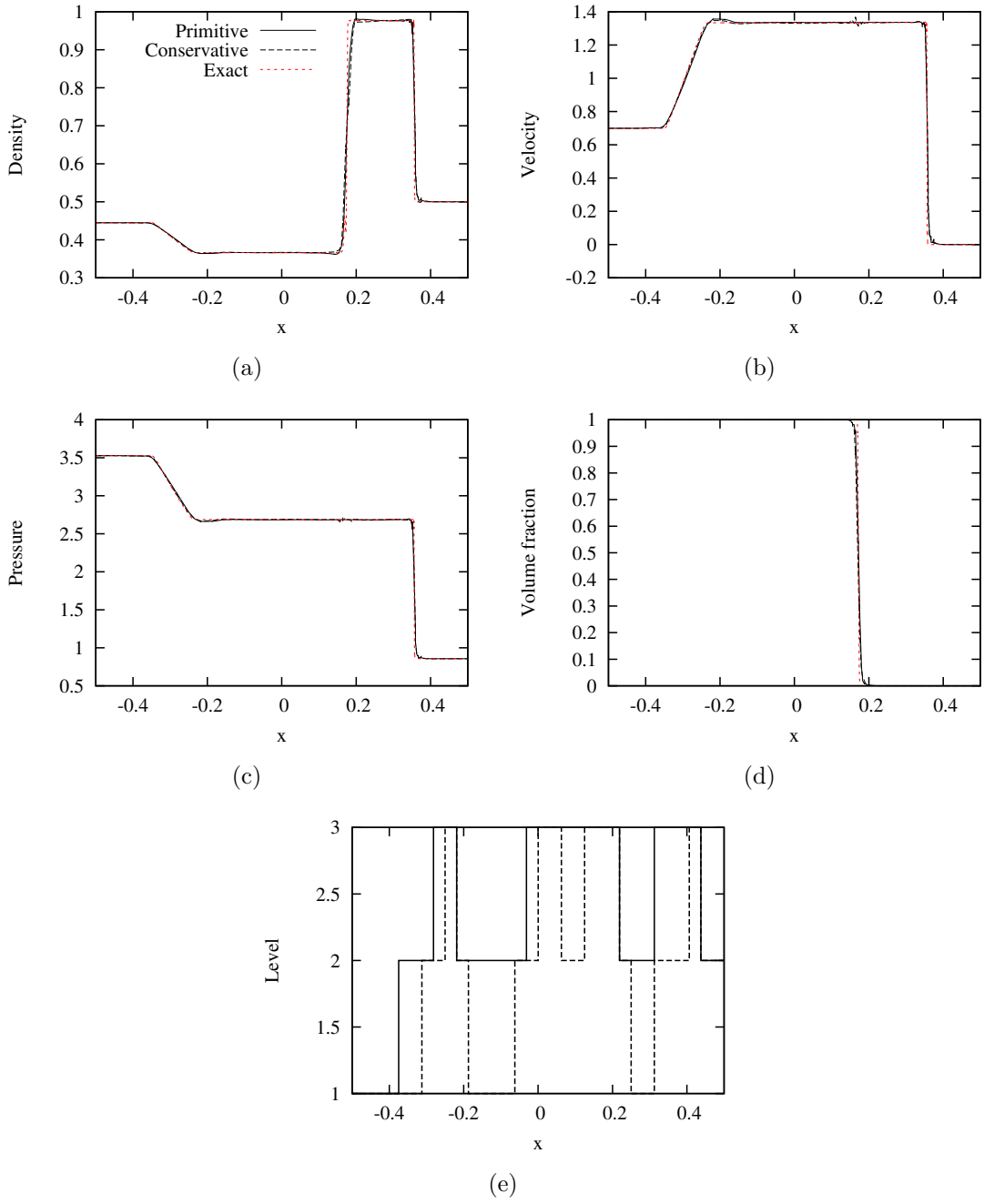
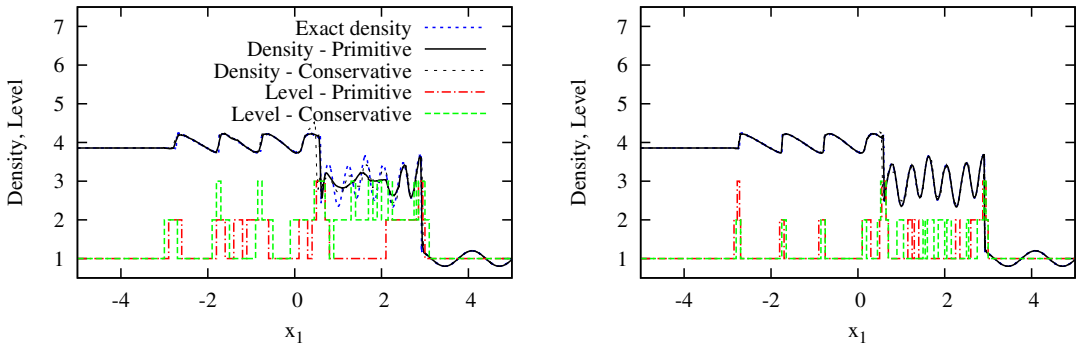


Figure 61: 2-phase Lax problem at $t = 0.13$



(a) 100 elements.

(b) 200 elements.

Figure 62: Density profile at $t = 1.8$ for the shock-entropy interaction case.

8.2.5 Shock tube close to the critical point

Here we study a shock tube problem at different conditions close to the critical point, compare possible thermodynamic approaches, and evaluate the implementation of the thermodynamic closures. The shock tube has a length of 1 m and the discontinuity is located in the middle. The domain has 128 elements and $l_{\max} = 3$. The substance used is carbon dioxide, for which $MW = 0.04401\ kg/mol$, $P_c = 7.382\ MPa$, and $T_c = 304.19\ K$. For constant specific heat c_P^0 we use $c_P^0 = 873\ J/(kg\ K)$. The temperature is the same on both sides, $T = T_c$. Two conditions are tested:

1. $P_{left} = P_c/2$, $P_{right} = P_c/4$ (sub-critical pressures)

2. $P_{left} = 4P_c$, $P_{right} = 2P_c$ (super-critical pressures)

In order to study real-gas effects we simulate cases with the IG EOS, the VDW EOS, and the PR EOS. The results at $t = 0.5\ ms$ are shown in Figs. 63 and 64. The "-C" and "-V" cases refer to constant and variable c_P^0, respectively. Given that the temperature range is small, constant or variable c_P^0 provide similar solutions for all the EOSs. The differences between the different EOSs at high pressures show that low density assumption of the IG EOS is not acceptable. Even within the case with sub-critical pressures, a larger difference between the EOSs is observed for high pressure.

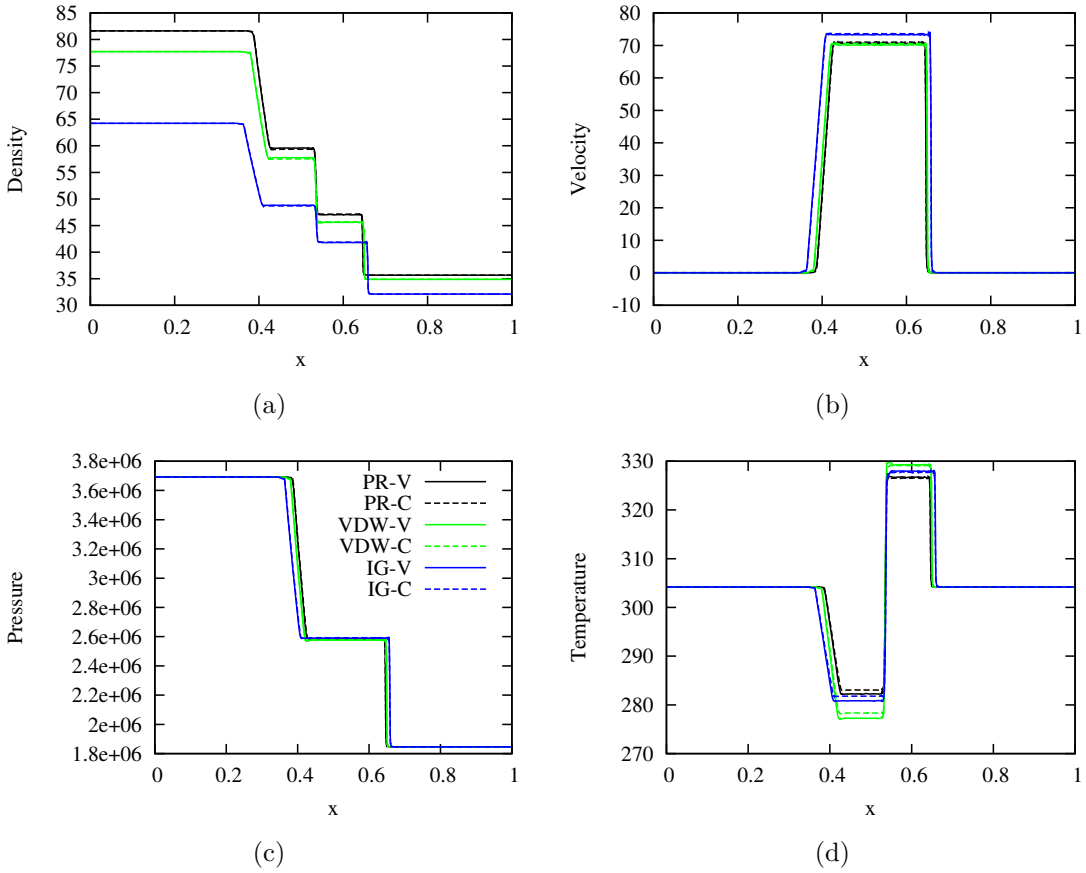


Figure 63: Shock tube 1, at sub-critical pressures.

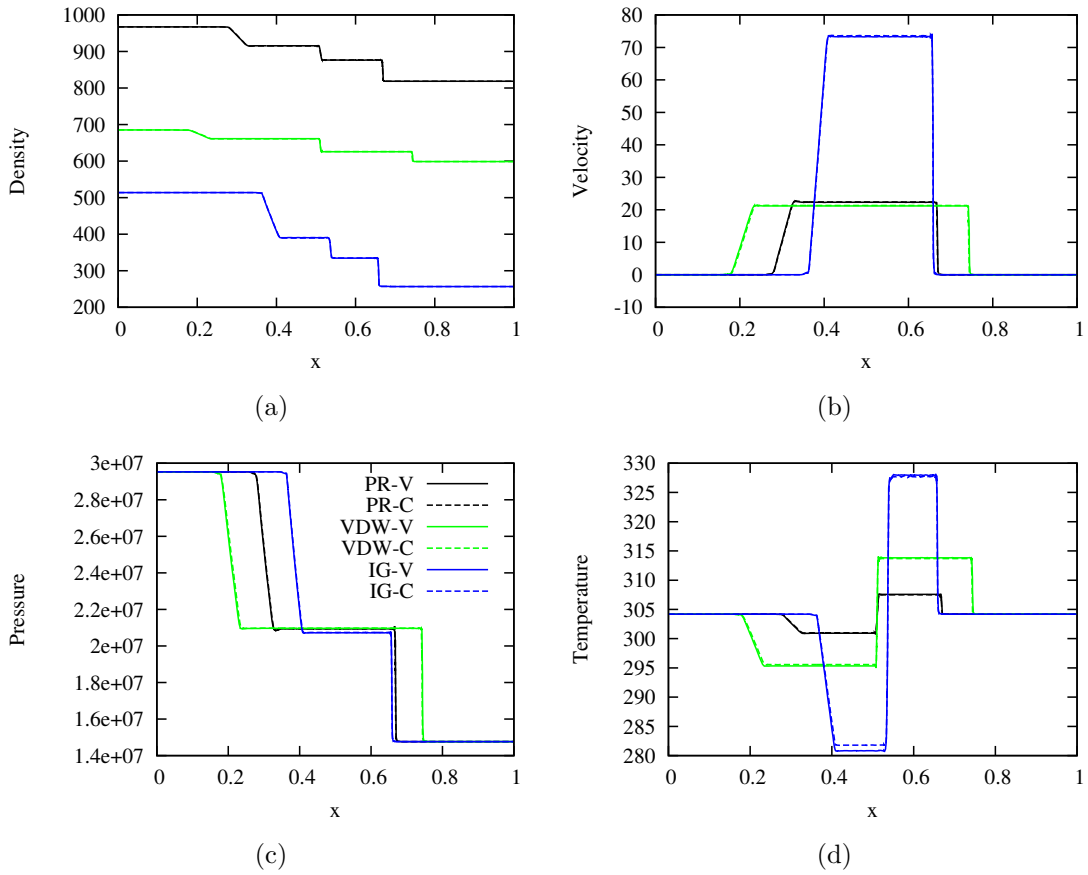


Figure 64: Shock tube 2, at super-critical pressures.

These results highlight the need of the appropriate EOS based on the conditions studied.

8.2.6 Multiple EOS

A shock-tube problem with a contact discontinuity is simulated with mixed EOSs and compared to the literature. The 1-D domain $[0, 1]$ m has 100 elements and $l_{\max} = 3$. The initial value is:

$$(\rho, u_1, P) = \begin{cases} (50\ kg/m^3, 0\ m/s, 10^9\ Pa) & x < 0.4\ m \\ (500\ kg/m^3, 0\ m/s, 10^5\ Pa) & x \geq 0.4\ m \end{cases}$$

The material at $x < 0.4$ m is modeled with the VDW EOS or the PR EOS, and has $T_c = 4.99$ K, $P_c = 185.2$ kPa, $w = 0.04$, $MW = 0.028$ kg/mol, and $c_P^0 = 1039.4$ $J/(kg\ K)$. The substance at $x \geq 0.4$ m uses the SG EOS with $\gamma = 4.4$, $P_\infty = 6.0 \times 10^8$ Pa, and $c_V = 1000$ $J/(kg\ K)$. The results at $t = 0.19$ ms are shown in Fig. 65 and compared to earlier results [1], where the VDW EOS was used. Note that the only non-negligible difference between the VDW and the PR EOS is the temperature. This is expected since at equal density and pressure, the temperature may differ for different EOSs, particularly at high pressure. If the conditions for $x < 0.4$ m had been closer to the critical point, the results would have differed even more. Our technique for multiple EOSs is validated since it shows the same results as the original scheme in [1].

8.2.7 Underwater explosion

An air bubble exploding underwater in 2D is conducted for conditions used earlier [124, 160] to study the isotropy of the scheme. The domain is the unit square $[0, 1]m \times [0, 1]m$ with periodic boundaries, 20×20 root cells, and $l_{\max} = 4$. The material properties are $\gamma_1 = 1.4$, $P_{\infty,1} = 0.0$ Pa, $\gamma_2 = 5.5$, and $P_{\infty,2} = 1.505$ Pa. The center of the bubble is at the center of the domain and its radius is 0.2. Initially, the flow is stagnant and inside the bubble $\rho = 1.241$ kg/m^3 and $P = 2.753$ Pa, while

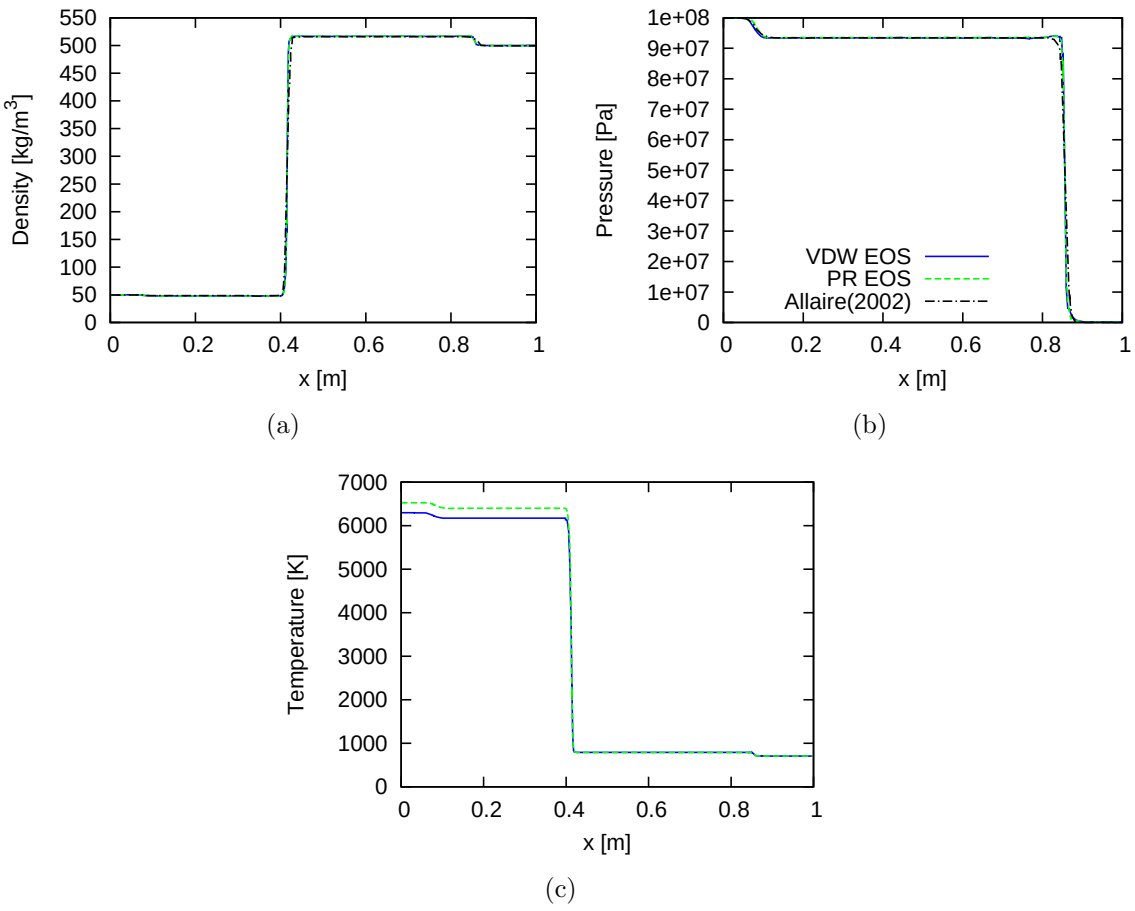
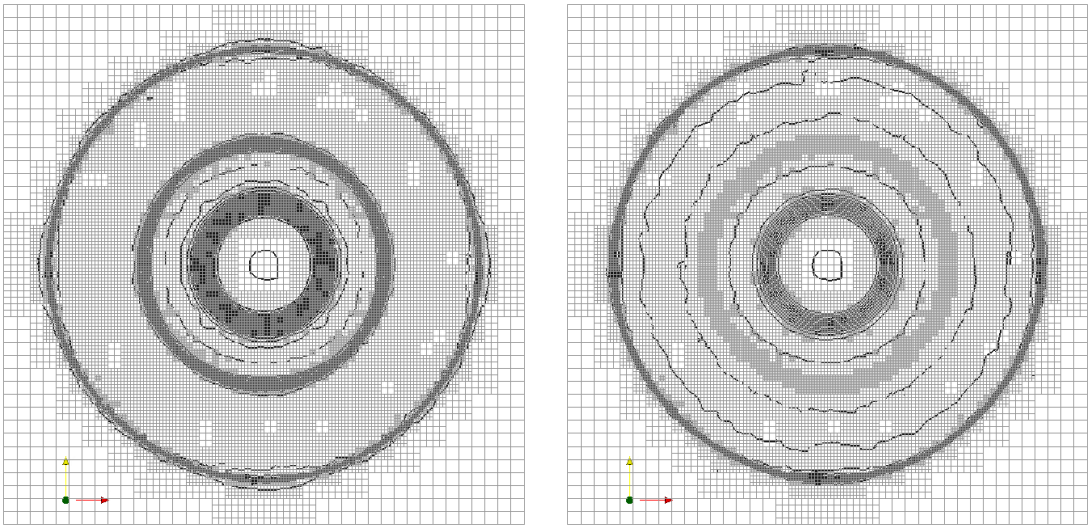


Figure 65: Shock-tube problem at $t = 0.19\ ms$ with different EOS, SG EOS for $x \geq 0.4\ m$, and VDW EOS or PR EOS for $x < 0.4\ m$.

outside the bubble $\rho = 0.991\ kg/m^3$ and $P = 3.059 \times 10^{-4}\ Pa$. Figure 66 shows 30 density and pressure contours at $t = 0.058\ s$ together with the adapted grid. The density and pressure along a horizontal centerline are presented in Fig. 67. The current results (solid line) are compared with those presented in [124] (dashed line) using a front-tracking algorithm, 500 mesh points, and a one-dimensional grid with radial symmetry. The rounded corners of the current results could be sharpened increasing

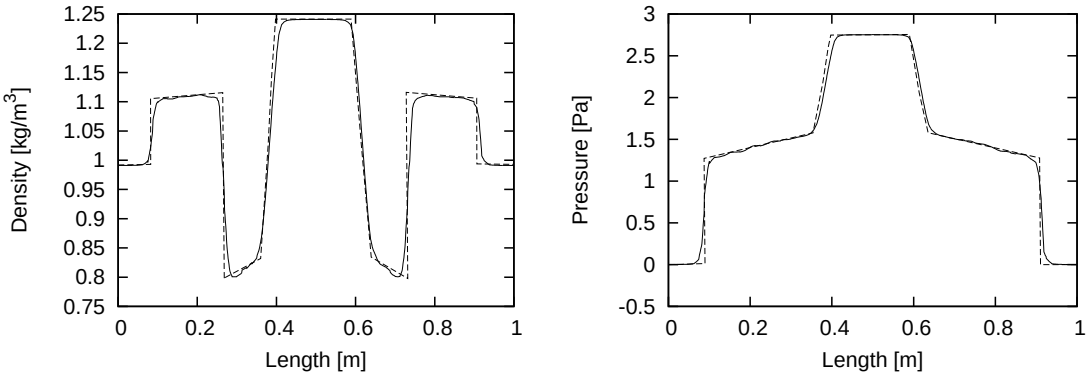


(a) Density. (b) Pressure.

Figure 66: Contour plots for the cylindrical explosion at $t = 0.058\ s$ with the adapted grid.

the resolution.

Now, the same case is conducted in 3D, but in the unit cube $[0,1]m \times [0,1]m \times [0,1]m$ with $20 \times 20 \times 20$ root cells and $l_{\max} = 4$. The final number of leaf-cells is $237,957$, what represents $6,424,839$ nodes. An equivalent uniform grid would have had $4,096,000$ cells and $110,592,000$ nodes. The density and pressure over four lines that go through the center are shown in Fig. 68 at $t = 0.058\ s$. The first three centerlines are aligned with the x_1, x_2, and x_3 axes, while the fourth one



(a) Density. (b) Pressure.

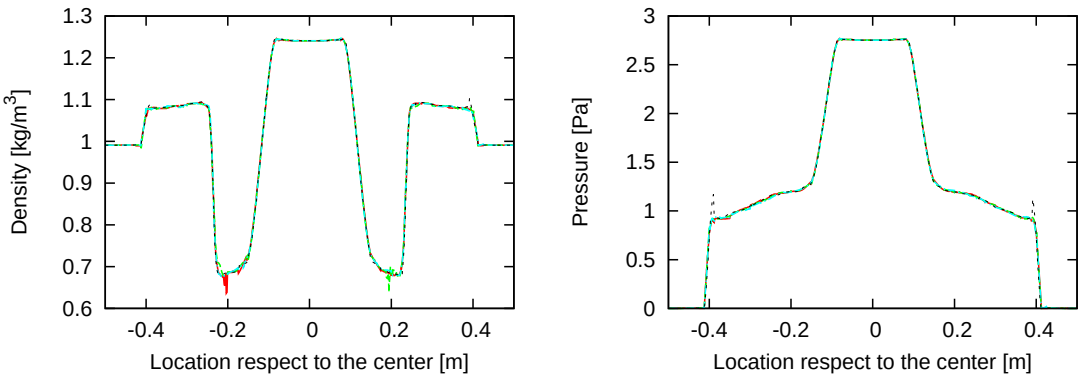
Figure 67: Profile along a horizontal centerline for the cylindrical explosion at $t = 0.058\ s$. Current results are solid lines, and reference [124] are dashed lines.

goes through the origin. Note that the four directions show the same results, what verifies the isotropy of the scheme in 3D. There are some small oscillations close to the material jump in the density and at the shock in the pressure in the (diagonal direction only). We observed that these instabilities can be reduced by applying a more aggresive detector and/or limiter, however, that usually carries the cost of more numerical diffusion in other areas. Nonetheless, these fluctuations are relatively small and do not reach unphysical values.

