THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (Toulouse INP)

Discipline ou spécialité :

Dynamique des fluides

Présentée et soutenue par :

M. SOUMYO SENGUPTA

le mardi 22 novembre 2022

Titre :

Méthodes avancées pour les maillages dans le calcul haute performance des simulations d'explosion

---

Ecole doctorale :

Mécanique, Energétique, Génie civil, Procédés (MEGeP)

Unité de recherche :

Centre Européen de Recherche et Formation Avancées en Calcul Scientifique (CERFACS)

Directeur(s) de Thèse :

M. LAURENT GICQUEL

M. GABRIEL STAFFELBACH

Rapporteurs :

M. JENS MUELLER, QUEEN MARY UNIVERSITY OF LONDON

M. VINCENT MOUREAU, TURBOMECA

Membre(s) du jury :

MME NABIHA CHAUMEIX, CNRS ORLEANS, Président

M. ANDREAS KEMPF, , Membre

M. GABRIEL STAFFELBACH, CERFACS, Invité(e)

M. LAURENT GICQUEL, CERFACS, Membre

Abstract

Tremendous improvements have been made in numerical simulation using LES (Large Eddy Simulation) over the past thirty years. This was possible with the introduction of more robust numerical methods, improved boundary condition modeling, and more accurate, detailed chemistries in the field of combustion. High-performance computing is a key factor so that these simulations can be performed in a reasonable time frame and use realistic large-scale representative geometries. Even with the advances in all of these mentioned fields, the bottleneck for solving these problems still remains the initial mesh resolution/quality. The focus of this thesis is to first understand why the mesh resolution is important through global stability analysis and then to alleviate this problem by using run-time dynamic mesh adaptation for combustion related explosion problems.

The first part of the manuscript deals with global stability analysis (GSA) of the linear convection-diffusion equation (LCDE) and linear convection-diffusion-reaction equation (LCDRE) for non-reacting and reacting flows respectively. This analysis shows the importance of the non-dimensional parameters such as CFL number N_c, Peclet number Pe and Damkohler number, Da on stability, dispersive and diffusive nature of the numerical scheme chosen (Lax-Wendroff and TTGC schemes). Through the analysis, the importance of mesh resolution to obtain an accurate, stable solution for any numerical problem is highlighted. Especially, when solving realistic reacting flow problems, it is of paramount importance to resolve the flame front adequately to obtain accurate solutions. To overcome this issue, when solving realistic large-scale reacting flow simulations it is useful to use run-time dynamic mesh refinement for accuracy and cost benefits.

In the second part of the manuscript, two different dynamic mesh adaptation techniques are explained. Although using a similar generic algorithm, the differences between the two techniques are detailed. Several test cases are simulated to validate the adaptation techniques. An appropriate quantity of interest (QOI) is chosen according to the case studied. Using this quantity of interest, two large-scale, compressible, and turbulent reacting flow test cases are simulated using dynamic mesh adaptation with the goal of obtaining the same accuracy while obtaining performance benefits.

Résumé

Des améliorations considérables ont été apportées à la simulation numérique par LES (Large Eddy Simulation) au cours des trente dernières années. Cela a été possible grâce à l'introduction de méthodes numériques plus robustes, à l'amélioration de la modélisation des conditions aux limites et à l'utilisation de produits chimiques plus précis et détaillés dans le domaine de la combustion. Le calcul haute performance est un facteur clé Le calcul haute performance est un facteur clé pour que ces simulations puissent être réalisées dans un délai raisonnable et utiliser des géométries représentatives réalistes à grande échelle. Malgré les avancées dans tous ces domaines, le goulot d'étranglement pour la résolution de ces problèmes reste la résolution/qualité du maillage initial. L'objectif de cette thèse est d'abord de comprendre pourquoi la résolution du maillage est importante grâce à l'analyse de la stabilité globale, puis d'atténuer ce problème en utilisant l'adaptation dynamique du maillage pour les problèmes d'explosion liés à la combustion.

La première partie du manuscrit traite de l'analyse de stabilité globale (GSA) de l'équation de convection-diffusion linéaire (LCDE) et de l'équation de convection-diffusion-réaction linéaire (LCDRE) pour les écoulements sans réaction et avec réaction respectivement. Cette analyse montre l'importance des paramètres non dimensionnels tels que le nombre CFL N_c, le nombre de Peclet Pe et le nombre de Damkohler, Da sur la stabilité, la nature dispersive et diffusive du schéma numérique choisi (schémas Lax-Wendroff et TTGC). L'analyse met en évidence l'importance de la résolution du maillage pour obtenir une solution précise et stable pour tout problème numérique. En particulier, lors de la résolution de problèmes réalistes d'écoulement réactif, il est primordial de résoudre le front de flamme de manière adéquate pour obtenir des solutions précises. Pour surmonter ce problème, lors de la résolution de simulations réalistes d'écoulement réactif à grande échelle, il est utile d'utiliser le raffinement dynamique du maillage en cours d'exécution pour des raisons de précision et de coût.

Dans la deuxième partie du manuscrit, deux techniques différentes d'adaptation dynamique du maillage sont expliquées. Bien qu'utilisant un algorithme générique similaire, les différences entre les deux techniques sont détaillées. Plusieurs cas de test sont simulés pour valider les techniques d'adaptation. Une quantité d'intérêt (QOI) appropriée est choisie en fonction du cas étudié. À l'aide de cette quantité d'intérêt, deux cas d'essai d'écoulement réactif à grande échelle, compressible et turbulent sont simulés en utilisant l'adaptation dynamique du maillage dans le

but d'obtenir la même précision tout en obtenant des avantages en termes de performance.

Acknowledgements

Firstly, thanks to the reviewers Jens Mueller and Vincent Moureau who have agreed to read the thesis and offer their comments. These comments have been vital in improving the quality of the manuscript. I would also like to thank Nabiha Chaumeix and Andreas Kempf who agreed to be a part of the jury and for their insightful comments and questions on the day of the defense. Overall, a big thank you to all of the external members to make the culmination of this thesis enjoyable and for me personally, a day to remember.

This work would have been incomplete if not for the support, patience and encouragement shown by my advisors Gabriel Staffelbach and Laurent Gicquel. Thank you for being patient, giving me space to try new things and for encouraging me throughout the writing phase. I sincerely thank you both from the bottom of heart and without your support and guidance it would have been impossible.

I would also like to thank Thierry Poinsot who saw someone who is capable to work at CERFACS and gave me the chance to explore this interesting research topic. I thank him for all the ideas and comment during the various meetings of LEFEX that we had. Also a big thanks to Pavanakumar Mohanamuraly who has been like an elder brother who has guided and gave wonderful ideas throughout my stay at CERFACS.

Finally a big thank you to all the colleagues, seniors, support staff at CERFACS without whose constant support this work would not have reached this stage. Special thanks to the CSG team who have constantly and selflessly helped all the students at CERFACS as we had problems. Special thanks to Quentin, Olivier and Thomas for their wonderful discussions which was useful to complete this work. Also thanks to many friends I have made during this journey - Abhijeet, Varun, Shubham, Sreejith, Sriram, Pavan, Suman, Victor and many more. I am sorry If I have forgotten anyone. Also special thanks to Michele, Brigitte and Lydia who support and help has been crucial for a peaceful stay in France.

The biggest thanks is reserved for my family (maa, baba and my sister Aditi) back at home who have been a pillar of constant support through COVID and thesis writing and the entire PhD process. Thank you for understanding when I am not in the best of moods and for tolerating my behavior. I couldn't have completed this big achievement in my life without all of you

Contents

CHAPTER 1

Introduction

Contents

1.1 Background

Combustion is the process involving chemical reactions between two or more species accompanied by the emission of energy mostly in the form of heat. It has been the primary source of energy for mankind and the basic use of combustion in the olden days namely, lighting and heating, were superseded by its use in harnessing mechanical power after the industrial revolution in the 18th century. One of the first prime movers, the steam engine, used coal as fuel to burn and produced energy which was used to vaporize water to steam and produce power.

Even after two and a half centuries since the industrial revolution, our dependence on combustion as the main source of energy has not changed. Fig. 1.1 shows the historical and forecasted energy production (in QuadrillionBTUs) by various fuel sources in the United States as documented by Energy Information

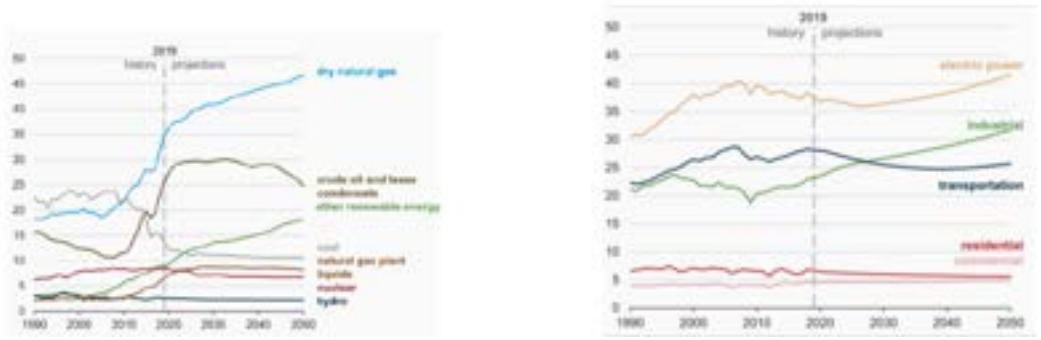


Figure 1.1: Historical and projected energy production in United States by fuel in BTUs and Historical and projected energy consumption in United States by sector in Quadrillion BTUs. [1]

Administration in its 2020 report [1]. Even though energy production from renewable sources is gaining momentum, the technology has not still evolved to fully meet the large global energy demand. While energy production from crude oil, nuclear and hydropower plants is expected to stabilize in the future, natural gas is expected to be the prime source of energy in the future owing to lower CO_2 emission, low cost and wide availability. Even though the long-term impact of Covid-19 on global energy demands is beyond comprehension at this point, the trends indicate that non-renewable sources of energy shall continue to dominate the global energy market. The field of combustion has varied uses majorly in the following fields:

1.1.1 Transport Industry

Gas turbines dominate aviation propulsion, and while they have traditionally always been considered to be voracious fuel consumers, the demands of modern aviation have resulted in a much more constrained fuel tolerance. Aviation gas turbine engines are very efficient machines for converting stored chemical energy into jet fuel. In contrast to ground transportation vehicles, aircraft-specific fuel consumption has continually decreased (improved) since commercial aviation converted to the gas turbine engine in the 1960s. Aviation gas turbines must operate over a wide range of ambient conditions while increasing efficiency and meeting specific emission targets. Many different types of essential studies are performed on gas turbine engines for the aviation industry. Studies have been performed to ignite or reignite a gas turbine engine in a cold and rarefied atmosphere which is a critical issue for many manufacturers [2]. In these studies, many factors affect the results like the influence of spark energy [3, 4, 5, 6], mean flow at the spark location [7, 8], detailed kinetics [9, 10, 11] and pressure [12]. Significant progress

in developing alternative (non-petroleum) fuels for aviation is an ongoing field of importance [13, 14].

Another important field of study for gas turbine engines is thermoacoustic instabilities [15]. Gas turbines work in lean operating conditions because of environmental restrictions. Operating at these conditions will increase the possibility of self-sustained oscillations and dynamic instabilities within the combustion system due to pressure and temperature fluctuations. The dynamic instabilities refer to thermoacoustics which is triggered when the heat release from the combustion is coupled with the acoustic pressure fluctuation leading to high sound pressure level (SPL) [16, 17]. These kinds of pressure oscillations could damage the combustor's internal parts. In some cases, small pieces of the combustor could break apart and travel with the exhaust gases through the exhaust section and may damage the gas turbine components [18]. Modern gas turbine combustors are still under development to provide robustness, low emissions, and fuel-efficient requirements.

Internal combustion (IC) engines operating on fossil fuel oil provide about 25 percent of the world's power. The transportation of people and goods is required in modern society. Currently transport is almost entirely powered by ICEs [19] using liquid fuels as it is readily available, convenient, and affordable. There are still no natural alternatives that can compete with the IC engine over the entire range of applications that they cover, and IC engines are still undergoing further improvement. It is possible that future IC engines will likely be characterized by a mix of solutions, involving battery electric and hybrid electric vehicles (BEV and HEV), fuel cell electric vehicles (FCEVs) and conventional vehicles. The combustion engine will still play a central role, whether used for power generation or for powering the vehicle itself. As a result, there is great interest in improving the thermal efficiency of IC engines without significant increases in the purchase and operating costs. In the medium-to-long term, there is scope for improving engines by co-designing fuel/engine systems for optimal performance. Single- and dual-fuel technologies, such as homogeneous charge compression ignition (HCCI) [20], premixed controlled compression ignition (PCCI) [21], and reactivity controlled compression ignition (RCCI) [22] offer promise for improving efficiency and reducing exhaust emissions.

1.1.2 Energy

A thermal power station is a power station in which heat energy is converted to electricity. Typically, fuel is used to boil water in a large pressure vessel

to produce high-pressure steam, which drives a steam turbine connected to an electrical generator. The low-pressure exhaust from the turbine passes through a steam condenser and is recycled to where it was heated, known as a Rankine cycle. Similarly, Natural gas can also be burnt directly in a gas turbine connected to a generator.

The combustion (gas) turbines being installed in many of today's natural-gas-fueled power plants are complex machines which involve three main sections: (i) The compressor that draws air into the engine, pressurizes it, and feeds it to the combustion chamber. (ii) The combustion system is typically made up of a ring of fuel injectors that inject a steady stream of fuel into combustion chambers where it mixes with the air. The combustion produces a high-temperature, high-pressure gas stream that enters into the turbine section. (iii) The turbine is an array of alternate stationery and rotating airfoil-section blades. As hot combustion gas expands through the turbine, it spins the rotating blades. The rotating blades perform a dual function: they drive the compressor to draw pressurized air into the combustion section, and they spin a generator to produce electricity. Many studies have been performed to understand the effect of ambient temperature and operation conditions (compression ratio, turbine inlet temperature, air to fuel ratio, and efficiency of compressor and turbine) on the performance of gas turbine power plant [23, 24, 25].

1.1.3 Chemical Engineering

Olefins are the basic building blocks of the chemical industry, and their derivatives are used as raw materials in manufacturing products used in our day-to-day life. Steam cracking is the principal process of producing light olefins. In the United States, 70 percent of ethylene is produced by steam cracking naphtha and the remaining 30 percent is produced by cracking ethane [26]. In this method, lighter hydrocarbon feed stocks such as ethane, propane and even higher hydrocarbons such as naphtha and gas oil are heated at high temperature and at specified pressure for a stipulated amount of time to produce light olefins such as ethylene and propylene. Steam cracking also produces the least amount of CO_2 per ton of ethylene produced [27]. Steam cracking is expected to continue to be the primary technology for olefin production at least for the coming couple of decades.

Different designs are used based on their NO_x emissions and can be classified as follows

1. Conventional burner - burner designs were aimed at creating quick mixing of fuel and air and, thereby, a short and compact flame. NO_x, CO and

Unburnt Hydrocarbons (UHC) emissions and their control did not play any role in the design process.

2. Low NO_x burner - burner designs were aimed at producing lesser NO_x emissions. NO_x reduction is achieved by forcing the combustion to occur under lean or rich conditions so that the flame temperature and thermal NO_x is low.

3. Ultra-Low NO_x burner - NO_x can be further reduced by mixing the air used for combustion with flue gas from the burner. Flue gases reduce NOx emissions in two ways. Firstly, mixing with flue gas reduces the O_2 concentration in the air available for combustion, thereby reducing NO_x production. Secondly, after losing heat to the process gas, flue gas temperature is lower than the flame temperature.

4. Lean premix burner - Under rich conditions, lower NO_x is produced in both premix and non-premixed flames. However, under lean conditions, non-premixed flames produce much higher NO_x when compared to premixed flames. Hence, if fuel staging is implemented with a primary, lean premix flame followed later by a rich staged combustion zone, then the NO_x produced is even lower than Ultra-low NO_x burners.

Many design improvements are being made in the Ultra-low NO_x burners to reduce the amount of NO_x produced. One method is to improve the mixing of flue gas with fuel and air. This is implemented by incorporating a Coanda surface on the burner tiles close to the fuel injection ports. Coanda surface can entrain the fuel, air, and flue gas efficiently, thereby enhancing mixing and burning at low temperatures, producing lower NO_x. The other method to reduce NO_x is to increase the flame surface area to enhance mixing. This is incorporated by modifying the burner tile geometry to alter the flame shape and increase the flame surface area. The burner tile is divided circumferentially into multiple segments over which the secondary fuel injection occurs. The increase in flame surface area, even while keeping the flame compact, aids in enhancing fuel-air mixing and reducing NO_x.

1.1.4 Fire safety studies

A variety of studies are performed for fire safety. The burning characteristics of wood cribs in a confined space are studied in [28]. The results show that the burning conditions of wood cribs in a confined space fall into two regimes:

the smaller porosity factor burning regime and the larger porosity factor burning regime. The theoretical heat of combustion of a given product is the maximum amount of heat that a product liberates in a fire as a result of complete combustion [29]. Depending on the context of use, this parameter is also referred to as the enthalpy of combustion, the calorific, or the heating value. The theoretical (net) heat of combustion of materials is basic information necessary for fire safety studies. Three combustion models, the volumetric heat source (VHS) model, the eddy break-up model, and the presumed probability density function (prePDF) model, are examined in enclosure fire simulation [30]. The combustion models are compared and evaluated for their performance in predicting three typical enclosure fires, a room fire, a shopping mall fire, and a tunnel fire. Study for enhanced fundamental combustion understanding where buoyancy effects are eliminated, and the new challenges of fire safety considerations in non-buoyant (spacecraft) environments [31]. To predict fire spread and extinguishment using water sprays, a computational fluid dynamics (CFD) model of fire spread with the description of the CFD model, the solid pyrolysis model (wood in this case) and the coupling of these models is studied. [32].

In all of the fields of combustion mentioned above, it is essential to perform accurate and efficient numerical modeling to evaluate designs, system efficiencies and damages properly. Many of these furnaces, gas turbines, and fire safety studies are of major sizes. It is, therefore, not economically feasible to perform numerous experiments on such large-scale setups, especially for design purposes. It is difficult to place probes and measure flow data at many locations in an experimental rig. Therefore, it is essential to perform numerical simulations to obtain a better understanding of the problem at hand due to the availability of all the data in a numerical simulation. It is also economically beneficial to use numerical simulations to obtain design parameter effects for large-scale applications. Different numerical approaches are, however available today, as detailed hereafter.

1.2 Numerical Simulations

Numerical modeling involves calculating the numerical solution of governing equations describing relevant physical phenomena, subject to the appropriate (realistic) boundary and initial conditions. While earlier numerical modeling of external aerodynamic flow configurations focused on solving the linear and later the full potential flow equations, today, the CFD community depends on mainly broad

classes of numerical approximations, namely, the Reynolds Averaged Navier-Stokes Equations (RANS), Unsteady Reynolds Averaged Navier-Stokes Equations (URANS), Detached Eddy Simulation (DES), Large Eddy Simulation (LES), Direct Numerical Simulation (DNS).

RANS governing equations are obtained by taking the time average of the full Navier-Stokes Equations[33]. One then arrives at steady non-linear equations with mean flow properties as the dependent variables. The averaging procedure also gives rise to the well known closure problem in turbulence modeling due to the mathematical inability to derive closure relations for the higher moments of the non-linear terms arising from averaging. These terms are often modeled using the Boussinesq Approximation [34], or by solving transport equations for the unclosed terms (which in turn have even higher order moments). RANS provides time-averaged flow field results since only a "statistically steady" solution is seeked by solving the governing equations. As one is aware, turbulent flows possess scales of a wide spectrum depending on the Reynolds number of the flow (for incompressible, non-reacting flows). RANS does not resolve any of these scales; all the flow scales are modeled instead. This is an inherent short coming of RANS methods which limits its applicability to a narrow class of flow situations to which the model constants are designed for. The main disadvantage of this approach is that it cannot capture the most subtle effects of a turbulent flow. So, even if it is based on averaged fields, these might be completely off if there is a transitional effect strong enough to create an instability [35]. DES is a modification of a RANS model in which the the model switches to a subgrid scale formulation in regions fine enough for LES calculations [36].

DNS relies on solving the full Navier-Stokes equations without relying on any averaging or filtering [37]. However, to resolve all the scales of the flow, one would need to spatially discretize the domain to scales close to the Kolmogorov scale [38]. As the ratio of the Kolmogorov to that of the integral length scale of flow is proportional to the ratio of $Re^{9/4}$, where Re denotes the Reynolds number, very fine grid resolution is imperative. However, the advantage of this method is the possibility to gain insight into the turbulent structure [39] in a way considered as exact. This is aggravated by the requirement for computationally costly, higher-order numerical methods to reduce discretization, dispersion, and aliasing errors. While DNS of low Reynolds number flows have been performed in the past, the current computational abilities does not permit DNS of any practical reacting turbulent flows of interest.

The third approach is that of LES. LES solves the Navier-Stokes equations after a spatial filtering operation is performed on the governing equations. This

filtering operation also induces additional unclosed terms (sub-grid scale stresses) in the governing equations like RANS. The filter size should be small enough to solve approximately 80% of the turbulent kinetic energy (TKE) as stated [33]. LES is an alternative modeling approach to RANS, which solves the larger scales of turbulence responsible for the interaction with the mean flow [40] while modeling the smallest scales. However, these terms only represent the effect of flow features which are not resolved by the grid due to the lack of grid resolution. At high Re, these small scales are found to be universal in their statistical nature and hence are modeled using sub-grid scale models, while the larger flow scales are resolved.

While RANS computations are comparatively the cheapest today, they suffer from their inherent inability to predict the physics of the flow beyond a region of operating space for which the model is designed. On the other hand, DNS can provide finer details of information in the flow but is exorbitantly costly. A compromise between these two approaches is LES, where accurate flow information can be obtained while resolving the large flow structures and modeling only the smaller ones, which exhibit universal behavior in their statistics. LES has hence been increasingly used in the design and analysis of practical reacting flow configurations. As a result, we will use only LES from now on when we refer to CFD in later chapters, and there is no use of RANS or DNS in the present work.

1.3 Factors affecting accuracy and efficiency of LES

There are two key factors which affect the accuracy and efficiency of LES simulations:

1. Subgrid-scale modeling (SGS),

2. Mesh resolution

1.3.1 The sub grid scale (SGS) turbulent flux

Subgrid-scale modeling refers to the representation of critical small-scale physical processes that occur at length scales that cannot be adequately resolved on a computational mesh. In Large-Eddy Simulation, subgrid-scale (SGS) modeling is used to represent the effects of unresolved small-scale fluid motions (small eddies, swirls, vortices) on the equations governing the large-scale motions that are resolved through computer models. The formulation of physically realistic

SGS models requires an understanding of the physics and the statistics of scale interactions in hydrodynamic turbulence and is an open research question since turbulence remains an unsolved problem in classical physics.

The SGS stress tensor $\overline{\tau_{ji}^t}$ expressed as

$$\overline{\tau_{ij}}^t = -\bar{\rho}\left(\widetilde{u_i u_j} - \tilde{u}_i \tilde{u}_j\right) \tag{1.1}$$

Based on the Boussinesq assumption, this term is modeled as,

$$\overline{\tau_{ij}}^t = 2\bar{\rho}\nu_t \left(\widetilde{S}_{ij} - \frac{1}{3}\delta_{ij}\widetilde{\tau}_{ll}\right) \tag{1.2}$$

where, ν_t is the SGS turbulent viscosity, $\tilde{u}_i$ is the Favre filtered velocity vector and $\widetilde{S}_{ij}$ is the resolved strain rate tensor.

The Boussinesq approximation [34] is based on the assumption that the turbulent flux terms follow a similar function model as molecular diffusion. In addition, the approximation is also based on the hypothesis of scale separation and local equilibrium [41]. The Boussinesq approximation is widely used in most of the LES codes available today even though experimental observations contrary to it have been reported [40].

The sub grid scale heat flux is given by the term,

$$\overline{q_i}^t = \bar{\rho}\left(\widetilde{u_i E} - \tilde{u}_i \tilde{E}\right) \tag{1.3}$$

and is modeled as,

$$\overline{q_i}^t = -\lambda_t \frac{\partial \widetilde{T}}{\partial x_i} + \sum_{k=1}^{N} \overline{J_{i,k}}^t \widetilde{h}_{s,k} \tag{1.4}$$

where,

$$\lambda_t = \frac{\mu_t \overline{C_p}}{Pr^t} \tag{1.5}$$

In Eqn. 1.5, μ_t and Pr^t are the turbulent viscosity and turbulent Prandtl number respectively.

The sub grid scale species diffusion is given by,

$$\overrightarrow{J_{i,k}}^t = \bar{\rho}\left(\widetilde{u_i Y_k} - \widetilde{u}_i \tilde{Y}_k\right) \tag{1.6}$$

This flux is modeled as,

$$\overline{J_{i,k}}^t = -\bar{\rho}\left(D_k^t \frac{W_k}{W}\frac{\partial \tilde{X}_k}{\partial x_i} - \widetilde{Y}_k \widetilde{V}_i^{c,t}\right) \tag{1.7}$$

where the turbulent species diffusion coefficient D_k^t is obtained from,

$$D_k^t = \frac{\nu_t}{Sc_k^t} \tag{1.8}$$

Here Sc_k^t is the turbulent Schmidt number which is also specified by the user.

It is not in the scope or intention of this thesis to improve or analyze the subgrid-scale modeling. The SGS models used here are indeed well known [42, 43, 44] and have been used for a variety of LES simulations.

1.3.2 Mesh Resolution

In all numerical simulations, a finer mesh resolution leads to a reduction in computational errors. However this also results in a substantial increase in computational overhead. Note that aside from this specific known behavior, for a specific chosen mesh resolution, the errors relating to a certain numerical scheme have to be analyzed and if possible eradicated/delayed to obtain accurate results irrespective of the type of problem to be solved.

1.3.2.1 Numerical analysis

However, in many situations, finding an analytic solution to partial differential equations is not always possible, and adequately qualifying a numerical solution becomes rather difficult. However, a numerical scheme for which the numerical dispersion relation is very close to the physical dispersion relation given a governing equation is an alternative way of looking at the above-mentioned problem. Note that, Dispersion Relation Preservation, by definition, implies the consideration of space-time discretization simultaneously. Despite this, in many past efforts, DRP schemes were developed by considering only spatial discretization. This can be explained by the absence for a while of an appropriate metric for the numerical dispersion errors. Indeed no major systematic efforts were made in this direction, except for some qualitative work reported in Vichnevetsky - Bowles [45], Trefethen [46].

In the specific case of fluid dynamics, computations require capturing the propagation of disturbances through the flow. These disturbances usually take the form of either waves or wave packets. It is therefore natural to demand that numerical methods capture wave/wave packets realistically. First, the spectral analysis tools [47] were developed and reported to try to meet this purpose. Correct numerical dispersion expressions in terms of numerical group velocity were indeed reported [48, 49]. For many problems, errors and signals are of comparable size, and the different wave number components of the physical signal

should transmit their energy at the correct speed. It is hence mandatory to correctly understand errors and their propagation. Unfortunately, the original error analysis for discrete computing due to von Neumann is incomplete due to the starting assumption that error and true/physical signals follow the same dynamics. In fact, a more appropriate error analysis in discrete computing was reported in [50] for the 1D convection equation, using a Fourier-Laplace spectral analysis. This analysis shows that the discrete computing error dynamics is governed by the basic numerical properties of the adopted space-time algorithm. It results from such a study that to avoid error in numerical computation, one hopes to have the following:

1. A numerical scheme that can resolve all physical scales of the flow,
2. A numerical scheme that is neutrally stable with no numerical dissipation/anti-dissipation,
3. Individual wave and the collective group must propagate its phase and energy at the correct phase and group velocity free from spurious dispersion.

Apart from the issues of dissipation and numerical dispersion, other factors are responsible for errors in discrete computing. For example, anisotropy of numerical schemes is an important factor in multi-dimensional cases. Fourier-Laplace spectral analysis can also be used to quantitatively assess the resolution and the isotropy of numerical wave solutions. The aspect ratio of the grid in two dimensions along with the wave propagation angle are the parameters used to qualitatively and quantitatively assess the anisotropy of the numerical solutions. It has also been seen that some simulations result in the simultaneous creation of physical and spurious waves in discrete computing. The physical waves (acoustic, vortical), which are formed naturally, have been termed p-waves, while non-physical waves originating from numerical methods are termed q-waves [45, 51]. Spurious numerical waves have been identified as parasite waves by Trefethen [46], who conjectured that this is linked to the numerical group velocity. However, no quantitative measure was provided although the author suggested that the parasite waves occur for wavenumbers very close to the Nyquist limit at the highest resolved wavenumber. Authors [45] have reported semi-discrete analysis for second order central (CD2) spatial discretization scheme and Galerkin finite element methods for 1D convection equation to explain q-waves qualitatively. It has been shown that the existence of q-waves is related to the numerical group velocity.

In reality, no scheme exists that resolves all wavenumbers perfectly, especially for reacting flows. Indeed, additional temporal and spatial scales are introduced

by chemical reactions, hence theoretically requiring further local mesh than the one for equivalent non-reacting flow. Considering the computational cost and needed resolution, the area around the important flow features should be resolved with finer grids while a relatively coarser grids could be used in the other areas. This kind of issue can be properly addressed by using dynamic adaptive mesh refinement (AMR) to refine the mesh in regions of interest locally. The need for remeshing during a numerical simulation may arise from two different reasons: either to reduce the numerical inaccuracy induced by an inadequate mesh (most often because it is too coarse; yet, too fine meshes may induce numerical error and waste too much compute time), or because of a constantly evolving reacting flow field/changing geometry (moving bodies, deformations).

1.3.2.2 Adaptive mesh refinement

In LES, the largest scales are solved explicitly, while the smallest scales are modeled. The scale separation is performed by a filtering operation. The filtering operation is applied to the non-linear terms of the Navier-Stokes equation resulting in the sub-filter terms which are modeled in the LES approach. In CFD codes, the filtering operation is performed implicitly by the mesh size, assuming that filtering width and mesh size are equal. The sub-filter terms that are to be modeled are called the subgrid-scale terms (SGS) and is dependent on the mesh size. The influence of the SGS models decreases with mesh size and therefore, grid independency can not be achieved in LES. When the mesh size is very small, allowing us to solve for all ranges of scales then the SGS model becomes negligible. To simulate a realistic complex geometry in LES, an adequate mesh must be chosen to make sure that the simulations are accurate to guarantee the validity of SGS closure models.

Most commonly, meshes consist of quadrilaterals or triangles in 2D and hexahedra or tetrahedra in 3D. Triangles and tetrahedra are usually used to mesh the domain for complex domains [52]. However, using quadrilaterals and hexahedra results in using fewer elements to model the same domain size. Many numerical schemes also exploit the ease of implementation on these element types [53, 54]. In many industrial types of applications, advantages of both type of elements is required, which motivates the use of hybrid meshes with multiple element types. It may also be necessary to use prisms or pyramids to transition between hexahedra and tetrahedra elements [55, 56].

To overcome and satisfy the challenges discussed above, one of the most successful tools to improve the performance of numerical simulations is adaptive mesh refinement(AMR). In all numerical methods, a finer mesh resolution results

in a reduction of the computational error. AMR introduces the concept of changing the mesh resolution locally. Using AMR, the mesh can be refined in regions where it is required while keeping other parts of the mesh untouched, which is usually done through local error estimators [57]. AMR can refine the number of mesh elements, reducing the memory footprint during run-time. However, implementing AMR in a true parallel high-performance environment results in huge overhead in mesh management. The most demanding tasks include refining and coarsening of the mesh, re-partitioning in parallel, creating ghost elements and many more.

Mesh refinement is relatively less computationally expensive than remeshing, as the existing connectivity is used as the starting point, simplifying the mapping of the data associated with the original nodes and elements to the new ones. Especially if the refinement is performed by adding new nodes along the edges of the original elements, interpolating data to the new mesh is an easier operation. The main challenge to implement AMR on HPC systems is storing and load balancing the mesh in parallel, especially when the mesh frequently changes during the computation. Many different algorithms and approaches are found in the literature that describe different approaches to the splitting of elements. Most techniques involve edge subdivision, which is very effective and easy to implement in a serial context. However, it is still difficult to implement the successful serial algorithms to the distributed environment. Some of the early work on local mesh adaptation can be found in the works of Babuska [58]. After that, local and global mesh refinement and coarsening of triangular and tetrahedral meshes have been implemented for finite element methods [59], compressible flows [60], magnetostatic problems [61], electromagnetic problems [62] and to generate an optimal mesh based on equally distributed discretization error with the help of error estimates [63].

Today's large-scale simulations can only be run in parallel because many meshes are now too big to fit in the memory of a single computer. Since shared-memory architectures are subject to memory bottlenecks, scalability can only be achieved using distributed memory architectures such as workstation clusters. Therefore, a prerequisite for such simulations is to be able to generate huge meshes in a parallel, distributed-memory fashion. Moreover, if users would like to perform mesh adaptation, the latter must also be performed in parallel. There are also works in literature for parallel mesh refinement algorithms for triangular and tetrahedra meshes, which use bisection schemes [64, 65]. Zhang [66] presented a parallel algorithm for distributed memory parallel computers using bisection, characterized by simultaneous refinement of submeshes without the

need for a central coordinator process for managing new vertices. The mesh is partitioned into submeshes as many as the number of MPI processors. Partitioning is done using METIS. After the mesh is partitioned, submeshes are refined independently, with shared faces treated as boundary faces. The shared tetrahedra, which have been bisected during the first phase, is exchanged between neighboring submeshes, and the tetrahedra with one or more hanging shared faces are bisected. The process is continued till global conformity of the mesh is reached. De Cougny [64] used edge-based subdivision templates for refinement. Mesh refinement using the Delaunay method has been implemented for solving magnetostatic problems [61]. It has been successfully applied to varied cases with large-scale flows in parallel and will be used in present work to solve complex, large-scale, reacting, and compressible flow cases.

1.4 Objectives and organization of the thesis

Part 1 of the thesis covers the so called Global spectral analysis of the widely used numerical schemes - Lax Wendroff [67] and TTGC [68] schemes. This part of the thesis, focuses on the errors related to stability, dispersion, diffusion for the non-reacting and the reacting flow problems using the model Convection-Diffusion (CD) and Convection-Diffusion-Reaction (CDR) equation respectively.

In chapter 1, the non-reacting flow problem is studied using the CD equation whenever solved using the schemes of interest. Obtained the numerical property charts for each scheme to show significant combined effects of the CFL number, the Peclet number, and the dependency on the wavenumbers with different regions of stable and unstable simulations. These property charts are then validated using well designed numerical tests of a wavepacket problem. A 2D extension of the previous problem is then used to highlight the effect of anisotropy and wave propagation angle on the scheme property charts. Finally, the 2D stability charts are validated by solving a 2D Taylor-Green Vortex problem to highlight the importance and the importance of Global spectral analysis.