8.3 Bubble collapse

A more challenging case is presented with greater material differences, very large pressure jumps. This case is of practical interest for many applications.

Bourne and Field [16] studied experimentally the interaction between an air cylinder in a substance similar to water at atmospheric pressure for shock pressures from 0.3 GPa to 3.5 GPa. Ball et al. [7] focused on a numerical test of an air bubble with a diameter of 6 mm submerged in water at atmospheric pressure and impacted by a shock of 1.9 GPa. They showed that viscous effects could be neglected. They also estimated that the error in temperature due to the ideal gas assumption at the time of



(a) Density.

(b) Pressure.

Figure 68: Profile over four lines that cross the center for the spherical explosion at $t = 0.058\ s$. The lines go through the points $(0, 0.5, 0.5)$ (solid red line), $(0.5, 0, 0.5)$ (dashed green line), $(0.5, 0.5, 0)$ (dot-dashed cyan line), and $(0, 0, 0)$ (dotted black line).

impact is around 2.6% and around 13.0% at later stages respect to the van der Waals equation of state. According to their analysis [7], the vibrational relaxation time is in the order of magnitude of the collapse time, leading to specific heats in between the equilibrium and the frozen conditions. Furthermore, a non-dimensional analysis using the Weber number (We) suggests for this case that surface tension effects are negligible since $We \gg 1$ ($We = \rho u^2 D/\sigma$, where ρ is the density of the environment, u is the characteristic velocity, D is the diameter of the bubble, and σ is the surface tension coefficient). Similar conditions as in [7] were also studied numerically [126, 32] assuming constant specific heat and ideal gas for the air in the bubble.

Given that a cylinder was tested in [16], a 2-D geometry should be sufficient to capture the main features properly. The geometry is described in Fig. 69 and the initial resolution is 50×20. Two maximum levels are used to verify grid dependency: $l_{\max} = 4$ and $l_{\max} = 5$. Initially, for the air bubble ($\gamma_1 = 1.4$, $P_{\infty,1} = 0.0\ Pa$):

$$(\rho, v_1, v_2, P, \alpha_1)^T = \left(1.18\ kg/m^3, 0.0\ m/s, 0.0\ m/s, 101325\ Pa, 0.99999\right)^T,$$

for the liquid surrounding the bubble ($\gamma_2 = 4.4$, $P_{\infty,2} = 6.0 \times 10^8\ Pa$):

$$(\rho, v_1, v_2, P, \alpha_1)^T = \left(1000\ kg/m^3, 0.0\ m/s, 0.0\ m/s, 101325\ Pa, 0.00001\right)^T,$$

and for the post-shock:

$$(\rho, v_1, v_2, P, \alpha_1)^T = \left(1323.65\ kg/m^3, -681.577\ m/s, 0.0\ m/s, 1.9 \times 10^9\ Pa, 0.00001\right)^T.$$

The temporal evolution of the interface and pressure are shown in Fig. 70 for a shock

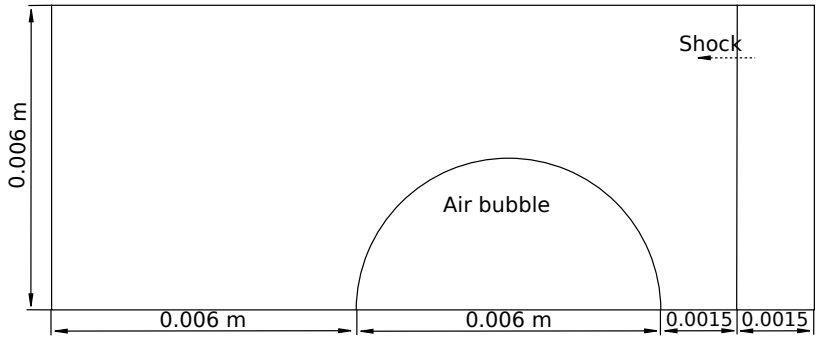
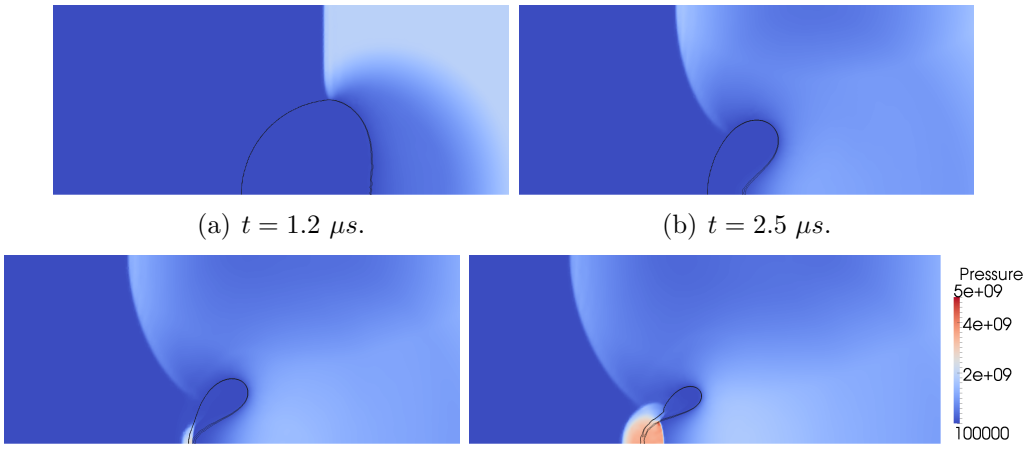


Figure 69: Geometry used for the 2-D interaction of a shock and an air bubble in water.

wave with a post-shock pressure of $1.9 \times 10^9\ Pa$. These results are very close to Ball et al.[7].

Ding and Gracewski [32] defined three regimes. At small post- over pre-shock pressure ratios ($P_r < 300$) the shock is relatively weak and the bubble collapses symmetrically. In the second regime, mixed effects are observed. In the third regime, defined by large post- over pre-shock pressure ratios ($P_r > 5000$), a jet is formed and an asymmetric collapse occurs, as in the current case. The speed of the jet is greater than the post-shock particle velocity, and the difference increases with P_r [16]. Note that after the jet impacts the distal side, the pressure increases due to a "water-hammer" effect. In the current simulations we found a peak pressure of $4.6 \times 10^9\ Pa$, very close to the results by Ball et al. [7]. It seems that some air is caught in between the interface after the collision. A similar phenomenon was observed by Ball et



(a) $t = 1.2\ \mu s$. (b) $t = 2.5\ \mu s$.

(c) $t = 2.9\ \mu s$. (d) $t = 3.2\ \mu s$.

Figure 70: Pressure and interface for a gas bubble collapsing in liquid for a shock with $1.9 \times 10^9\ Pa$.

Table 19: Post-shock conditions for a cylindrical air bubble in water.

Case	$\rho\ [kg/m^3]$	$v_1\ [m/s]$	$P\ [GPa]$
1	1095.2	−161.456	0.3
2	1143.23	−250.257	0.5
3	1323.65	−681.577	1.9
4	1407.43	−1006.56	3.5
5	1437.3	−1170.08	4.5
6	1458.7	−1315.1	5.5

al. [7], but they could not confirm if that was a numerical issue or actually physical. According to Qian and Law [105], when two liquid interfaces collide under a certain regime, small gas bubbles should be retained in the liquid, however, it is very difficult to show experimentally and numerically.

The collapse time t_c, defined as the time required for the jet to hit the distal side, was calculated for different post-shock pressures. Table 19 shows for each case analyzed the corresponding post-shock density, velocity, and pressure based on the Rankine-Hugoniot relations. The collapse time t_c obtained are shown and compared with the simulations by Shyue [126] and experiments by Bourne and Field [16] in Table 20. No significant differences between $l_{\max} = 4$ and $l_{\max} = 5$ are observed,

Table 20: Collapse time for a cylindrical air bubble in water.

Case	t_c [μs]	t_c [μs] (Exp.[16])	t_c [μs] (Num.[126])	t_c [μs] (Num.[7])
1	7.95	40 ± 1	–	–
2	5.95	5 ± 1	–	–
3	2.85	1.8 ± 0.2	–	3.1
4	2.05	1 ± 0.5	2.072	–
5	1.79	–	–	–
6	1.61	–	–	–

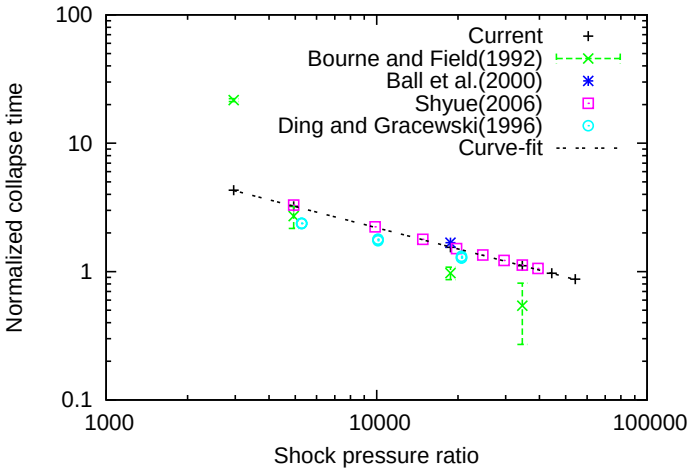


Figure 71: Normalized collapse time, $t_c c_L/R_0$, for a cylindrical air bubble in water. The curve-fit is based on the current study and it is defined as: $344.2P_r^{-0.5488}$.

what suggests that the grid has converged for the current studies. Figure 71 shows the normalized collapse time versus the initial shock pressure ratio in a logarithmic plot together with past results ([16, 7, 126, 32]).The time is normalized as $t_c c_L/R_0$, where c_L is the speed of sound of the liquid initially surrounding the bubble, and R_0 is the initial radius. A curve-fit of the current data is included and it corresponds to $344.2P_r^{-0.5488}$. Ding and Gracewski [32] showed that the collapse time is proportional to the radius of the bubble. Thus, studies with different radius should collapse when the time is being normalized by R_0.

The area of the bubble decreases due to the impact of the shock until it reaches a minimum ($A_{\min}$) after the jet impacts the downstream side. The time to reach the

Table 21: Time taken to reach the minimum area for a cylindrical air bubble in water.

Case	$t_{A\min}$ $[\mu s]$	$t_{A\min}$ $[\mu s]$ (Num.[7])
1	8.94	–
2	7.20	–
3	3.64	3.7
4	2.72	–
5	2.42	–
6	2.20	–

minimum area, $t_{A\min}$, is shown in Table 21. Again, similar results were obtained with $l_{\max} = 4$ and $l_{\max} = 5$. For $P = 1.9\ GPa$ Ball et al. [7] reported $t_{A\min} = 3.7 \times 10^{-6}\ s$. The time (normalized by $t_{A\min}$) vs. the area of the bubble (normalized by $A_{\min}$) are shown in Figure 72. The predicted shape agrees well with earlier numerical and experimental results in [7] (at $P = 1.9\ GPa$) and [130] (at a pressure ratio of $P_r = 421$), respectively. Both references and current studies agree in a linear scale, for which we find a slope of approximately $-3/4$. Note that even though the experimental case is below the asymmetric collapse regime, it shows the same behavior. Bourne and Field [16] found for $P = 0.3\ GPa$ that the area reduction is linear in time. Their plot shows only until the jet impacts, so they miss the deceleration zone approaching the minimum volume. They do not observe the acceleration area right after the impact as well, but that could be due to accuracy limitation of the instrument. The data in Table 21 is shown in Fig. 73. A curve-fit to the current data is included, and it can be expressed as $242.6 P_r^{-0.4881}$.

Shyue [126] and Bourne and Field [16] compared the shock velocity to the jet velocity. Even though their results differ quantitatively, they are qualitatively similar. Their plots explain that below a certain pressure ratio the shock travels faster than the jet. Even at pressure ratios slightly below that point the jet impacts the distal side before the shock reaches the zone of collision, because the shock has to go around

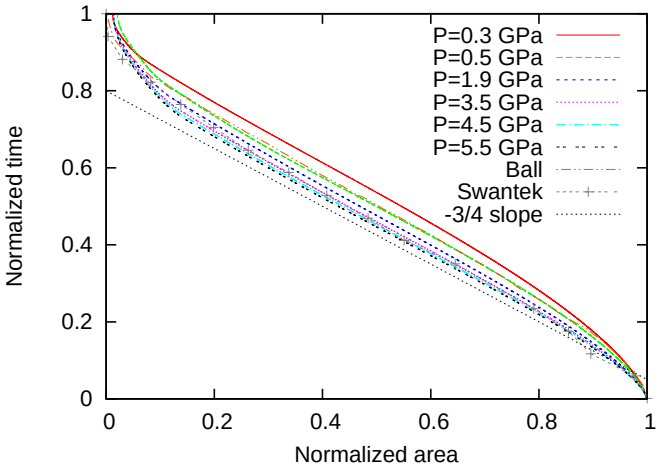


Figure 72: Time since impact (normalized by the time of minimum area) as a function of the area of the cylindrical bubble (normalized by the initial area). Compared with the numerical study by Ball et al. [7] and the experimental study with one bubble by Swantek and Austin [130].

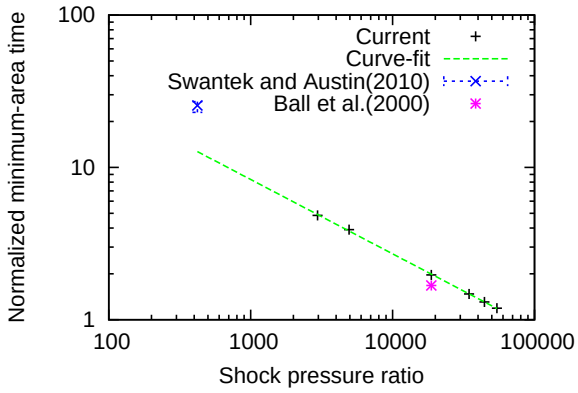
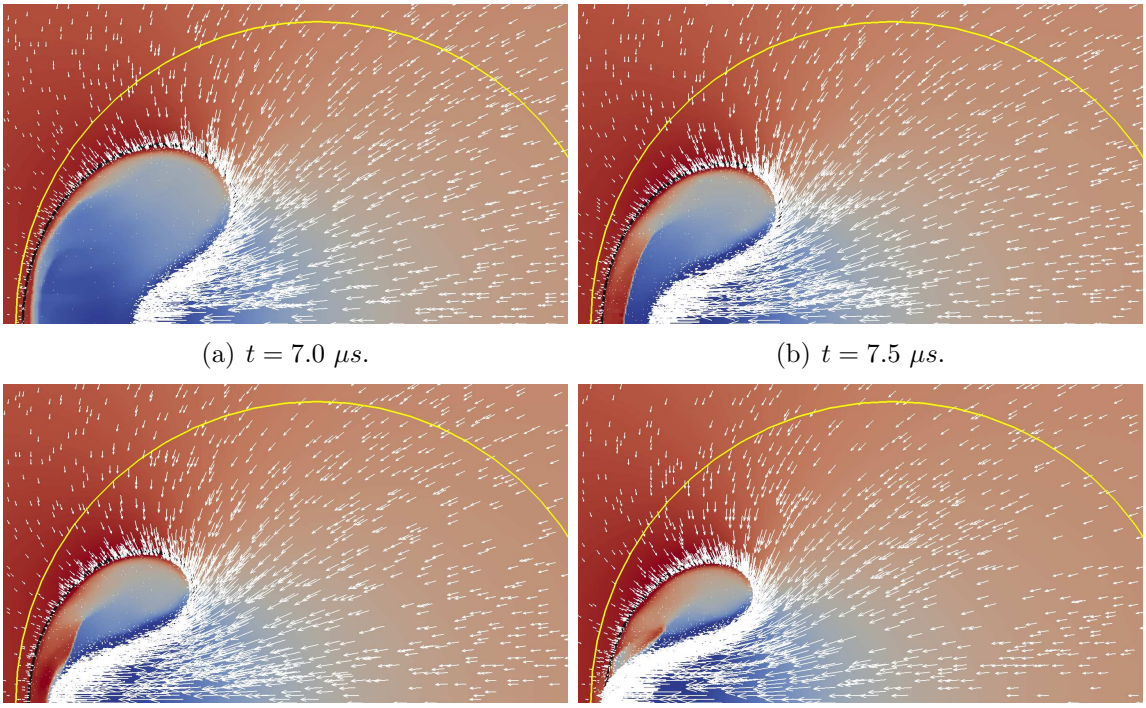


Figure 73: Normalized minimum-area time, $t_{A\,\min} c_L / R_0$, for a cylindrical air bubble in water. The curve-fit is based on the current study and it is defined as: $242.6 P_r^{-0.4881}$.



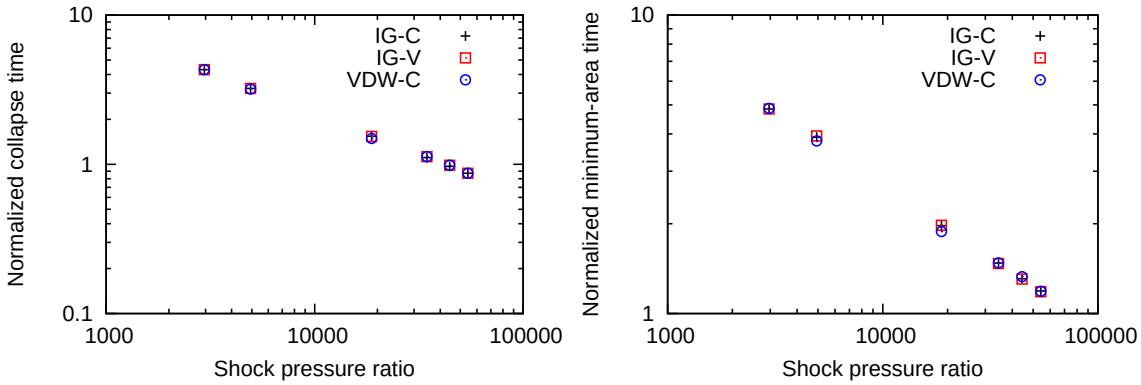
(a) $t = 7.0\ \mu s$. (b) $t = 7.5\ \mu s$.

(c) $t = 7.7\ \mu s$. (d) $t = 7.9\ \mu s$.

Figure 74: Detailed evolution of Case 1 near the jet impact. Contour colored based on the X-velocity. Original shape of the bubble in yellow. Instantaneous bubble interface in black.

the bubble. At low-enough pressure ratios the shock embeds into the bubble before the impact of the jet. Case 1, with $P = 0.3\ GPa$ and $P_r = 2961$, is one of these cases. When the shock passes the whole bubble, the downstream side of the interface is subjected to a large pressure ratio, which then generates an internal shock that propagates upstream that later impact the coming jet. At the same time, the downstream side starts moving upstream. This is observed in weak regimes ($P_r < 300$) where symmetric collapse occurs. Figure 74 shows the evolution of case 1 in detail, which presents the contour plot of the velocity in the X direction, the original location of the bubble in yellow, the velocity vectors in white, and the material interface in black.

Ball et al. [7] mentioned that the equation of state might produce some appreciable difference. In order to assess the influence of the thermodynamic closure, particularly



(a) Collapse time.

(b) Minimum area.

Figure 75: Comparison between different thermodynamic closure.

at very high pressures, we replace for the air inside the bubble the IG EOS with constant c_P^0 (IG-C) for the IG EOS with variable c_P^0 (IG-V) and the VDW EOS with constant c_P^0 (VDW-C). The coefficients for VDW are based on a mix in volume of 79% Nitrogen ($P_c = 3.3978\ MPa$, $T_c = 126.19\ K$) and 21% Oxygen ($P_c = 5.05\ MPa$, $T_c = 154.6\ K$), assuming that $a = \sum_i \sum_j X_i X_j (a_i + a_j)/2$ and $b = \sum_i X_i b_i$, where X is the molar fraction. The results are shown in Fig. 75. The differences between the models are negligible, suggesting that the equation of state used may not be the cause of mismatch between experiments and simulations.

8.4 Discussion

The Local Discontinuous Galerkin (LDG) method [24] together with special treatment for the Hamilton-Jacobi equation in [151] was shown to successfully solve inviscid multi-phase flows. In addition, the current thermodynamic closure perfectly handled two phases with different EOSs for constant and varying specific heats, including real gas models.

Troubled-cells treated with a moment limiter (ML) [72] modified for non-uniform meshes with hanging nodes. The limiting stage is done using primitive and conservative variables. For flows with more than one phase the first set of variables resulted to

be significantly more effective than the second one. However, when only one phase is present the benefit is reduced. An additional limiter to avoid non-physical values of two-phase models was adapted from [159] where it was used for the Euler equations. In the current study, we extended the approach to the 5-eq. model. This approach results in smaller errors than cropping the high-order modes.

The solver was used to study the interaction of an air cylindrical bubble in water with a shock for high pressures and different thermodynamic closures. The collapse time, mean jet speed, and volume evolution were measured. The results matched very well with numerical results in the literature, and agreed well with published experimental studies. We also extended the regime usually studied in the literature and the same trend was observed. Given the capabilities of the solver, high detail of the dynamics of these complex flows was captured. A significant quantitative difference is observed between numerical and experimental studies in the literature. Ball et al. [7] suggested that a non-negligible error could be caused by the thermodynamic closure of the gas phase. In order to address this issue, we used 3 different thermodynamic closures for the bubble: ideal gas equation of state (IG EOS) with constant specific heat, IG EOS with variable specific heat, and van der Waals (VDW) EOS with constant specific heat. However, all three options produces almost identical collapse time and time to minimum area. The fact that constant specific heat and variable specific heat for the same EOS produced almost identical times suggests that vibrational relaxation should not affect appreciably the results.

CHAPTER IX

DROP-WALL INTERACTION USING LDG

9.1 Introduction

Surface tension is a phenomena that controls several multi-phase problems. Even though its effects can easily be modeled and formulated, their implementation is more difficult. Numerical schemes usually present some degree of difficulty capturing discontinuities. Given that surface tension is manifested in this region, it is expected to observe additional challenges. Thus, in this chapter the time-step size required to preserve positivity Δt_{c2} is used instead of Δt_{c1}, which is less restrictive and is needed only to provide stability.

The technique derived in this study for the LDG method is evaluated under steady and unsteady conditions. Most numerical method present spurious oscillations at the interface. The oscillations observed here are compared with the literature. Finally, the solver is applied to drop-wall collisions.

9.2 Verification and Validation of surface tension

The details of the test cases and the rationals for them are summarized in Table 22.

Table 22: Summary of cases to verify and validate the formulation and implementation of surface tension.

Test case	Purpose	Section
Capillary effects	Measure the spurious oscillations and the error for a stationary case.	9.2.2
Oscillating drop	Measure the error for an unsteady problem.	9.2.3
Falling drop	Validation of the contact line.	9.2.4

9.2.1 Accuracy

The computation of the curvature has to be evaluated. For this purpose, we use a static field with a volume fraction that varies as $\alpha_1 = \sqrt{(x_1+1)^2+(x_2+1)^2}$. The exact solution for the curvature is $\kappa = -1/\alpha_1$. The L_1-norm error is measured in the domain $[0,1]^2$ with $p=2$ and $p=4$ elements. The result is shown in Fig. 76. The

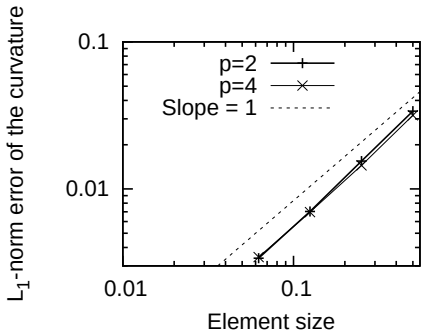


Figure 76: Order of accuracy of the computation of the curvature.

order of accuracy observed is approximately 1 in spite of the order of the elements. This is expected because the solution of the curvature in the current formulation is constant inside each element.

9.2.2 Capillary effects

In equilibrium the pressure jump at an interface is given by $\Delta P = \sigma(\kappa_1 + \kappa_2)$, where κ_1 and κ_2 are the main local interface curvatures. Under no external body force and stagnant conditions a non-spherical (or non cylindrical in 2D) shape evolves into a sphere (or cylinder in 2D). Once equilibrium is obtained the pressure difference between inside and outside the drop is $\Delta P = 2\sigma/r$ for a sphere (3D) and $\Delta P = \sigma/r$ for a cylinder (2D). We test the solver with similar initial conditions as in [88]. Both fluids have the same properties: $\gamma_i = 1.4$, $P_{\infty,i} = 0.0\ Pa$, $c_{v_i} = 720.0\ J/(kg\ K)$, $\mu_i = 1.0\ Pa\ s$, $\kappa_i = 0.1\ W/(m\ K)$, and $\sigma = 100.0\ N/m$. Initially, $\rho_1 = 1.0\ kg/m^3$, $\rho_2 = 1.0\ kg/m^3$, and $P = 101325.0\ Pa$. The initial shape is a square of length $\sqrt{\pi}/4\ m$ located at the center of a unit square. All the boundaries are at farfield.

After long enough time the square evolves into a circle and the velocities should decay and converge to zero. However, almost every method reported in the literature presents some spurious oscillations at the interface. These seem to depend on the method and on the interaction between surface tension and viscosity. The Laplace number ($La = \rho_{\text{circle}} D\sigma/\mu^2_{\text{circle}}$, where D is the diameter of the circle) relates these two effects. For this problem $La = 67.6$, and the viscosity and density ratios are equal to 1. The fluctuations are measured in a non-dimensional way through the Capillary number $Ca = \mu_{\text{circle}} V/\sigma$, where V is the maximum velocity in the whole domain after a long time. Table 23 shows Ca for different grids. The values observed in the

Table 23: Spurious oscillations for different grids.

Number of elements	p	Ca
16×16	2	2.1×10^{-3}
32×32	2	2.2×10^{-3}
16×16	4	3.0×10^{-3}
32×32	4	5.0×10^{-3}

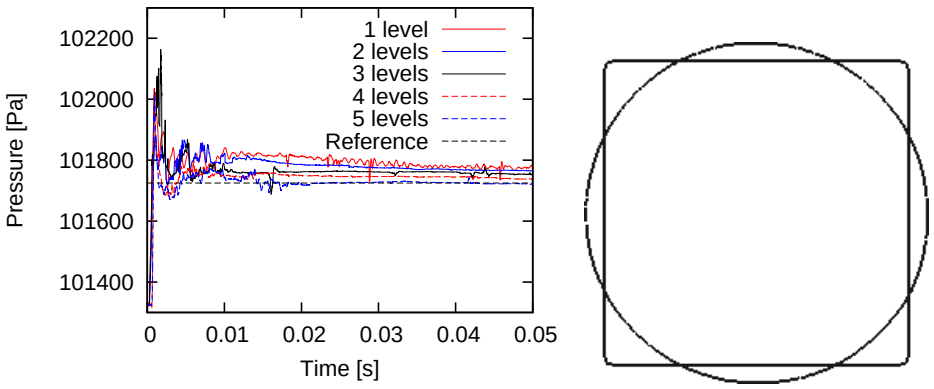
current study are comparable to the CSS and CSF methods reported in [137], where they used a finite difference scheme with 32 grid points per diameter. For $La = 0.357$, their capillary numbers were 3×10^{-3} and 1.2×10^{-2} for CSS and CSF, respectively. For $La = 2^6$, their capillary number were 3×10^{-4} and 5.0×10^{-4} for CSS and CSF respectively. A larger domain was used to assess if the boundary conditions were affecting the solution but the same values were observed.

Marchandise et al. [88] observed a first order convergence for Ca with respect to h, where h is the element size. In their case the level set function was used and its solution was independent on the grid size. However, in our case it seems that increasing the resolution does not help decrease the oscillations. This can be explained by the fact that the gradient of the volume fraction at the interface is proportional to h^{-1}. Thus, increasing the resolution makes the source term larger and the problem

numerically more difficult. An alternative could be to treat the surface tension in a fully conservative way as in [100], and this will be conducted in the near future.

Given that the current formulation is compressible and conservative, the volume of the drop changes, but the mass does not. Due to surface tension effects the drop is compressed. If we neglect the change in density, the pressure jump would be 400.0 Pa. However, if we assume that the initial and final temperature are the same, the pressure jump would be 400.79 Pa. We use the later assumption as a reference value to compute the error. Nonetheless, this error estimating of the exact solution is smaller than the other errors (see below).

A grid of 16×16 with $p = 2$ elements is used with $l_{max} = 1, 2, 3, 4, 5$. Figure 77(a) shows the pressure at the center of the domain in time. As the resolution increases the pressure converges to the expected value. Figure 77(b) presents the initial and the final ($t = 0.05$ s) location of the interface, defined as the contour line where the volume fraction is equal to 0.5 s.

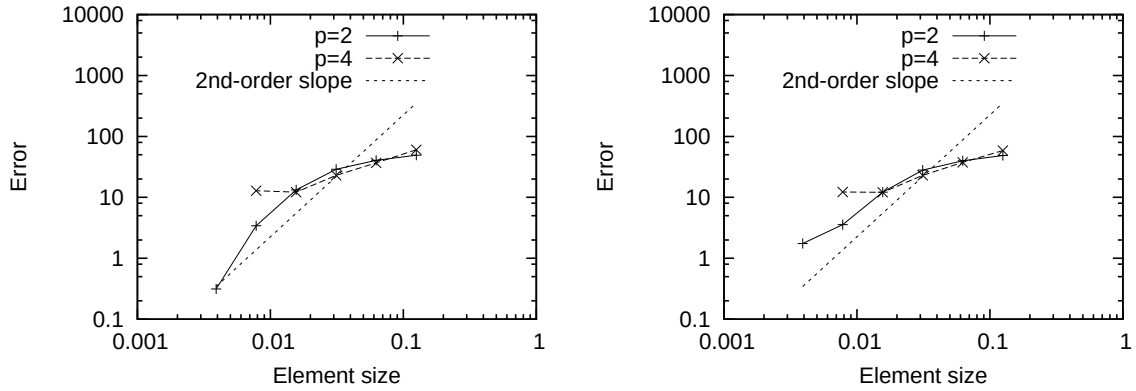


(a) Pressure evolution at the center of the domain. (b) Initial and final shapes.

Figure 77: Square evolving into a cylinder due to surface tension effects.

As the grid is refined, due to the presence of spurious oscillations it is not possible to make the error converge to very low values. The convergence is analyzed for different spatial orders: $p = 2$ and $p = 4$. The error is quantified based on the pressure at the center of the domain averaged in time. The result for an average

from $t = 0.045\ s$ until $t = 0.05\ s$ is shown in Fig. 78(a), while for an average from $t = 0.0475\ s$ until $t = 0.05\ s$ is shown in Fig. 78(b). In this context, the effect of the



(a) Average from $t = 0.045$ to $t = 0.05$.

(b) Average from $t = 0.0475$ to $t = 0.05$.

Figure 78: Order of accuracy for a square evolving into a cylinder due to surface tension effects.

spurious oscillations is twofold: it makes the error vary in time and prevents the error from decreasing to machine precision. Nonetheless, given that the order is cropped when computing the curvature, the observation in Fig. 78 are consistent in the sense that the order is somewhat independent of the order of the element. Note that as seen in Table 23, the error for $p = 4$ does not decrease as expected. This could be due to deficiencies in the limiter under very challenging conditions.

9.2.3 Oscillating drop

Now the dynamic behavior of the surface tension is studied. As mentioned above, drops tend to a spherical or cylindrical shape due to surface tension effects. If these effects are greater than the dissipation effects, the drop oscillates as it converges to equilibrium. With a linearized inviscid analysis the natural modes can be derived [39]:

$$\omega_n^2 = (n^3 - n)\frac{\sigma}{(\rho_1 + \rho_2)r^3} \tag{150}$$

Table 24: Period for an oscillating drop using different grids. Theoretical value $\tau = 0.3628\ s$

Number of elements	$\tau\ [s]$	Error $[s]$
16^2	0.3925	0.0298
32^2	0.4120	0.0412

where n is the mode of interest, r is the unperturbed radius and the period is $\tau = 2\pi/\omega_n$. r is computed as the radius of the equivalent shape in equilibrium which has the same area or volume. The drop is initialized with an elliptical shape:

$$\left(\frac{x_1}{0.2}\right)^2 + \left(\frac{x_2}{0.12}\right)^2 = 1$$

The fluid properties are: $P_{\infty,1} = 0$, $\gamma_1 = 1.4$, $\mu_1 = 0$, $\lambda_1 = 0$, $P_{\infty,2} = 0$, $\gamma_2 = 1.4$, $\mu_2 = 0$, $\lambda_2 = 0$, and $\sigma = 342.0$. Initially, $P = 101325.0$, $\rho_1 = 1.0$, and $\rho_2 = 1.0$. The mode of interest is $n = 2$. Thus, $r = 0.15492$.

The axes in the Cartesian directions are measured during the simulations, assuming that the interface is where the volume fraction is 0.5. Then, the time between two instants when both axes are equal is half period. The results in Table 24 were computed for different grids all with $l_{\max} = 3$. In the study of the spurious oscillations we observed that they increased slightly when the grid was refined. Probably, this same spurious oscillations are the cause of a larger error when the grid is refined.

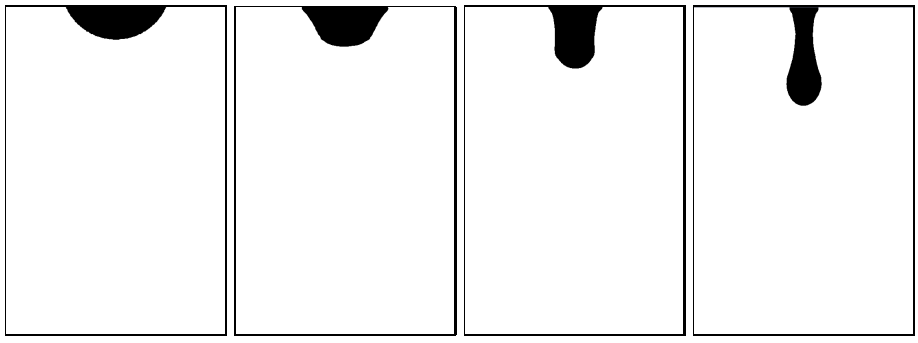
9.2.4 Falling drop

A case commonly found in the literature to study the effect of wall effect and break up consist of a drop initially in contact with the roof and that start falling due to the effect of gravity.

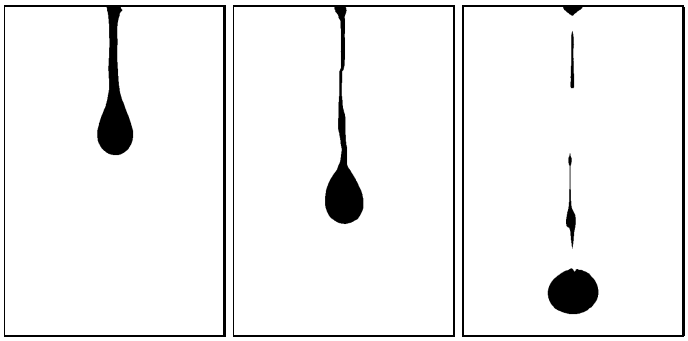
For numerical purposes, non-physical values are used here. However, the non-dimensional parameters that governed this problem have physical meaning, and these are the density ratio ρ_l/ρ_g and the Bond number, $Bo = \rho_l g D^2/\sigma$. The domain has a width of 1.0 and a hight of 1.5. Initially, the center of the drop is at $[0.5; 1.6]$ and

it has a radius of 0.25. The gravity is -25.0, the surface tension is 175, and the equilibrium contact angle is $\vartheta_{eq} = 25^o$. This same case was shown in the literature in several cases (e.g., [100]).

The grid has 48×72 elements with 3 levels. The results are shown in Fig. 79. Rapidly, the surface of the drop adapts due to the contact angle. Given the effect of the gravity the drop starts falling. Once the ligament is long and thin enough, it break up. Drops with lower Bond number, tend to remain hanging from the ceiling. This is the case of the small drop in Fig. 79(g), where the effect of surface tension is greater than the weight.



(a) $t = 0$ s. (b) $t = 0.075$ s. (c) $t = 0.15$ s. (d) $t = 0.225$ s.



(e) $t = 0.3$ s. (f) $t = 0.375$ s. (g) $t = 0.45$ s.

Figure 79: Drop falling from the ceiling due to gravity.

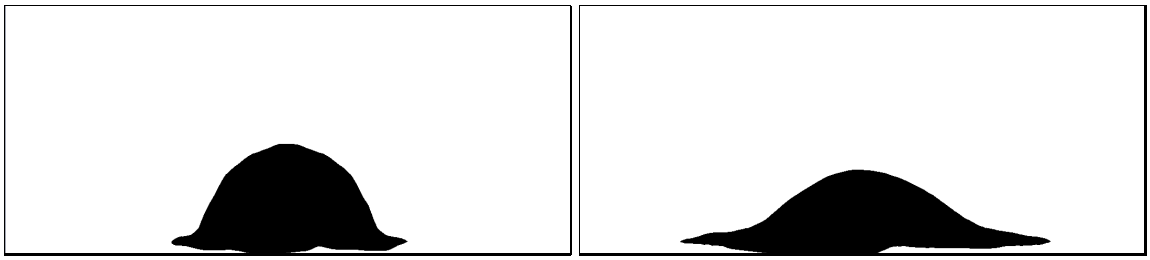
9.3 Drop-wall collision

Xu et al. [147] found that splashing produced after a drop impacts a dry wall can be avoided by decreasing the pressure of the surround gas. That means that the

properties of the surrounding gas play a critical roll in the behavior of the drop. Also, the same authors concluded based on their observations that the compressibility of the gas is a key element needed to quantify the destabilization of the advancing front and produce the splashing. The experiment used in [147] is reproduced numerically to study this phenomenon. A drop of alcohol with an initial diameter of $D_0 = 3.4\ mm$ moves perpendicularly towards a wall at a speed of $V_0 = 3.74\ m/s$. Initially, we locate the center of the drop at a distance of 1.5 radii. For the surrounding gas we use $\gamma = 1.4$, $P_\infty = 0\ Pa$, $c_v = 720.0\ J/(kg\ K)$, and $\mu = 1.8 \times 10^{-5}\ Pa\ s$. For the liquid we use $\gamma = 4.1$, $P_\infty = 5 \times 10^7\ Pa$, $c_v = 69\ J/(kg\ K)$, and $\mu = 1.96\ \times 10^{-3}\ Pa\ s$. The corresponding threshold pressure is 38.4 kPa. First, the test is conducted in 2D with a domain of 0.0136 $m \times 0.00595\ m$, 32×14 elements with $p = 2$, and 3 AMR levels. Grid dependency is not reached in this problem, but the largest resolution that can be run in a convenient amount of time is used. Two different initial pressures are used: 17.2 kPa and 100 kPa. The first one is below the threshold pressure and the drop should stick to the wall, while the second one is above the the threshold pressure and the drop should produce splashing. Even in 2D, it can be observed that the drop is not capable of fully displacing the gas at high pressure, see Fig. 80. Note that the threshold pressure should not necessarily be the same as in 3D, since, according to [147], the threshold pressure depends on the velocity of propagation of the front, which depends on the geometry of the drop.

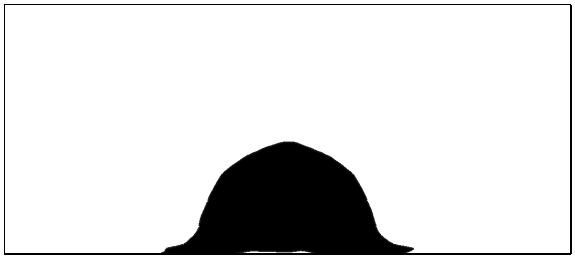
The case at $P = 17.2\ kPa$ is repeated in 3D. Figure 81 shows the result together the images from reference [147]. The current numerical results qualitatively resemble the pictures in reference [147].

The radius (R) and velocity (dR/dt) of the front of the lamella are measured from the 3-D simulation, see Fig. 82. During the initial stage, theoretical estimations predict that the velocity evolves proportional to $t^{-1/2}$ [131,]. The curve fit of the current data is $0.66674(t\ V_0/D_0)^{-0.46277}$.

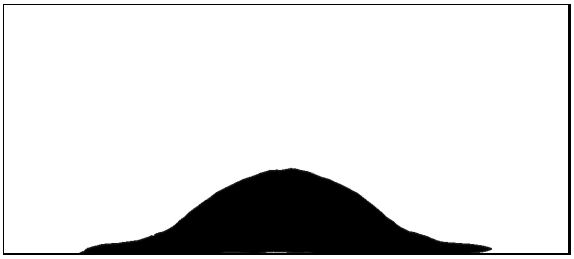


(a) $t = 0.276\ ms$, $P = 100\ kPa$.

(b) $t = 0.552\ ms$, $P = 100\ kPa$.

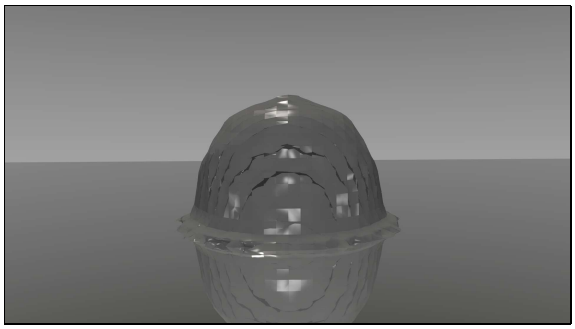


(c) $t = 0.276\ ms$, $P = 17.2\ kPa$.

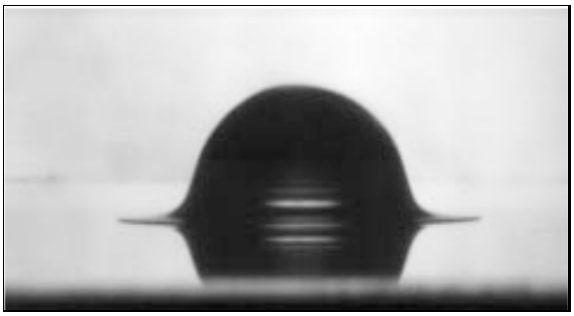


(d) $t = 0.552\ ms$, $P = 17.2\ kPa$.

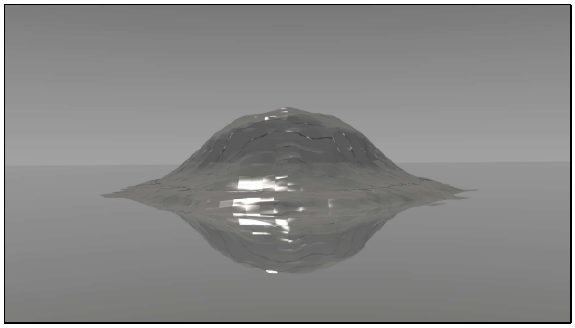
Figure 80: Collision of a drop against a wall in 2D.



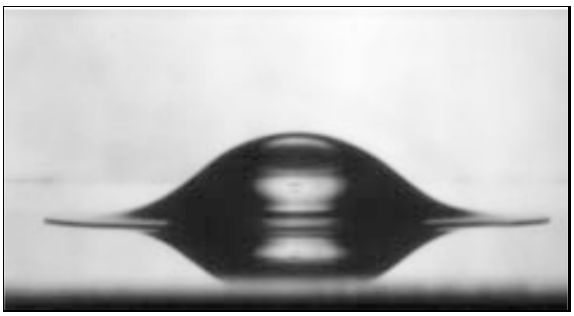
(a) Current simulation at $t = 0.276\ ms$.



(b) Experimental picture in [147] at $t = 0.276\ ms$.



(c) Current simulation at $t = 0.552\ ms$.



(d) Experimental picture in [147] at $t = 0.552\ ms$.

Figure 81: Collision of a spherical drop against a wall at $P = 17.2\ kPa$.

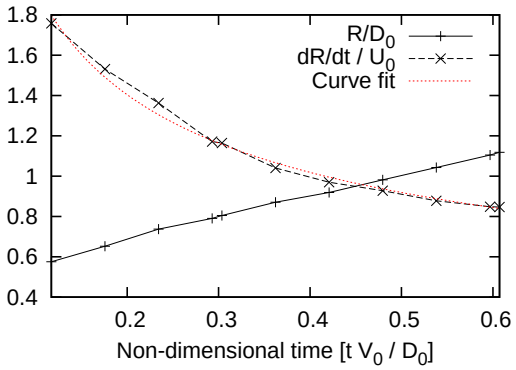


Figure 82: Radius and velocity of the front of the lamella for $P = 17.2\ kPa$.

9.4 Discussion

Using the Local Discontinuous Galerkin (LDG) method, we presented a multi-fluid solver that is capable of resolving surface tension effects, including the contact angle formed between the interface and a wall.

The multi-fluid model used is the 5-eq. model, which conserves mass, momentum, and total energy, and assumes compressible fluids. Even though this final aspect together with explicit time integration makes the time-step size very small, it allows for regimes where compressibility is important, like in some drop-wall collision cases [80, 34]. The time step is constrained further by the need to preserve positivity. Nonetheless, it allows for usage of schemes with minimal numerical diffusion. Future work could involve an algorithm to control the time-step size between the requirements for stability and positivity.

Modeling surface tension effects has numerical difficulties for most schemes, and it usually generates spurious oscillations which may even be destructive. The fluctuations found in our new approach have comparable magnitudes to other techniques commonly used. The technique suggested here to compute the curvature is simple, effective and stable, but it has an order of accuracy of 1. It was observed that elements of 4^{th}-order may present larger errors than elements of 2^{nd}-order. The cause of this

could be the effects at the interface that challenge the numerical scheme. The surface tension effect was applied as a source term, as it is commonly executed in most cases, but it could be implemented in a conservative way as in [100], which could potentially reduce spurious fluctuations.

The solver was applied to the interaction between a drop and a wall. The experimental observations by [147] showed that the ambient pressure can affect the capability of the drop to displace the air after impact. A similar effect was observed in 2-D simulations. The 3-D simulation of a case in [147] suggests an evolution of the velocity of the front of the lamella proportional to $t^{-0.46277}$, close to the theoretical estimation of $t^{-1/2}$. As mentioned by [147], compressibility of the fluids is a main element to study this problem effectively and our framework provided this capability.

CHAPTER X

TOWARD APPLIED FLOWS WITH THE LDG METHOD

10.1 Introduction

Several aspects relevant to applied flows and not evaluated so far are tested and discussed in this chapter.

10.2 Code performance

10.2.1 Scaling

Weak (constant load) scalability was measured on five clusters: 2 Linux+Intel, 1 Cray XT4, 1 Cray XT5, and 1 Cray XE6. Each core has 1024 elements with $p = 2$. The geometry is Cartesian, has 3 dimensions, and wall boundaries on all 6 faces. No refinement is used and the simulation is run for 100 time-steps. The speed-up and efficiency are shown in Fig. 83. Speed-up is computed as:

$$\text{speed-up}(n_p) = \frac{t_1}{t_{n_p}} \tag{151}$$

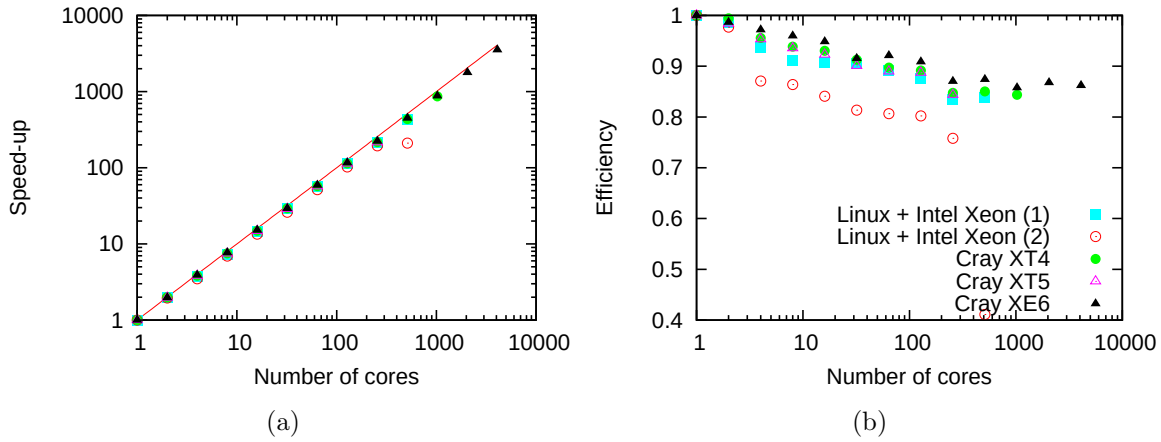
where t_i is the time taken using i processors. The efficiency is computed as:

$$\text{efficiency}(n_p) = \frac{t_1}{n_p * t_{n_p}} \tag{152}$$

All platforms showed very good scaling except for one of the Linux+Intel clusters above 512 cores. The most modern computer tested (Cray XE6) showed the best performance. The maximum number of cores used was 4096 and their efficiency was approximately 86%.