In chapter 2, the focus is shifted to the reacting flow problems using the CDR equation. A constant unity reaction source term is the first problem analyzed. The property charts are designed for both the numerical schemes and the validated through a selection of numerical experiments. To develop a source term that can be used for a practical combustion case, a reaction term formulation which is faithful to realistic gaseous combustion chemistry is presented. The designed reaction source term is similar to the Pfitzner reaction model. The stability limits are validated by solving the propagation of a 1D fully premixed flame

using AVBP. Finally, the stability limits (grid resolution) when solving a realistic reacting flow are established using GSA. Using the grid resolution, combined effects of the CFL and Pe numbers at a specific Da can be analyzed to check the effect of grid resolution on the stability of reacting flows.

Part 2 focuses on the fact that practically no numerical scheme can resolve all the wavenumbers and especially in reacting flows, additional temporal and spatial scales are introduced by chemical reactions. Hence we need the local finer meshes obtained through dynamic mesh refinement.

In chapter 3, the AMR methodology and its implementation based on two different techniques are introduced and explained in detail. The steps involved in the algorithm for dynamic parallel mesh adaptation for both adaptation techniques are explained. The details about the load balancing, partitioning, and interpolation techniques used in adaptation techniques are differentiated. Two simple test cases are chosen to validate the adaptation techniques: Firstly, the test case chosen is the 2D Karman vortex street case (non-reacting flow), and secondly, a planar flame in a 3D chamber (reacting flow) is chosen.

In chapter 4, we use dynamic mesh refinement for two compressible, reacting, turbulent flow cases. These cases are perfect candidates for dynamic mesh refinement as the flame front is constantly evolving and unsteady. The first case is a closed channel with one obstacle consisting of an upper and lower wall mounted fence-type obstacle. The second case is a channel with removable baffle plates and a single central obstacle. In both these test cases, the two adaptation techniques are compared with the experimental data available to validate the simulations. Appropriate combinations of mesh adaptation criteria were chosen to obtain accurate results. For these cases, a speed up in total wall time was observed while maintaining accuracy of the results when compared to the reference simulation.

1.5 Acknowledgment

This work has been partially supported by the EXCELLERAT project which has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 823691 and CERFACS.

Part I

Numerical Analysis

Numerical analysis: Convection - Diffusion equation

Contents

2.1 Introduction

This chapter deals with the numerical solution strategies used to solve equations in the LES solver used in the present study, ie. AVBP. AVBP is a predominantly MPI dependent, explicit time integration based hybrid Finite Volume (FV)-Finite Element (FE) solver that works on unstructured meshes. Compressible reactive, Navier-Stokes equations are solved in AVBP with the problem of combustion being addressed using the species transport framework. The convective numerical schemes available include purely FV schemes (Centered, Lax-Wendroff (LW)) [67] and FE schemes (variants of Taylor-Galerkin schemes such as TTGC [68] etc.).

Out of all these schemes, LW and TTGC are the most widely used. While LW enjoys the advantage of low computational cost, TTGC benefits from its superior accuracy and resolution properties.

Numerical methods for CFD have been a widely studied topic and the most popular among all the methods is definitely the Von-Neumann analysis. The Von-Neumann analysis [69] originates from the Los Alamos Laboratory and appeared in 1950s. This analysis involves expressing the numerical solution of periodic problems in terms of the Fourier frequencies and identifying the modes of the numerical difference equation that may potentially lead to numerical instability. The second method widely used today is the Matrix Stability method [70]. In this method, the convective (and diffusive) operators are expressed in the form of a matrix. The effect of boundary conditions can also be included in this matrix representation. The eigenvalues of the matrix then determine the stability of the numerical scheme studied. The third and the final method available is the so called Global Stability Analysis (GSA) [71]. This method analyses the numerical solution in the spectral space. The conditions for stability of the numerical scheme is obtained by looking at the amplification factors obtained from the GSA expressions. In addition to the issue of stability, emphasis is also usually put on the notion of dispersion relation preservation [72], a critical aspect to be considered for high performance computing.

While all of the above methods predict the numerical stability of linear methods accurately, they differ from each other by studying other numerical aspects of schemes as explained in what follows.

First, the Von-Neumann method is applicable only to periodic problems. Another shortcoming of this approach is the assumption that the numerical solution follows the exact convection speed: ie. the one prescribed by the governing PDE. This hypothesis is seldom true that can be seen later for correct dispersion relations. The numerical phase and group velocity are indeed scheme dependent and vary based on the Courant-Friedlich-Lewy (CFL) number and the wavenumber of the solution. This assumption by Von-Neumann has been used to derive an evolution equation for the numerical error. Similar to the convection speed, the other physical properties of the governing equations such as the diffusion coefficient (for the convection-diffusion equation) and the source term (for the convection-diffusion-reaction equation) also are not necessarily constants when solved numerically.

Next, the Matrix Method on the other hand, provides information on the stability of schemes by looking at the spectral radii of the matrix produced by

the numerical operators. If the spectral radius is larger than one, then numerical instability is expected during long time integration. This method predicts instability (only) after a considerable duration of time integration.

Another major drawback suffered by the previously mentioned methods is the lack of information on the dispersion relation properties of schemes. Dispersion relation of a PDE refers to the expression relating the spatio-temporal properties in the spectral space. Any numerical method must be able to satisfy this relation numerically to the maximum wavenumber extent possible. This particular aspect is critical and seldom studied in high performance computing such as in DNS and LES. Indeed, a stable scheme does not necessarily ensure an accurate numerical solution. In fact, the numerical method should be able to accurately resolve the speed of propagation of information, diffusion of information and the production (or destruction) of information in addition to being stable. Only when the error associated with all the above properties is minimized, a stable as well accurate simulation results are obtained. Hence, while choosing numerical schemes for LES or DNS solvers, one should ensure that the scheme reproduces the right amplitude of the solution (dissipation), the right speed (dispersion) and the right diffusive (diffusion) properties of the governing PDE. Any scheme which satisfies these conditions is called a Dispersion Relation Preserving (DRP) scheme [73]. In contrast to the previously mentioned methods, GSA addresses these properties, that is it predicts not only the stability of the scheme but also the numerical dispersion, diffusion and source term properties of schemes (DRP properties). In addition to this, GSA also has the additional capability of studying the effect of boundary closures on numerical schemes. This is why in this study, GSA is used to carry out the numerical analysis of the linear Convection-Diffusion equation (LCDE) [74] and the linear Convection-Diffusion-Reaction equation (LCDRE) [75].

In most natural phenomena, convection and diffusion are fundamental processes that often compete yielding complex solutions. The importance of these processes however varies since these can occur over a wide range of disciplines and regimes like in fluid flows, climate studies, astrophysics or various branches of mathematics related to pattern formation [76, 77]. In fluid dynamics, the interplay between convection and diffusion is key to determining the stability and dynamics, for example in the study of fluid stability in differentially heated fluid layers [78]. It is also at the origin of turbulent flow closures like the (k,ϵ)-turbulence models [79]. For the specific problem of the geomagnetic storm present near Earth, authors [80] discussed the problem of plasma pressure diffusion or even the observed process of anti-diffusion at the origin of the so called “entropy anti-diffusion instability”. It is finally at the root of studies on granular transport

problems for which flow criticality and clustering [81] arise due to the coupling of a heat flow with diffusion and creates order in a chaotic system [82]. The governing equations describing the above discussed phenomena are however usually highly non-linear and thereby preclude from having access to exact solutions. Two approaches are therefore possible, either use simplified model equations such as linear convection, linear diffusion or linear convection-diffusion equations or develop numerical methods that solve the non linear problem using well validated numerical schemes. Note that this last step, in all cases requires the use of simplified models for which the general nature of the solution is known and restricted to only a few linear problems. Adopted numerical analysis methods are however multiple and with limits. It is therefore still useful to investigate numerical methods to fully understand the behavior of a chosen numerical scheme especially in complex problems. GSA has been used for variety of popularly used numerical schemes for solving reacting and non-reacting problems [74, 75].

Although this chapter includes the description of the numerical methods used in AVBP, emphasis is on the numerical analysis of the popular schemes used in CFD namely the Lax-Wendroff LW scheme and Taylor-Galerkin, TTGC scheme. In the present study, the LW convective scheme is used in conjunction with a centered scheme, hereafter denoted by LW-CD_2 and Taylor-Galerkin convective scheme as TTGC-CD_2. Note that such a combination is quite popular and is often used in LES of engineering applications [83], owing to its superior resolution in comparison to other second-order schemes as well as for its low computational cost. Although the stability analysis of the schemes have been performed in the past using Von-Neumann analysis, GSA of the above mentioned schemes has not been attempted yet. The objective is thus to capture by GSA the diffusive as well as the dispersive properties of the resulting scheme for the $1D$ and $2D$ convection-diffusion equations. For the $2D$ convection-diffusion equation, in addition to the above mentioned properties, the isotropic nature of the scheme in two-dimensions is also analyzed, where one looks for the properties of such a scheme as a function of the angle of wave front direction. The validity and applicability of GSA are demonstrated not only for this linear $2D$ problem but also for the non-linear Navier-Stokes equations, thereby proving the efficacy of GSA in studying realistic flows using non-linear academic and industrial CFD solvers.

There are three main reasons why numerical analysis of schemes using GSA are beneficial. First, such an analysis can highlight the numerical properties of these schemes (especially combined effect of Convection-Diffusion) and could be helpful in interpreting the obtained numerical solution. Second, the specific analysis of schemes when applied to governing equations involving diffusion and

convective terms simultaneously are scarce and still needed to fully understand the numerics (stability, dispersion and dissipation) of such problems . Finally, presenting such an analysis serves as a necessary precursor (obtaining stability limits as well as limits for accurate numerical solution) to the following chapter where the same analysis is extended to study the effect of chemical source term on the numerical resolution of the Convection-Diffusion-Reaction equation. These mentioned reasons are the objectives for the current chapter of the manuscript.

Keeping these objectives in mind, this chapter is organized as follows. In the following, a theoretical analysis of the convection-diffusion equation is first provided as an introduction to GSA of the LW and TTGC schemes. Of particular interest are the relationships provided by GSA between the numerical scheme properties: *i.e.* amplification factor, dissipation and dispersion errors as function of the non-dimensional computational variables that are the non-dimensional wavenumber (kh), CFL number (Nc) and Peclet number (Pe). GSA provides valuable information on the scheme stability as well as the diffusive and the dispersive properties of the scheme allowing to finely identify the limited range of acceptable Nc-Pe combinations ensuring DRP. The analysis of the $1D$ problem is first detailed along with dedicated numerical experiments to illustrate and validate the various findings. The $2D$ convection-diffusion problem is then addressed to finish with a $2D$ Taylor-Green vortex problem simulated using the full non-linear Navier-Stokes equations in an attempt to illustrate the contribution of GSA in explaining specific observed behaviors.

2.2 Analysis of the 1D linear convection-diffusion equation

The theoretical and numerical analysis of the linear $1D$ convection-diffusion equation is detailed next, to be then followed by the use of GSA on the LW scheme. The analysis here is mainly devoted to stability, dispersion and dissipation as well as group velocity responses. This section ends with the validation of the findings using a variety of numerical test cases.

2.2.1 Linear $1D$ convection-diffusion equation properties

The $1D$ linear convection-diffusion equation reads as,

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \alpha\frac{\partial^2 u}{\partial x^2}, \tag{2.1}$$

here, c and α are constant real numbers corresponding respectively to the convection speed and diffusion coefficient. For spatially periodic configurations, the above expression can be re-expressed applying a spatial Fourier Transform to the unknown function $u(x,t)$, in the hybrid spectral plane [47, 71]. In this case, Eqn. 2.1 can be written as,

$$\frac{d\hat{U}}{dt} + ick\hat{U} = -\alpha k^2 \hat{U}. \tag{2.2}$$

Here, $\hat{U}$ is the Fourier amplitude and k represents the wavenumber which is a real number. This new equation can be solved analytically for a general initial condition of the form, $u(x,0) = f(x) = \int \hat{F}(k) e^{ikx} dk$, so that,

$$\hat{U}(k,t) = \hat{F}(k)\, e^{-\alpha k^2 t} e^{-ikct}. \tag{2.3}$$

For the physical dispersion properties of such a system, the unknown function $u(x,t)$, needs to be represented by a bi-dimensional Fourier-Laplace transform (which implies a time as well as space periodic problem) such that,

$$u(x,t) = \int\int \bar{U}(k,\omega)\, e^{i(k\,x - \omega t)}\, dk d\omega, \tag{2.4}$$

where, $\bar{U}$ and ω refer to the Fourier-Laplace amplitude and the complex circular frequency respectively. Substituting the above expression in Eq. (2.1), the physical dispersion relation is obtained and is expressed as,

$$\omega = c\,k - i\,\alpha\,k^2. \tag{2.5}$$

This dispersion relation is an important property and governs the wave propagation / diffusion in a given medium. The expression also indicates that ω is a complex number, determined by the wavelength of interest, the convection speed, c and the diffusion coefficient, α. The expression also details the phase and group velocities of signal propagation by considering only the real part of ω. Any numerical scheme used to solve the above convection-diffusion equation must obey this dispersion relation to minimize phase and dispersion errors [50, 71]. Different physical phase speeds can be obtained from the above dispersion relation, using,

$$c_\varphi = \frac{\omega}{k} = c - i\,\alpha k, \tag{2.6}$$

from which the actual speed at which physical waves travel is obtained by considering only the real part of the expression: *i.e.* $c_{phys} = \Re(c_\varphi) = c$, while the

imaginary part is the contribution of the diffusion process which is pure damping of the amplitude of wave k with time.

Similarly, the physical group velocity which defines the velocity at which energy propagates in this same medium is represented by, $v_{g,phys} = \Re(v_{g,\varphi}) = c$ with,

$$v_{g,\varphi} = \frac{\partial \omega}{\partial k} = c - 2\,i\,\alpha k. \tag{2.7}$$

Here as well, the imaginary part is a direct result of the energy damping as a function of time and contained in a wave k. The importance of the complex group velocity or the energy propagation speed is detailed in [74]. Note here, both retrieved speeds are effectively complex in nature due to the diffusion process taking place at the same time as convection.

Another important property of the above problem is that it also provides access to the physical amplification factor: *i.e.* comparing the solution amplitude at two distinct instants separated by Δt,

$$G_{phys} = \frac{\hat{U}(k, t + \Delta t)}{\hat{U}(k, t)} = e^{-\alpha\, k^2 \Delta t} e^{-i\, k\, c\, \Delta t} = e^{-i\, \omega\, \Delta t}. \tag{2.8}$$

For the problem at hand, G_{phys} solely depends on the speed at which information travels through the medium while it is diffused as expressed through Eq. (3.9).

2.2.2 Discrete $1D$ linear convection-diffusion system properties

The numerical solution of Eq. (2.1) involves the use of two non-dimensional parameters, which depend upon the choice of the space-time discretization of the governing equation. These numbers are Nc and Pe, which both relate to the two physical processes of convection and diffusion. These follow the algebraic expression linking the fixed chosen temporal and spatial steps respectively, Δt and h: $Nc = \frac{c\Delta t}{h}$, $Pe = \frac{\alpha \Delta t}{h^2}$. Upon substituting these expressions in Eq. (3.9), the physical amplification factor as a function of the introduced non-dimensional numbers is obtained and reads,

$$G_{phys} = e^{-Pe\,(kh)^2}\; e^{-i\,N_c\,(kh)}. \tag{2.9}$$

The CFL number, Nc, is a measure of the non-dimensional distance traveled by the signal for a given time step and is directly related to the convection speed

in the governing equation for a pure convection problem. Likewise, Pe quantifies the decay rate in the diffusion equation. In numerical simulations, contrary to the popular assumption in the classical Von-Neumann analysis, the numerical convection speed, c_{num}, and the numerical diffusion, α_{num}, show significant dependency on the wavenumber, grid spacing h, Nc and Pe as demonstrated in [50, 71]. As a result, every numerical scheme has a corresponding numerical amplification factor, G_{num}, that corresponds to a numerical dispersion relation which governs the evolution of the solution in time. Ideally, it is highly desirable that for a numerical scheme, G_{num} is as close as possible to G_{phy}. Therefore, it is essential to perform the equivalent detailed analysis as in section 2.2.1 in the context of the discrete system.

Presuming that the discretization produces a consistent representation of the continuous problem, it should result in a dispersion relation equivalent to Eq. (2.5) as demonstrated in [50, 84]. As a result, the numerical dispersion properties can be expressed using,

$$\omega_{num} = c_{num}\, k - i\, \alpha_{num}\, k^2, \tag{2.10}$$

where ω_{num} is the complex numerical circular frequency which differs from its exact expression due to the fact that the effective numerical speed c_{num} and diffusion coefficient α_{num} are both real functions which vary with k as well as the numerical parameters, Nc and Pe. A similar relation has been reported and its consequences described in the context of the physical process of anti-diffusion and focusing by [74]. Similarly, the numerical amplification factor can be obtained and expressed as a function of the two real functions: α_{num}, the measure of the effective decay rate of the numerical solution and c_{num}, the effective speed at which information travels,

$$G_{num} = e^{-\alpha_{num}\, k^2 \Delta t}\, e^{-i\, k\, c_{num} \Delta t} = e^{-i\, \omega_{num} \Delta t}. \tag{2.11}$$

Using the previous relationships identified above, and applied here, one notes that:

$$c_{num} = \Re\left(\frac{\omega_{num}}{k}\right), \tag{2.12}$$

and,

$$v_{g,num} = \Re\left(\frac{\partial \omega_{num}}{\partial k}\right). \tag{2.13}$$

To obtain c_{num}, the notion of numerical phase shift per time step [50, 71]: *i.e.* the ratio between the imaginary and real part of G_{num}, is used:

$$tan(\beta) = -\frac{\Im(G_{num})}{\Re(G_{num})} = tan(c_{num} k \Delta t). \tag{2.14}$$

This allows to express the non-dimensional effective numerical speed with respect to a purely convective problem, that is,

$$\frac{c_{num}}{c_{phys}} = \frac{\beta}{kc\Delta t} = -\frac{1}{(kh)N_c}\, tan^{-1}\left[\frac{\Im(G_{num})}{\Re(G_{num})}\right]. \tag{2.15}$$

Similarly, the numerical group velocity can be obtained by noting that,

$$v_{g,num} = \Re\left(\frac{\partial \omega_{num}}{\partial k}\right) = \frac{1}{\Delta t}\frac{\partial \beta}{\partial k}, \tag{2.16}$$

which results after normalization into,

$$\frac{v_{g,num}}{v_{g,phys}} = \frac{1}{N_c}\frac{d\beta}{d(kh)}. \tag{2.17}$$

Based on Eq. (3.12), it can be noted that the modulus of the numerical amplification factor is solely dependent by construction on α_{num},

$$|G_{num}| = e^{-\alpha_{num} k^2 \Delta t}, \tag{2.18}$$

or after introducing the non-dimensional parameter Pe,

$$ln|G_{num}| = -\frac{\alpha_{num}}{\alpha}(kh)^2\, Pe. \tag{2.19}$$

The numerical amplification factor is a clear function of kh and Pe as well as Nc which can appear through the actual expression of α_{num}. Alternatively, Eq. (3.17) provides a direct access to the numerical diffusion coefficient which upon normalization results in,

$$\frac{\alpha_{num}}{\alpha} = \frac{-ln|G_{num}|}{(kh)^2 Pe}. \tag{2.20}$$

The above expression clearly expresses the fact that a numerical scheme mimics the exact diffusion process when the above ratio is unity. Higher coefficients correspond to excessive diffusion while lower diffusion is obtained for values of α_{num}/α between 0 and 1. Negative values are also possible as seen later and indicate anti-diffusion. These are nonphysical and lead to numerical instability.

2.2.3 GSA of numerical schemes for the $1D$ convection-diffusion equation

The above discrete expressions that have been derived can be obtained for specific numerical schemes. Here, the focus is given to LW and TTCG schemes for the convective fluxes and the second central difference diffusive fluxes, namely LW-CD_2 and TTGC-CD_2.

LW-CD_2 scheme

Now, the above analysis is applied to the LW-CD_2 scheme. Here, the LW scheme is used for the spatio-temporal discretization of the convective terms while the diffusive term is approximated using a second order central difference (CD_2) scheme. Applied to the linear convection-diffusion equation at j^{th} node, at time level n, and introducing the non-dimensional numbers, the above scheme results in the expression,

$$u_j^{n+1} = u_j^n - \frac{N_c}{2}(u_{j+1}^n - u_{j-1}^n) + \frac{1}{2}\,\zeta(u_{j+1}^n - 2u_j^n + u_{j-1}^n), \tag{2.21}$$

where ζ is defined as $(N_c^2 + 2\,Pe)$.

Upon substituting the Fourier representation of the unknown variable u and upon simplification, Eq. (2.22) provides the expression for the numerical amplification factor. For LW-CD_2, this factor reads,

$$G_{num} = 1 - i\,Nc\,sin(kh) + [\zeta(cos(kh) - 1)]. \tag{2.22}$$

Using Eq. (3.13), the expression for the numerical phase shift becomes,

$$\beta = tan^{-1}\left(\frac{N_c\,sin(kh)}{1 + \zeta(cos(kh) - 1)}\right), \tag{2.23}$$

resulting in, the non-dimensional numerical phase and group velocities,

$$\frac{c_{num}}{c} = -\frac{1}{(kh)N_c}tan^{-1}\left(\frac{N_c\,sin(kh)}{1 + \zeta(cos(kh) - 1)}\right), \tag{2.24}$$

$$\frac{v_{g,num}}{c} = \left[\frac{cos(kh) + \zeta(1 - cos(kh))}{(1 + \zeta(cos(kh) - 1))^2 + N_c^2\,sin^2(kh)}\right]. \tag{2.25}$$

Finally, the non-dimensional numerical diffusion coefficient is obtained as,

$$\frac{\alpha_{num}}{\alpha} = \left[\frac{-ln\left(\sqrt{(1 + \zeta(cos(kh) - 1))^2 + N_c^2\,sin^2(kh)}\right)}{(kh)^2 Pe}\right]. \tag{2.26}$$

Before analyzing the plots for LW-CD_2 scheme, similar expressions are obtained for TTGC-CD_2 scheme.

TTGC-CD_2 scheme

The TTGC is a two step scheme used to treat the spatio-temporal discretization of the convective terms [68]. When applied to the LCDE, it yields the following relationships,

$$\frac{(\tilde{u}^n_{i+1} + 4\tilde{u}^n_i + \tilde{u}^n_{i-1})}{6} = \frac{(u^n_{i+1} + 4u^n_i + u^n_{i-1})}{6} - \alpha_{TTGC}\, N_c \frac{(u^n_{i+1} - u^n_{i-1})}{2} + (\beta_{TTGC}\, N_c^2 + Pe)\,(u^n_{i+1} - 2u_i + u^n_{i-1}) \tag{2.27}$$

$$\frac{(u^n_{i+1} + 4u^n_i + u^n_{i-1})}{6} = \frac{(u^n_{i+1} + 4u^n_i + u^n_{i-1})}{6} - N_c \frac{(\tilde{u}^n_{i+1} - \tilde{u}^n_{i-1})}{2} + (\gamma_{TTGC}\, N_c^2 + Pe)\,(u^n_{i+1} - 2u_i + u^n_{i-1}) \tag{2.28}$$

The numerical properties of the TTGC scheme are then obtained using,

$$G_{num} = 1 + \frac{\hat{A}\,\hat{S}\,\varsigma_2}{N_c} - \alpha_{TTGC}(\hat{A}\hat{L})^2 - i\hat{A}\hat{L}\left(1 + \frac{\hat{A}\hat{S}\varsigma_1}{N_c}\right), \tag{2.29}$$

The expressions for $\frac{\alpha_{num}}{\alpha}$ is calculated using the above relation for G_{num} from equation 3.19.

for which,
$\hat{A} = \frac{3N_c}{2+cos(kh)}$, $\hat{S} = 2cos(kh) - 1$, $\hat{L} = sin(kh)$

The above expressions obtained for both schemes clearly confirm a dependency of the numerical properties on all parameters: *i.e.* kh, Nc and Pe. These aspects are specifically detailed next, since they have important effects on the scheme stability, effective diffusion, convection speed and group velocity.

As mentioned before, it is ideal if numerical methods used for LES and DNS follow the dispersion relation of the original governing equation as closely as possible. This implies that the numerical solution should not only be stable, but also be capable of resolving the magnitude, the speed and the diffusion characteristics of the exact solution. When it comes to the scheme behavior, its accuracy can be first assessed by looking at the relative amplification factor modulus $G_{rel} = \frac{|G_{num}|}{|G_{phys}|}$. Likewise, stability appears by looking at the condition, $|G_{num}| < 1$, while the effective diffusive process as well as the potential appearance of anti-diffusion is performed by investigating, α_{num}. Lastly, the non-dimensionalized convection velocity, c_{num}/c_{phys}, and the non-dimensional group velocity, $v_{g,num}/v_{g,phys}$, characterize the scheme's ability to recover the true convective process of the studied

system. Note that to identify the regions in the kh-Nc-Pe space where the dispersion relation results in satisfying the DRP condition, a tolerance of $\pm 1\%$ deviation is applied hereon.

G_{rel} contours issued by LW-CD$_2$ are plotted in Fig. 2.1 for increasing values of Pe in the Nc-kh plane. G_{rel} is a measure of the effective numerical diffusion process relative to the exact solution. Excessive damping is therefore observed if $G_{rel} < 1$ and under-diffusion or even instability can be obtained if $G_{rel} > 1$. Note that the condition $G_{rel} > 1$ does not necessarily imply numerical instability since the solution can be bounded and still be larger than the exact solution (under diffusive). To obtain the numerical instability limit, the absolute numerical amplification factor $|G_{num}|$ (from which instability corresponds to $|G_{num}| > 1$) should be considered as well as G_{rel}. The numerically unstable regions are shaded in orange in Fig. 2.1. Likewise, the DRP regions are identified by allowing a 1% deviation tolerance ($0.99 < G_{rel} < 1.01$) and are indicated by the grey shaded regions.

We begin by focusing on the $Pe \leq 0.01$ cases which are shown in Fig. 2.1 (a) & (b). For such cases, when compared to the standard LW scheme (corresponding to $Pe = 0$ in Fig. 2.3), it is clear (as expected due to the addition of diffusion) that a modified range of stability is obtained as shown in Fig. 2.3(a). Due to the added physical / numerical diffusion, the standard LW critical stability criterion for pure convection, $Nc = 1$ does not hold strictly anymore for all values of kh. Indeed, the addition of the physical diffusion makes the scheme stable at all wavenumbers for larger Nc values. It is also observed that in these cases ($Pe = 0.0001, 0.01$), DRP is respected with minimal error in the solution. For too low or too high Nc values, it can however result in errors and more specifically under-diffusion for high wavenumbers. Here, enhanced stability is due to the dominance of the diffusion process in the physics of the problem rendering the convective process marginal and therefore results in the improved stability range. When it comes to potential stability issues with LW-CD$_2$, as Pe increases, the acceptable range of $Nc - kh$ values decrease and stability limits appear around $Pe \approx 0.1$ as shown by Fig. 2.1(c). Ultimately, for $Pe \approx 0.5$, instability appears for all Nc values mainly due to the Nyquist limiting wavelength $kh = \pi$, which is in agreement with the well known pure diffusive problem which is unstable for this value. In addition, numerical instability can occur for intermediate values of Pe and different Nc ranges. For example, when $Pe = 0.1$, LW-CD$_2$ is unstable for $Nc > 0.9$ and $kh > 1.5$. For $Pe \geq 0.5$ problems in the range $Nc \leq 1$ and $kh < 1$ are possibly stable although round-off errors will ultimately render these simulations unstable after long time integration. These specific properties are

overlooked by conventional Von-Neumann analyses since this approach predicts a blanket condition of stability for values of $Pe < 0.5$ when analyzing a purely diffusive problem and $Nc < 1$ for a purely convective problem. It is also worth noting that as Pe increases, the DRP region shrinks to lower kh values. As a result, an appropriate value of Nc should be chosen to obtain accurate and stable solution. The common choice of Nc and Pe values is clearly not transparent and surely not as simple as a linear combination of both individual processes as often addressed in CFD codes.

The G_{rel} contours for the TTGC-CD_2 scheme are plotted in Fig. 2.2 for different values of Pe in the Nc-kh plane. In the cases where, $Pe \leq 0.001$ shown in Fig. 2.2 (a) & (b), there exists a large region of DRP present especially for low wavenumbers with no instability noticed. For values of Nc up to 0.2, there is a stable DRP preserving region for all wavenumbers which appears to be wider than the one obtained with the LW-CD_2 scheme. As Pe increases, the acceptable range of $Nc - kh$ values decreases and a first region of instability appears around $Pe \approx 0.1$, on Fig. 3.2(d). This behavior is similar to the one of the $LW - CD_2$ scheme where the first instability is also noticed at $Pe \approx 0.1$. Finally, for $Pe \approx 0.17$, instability appears for all Nc values mainly due to the Nyquist limiting wavelength $kh = \pi$: ie. earlier than for the $LW - CD_2$ scheme for which instability for all N_c values appears at $Pe \approx 0.50$.

Figure. 2.3(a) shows the stability map in the Nc-Pe plane, by considering the value of $|G_{num}|$ across all values of kh in $[0, \pi]$ for the LW-CD_2 scheme. It can be observed that for vanishing diffusion ($Pe \to 0$), the stability condition for linear convection equation ($Nc \leq 1.0$) is recovered while for pure diffusion ($Nc \to 0$), the well known stability limit [85] ($Pe \leq 0.5$) is obtained. To better understand the non-negligible combined effects of Nc and Pe on the stability, Fig. 2.3(b) shows the same stability map in the Nc-Pe plane, but considering only values of kh in the range 0 to $\frac{\pi}{2}$: *i.e.* signals without poorly discretized wavelengths. For such a case, the region of stability marked in the white shade region increases for a wider range of Nc and Pe combinations. The differences in the graphs of frame (a) and (b) highlight the region where instability is triggered. In the first case, it is predominantly due to the Nyquist frequency limit which is not the scenario for the case corresponding to frame (b) where well resolved scales can still become unstable. In frame (c), the deep blue region indicates the same stable region as shown in frame (a) for all values of kh considered within the range $[0, \pi]$. The remaining contours indicate the kh value at which the instability is first observed ($|G_{num}| > 1$) thus, indicating the critical wavenumber that can be resolved by the scheme when choosing a certain combination of

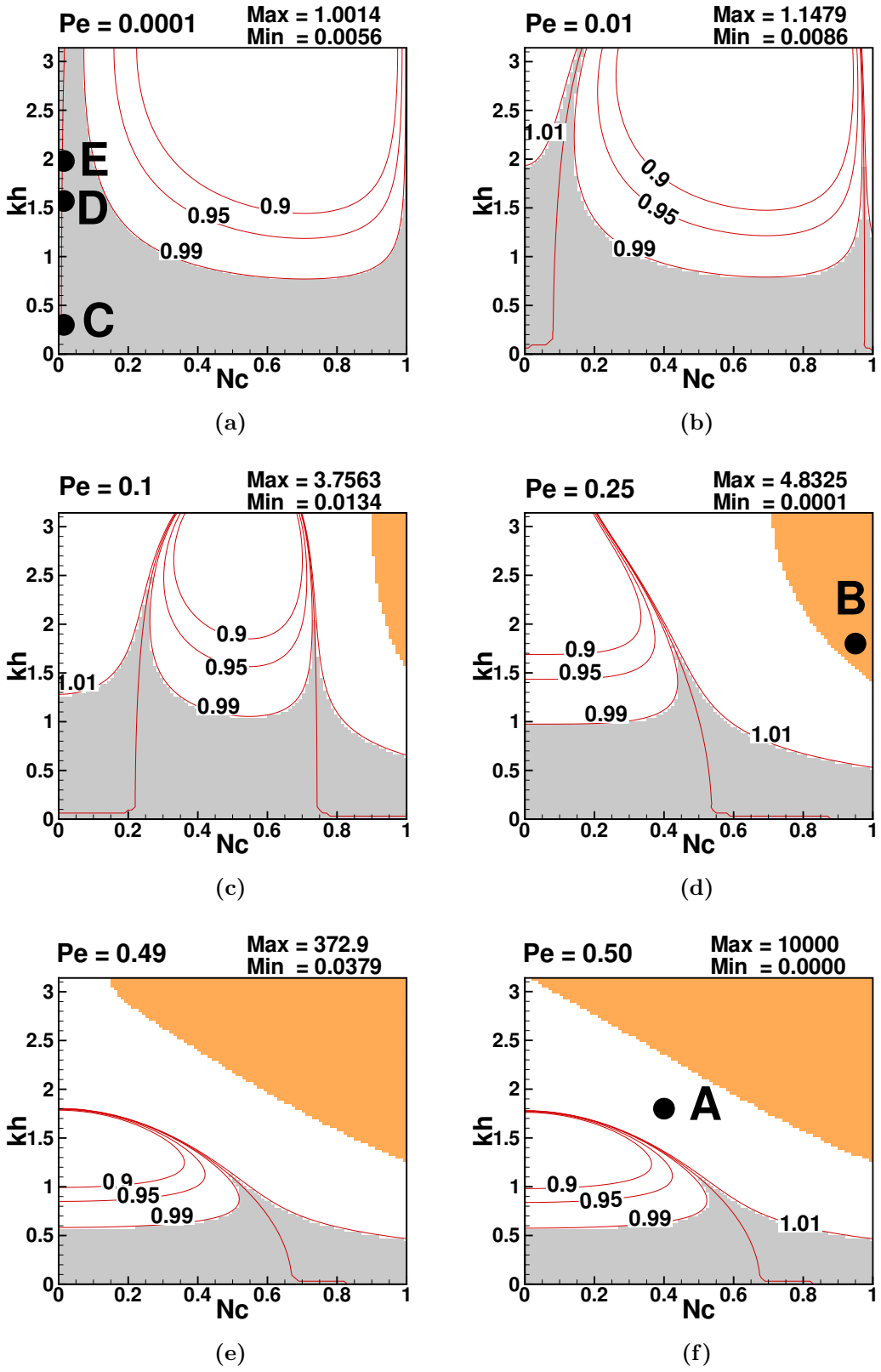


Figure 2.1: Contours of $|G_{num}|/|G_{phy}|$ as a function of kh and Nc for mentioned values of Pe for the **LW-CD$_2$** scheme. Regions of numerical instability are colored in orange, while DRP regions are shaded in grey. Five numerical test cases are marked (A to E) and explained later.