10.2.2 Convection of a vortex at very low Mach number in 2-D

The appropriate way to compare the efficiency of two schemes is using a solution for which you can compute the error accurately. This allows you to compare absolute



(a) (b)

Figure 83: Scaling at constant load (1024 elements per processor) for a cube. The straight solid line shows the ideal speed-up.

accuracy with CPU time, and avoids using the number of nodes in the comparison since it is not a direct measure of the efficiency.

A vortex is initialized in the center of the domain according to:

$$\begin{cases} v_1 = V_0 + \frac{\partial \psi}{\partial y} & v_2 = -\frac{\partial \psi}{\partial x} \\ \psi = \Gamma \exp\left(-\frac{(x-x_c)^2+(y-y_c)^2}{2R_c^2}\right) \\ \rho = \rho_0 \\ P = P_0 + \frac{\rho \Gamma^2}{2R_c^2} \exp\left(-\frac{(x-x_c)^2+(y-y_c)^2}{R_c^2}\right) \end{cases}$$

where $V_0 = 35\ m/s$, $\Gamma = 3.59157 \cdot 10^{-2}$, $R_c = 0.01556\ m$, $\rho_0 = 1.17170407\ kg/m^3$ and $p_0 = 101325\ Pa$. The domain is a 2-D square with a length of $0.3112\ m$ and periodic boundaries.

The L^1-norm error for the X-momentum after 5 periods is compared for different orders and number of cells using Leslie (FV code) and DiGGIT (DG code). The initial value is used as the reference solution after 5 periods. In addition, the DG code was run with and without the Cartesian assumption. See Fig. 84. All the combinations were run using 4 cores on an Linux - Intel Xeon cluster (Garuda). The O(2) and O(4) curves correspond to Leslie with the 2^{nd} and 4^{nd} order of accuracy respectively. The other curves were obtained with DiGGIT for $p = 2, 4, 6$. Curves closer to the bottom

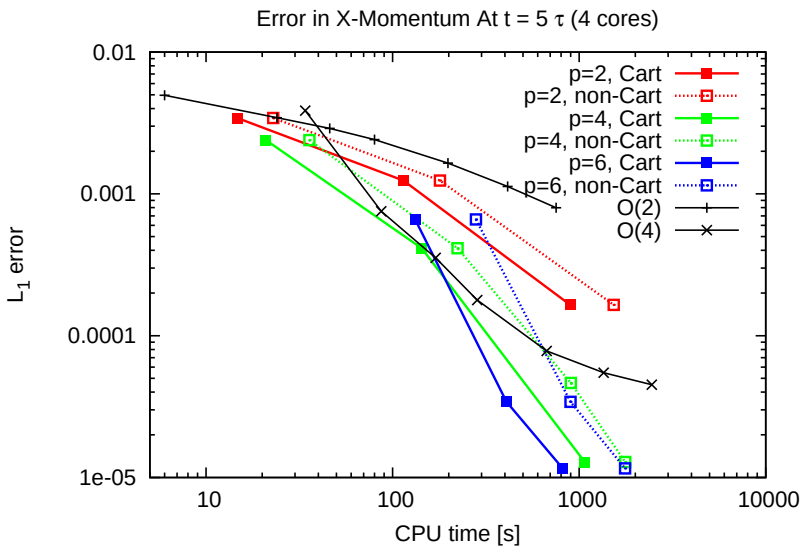


Figure 84: Efficiency for the convection of a vortex at low Mach number in 2-D.

left corner indicate a more efficient scheme. The efficiency increases for higher orders for both codes, except in the high-error region. Nonetheless, the error is very large in this area and the solution is useless, so this high-error region can be ignored. The difference between with and without the Cartesian assumption for DiGGIT produces a change in the speed with a ratio of 1.5 for $p = 2$, and increases with the order up to 2.1 for $p = 6$.

10.2.3 Convection of a vortex at very low Mach number in 3-D

The same test case as in previous section is studied here but in 3-D. The domain is now a cube and the solution is uniform and periodic in the new dimension. Leslie is run with the O(2) scheme and a grid of 80^3 cells, while for DiGGIT the order and the number of cells are varied. The non-Cartesian cases are omitted here for the sake of brevity. The results at $t = \tau$ are shown in Fig. 85. Even though only one point is shown for Leslie, it is enough to give us an idea of the situation. The efficiency increases slightly with the order. At equal error, Leslie was 7.5 times faster. However, the results are at a different time than in the 2-D case. It could be that as

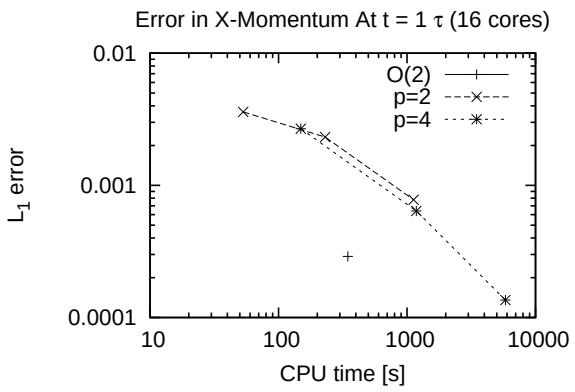


Figure 85: Efficiency for the convection of a vortex at low Mach number in 3-D.

time progresses the error for Leslie increases faster than for DiGGIT.

Besides the mismatch between the 2-D and the 3-D cases, the difference in performance between 2-D and 3-D for DG should be larger than for FV. Assuming a cube, FV scales with $n_x^{n_d}$ where n_d is the number of dimensions, while DG (assuming $2p+2$ quadrature points in each direction) scales with $[(2p+2)n_x]^{n_d}$. A detailed comparison of the codes would imply to run several different cases, which is not the purpose of this thesis so it is omitted.

10.2.4 Timing

The computational time spent in each procedure depends on the problem being simulated. In order to get an estimate of computational load of each section of the code, the profiling is done in serial for the case studied in Section 10.4 with LDKM. The result is shown in Table 25 for the main procedures assuming constant Jacobians (i.e. Cartesian grid) with hexahedral elements. The inter-element communication has a significant load because the procedure does not make any assumption about how the elements are being coupled. Some assumptions that could be done to reduce this cost drastically are: element faces always matching spatially and constant order.

All the terms (inviscid, viscous, and source) are integrated with the same number of quadrature points. Viscous terms usually require half as much in each direction

Table 25: Timing of the main procedures.

Procedure	Computational load [%]
Inter-element communication	20.01
Source term (due to LDKM)	15.62
Derivatives	11.60
Explicit filtering and model parameters for LDKM	7.69
Time integration	5.75
Inviscid flux	4.75
Inviscid numerical flux	4.75
Viscous flux	4.09
Viscous numerical flux	4.72
Spatial integration	2.56
Positivity check	1.64

than the inviscid terms. Therefore, using different rules for viscous term would reduce the their computational cost approximately by 8 times.

10.3 Verification of the boundary conditions

The details of the test cases and rationals for them are summarized in Table 26.

Table 26: Summary of cases to verify the boundary conditions.

Test case	Purpose	Section
Flow past a cylinder	Verification for high-order curved wall.	10.3.1
Two-phase farfield	Verify fluctuation generated when the interface crosses the outflow.	10.3.2
Two-phase in 2-D	Same as previous case but with a curved interface.	10.3.3
1-D acoustic waves	Verify the reflectivity of artificial BCs.	10.3.4
2-D acoustic wave	Same as previous case but in multi-dimensions.	10.3.5
Vortex convection	Quantify the behavior of the absorbing sponge layer.	10.3.6

10.3.1 Subsonic inviscid flow past a cylinder

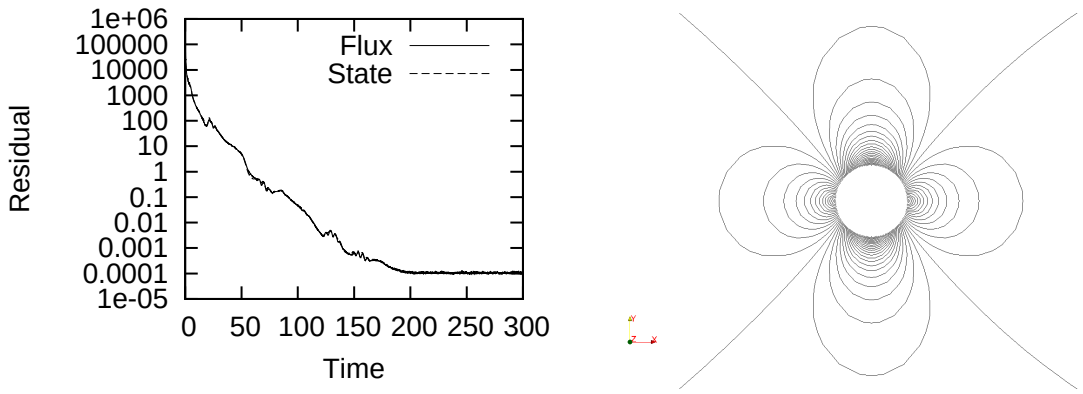
Elements that are on a curved boundary must accurately represent the shape of the boundary. Krivodonova and Berger [73] showed that the nodes on a face of a high-order element do not represent the curved boundary the solution has large errors in spite of the number of elements. We conduct the same inviscid flow past a cylinder case as in [73] to show that curved boundaries are correctly represented. In addition, the flux based and the state based formulations are used to show that both produce almost identical results, at least for this problem.

The grid consists of a disc with an inner radius of 0.5 and an outer radius of 10. 8 elements are located in the radial direction, and 16 in the circumferential direction. The grid stretches out as the radius increases as in [73]. The order of the elements is $p = 4$. The boundary conditions are inviscid wall and farfield for the inner and outer boundaries, respectively. The flow has a Mach number of 0.38. The time integration is done with 3^{rd}-order TVD-RK. Given that the flow is smooth, no limiter is used. The numerical flux used is the HLLC.

Figure 86(a) presents the residual for both boundary formulations. Note that the difference between the two is barely noticeable. The residual is computed as the sum of difference in $\mathbf{d}_{i,j}$ (for every variable and every mode) at time-step n and $n+1$. Figure 86(b) shows the Mach lines once the solution converges. The lines are symmetric what according to [73] means a correct representation of the boundary. Otherwise, it would not be symmetric, viscous instabilities would be observed downstream, and the residual would not decay.

10.3.2 Two-phase farfield

The implementation of the boundary conditions for the two phase model is first tested for a one dimensional problem. The domain used is 2 mm long, has farfield boundary conditions on both ends, and 64 $p = 2$ elements. A constant velocity of

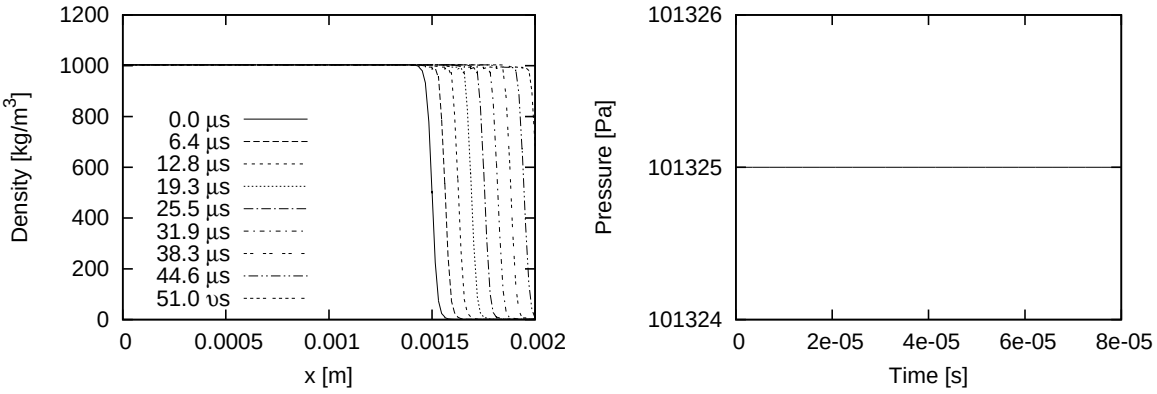


(a) Residual.

(b) Mach lines.

Figure 86: Subsonic inviscid flow past a cylinder.

10 m/s and pressure of 101325 Pa were initially imposed. The initial volume fraction is defined as $\alpha_1(x, t = 0) = 0.5 + 0.4999 \tanh[(x - 0.0015)/40000]$. This smooth initialization prevents numerical waves as a result of the initial value, isolating one possible cause of instabilities. The initial pressure in 101325 Pa and the temperature is 298 K. The material properties are: $c_{v,1} = 720.0\ J/(kg\ K)$, $\gamma_1 = 1.4$, $P_{\infty,1} = 0\ Pa$, $c_{v,1} = 474.81\ J/(kg\ K)$, $\gamma_1 = 4.1$, and $P_{\infty,1} = 4.4 \cdot 10^8\ Pa$. The transport properties for both phases are zero. The results are shown in Fig. 87. The interface successfully



(a) Density for different time steps.

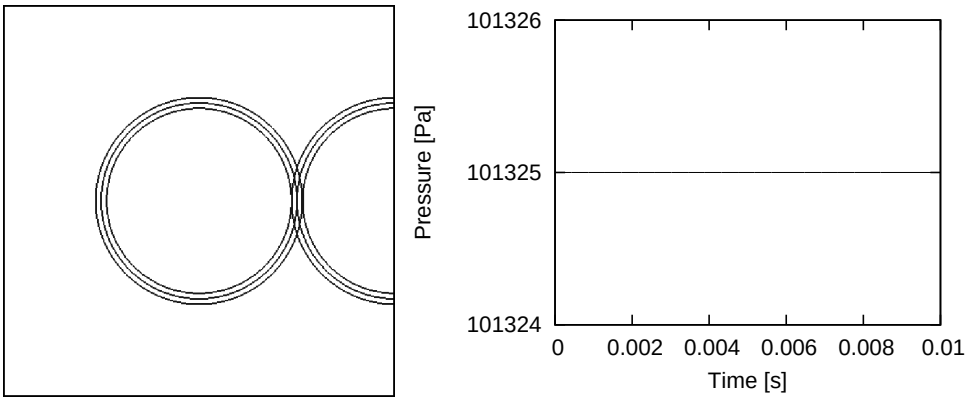
(b) Pressure at $x = 0.002\ m$.

Figure 87: Liquid-gas interface propagating in a one dimensional domain and exiting the domain.

exits the domain without causing any type of appreciable oscillation.

10.3.3 Two-phase convection in 2-D

A circle is convected in a 2-D squared domain with a length of 1.0 m. The circle has a radius of 0.25 m and it is initially located at the center of the geometry. The velocity is uniform in the x_1 direction and the magnitude is 100 m/s. The material properties are the same as in Section 10.3.2. The interface is shown at different time step in Fig. 88(a). The contour lines are for $\alpha = 0.25, 0.5, 0.75$. The pressure at $(1, 0.5)$ m is shown in Fig. 88(b). The circle leave the domain without generating any pressure



(a) Interface at $t = 0$ ms and $t = 5$ ms.

(b) Pressure at $(1, 0.5)$ m.

Figure 88: Liquid-gas interface propagating in a 2-D domain and exiting the domain.

instability and keeping the circular shape.

10.3.4 1-D acoustic wave

A one-dimensional acoustic wave propagating toward the left is used to test the reflectivity of the inflow and farfield boundaries. The domain is 2 mm long and has 64 elements with $p = 2$. The initial value can be expressed as:

$$\begin{aligned} v &= v_0 + A \cdot exp\left[-\left(B\frac{x - L/2}{L}\right)^2\right] \\ P &= P_0 - \rho_0 \cdot c_0 \left(v - v_0\right) \\ \rho &= \rho_0 - \frac{\rho_0 \left(v - v_0\right)}{c_0} \end{aligned} \tag{153}$$

Where the subindex 0 refers to the uniform value, $v_0 = 1\ m/s$, $P_0 = 101325\ Pa$, and A and B are constants that specify the amplitude and width of the wave. For the following cases $A = 7.5$ and $B = 12.0$.

The right hand side boundary is farfield, while the left hand side boundary is tested with inflow and farfield. Figure 89 shows the pressure (P) and velocity (v) at different time steps for inflow BC, whereas Fig. 90 shows the same but using farfield BC on both ends. In the first case, the acoustic wave reflects at the inflow boundary.

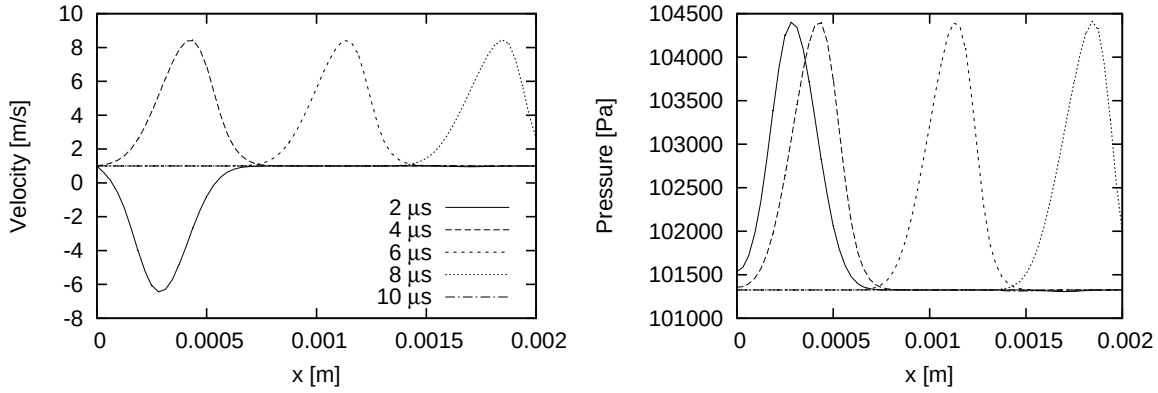


Figure 89: Reflectivity test of boundary conditions in 1D with inflow and farfield.

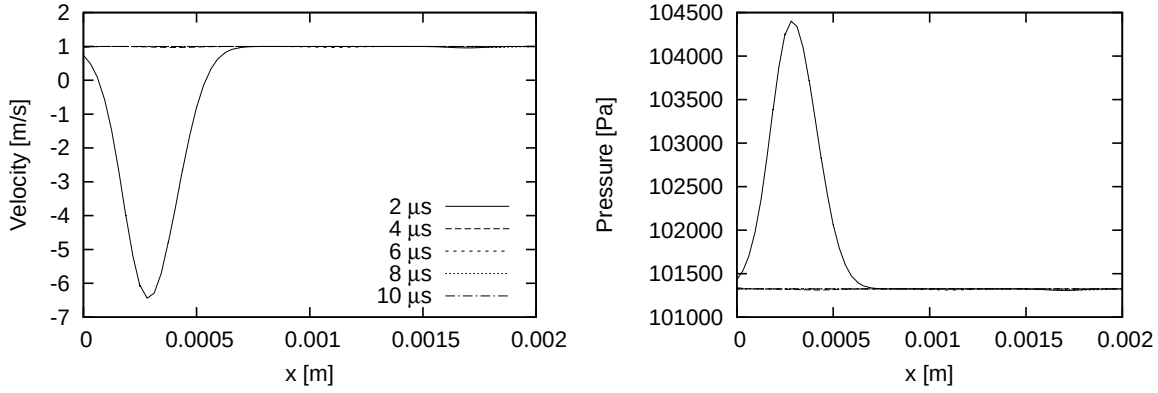


Figure 90: Reflectivity test of boundary conditions in 1D with farfield on both ends.

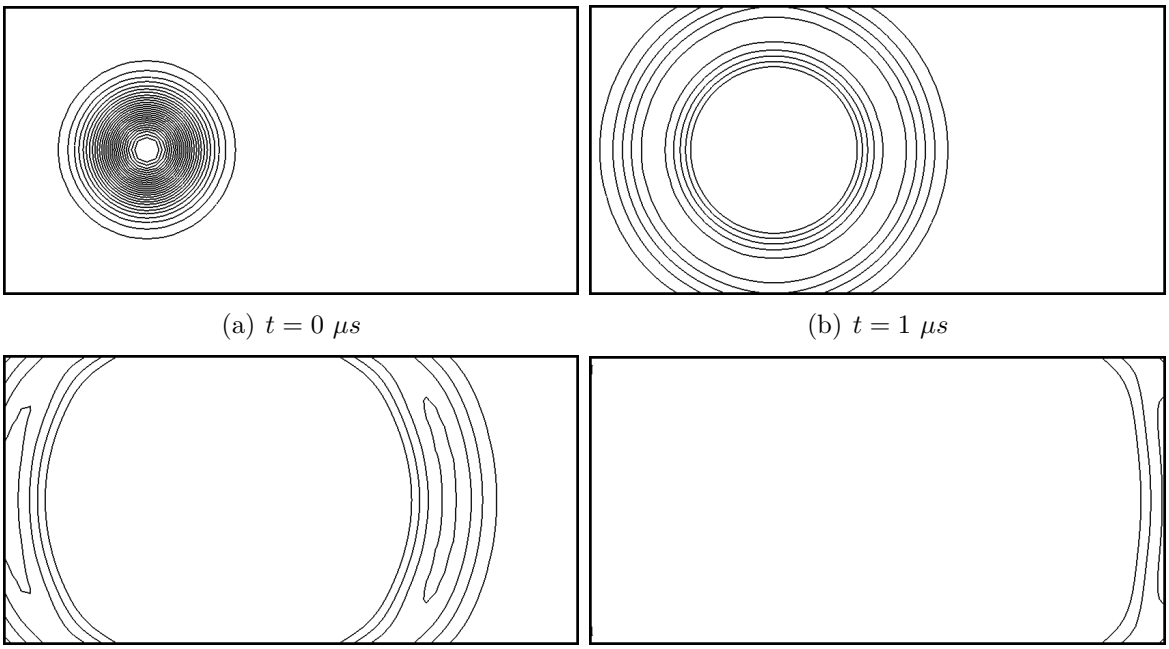
Meanwhile, in every case, the farfield boundaries offer very good absorption of the wave.

10.3.5 2-D acoustic wave

The non-reflection of the boundaries is tested with a two-dimensional acoustic wave initially positioned inside the domain. The domain size is $2 \times 1\ mm$ with 32×16 elements. All the boundaries are farfield. The initial velocity is $140\ m/s$ to the right. The initial pressure distribution is computed as:

$$P = P_{ref} + (P_{MAX} - P_{ref}) \exp\left(-12^2 \frac{(x - x_0)^2 + (y - y_0)^2}{l_x^2 + l_y^2}\right) \tag{154}$$

where x_0 and y_0 specify the location of the maximum pressure (P_{MAX}), $x_0 = 0.5\ mm$, $y_0 = 0.5\ mm$, and l_x and l_y are the size of the domain. The density is computed based on the isentropic condition: $\rho = \rho_{ref}(P/P_{ref})^{1/\gamma}$, where $\gamma = 1.4$. For the presented test $P_{ref} = 101325\ Pa$, $P_{MAX} - P_{ref} = 10132.5\ Pa$ and $\rho_{ref} = 1.1768\ K$. The resulted pressure contours are shown in Fig. 91 for different time steps. The iso-contours keep the circular shape relatively well, although not perfectly.



(a) $t = 0\ \mu s$ (b) $t = 1\ \mu s$

(c) $t = 2\ \mu s$ (d) $t = 3\ \mu s$

Figure 91: Pressure contours for an acoustic wave going through farfield BCs in 2D.