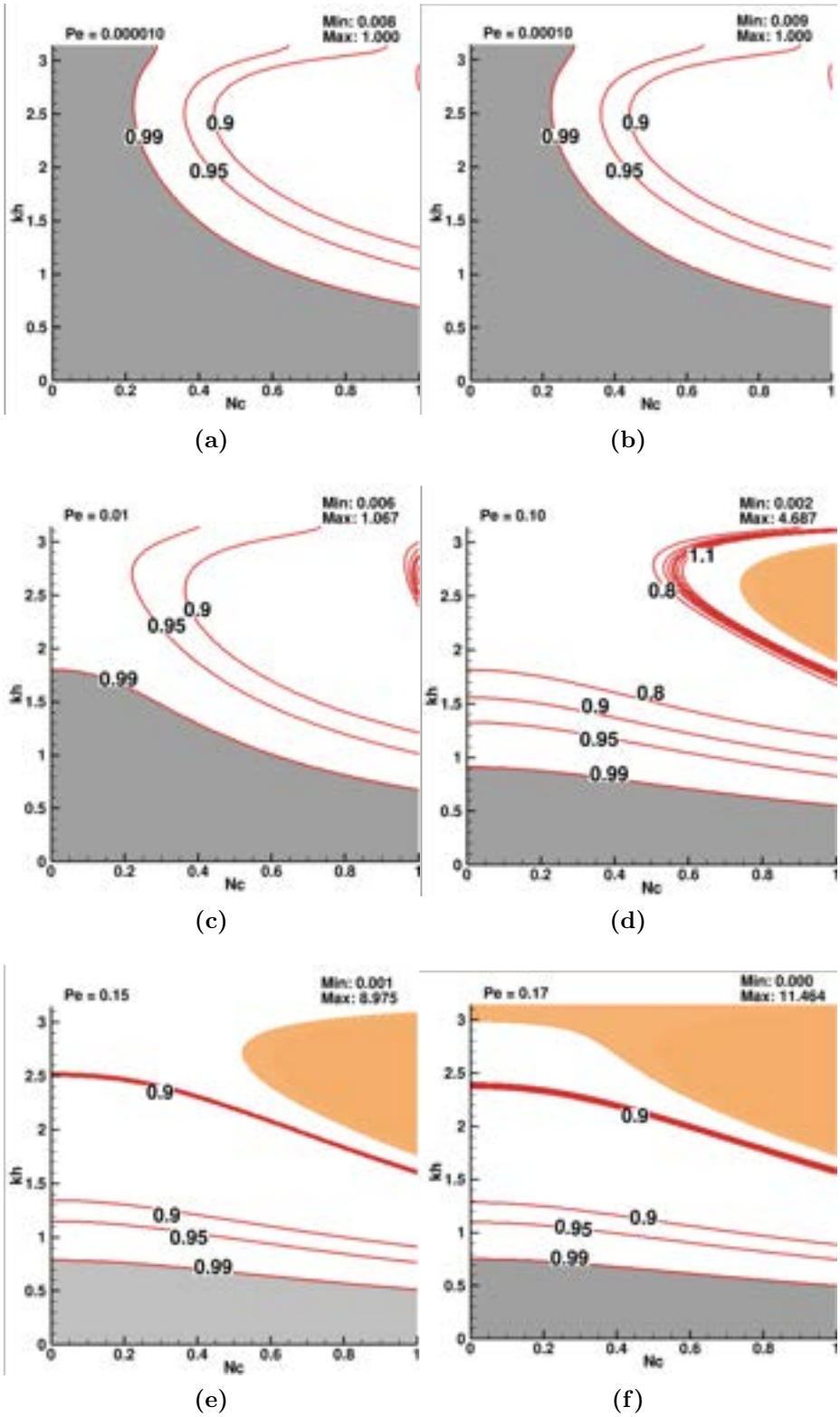


Figure 2.2: Contours of $|G_{num}|/|G_{phy}|$ as a function of kh and Nc for mentioned values of Pe for the TTGC-CD_2 scheme. Regions of numerical instability are colored in orange, while DRP regions are shaded in grey.

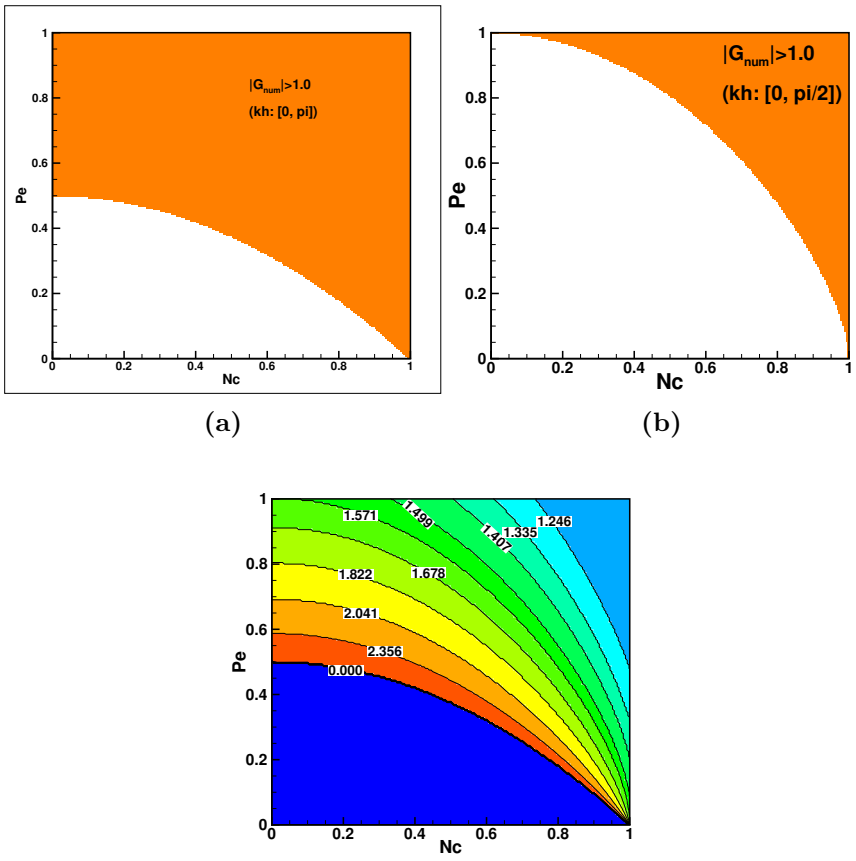


Figure 2.3: Stability region in the $Nc - Pe$ plane for ranges of (a) kh from 0 to π. (b) kh from 0 to $\frac{\pi}{2}$ for the LW-CD$_2$ scheme. The orange shaded region identifies the unstable region, $|G_{num}| > 1$. (c) denotes the critical value of kh at which numerical instability is first observed.

Nc and Pe. When approaching higher Nc and Pe combinations, the critical kh value that can be resolved keeps on decreasing leading to numerical instability at all wavenumbers above the critical value. The same trends are observed for TTGC-CD$_2$ scheme, as a result, not presented here.

As mentioned previously, G_{rel} is an indirect measure of the solution accuracy expressing the error through the inaccurate evaluation of the diffusive process taking place numerically. Another measure of the numerical errors in the diffusion process is α_{num}/α as shown in Fig. 2.4 for the LW-CD$_2$ scheme. The contours of α_{num}/α exhibit similar variations as that observed for $|G_{num}|$. This is expected, since diffusion is the only mechanism through which damping of the solution occurs in convection-diffusion phenomena and is captured by $|G_{num}|$. For this same reason, variations of the non-dimensional effective diffusion coefficient, α_{num}/α is also an indicator of numerical stability since it is the only mechanism contributing to the evolution of the amplitude of a wave in time as evidenced

by Eq. (3.16). Typically for negative numerical diffusion coefficients, the role of diffusion on the solution is changed and results into focusing the solution at the location of instability generation in the parameter space. This has been demonstrated in the literature as anti-diffusion and such a numerical behavior is not only unphysical, but also results in solution blow off [86]. Here, the stability limit is reported on the basis of the condition: $\alpha_{num}/\alpha > 0$. These property charts exhibit similar features as in Fig. 2.1 (as $\alpha_{num}/\alpha \leq 0$ corresponds to $|G_{num}| > 1$), where the unstable region is shaded in orange. However, the DRP region, shown in grey indicates a much narrower acceptable region that can be used for numerical simulations. As a result, this tolerance ($0.99 \leq \alpha_{num}/\alpha \leq 1.01$) is much more restrictive to obtain accurate and stable solutions.

Similarly, the graphs of the numerical errors in the diffusion process (α_{num}/α) for the TTGC-CD_2 scheme are shown in Fig. 2.5. The contours for this variable is very similar to those seen in the G_{rel} contours. In the case of negative numerical diffusion, anti-diffusion results in a focusing of the solution at the location of instability in the parameter space. As Pe increases, the DRP region initially increases but does not change much especially for larger Pe values till the critical value of $Pe = 0.17$, for which there is no value of Nc which is stable for all wavenumbers. The DRP region for this scheme is obtained for a smaller range of N_c and kh when compared to the plots observed above for the LW-CD_2 scheme.

Another property of interest is the ability of a scheme to propagate the information at the right speeds. This is illustrated through the contours of the non-dimensional phase speed, as shown in Fig. 2.6. For low values of Pe (0.0001 and 0.01), the DRP regions are limited to a smaller range of kh ($kh < 0.3$) for $Nc < 0.7$ and a larger range of kh ($kh < 1.3$) for $Nc > 0.8$. Hence, from the perspective of resolving the phase speed accurately, it is desirable to carry out computations at high values of Nc ($Nc > 0.9$) for low Pe values. As Pe is increased, the DRP region is reduced. For $Pe > 0.25$, the DRP regions are limited to $kh < 0.4$ for all values of Nc. It is also interesting to note that for high values of Pe, the numerical phase speed is always higher than the physical phase speed. It can be also noted that no region with negative phase speed is observed for any value of Pe. This observation is of interest since negative phase and group speeds can be detrimental to numerical solutions as they imply non-physical waves moving in the upstream direction to the flow. When it comes to real flow simulations, it is important to underline the fact that the flow involves a range of wavenumbers. In such cases, phase speed is of interest but so is the speed at which groups of waves travel (group velocity) along with the phase speed. For the linear CD problem, both speeds are the same: *i.e.* constant c. This translates to the fact

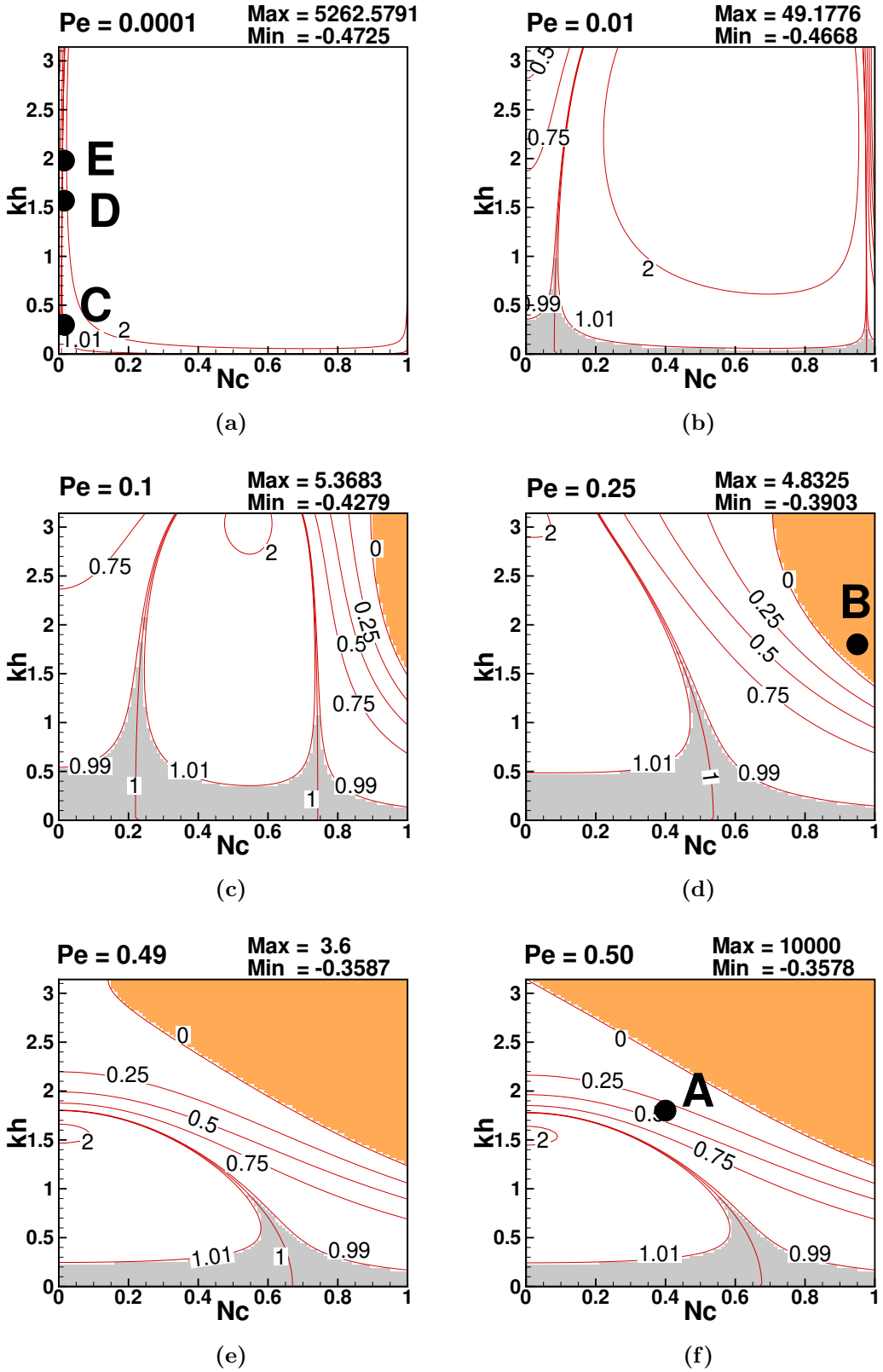


Figure 2.4: Effective measure of the numerical diffusion process, α_{num}/α in the $Nc - Pe$ plane for the LW-CD$_2$ scheme. Region of instability is identified in orange by $|G_{num}| \leq 1$; the grey region corresponds to the 1% DRP zone. Five numerical test cases are marked (A to E) and explained later.

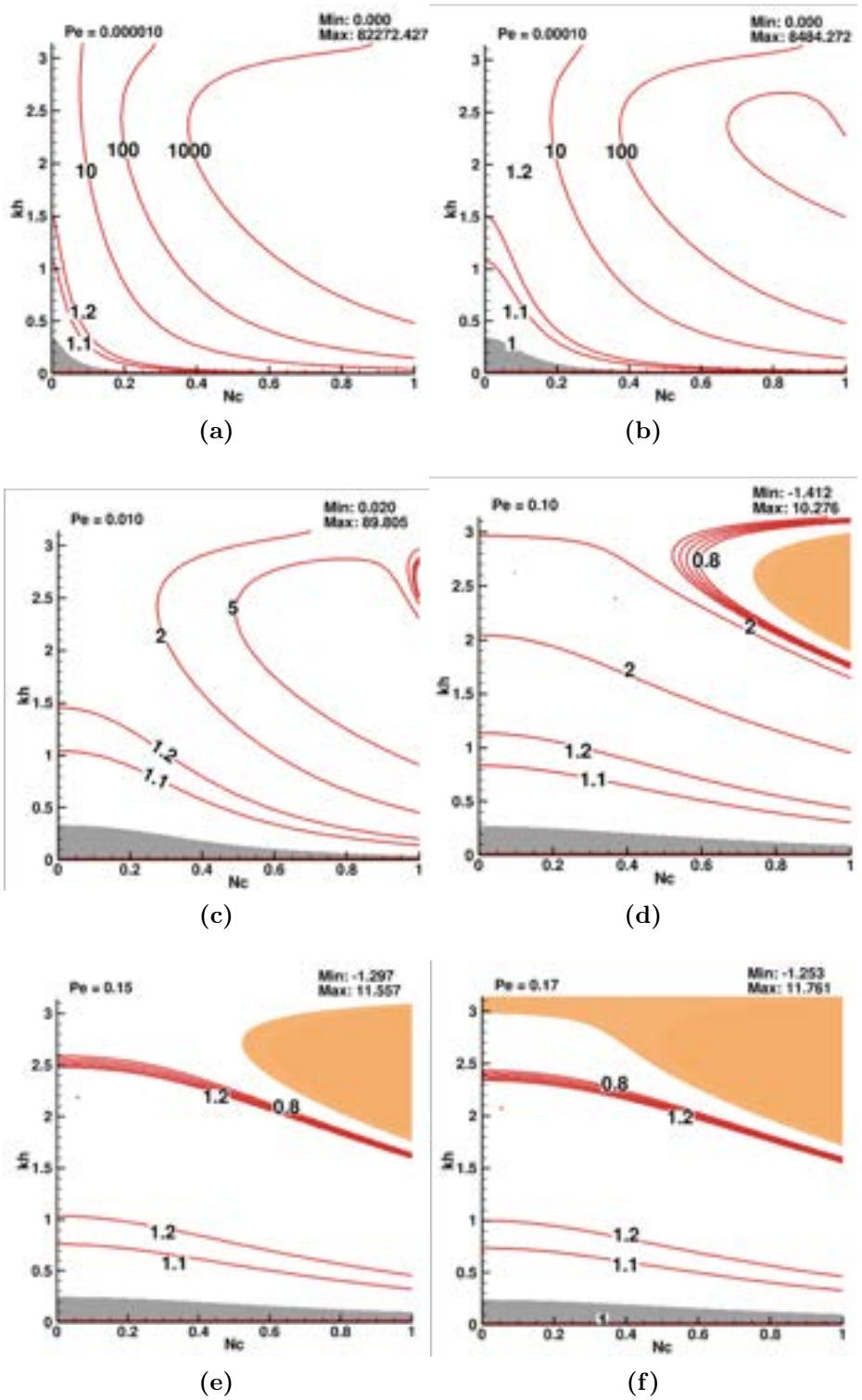


Figure 2.5: Effective measure of the numerical diffusion process, α_{num}/α in the $Nc - Pe$ plane for the TTGC-CD$_2$ scheme. Region of instability is identified in orange by $|G_{num}| \leq 1$; the grey region corresponds to the 1% DRP zone

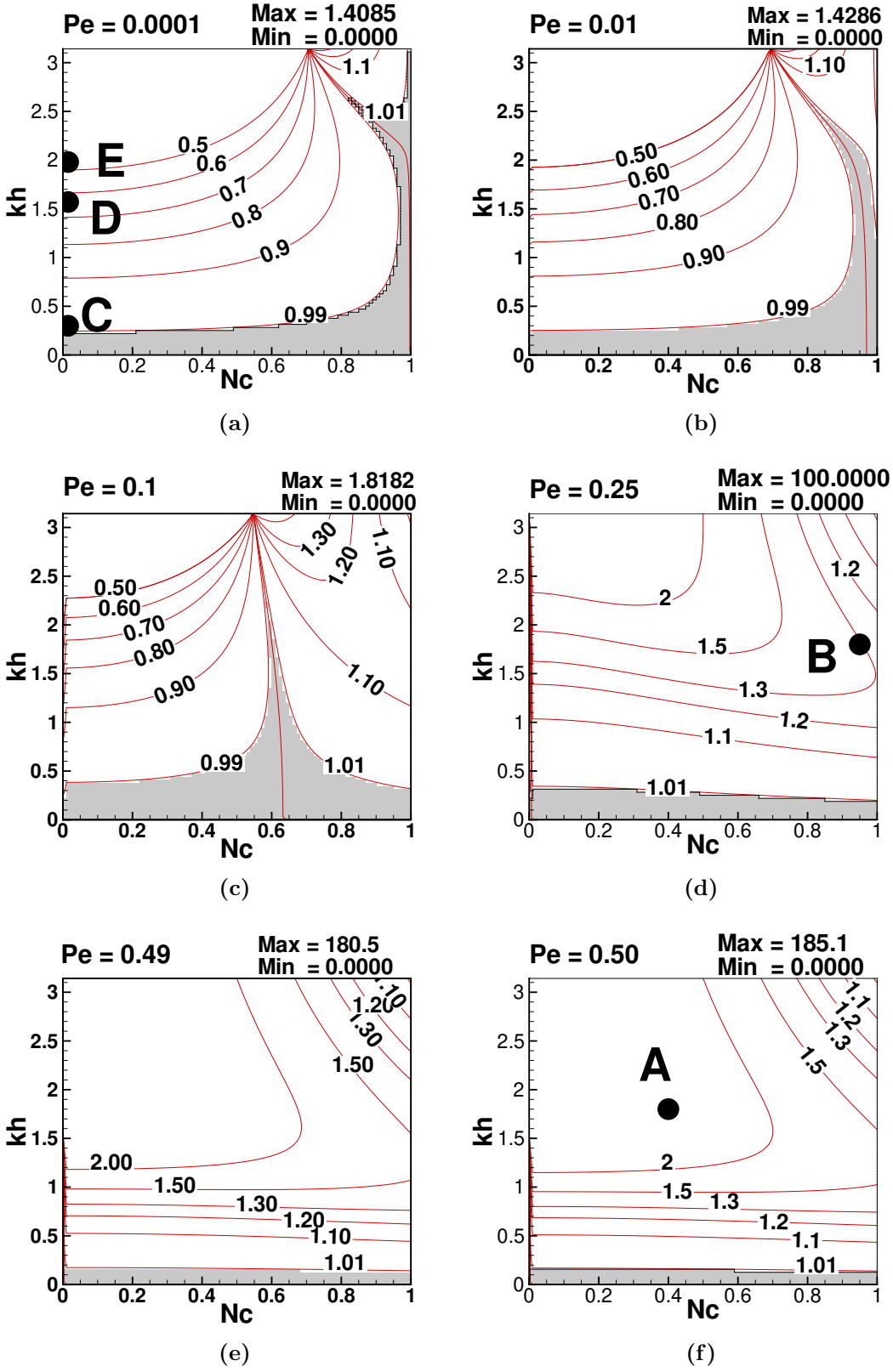


Figure 2.6: Contours of $\frac{c_{num}}{c_{phy}}$ as a function of kh and Nc for mentioned values of Pe for the LW-CD$_2$ scheme. DRP regions are shaded in grey. Five numerical test cases are marked (A to E) and explained later.

that the real part of the dispersion relation is linear in k which is not guaranteed by numerics. In that respect, looking at the group velocity speed is more constraining than looking at the phase speed alone. Therefore, in the rest of the discussions, the dispersive property of the scheme is solely analyzed using group velocity $V_{g,num}$ plots. As a result, to compare the dispersive property of or the LW-CD_2 scheme and TTGC-CD_2 scheme, the numerical group velocity of both the schemes are shown.

Figure. 2.7 shows the non-dimensional group velocity of the scheme in the Nc-kh plane for different values of Pe for the LW-CD_2 scheme. Similarly to the previous discussions, the DRP regions are identified by a tolerance of 1% deviation and shown by the grey shaded regions. The non-dimensional group velocity exhibits similar behavior as the non-dimensional phase speed in terms of its variations with kh and Nc for all the Pe values shown. The strong resemblance in the DRP regions of both the phase speed and group velocity plots are clearly evident. At very low values of Pe, Fig. 2.7 (a) & (b), negative values of group velocity (q-waves) [87] are observed for high wavenumbers ($kh > \pi/2$) and for $Nc < 0.7$. Computations performed in these regions will produce upstream propagating, non-physical waves. While this may not be of great concern at higher values of Nc due to the low values of $|G_{num}|$, which will contribute to a significant damping of these waves, they can be detrimental when simulations are performed at low Nc values since these waves can propagate upstream remaining undamped, potentially resulting in non-physical flow events. As Pe increases, Fig. 2.7 (c) & (d), the region of q-waves shrinks to finally disappear from the Nc-kh plane around $Pe = 0.25$. Similar observations apply to the DRP region, that remains quite narrow: *i.e.* for $kh < 0.5$ for all Nc values, with the region decreasing in size with increasing Pe.

Similarly, Fig. 2.8 shows the non-dimensional group velocity of the TTGC-CD_2 scheme in the Nc-kh plane for different values of Pe. At very low values of Pe, Fig. 2.8 (a) & (b), negative values of the group velocity (q-waves) are observed for high wavenumbers ($kh > 2.2$) and when $Nc < 0.8$. The region of the q-waves remains similar up to $Pe = 0.01$. As Pe further increases, Fig. 2.8 (d) & (e), the region of q-waves changes shape and finally settles for all wavenumbers $kh > 2.2$ for higher values of Pe. In this case, as the value of Pe increases, the q-wave region in the $kh - Nc$ plane doesn't disappear as it does for the LW-CD_2 scheme. As for the DRP region, it changes slightly with increasing Pe all through. This variation in DRP region is not significant for this numerical scheme compared to the LW-CD_2 scheme.

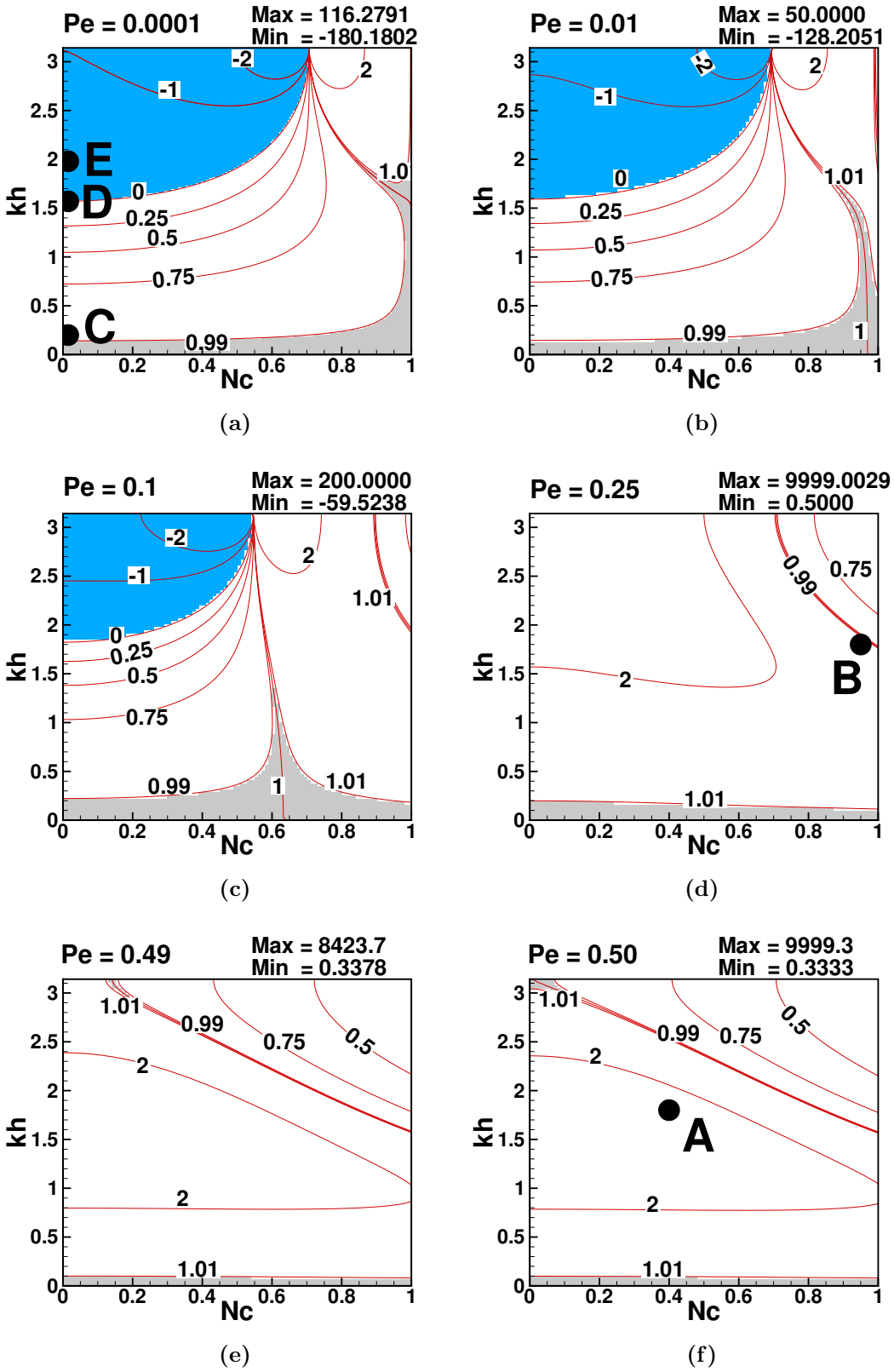
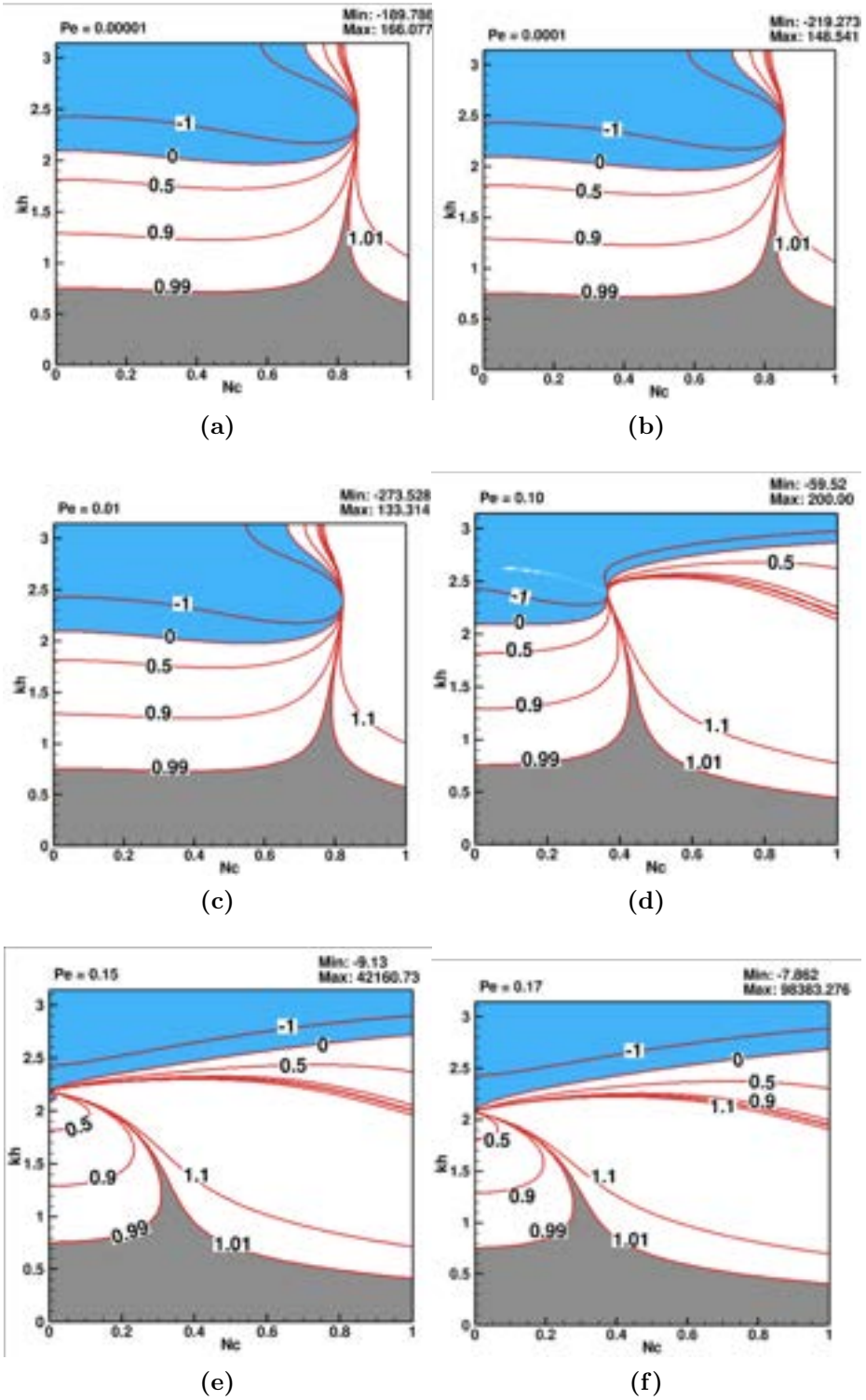


Figure 2.7: Contours of $v_{g,num}/v_{g,phy}$ as a function of kh and Nc for mentioned values of Pe for the LW-CD$_2$ scheme. The regions of q-waves are colored in blue, while DRP regions are shaded in grey. Five numerical test cases are marked (A to E) and explained later.



(a) (b) (c) (d) (e) (f)

Figure 2.8: Contours of $v_{g,num}/v_{g,phy}$ as a function of kh and Nc for mentioned values of Pe for the TTGC-CD_2 scheme. The regions of q-waves are colored in blue, while DRP regions are shaded in grey.

To conclude the numerical analysis, the GSA analysis of the scheme indicates that the scheme can be unstable over a large range of wavelengths and such a property is strongly affected by both parameters Nc and Pe. Also and as anticipated, the quality of a simulation, even if stable, is a function of all these parameters. The next sub-section illustrates all above findings through dedicated numerical experiments to confirm above conclusions and observations. The same numerical experiments for the 1D LCDE are performed for both the schemes, however, the results of the LW-CD_2 scheme are reported here only, as both schemes follow the property charts and numerical experiments faithfully.

2.2.4 Numerical experiments:

To illustrate the findings from the previous section, the convection-diffusion equation, Eq. (2.1), is numerically solved in the following using the LW-CD_2 scheme. The same numerical experiments presented here, were performed for the TTGC-CD_2 scheme and showed the same trends and conclusions. As a result, only results for the LW-CD_2 scheme are shown.

For the chosen numerical problem, the one-dimensional computational domain arbitrarily extends from $x = -50$ to $x = 50$ and is uniformly discretized using 200 grid elements. To be in agreement with the previous numerical analysis, spatially periodic boundary conditions are enforced on the domain boundaries and the constant c is fixed and given the value of 1.0. The initialization of u is prescribed to be a wavepacket of the form given by Eq. (2.30) for which σ is a parameter used to fix the width of the wavepacket (fixed to 0.05) and x_0 is the location of its center initially fixed at $x = x_0 = 0$,

$$u(x) = e^{-\sigma(x-x_0)^2} sin(k_0\, x). \tag{2.30}$$

Numerical tests are carried out to highlight the numerical properties of the scheme found previously and their variations with Nc, Pe and kh. The values of Nc and Pe are specifically chosen (Cases A to E) in each of these tests to highlight behaviors identified from the property charts detailed and discussed in the previous sections. Note that in terms of procedure, the diffusion coefficient α and the time step, Δt, are fixed once Nc, Pe, h and c are chosen. All the test cases explained in this section are detailed in Table. 2.1 and explained thereafter.

Numerical instability is often described in the literature as the sudden amplification of signals at the Nyquist frequency although as discussed earlier this can arise at other wavelengths. In the following, two cases are specifically detailed to

Case number	k_0h	Nc	Pe
A	1.80	0.40	0.50
B	1.80	0.95	0.25
C	0.30	0.0057	0.0001
D	1.57	0.0047	0.0001
E	1.98	0.0094	0.0001

Table 2.1: List of numerical cases analyzed with its important parameters. These cases (A to E) were marked in the property charts in the previous section.

illustrate the appearance of the instability at the Nyquist frequency, **Case A** for $Pe = 0.5$: *i.e. at the diffusive limiting value*, and an instability that arises at a frequency other than the Nyquist limit, **Case B** for $Pe = 0.25$, theoretically stable from a purely diffusive point of view.

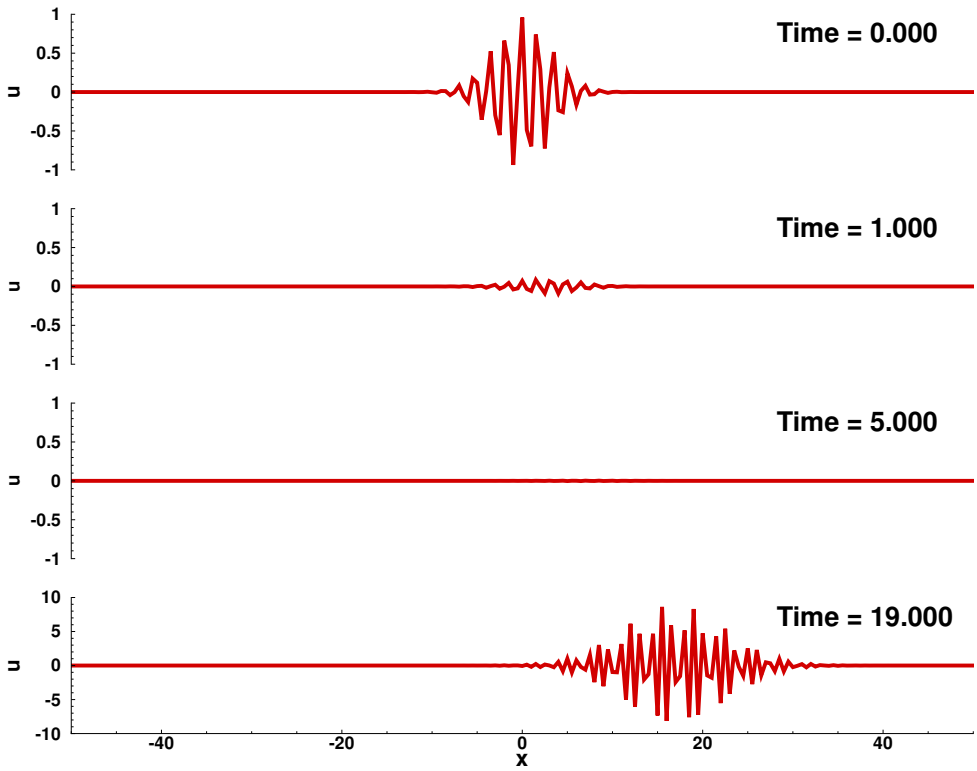


Figure 2.9: Numerical solution u as a function of the spatial coordinate for Case A at four mentioned time instants. ($Pe = 0.5$, $Nc = 0.4$, $kh = 1.8$)

The numerical solution, u, as a function of the spatial coordinate for **Case A** and at four mentioned time instants is shown in Fig. 2.9. In this case, the numerical solution is seen to be damped to a tenth of its initial amplitude by $t = 1$ and completely vanishes by $t = 5$. This is in accordance with the direct GSA prediction for the particular value of Nc, Pe and kh which is far from the DRP

region, as marked in Figs. 2.1& 2.4. However, as the simulation proceeds, the presence of random round off errors emerge. These errors occur primarily at the Nyquist limit and since this specific wavelength is always unstable as evidenced from Fig. 2.3, numerical instability arises at the indicated times. This test case demonstrates that although the initial solution falls in the stable amplification region and is eventually damped, over long time integration it leads to instability despite the presence of diffusion.

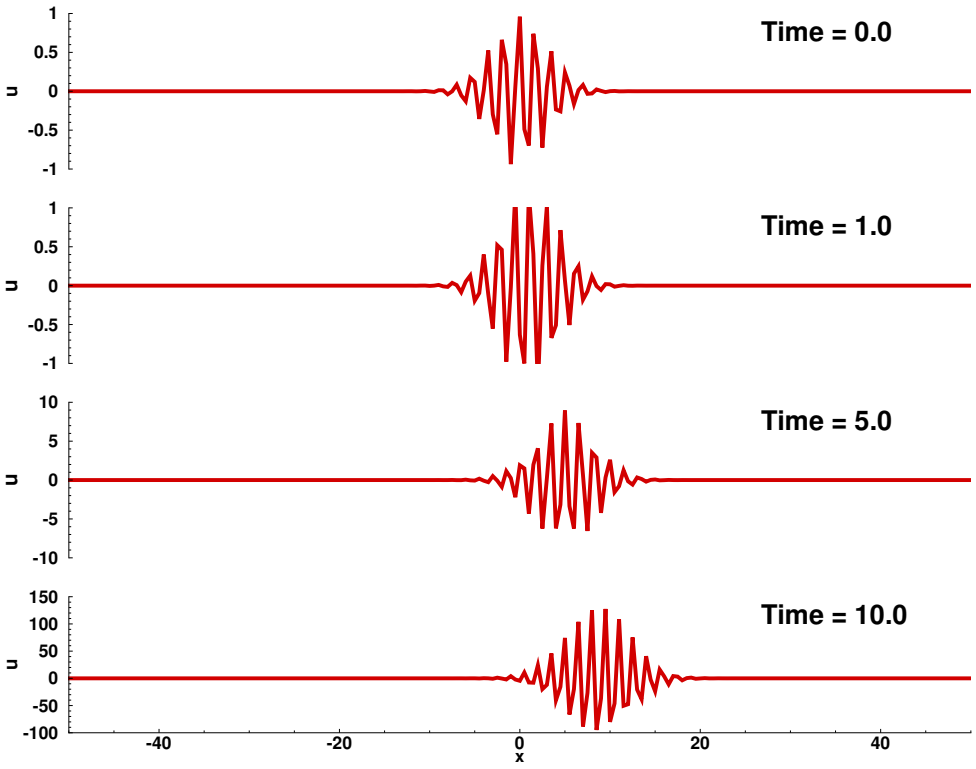


Figure 2.10: Numerical solution u as a function of the spatial coordinate for Case B at four mentioned time instants. ($Pe = 0.25$, $Nc = 0.95$, $kh = 1.8$). Please note the change of scale of the sub-figures.

Case B corresponds to values of $G_{rel} = 2.63$ and $|G_{num}| = 1.17$. Since $|G_{num}| > 1$, numerical instability should arise despite the fact that the CFL condition of LW is satisfied as is the purely diffusive limit. Note that the physical amplification factor for this case is 0.448 indicating that the exact solution is damped as expected although the numerical solution is amplified. Figure 2.10 confirms the findings from GSA; the solution of **Case B** is continuously amplified and is indicative of an instability arising away from the Nyquist limit and due to the improper interaction between the convective and diffusive discrete schemes. Such a behavior is clearly dependent on the chosen (Nc, Pe) combination and although a diffusive and stabilizing term is added to the problem, it does not

guarantee stability of the simulation as highlighted here. In this case, the last time instant shown is $t = 10$ and for which the wave packet has amplified significantly in amplitude compared to **Case A** is clearly noted in the change of scale needed for the vertical axis.

Next, **Cases C, D and E** are analyzed as follow ups to the previous set of numerical test cases which addressed specifically the notion of numerical stability. The following cases instead concentrate on the notion of group velocity. As mentioned previously, although the convection velocity is of interest to qualify the quality of a numerical prediction, enforcing a proper group velocity is a stronger condition: *i.e.* minimized error in the convective velocity does not guarantee for a proper energy propagation speed especially for a wide range of wavelengths. This is even more critical as most problems are highly rich signals which result in complex energy speeds at least numerically. The objective of the following discussion is therefore to validate the group velocity property charts for the LW-CD_2 scheme. **Case C** represents the scenario where the numerical group velocity and the numerical amplification factor is close to unity: *i.e.* both conditions satisfying the DRP condition. In such a scenario, it is expected that the numerical and exact solution move at the same speed as well as have similar amplitudes as the wavepacket propagates in time. **Case D** corresponds to the scenario where the numerical amplification is further away from the desired value of unity *i.e.* satisfying the DRP requirement, while the numerical group velocity is near zero. The numerical wave is therefore not expected to move and should remain around its initial location $x = x_0 = 0$. The amplitude of the numerical wavepacket should nonetheless decrease faster than the exact solution. In **Case E**, the group velocity has a negative value and the numerical amplification factor is much smaller than unity. The wavepacket should hence move in the opposite direction compared to the exact solution while being attenuated in time.

Figure 2.11 exposes all spatial resolutions issued by all three test cases in comparison to the respective exact solutions at $t = 20$. As observed in the respective GSA property charts, **Case C**, shows minimal error between the exact and the numerical solution. The second case, **Case D**, has a numerical group velocity close to zero as again confirmed by the numerical experiment, the exact solution having moved to the right. Note that the numerical solution has also a diminishing amplitude in time due to a smaller numerical amplification factor than in the exact solution. Finally, **Case E**, has been specifically constructed to evidence the so called q-waves which arise from high wavenumber components in simulations. For such waves, upstream group velocity propagation arises which is again confirmed by the prediction. Compared to the exact solution which

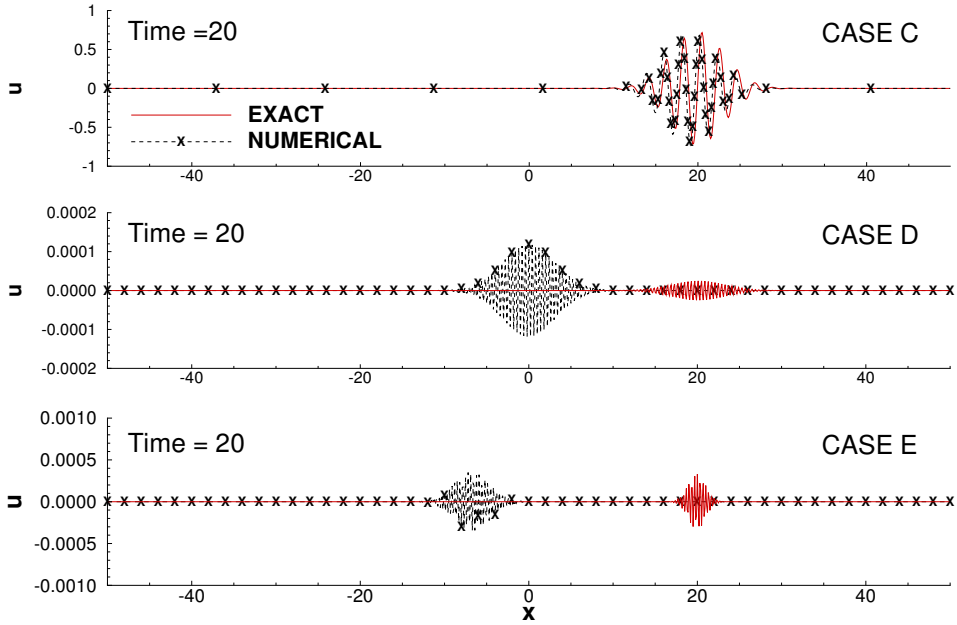


Figure 2.11: Exact and numerical wave-packet solution for Cases C, D and E to highlight the difference in numerical group velocity for LW-CD_2 scheme. Different ranges are chosen for the y-axis to highlight the differences in numerical amplification for the three cases which start with the same y-extent from the initial solution.

travels downstream, the numerical solution moves in the opposite direction and has an amplitude that is also significantly decreased when compared to the initial solution due to the numerical amplification factor that is much smaller than unity for these high wavenumbers. Once again, this test highlights how the choice of wavenumber, Nc and Pe play an important role in determining the numerical amplification factor as well as the numerical group velocity. This makes it clear that the choice of all the numerical parameters is crucial to obtain stable, accurate and correct numerical solutions. Next, the same three cases discussed previously **(Cases C, D and E)** are performed at a higher value of $Pe = 0.01$, to study its influence on the numerical properties.

Although Pe has changed, the overall behavior of the cases does not change compared to the previous study. Differences in numerical properties are present when Pe is changed as shown by Table 2.2. One however notes that no significant effect of Pe on the numerical group velocity is observed. By comparing the numerical amplification factor, differences are more visible at the two Pe values.

To conclude, this first series of tests confirms the GSA properties and maps detailed in the theoretical introduction. It furthermore highlights the paramount importance of understanding the effect of each parameter on the numerical properties which eventually affects the accuracy and stability of the numerical solution

	Gnum		Vg,num	
	Pe = 0.0001	Pe = 0.01	Pe = 0.0001	Pe = 0.01
Case C	0.9999	0.9991	0.9552	0.9578
Case D	0.9997	0.9799	0.0010	0.0216
Case E	0.9996	0.9719	-0.399	-0.3933

Table 2.2: Values of numerical amplification factor and numerical group velocity for the Cases C, D and E at two different Pe values, ($Pe = 0.0001, Pe = 0.01$).

when using particular schemes. Performing similar analyses in $2D$ is nonetheless mandatory for any use in a CFD context. Indeed, a $2D$ spatial discretization comes with the addition of two new parameters namely the spatial cell aspect ratio, $AR = \frac{h_y}{h_x}$ and the imposed wave propagation angle, $\theta = tan^{-1}(\frac{c_y}{c_x})$, which are detailed in the following section.

2.3 Extension to the linear $2D$ convection-diffusion problem

The same analysis as the one performed for the $1D$ convection-diffusion equation can be extended to $2D$, as detailed next. Similarly to the $1D$ problem, theoretical and numerical analysis are briefly recalled followed by the GSA analysis of the $2D$ numerical scheme. Numerical experiments are then performed to verify the findings.

2.3.1 Theoretical and numerical developments

The $2D$ convection-diffusion problem reads,

$$\frac{\partial u}{\partial t} + c_x \frac{\partial u}{\partial x} + c_y \frac{\partial u}{\partial y} = \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \tag{2.31}$$

Here, c_x, c_y are constants specifying the components of the convection speed in the x- and y-directions while α is the diffusion coefficient of the problem. Use of GSA essentially follows the details given for the $1D$ convection-diffusion equation with the exception that now expressions contain the vector representation of

the convection speed and wavelengths. Aside from these differences in terms of notations, the physical dispersion relation reads,

$$\omega = \vec{c}\,.\,\vec{k} - i\,\alpha\,||\vec{k}||^2 = (c_x\,k_x + c_y\,k_y) - i\,\alpha\,(k_x^2 + k_y^2). \tag{2.32}$$

It also results in a vector phase speed, $\vec{c}_\varphi = \omega\ \vec{k}/||\vec{k}||^2$, with complex components that are respectively,

$$c_{\varphi,x} = \frac{\omega\,k_x}{||\vec{k}||^2} = \frac{(c_x\,k_x + c_y\,k_y)\,k_x}{||\vec{k}||^2} - i\,\alpha\,k_x, \tag{2.33}$$

and,

$$c_{\varphi,y} = \frac{\omega\,k_y}{||\vec{k}||} = \frac{(c_x\,k_x + c_y\,k_y)\,k_y}{||\vec{k}||^2} - i\,\alpha\,k_y. \tag{2.34}$$

Likewise, the physical group velocity , $\vec{v}_g = \vec{\nabla}_k\,(\omega)$, has also two complex components,

$$v_{g,x} = \frac{\partial\omega}{\partial k_x} = c_x - 2\,i\,\alpha\,k_x, \tag{2.35}$$

$$v_{g,y} = \frac{\partial\omega}{\partial k_y} = c_y - 2\,i\,\alpha\,k_y. \tag{2.36}$$

From the above expressions, $\vec{c}_{phys} = \Re(\vec{c}_\varphi) = (\vec{c}\,.\,\vec{k})\,\frac{\vec{k}}{||\vec{k}||^2}$ and $\vec{v}_{g,phys} = \Re(\vec{v}_g) = \vec{c}$.

The physical amplification factor, G_{phys}, can then be expressed in terms of the x and y components of Nc and Pe as,

$$G_{phys} = e^{-[Pe_x\,(k_x\,h_x)^2 + Pe_y\,(k_y\,h_y)^2]}\,e^{-i\,[Nc_x\,(k_x\,h_x) + Nc_y\,(k_y\,h_y)]}. \tag{2.37}$$

Equivalently to the $1D$ problem, the numerical discrete correspondence can be called upon so that,

$$\omega_{num} = (c_{num,x}\,k_x + c_{num,y}\,k_y) - i\,\alpha_{num}\,(k_x^2 + k_y^2), \tag{2.38}$$

where $c_{num,x}$, $c_{num,y}$ and α_{num} are no more constants but depend on the nature of the discrete numerical scheme (*i.e.* numerical scheme but also the grid) adopted for the simulation. The numerical amplification factor can hence be represented as,

$$G_{num} = e^{-\alpha_{num}\,(k_x^2 + k_y^2)\,\Delta t}\,e^{-i\,(c_{num,x}\,k_x + c_{num,y}\,k_y)\,\Delta t}. \tag{2.39}$$

The numerical phase shift per unit time step follows,

$$tan(\beta_{num}) = -\frac{\Im(G_{num})}{\Re(G_{num})}, \tag{2.40}$$

where, $\beta_{num} = (c_{num,x}\, k_x + c_{num,y}\, k_y)\, \Delta t$.

Therefore, the non-dimensional numerical phase speeds in the x, y directions are,

$$\frac{c_{num,x}}{Re(c_{\varphi,x})} = \frac{c_{num,y}}{Re(c_{\varphi,y})} = -\frac{1}{Nc_x(k_x h_x) + Nc_y(k_y h_y)}\, tan^{-1}\left[\frac{\Im(G_{num})}{\Re(G_{num})}\right]. \tag{2.41}$$

The numerical group velocity can be calculated from the numerical dispersion relation using, $(v_{g,j})_{num} = \frac{\partial}{\partial k_j}(\omega_{num})$, which upon simplification results in the expressions,

$$\frac{(v_{g,x})_{num}}{Re(v_{g,x})} = \frac{1}{Nc_x}\frac{\partial \beta_{num}}{\partial (k_x h_x)}, \tag{2.42}$$

$$\frac{(v_{g,y})_{num}}{Re(v_{g,y})} = \frac{1}{Nc_y}\frac{\partial \beta_{num}}{\partial (k_y h_y)}. \tag{2.43}$$

The physical group velocity in the x- and y-directions are therefore for the present study equal to c_x and c_y respectively.

The numerical diffusion coefficient can be evaluated from Eq. (2.39) as,

$$|G_{num}| = e^{-\alpha_{num}(k_x^2 + k_y^2)\Delta t}. \tag{2.44}$$

By introducing the non-dimensional parameters (Pe_x, Pe_y, Nc_x and Nc_y), one gets,

$$ln|G_{num}| = -\left(\frac{\alpha_{num}}{\alpha}\right)[Pe_x(k_x h_x)^2 + Pe_y(k_y h_y)^2]. \tag{2.45}$$

The expression for the non-dimensional numerical diffusion coefficient is finally expressed as,

$$\frac{\alpha_{num}}{\alpha} = \frac{-ln|G_{num}|}{[Pe_x(k_x h_x)^2 + Pe_y(k_y h_y)^2]}. \tag{2.46}$$

The above analysis indicates the importance of two additional parameters that arise due to the $2D$ nature of the convection diffusion problem [88]. These are: the spatial aspect ratio, $AR = \frac{h_y}{h_x}$ for non isotropic meshes and the imposed wave propagation angle, $\theta = tan^{-1}(\frac{k_y}{k_x})$, or the convection velocity ratio, $\frac{c_y}{c_x}$. As detailed next, the above analysis when applied to a scheme gives access to this scheme's fundamental behavior.

2.3.2 GSA of the $2D$ LW-CD_2 scheme for the convection-diffusion equation

First, in the following, the LW scheme is used for the spatio-temporal discretization of the convective terms and the second order central difference (CD_2) scheme is used for the diffusive term. For the 2D analysis, in the case of TTGC-CD_2, the mass matrix M is tedious to handle in 2D, as a result, only the LW-CD_2 is used and presented. The numerical scheme when applied to any node (i, j) is thus given by,

$$\begin{aligned} u_{i,j}^{n+1} = u_{i,j}^{n} - \left[\frac{Nc_x}{2}(u_{i+1,j}^{n} - u_{i-1,j}^{n}) + \frac{Nc_y}{2}(u_{i,j+1}^{n} - u_{i,j-1}^{n})\right] \\ + \frac{1}{2}\,\zeta_x(u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}) + \frac{1}{2}\,\zeta_y(u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}), \end{aligned} \tag{2.47}$$

where, $\zeta_x = Nc_x^2 + 2\,Pe_x$ and $\zeta_y = Nc_y^2 + 2\,Pe_y$

Following the same procedure as in section 2.2.3, the numerical properties are,

$$\begin{aligned} G_{num} &= 1 - i\,[Nc_x\,sin(k_xh_x) + Nc_y\,sin(k_yh_y)] \\ &\quad + \zeta_x(cos(k_xh_x) - 1) + \zeta_y(cos(k_yh_y) - 1) \\ \frac{(v_{g,x})_{num}}{c_x} &= \frac{(1 + B_x + B_y)\,cos(k_xh_x) + (A_x + A_y)Nc_x\,sin(k_xh_x)}{((1 + B_x + B_y)^2 + (A_x + A_y)^2)} \\ \frac{(v_{g,y})_{num}}{c_y} &= \frac{(1 + B_x + B_y)\,cos(k_yh_y) + (A_x + A_y)Nc_y\,sin(k_yh_y)}{((1 + B_x + B_y)^2 + (A_x + A_y)^2)} \\ \frac{\alpha_{num}}{\alpha} &= \left[\frac{-ln\left(\sqrt{(1 + \zeta_x(cos(k_xh_x) - 1) + \zeta_y(cos(k_yh_y) - 1))^2 + (A_x + A_y)^2}\right)}{(k_xh_x)^2Pe_x + (k_yh_y)^2Pe_y}\right], \end{aligned} \tag{2.48}$$

where,

$$\begin{aligned} A_x &= Nc_x\,sin(k_xh_x), \\ A_y &= Nc_y\,sin(k_yh_y), \\ B_x &= \zeta_x\,(cos(k_xh_x) - 1), \\ B_y &= \zeta_y\,(cos(k_yh_y) - 1). \end{aligned} \tag{2.49}$$

As mentioned previously, GSA of the $2D$ convection-diffusion equation on a uniform mesh results in an additional degree of freedom when compared to the $1D$ analysis: namely the angle made by the wave normal direction θ which satisfies $tan(\theta) = k_y/k_x$. Additionally, the numerical properties of the scheme now depend

on 4 parameters: Nc_x, Nc_y, Pe_x and Pe_y. Note also that by construction, Nc_x and Nc_y are not truly independent and are related by,

$$Nc_y = (AR)\,\gamma\,Nc_x, \tag{2.50}$$

where AR is the grid aspect ratio, $AR = h_x/h_y$ and γ is the velocity ratio defined by c_y/c_x. Similarly, Pe_y and Pe_x are related through,

$$Pe_y = (AR)^2\,Pe_x. \tag{2.51}$$

These four parameters can therefore be independently chosen provided that a velocity ratio γ and a grid aspect ratio AR are fixed. For the sake of simplicity and unless specified otherwise, in the following, γ and AR are both taken to 1.0: *i.e.* $Nc_x = Nc_y = Nc$ and $Pe_x = Pe_y = Pe$. The stability and amplification factor properties of the scheme as a function of the wave direction angle, θ are first addressed next.

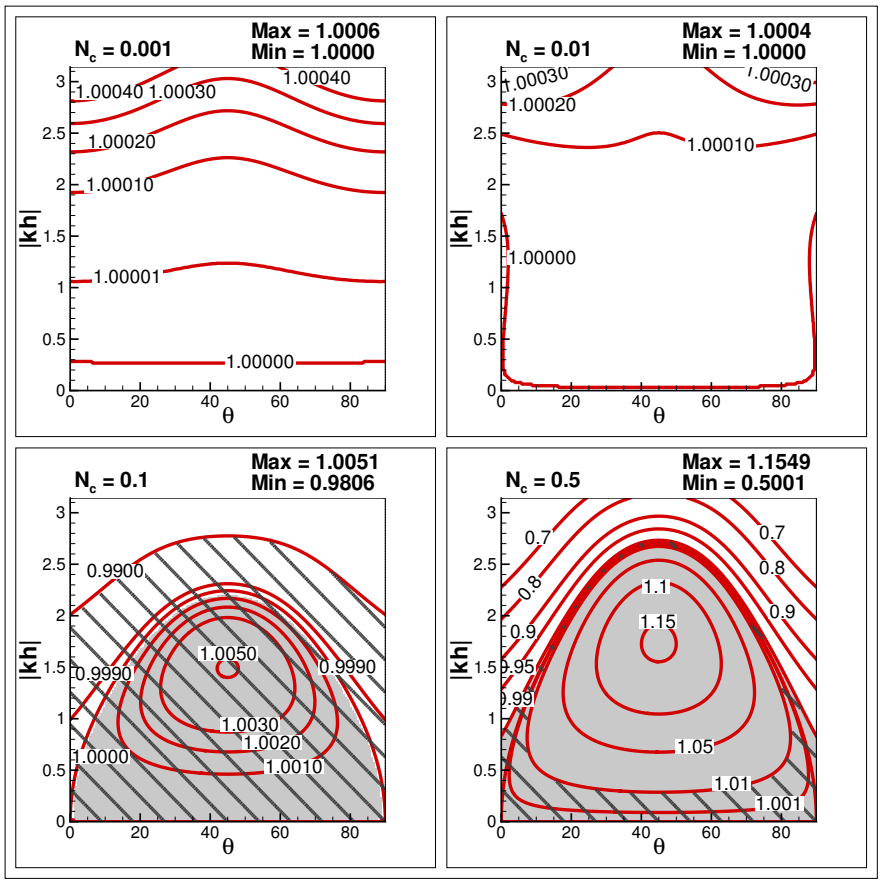


Figure 2.12: Effect of Nc on the contours of $\frac{|G_{num}|}{|G_{phys}|}$ for $Pe = 0.0001$. DRP region is hashed with the black solid line and the region of the instability ($|G_{num}| > 1.0$) is colored with light grey.

First, the effect of the wave direction angle, θ, on the numerical amplification factor for various wavenumbers and different values of Nc at $Pe = 0.0001$ are

shown in Fig. 2.12. Second, the same dependency but for a fixed $Nc = 0.1$ and various Pe is detailed in Fig. 2.13. In the following figures, the y-axis is $|kh| = \sqrt{(k_x h_x)^2 + (k_y h_y)^2}$. For all discussed results, note that the DRP region is hashed with black solid lines and the region of instability ($|G_{num}| > 1.0$) is shaded in light grey. As shown before, a region of instability ($|G_{num}| > 1.0$) coincides with a negative numerical diffusion or anti-diffusion ($\frac{\alpha_{num}}{\alpha} < 0$) as shown in the $1D$ GSA analysis. Finally, in the case of the present $2D$ analysis, by construction, the provided maps are symmetric with respect to $\theta = 45°$ as anticipated when using a uniform grid $h_x = h_y$ with $c_x = c_y$. Based on these various maps, it is evident that most of the deviations of the numerical solution from the exact solution appears along $\theta = 45°$. For small values of Nc (0.001 and 0.01), the amplitude of the relative amplification factor $G_{rel} = \frac{|G_{num}|}{|G_{phys}|}$ remains only slightly higher than 1 for all wavenumbers and all wave directions, θ. At these values of Nc, the DRP region is present for the entire range of theta and $|kh|$ values. However, for higher values of Nc (0.1 and 0.5), the effect on the numerical stability becomes more pronounced as θ approaches 45° even for small wavenumbers. As Nc increases, the DRP region is seen to shrink while the region of instability increases as shown in grey. High frequency wave components are however more damped with increasing values of Nc. For higher values of Nc, at small wavenumbers, instability is observed for all angles. The instability region peaks around $\theta = 45°$ and is observed to be least at $\theta = 0°$ and 90° degrees.

Figure 2.13 illustrates the effect of Pe on the stability of the numerical solution for $Nc = 0.1$. As Pe increases, the instability region shaded in grey shrinks and is present for a smaller range of θ and disappears for higher values of Pe, emphasizing the positive effect of diffusion on stability. At smaller values of Pe, instability is maximum at 45 degrees and least at 0 and 90 degrees. The deviation of the relative amplification factor (G_{rel}), is maximum at $\theta = 0°$ and 90° when $Pe = 0.1$. A larger DRP region is observed for all values of Pe around $\theta = 45°$. At $Pe = 0.1$, the DRP region is max for all wavenumbers and symmetric about $\theta = 45°$.

The dispersive properties of the scheme can be directly analyzed by studying the variations of $(v_{g,x})_{num}/c_x$ as shown in Fig. 2.14. Note that the variations of $(v_{g,y})_{num}/c_y$ are not shown since it is the symmetric of the x-component group velocity. Here, the value of Nc used is 0.1 and Pe values are changed. As Pe increases, the DRP region marked in solid hashed lines, remains almost similar and maximum at around $\theta = 90°$. For $Pe = 0.1$, the DRP region looks different from the other Pe maps, with a DRP region shifting to angles closer to 70 degrees. In all these cases, a DRP region covers all wave propagation angles but only for

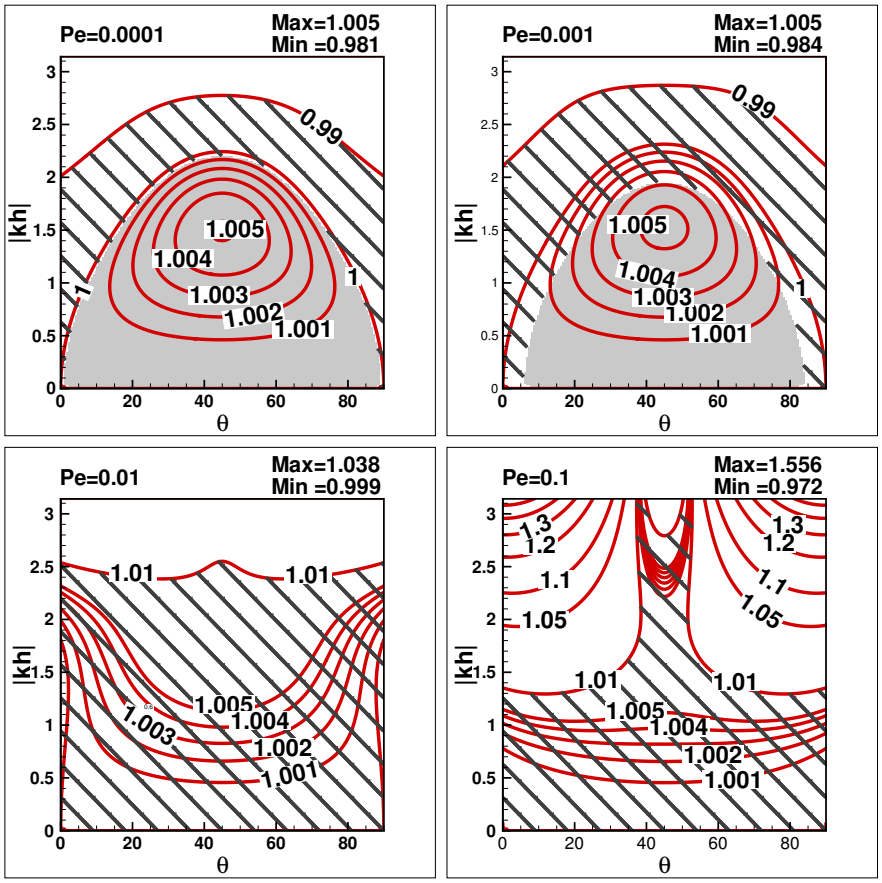


Figure 2.13: Effect of Pe on contours of $\frac{|G_{num}|}{|G_{phys}|}$ for $Nc = 0.1$. DRP region is hashed with the black solid line and the region of the instability ($|G_{num}| > 1.0$) is colored with light grey.

small wavenumbers. Finally, it is observed that the region of q-waves, where the group velocity is negative, is present for $\theta < 60°$ and as Pe increases, the q-wave region remains similar with negligible differences. Only for $Pe = 0.1$, can one see a significant reduction in the size of the q-wave region.

The effect of Nc on the dispersive properties are next analyzed by studying the variations of $v_{gx,num}/c_x$, as shown in Fig. 2.15. It is noted that the scheme satisfies the dispersion relation ($v_{gx,num}/c_x > 0.99$) for values of θ close to 90° and for all wavenumbers. For smaller values of θ, the DRP region shifts to a smaller range of wavenumbers. It is also noticeable that the region of q-waves is present for $\theta < 60°$. For lower values of Nc (0.001 and 0.01), the DRP region remains close to $\theta = 90°$ and as theta reduces, the DRP region shrinks to a lower range of wavenumbers. For $Nc_x = 0.5$, the DRP region is limited to small values of wavenumbers ($|kh| < 0.1$). Regarding the q-wave region, it shrinks with increasing values of Nc_x thereby reducing chances of upstream propagating high wavenumber phenomena.

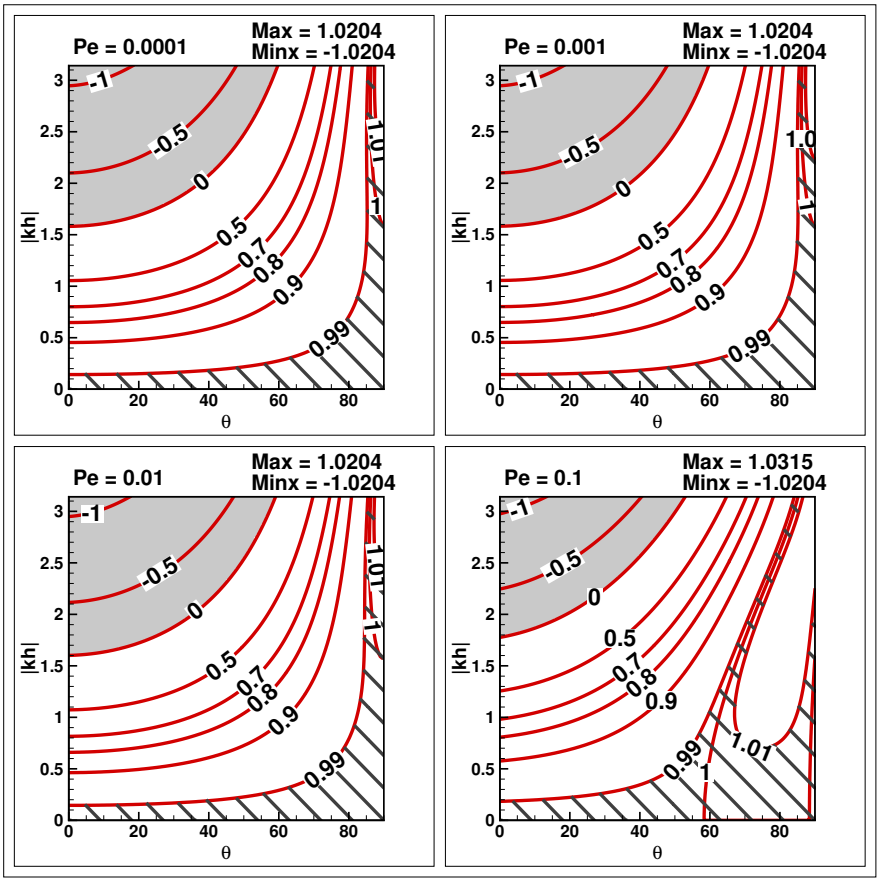


Figure 2.14: Effect of Pe on contours of $\frac{Vg_{num,x}}{cx}$ for $Nc = 0.1$. DRP region is hashed with the black solid line and q-wave region is colored with light grey.