Sponge layers are added on 4 boundaries, each with a thickness of $0.2\ mm$. Two

values for the strength β_0 are used, 10^6 and 10^7. The density at $(1, 0.5)$ mm is shown in Fig. 92. The initial range of time is not shown in order to focus on the small bounced waves, which are much smaller than the original wave. For this problem,

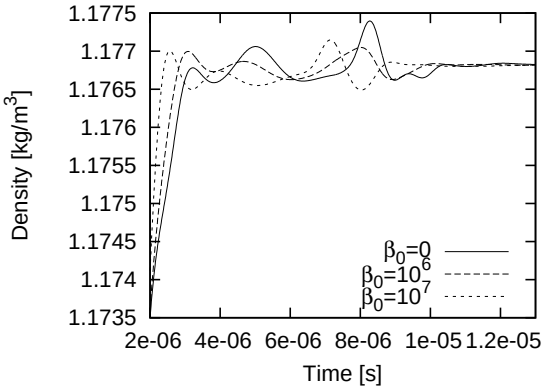
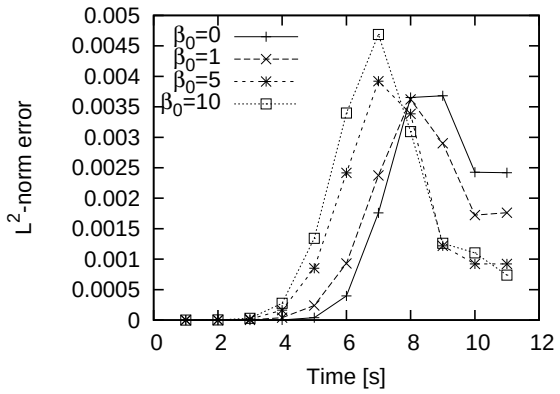


Figure 92: Density at $(1, 0.5)$ mm for an acoustic wave going through farfield BCs in 2D.

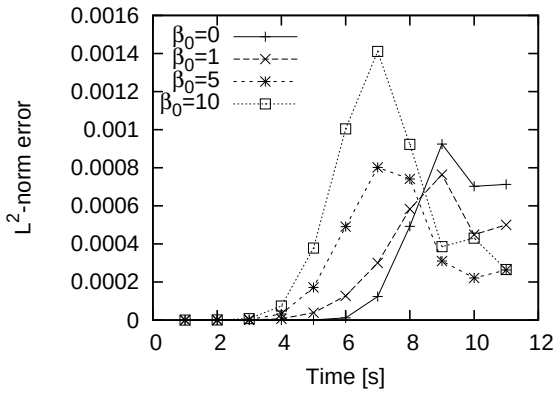
the presence of the sponge layers helps reduce the reflections only partially. Also, note that a higher strength seems to make the waves partially reflect on the sponge layer itself since the waves seem to arrive faster. Actually, the time difference is approximately twice the speed of sound divided by the thickness of the sponge layer.

10.3.6 Vortex convection with absorbing sponge layer

The same case as in Section 6.2.12 is simulated here but with farfield BCs on 4 boundaries instead of periodic conditions. Also, an absorbing sponge layer of 0.2 m-wide is used on all boundaries. This case is run for different sponge strengths (β_0) and the error at given times-steps is computed respect to a reference solution. This reference is obtained without the sponge layer and with a domain twice as long. The results are shown in Fig. 93. In Fig. 93(a) the error is computed over the whole domain (including the sponge layers), whereas in Fig. 93(b) the error is computed without taking into account the thickness of the sponge layers. Even though it is more appropriate to exclude the sponge layers when computing the error since the



(a) L^2-norm error for the whole domain.



(b) L^2-norm error for the domain not including the sponge layers.

Figure 93: L^2-norm error for a vortex going through the boundary for different absorbing sponge layer strength.

governing equation inside them is modified, the trend in both cases is the same. Higher values of β_0 start affecting the solution earlier, but achieve a lower error at the end, what means better boundary absorption. If the layer were thinner, the solution would not be affected during the earlier stages, however, the strength value would need to be higher, and may cause some reflectivity from the layer itself, as observed in Section 10.3.5. In conclusion, β_0 has to be adjusted empirically based on the specific case under consideration.

10.4 Turbulent flows

10.4.1 Stability limit for a convection-diffusion equation

The size of the time step in an explicit time stepping is known to be limited by stability criteria that is scheme dependent. The convective term imposes a well known stability limit. The viscous term also imposes a limit that is usually less known. Here we try to estimate this coupled limit using the linear convection-diffusion equation:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = a\frac{\partial^2 u}{\partial x^2} \tag{155}$$

where u is a scalar, c is the convection velocity, and a is the viscosity coefficient. A 1D domain with a length of 2π with periodic boundaries is used. The initial solution is $u = \sin(x)$. The values of c and a, and the order p are varied while the time step and the grid size are kept constant. This provides a stability map with stable and unstable results. Figure 10.4.1 shows this map, where Δt_c and Δt_v are:

$$\Delta t_c = \frac{\Delta x}{c\ 2\ (p+1)} \tag{156}$$

$$\Delta t_v = \frac{\Delta x^2}{ap^4} \tag{157}$$

The stability limit in figure 10.4.1 was obtained heuristically, and leads to the following stability limit:

$$1/\Delta t = 1/\Delta t_c + 4/\Delta t_v \tag{158}$$

What can be rewritten as:

$$\Delta t = \frac{\Delta t_c\ 4\ \Delta t_v}{\Delta t_c + 4\Delta t_v} \tag{159}$$

The stability was studied for a 1-D linear equation. In the general case, a system of multi-dimensional, non-linear equations may have a limit even lower, thus a safety margin should be adopted for all the other cases.

10.4.2 Isotropic decaying turbulence

10.4.2.1 Decaying isotropic turbulence

A set of finite difference schemes were compared under decaying isotropic turbulence in [67]. In order to compare against their results the same procedure and values are used to generate the initial condition. Note that the authors in [67] used shock-capturing schemes for this case since it may form weak shock waves (eddy shocklets) due to the high turbulent Mach number. Nonetheless, our scheme stayed stable and showed good results without requiring the moment limiter. Only the limiter

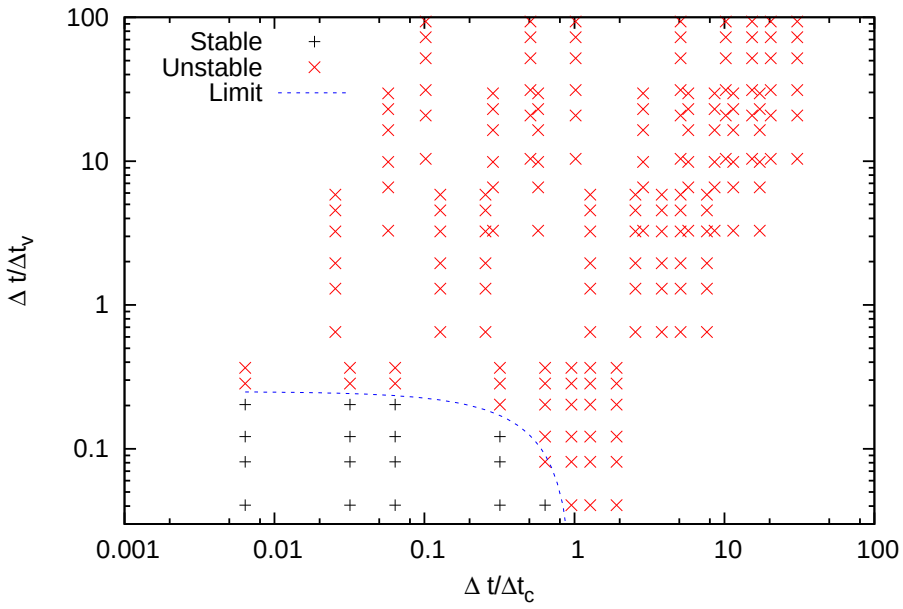


Figure 94: Stability map

to preserve positivity was used. In fact, shock-capturing schemes tend to be over dissipative to simulate isotropic turbulence [40, 67], so it is better to avoid the limiter.

The relevant parameters are turbulent Mach number and the Taylor-scale Reynolds number

$$M_t = \frac{\sqrt{\langle u_i u_i \rangle}}{\langle c \rangle} \tag{160}$$

$$Re_\lambda = \frac{\langle \rho \rangle \, u_{rms} \lambda}{\langle \mu \rangle} \tag{161}$$

where

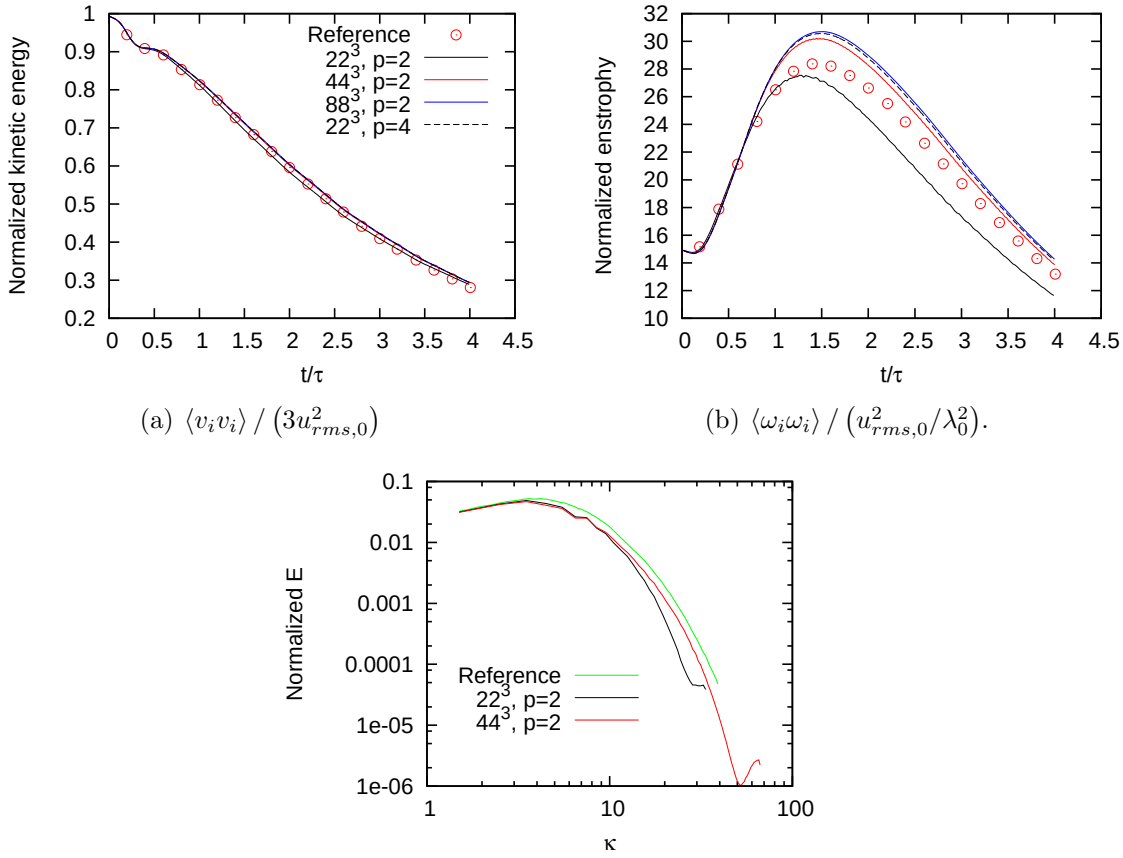
$$u_{rms} = \sqrt{\frac{\langle u_i u_i \rangle}{3}} \tag{162}$$

$$\lambda = \sqrt{\frac{\langle u_1^2 \rangle}{\left(\langle \frac{\partial u_1}{\partial x} \rangle \right)^2}} \tag{163}$$

Initially, the pressure and density are constant while the velocity fluctuates following the rule

$$E(k) = k^4 \exp\left(-2(k/k_0)^2\right) \tag{164}$$

This spectrum leads to $\lambda_0 = 2/k_0$. As in [67]: $k_0 = 4$, $M_t = 0.6$, $Re_\lambda = 100$. The domain is a periodic cube with size $(2\pi)^3$. Four grids are tested. These have 22^3, 44^3, and 88^3 elements with $p = 2$, and the fourth grid has 22^3 elements with $p = 4$. The normalized kinetic energy, enstrophy $(\omega_i\omega_i)$, and energy spectrum are compared against [67] in Fig. 95. The time is normalized by the eddy turn-over time, $\tau = \lambda_0/u_{rms,0}$. The spectrum is computed at $t = 4\tau$. The results are very close to the reference. Also, the skewness and kurtosis are shown in Fig. 96, defined



(a) $\langle v_i v_i \rangle \,/\, \left(3u^2_{rms,0}\right)$

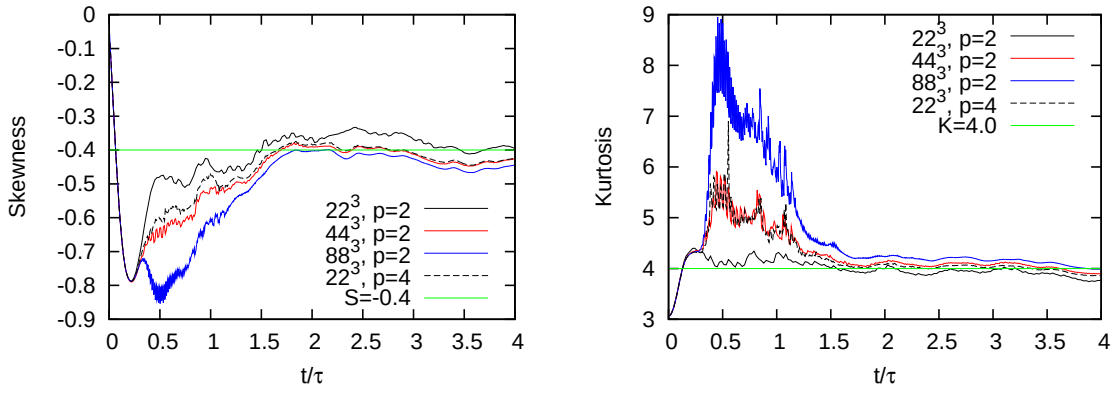
(b) $\langle \omega_i \omega_i \rangle \,/\, \left(u^2_{rms,0}/\lambda^2_0\right)$.

(c) $E/\left(3/2u^2_{rms,0}\right)$

Figure 95: Results for decaying isotropic turbulence compared against [67].

as $\left\langle \left(\frac{\partial v_1}{\partial x_1}\right)^n \right\rangle / \left\langle \left(\frac{\partial v_1}{\partial x_1}\right)^2 \right\rangle^{n/2}$ with $n = 3$ and $n = 4$, respectively. The plots include estimations of the expected values based on the Re_λ as measured in [5]. Both variables show a behavior that is grid dependent for $t < 2\tau$. This should be caused by the

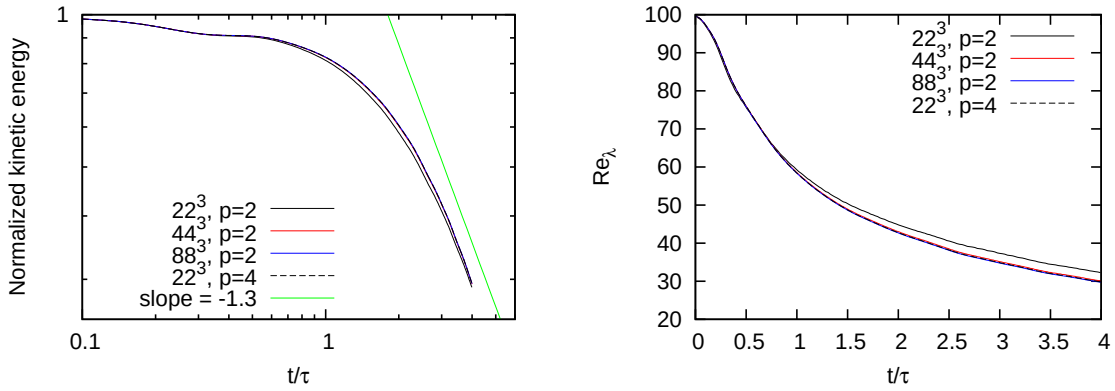
initial value that is non-physical. Note that after a certain period the problems in the initial solution decay and the variables tend to the expected values. The normalized



(a) (b)

Figure 96: Velocity-derivative skewness and kurtosis for decaying isotropic turbulence.

kinetic energy is shown in Fig. 97(a) with logarithmic axes. Note that the slope of the decay is close to -1.3 as expected [104]. Finally, the Re_λ is shown in Fig. 97(b). Based on these plots, the grid with 22^3 elements and $p = 2$ is not converged. It is



(a) (b)

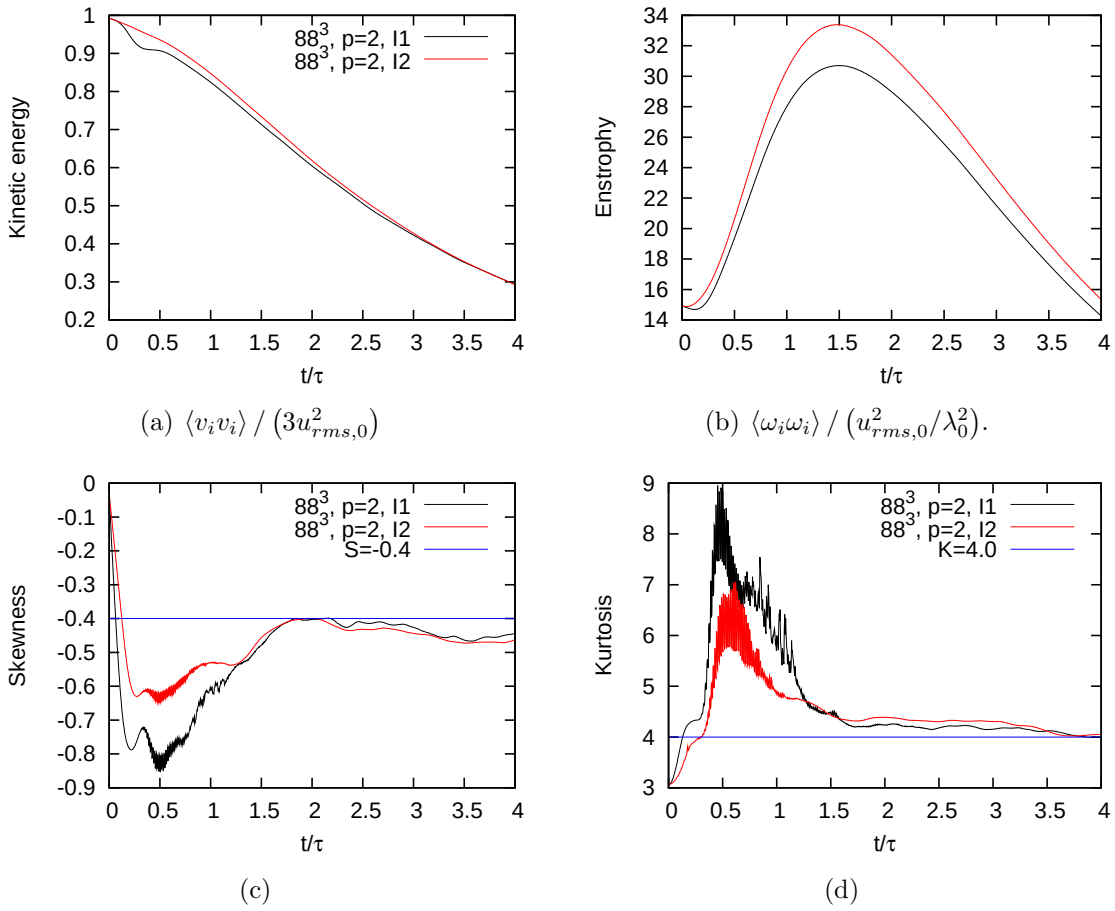
Figure 97: Kinetic energy and Re_λ for decaying isotropic turbulence.

interesting to note that the grid with 22^3 elements and $p = 4$, and the grid with 88^3 elements and $p = 2$ provide almost the same results. However, the main difference is

that with 64 processors, the grid with $p = 4$ was 13 times faster.

10.4.3 Effects of the initialization

The flow in the last section was initialized with constant pressure in order to match reference [67]. However, the technique in [115] allows the flow to develop more naturally. Let's refer to the former and later initialization as I1 and I2, respectively. The simulation is conducted again with 88^3 elements with $p = 2$. The comparison between I1 and I2 is shown in Figs. 98 and 99. I2 reduces the initial bump in the kinetic



(a) $\langle v_i v_i \rangle \,/\, \left(3u^2_{rms,0}\right)$

(b) $\langle \omega_i \omega_i \rangle \,/\, \left(u^2_{rms,0}/\lambda^2_0\right)$.

(c)

(d)

Figure 98: Results for decaying isotropic turbulence with initializations I1 and I2.

energy, the peaks in the skewness and kurtosis, and a small bump in the Reynolds number. However, the flow does not present a physical behavior for I2 until $t = 2\tau$,

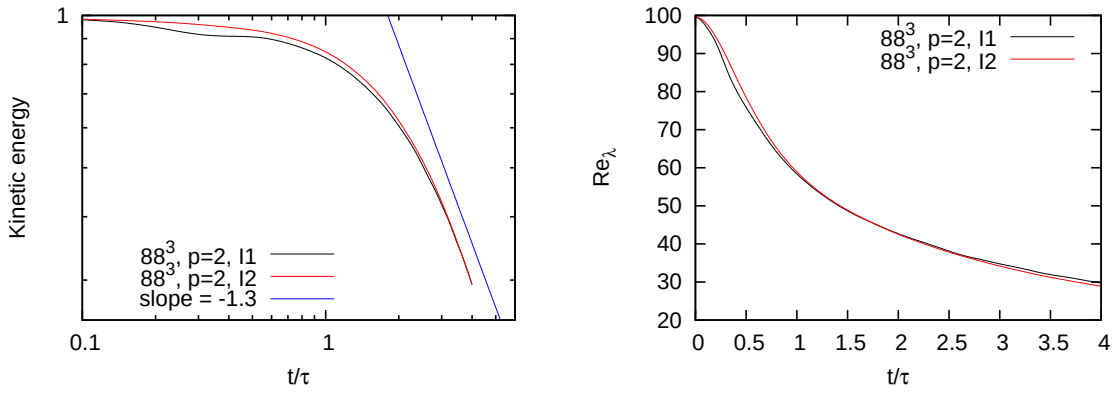
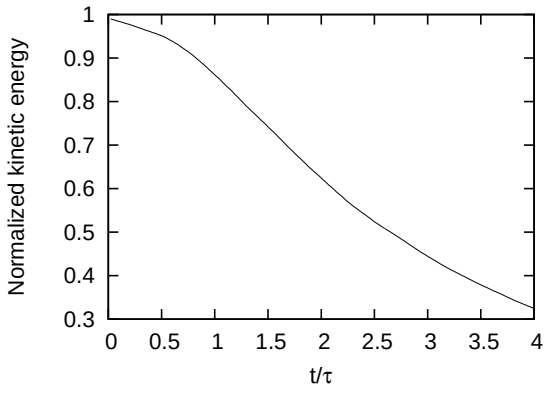


Figure 99: Kinetic energy and Re_λ for decaying isotropic turbulence with initializations I1 and I2.

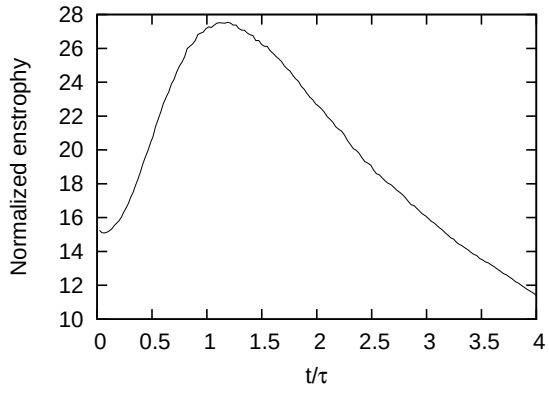
just like I1. In sumarry, during the initial transient I1 behaves less physically, but both reach physical coherence at the same time.