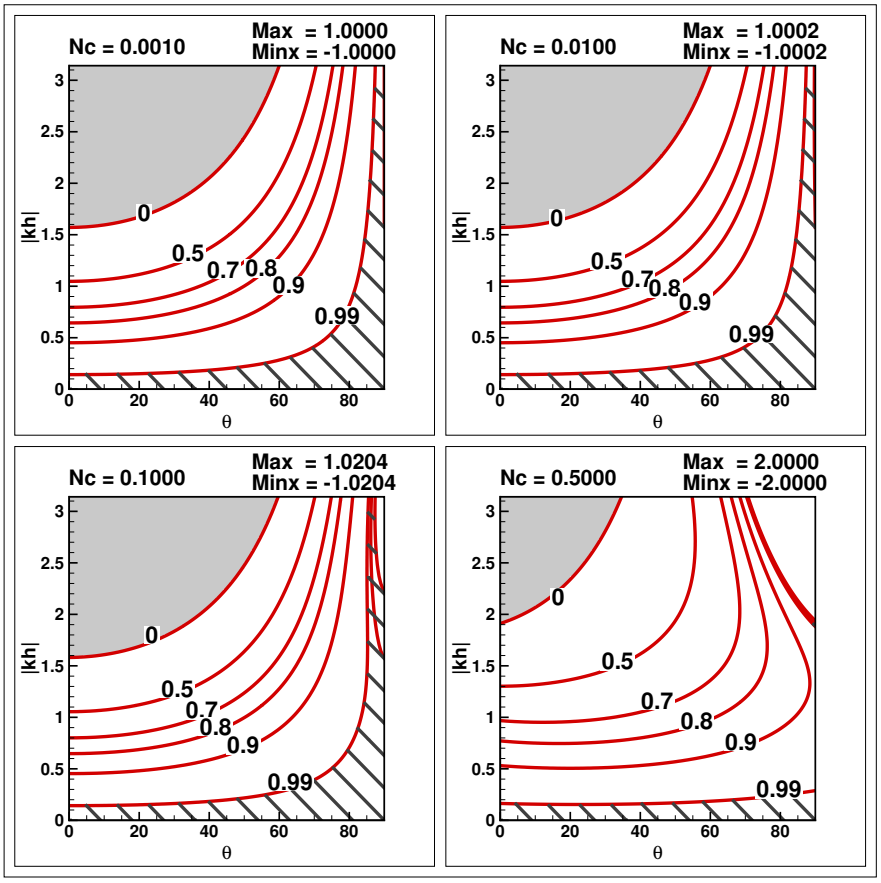


Figure 2.15: Effect of Nc on contours of $\frac{Vg_{num,y}}{cy}$ for $Pe = 0.0001$. DRP region is hashed with the black solid line and q-wave region is colored with light grey.

The primary information to retain from the previous $2D$ GSA is the identification of the stable regions of the LW-CD_2 scheme for Nc_x , Nc_y and Pe in the specific case of $AR = 1$ and $\gamma = 1$. The importance in the discussion around stability and its dependency on the wave propagation angle θ is shown in Fig. 2.16. The numerically unstable region corresponds to the white shaded region, where $|G_{num}| > 1$, while $|G_{num}| \leq 1$ indicated by blue shades corresponds to the stable region. Three specific θ values are here detailed for a mesh aspect ratio, $AR = 1$ in Fig. 2.16 and in the Nc and $Pe = Pe$ - planes. The plot emphasizes the fact that numerical instability indeed depends on θ for this specific choice of $AR = 1$. For lower Nc, the critical diffusive value of $Pe_{cr} = 0.50$ is recovered for all angles and this value decreases as Nc increases upto a certain limit denoted by $Nc_{cr} \approx 0.5$ which differs for the angles considered and certainly does not reach the $1D$ limit of pure convection: *i.e.* 1. Clearly, the wave propagation angle which shows the minimum critical value $Nc_{cr} = 0.5007$, appears for $\theta = 45°$. This minimal critical value will however correspond to stable computations for all other θ values since they have larger stability regions. The region shaded in blue shows that if $\theta = 45°$ is chosen, its satisfies the stability criterion for all other angles. This specific value is then used for simulating the Navier-Stokes equation in the last part of this chapter with all the 8 cases marked on the figure showing whether these cases lie in the stable or unstable region. Next, all above findings from the $2D$ GSA are validated through a $2D$ wavepacket problem. Here, again only LW-CD_2 scheme is used due to the tedious nature of the mass matrix in 2D for the TTGC-CD_2 scheme.

2.3.3 Numerical experiments

In order to validate the above observations obtained from the $2D$ GSA property charts, numerical simulations of the $2D$ convection-diffusion equation is carried out for a $2D$ propagating wavepacket problem. To do so, a square domain of side 100 units in length is chosen for the study, with periodic boundary conditions applied on all four sides. Both the velocity components c_x and c_y are chosen as unity so that $\gamma = 1$ and the domain is discretized using uniform, orthogonal quadrilateral elements with edge length of 0.5 units (so $h_x = h_y = 0.5$ and $AR = 1$). The problem is then initialized with a $2D$ wavepacket as follows,

$$u(x, y, 0) = e^{[-\sigma((x-x_0)^2+(y-y_0)^2)]}\, sin(k_x x + k_y y). \tag{2.52}$$

Here, σ determines the size of the wavepacket which is fixed at 0.05, k_x and k_y are obtained from the values of $|kh|$ and different values of θ are chosen for

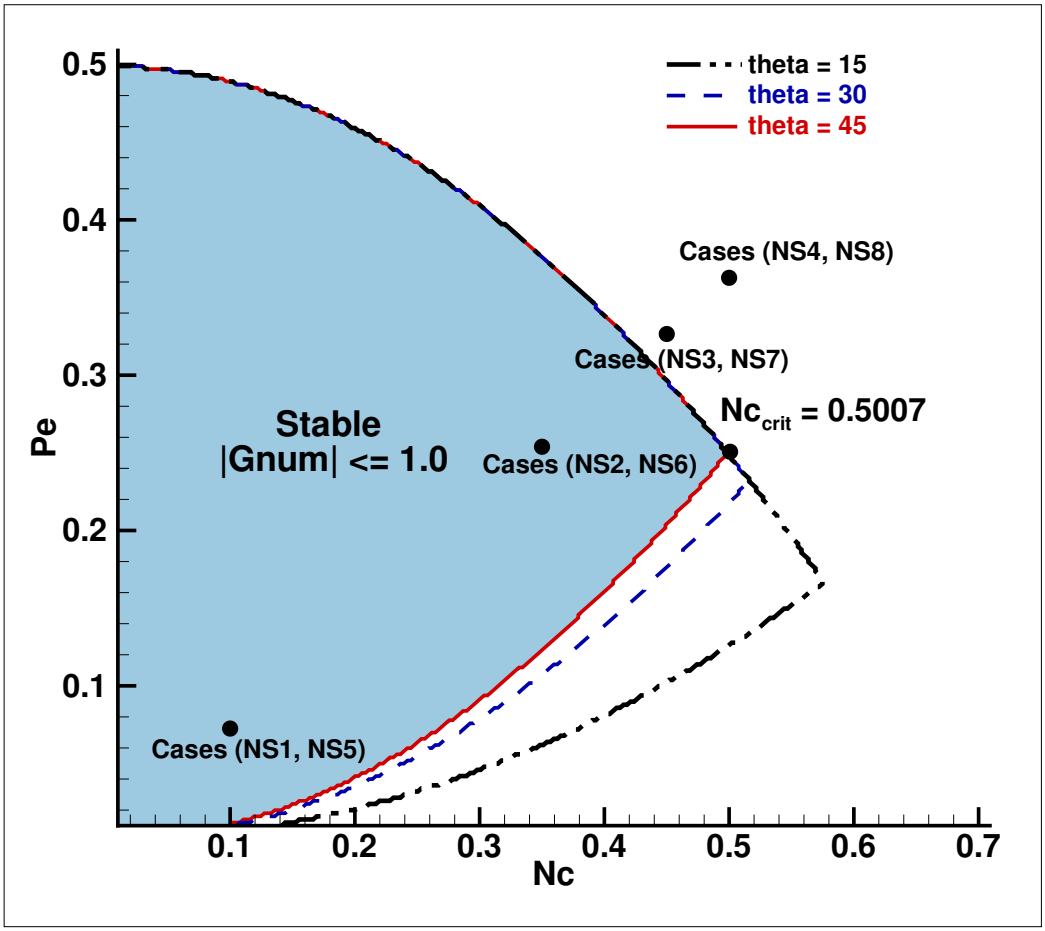


Figure 2.16: Influence of wave propagation angle θ on critical values of Pe and Nc for a grid with aspect ratio, AR = 1. All marked cases are solved later for the Taylor Green Vortex (Navier -Stokes equation) problem.

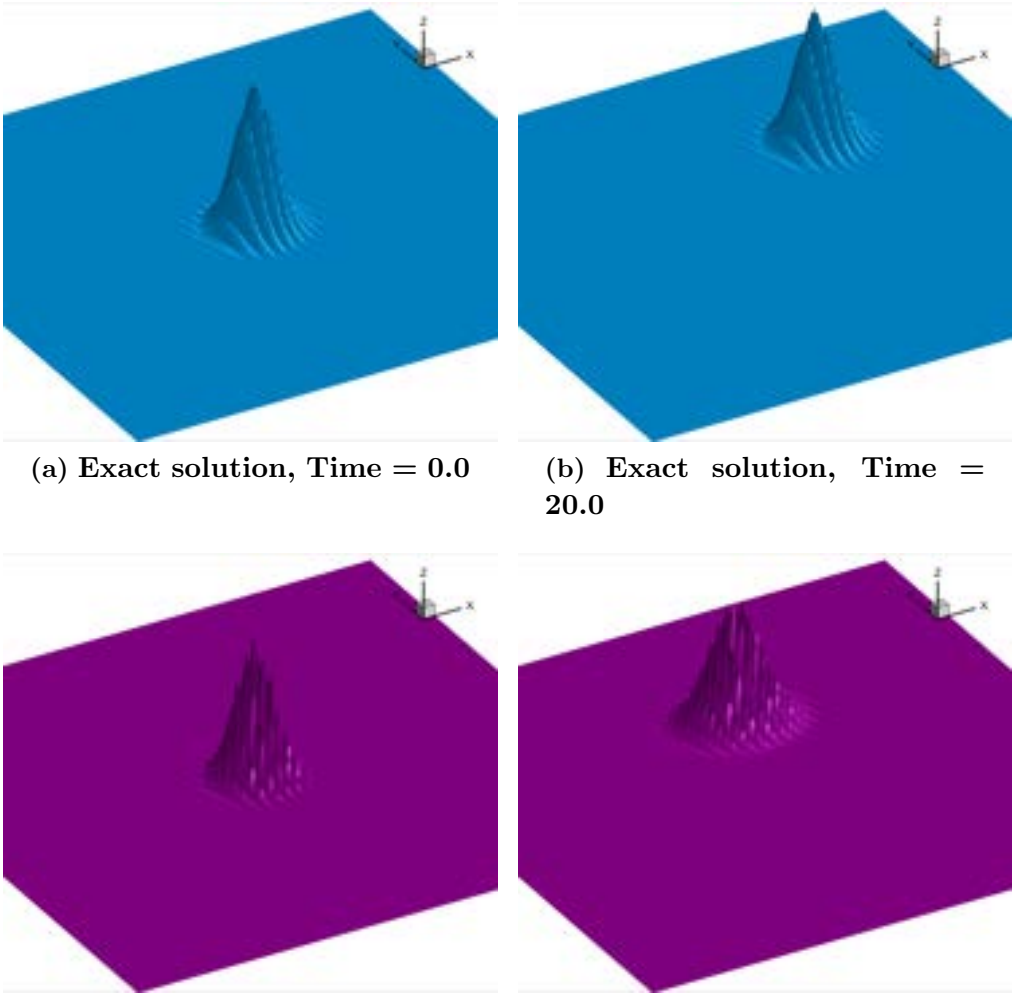
the study. To validate the analysis detailed in the previous sub-section, two simulations are carried out. The value of $|kh|$ is set to 1.5 for both the simulations and θ is fixed to 15° and 45° respectively. Nc_x and Pe_x are prescribed at 0.1 and 0.0001 respectively. These values are carefully chosen so that one computation ($\theta = 15°$) is stable while the other ($\theta = 45°$) is unstable, as can be seen in Fig. 2.12. For both simulations, the computation is carried out until a flow time of 20 units and the corresponding exact solution is obtained using discrete Fourier transform of the initial solution. The instantaneous solution is then obtained by time integrating the dispersion relation in the spectral space. A 4^{th} order Runge-Kutta scheme is used for this purpose.

Figure 2.17 shows the exact and numerical solutions for the $\theta = 15°$ test case at the two time instants of interest. The comparison of the numerical with the exact results for $\theta = 15°$ indicates that the magnitude of numerical wavepacket is in excellent agreement with that of the exact solution at $t = 20$, as the relative amplification factor for this case is close to unity.

Although the wave amplitude is accurately captured for the $\theta = 15°$ case, its numerical speed is incorrect compared to the exact solution. This specific error is also present for the numerical solution of the $\theta = 45°$ test case shown in Fig. 2.18 with the additional issue that here the simulation is unstable and the signal amplitude is amplified although the direction of propagation is better captured. The mismatch in the numerical and the exact wave speeds can be explained using Fig. 2.14. The (x, y) components of the numerical group velocity for the chosen values of Nc_x, Pe_x and $|kh|$, as predicted by GSA are $(0.134, 0.922)$ respectively for $\theta = 15°$ clearly highlighting the observed differences in propagation direction. These values are very close to those retrieved numerically $(0.135, 0.92)$ shown in Fig. 2.19. Similarly, the values of the two group velocity components obtained in the computed solutions ($V_{gx} = V_{gy} = 0.48$) are in good agreement with the values of their predicted counterparts as shown in Fig. 2.15 ($V_{gx} = V_{gy} = 0.49$). For the $\theta = 45°$ case, the wave propagation angle is captured accurately as shown by the white arrow but at approximately half the propagation speed of the exact solution. Both of these results emphasize the effect and importance of the wave direction angle and the numerical parameters (Nc_x and Pe_x) on the stability and dispersive properties of the scheme.

2.3.4 Links with a $2D$ Navier-Stokes CFD

CFD predictions have been produced in [86] to show the one-to-one correspondence between numerical anti-diffusion and focusing for unsteady flows inside a



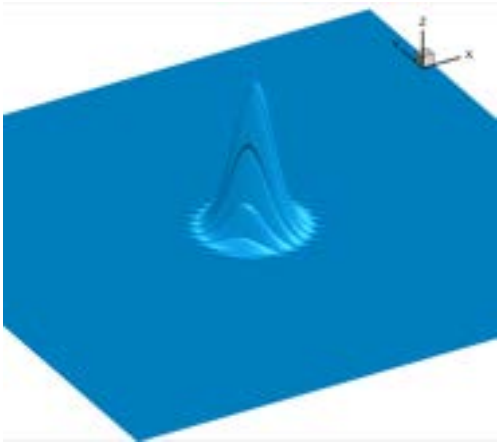
(a) **Exact solution, Time = 0.0**

(b) **Exact solution, Time = 20.0**

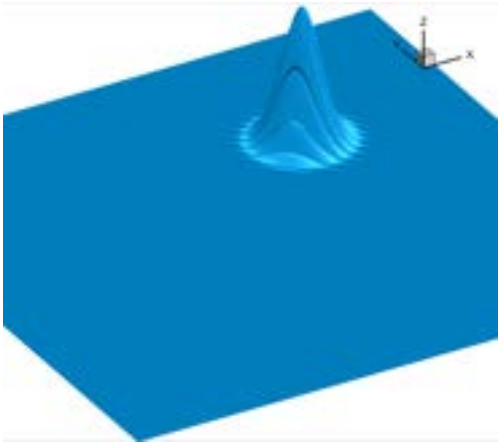
(c) **Numerical solution, Time = 0.0**

(d) **Numerical solution, Time = 20.0**

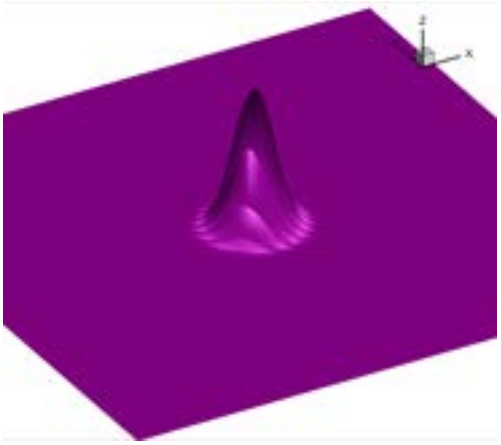
Figure 2.17: Exact and computed solutions of 2D convection-diffusion equation for $\theta = 15^{\circ}$, $Nc_x = 0.1, Pe_x = 0.0001$ and $|kh| = 1.5$ with LW-CD$_2$ scheme.



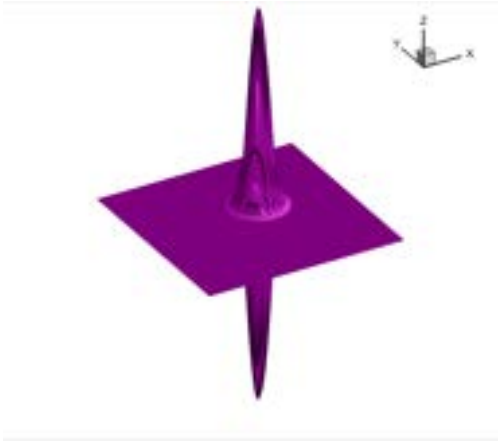
(a) **Exact solution, Time = 0.0**



(b) **Exact solution, Time = 20.0**

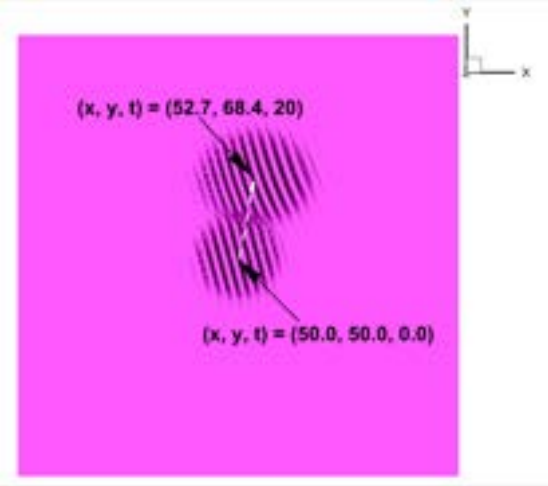


(c) **Numerical solution, Time = 0.0**

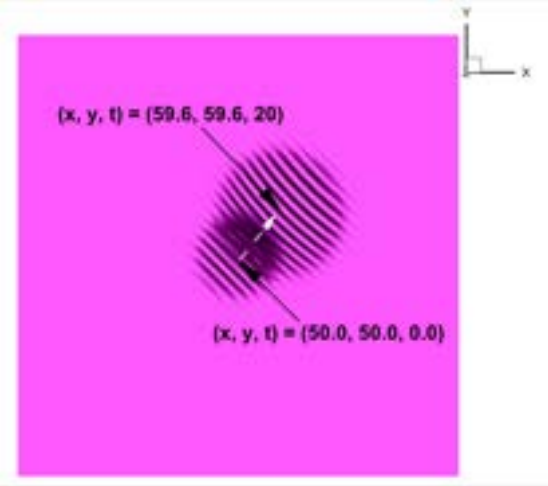


(d) **Numerical solution, Time = 20.0**

Figure 2.18: Exact and computed solutions of 2D convection-diffusion equation for $\theta = 45^\circ$ $Nc_x = 0.1, Pe_x = 0.0001$ and $|kh| = 1.5$ with LW-CD$_2$ scheme.



(a) Exact solution at $t = 0$ and $t = 20$, $\theta = 15^{\circ}$



(b) Exact solution at $t = 0$ and $t = 20$, $\theta = 45^{\circ}$

Figure 2.19: Z-view of the numerical solutions of 2D convection-diffusion equation for $\theta = 15^{\circ}$ and 45°. $Nc_x = 0.1, Pe_x = 0.0001$ and $|kh| = 1.5$

2D lid driven cavity (LDC) for a Reynolds number of 10,000. This specific process was shown by the discretization of the diffusion term by the chosen numerical scheme. Based on this specific experience, the Taylor-Green vortex (TGV) problem is addressed hereafter to link retrieved information from GSA with the specific context of a CFD simulation.

The TGV problem is a commonly used flow test case for CFD solver validation. In this problem, multiple pairs of vortices are superimposed on a viscous fluid in a square periodic domain with a fixed initial velocity and pressure fields. This particular test case is chosen for the following reasons. Its $2D$ nature makes computations cheaper while relevant with the findings of our 2D GSA analysis. Finally, the absence of boundary condition helps isolating the nature of the numerical scheme alone.

For the specific problem addressed here, the following initial solution is enforced,

$$u(x, y, 0) = U_0 sin(\frac{2\pi x}{L}) cos(\frac{2\pi x}{L}) \tag{2.53}$$

$$v(x, y, 0) = -U_0 cos(\frac{2\pi x}{L}) sin(\frac{2\pi x}{L}) \tag{2.54}$$

$$p(x, y, 0) = \frac{\rho}{4}(cos(\frac{2\pi x}{L}) + sin(\frac{2\pi x}{L})). \tag{2.55}$$

Here u, v and p are the x, y velocity components and pressure respectively while the density ρ is kept constant. The dynamic viscosity of the fluid (μ) is also kept constant. The constant velocity U_0 is provided in Table 2.3 along with other parameters of importance for the cases considered. The $2D$ TGV problem on a square domain in the cartesian $x - y$ plane, bounded by $0 \leq x, y \leq L$ is chosen. Since, the GSA results are valid for any chosen grid size, and to have sufficient flow resolution, the domain is discretized using 512×512 square cells. The Reynolds number for the cases is thus defined as,

$$Re = \frac{\rho U_0 L}{\mu}. \tag{2.56}$$

For the present study, Re of 10 and 100 are chosen to show that increasing the Re has no effect on the stability limits when applied to the 2D NSE. These values of Re furthermore ensure that the solution is in the linear regime. This is a diffusion dominated problem where the vortex pairs in the domain are non-interacting and the strength of the vortices continuously decays with time. The objective of choosing these specific test cases are to verify the GSA stability plots as shown in Fig. 2.16 for the 2D convection-diffusion equation applied to the 2D Navier-Stoked equations. In order to validate the GSA predictions and to

demonstrate its independence with respect to Reynolds number, four different combinations of N_c and Pe are chosen on Fig. 2.16 for the two different Re. The $N_c - Pe$ combinations, Re and the calculated maximum values of numerical amplification factors are recapitulated in Table 2.3.

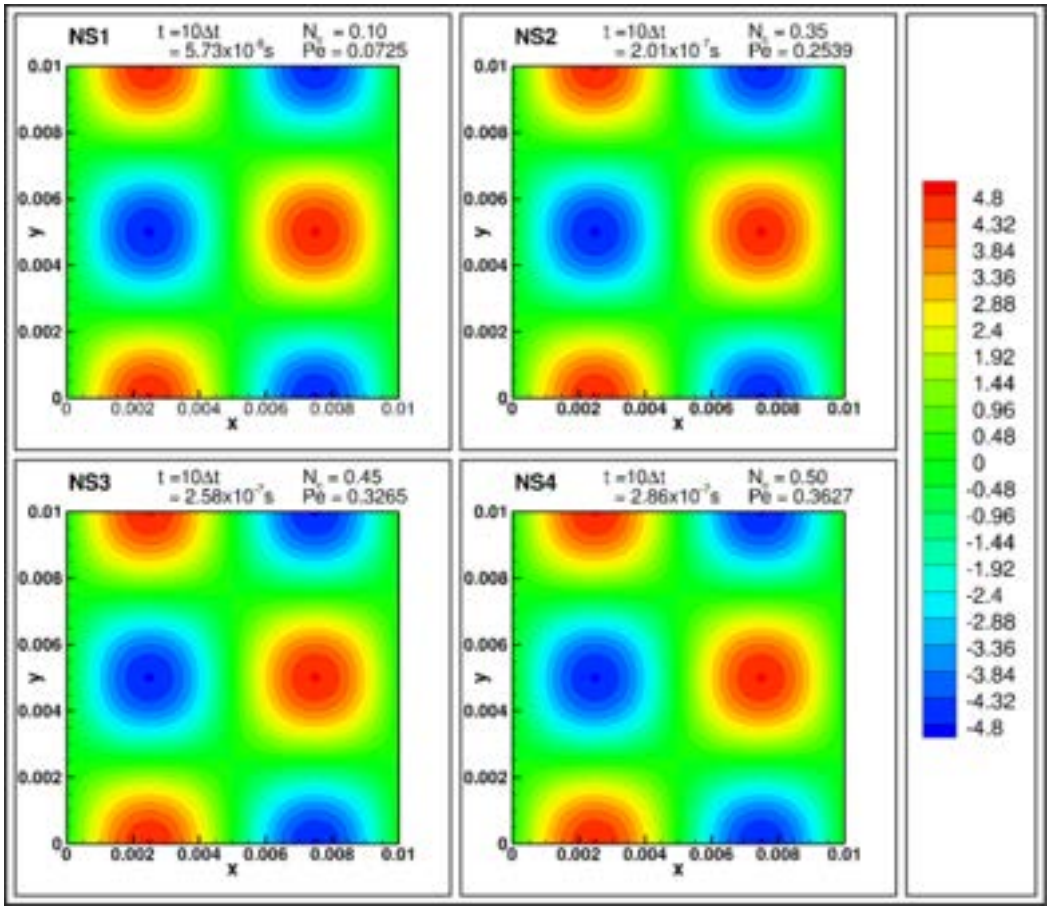


Figure 2.20: Contours of horizontal velocity components after 10 **time steps for cases NS1-NS4 from Table 2.3. The** Nc **and** Pe **numbers corresponding to each case are shown on the top right corner.**

All the cases in this section are named as NS1-8 (Navier-Stokes cases 1 to 8 as listed in Table 2.3). These values are chosen from the GSA analysis so that the two $Nc - Pe$ combinations for each Re **(Cases NS1, NS5 and Cases NS2, NS6)** correspond to stable simulations (Max. $G_{num} = 1$) while the other cases **(Cases NS3, NS7 and Cases NS4, NS8)** are unstable (Max. $G_{num} > 1$). Note that, by fixing the values of Re, Nc and Pe, and choosing the grid size, the velocity scale gets fixed. In such cases, the domain size L needs to be scaled to ensure that the flow remains in the linear regime as well as incompressible as demanded by this TGV problem. For all the cases addressed numerically and shown in the present subsection, the cell aspect ratio $AR = 1$ and the numerical parameters specific to the wave propagation angle $\theta = 45°$ are prescribed as

Case number	Re	Nc	Pe	U_0	Max.G_{num}
NS1	10	0.10	0.0725	4.8277	1.00
NS2	10	0.35	0.2539	4.8305	1.00
NS3	10	0.45	0.3265	4.8314	1.11
NS4	10	0.50	0.3627	4.8303	1.45
NS5	100	0.10	0.0725	48.2766	1.00
NS6	100	0.35	0.2539	48.3051	1.00
NS7	100	0.45	0.3265	48.3136	1.11
NS8	100	0.50	0.3627	48.3032	1.45

Table 2.3: List of TGV cases analyzed with its important parameters.

explained earlier. Simulations of all 8 cases from Table 2.3 are then carried out using 48 computing processors till $t = 5$ seconds.

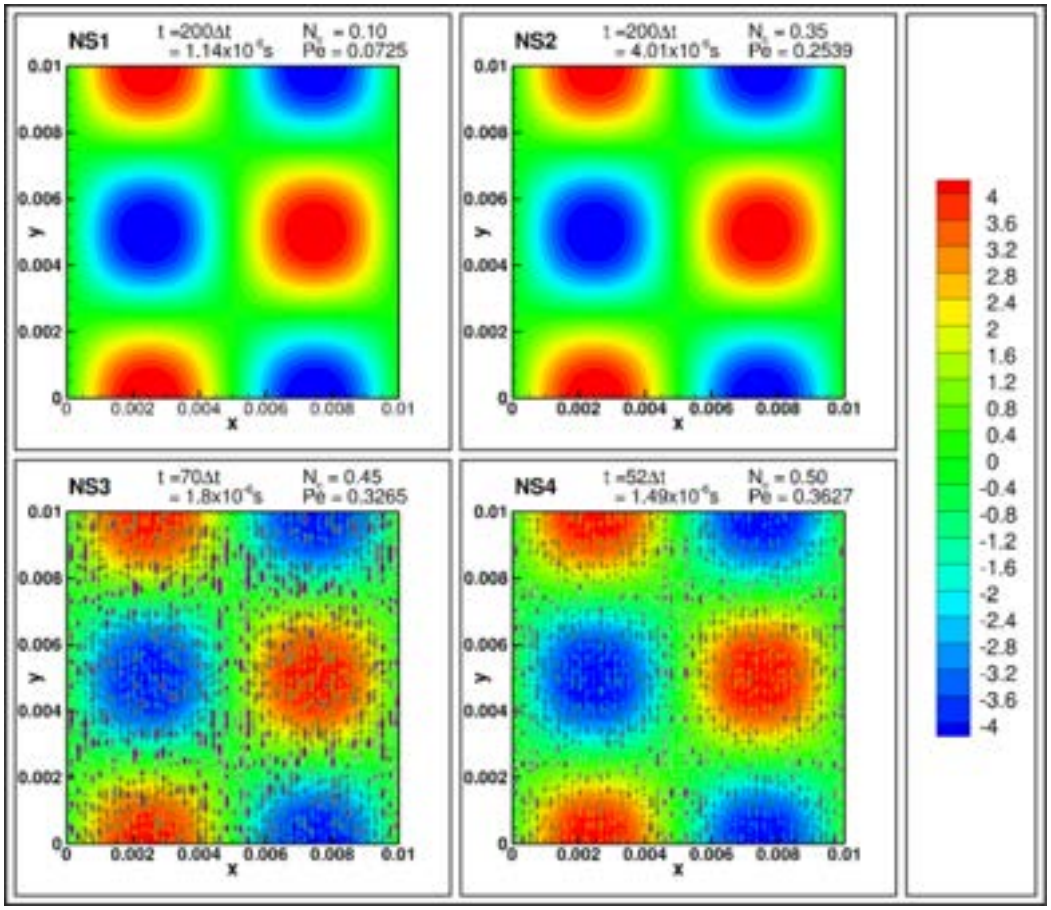


Figure 2.21: Contours of horizontal velocity components just before simulations crash for cases NS3-NS4 Cases NS1 & NS2 are observed to be numerically stable as predicted by GSA. The Nc and Pe numbers corresponding to each case are shown on the top right corner.

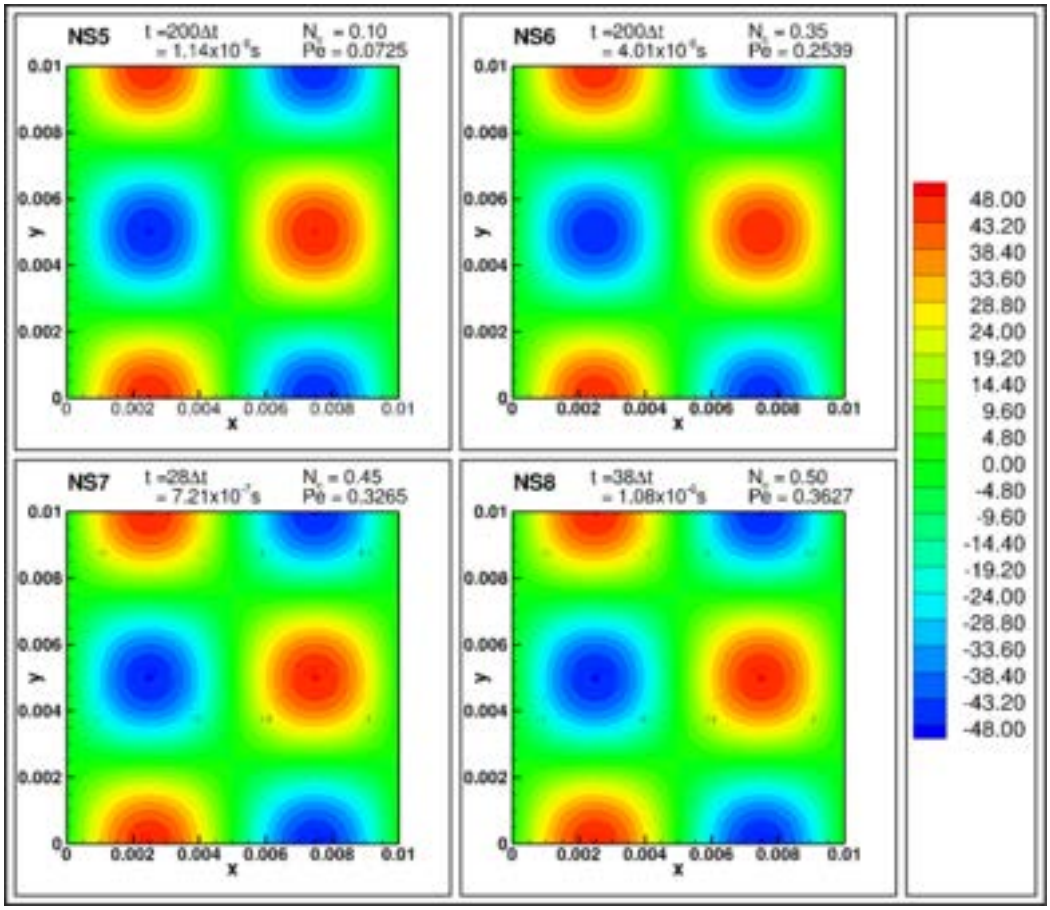


Figure 2.22: Contours of horizontal velocity components after 38 time steps for cases NS5-NS8. The Nc and Pe numbers corresponding to each case are shown on the top right corner.

For cases **Cases NS1-NS4**, for which $Re = 10$, iso-contours of velocity in the x-direction after 10 time steps (exact time indicated on each frame of cases NS1-NS4) are shown in Fig. 2.20. At this time instant, all the cases (both stable and unstable as predicted by GSA), do not show any indication of instability. At a later time instant, however, $t = 1.8$ x 10^{-6} s, (70 time steps), **Case NS3** corresponding to $Nc = 0.45$ and $Pe = 0.3265$ shows signs of numerical instability as observed in Fig. 2.21 leading to the solution blow-up. **Case NS4** with $Nc = 0.50$ and $Pe = 0.3627$ shows instability at an even earlier time, $t = 1.49$ x 10^{-6} s, (52 time steps) due to the higher value of Max. (G_{num}) for this case as reported by Table 2.3. The development of the numerical instability is exactly in accordance with the prediction made by the GSA analysis. The x-component velocity contours for all 4 cases just before the simulation blow up are shown in Fig. 2.21. **Case NS1 and NS2** results are shown at $t = 1.14$ x 10^{-6} and $t = 4.01$ x 10^{-6} respectively (200 time steps) to confirm their stability, while the results for **Case NS3** and **Case NS4** exhibit simulation blow up much earlier

as predicted by GSA. A similar observation is also made for **Cases NS5-NS8**, for which $Re = 100$. While **Cases NS5 and NS6** are observed to be stable till $t = 5s$, **Cases NS7 and NS8** develop an instability at $t = 7.21$ x 10^{-7} and $t = 1.08$ x 10^{-6} respectively (28 and 38 time steps) as shown by Fig. 2.22.