10.4.4 Large-eddy Simulation

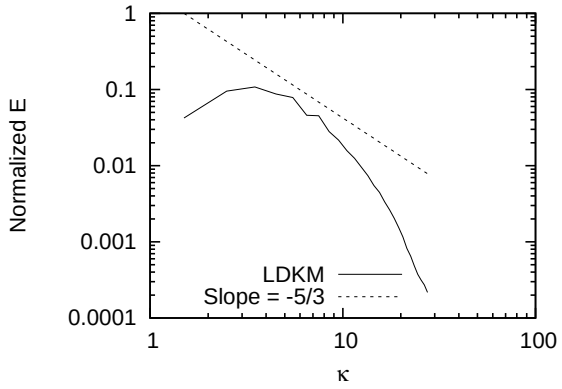
Given that the solver can successfully capture isotropic turbulence with DNS, we now evaluate LES with LDKM at increased Reynolds number to increase the inertial range. The field is initialized with the "I2" method, which was shown to be more natural. The same parameters are used except for $Re_\lambda = 200$. The grid has 11^3 elements with $p = 4$. The kinetic energy, enstrophy, and energy spectrum are shown in Fig. 100. In this case, the energy spectrum is calculated at $t = 2\tau$ (not at $t = 4\tau$ as in the previous section). It would seem that there is an inertial where the slope is close to $-5/3$. The skewness and kurtosis of the velocity derivatives are shown in Fig, 101. Here also the skewness reaches a steady value approximately at $t = 2\tau$. The kurtosis does not present a large peak as in the previous section, but that was grid dependent during the non-physical initial transient and at least for $t > 2\tau$ the approximate value is 4. The skewness and kurtosis are more noisy than in the DNS case. The reason has not been determined, but it could be related to the discontinuity



(a) $\langle v_i v_i \rangle \,/\, \left(3u_{rms,0}^2\right)$



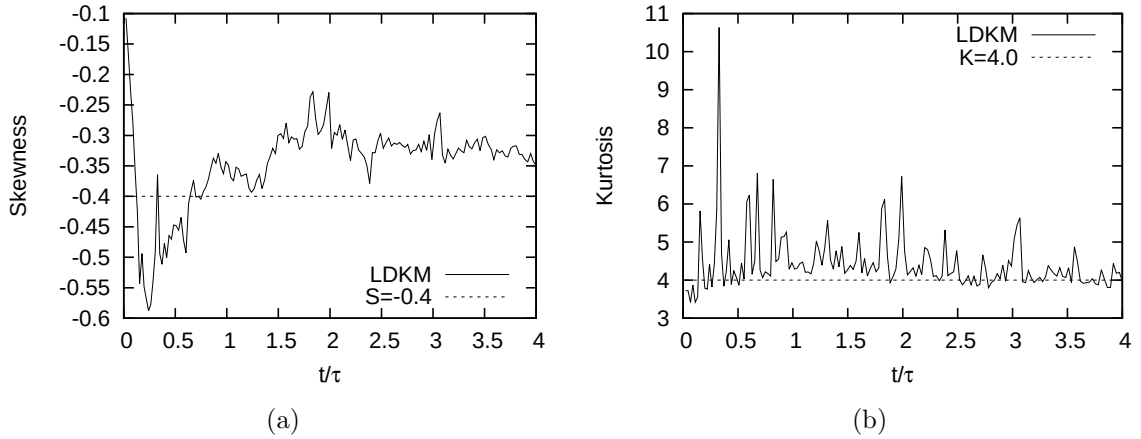
(b) $\langle \omega_i \omega_i \rangle \,/\, \left(u_{rms,0}^2/\lambda_0^2\right)$.



(c) $E/\left(3/2u_{rms,0}^2\right)$

Figure 100: Results for decaying isotropic turbulence with LES-LDKM.

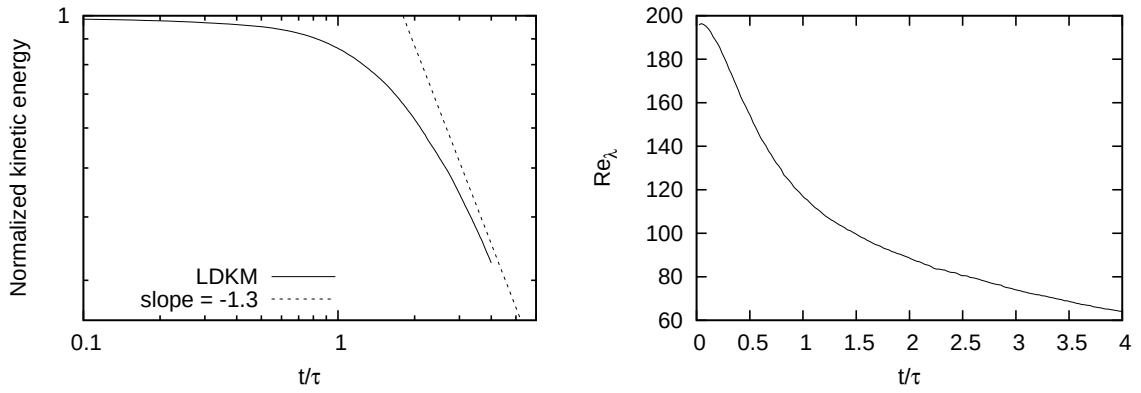
between elements in the model's coefficients. The kinetic energy in Fig. 102(a) decays



(a) (b)

Figure 101: Velocity-derivative skewness and kurtosis for decaying isotropic turbulence with LES-LDKM.

at the expected rate of -1.3. Figure 102(b) shows that at $t = 2\tau$ $Re_\lambda \cong 88$, which is the time when the spectrum was calculated, and such value of Re_λ verifies that some inertial range should exist.



(a) (b)

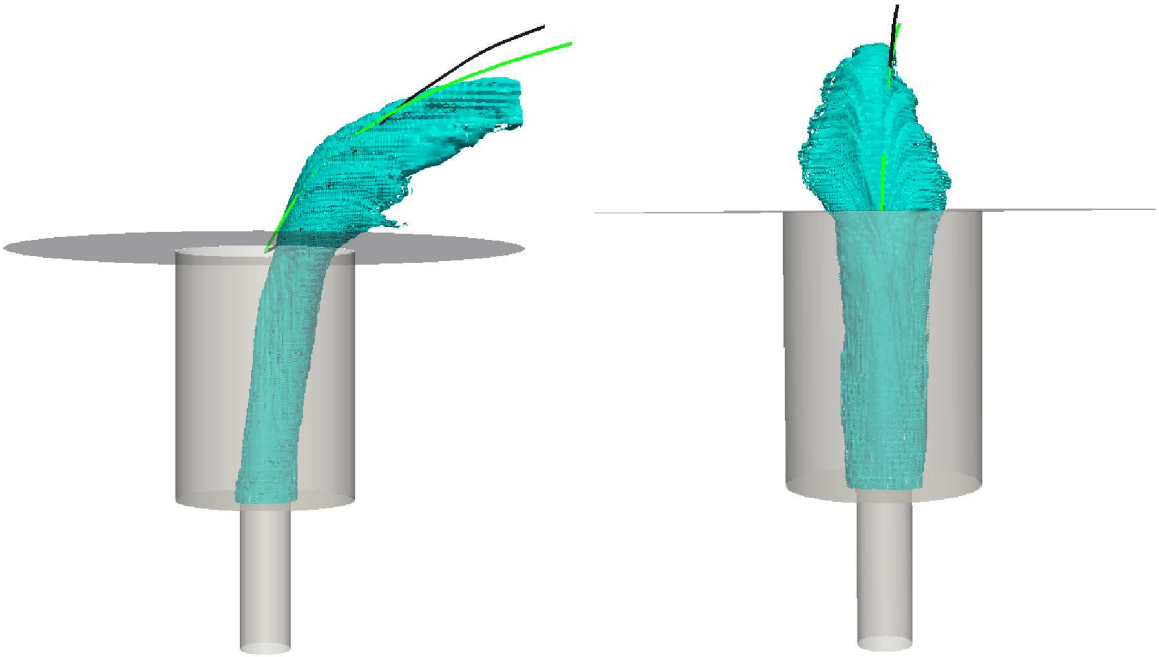
Figure 102: Kinetic energy and Re_λ for decaying isotropic turbulence with LES-LDKM.

10.5 Liquid jet in cross flow

The Combustion Lab at Georgia Tech, under the supervision of Ben Zinn, conducted a set of experimental tests of jets in cross flow (JICF) at sub- and super-critical conditions. Their case MCF01 is considered, which makes use of the Spraywell injector for numerical study. The parameters that characterize this case are: $J = 4.86$, $We = 2267$, $We_{\text{aero}} = 157$, $P = 1.93\ MPa$, $T_{\text{fuel}} = 298.0\ K$, and $T_{\text{air}} = 588.6\ K$. The diameter of the injector and cavity are $D = 0.671\ mm$ and $2.3622\ mm$, respectively, while their lengths are $1.9558\ mm$ and $3.2512\ mm$, respectively. See [163] for more details.

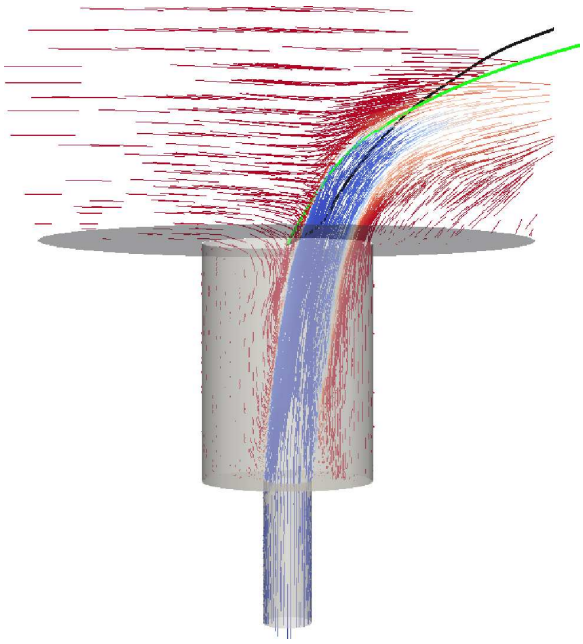
Initially, a small cylindrical geometry was used with a diameter in the cross-flow region of $11D$, see Fig. 103. The black line is the mean trajectory measured experimentally and the green line is the corresponding regression suggested in [163]. The simulated jet matches the experiments in the initial stages, however, the reduced size of the geometry affects the evolution of the trajectory.

A larger grid was simulated with a size from the center of the injector to the downstream end of $15D$. The downstream boundary has a sponge layer of $1D$. Figure 104 shows the interface based on a volume fraction of gas of 0.9 with velocity streamlines in the boundary layer of the cross flow. Figure 105 shows the measured trajectory in black and the regression in red together with a volumetric representation of the fuel. As the liquid leaves the injector and is exposed to the cross flow, the high shear makes the interface of the jet diffuse. Figure 106 shows the interface ($\alpha_{gas} = 0.9$) together with the volumetric plot. The mean trajectory of the fuel seems to be matched very well with the larger geometry. The RMS value of the trajectory from the simulation and the experiment are shown in Fig. 107. The first point should be ignored, since the measuring error there is large. The shape of the curves are similar, but there is a significant offset between them. One cause of the mismatch could the diffusion of the interface. The unphysical dispersion of the interface produces an apparent greater



(a) Side view.

(b) Front view.



(c) Velocity vectors colored by the volume fraction (red for air and blue for fuel).

Figure 103: Case MCF01 with a reduced geometry. The interface is based on a volume fraction of gas of 0.9.

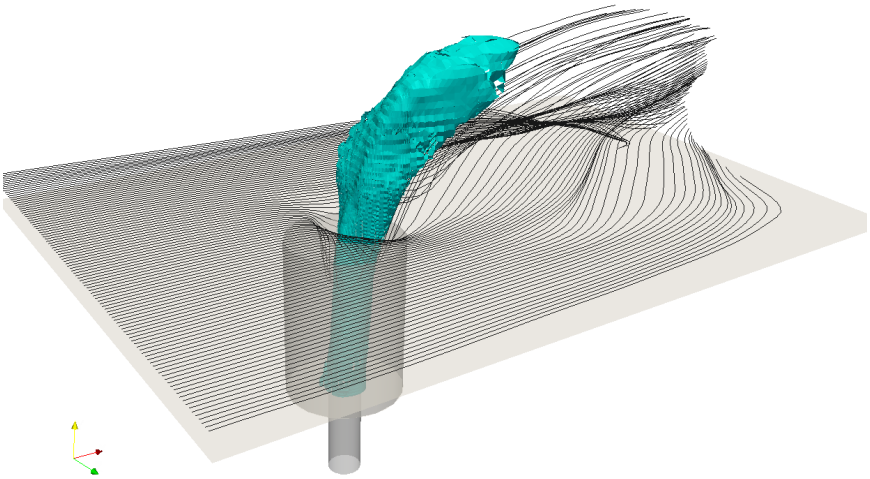


Figure 104: Case MCF01. The interface based on a volume fraction of gas of 0.9.

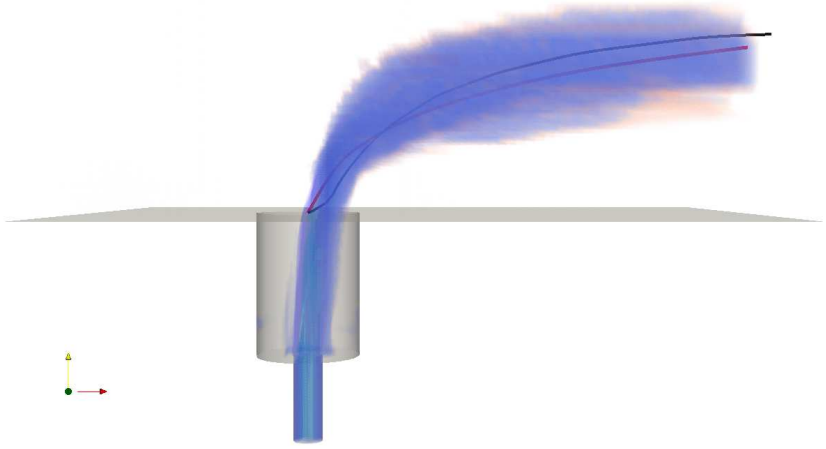


Figure 105: Case MCF01. Volumetric plot of the fuel.

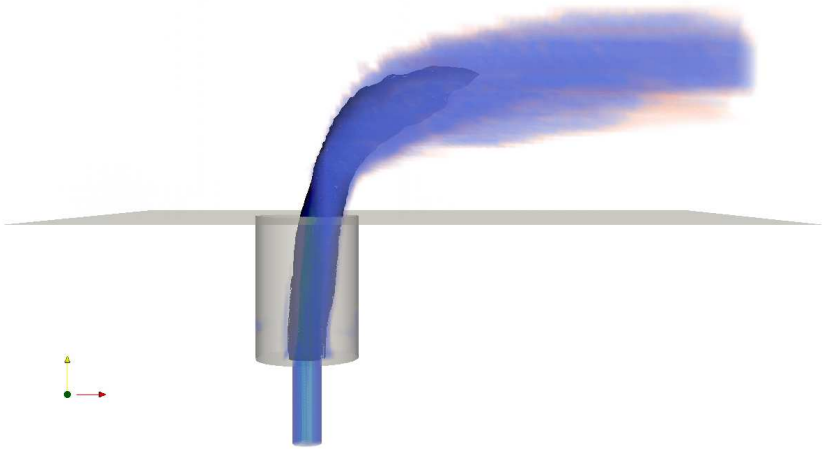


Figure 106: Case MCF01. The interface based on a volume fraction of gas of 0.9.

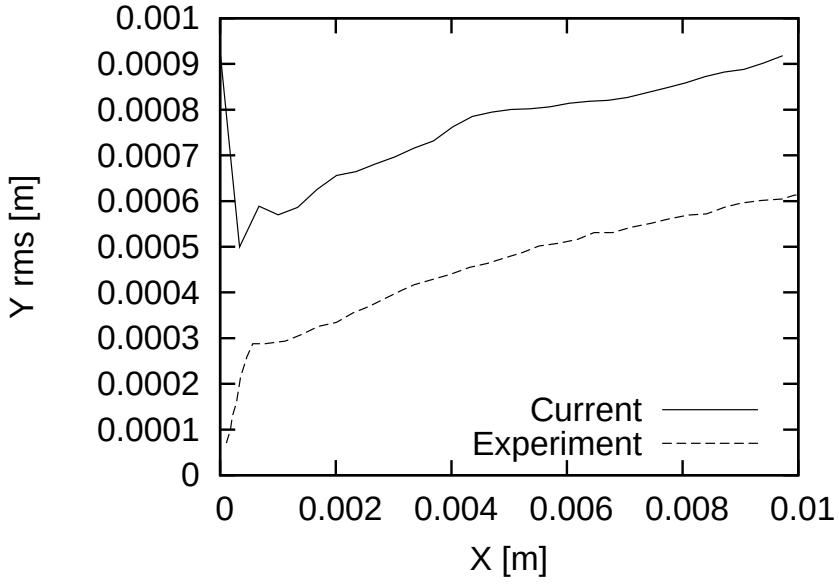


Figure 107: Case MCF01. RMS value of the trajectory.

distribution of the jet.

The boiling temperature of Jet-A at $P = 1.93\ MPa$ is approximately 583.2 K. Which is slightly below the temperature of the cross-flow. The temperature on the vertical center-plane is shown in Fig. 108 together with the interface ($\alpha_{gas} = 0.9$) in black. By the time the fuel reaches the boiling temperature, it is already widely

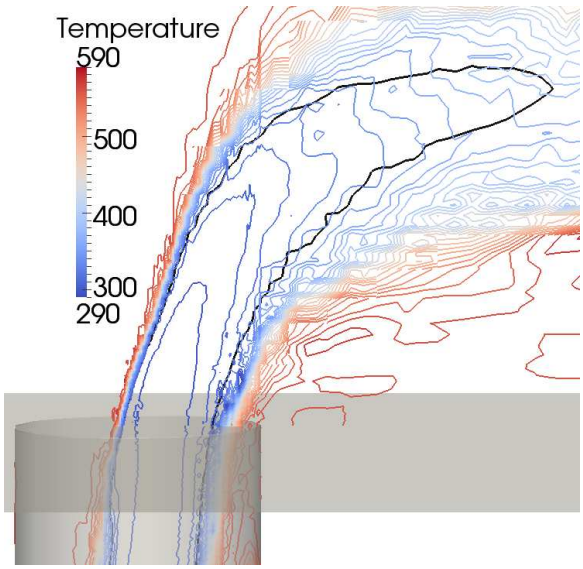


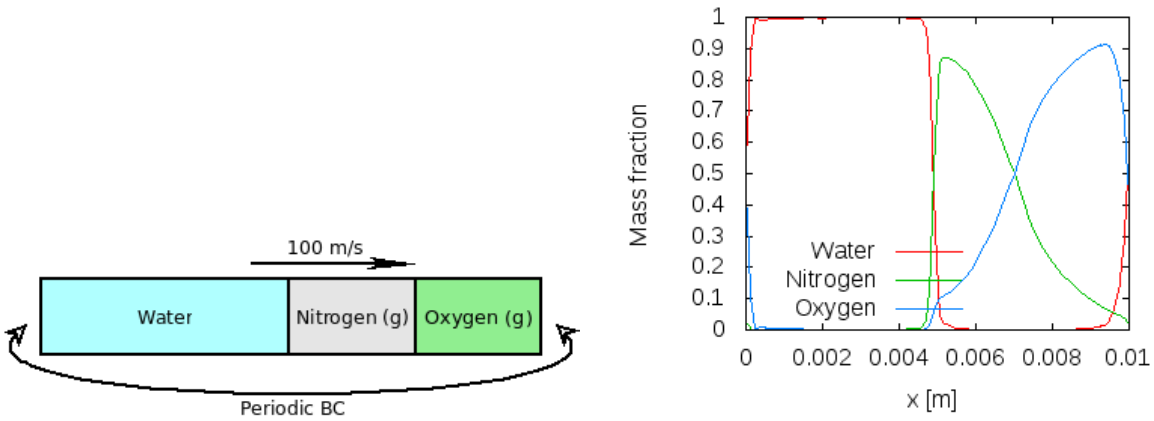
Figure 108: Case MCF01. Temperature on the center-plane. The interface based on a volume fraction of gas of 0.9 is in black.

diffused. However, before the boiling condition is reached, evaporation should be occurring, but its physics are not being modeled in this simulation. It would seem that in order to improve the simulation of this case interface sharpening and evaporation are needed.

10.6 Multi-component multi-phase model

The first model presented in Chapter 3 for multiple phases with multiple components is demonstrated here. A 1-D domain of 0.01 m long with periodic boundaries is used. Initially, the components are distributed as: water for $x < 0.005\ m$, nitrogen (N_2) for $0.005\ m < x < 0.0075\ m$, and oxygen (O_2) $0.0075\ m < x$ (see Fig. 109(a)).

Water is part of one phase, while nitrogen and oxygen belong to a different phase. It is assumed that the two phases do not mix with each other. Initially, there is a uniform velocity of 100 m/s. The viscous coefficients were increased several orders of magnitude respect to their real values in order to accelerate the process for this demonstration. The result after one period ($t = 0.1\ ms$) is shown in Fig. 109(b). The two gases diffuse into each other, while water, which is forms another immiscible



(a) Initial condition.

(b) After one cycle.

Figure 109: Convection of a multi-component multi-phase flow.

phase, does not diffuse as expected.

This simple test is the first step toward evaporation, which, for example, could be captured the following way. Phase 1 is composed by liquid fuel. Phase 2 is composed by oxygen, nitrogen, and fuel vapor. The components of phase 2 are all gases than can diffuse into each other. The initial concentration of liquid fuel in phase 1 is 100%, while the concentration of fuel vapor in phase 2 is 0%. Given an appropriate formulation for the mass transfer from phase 1 to phase 2, fuel will start to evaporate. The mass of phase 1 will decrease, while the mass of phase 2 and the concentration of fuel vapor in it will increase. This effect could be validated with the classical Steffan problem. A similar approach could be done to include boiling and cavitation as done in [102].

CHAPTER XI

CONCLUSION

Diffused Interface Methods (DIM) were solved with a Discontinuous Galerkin (DG) method. DIM cannot naturally control the thickness of the interface and requires an artificial numerical technique for this purpose. The interface sharpening approaches found in the literature until the date when the current study was conducted, presented several drawbacks. Hence, a new sharpening method was developed, which respect to [70] generalized the strength parameter to make it problem independent. It presented great properties in 1D, but for 2D some problems where noted, like violation of conservation properties. The other techniques in the literature were not appropriate for our compressible DIM and applications, or were not better than our sharpening scheme. So & Adams [127] recently published a method that solves all these issues. In principle, this method could be applied to our new approach and discuss it further in the Future Work (Chapter 12).

DIM presents a large range of scales since the interface has to be captured and kept much smaller than all the other features of the flow. Given this issue and the numerical diffusion, a scheme with Adaptive Mesh Refinement (AMR) and minimal numerical diffusion was needed. The Discontinuous Galerkin (DG) method is well known for being friendly with AMR and for presenting very low numerical diffusion, especially due to its flexibility to increase the order of accuracy easily.

I extended the popular Moment Limiter (ML) found in the literature to hexahedral meshes with AMR. Also, I adapted the "troubled-cell" detector known as AP-TVD to our DG solver. The AP-TVD detector was developed in the literature for Spectral Differences (SD) with averaged-derivative basis, and I adapted it for DG with a

Legendre basis. Our approach resulted to be more efficient for the cases studied. A new error estimator was presented to control the refinement/coarsening criteria for AMR. This error estimator is more efficient for the cases studied than other common techniques in the literature. In addition, the solver was tested for high orders in space and time. When discontinuities dominate, high order of accuracy in space may not improve the efficiency of the solver. Probably, the combination of h and p adaptivity would improve the efficiency by reducing h in discontinuous zones and increasing p in smooth zones. Time integration was done explicitly with the TVD Runge-Kutta scheme and with the Spectral Deferred Correction (SDC) method. The second one has the advantage of being easy to extend to any order of accuracy. It was observed that when the time-step size required for stability is close to the requirement for accuracy a higher order in time is more efficient. Thus, in fluid mechanics simulations at relatively low Mach number, high order in time is not needed. For a larger discussion on this see Section 6.3. The scalability of all these features was evaluated on several platforms with the weak-scaling technique. The efficiency observed at 4096 cores was approximately 86%.

I presented the first application in the literature of DG to DIM (see Chapter 7. In this case, I used the inviscid 4-eq. model, which is a simple DIM. The numerical flux used was a Roe-type flux, which can violate the entropy rule. This scheme was applied to shock-bubble interaction problems and the numerical flux did not present any problem in the cases studied.