For both Re, the presence of instabilities for the higher Nc and Pe number combinations, is due to the error arising because of focusing which occurs at high wavenumbers. This focusing arises from the anti-diffusion process at the Nyquist limit where α_{num}/α is negative ($|G_{num}| > 1$) as shown in the $2D$ property charts of the earlier section. Note that, although the error is initiated at the Nyquist limit, as the solution proceeds, the error propagates to all wavenumbers which results in an eventual solution blow-up. This highlights that although the GSA was performed on the 2D linear convection-diffusion equation, the results from the analysis can be extended to the weakly non-linear Navier-Stokes equations. All the results shown here are for LW-CD_2 scheme, as mentioned before the mass matrix in 2D for TTGC is tedious to handle, as a result, it has been omitted for the present study.

2.4 Conclusions and Perspectives

Most academic and industrial codes often have issues when solving complex problems for a variety of reasons (bad resolution, wrong numerical parameters etc). To overcome this problem, a thorough analysis is required which is done via the help of Global Spectral Analysis. GSA analysis is introduced to the LCDE framework in 1D for the well know and widely used LW-CD_2 scheme. The property charts and expressions for stability, diffusion, dispersion are introduced in terms of non-dimensional parameters. The property charts for the LW-CD_2 and TTGC-CD_2 schemes are analyzed for the LCDE rather than convection and diffusion procedures studied separately. This resulted in different critical values of N_c that can be used compared to a pure convective process ($N_c = 1$). Well designed experiments show the presence of standing or stationary waves as well as q-waves. This highlighted the fact that the property charts are sensitive to the parameters like N_c, Pe, and kh.

The analysis is then extended to the 2D LCDE problem discretized using the LW-CD_2 scheme. Two additional important parameters have been introduced, namely the spatial aspect ratio, AR and the wave propagation angle, θ. The stability and dispersive properties of the scheme has been studied for varying Nc and Pe and shows a strong dependence on θ which is validated by numerically solving the 2D wavepacket propagation problem. Numerical focusing

due to anti-diffusion has been demonstrated for the unsteady, 2D incompressible Navier-Stokes equation, solved using LW-CD_2 scheme for the Taylor Green Vortex problem at two different Reynolds numbers. Such an extensive analysis of LCDE problem both in 1D and 2D highlights the benefits of GSA to better understand stability, diffusion and dispersion of a specific numerical scheme. This highlights the ability of such a powerful tool and the variety of analysis that can be performed accurately. Therefore, it is imperative to extend GSA for the linear Convection-Diffusion-Reaction Equation (LCDRE) for analysis and resolution of fully premixed reacting problems next.

Numerical analysis: Convection - Diffusion-Reaction equation

Contents

3.1 Introduction

In the present investigation, the canonical problem involving a single variable, whose evolution is governed by the simultaneous actions of convection, diffusion and reaction is analyzed. The importance of this equation is related to the fact that the results can be of relevance to multiple fields such as acoustics [89], astrophysics [90], various branches of mathematical physics related to pattern formation and nonlinear dynamics [91, 92, 93]. In the field of fluid dynamics, this equation has been used to develop the constitutive equations for the (k, ϵ)-turbulence models [79] or the study of the onset of convection with differentially heated fluid layers [94]. This equation has also been used to study the parametric

variation of the viscosity coefficient with temperature causing thermal convection which has applications on heat transfer [95], geothermal studies [96] or the formation of the Rayleigh-Benard roll pattern [97]. This equation has been finally used for many ecological studies of predator-prey dynamics in [98, 99, 100]. It is also of importance for combustion and reactive models [101, 102, 103, 104]. For example, in [103], the detailed chemical processes are represented by an equivalent reduced system that tracks scalars. Two such scalars are considered; (i) A mixture fraction variable, that tracks the mixing of fuel and oxidizer and (ii) a progress variable, which tracks the global extent of reaction of the local mixture of fuel. in this approach, the progress variable (C) is governed by the following scalar transport equation,

$$\frac{\partial \rho C}{\partial t} + \nabla \cdot (\rho \vec{V} C) = \nabla \cdot [\rho \alpha \nabla C] + f$$

where ρ is the density, $\vec{V}$ is the convective velocity field; α is a coefficient of diffusion (may include the turbulent diffusion component) and f is the forcing due to the chemical reaction.

Past efforts have been performed based on the numerical analysis of the LCDRE [105, 106, 107, 108]. Fourier analysis of the LCDRE is reported in [108], where the authors demonstrate that their positivity preserving variational method is superior to the Galerkin/Least Squares (GLS) and sub-grid scale (SGS) methods. According to the authors, the formulation proposed shows superiority for both reaction-dominated as well as convection-dominated flow regimes, due to the minimization of spurious oscillations. The Global Spectral Analysis approach for this specific problem was performed for two commonly used finite difference schemes where a model reaction term allows for the attention of a theoretical solution [109].

To the knowledge of the present authors, no detailed analysis of the numerical methods used here (LW and TTGC) have been however reported for LCDRE. Such an analysis is however accessible for the individual problems that are: the convection equation [50]; the diffusion equation [84] and the convection-diffusion equation [74]. In these latter references, the authors have shown that for the linear equations, the numerical convection speed and the numerical coefficient of diffusion become a function of the numerical and physical length scales. In the present work, the GSA analysis is performed on LCDRE with the objective of being applicable to a realistic reaction term. To do so, the reaction term is constructed based on the Pfitzner's model [110], itself representative of premixed flames which is then validated through numerical experiments.

The chapter is organized as follows. The analysis of the 1D LCDRE with a constant reaction term is first discussed in Section 3.2. In this section, the mathematics involving GSA is detailed along with the resulting property charts. The corresponding formulation issued by the use of the two schemes (LW-CD_2 and TTGC-CD_2) if applied to the resolution of the LCDRE are shown in Sec. 3.3. The corresponding numerical property charts are then used to explain and pinpoint differences between the LW and TTGC schemes in Section 3.4. The effect of the reaction source term on these property charts for both numerical schemes is detailed in Section 3.5. A more faithful reaction term is presented in Section 3.6. Here, the details of how the reaction term is modeled as well as its comparison with a realistic Pfitzner model are shown. The GSA analysis is then performed on this model problem for the LW-CD_2 scheme. From the obtained property charts, stability limits are derived and the findings are validated by simulating a 1D fully premixed flame using the Pfitzner model by CFD. To finish, conclusions and perspectives are provided.

3.2 Analysis of the 1D linear convection-diffusion-reaction equation with constant reaction source term

The theoretical and numerical analysis of the linear $1D$ convection-diffusion-reaction equation is shown next. Prior to its re-expression in a discrete form by use of the LW and TTGC schemes, the analysis is here presented to provide the exact effects of the underlying physics on dispersion, dissipation as well as the group velocity response.

For the sake of derivation, the Linear Convection-Diffusion-Reaction equation (LCDRE) is written as,

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} + su = \alpha\frac{\partial^2 u}{\partial x^2}. \tag{3.1}$$

Here c, α and s are constants specifying the convection speed, diffusion and reaction coefficients, respectively. As done in the previous chapter, GSA expresses the unknown $u(x,t)$ in the hybrid spectral plane [47, 71] that,

$$u(x,t) = \int \hat{U}(k,t)e^{ikx}dk. \tag{3.2}$$

Here $\hat{U}$ is the Fourier amplitude and k represents the wavenumber. Substituting this expression in Eqn. (3.1), the following transformed expression is obtained

in the spectral space,

$$\frac{d\hat{U}}{dt} + ick\hat{U} + s\hat{U} = -\alpha k^2 \hat{U}. \tag{3.3}$$

Note that, Eqn. (3.3) can be solved for a given initial condition of the form, $u(x,0) = f(x) = \int \hat{F}(k)e^{ikx}dk$, yielding the exact solution,

$$\hat{U}(k,t) = \hat{F}(k)e^{-\alpha k^2 t}e^{-ikct}e^{-st}. \tag{3.4}$$

To access the physical dispersion relation, the unknown $u(x,t)$ can be written using the bi-dimensional Fourier-Laplace transform so that,

$$u(x,t) = \int\int \bar{U}(k,\omega)e^{i(kx-\omega t)}dkd\omega, \tag{3.5}$$

which upon substitution in Eqn. (3.1) yields the physical dispersion relation,

$$\omega = ck - is - i\alpha k^2. \tag{3.6}$$

Therefore, based on the above expression, a numerical scheme used to solve the LCDRE must obey this dispersion relation to minimize phase as well as dispersion errors [50, 71]. Indeed, for the above problem one finds that the complex phase speed should read,

$$c_{phys} = \frac{\omega}{k} = c - \frac{is}{k} - i\alpha k. \tag{3.7}$$

Likewise, the physical group velocity is,

$$V_{g,phys} = \frac{\partial \omega}{\partial k} = c - 2i\alpha k. \tag{3.8}$$

For the physical group velocity, the velocity at which energy propagates in the medium is: $v_{g,phys} = \Re(c - 2i\alpha k) = c$.

The physical amplification factor can be expressed from Eqn. (3.4), introducing the notion of a time increment, Δt. In that case,

$$G_{phys} = \frac{\hat{U}(k,t+\Delta t)}{\hat{U}(k,t)} = e^{-\alpha k^2 \Delta t}e^{-ikc\Delta t}e^{-s\Delta t}. \tag{3.9}$$

Upon discretization to introduce the realm of numerical resolution, Eqn. (3.1) yields several non-dimensional parameters which are functions of the space-time discretization choice, i.e.. the CFL number (N_c), the Peclet number (Pe) and the Damkohler number (Da). Each of these numerical parameters correspond to a physical process. These are convection, diffusion and reaction that are present in the original governing equation. These numbers follow the expressions:

$$N_c = \frac{c\Delta t}{h},\ Pe = \frac{\alpha \Delta t}{h^2},\ Da = \frac{sh}{c},$$

where h stands for the grid spacing and Δt is the chosen numerical time step. If these non-dimensional parameters are substituted into Eqn. (3.9), one obtains,

$$G_{phys} = e^{-Pe(kh)^2} e^{-i(kh)N_c} e^{-N_c Da}. \tag{3.10}$$

Although, the CFL number, N_c, is directly related to the convection process in the governing equation and physically equals the phase speed, c, in simulations, it has been shown [50, 71] that its numerical counterpart, c_{num}, is a function of the wavenumber, k due to the presence of dispersion and phase errors. Equivalently, the Peclet number, Pe, which relates to the physical diffusion process, has been shown [74, 84] if not treated properly, i.e. its numerical counterpart, α_{num} , does not remain constant but again depends on k. Proceeding with the same logic, the reaction process, represented by the reaction coefficient (s) can thus also become dispersive, i.e. $s = s_{num}(k)$, and impact the numerical Damkohler counterpart.

To extract such a behavior from a numerical discretization scheme, the numerical dispersion relation can be expressed by analogy with Eqn. (3.6) [50, 84] into,

$$\omega_{num} = c_{num}k - is_{num} - i\alpha_{num}k^2. \tag{3.11}$$

Note that, for the convection-diffusion equation, a similar relation has been reported [74]. The numerical amplification factor with added reactions can hence be similarly recast into,

$$G_{num} = e^{-\alpha_{num}k^2\Delta t} e^{-ikc_{num}\Delta t} e^{-s_{num}\Delta t}. \tag{3.12}$$

Based on this last expression, the reaction source term is seen to play a similar role as diffusion which was explained in the previous chapter. The reaction source term therefore potentially reduces the acceptable limits and the regions under which the solution remains stable. Manipulating the last expression, numerical phase shift per time step is given by [50, 71],

$$tan(\beta) = -\frac{\Im(G_{num})}{\Re(G_{num})}, \tag{3.13}$$

with the ratio of the imaginary and real parts of the numerical amplification factor. The non-dimensional numerical phase speed can therefore be retrieved as,

$$\frac{c_{num}}{c} = \frac{\beta}{kc\Delta t} = -\frac{1}{(kh)N_c}\, tan^{-1}\left[\frac{\Im(G_{num})}{\Re(G_{num})}\right]. \tag{3.14}$$

Similarly, the numerical group velocity can be extracted from the numerical dispersion relation by using its definition, $V_{g,num} = \frac{\partial \omega_{num}}{\partial k}$, which upon simplification yields the following non-dimensional expression,

$$\frac{V_{g,num}}{V_{g,phys}} = \frac{1}{N_c}\frac{d\beta}{d(kh)}. \tag{3.15}$$

Recall that, the physical group velocity ($V_{g,phys}$) for the present study equals c.

The numerical diffusion and reaction coefficients will be evaluated from Eqn. (3.12), by noticing that α_{num}, c_{num} and s_{num} are real quantities, so the numerical amplification factor can be reduced to,

$$|G_{num}| = e^{-\alpha_{num}k^2\Delta t}e^{-s_{num}\Delta t}. \tag{3.16}$$

Introducing the non-dimensional parameters (Pe, N_c &Da), one obtains,

$$ln|G_{num}| = -\frac{\alpha_{num}}{\alpha}(kh)^2 Pe - \frac{s_{num}}{s}N_c Da. \tag{3.17}$$

The effect of the reaction source term is evident on the numerical amplification relation shown in Eqn. 3.17.It plays a similar role as the diffusion coefficient. As the reaction source term is increased in strength, the range of parameters over which the problem is stable shrinks. From the above expression, the specific case $k = 0$ is used to highlight the complexity of the problem. In this specific case,

$$\frac{s_{num}}{s} = -\left(\frac{ln|G_{num}|_{k=0}}{N_c Da}\right), \tag{3.18}$$

which indicates that for a given numerical scheme, the numerical amplification factor is a function of all four non-dimensional parameters, kh, N_c, Pe and Da. Note also that the non-dimensional numerical reaction coefficient can be impacted by N_c and Da. Substituting the numerical reaction coefficient obtained, in Eqn. (3.17), one obtains a non-dimensional numerical diffusion coefficient that reads,

$$\frac{\alpha_{num}}{\alpha} = \frac{ln|G_{num}|_{k=0} - ln|G_{num}|}{(kh)^2 Pe}. \tag{3.19}$$

As noted in [74] for the convection-diffusion equation, here also the ratio is related to Pe.

3.3 GSA of the numerical schemes

The details of the impact of a chosen numerical scheme on the above discussed properties are given hereafter. First, the LW-CD_2 scheme is considered, followed by, the same analysis performed to the two-step TTGC-CD_2 scheme.

3.3.1 LW-CD_2 scheme

Here, the LW scheme is used for the spatio-temporal discretization of the convective terms while the diffusive term is approximated using a second order central difference (CD_2) scheme. Applied to the LCDRE at the j^{th} node of a 1D uniform mesh at time level n and introducing the non-dimensional numbers, the above scheme results in the discrete relationship,

$$u_j^{n+1} = u_j^n - \frac{N_c}{2}(u_{j+1}^n - u_{j-1}^n) + \frac{1}{2}\,\zeta(u_{j+1}^n - 2u_j^n + u_{j-1}^n) + DaN_c, \tag{3.20}$$

where ζ is defined as $Nc^2 + 2\,Pe$.

Compared to the expression obtained in the previous chapter, the reaction source term modifies the expression by a constant, DaN_c, so the new LW-CD_2 scheme numerical amplification factor reads,

$$G_{LW-CD_2} = 1 - i\,Nc\,sin(kh) + [\zeta(cos(kh) - 1)] + DaN_c. \tag{3.21}$$

Using Eqn. 3.21, all the subsequent expressions characterizing the scheme behavior can be easily obtained.

3.3.2 TTGC-CD_2 scheme

The TTGC is a two step scheme used for the spatio-temporal discretization of the convective terms [68]. When applied to the LCDRE, the source term is added only in the second stage of time integration scheme, to get,

$$\begin{aligned}\frac{(\tilde{u}_{i+1}^n + 4\tilde{u}_i^n + \tilde{u}_{i-1}^n)}{6} = \frac{(u_{i+1}^n + 4u_i^n + u_{i-1}^n)}{6} - \alpha_{TTGC}\,N_c\frac{(u_{i+1}^n - u_{i-1}^n)}{2} \\ + (\beta_{TTGC}\,N_c^2 + Pe)\,(u_{i+1}^n - 2u_i + u_{i-1}^n),\end{aligned} \tag{3.22}$$

$$\begin{aligned}\frac{(u_{i+1}^n + 4u_i^n + u_{i-1}^n)}{6} = \frac{(u_{i+1}^n + 4u_i^n + u_{i-1}^n)}{6} - N_c\frac{(\tilde{u}_{i+1}^n - \tilde{u}_{i-1}^n)}{2} \\ + (\gamma_{TTGC}\,N_c^2 + Pe)\,(u_{i+1}^n - 2u_i + u_{i-1}^n) + Su_i^n.\end{aligned} \tag{3.23}$$

The numerical properties of the TTGC scheme are then obtained using,

$$G_{TTGC-CD_2} = 1 + \frac{\hat{A}\,\hat{S}\,\varsigma_2}{N_c} - \alpha_{TTGC}(\hat{A}\hat{L})^2 - i\hat{A}\hat{L}\left(1 + \frac{\hat{A}\hat{S}\varsigma_1}{N_c}\right) + Da\,N_c \quad (3.24)$$

$\hat{A} = \frac{3N_c}{2+cos(kh)}$, $\hat{S} = 2cos(kh) - 1$, $\hat{L} = sin(kh)$.

For both schemes, expressions clearly confirm the dependency of the numerical properties on all parameters: *i.e.* kh, Nc, Pe and Da. These aspects are specifically detailed next, since they have important effects on the scheme stability, effective diffusion, convection speed and group velocity.

3.4 Comparison of the property charts for the LW and TTGC schemes applied to the LCDRE

For any numerical method, it is desired that the scheme follows the dispersion relation of the original governing equation as closely as possible. As a result, the numerical solution should not only be stable, but also be able to resolve the speed and the diffusion characteristics of the exact solution. The accuracy of the scheme is obtained by looking at the relative amplification factor modulus $G_{rel} = \frac{|G_{num}|}{|G_{phys}|}$. While the stability of the scheme is assessed by ensuring that $|G_{num}| < 1$. The effective diffusive process and potential appearance of anti-diffusion is obtained by looking at α_{num}. Similarly, the non-dimensionalized convection velocity, c_{num}/c_{phys}, and the non-dimensional group velocity, $v_{g,num}/v_{g,phys}$, detail the scheme's ability to recover the true convective process. Finally, the strength of the reaction term is non-dimensionalized through the problem's Damkohler number, Da. To finish and to identify the regions in the kh-Nc-Pe-Da space where the dispersion relation results in satisfying the Dispersion Relation Preserving (DRP) condition, a tolerance of $\pm 1\%$ deviation is chosen.

The G_{rel} contours for the LW-CD_2 scheme are plotted in Fig. 3.1 for increasing values of Pe in the Nc-kh plane for $Da = 0.10$. The measure of the effective numerical diffusion process relative to the exact solution is indeed obtained via G_{rel}. If there is excessive damping, $G_{rel} < 1$, while under-diffusion is obtained when $G_{rel} > 1$. Note, that the condition $G_{rel} > 1$ alone doesn't imply numerical instability as the solution could be bounded although with amplitudes larger than the one issued by the exact solution (under diffusive). The numerical instability limit is obtained only when both the absolute numerical amplification factor $|G_{num}|$ as well as G_{rel} are considered. In such a case, unstable regions are obtained when $|G_{num}| > 1$ and are shaded in orange in all figures. When it comes to the

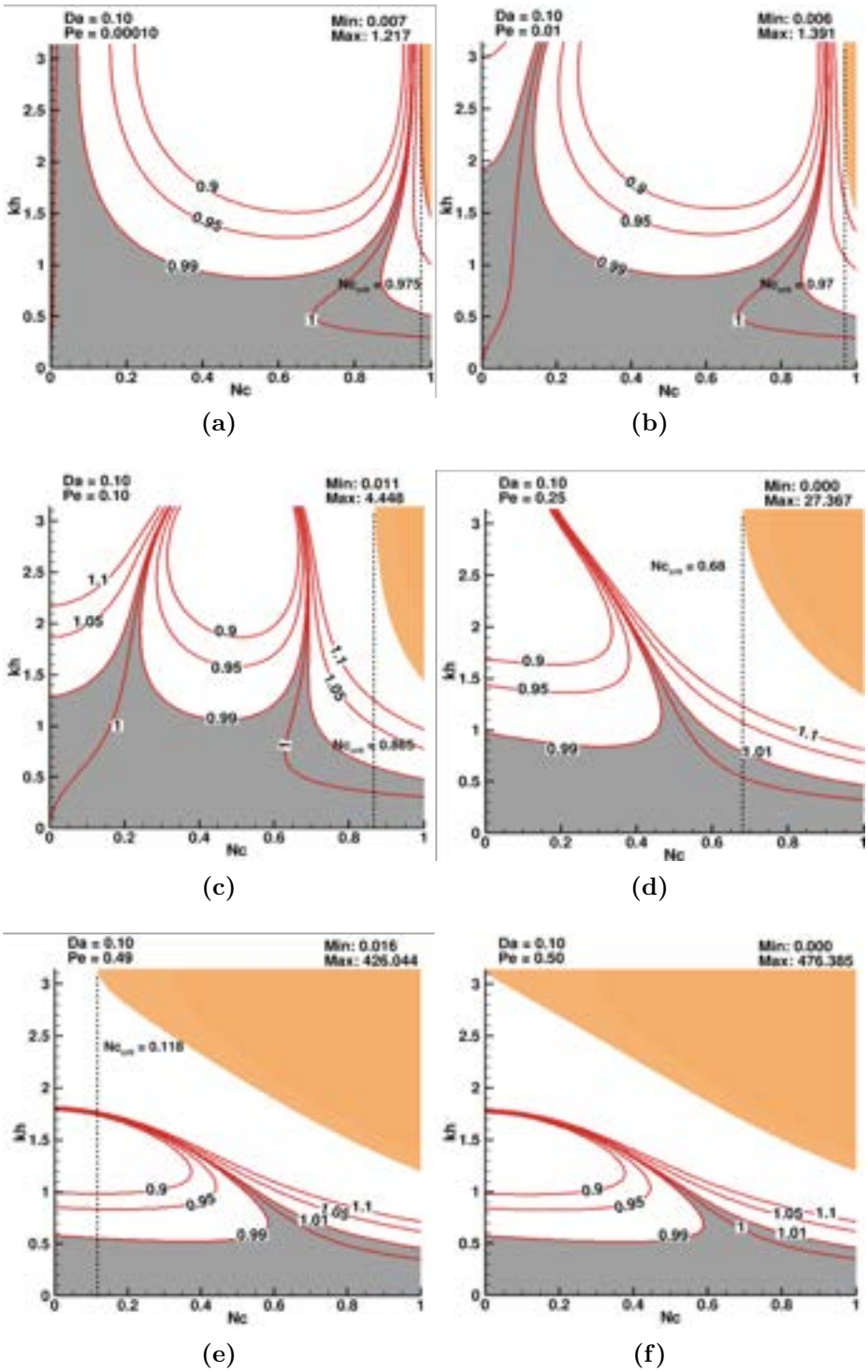


Figure 3.1: Contours of $|G_{num}|/|G_{phy}|$ as a function of kh and Nc for various value of Pe at $Da = 0.1$ for LW-CD$_2$ scheme. The regions of numerical instability are colored in orange. DRP regions are shaded in grey.

DRP regions, they are identified by allowing a 1% deviation tolerance ($0.99 < G_{rel} < 1.01$) and are evidenced by grey shaded areas.

Looking first at the cases $Pe \leq 0.01$ of Fig. 3.1 (a) & (b), because of the added physical / numerical diffusion as well as the reaction term, the standard LW critical stability criterion for pure convection, $Nc = 1$, does not hold anymore for all values of kh. It is also observed that in these low Pe cases, DRP is respected with minimal error especially for low wavenumbers. The enhanced stability compared to a purely convective process is due to the dominance of the diffusion process in the physics of such a problem which results in the improved stability range. Here, the plots are shown for $Da = 0.1$, while in the previous chapter the same plots are shown for LCDE ($Da = 0$). The addition of the reaction source term affects the property charts in different ways. First, the DRP region shrinks to a smaller region especially at high values of N_c. Second, from the LCDE plots of the previous chapter, instability (orange region) is first observed at $Pe = 0.1$. Now, the instability is observed at a much smaller value: $Pe = 0.0001$. This confirms that the scheme stability limit is affected by the reaction source term, the critical value of N_c observed in the LCDRE being lower than the one observed on the corresponding LCDE plots. As Pe increases, the acceptable range of $Nc - kh$ values decreases as shown by Fig. 3.1(c). Ultimately for $Pe \approx 0.5$, instability appears for all Nc values mainly due to the Nyquist limiting wavelength $kh = \pi$, which is in agreement with the well known purely diffusive problem. For $Pe \geq 0.5$, the range $Nc \leq 1$ and $kh < 1.5$ is initially stable however round-off errors will ultimately render these simulations unstable after a long time integration. Finally, as Pe further increases, the DRP region shrinks to lower kh values.

Investigation of G_{rel} contours for the TTGC-CD_2 scheme are shown in Fig. 3.2 for different values of Pe in the Nc-kh plane for $Da = 0.1$. In the cases where, $Pe \leq 0.0001$, Fig. 3.2 (a) & (b), there exists a large region of DRP present especially for low wavenumbers and no instability is noticed. For values of Nc up to 0.2, there is a stable DRP preserving region for all wavenumbers which appears wider than the one obtained for LW-CD_2 in Fig. 3.1 (a) & (b). Here, the differences between the LCDE (from previous chapter) and the LCDRE is not that pronounced. The stability limits of the scheme are affected slightly as observed with LW-CD_2. First instability is observed at $Pe \approx 0.1$ as was the case in LCDE. The critical value of N_c reduces slightly for the LCDRE case due to the addition of the source term. Finally, for $Pe \approx 0.17$, instability appears for all Nc values mainly due to the Nyquist limiting wavelength $kh = \pi$, i.e. earlier

than for the LW-CD_2 scheme for which instability for all Nc values appears at $Pe \approx 0.50$.

Note that, G_{rel} is a measure of the solution accuracy, which here expresses the inaccurate evaluation of the numerical diffusive process . Such an error is expressed through, α_{num}/α shown in Fig. 3.3 for the LW-CD_2 scheme at $Da = 0.1$. As expected, similar variations/contours as for $|G_{num}|$ are observed. Note that, α_{num}/α is also an indicator of numerical stability since it is for the problem considered the only mechanism contributing to the evolution of the amplitude of a wave with time as evidenced by Eqn. (3.19). Here, the stability limit is reported on the basis: $\alpha_{num}/\alpha > 0$ and the property charts exhibit identical features as in Fig. 3.1 , the unstable region being shaded in orange. The DRP region is almost non-existent and is available for only small patches for all values of Pe. Comparison of the plots for LCDRE with the plots for LCDE of the previous chapter, differences can be noted. Instability (orange region) is first observed at $Pe = 0.1$. However, here the first instability appears at a smaller value: $Pe = 0.0001$. Clearly, the stability limit is affected by the reaction source term, the critical value of N_c observed in the LCDRE is lower than the one observed for the corresponding LCDE plots. Note also that the DRP region has shrunk significantly, due to the reaction source term.

Similarly, the graphs for the TTGC-CD_2 scheme are shown in Fig. 3.4. The contours for this variable are very similar to those seen in the G_{rel} contours. In the case of negative numerical diffusion, anti-diffusion results in a focusing of the solution at the location of instability in the parameter space. As Pe increases, the DRP region initially increases slightly but does not change much especially for larger Pe values till the critical value of $Pe = 0.17$, for which there is no value of N_c which is stable for all wavenumbers. The effect of the reaction source term here is smaller compared to the case of the LW-CD_2 scheme. First, instability is observed at $Pe \approx 0.1$ as was the case for LCDE for LW-CD_2 scheme. The critical value of N_c reduces only slightly due to the addition of the source term. The DRP region has also reduced slightly.

On the basis of the discussion of the previous chapter, the non-dimensional group velocity is deemed sufficient and the non-dimensional phase speed is not specifically detailed here. Figure. 3.5 shows the non-dimensional group velocity of the LW-CD_2 scheme in the Nc-kh plane for different values of Pe for $Da = 0.1$. At lower Pe, Fig. 3.5 (a), (b) & (c), negative values of the group velocity (q-waves) are observed for high wavenumbers ($kh > \pi/2$) and for $Nc < 0.7$. Computations performed in these regions will hence produce upstream propagating non-physical waves. As Pe increases, Fig. 3.5 (c) & (d), the region of q-waves shrinks to finally

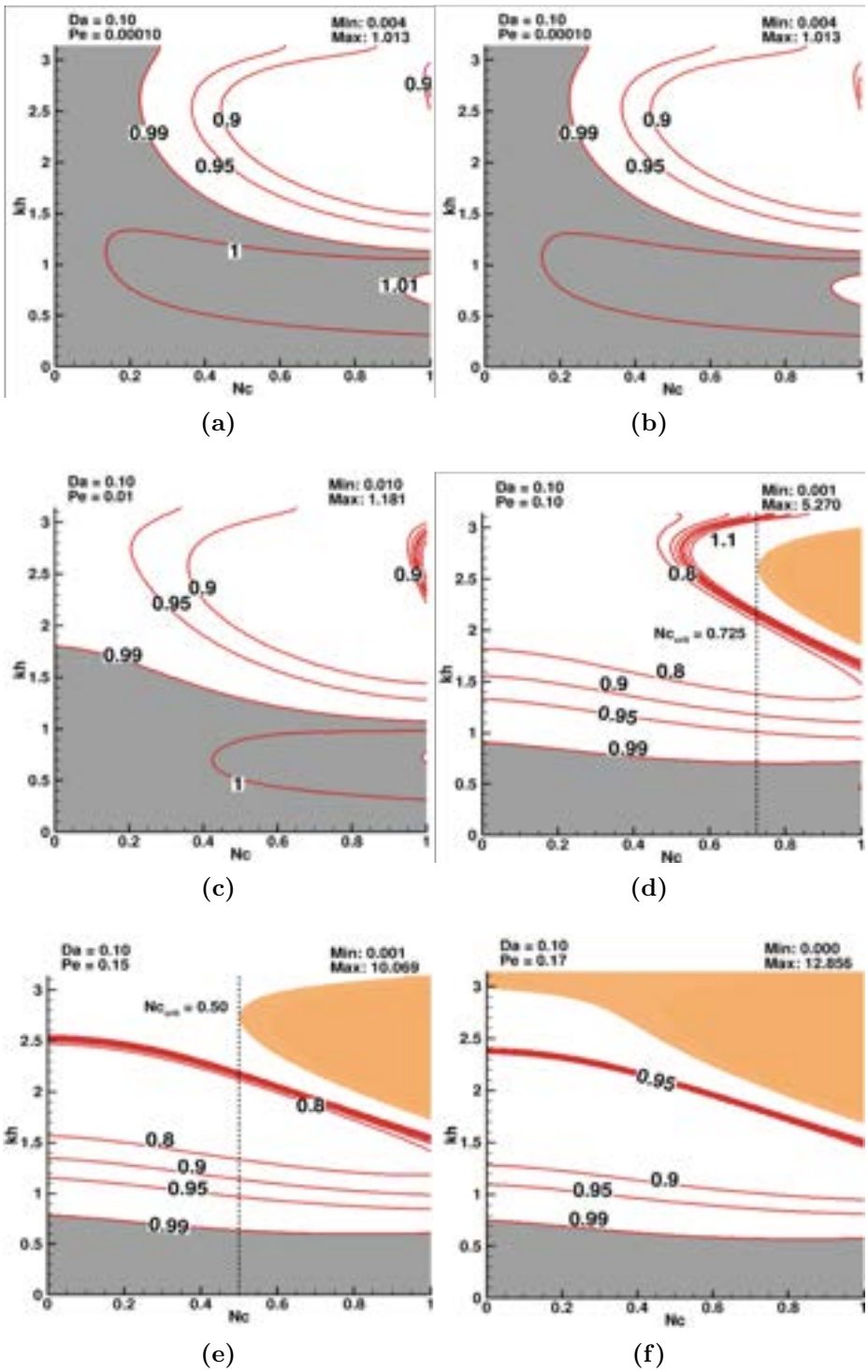


Figure 3.2: Contours of $|G_{num}|/|G_{phy}|$ as a function of kh and Nc for various value of Pe at $Da = 0.1$ for TTGC-CD_2 scheme. The regions of numerical instability are colored in orange. DRP regions are shaded in grey.

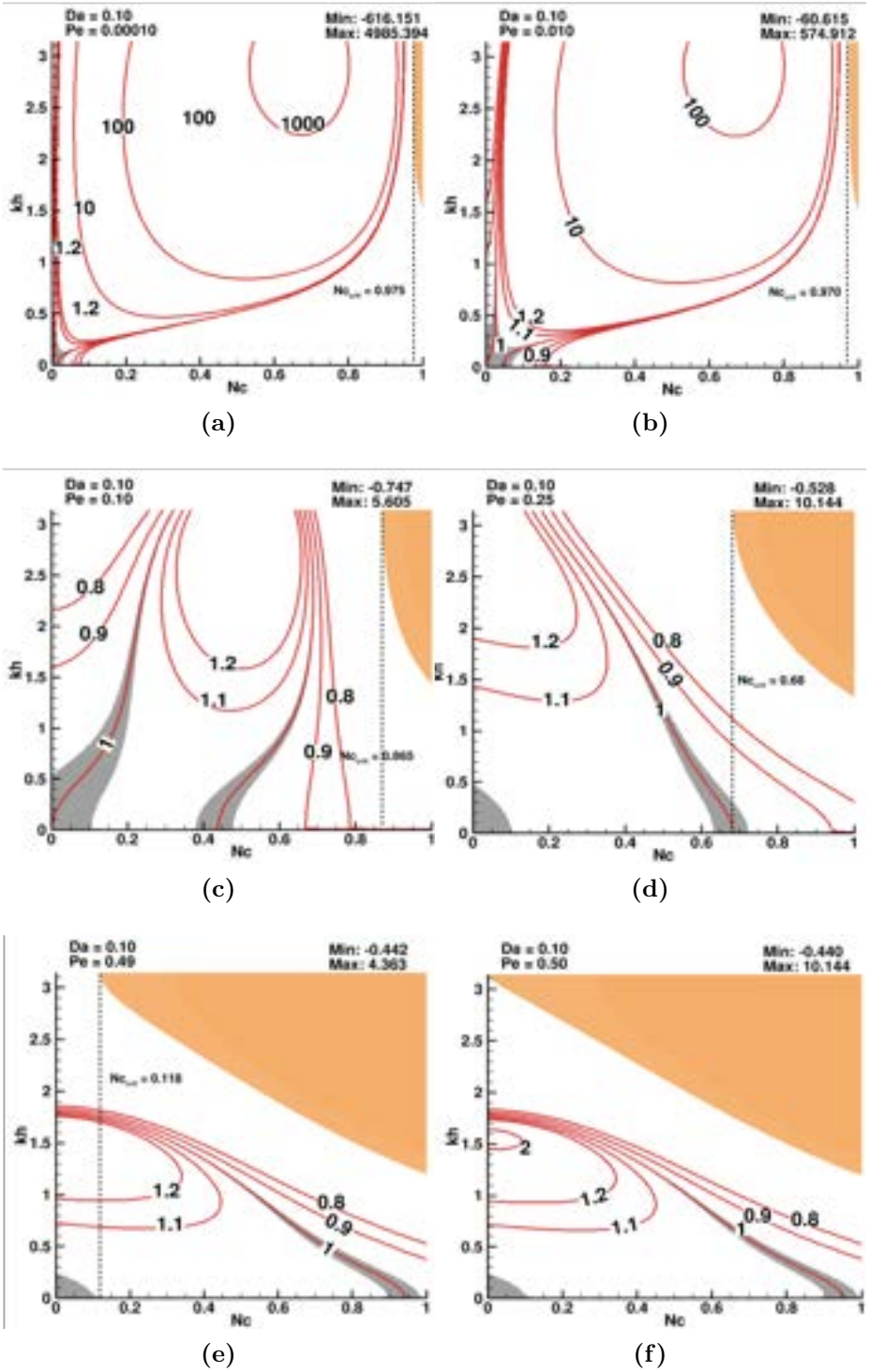


Figure 3.3: Effective measure of the numerical diffusion process, α_{num}/α taking place with the LW-CD$_2$ scheme in the $Nc - Pe$ plane at $Da = 0.1$. Note that instability is identified in orange by $|G_{num}| \leq 1$; the grey region corresponds to the 1% DRP zone.