In Chapter 8 I showed a new scheme that avoids Roe-type numerical flux and is based on the Local Discontinuous Galerkin (LDG) method. This study is the first one in the literature to use LDG for for a multi-phase model. In this case, I used the 5-eq. model, which is a very common DIM. I tested the limiter with conservative variables and primitive variables. The second option is more accurate than the first one, at

least for the test cases studied. Characteristic variables cannot be used in multi-dimensions when using ML. In addition, a new technique to treat different Equations of State (EOS) was shown. Four EOSs were tested: Ideal Gas, Stiffened Gas, van der Waals, and Peng-Robinson. This allowed for simulations of sub- and super-critical flows. Also, an additional limiter to preserve positivity while minimizing numerical diffusion was derived. This limiter is an extension of the approach in [159] to multi-phase models. DIM is a first-order approach because the interface has a thickness proportional to the size of the elements. Thus, high-order elements resulted to be less efficient than low-order elements. Finally, the solver was applied to the study of bubble collapse under water at very high pressures. The current approach matched very well other numerical studies, which are close to experiments but the extremes of the regime tested could be improved. The van der Waals EOS was used as an attempt to improve the accuracy of the mathematical model, but no difference was observed. Thus, the partial mismatch between numerical and experimental studies has not been explained yet.

In Chapter 9 I presented the 5-eq. model with surface tension using the LDG method. This study is the first one in the literature to use LDG for surface tension. Surface tension presents numerical issues with most common schemes, mostly due to the need to compute high-order derivatives at the interface. Marchandise et al. [88] used DG to convect the level-set equation, and to reduce numerical oscillations smoothed out the interface using a least-squared approximation. Our method is more compact and it avoids the computational cost of that approach. Most capturing methods present spurious oscillations, the ones I observed with the current technique are of comparable magnitude as other common capturing schemes (e.g., CSS and CSF). The solver was applied to the interaction of single-drops with a wall.

Decaying isotropic turbulence was studied with DNS and LES. The former was done with $p = 2$ and $p = 4$. The higher order resulted to be significantly more

efficient. Thus, $p = 4$ was used for the LES analysis. Higher Reynolds number was used for LES and the results matched the theoretical expectations. This is the first study of DG with LDKM in the literature up to the author's knowledge.

Most of the techniques discussed in this thesis were applied together in a liquid jet in cross flow (see Section 10.5). The trajectory was adequately captured. The dispersion of the jet was over estimated, probably, due to the unphysical diffusion of the interface. Another relevant effect that was not included in the simulation is evaporation. Section 10.6 demonstrated some of the capabilities of the multi-component multi-phase model. This model could form the foundation for evaporation in future research. This is an area of active research and it is suggested in the following chapter as future work.

DG was proven to be a modern versatile scheme with a great potential. The only main disadvantage of DG I observed is its computational cost (CPU and memory). However, it should be a competitive approach when complex geometries are present since a very high order can still be used, while FV and FD cannot handle complex geometries with high order.

CHAPTER XII

FUTURE WORK

In this concluding section, I discuss possible extensions to the research presented in this work.

Phase change. Evaporation is a phenomenon present in combustors and it is a key element of the combustion process. The modeling of the process in complex systems is in its very initial stages in the numerical field. The formulation of the model and the scheme for complex systems is not simple and deserves special attention. The elements that must be present for evaporation to occur are multiple phases and multiple species. Hence, the multi-component multi-phase model introduced in this thesis can be the foundation for future developments of phase transition (see Section 10.6).

Couple two-phase model with Lagrangian drops. Several multiphase applications involve a large range of scales. For example, liquid fuel injection requires to simulate from the combustor down the small drops formed in the spray. The individual drops can be modeled as Lagrangian points relatively well. However, the accuracy of the models to represent the formation of the spray is limited. Thus, coupling the (Eulerian) multi-fluid approach with an Eulerian-Lagrangian formulation can ease the computational need and make such applications feasible. A few researchers have already shown some techniques (e.g., see [83]). Fundamental research can be done regarding the closure used to transition from Eulerian to Lagrangian representation. After that, a wide range of applied research can be done, for example, for liquid jets in cross flow. DiGGIT already interacts with a Lagrangian module demonstrated in Appendix A.

Implement an interface sharpening algorithm. The interface can be widely diffused in regions of large shear even with low-diffusion schemes when using DIM. Interface sharpening algorithms try to revert the undesired effect of the numerical scheme and the drawback of the mathematical model. A sharpening schemes was presented in this thesis and a few other approaches can be found in the literature [71, 123]. However, all of these techniques lack of conservation properties. The most advance approach I have seen was published recently in [127] and it conserves mass, momentum, and total energy. Based on the literature available until this thesis was written, I recommend to implement the method in [127] and probably use some ideas from [128]. The approaches in [127] and [128] are based on FV and FD formulation, respectively, but they could be adapted for DG.

Research into Variational Multi-Scale methods. One of the most proper ways to conduct LES with AMR and DG is using Variational Multi-Scale (VMS) methods. Not only does it provide a mathematically correct framework for LES, but it also improves the accuracy of simple models [28, 13]. The models proven in this thesis could be used in a VMS framework. Given that it is a relatively new technique useful for engineering applications, it opens opportunities for fundamental and applied research.

Conservative surface tension. Almost every scheme to model surface tension presents spurious oscillations. The most common approach is the CSF applied as source terms. However, it seems to be [60, 100] that if it is applied in a conservative way, modifying the fluxes and the state vector, these fluctuations decrease. This topic opens opportunities for fundamental research since the claim has not been thoroughly studied for models like the 5-eq. model. Also, this improvement should increase the accuracy of applied simulations.

Addition of tetrahedral elements. As shown in this thesis, non-Cartesian grids with hexahedral elements have variable Jacobian. When the Jacobian varies the quadrature-free approach cannot be used and the computational cost is higher [4]. Tetrahedral elements with flat faces can be used for complex geometries while having constant Jacobian, so the quadrature-free technique can be used. This is usually relaxed at curved boundaries where the elements must be curved and, thus, they have a variable Jacobian. Even though hexahedral elements may be more efficient than tetrahedral elements with Cartesian grids, generic cases would run faster with tetrahedral elements and their grids would be much easier to generate. The limiter and detector presented in this thesis cannot be used for tetrahedral or triangular elements, but other options exist in the literature [154, 149].

APPENDIX A

LAGRANGIAN FORMULATION

A.1 Formulation

When lots of drops of particles are present and their sizes are too small to resolve for practical computations, the entities are usually modeled as point particles making use of a Lagrangian formulation. Fluid particles (massless particles) are tracked with this ODE:

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{v} & \text{for } t > 0 \\ \mathbf{x} = \mathbf{x}_0 & \text{for } t = 0 \end{cases} \tag{165}$$

where $\mathbf{v}$ is the velocity of the Eulerian field at location $\mathbf{x}$, and $\mathbf{x}_0$ is the initial position. Drops and particles with mass are tracked with this ODE:

$$\begin{cases} \ddot{\mathbf{x}} = \mathbf{F}/m & \text{for } t > 0 \\ \mathbf{x} = \mathbf{x}_0 & \text{for } t = 0 \\ \mathbf{v} = \mathbf{v}_0 & \text{for } t = 0 \end{cases} \tag{166}$$

where $\mathbf{v}_0$ is the initial velocity, $\mathbf{F}$ is a modeled force due to the interaction with the surrounding fluid, and m is its mass. Several techniques with different degrees of accuracy and complexity exist to model the force $\mathbf{F}$. Not only does the fluid affect these modeled bodies through $\mathbf{F}$, but also the fluid can be affected by them, what is usually called "two-way coupling". Other effects could also be considered, like energy and mass transfer. These factors have already been studied elsewhere [97]. Thus, the heat and mass transfer are neglected in the current study.

A.2 Method

The governing equations for the Lagrangian formulation are easily integrated in time with the same scheme as used for the Eulerian fluid.

In order to extract properties of the fluid surrounding a Lagrangian body its location within an element is needed. Known the physical location $\mathbf{x}$ and the element where its located, the computational coordinate ξ is computed with an iterative Newton-Raphson solver knowing that

$$\mathbf{x}_j = \sum_{i=0}^{(p+1)^d-1} \phi_i(\xi)\chi_{ij} \tag{167}$$

where χ_j is the modal coordinates of element j. If the iterative solver leaves the range $-1 < \xi < 1$ it means that the body might be in a neighboring element. Thus, the iteration restarts at the center of the neighbor ($\xi = 0$). Figure 110 depicts this algorithm, where the dotted arrows represent the path of the search algorithm, filled circles show the locations at sub-steps n and $n+1$, and the unfilled circles represent the element-centers to which the search jumps to each time a boundary is crossed. The latest computational location (element plus computational coordinate ξ) is stored

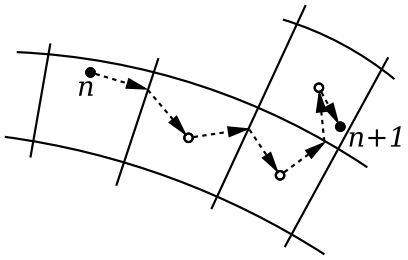


Figure 110: Lagrangian-body search algorithm.

for each Lagrangian body at each sub-step in time to make the location search faster. This approach lets the body move more than one element in one iteration without affecting the stability of the scheme.

If the iterative solver tries to leave an element and there is no neighbor, a global element-by-element search is done among all the elements in the current processor. Nonetheless, this situation is not common.

If the body enters a ghost element, the body gets transfered to the processor where this ghost element is hosted. In the case of parallel simulations it is easier to track

Lagrangian bodies that do not cross more than one element boundary per iteration.

A.3 Verification and demonstration

A.3.1 Verification

Here we track individual point particles based on the velocity of the flow such that the velocity of a particle is $\mathbf{V}(t) = \mathbf{v}(\mathbf{x}, t)$. A helical trajectory is considered as in [155] to verify the accuracy. A grid of dimension $[-\pi, \pi]^3$ has a steady velocity field defined as:

$$\mathbf{v}(\mathbf{x}) = \begin{pmatrix} 0.4 \\ x_3\ \omega(\mathbf{x}) \\ -x_2\ \omega(\mathbf{x}) \end{pmatrix} \tag{168}$$

where for $r = \sqrt{x_2^2 + x_3^2}$

$$\omega(\mathbf{x}) = \begin{cases} 1 - 3\ (r/\pi)^2 + 3\ (r/\pi)^4 - (r/\pi)^6 & \text{for } r \leq \pi \\ 0 & \text{otherwise} \end{cases} \tag{169}$$

A particle is initially located at $(-2, 0, 2)$ and evolves until $t = 10$ with a fixed time-step of $\Delta t = 0.05$. The trajectory is plot in Fig. 111. The final location can

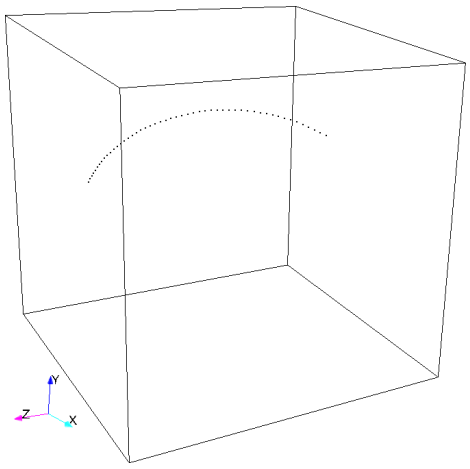


Figure 111: Helical trajectory for a Lagrangian particle shown at every $\Delta t = 0.2$.

be calculated analytically, so the final error can be quantified. For a grid with 8^3 elements, the error decreases as the order p increases, see Table 27.

Table 27: Error of a Lagrangian point for varying p.

p	1	3	5
Error	$1.752 \cdot 10^{-2}$	$5.3910 \cdot 10^{-5}$	$1.2651 \cdot 10^{-5}$

A.3.2 Fluid particles past a cylinder

A subsonic viscous flow is simulated past a cylinder. Fluid particles are injected upstream the cylinder. Figure 112 shows the flow colored by the momentum in X and the fluid particles are colored by their age. The particles follow the flow and

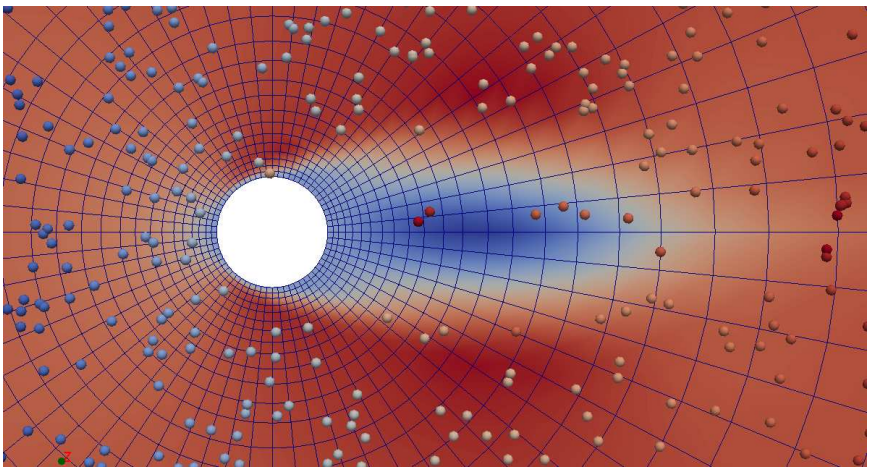


Figure 112: Fluid particles past a cylinder.

successfully stay inside the domain.

A.3.3 Jet in cross flow with particles

A gas jet is injected at 102.94 m/s together with particles into a cross flow with 67.72 m/s. The simulation is conducted in 2D. The particles have a mass of 5.2×10^{-9} kg and a diameter of 5 mm. Figure 113 shows the results at 1.1 ms. The drops are colored based on their age. Note that there is only one-way coupling, i.e. the flow is not affected by the particles.

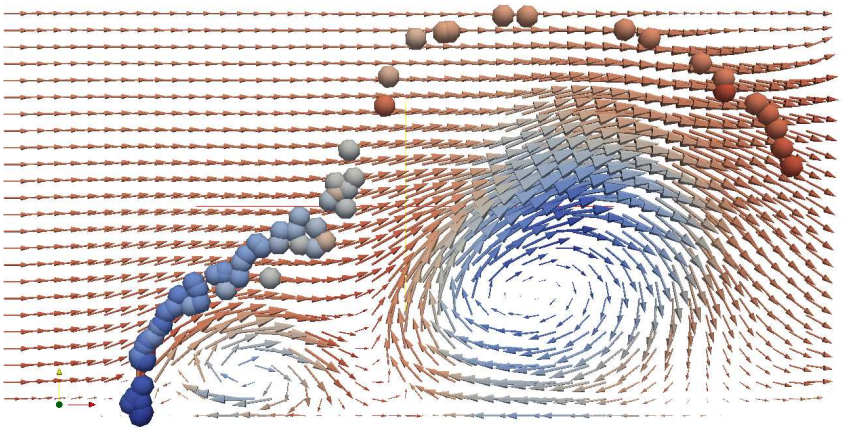


Figure 113: Demonstration of a jet in cross flow with particles.

REFERENCES

[1] Allaire, G., Clerc, S., and Kokh, S., "A five-equation model for the simulation of interfaces between compressible fluids," *Journal of Computational Physics*, vol. 181, pp. 577–616, 2002.

[2] Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, L. D., "Unified analysis of discontinuous Galerkin methods for elliptic problems," *SIMA Journal of Numerical Analisis*, vol. 39, no. 5, pp. 1749–1779, 2002.

[3] Atkins, H. L., "Continued development of the discontinuous Galerkin method for computational aeroacoustic applications," in *Computational Aeroacoustic Applications*, no. AIAA-1997-1581, 1997.

[4] Atkins, H. L. and Shu, C. W., "Quadrature-free implementation of discontinuous Galerkin method for hyperbolic equations," *AIAA Journal*, vol. 36, no. 5, p. 775, 1998.

[5] Atta, C. W. V. and Antonia, R. A., "Reynolds number dependence of skewness and flatness factors of turbulent velocity derivatives," *Physics of Fluids*, vol. 23, pp. 252–257, 1980.

[6] Baer, M. and Nunziato, J., "A two-phase mixture theory for deflagration to detonation transition in granular materials," *International Journal of Multiphase Flows*, vol. 181, no. 577-616, 1986.

[7] Ball, G., Howell, B., Leighton, T., and Schofield, M., "Shock-induced collapse of a cylindrical air cavity in water: a free-Lagrange simulation," *Shock Waves*, vol. 10, no. 265-276, 2000.

[8] Barter, G. E. and Darmofal, D. L., "Shock capturing with higher-order, PDE-based artificial viscosity," in *18th AIAA Computational Fluid Dynamics Conference*, no. AIAA-2007-3823, (Miami, FL), June 2007.

[9] Bassi, F. and Rebay, S., "A high-order accurate discontinuous Finite Element method for the numerical solution of the compressible Navier-Stokes equations," *Journal of Computational Physics*, vol. 131, pp. 267–279, 1997.

[10] Baum, M., Poinsot, T., and Thevenin, D., "Accurate boundary conditions for multicomponent reactive flows," *Journal of Computational Physics*, vol. 116, pp. 247–261, 1994.

[11] Biswas, R., Devine, K., and Flaherty, J., "Parallel, adaptive Finite Element methods for conservation laws," *Applied Numerical Mathematics*, vol. 14, pp. 255–283, 1994.

[12] Blackburn, H. M. and Schmidt, S., "Spectral Element filtering techniques for Large Eddy Simulation with dynamic estimation," *Journal of Computational Physics*, vol. 186, pp. 610–629, 2003.

[13] Bochev, P. B., Collis, S. S., Jones, R. E., Lehoucq, R. B., Parks, M. L., Scovazzi, G., Silling, S. A., Templeton, J. A., and Wagner, G. J., "A mathematical framework for multiscale science and engineering: the variational multiscale method and interscale transfer operators," Sandia Report SAND2007-6179, Sandia, 2007.

[14] Bokanowski, O. and Zidani, H., "Anti-dissipative schemes for advection and application to Hamilton-Jacobi-Bellmann equations," *Journal of Scientific Computing*, vol. 30, pp. 1–33, January 2007.

[15] Bouchut, F., "An antidiffusive entropy scheme for monotone scalar conservation laws," *Journal of Scientific Computing*, vol. 21, pp. 1–30, August 2004.

[16] Bourne, N. and Field, J., "Shock-induced collapse of single cavities in liquids," *Journal of Fluid Mechanics*, vol. 244, pp. 225–240, 1992.

[17] Bowden, F. P. and Yoffe, A. D., *Initiation and Growth of Explosion in Liquids and Solids*. Cambridge University Press, 1952.

[18] Brackbill, J., Kothe, D., and Zemach, C., "A continuum method for modeling surface tension," *Journal of computational physics*, vol. 100, pp. 335–354, 1992.

[19] Chase, M. W. and National Institute of Standards and Technology (U.S.), *NIST-JANAF thermochemical tables*. American Chemical Society, 1998.

[20] Cheng, Y. and Shu, C. W., "A discontinuous Galerkin Finite Element method for directly solving the Hamilton-Jacobi equations," *Journal of Computational Physics*, vol. 223, pp. 398–415, 2007.

[21] Cockburn, B., Hou, S., and Shu, C. W., "TVB Runge-Kutta local projection discontinuous Galerkin Finite Element method for conservation laws IV: The multidimensional case," *Math. Comp.*, vol. 54, p. 545, 1990.

[22] Cockburn, B., Lin, S. Y., and Shu, C. W., "TVB Runge-Kutta local projection discontinuous Galerkin Finite Element method for conservation laws III: One dimensional systems," *Journal of Computational Physics*, vol. 84, p. 90, 1989.

[23] Cockburn, B. and Shu, C. W., "TVB Runge-Kutta local projection discontinuous Galerkin Finite Element method for scalar conservation laws II: General framework," *Math. Comp.*, vol. 52, p. 411, 1989.

[24] Cockburn, B. and Shu, C. W., "The local discontinuous Galerkin method for time-dependent convection-diffusion systems," *SIAM Journal of Numerical Analysis*, vol. 35, pp. 2440–2463, 1998.

[25] Cockburn, B. and Shu, C. W., "The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems," *Journal of Computational Physics*, vol. 141, pp. 199–224, 1998.

[26] Cockburn, B. and Shu, C.-W., "Runge-Kutta discontinuous Galerkin methods for convection-dominated problems," *Journal of Scientific Computing*, vol. 16, no. 3, p. 173, 2001.

[27] Coleman, A. J., Crum, J. E. S. L. A., and Dyson, M., "Acoustic cavitation generated by an extracorporeal shockwave lithotripter," *Ultrasound in Medicine & Biology*, vol. 13, no. 2, pp. 69–76, 1987.

[28] Collis, S. S., "The dg/vms method for unified turbulence simulation," in *32nd AIAA Fluid Dynamics Conference and Exhibit*, no. AIAA-2002-3124, 2002.

[29] Collis, S. S., "Discontinuous Galerkin methods for turbulence simulation," in *Proceedings of the Summer Program 2002*, p. 155, Center for Turbulence Research, 2002.

[30] Colonius, T., "Modeling artificial boundary conditions for compressible flow," *Annual Review in Fluid Mechanics*, vol. 36, pp. 315–345, 2004.

[31] Després, B. and Lagoutière, F., "Contact discontinuity capturing schemes for linear advection and compressible gas dynamics," *Journal of Scientific Computing*, vol. 16, pp. 479–524, December 2001.

[32] Ding, Z. and Gracewski, S. M., "The behaviour of a gas cavity impacted by a weak or strong shock wave," *Journal of Fluid Mechanics*, vol. 309, pp. 183–209, 1996.

[33] Dubois, T. and Jauberteau, F., *Dynamic multilevel methods and the numerical simulation of turbulence*. Cambridge University Press, 1999.

[34] Dupuy, P. M., Kleinohl, N., Fernandino, M., Jakobsen, H. A., and Svendsen, "Droplet-surface impact at high pressures," *Chemical Engineering Science*, vol. 65, no. 5320-5343, 2010.

[35] Dutt, A., Greengard, L., and Rokhlin, V., "Spectral deferred correction methods for ordinary differential equations," *BIT*, vol. 40, pp. 241–266, 2000.

[36] Fedkiw, R. P., Aslam, T., Merriman, B., and Osher, S., "A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method)," *Journal of Computational Physics*, vol. 152, pp. 457–492, 1999.

[37] Flaherty, J. E., Krivodonova, L., Remacle, J. F., and Shephard, M. S., "Aspects of discontinuous Galerkin methods for hyperbolic conservation laws," *Finite Elements in Analysis and Design*, vol. 38, pp. 889–908, 2002.

[38] Fuster, D., Bague, A., Boeck, T., Moyne, L. L., Leboissetier, A., Popinet, S., Ray, P., Scardovelli, R., and Zaleski, S., "Simulation of primary atomization with an octree adaptive mesh refinement and VOF method," *International Journal of Multiphase Flow*, vol. 35, pp. 550–565, 2009.

[39] Fyfe, D. E., Oran, E. S., and Fritts, M. J., "Surface tension and viscosity with lagrangian hydrodynamics on a triangular mesh," *Journal of Computational Physics*, vol. 76, pp. 349–384, 1988.

[40] Garnier, E., Adams, N., and Sagaut, P., *Large Eddy Simulation for Compressible Flows*. Springer, 2009.

[41] Giraldo, F. X., Hesthaven, J. S., and Warburton, T., "Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations," *Journal of Computational Physics*, vol. 181, pp. 499–525, 2002.

[42] Glimm, J., Grove, J. W., Li, X. L., Shyue, K. M., Zeng, Y., and Zhang, Q., "Three-dimensional front tracking," *SIAM Journal of Scientific Computing*, vol. 19, no. 3, pp. 703–727, 1998.

[43] Gottlieb, S., Ketcheson, D., and Shu, C., "High order Strong Stability Preserving time discretizations," *Journal of Scientific Computing*, vol. 38, pp. 251–289, 2009.

[44] Gottlieb, S. and Shu, C., "Total Variation Diminishing Runge-Kutta schemes," *Mathematics of Computations*, vol. 67, pp. 73–85, January 1998.

[45] Gottlieb, S., Shu, C., and Tadmor, E., "Strong Stability-Preserving high-order time discretization methods," *SIAM Review*, vol. 43, no. 1, pp. 89–112, 2001.

[46] Grooss, J. and Hesthaven, J., "A level set discontinuous Galerkin method for free surface flows," *Comput. Methods Appl. Mech. Engrg.*, vol. 195, pp. 3406–3429, 2006.

[47] Gueyffier, D., Li, J., Nadim, A., Scardovelli, R., and Zaleski, S., "Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows," *Journal of Computational Physics*, vol. 152, pp. 423–456, 1999.

[48] Haas, J.-F. and Sturtevant, B., "Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities," *Journal of Fluid Mechanics*, vol. 181, pp. 41–76, 1987.

[49] Harlow, F. and Amsden, A., *Fluid Dynamics*. No. LA-4700 in Monograph, Los Alamos, NM.: Los Alamos National Laboratory, 1971.

[50] Harstad, K. G., Miller, R. S., and Bellan, J., "Efficient high-pressure state equations," *AIChE Journal*, vol. 43, pp. 1605–1610, June 1997.

[51] Harten, A., "The artificial compression method for computation of shocks and contact discontinuities. iii. self-adjusting hybrid schemes.," *Math. of Comp.*, vol. 32, no. 142, p. 363389, 1987.

[52] Harten, A., "ENO schemes with sub-cell resolution," *Journal of Computational Physics*, vol. 83, pp. 148–184, 1989.

[53] Harten, A., "ENO schemes with sub-cell resolution," Tech. Rep. 87-56, ICASE Report No. 87-56, August 1987.

[54] Hirschfelder, J. O. and Curtiss, C. F., *Molecular Theory of Gases and Liquids*. John Wiley & Sons, Inc., 1954.

[55] Hu, C. and Shu, C.-W., "A discontinuous Galerkin Finite Element method for Hamilton-Jacobi equations," *SIAM Journal of Scientific Computing*, vol. 21, no. 2, pp. 666–690, 1999.

[56] Hu, F. Q. and Atkins, H. L., "A discrete analysis of non-reflecting boundary conditions for discontinuous Galerking method," in *9th AIAA/CEAS Aeroacoustics Conference and Exhibit*, no. AIAA-2003-3301, January 2003.

[57] Hughes, T., Mazzei, L., and Oberai, A., "The multiscale formulation of Large-Eddy Simulation: Decay of homogeneous isotropic turbulence," *Physics of Fluids*, vol. 13, no. 2, pp. 505–512, 2001.

[58] Hughes, T., Oberai, A., and Mazzei, L., "Large Eddy Simulation of turbulent channel flows by variational multiscale method," *Physics of Fluids*, vol. 13, no. 6, pp. 1784–1799, 2001.

[59] Israeli, M. and Orszag, S. A., "Approximation of radiation boundary conditions," *Journal of Computational Physics*, vol. 41, pp. 115–135, 1981.

[60] Jacqmin, D., "An energy approach to the continuum surface tension method," in *34th Aerospace Sciences Meeting and Exhibit*, no. AIAA-96-0858, (Reno, NV), January 1996.

[61] Jacqmin, D., "Calculation of two-phase Navier-Stokes flows using phase-field modeling," *Journal of Computational Physics*, vol. 155, pp. 96–127, 1999.

[62] Jaffre, J., Johnson, C., and Szepessy, A., "Convergence of the discontinuous Galerkin Finite Element method for hyperbolic conservation laws," *Mathematical Models and Methods in Applied Sciences*, vol. 5, pp. 367–386, 1995.

[63] Jamet, D., Lebaigue, O., Coutris, N., and Delhaye, J. M., "The second gradient method for the direct numerical simulation of liquid-vapor flows with phase change," *Journal of Computational Physics*, vol. 169, pp. 624–651, 2001.

[64] Jeon, B., Kress, J. D., Collins, L. A., and Grønbech-Jensen, N., "Parallel tree code for two-component ultracold plasma analysis," *Computer Physics Communications*, vol. 178, pp. 272–279, 2008.

[65] Jiang, G. and W.Shu, C., "Efficient implementation of weighted ENO schemes," *Journal of Computational Physics*, vol. 126, pp. 202–228, 1996.

[66] Johnsen, E. and Colonius, T., "Implementation of WENO schemes in compressible multicomponent flow problems," *Journal of Computational Physics*, vol. 219, pp. 715–732, 2006.

[67] Johnsen, E., Larsson, J., Bhagatwala, A. V., Cabot, W. H., Moin, P., Olson, B. J., Rawat, P. S., Shankar, S. K., Sjgreen, B., Yee, H. C., Zhong, X., and Lele, S. K., "Assessment of high-resolution methods for numerical simulations of compressible turbulence with shock waves," *Journal of Computational Physics*, vol. 229, pp. 1213–1237, 2010.

[68] Kemenov, K. and Menon, S., "Explicit small-scale velocity simulation for high-re turbulent flows," *Journal of Computational Physics*, vol. 220, pp. 290–311, 2006.

[69] Khokhlov, A. M., "Fully threaded tree algorithms for adaptive refinement fluid dynamics simulations," *Journal of Computational Physics*, vol. 143, pp. 519–543, 1998.

[70] Kokh, S. and Allaire, G., "Numerical simulation of 2-d two-phase flows with interface," in *Godunov Methods: Theory and Applications* (Toro, E. F., ed.), pp. 513–518, Kluwer Academic/Plenum Publishers, 2001.

[71] Kokh, S. and Lagoutiére, F., "An anti-diffusive numerical scheme for the simulation of interfaces between compressible fluids by means of a five-equation model," *Journal of Computational Physics*, vol. 229, pp. 2773–2809, 2010.

[72] Krivodonova, L., "Limiters for high-order discontinuous Galerkin methods," *Journal of Computational Physics*, vol. 226, pp. 879–896, 2007.

[73] Krivodonova, L. and Berger, M., "High-order accurate implementation of solid wall boundary conditions in curved geometries," *Journal of Computational Physics*, vol. 211, pp. 492–512, 2006.

[74] Krivodonova, L., Xin, J., Remacle, J.-F., Chevaugeon, N., and Flaherty, J., "Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws," *Applied Numerical Mathematics*, vol. 48, pp. 323–338, 2004.

[75] Laval, J., Dubrule, B., and Nazarenko, S., "Nonlocality of interaction of scales in the dynamics of 2d incompressible fluids," *Phys. Rev. Let.*, vol. 83, pp. 4061–4064, 1999.

[76] Laval, J., Dubrule, B., and Nazarenko, S., "Nonlocality and intermittency in three-dimensional turbulence," *Physics of Fluids*, vol. 13, pp. 1995–2012, 2001.

[77] Layes, G. and Métayer, O. L., "Quantitative numerical and experimental studies of the shock accelerated heterogeneous bubbles motion," *Physics of Fluids*, vol. 19, p. 042105, 2007.

[78] Leicht, T. and Hartmann, R., "Error estimation and anisotropic mesh refinement for 3d laminar aerodynamic flow simulations," *Journal of Computational Physics*, vol. 229, pp. 7344–7360, 2010.

[79] Lepsky, O., Hu, C., and Shu, C.-W., "Analysis of the discontinuous Galerkin method for Hamilton-Jacobi equations," *Applied Numerical Mathematics*, vol. 33, pp. 423–434, 2000.

[80] Lesser, M. B. and Field, J. E., "The impact of compressible liquids," *Ann. Rev. Fluid Mech.*, vol. 15, pp. 97–122, 1983.

[81] Levin, J. G., Iskandarani, M., and Haidvogel, D. B., "A spectral filtering procedure for eddy-resolving simulations with a Spectral Element ocean model," *Journal of Computational Physics*, vol. 137, pp. 130–154, 1997.

[82] Li, F. and Shu, C.-W., "Reinterpretation and simplified implementation of a discontinuous Galerkin method for Hamilton-Jacobi equations," *Applied Mathematics Letters*, vol. 18, pp. 1204–1209, 2005.

[83] Li, X., Soteriou, M. C., Arienti, M., and Sussman, M. M., "High-fidelity simulation of atomization and evaporation in a liquid jet in cross-flow," in *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, no. AIAA-2011-99, (Orlando, FL), January 2011.

[84] Liu, Y., Shu, C.-W., and Zhang, M., "Strong stability preserving property of the deferred correction time discretization," *Journal of Computational Mathematics*, vol. 26, no. 5, pp. 633–656, 2008.

[85] Liu, Y.-J., Shu, C.-W., , and Xu, Z., "Hierarchical reconstruction with up to second degree remainder for solving nonlinear conservation laws," *Nonlinearity*, vol. 22, pp. 2799–2812, 2009.

[86] Liu, Y.-J., Shu, C.-W., Tadmor, E., and Zhang, M.-P., "Central discontinuous Galerkin methods on overlapping cells with a non-oscillatory hierarchical reconstruction," *SIAM Journal of Numerical Analysis*, vol. 45, pp. 2442–2467, 2007.

[87] Lodato, G., Domingo, P., and Vervisch, L., "Three-dimensional boundary conditions for direct and Large-Eddy Simulation of compressible viscous flows," *Journal of Computational Physics*, vol. 227, pp. 5105–5143, 2008.

[88] Marchandise, E., Geuzaine, P., Chevaugeon, N., and Remacle, J.-F., "A stabilized Finite Element method using a discontinuous level set approach for the computation of bubble dynamics," *Journal of Computational Physics*, vol. 225, pp. 949–974, 2007.

[89] Marchandise, E. and Remacle, J.-F., "A stabilised Finite Element method using a discontinuous level set approach for solving two phase incompressible flows," *Journal of Computational Physics*, vol. 219, pp. 780–800, 2006.

[90] Marchandise, E., Remacle, J.-F., and Chevaugeon, N., "A quadrature free discontinuous Galerkin method for the level set equation," *Journal of Computational Physics*, vol. 212, pp. 338–357, 2006.

[91] Massoni, J., Saurel, R., Nkonga, B., and Abgrall, R., "Some models and Eulerian methods for interface problems between compressible fluids with heat transfer," *International Journal of Heat and Mass Transfer*, vol. 45, pp. 1287–1307, 2002.

[92] Menard, T., Tanguy, S., and Berlemont, A., "Coupling level set/VOF/ghost fluid methods: Validation and application to 3d simulation of the primary break-up of a liquid jet," *International Journal of Multiphase Flow*, vol. 33, pp. 510–524, 2007.

[93] Minion, M., "Semi-implicit spectral deferred correction methods for ordinary differential equations," *Commun. Math. Sci.*, vol. 1, pp. 471–500, 2003.

[94] Moureau, V., Lartigue, G., Sommerer, Y., Angelberger, C., Colin, O., and Poinsot, T., "Numerical methods for unsteady compressible multi-component reacting flows on fixed and moving grids," *Journal of computational physics*, vol. 202, pp. 710–736, 2005.

[95] Nordstrom, J., Eriksson, S., and Eliasson, P., "Weak and strong wall boundary procedures and convergence to steady-state of the NavierStokes equations," *Journal of Computational Physics*, vol. 231, pp. 4867–4884, 2012.

[96] Osher, S. and Sethian, J. A., "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, pp. 12–49, 1988.

[97] Patel, N., *Simulation of hydrodynamic fragmentation from a fundamental and an engineering perspective*. PhD thesis, Georgia Institute of Technology, August 2007.

[98] Peng, D. Y. and Robinson, D. B., "A new two-constant equation of state," *Ind. Eng. Chem. Fundam.*, vol. 15, no. 1, pp. 59–64, 1976.

[99] Peraire, J. and Persson, P.-O., "The compact discontinuous Galerkin (cdg) method for elliptic problems," *SIAM Journal of Scientific Computing*, vol. 30, no. 4, pp. 1806–1824, 2008.

[100] Perigaud, G. and Saurel, R., "A compressible flow model with capillary effects," *Journal of Computational Physics*, vol. 209, pp. 139–178, 2005.

[101] Persson, P.-O. and Peraire, J., "Sub-cell shock capturing for discontinuous Galerkin methods," in *44th AIAA Aerospace Sciences Meeting and Exhibit*, no. 112 in 2006, (Reno, Nevada), AIAA, January 2006.

[102] Petitpas, F., Massoni, J., Saurel, R., Lapebie, E., and Munier, L., "Diffuse interface model for high speed cavitating underwater systems," *International Journal of Multiphase Flow*, vol. 35, pp. 747–759, 2009.

[103] Poinsot, T. and Lele, S., "Boundary conditions for direct simulations of compressible viscous flows," *Journal of computational physics*, vol. 101, pp. 104–129, 1992.

[104] Pope, S., *Turbulent Flows*. Cambridge Univeristy Press, 2000.

[105] Qian, J. and Law, C. K., "Regimes of coalescence and separation in droplet collision," *Journal of Fluid Mechanics*, vol. 331, pp. 59–80, 1997.

[106] Qiu, J. and Shu, C.-W., "A comparison of troubled-cell indicators for Runge-Kutta discontinuous Galerkin methods using weighted essentially nonoscillatory limiters," *SIAM Journal of Scientific Computing*, vol. 27, no. 3, pp. 995–1013, 2005.

[107] Qiu, J. and Shu, C.-W., "Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method ii: Two dimensional case," *Computers and Fluids*, vol. 34, pp. 642–663, 2005.

[108] Qiu, J. and Shu, C.-W., "Runge-Kutta discontinuous Galerkin methods using WENO limiters," *SIAM Journal of Scientific Computing*, vol. 26, no. 3, pp. 907–929, 2005.

[109] Quirk, J. and Karni, S., "On the dynamics of a shock-bubble interaction," *Journal of Fluid Mechanics*, vol. 318, pp. 129–163, 1996.

[110] Rault, A., Chiavassa, G., and Donat, R., "Shock-vortex interactions at high Mach numbers," *Journal of Scientific Computing*, vol. 19, no. 1-3, p. 347, 2003.

[111] Reed, W. and Hill, T., "Triangular mesh methods for the neutron transport equation.," Tech. Rep. LA-UR-73-479, Los Alamos National Labs, NM, 1973.

[112] Remacle, J.-F., Flaherty, J., and Shephard, M., "An adaptive discontinuous Galerkin technique with an orthogonal basis applied to Rayleigh-Taylor flow instabilities," *SIAM Review*, vol. 45, pp. 53–72, 2003.

[113] Remacle, J.-F., Li, X., Shephard, M. S., and Flaherty, J. E., "Anisotropic adaptive simulation of transient flows using discontinuous Galerkin methods," *International Journal for Numerical Methods in Engineering*, vol. 62, pp. 899–923, 2005.

[114] Renardy, Y. and Renardy, M., "Prost: a parabolic reconstruction of surface tension for the volume-of-fluid method," *Journal of Computational Physics*, vol. 183, pp. 400–421, 2002.

[115] Ristorcelli, J. R. and Blaisdell, G. A., "Consistent initial conditions for the dns of compressible turbulence," *Physics of Fluids*, vol. 9, no. 1, pp. 4–6, 1997.

[116] Saurel, R., Petitpas, F., and Berry, R. A., "Simple and efficient relaxation methods for interfaces separating compressible fluids, cavitating flows and shocks in multiphase mixtures," *Journal of Computational Physics*, vol. 228, pp. 1678–1712, 2009.

[117] Saurel, R. and Abrall, R., "A multiphase Godunov method for compressible multifluid and multiphase flows," *Journal of Computational Physics*, vol. 150, pp. 425–467, 1999.

[118] Saurel, R. and LeMetayer, O., "A multiphase model for compressible flows with interfaces, shocks, detonation waves and cavitation," *Journal of Fluid Mechanics*, vol. 431, pp. 239–271, 2001.

[119] Sengupta, K., Jacobs, G. B., and Mashayek, F., "Large-Eddy Simulation of compressible flows using a spectral multidomain method," *International Journal for Numerical Methods in Fluids*, vol. 61, pp. 311–340, 2009.

[120] Sengupta, K. and Mashayek, F., "Large-Eddy Simulation using a discontinuous Galerkin Spectral Element method," in *45th AIAA Aerospace Sciences Meeting and Exhibit*, no. AIAA-2007-402, (Reno, Nevada), January 2007.

[121] Shelton, A., *A Multi-Resolution Discontinuous Galerkin method for unsteady compressible flows*. PhD thesis, Georgia Institute of Technology, 2008.

[122] Shu, C.-W. and Osher, S., "Efficient implementation of Essentially Non-Oscillatory shock-capturing schemes, II," *Journal of Computational Physics*, vol. 83, pp. 32–78, 1989.

[123] Shukla, R., Pantano, C., and Freund, J. B., "An interface capturing method for the simulation of multi-phase compressible flows," *Journal of Computational Physics*, vol. 229, pp. 7411–7439, 2010.

[124] Shyue, K.-M., "An efficient shock-capturing algorithm for compressible multicomponent problems," *Journal of Computational Physics*, vol. 142, pp. 208–242, 1998.

[125] Shyue, K.-M., "A fluid-mixture type algorithm for compressible multicomponent flow with mie-gruneisen equation of state," *Journal of Computational Physics*, vol. 171, pp. 678–707, 2001.

[126] Shyue, K., "A volume-fraction based algorithm for hybrid barotropic and non-barotropic two-fluid flow problems," *Shock Waves*, vol. 15, pp. 407–423, 2006.

[127] So, K. K., Hu, X. Y., and Adams, N. A., "Anti-diffusion interface sharpening technique for two-phase compressible flow simulations," *Journal of Computational Physics*, vol. 231, pp. 4304–4323, 2012.

[128] Steinhoff, J. and Chitta, S., "Long range numerical simulation of short waves as nonlinear solitary waves," *Mathematics and Computers in Simulation*, vol. 80, pp. 752–762, 2009.

[129] Sutherland, J. and Kennedy, C., "Improved boundary conditions for viscous, reacting, compressible flows," *Journal of Computational Physics*, vol. 191, pp. 502–524, 2003.

[130] Swantek, A. and Austin, J., "Collapse of void arrays under stress wave loading," *Journal of Fluid Mechanics*, vol. 649, pp. 399–427, 2010.

[131] Tabakova, S., Feuillebois, F., Mongruel, A., Daru, V., and Radev, S., "First stages of drop impact on a dry surface: asymptotic model," *Z. Angew. Math. Phys.*, vol. 63, pp. 313–330, 2012.

[132] Teigen, K. E., Song, P., Lowengrub, J., and Voigt, A., "A diffuse-interface method for two-phase flows with soluble surfactants," *Journal of computational physics*, vol. 230, pp. 375–393, 2011.

[133] Tennekes, H. and Lumley, J. L., *A first curse in turbulence.* MIT Press, 1972.

[134] Thompson, K. W., "Time-dependent boundary conditions," *Journal of Computational Physics*, vol. 89, pp. 439–461, 1990.

[135] Toro, E. E., *Riemann Solvers and Numerical Methods for Fluid Dynamics.* Springer, 1999.

[136] Toulopoulos, I. and Ekaterinaris, J. A., "Artificial boundary conditions for the numerical solution of the Euler equations by the discontinuous Galerkin method," *Journal of Computational Physics*, vol. 230, pp. 5974–5995, 2011.

[137] Tryggvason, G., Scrdovelli, R., and Zaleski, S., *Direct Numerical Simulations of Gas-Liquid Multiphase Flows.* Cambdridge University Press, 2011.

[138] Unverdi, S. O. and Tryggvason, G., "A front tracking method for viscous, incompressible, multifluid flows," *Journal of Computational Physics*, vol. 100, p. 25, 1992.

[139] van Leer, B. and Nomura, S., "Discontinuous Galerkin for diffusion," in *17th AIAA Computational Fluid Dynamics Conference*, no. AIAA-2005-5108, (Toronto, Ontario Canada), AIAA, June 2005.

[140] Wang, C. and Shu, C. W., "An interface treating technique for compressible multi-medium flow with Runge-Kutta discontinuous Galerkin method," *Journal of Computational Physics*, vol. 229, pp. 8823–8843, 2010.

[141] Wang, C., Zhang, X., Shu, C. W., and Ning, J., "Robust high order discontinuous Galerkin schemes for two-dimensional gaseous detonations," *Journal of Computational Physics*, vol. 231, pp. 653–665, 2012.

[142] Wang, Z., Yang, J., Koo, B., and Stern, F., "A coupled level set and volume-of-fluid method for sharp interface simulation of plunging breaking waves," *International Journal of Multiphase Flow*, vol. 35, pp. 227–246, 2009.

[143] Wong, D. S. H. and Sandier, S. I., "A theoretically correct mixing rule for cubic equations of state," *AIChE Journal*, vol. 38, pp. 671–680, May 1992.

[144] Woordward, P. and Colella, P., "The numerical simulation of two-dimensional fluid flow with strong shocks," *Journal of Computational Physics*, vol. 54, pp. 115–173, 1984.

[145] Xia, Y., Xu, Y., and Shu, C.-W., "Efficient time discretization for local discontinuous Galerkin methods," *Discrete and Continuous Dynamical Systems*, vol. 8, pp. 677–693, 2007.

[146] Xin, J. and Flaherty, J. E., "Viscous stabilization of discontinuous Galerkin solutions of hyperbolic conservation laws," *Applied Numerical Mathematics*, vol. 56, pp. 444–458, 2006.

[147] Xu, L., Zhang, W. W., and Nagel, S. R., "Drop splashing on a dry smooth surface," *Physical Review Letters*, vol. 94, p. 184505, 2005.

[148] Xu, Y. and Shu, C.-W., "Local discontinuous Galerkin methods for high-order time-dependent partial differential equations," *Communications in Computational Physics*, vol. 7, pp. 1–46, 2010.

[149] Xu, Z., Liu, Y., and Shu, C.-W., "Hierarchical reconstruction for discontinuous Galerkin methods on unstructured grids with a WENO type linear reconstruction with partial neighboring cell," *Journal of Computational Physics*, vol. 228, pp. 2194–2212, 2009.

[150] Xu, Z. and Shu, C., "Anti-diffusive flux corrections for high order finite difference WENO schemes," *Journal of Computational Physics*, vol. 205, pp. 458–485, 2005.

[151] Yan, J. and Osher, S., "A local discontinuous Galerkin method for directly solving Hamilton-Jacobi equations," *Journal of Computational Physics*, vol. 230, pp. 232–244, 2011.

[152] Yan, J. and Shu, C. W., "Local discontinuous Galerkin methods for partial differential equations with higher order derivatives," *Journal of Scientific Computing*, vol. 17, no. 1-4, pp. 27–47, 2002.

[153] Yang, H., "An artificial compression method for ENO schemes: The slope modification method," *Journal of Computational Physics*, vol. 89, pp. 125–160, 1990.

[154] Yang, M. and Wang, Z. J., "A parameter-free generalized moment limiter for high-order methods on unstructured grids," in *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, no. 605, (Orlando, Florida), AIAA, January 2009.

[155] Yeung, P. K. and Pope, S. B., "An algorithm for tracking fluid particles in numerical simulations of homogeneous turbulent," *Journal of Computational Physics*, vol. 79, pp. 373–416, 1988.

[156] Yoo, C. and Im, H., "Characteristic boundary conditions for simulations of compressible reacting flows with multi-dimensional, viscous and reaction effects," *Combustion Theory and Modelling*, vol. 11, pp. 259–286, April 2007.

[157] Yoo, C., Wang, Y., Trouvé, A., and Im, H., "Characteristic boundary conditions for direct simulations of turbulent counterflow flames," *Combustion Theory and Modelling*, vol. 9, pp. 617–646, November 2005.

[158] Yuan, L. and Shu, C.-W., "Discontinuous Galerkin method based on non-polynomial approximation spaces," *Journal of Computational Physics*, vol. 218, pp. 295–323, 2006.

[159] Zhang, X. and Shu, C., "On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes," *Journal of Computational Physics*, vol. 229, pp. 8918–8934, 2010.

[160] Zheng, H., Shu, C., and Chew, Y., "An object-oriented and quadrilateral-mesh based solution adaptive algorithm for compressible multi-fluid flows," *Journal of Computational Physics*, vol. 227, pp. 6895–6921, 2008.

[161] Zhou, T., Li, Y., and Shu, C., "Numerical comparison of WENO finite volume and Runge-Kutta discontinuous Galerkin methods," *Journal of Scientific Computing*, vol. 16, no. 2, pp. 145–171, 2000.

[162] Zhu, H. and Qiu, J., "Adaptive Runge-Kutta discontinuous Galerkin methods using different indicators: One-dimensional case," *Journal of Computational Physics*, vol. 228, pp. 6957–6976, 2009.

[163] Zinn, B., Lubarsky, E., Bibik, O., Shcherbik, D., and Bennewitz, J., "Liquid jet atomization study," Tech. Rep. 1606D57, School of Aerospace Engineering, Georgia Institute of Technology, 2011.