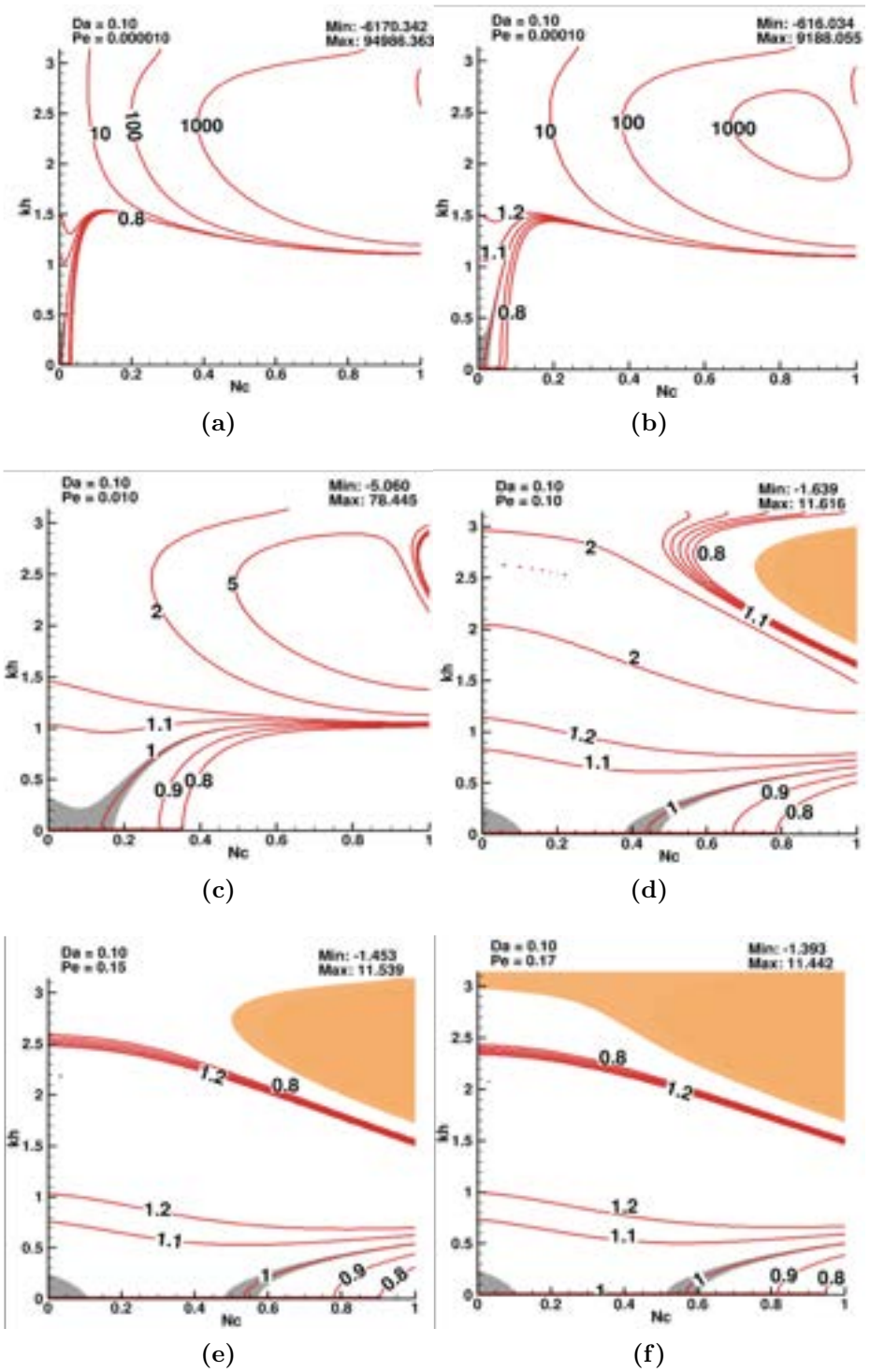


Figure 3.4: Effective measure of the numerical diffusion process, α_{num}/α taking place with the TTGC-CD$_2$ scheme in the $Nc-Pe$ plane at $Da = 0.1$. Note that instability is identified in orange by $|G_{num}| \leq 1$; the grey region corresponds to the 1% DRP zone.

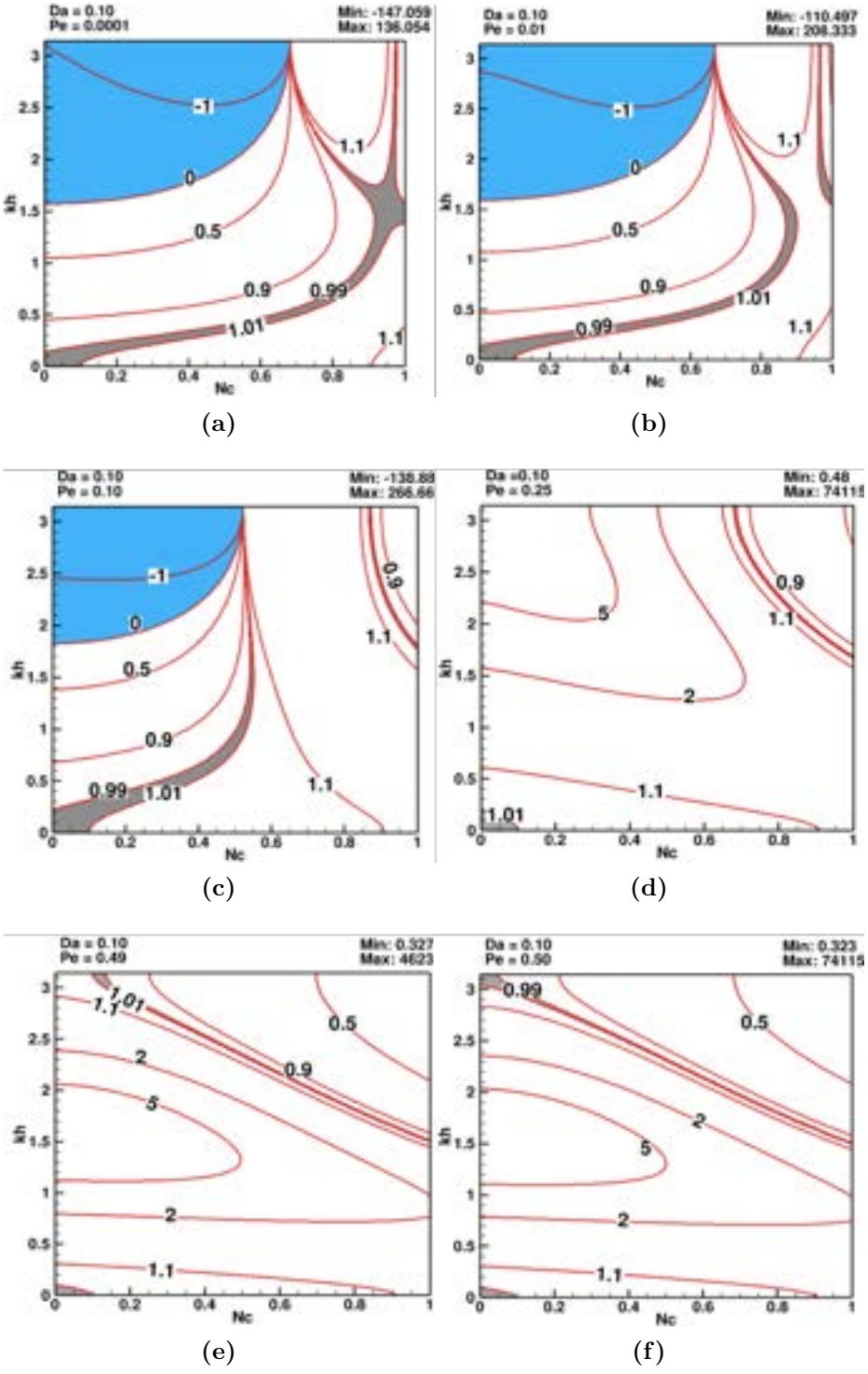


Figure 3.5: Contours of $v_{g,num}/v_{g,phy}$ as a function of kh and Nc for various value of Pe at $Da = 0.1$ for the LW-CD$_2$ scheme. The regions of q-waves are colored in blue. DRP regions are shaded in grey.

disappear from the Nc-kh plane around $Pe = 0.25$. Similarly, the DRP region shrinks with increasing Pe, and remains quite narrow: *i.e.* for $kh < 0.15$ for all Nc values. Comparisons of the results for LCDRE with LCDE from the previous chapter indicate that there is not much difference for the non-dimensional group velocity. The q-wave region disappears around $Pe = 0.25$ as in the case of the LCDE, while the DRP region is smaller and scarcer in the case of the LCDRE.

Similarly. Fig. 3.6 shows the non-dimensional group velocity of the TTGC-CD_2 scheme in the Nc-kh plane for different values of Pe and $Da = 0.1$. At very low values of Pe, Fig. 3.6 (a) & (b), negative values of group velocity (q-waves) are observed for high wavenumbers ($kh > 2.2$) and when $Nc < 0.8$. This region of q-waves remains similar up to $Pe = 0.01$. As Pe increases further, Fig. 3.6 (d) & (e), the region of q-waves changes in shape and finally settles for all wavenumbers $kh > 2.2$ for higher values of Pe. In this case, as the value of Pe increases, the q-wave region in the $kh - Nc$ plane does not disappear as it does for the LW-CD_2 scheme. Also, the DRP region changes slightly with an increasing Pe value all through. Finally, there is a negligible difference between plots for the LCDRE and the LCDE in terms of behavior of the q-waves although, the DRP region has shrunk significantly in the case of the LCDRE shown here for all values of Pe.

We have briefly explained the difference in the plots for the LCDE and LCDRE in this section. Next, to highlight the differences and effect of the reaction source term, the stability and numerical group velocity charts for different values of Da are compared.

3.5 Effect of reaction source term on the property charts of LW and TTGC schemes

The effect of Da on the numerical stability as well as dispersive properties for the LW-CD_2 scheme are shown in Fig. 3.7 by plotting G_{rel} and $v_{g,num}/c$ for different values of Da at a fixed $Pe = 0.10$. In frames (a) and (b), the plots for $Da = 0$ are shown, ie. the LCDE problem that is the case without reaction, explained in detail in the previous chapter. For this case, as explained before, in frame (a), there exists a large DRP region for all values of N_c for low wavenumbers. At this value of Pe, there exists an instability region shaded in orange and the critical value: $N_c = 0.895$. In frame (b), there exists a q-wave region shaded in blue. In frames (c) and (d), the property charts are shown for $Da = 0.1$ which show differences. In frame (c), the DRP region has indeed shrunk with the addition of the reaction source term, especially for higher values of N_C. Comparing the frames (a, b) for the LCDE with the frames (c, d) of the LCDRE ($Da = 0.1$),

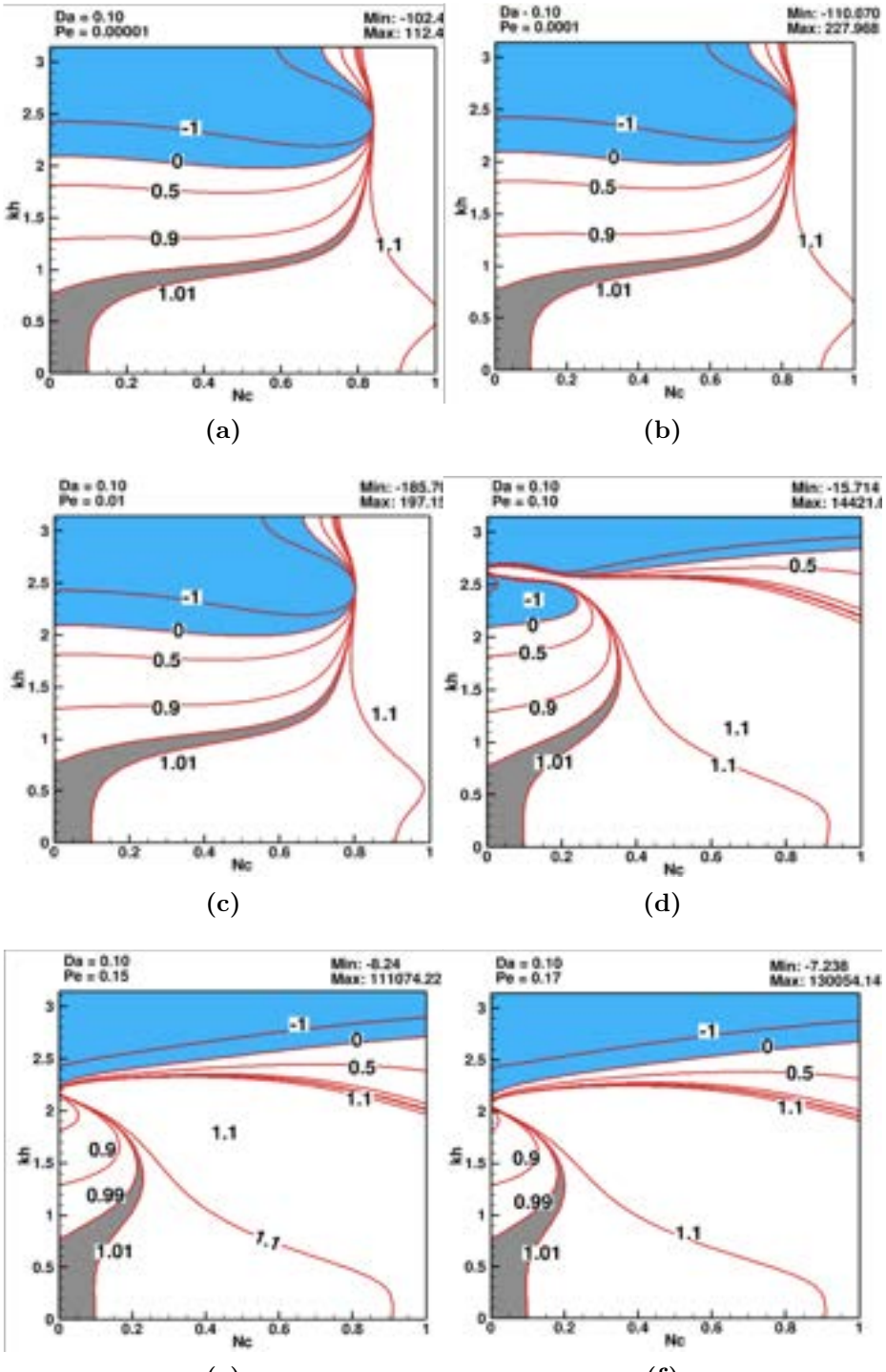


Figure 3.6: Contours of $v_{g,num}/v_{g,phy}$ as a function of kh and Nc for various value of Pe at $Da = 0.1$ for the TTGC-CD_2 scheme. The regions of q-waves are colored in blue. DRP regions are shaded in grey.

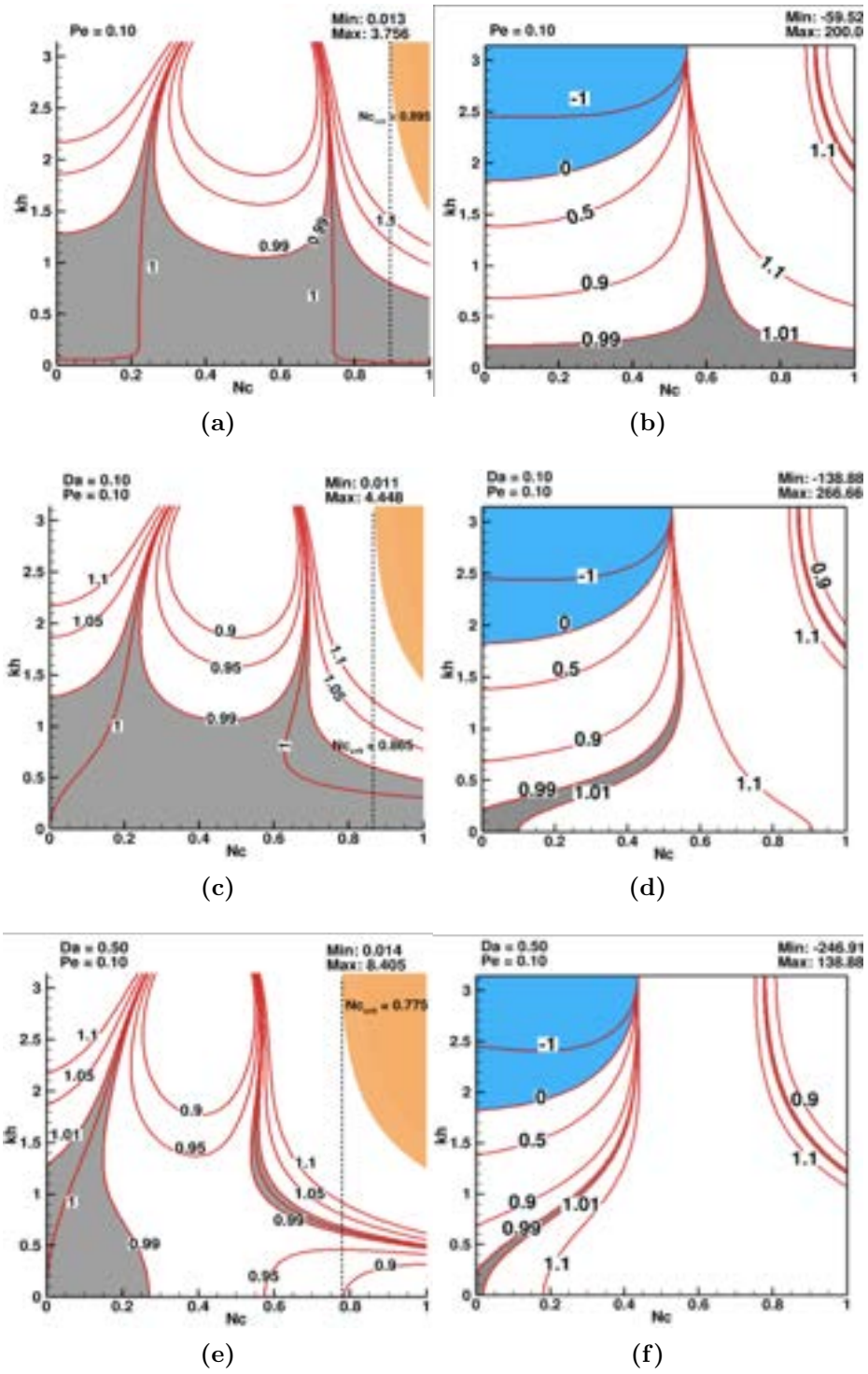


Figure 3.7: Effect of Da for LW-CD_2 scheme on numerical stability and dispersion at $Pe = 0.1$. Figures are plotted at (a),(b)$Da = 0.0$ (LCDE), (c),(d)$Da = 0.1$, (e),(f) $Da = 0.5$.

it can be observed that a smaller critical N_c can be used for performing stable simulations. In frame (d), the DRP region has significantly shrunk in comparison to the LCDE case shown in frame (b). There does not seem to be much difference for the q-wave region, $Da = 0.50$ LCDRE case is plotted in frames (e) and (f). For frame (e), the DRP region has significantly shrunk with the increase of the reaction source term, it now exists for a very small region, low values of CFL and wavenumbers. The critical value of CFL significantly reduces to $N_c = 0.775$. Finally, in frame (f), the DRP region shrinks further and hardly exists for the case chosen. Throughout all the figures, the q-wave region indicated by the blue shaded area remains almost unchanged.

Similarly, the effect of Da on the numerical stability as well as dispersive properties are shown in Fig. 3.8 for the TTGC-CD_2 scheme for different values of Da, at a fixed $Pe = 0.1$. In frames (a) and (b), the plots for $Da = 0$ are again recalled, i.e. the LCDE. As explained before, in frame (a), there exists a large DRP region for all values of N_c for low wavenumbers (here the DRP region exists for a lower wavenumber range than that observed with the LW-CD_2 scheme). At this value of Pe, there exists an instability region and the critical value of N_c equals 0.740. This critical value for the TTGC-CD_2 scheme is lower than the critical value (0.895) for the corresponding case with LW-CD_2 scheme. In frame (b), there exists a q-wave region shaded in blue. It exists for all values of N_c at high wavenumbers. In frames (c) and (d), the property charts are shown for $Da = 0.1$ and exhibit lesser difference than the one observed for the LW scheme. In frame (c), there is negligible difference in the DRP region and the critical value of CFL has reduced to $N_c = 0.725$. In frame (d), the DRP region has shrunk significantly and the q-wave region has also reduced slightly especially for low values of N_c. In frames (e) and (f), $Da = 0.50$ is plotted. Significant differences are observed especially for the DRP region. In frame (e), the DRP region has shrunk significantly with the increase of the reaction source term strength, yielding a critical value of N_c that has further reduced to $N_c = 0.670$. In the case of TTGC-CD_2, the critical value of N_c is lower than the LW-CD_2 scheme as expected when the reaction source term is added. Finally, in frame (f), the DRP region further shrinks and the q wave region also reduces for low values of N_c.

Overall, similar trends are observed for both numerical schemes when we keep increasing the reaction source term strength. The critical value of N_c for stable simulations indeed keeps decreasing with increasing Da. This effect is observed to be more critical with TTGC-CD_2 scheme. The DRP regions in both the stability and dispersive property charts decrease with an increasing strength of the reaction

source term. Finally, the negative q- wave region is not greatly affected by the reaction source term and almost remains unchanged for the LW-CD_2 scheme. While in the case of the TTGC-CD_2 scheme, only small differences are noticed at low CFL. In the analysis so far, a linear constant strength reaction source term was applied which is not representative of real flames. In the next section, a model is designed to mimic a more realistic reaction term allowing a GSA analysis on a more realistic problem.

3.6 Analysis of 1D linear convection-diffusion-reaction equation with a realistic reaction term

The previous application focused on the LCDRE with a constant source term which can be useful in terms of understanding. It is however not representative of a flame problem. To be fully applicable and useful a necessary step is to recast a flame problem in a LCDRE type that is amicable to such a numerical analysis. The first objective of the following is to identify such a model.

First, considering a 1D fully premixed flame front as the target for theoretical studies [51, 111], with adiabatic conditions, unity Lewis number for all species and a constant diffusivity, D. The transport equation describing such a problem can be reduced to the progress variable equation which reads [51]:

$$\frac{\partial \theta}{\partial t} + v\frac{\partial \theta}{\partial x} = D\frac{\partial^2 \theta}{\partial x^2} + \dot{\omega}_\theta \,, \tag{3.25}$$

where x, v are the spatial coordinate, the flame velocity, and $\dot{\omega}_\theta$ is the progress variable source term.

The progress variable source term is commonly modeled using an Arrhenius formulation, depending on the temperature field T and an activation temperature T_a: $\dot{\omega}_\theta \propto (1-\theta)\exp(-T_a/T)$ [51]. Another way to model such a reaction is to find analytical functions for θ and $\dot{\omega}_\theta$ which are solution of Eqn. 3.25. In this spirit, Pfitzner et al. [110, 112] proposes, for example,

$$\dot{\omega}_\theta = \frac{(\rho_u s_L)^2}{\rho D}(m+1)(1-\theta^m)\theta^{m+1} \,, \tag{3.26}$$

where S_L stands for the premixed laminar flame speed and m is a model coefficient that can be tuned to match a reference Arrhenius like chemistry model, and which will affect the laminar flame thickness. For any m value however, the laminar flame will always propagate at the laminar flame speed s_L specified in Eqn. 3.26. Unlike the linear reaction source term of Eqn. 3.1, this formulation vanishes for

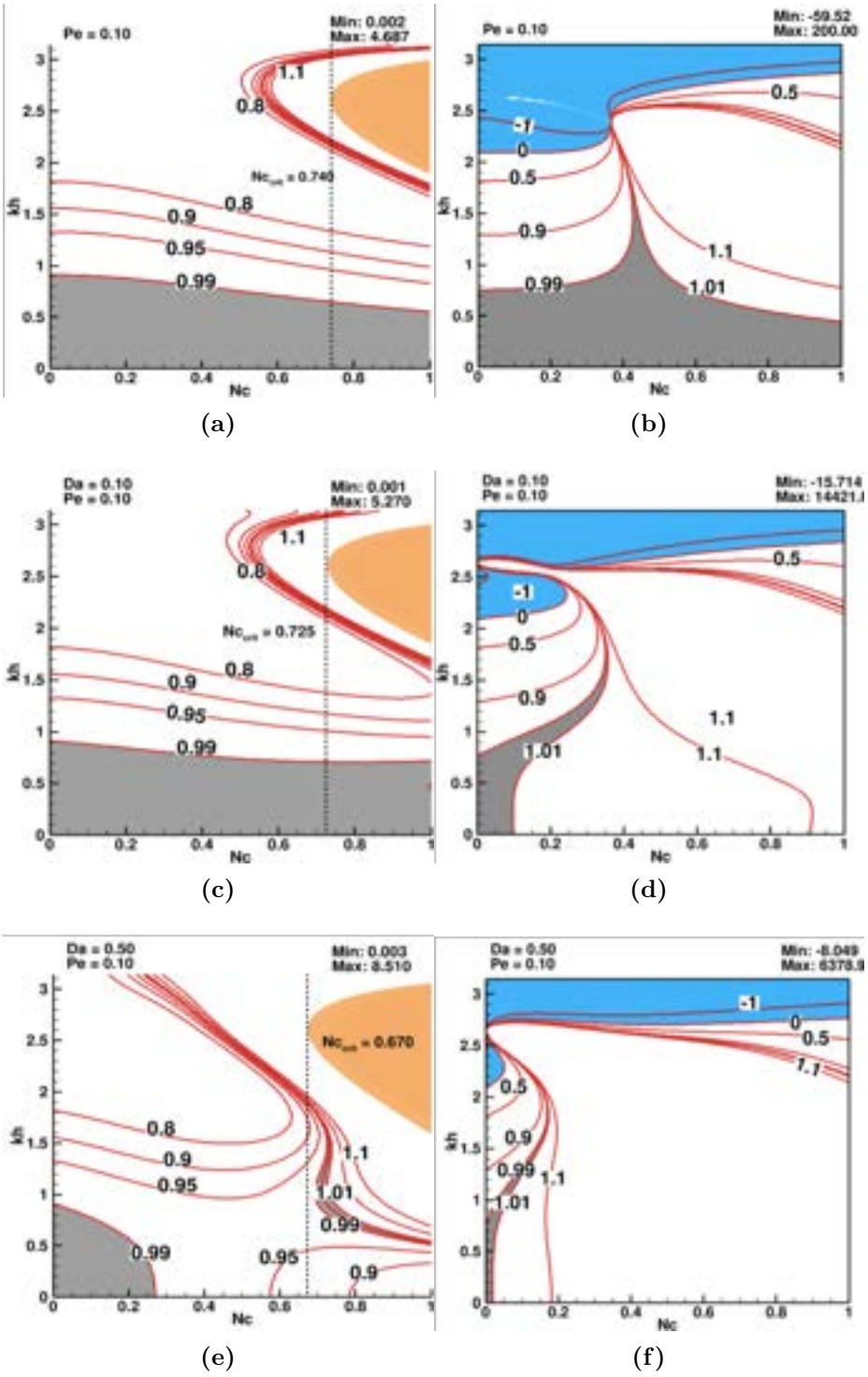


Figure 3.8: Effect of Da for $TTGC$-CD_2 scheme on numerical stability and dispersion at $Pe = 0.1$. Figures are plotted at (a),(b)$Da = 0.0$ (LCDE), (c),(d)$Da = 0.1$, (e),(f) $Da = 0.5$.

$\theta = 0$ as well as for $\theta = 1$ and peaks inside a thin reaction zone, thus behaving like a realistic combustion source term. However, such a model still renders impossible the linearization and therefore precludes from performing a GSA analysis for such a LCDRE.

To circumvent this difficulty while allowing a GSA analysis expressions for the Fourier transform of $\widehat{\dot{\omega}}_\theta$ as a function of $\widehat{\theta}$ is needed. To do so, an approximate reaction term R is proposed. While doing so, it further is presumed that R is also the result of the convolution of a high-pass filter B with the progress variable front θ. That is:

$$R(x) = B(x) \star \theta(x)\,, \tag{3.27}$$

so that, $\widehat{R}$ is simply the product of $\widehat{B}$ with $\widehat{\theta}$ (their Fourier counterparts),

$$\widehat{R}(k) = \widehat{B}(k)\widehat{\theta}(k)\,. \tag{3.28}$$

Note that, high-pass filtering of θ results in a R profile which goes to 0 in the fully burnt as well as unburnt states and peaks inside the flame front. Note also that if B is chosen so that R closely matches $\widehat{\dot{\omega}}_\theta$, their spectral behavior are expected to be the same and GSA can be performed by replacing $\widehat{\dot{\omega}}_\theta$ with the expression of $\hat{R}$ in Eqn. (3.28).

In the following, for simplicity B is chosen to be a first-order Butterworth filter whose frequency response reads,

$$\widehat{B}(k) = \frac{B_0}{\sqrt{1 + \left(\frac{k_c}{k}\right)^2}}\,, \tag{3.29}$$

where k_c is the cutoff wavenumber and B_0 its gain for $k \to +\infty$. Note that, k_c and B_0 control the width and the amplitude of the approximated reaction peak and they can be tuned to match a given Pfitzner source term, as illustrated in Fig. 3.9 or more complex expressions if needed. For the specific case considered in Fig. 3.9, the Pfitzner source term for m = 0.2, peaks inside the flame front and decays to 0 in the fresh (left of the figure) and burnt (right) states as expected. Furthermore, having $k_c = 265\,\mathrm{m}^{-1}$ and $B_0 = 3136\,\mathrm{s}^{-1}$, R clearly approaches $\dot{\omega}_\theta$ quite accurately. The computation non-dimensional Damkohler number, Da, for this problem is given by $Da = \frac{B_0 * h}{c}$. The Damkohler number is directly proportional to the amplitude of the approximated reaction peak, B_0. The largest errors arise near the cold boundary of the flame front and overall amount to less than 5% of the maximum source term value.

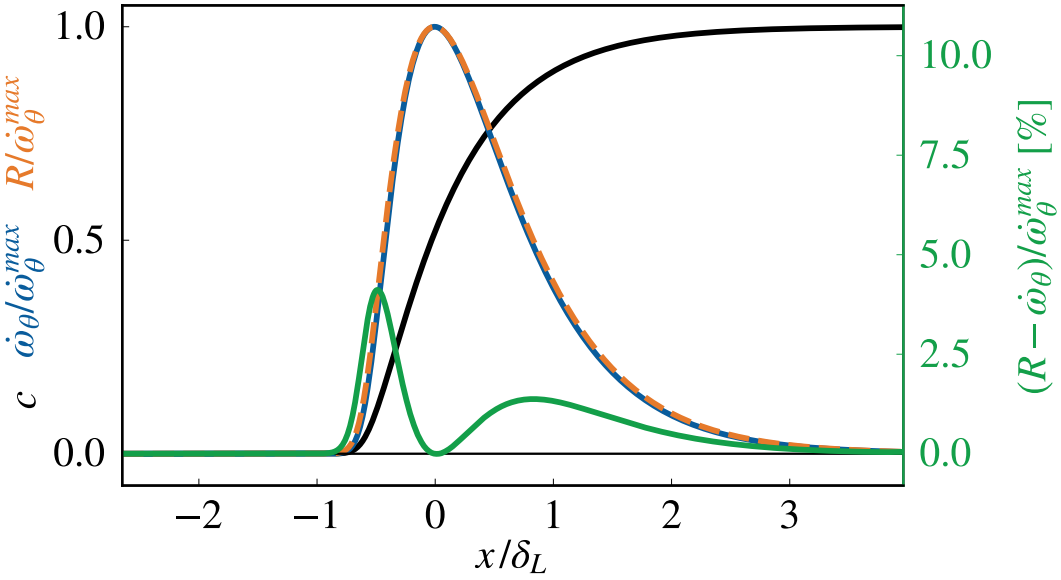


Figure 3.9: Profiles of progress variable θ (black), Pfitzner source term $\dot{\omega}_\theta$ (blue), filtered approximation R (orange, dashed), and error of the approximation (green, values on the right axis) from the simulation of a 1D propagating premixed flame. For visualization purposes, the last three quantities are normalized by the maximum value of $\dot{\omega}_\theta$ in the flame.

Using this reaction term, the expressions resulting from the use of the LW-CD_2 scheme yield,

$$|G_{num}| = 1 - iN_c sin(kh) + 2[\xi cos(kh) - 1)] + Da * Nc * \tau \tag{3.30}$$

where, $\tau = \frac{kh}{\sqrt{(kh)^2+(k_c h)^2}}$. Here, and in comparison to the previous case with a linear reaction source term, an additional parameter τ expresses the wavenumber, k, grid spacing, h, and dependency on the cutoff wavenumber, k_c.

A direct consequence is that the non-dimensional phase speed reads,

$$\frac{c_{num}}{c} = -\frac{1}{khN_c} tan^{-1} \frac{sin(kh)N_c}{1 + 2\xi(cos(kh) - 1) + DaN_c\tau}, \tag{3.31}$$

while, the non-dimensional group velocity can be expressed as functions of the real and imaginary parts of numerical amplification factor (G_r and G_i), as well as the derivatives with respect to (kh) noted respectively, G_r' and G_i', yields:

$$\frac{V_{g,num}}{V_g} = -\frac{1}{N_c} \frac{G_r G_i' - G_i G_r'}{(G_r^2 + G_i^2)}. \tag{3.32}$$

Introducing the non-dimensional parameters (Pe, N_c &Da) in Eqn. (3.30),

and further simplifying, obtains

$$ln|G_{num}| = -\frac{\alpha_{num}}{\alpha}(kh)^2 Pe - \frac{B_{0num}}{B_0}\tau * N_c * Da. \tag{3.33}$$

The non-dimensional numerical reaction coefficient can then be estimated by evaluating the above expression for $Pe = 0$,

$$\frac{B_{0num}}{B_0} = -\left(\frac{ln|G_{num}|_{Pe=0}}{N_c \tau Da}\right). \tag{3.34}$$

The numerical amplification factor is noted to be the function of all the four non-dimensional parameters, kh, N_c, Pe and Da while the non-dimensional numerical reaction coefficient is determined by N_c and Da. By substituting the numerical reaction coefficient in Eqn. (3.33) obtains the expression for the non-dimensional numerical diffusion coefficient,

$$\frac{\alpha_{num}}{\alpha} = \frac{ln|G_{num}|_{Pe=0} - ln|G_{num}|}{(kh)^2 Pe}. \tag{3.35}$$

Looking at the expression for the numerical amplification factor for this model with the LW-CD_2 scheme, the effect of the reaction source term can be directly observed by looking at Eqn. (3.33). The reaction source term plays an additional effect that is similar to the diffusion term observed in the previous chapter. The addition is here due to the dependency of τ, which is a function of k and k_c. Compared to the corresponding expressions for linear reaction source term, this addition is expected to effect the stability limits differently.

3.6.1 Extension to a reacting Navier-Stokes CFD

In the following, the propagation of a 1D fully premixed flame is simulated using the AVBP code [113, 114] by use of the LW-CD_2 scheme that is second-order accurate in space and time. Only the LW-CD_2 scheme has been shown here. The same has be successfully done for TTGC-CD_2 scheme but has not been presented in the manuscript as they show identical trends following the property charts for TTGC-CD_2 scheme. This test case is used to validate the findings from the property charts explained above. To do so, the problem is discretized on a uniform mesh with a grid spacing $h = 20\,\mu m$ and for a total length of $L = 20.48\,mm$. The fresh mixture is a stoichiometric methane-air mixture at a temperature $T = 300\,K$ and pressure $P = 1\,bar$. Combustion is modeled through a single irreversible chemical reaction $CH_4 + 2\,O_2 \rightarrow CO_2 + 2\,H_2O$. The species reaction rates are computed using the Pfitzner source term with $m = 0.2$ (as shown before), leading a laminar flame thickness $\delta_L = 940\,\mu m$. NSCBC inlet and

outlet conditions [115] are imposed at both ends of the domain. With an initial flow at rest, the flame propagates towards the inlet at its laminar flame speed $s_L = 0.38\,\mathrm{m\,s^{-1}}$.

Figure 3.9 shows the profiles of the progress variable θ, Pfitzner source term $\dot{\omega}_\theta$ along with its filtered approximation R. As said previously for $k_c = 265\,\mathrm{m^{-1}}$ and $B_0 = 3136\,\mathrm{s^{-1}}$ (as used before), R approaches $\dot{\omega}_\theta$ with a very low error. As all these parameters are fixed to match the theoretical Pfitzner model, the computational Damkohler number is also fixed, $Da = \frac{B_0 * h}{c} = 0.0009485$ for the present problem. Note that, the affect of this reaction source term is tested later on by solving another problem at a higher value, $Da = 0.9485$. For both the cases, the value of Pe is much smaller than the value of CFL because the viscosity of air (α) is of the order of $10^{-5}\ \frac{kg}{m*s}$.

The stability limits for the problem chosen in the $N_c - Pe$ plane is shown for the first case $Da = 0.0009485$ (solid line) and for the previously show LCDE problem (dashed line) using the LW-CD_2 scheme are shown in Fig. 3.10. The stability curves are zoomed in for low values of Pe, as we are using air in this case as mentioned before, also we zoom at the stability limits of N_c around unity to show effect of the reaction source term. The contours of $|G_{num}| > 1.0$ corresponds to numerical instability and is shaded in orange color. The two distinct lines show that even at such small values of Da, there is an effect on stability limits, although not a huge difference. The critical value of N_C reduced from $N_c = 1$ for LCDE to $N_c = 0.992$ for LCDRE at this low value of Da. As the values of k_c and h are fixed, $k_c * h$ is no longer a variable in the expressions for τ as discussed before. For the LCDRE case, Da being fixed, $N_c = \frac{c\Delta t}{h}$ and $Pe = \frac{\alpha \Delta t}{h^2}$ become only functions of the time step (Δt) to be chosen to perform the simulation.

Two test cases are and identified on the stability plots, 3.10, the details being given in Tab. 3.1. The two cases are solved by applying a slightly different constant time step (see Tab. 3.1). The first test case chosen, **case (a)** is a case which would have been stable for LCDE but due to the additional source term, here it is unstable as the operating N_c is above the critical value of N_c. This corresponds to a case with a constant time step, $\Delta t = 2.13$ x 10^{-7}. The second case chosen, **case (b)**, should be stable for the low Da value, LCDRE case with $\Delta t = 2.10$ x 10^{-7}. In this case, the operating N_c is within the value of the critical N_c. Next, let us validate these findings from the property charts by solving a reacting plane flame in a chamber using Pfitzner model.

First, **case (b)** with $\Delta t = 2.1$ x10^{-7} results are discussed, which according to the stability chart should be a stable simulation. Fig. 3.11 shows the line plot along the center-line of the domain of temperature in frames with the times

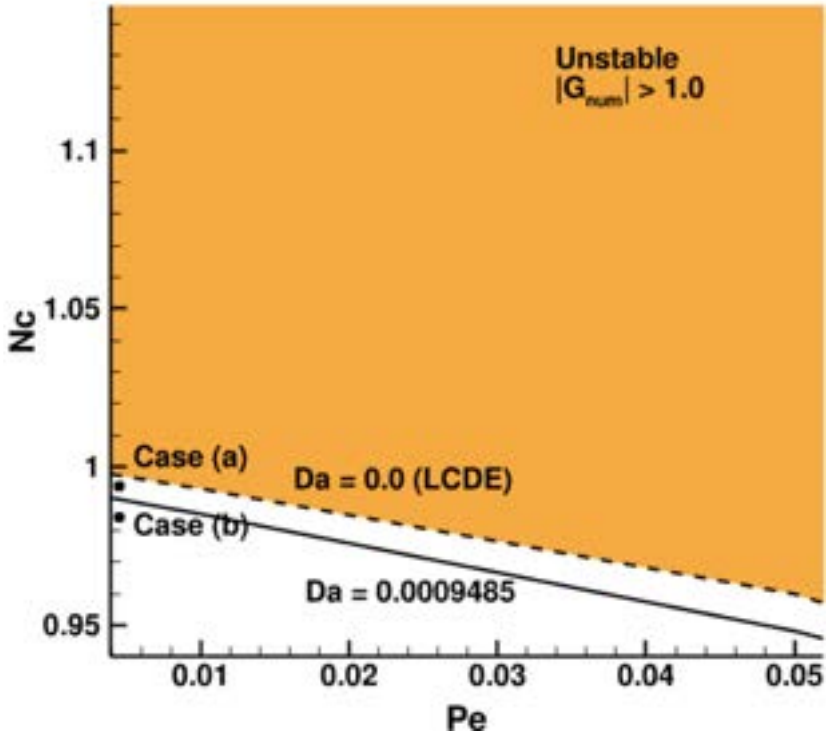
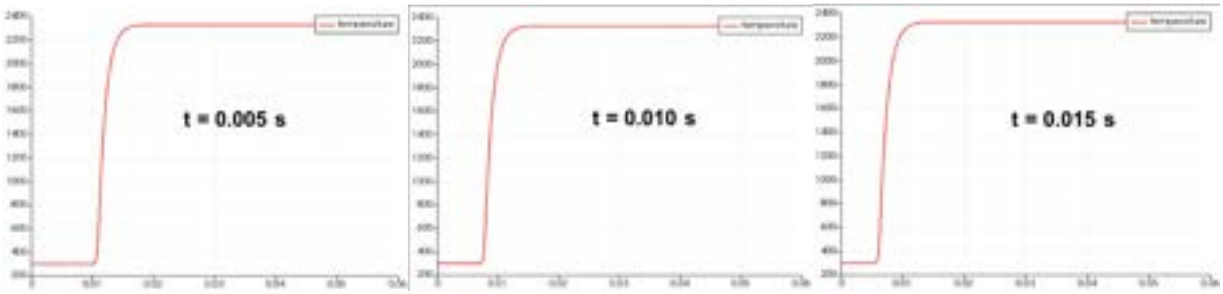


Figure 3.10: Stability curve for the cases at $Da = 0.0009485$ for the LW-CD$_2$ scheme compared with the LCDE case. The regions of numerical instability are colored in orange.

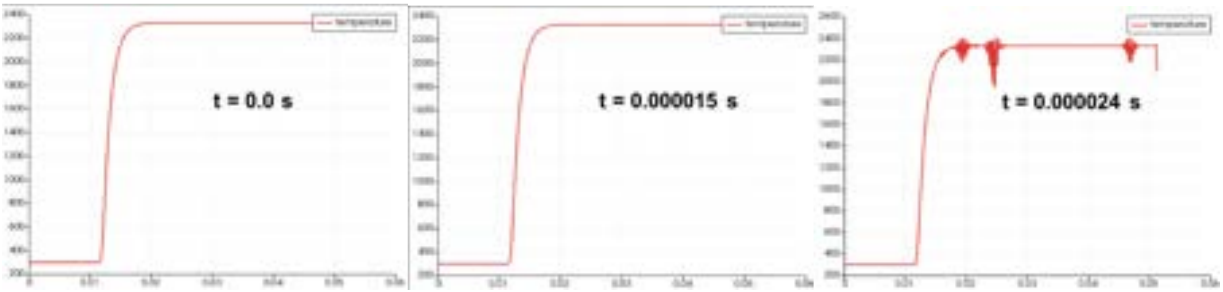
Δt	N_c	Pe	Nc_{crit} @ (Da = 0.0009485)
(a) 2.13×10^{-7}	0.995	1.89×10^{-3}	0.991 (U)
(b) 2.10×10^{-7}	0.981	1.86×10^{-3}	0.992 (S)

Table 3.1: Different values of time step used and the stability prediction obtained from GSA analysis at $Da = 0.0009485$.



(a) (b) (c)

Figure 3.11: Line plots of temperature along the center of the domain for the 1D fully premixed cases solved using AVBP at the mentioned times. This simulation was performed with the fixed time-step of $\Delta t = 2.1$ x10^{-7}, case (b) which is a stable time-step for this problem.



(a) (b) (c)

Figure 3.12: Line plots of temperature along the center of the domain for the 1D fully premixed case solved using AVBP at the mentioned times. This simulation was performed with the fixed time-step of $\Delta t = 2.13$ x 10^{-7}, Case (a) which is an unstable time-step for this problem.

mentioned in the frames. When the simulation is run for even long time, the temperature curve is stable and the flame is moving left towards the reactants. There is no disturbances or error noticed for this case when solving the reacting NSE problem.

Fig. 3.12 shows the line plot along the center-line of the domain of temperature in frames for **case (a)** with the times mentioned in the frames. The time step used for this present simulation is $\Delta t = 2.13$ x 10^{-7}, which according to property charts, should be a unstable simulation. Immediately, we can observe that within small time (few iterations) , instability is noticed in frames for both the temperature field and this results in solution blow up very soon afterwards in a few iterations. These oscillations start at the burnt gas and start disturbing the entire domain as can be seen through the movement of the oscillations. The values start increasing rapidly which eventually leads to solution blowup. So, by keeping everything else constant, and just by increasing the time step very

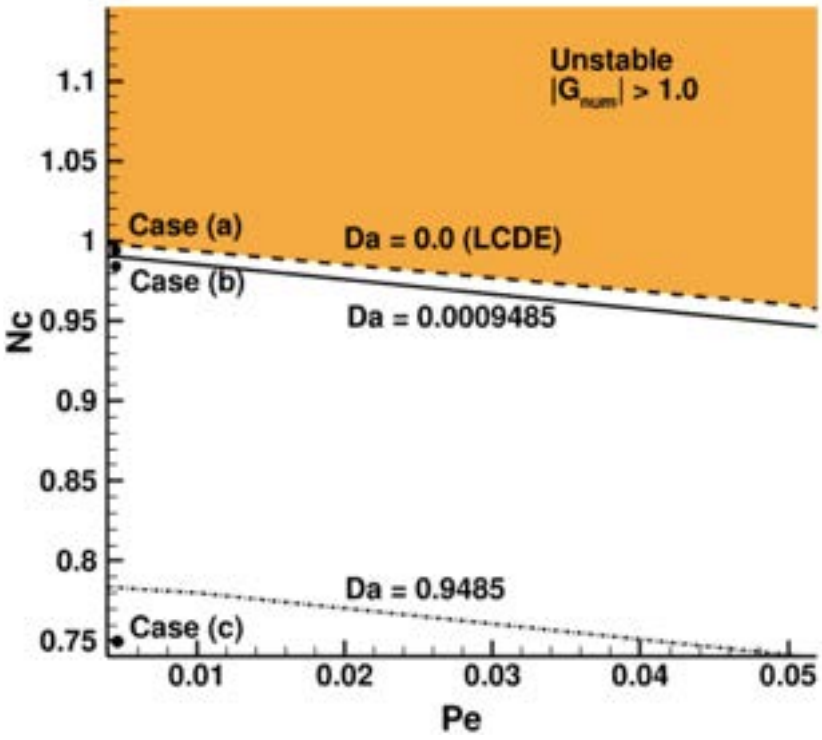


Figure 3.13: Stability curve for the cases at $Da = 0.9485$ for the LW-CD$_2$ scheme compared with $Da = 0.0009485$ and LCDE cases. The regions of numerical instability are colored in orange.

slightly, we go from a stable configuration **(case (b))** to an unstable one **(case (a))** and thus validates the findings on stability obtained by the GSA analysis for this realistic model and shows the importance of such an analysis when solving reacting flows.

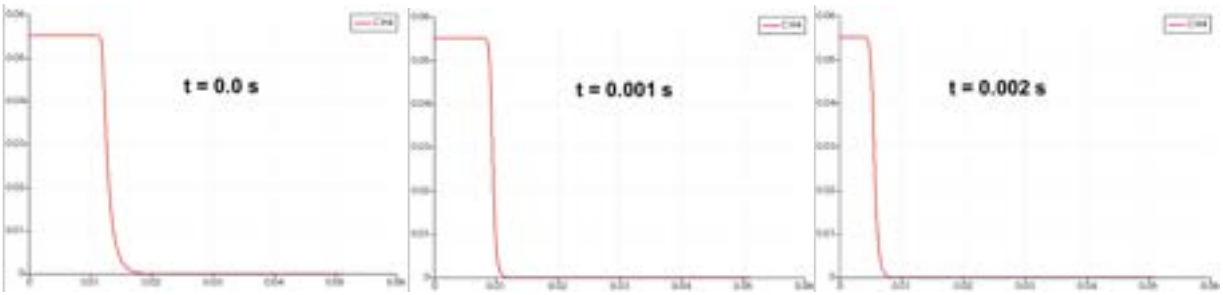
3.6.2 Effect of reaction source term on stability limits

The same test as in the previous section, is addressed here increasing the amplitude of B_0 by a factor of 1000. Since, the term B_0 only appears in the formulation for Da, everything else remains the same. The resulting value of Da is now $Da = 0.9485$. Similarly, as in the previous case, the stability contours in the $N_c - Pe$ plane are shown in Fig. 3.13 for the LW-CD$_2$ scheme. The stability limits, as shown in the previous subsection corresponding to LCDE and low value of $Da = 0.0009485$ are shown along with the new stability limits for the case at a much more realistic $Da = 0.9485$. At this value of Da, the critical value of N_c that can be used for stable simulations has reduced significantly from $N_c = 0.992$ (for low value of Da) to $N_c = 0.786$.

Three test cases are marked and detailed as given In Tab. 3.2. The first two

Δt	N_c	Pe	Nc_{crit} @ (Da = 0.0009485)
(a) 2.13×10^{-7}	0.995	1.89×10^{-3}	0.783 (U)
(b) 2.10×10^{-7}	0.981	1.86×10^{-3}	0.784 (U)
(c) 1.60×10^{-7}	0.748	1.42×10^{-3}	0.786 (S)

Table 3.2: Different values of time step used and the stability prediction obtained from GSA analysis at $Da = 0.9485$.

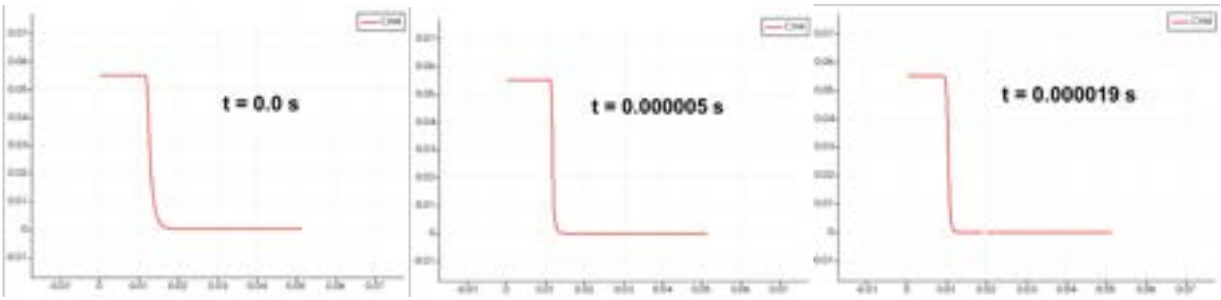


(a) (b) (c)

Figure 3.14: Line plots of methane mass fraction (Y_{CH4}) along the center of the domain for the 1D fully premixed case solved using AVBP at the mentioned times. This simulation was performed with the fixed time-step of $\Delta t = 1.6$ x10^{-7}, case(c) which is a stable time-step for this problem.

cases, **cases (a) and (b)**, are the same cases which were chosen before. However, at this value of Da, both cases are unstable as the operating value of N_c is much higher than the critical value of N_c. An additional case chosen, **case (c)**, should be stable for this higher Da value, LCDRE case with $\Delta t = 1.60$ x 10^{-7}. In this case, the operating N_c is within the value of the critical N_c. In conclusion, when the reaction source term was increased by a factor of 1000, the time step of stable simulation reduced from $\Delta t = 2.10$ x 10^{-7} to $\Delta t = 1.60$ x 10^{-7}.

Fig. 3.14 shows the line plot along the center-line of the domain of methane mass fraction with the times mentioned in the frames for **case(c)**. The time step used for this present simulation is $\Delta t = 1.6$ x 10^{-7}, which according to property charts discussed in the previous section should be a stable simulation. This is validated by the results plotted here for time evolution of fuel mass fraction. When the simulation is run for long time, the mass fraction curve is stable and the flame is moving left towards the reactants. There is no disturbances or error noticed in the simulation performed..



(a) (b) (c)

Figure 3.15: Line plots of methane mass fraction (Y_{CH4}) along the center of the domain for the 1D fully premixed case solved using AVBP at the mentioned times. This simulation was performed with the fixed time-step of Δt = 2.1 x10^{-7}, case (b) which is a unstable time-step for this problem.

Fig. 3.15 shows the line plot along the center-line of the domain of fuel mass fraction with the times mentioned in the frames for **case (b)**. The time step used for this present simulation is $\Delta t = 2.1$ x 10^{-7}, which according to property charts, should be a unstable simulation. Initially, the flame is moving towards the reactants, however, oscillations start appearing in this simulation in the fields of temperature, pressure etc which causes a discontinuity and crash as shown in frame (c). This time step corresponds to a stable simulation in the lower Da case explained before. The discontinuity is clearly visible which causes the simulation to crash. So, by keeping everything else constant, and just by increasing the reaction source term, we go from a earlier stable configuration to an unstable one (i.e. time step that was usable before results in a unstable simulation now). GSA analysis shows the importance of such an analysis when solving simple yet realistic, reacting flows. Here, in these studies the two variables that can be modified are the time step or grid spacing which are inversely proportional to each other. If timestep is increased we need to use a finer mesh to obtain instability. It is clear that when solving reacting flows its crucial to have sufficient resolution around a flame front otherwise we will run into trouble with stability of the scheme. In realistic, huge simulations it is impossible to have such a small mesh resolution throughout the domain. Therefore, its required to use a method like adaptive mesh refinement during the simulation to be able to have a properly resolved flame front. In the next part of the thesis, we will look at the algorithm and implementation of dynamic mesh refinement in our work.

3.7 Conclusions and Perspectives

GSA analysis is extended to the LCDRE now as it is important to perform the analysis performed in the previous chapter (CD equation) but for reacting flows. First, a constant reaction source term is used for both LW and TTGC schemes. The effect of the non-dimensional parameters on the amplification factor, group velocity for both numerical schemes were analyzed. The property charts for both the numerical schemes are compared to understand the differences in LW and TTGC schemes. The affect of adding the reaction source term in this chapter is analyzed on the property charts for both schemes - it results in the critical N_C value to reduce as the reaction source term becomes stronger.

A model with a realistic reaction term is designed: a reaction term formulation which is faithful to realistic gaseous combustion chemistry is explained. This has never been done before and is absolutely crucial for all future analysis using GSA for practical reacting flow problems. Property charts were used to obtain the stability limits for this model and then these findings were validated by solving a fully 1D premixed flame in a chamber. The effect of the reaction source term is studied on the stability limits and is validated using the same 1D premixed flame in a chamber. By increasing the reaction source term, a higher resolution or a smaller time step is required to remain within stability limits. Through solving the 1D premixed flame problem with NSE, it is evident that when solving reacting flows its crucial to have sufficient resolution around a flame front otherwise we will run into trouble with stability of the scheme. However, its is impractical to have such a fine scale resolution when solving large scale-realistic reacting flow problems. As a result, dynamic mesh refinement for realistic, large-scale reacting flows is introduced next to tackle this problem.

Part II

Dynamic Mesh Refinement

Dynamic mesh refinement: Methodology and simple validation cases

Contents

4.1 Introduction

In the previous two chapters, the numerical methodology and the spectral analysis of numerical schemes used in AVBP are presented. Thanks to this new analysis, it is confirmed that when solving reacting flows, it is essential to have sufficient resolution to resolve the problem suitably and obtain accurate solutions. However, only errors associated with the numerical schemes were discussed so far. Many important flow physics and features also need to be captured to obtain an accurate simulation for complex cases. Typically, time-dependent phenomena very often

change position in a given computational domain i.e. moving boundary problems, multi-phase fluid flows, shock-related problems, and many more. This is even more important when dealing with cases of transient combustion problems. In this last case, it is particularly difficult for a numerical problem to anticipate the process due to the coupling of the flow with the flame front and their combined effects. A huge challenge observed while solving these problems is the large scale of motions involved and ensuring an adequate resolution beforehand is nearly impossible unless a fully uniform mesh is used.

In this context, the main goal for the present work is to perform efficient LES of reactive and turbulent explosion based test cases. During such explosions of premixed gas mixtures, the main issue is the pressure increase (overpressure) which needs to be properly captured since it controls the intensity of the explosion. This overpressure can be fatal and cause the destruction of industrial facilities and buildings. However, these overpressures are difficult to predict since they are the result of complex and fully unsteady interactions between flame propagation, turbulence and the geometry. Different effects like turbulence and reaction will combine and affect the simulation. Two paths can be hence followed in terms of spatial resolution to try solving such problems. The first option is to make a well resolved mesh of the entire domain to ensure an accurate simulation. However, it is very often not economically feasible to proceed with this route. The second alternative is to track the important physics and adapt the mesh/grid dynamically as the simulation proceeds. The use of the second alternative although attractive requires however to understand the physics of importance for the underlying processes and the importance of the overpressure.

In configurations used to study gas explosions, chambers usually contain solid obstacles and the remaining volume is filled with a premixed flammable mixture. An initial flame kernel is then locally ignited and propagates initially in a laminar way in a flow initially at rest. In this phase, it is essential to properly resolve the flame front. However, very often the flame is too thin and cannot be fully resolved. A combustion model called the Thickened Flame model (TFLES) can be introduced to overcome this deficiency [116]. In this particular model, the flames are indeed artificially thickened so that the front is resolved on the available mesh without modifying the 1D laminar flame speed. As time evolves and the regime of combustion can change and it is essential to capture this thickened flame accurately in the chamber. Naturally, the tracking and capturing of the flame front becomes one of the necessary factors if a dynamic mesh adaptation is to be used.

As said before a critical phase is the transition of the laminar front to a turbulent front. This turbulent propagation phase mainly depends on the interaction of the flame front with the obstacles and surrounding flow that becomes unsteady or fully turbulent. Indeed due to the presence of obstacles, the flow becomes nonuniform giving rise to distortions of the flame which result in further modifications of the upstream flow. If turbulence is produced, the burning velocity will increase. This in turn increases the upstream flow velocity, producing more turbulence which further increases the burning velocity, and so on. There is therefore a positive feedback mechanism that continually accelerates the flame. This flame-induced turbulence leads to flames which can propagate at very large speeds compared to a standard laminar flame. In the worst scenario, the initial deflagration flame can even transition to a detonation. This acceleration of the flame front by obstacle-generated turbulence is the reason for many severe real-life industrial explosions [117]. For such cases, the main indicator of the quality of a simulation is the flame position visualization (flame speed) with time while the quantitative measure is the overpressure curve versus time. To calibrate and compare these quantities effectively, it is clear that one needs to track the turbulence generated by the interaction of the flow with the solid obstacles. As a result, a robust mesh adaptation scheme can be applicable for such problems only if the flame front and the turbulence generated in the chamber are tracked as the simulation proceeds and used to adapt locally the mesh quality.

Based on the above description, it is clear that Adaptive Mesh Refinement (AMR) or more generally the concept of changing the mesh resolution locally, thereby, maintaining the fine resolution in regions where it is required while keeping the mesh coarser everywhere else is key. Thanks to such an approach the number of mesh elements can indeed be significantly reduced. Provided an adequately refined initial mesh, it can be adapted while the physics of the flow evolves on the fly during the simulation. It results in significant improvement regarding memory at run-time as well as footprint compared to uniform meshes based prediction. However, using AMR in a parallel and high performance computing environments can yield substantial overhead due to mesh management difficulties and added CPU efforts. The advantages of using this method nonetheless outweighs the increase in overhead and complexity of implementation. Indeed if properly implemented, this approach requires less a priori knowledge on the evolution of the solution for the given physical system to be simulated [118]. As for the AMR implementation framework, it is usually eased thanks to available AMR-specialized libraries such as BoxLib [119], AMReX [120], libMesh [121], p4est [122], Peano [123], MMG [124] and others. Out of all of the libraries listed

here, MMG was used in our setup due to the ease of coupling it with our in house solver AVBP.

In this context, the AMR methodology and especially its implementation need to be strictly analyzed. To do so, in the following, two different algorithms are introduced and studied. One using a coupling with the YALES2 solver [125] (YALESadapt) and a second one using a library developed at CERFACS: Kalpataru (Treeadapt). The general steps in the algorithm laid out for any dynamic parallel mesh adaptation coupled with a LES solver is first explained in Section 4.2. This is followed by the discussion of the steps that are performed at the solver level before entering the mesh adaptation cycle algorithm (mask calculation based on the physics of interest, remeshing criteria triggering the adaptation). Next, the steps involving the mesh adaptation cycle are explained. The treatment and techniques followed to address differences in load balancing, partitioning and interpolation are underlined in Section 4.4. For a first assessment, two simple test cases are then chosen to validate the two tools. The first test case is the 2D Karman vortex street for which the results are discussed in Section 4.5.1. Finally in Section 4.5.2, a planar flame in a 3D chamber is simulated to validate the results obtained from both techniques in a reacting context. These simple test cases are chosen before moving on to complex reacting turbulent test cases, which are discussed in the next chapter. The goal of the current study is to implement an efficient mesh refinement algorithm and to implement it successfully for explosion cases.

4.2 Algorithm for parallel mesh adaptation

The general steps involved in parallel mesh adaptation if to be coupled with AVBP or any LES code solver are shown in Fig. 4.1.

The described logic applies to all tools discussed in this manuscript. Without going into the technical details, the steps involved are:

1. At every time integration step, on the current mesh, one needs to calculate sensors based on quantities of interest (QOI). Note that, the final sensor used i.e. the concatenation of the previous measures for all points of the mesh can be based on important flow features and is case specific.

2. Based on obtained values, a check is performed to see if we one is within the allowable threshold of the sensor chosen. Note that these limiting values are user defined and can be set according to the needs of the case.

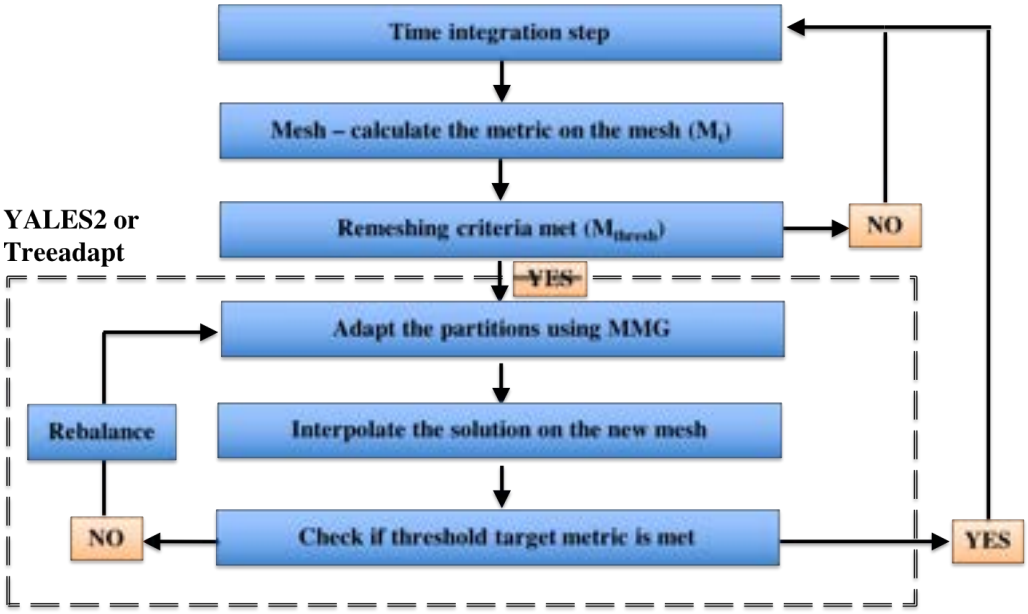


Figure 4.1: Overview of the iterative parallel mesh adaptation algorithm used here to perform simulations.

   (a) If adaptation is triggered, mesh adaptation is engaged as detailed in step 3.

   (b) If adaptation is not triggered, one goes back to step 1 for the next time integration step.

3. Each process enters the adaptation procedure by calling the MMG library [126].

4. After adaptation, interpolation of the sensor field as well as the solution field are produced to have solutions consistent with the adapted mesh.

5. The adapted mesh, the interpolated sensor field is once again verified to check if one is within the threshold of the target value.

   (a) If the objectives are not met, a mesh is re-partitioned and a new mesh adaptation attempt using MMG is activated as detailed in step 3.

   (b) If the target value is reached, the CFD solver proceeds to the next time integration step to go back with step 1.

These steps are followed by any generic dynamic mesh adaptation scheme irrespective of the LES solver. In Fig. 4.1, the first three boxes marked outside the dashed black lines are performed at the solver level. These are calculated at each time integration step and can be termed as **"precursor steps to the remeshing algorithm** and will be detailed next. In these steps, the parameters

used to identify the important physics related to a problem are to be provided. For any mesh adaptation algorithm to work successfully, this step is a key precursor to obtain accurate results. In the next section, how the physics identified for the problem is converted into a mask will be detailed.

In Fig. 4.1, the steps involved in the adaptation cycle are marked by the dashed solid black lines and they are performed by external softwares coupled with the LES solver. These steps can be performed by either of the mesh adaptation algorithms (YALESadapt or Treeadapt) and these are explained later in Section 4.4 and are termed as **"remeshing algorithm"**.

4.3 Precursor steps to the remeshing algorithm

The steps prior to the mesh adaptation call, are common to both mesh adaptation techniques. All of the following parameters are important and required before the mesh adaptation procedure can be initiated.

4.3.1 Identification of the physics of importance

The proposed mesh adaptation strategy relies on a mesh refinement procedure which can be applied to complex turbulent reacting flows. The criteria used to adapt the mesh should therefore be based on the important physics of such a problem [127]. This guides the choice of specific parameters also called Quantity of Interests which help highlighting all the important regions of flow associated with the specific problem and which eventually need to be modified through the mesh refinement process. In the cases of turbulent combustion, the two key parameters are the flame and the turbulence. The procedure to capture these quantities are detailed next. To accurately capture the proper simulation behavior, the mesh indeed needs to be refined enough in spatial zones where the flame and/or turbulence are of importance and present.

After identifying the right QOIs [128, 129] for explosion cases, the refinement algorithm is used in an iterative manner. Indeed fields/QOIs are calculated on a given initial mesh. To target zones that need to be refined given a mesh size and distribution, the concept of mask needs to be introduced. A mask M will indicate which zone needs to be refined, i.e where the mesh size should reach the target mesh size. The mask field will hence take values between 0 (where we want a coarse mesh) and 1 (where we want a refined mesh). For the problems discussed before, this mask field M is decomposed into a flame mask M_{flame} and

a turbulence mask M_{turb}, computed individually, and then assembled to get the final mask used in the adaptation algorithm. The strategy used to build the flame and turbulent masks is described in the following.

Flame detection and Flame mask

In LES of combustion problems, usually the flame is too thin and cannot be fully resolved. Therefore, a commonly used combustion model called the Thickened Flame model (TFLES) is used to overcome this deficiency [116]. In this particular model, the flames are artificially thickened so that the flame is resolved on the mesh without modifying the flame speeds [68, 130, 131, 132]. The flame is thickened by applying a thickening factor F to the diffusion terms while the reaction rates are divided by the same factor. This provides a flame propagating at the same laminar flame speed s_l^0 as the non-thickened flame but its thickness is increased and becomes $\delta_L^1 = F\delta_L^0$, alleviating the local mesh resolution requirement.

In such simulations, to guarantee that the flame is sufficiently resolved, 5 points (nF = 5) are usually needed in the flame front [133]. With such an objective if δ_L^0 is known, the value of the thickening factor F becomes a function of the desired number of points in the thickened front relative to the reference laminar flame and available local grid size. The local thickening factor F will therefore vary both in space and time due to the local grid variations.

In terms of approach the local thickening factor F is applied at the flame front thanks to a sensor which depends on the local temperature as well as mass fractions. This location is determined by a sensor θ_F which compares a local pseudo reaction rate of the flame Ω to a reference value Ω_0 extracted from a 1D laminar flame at the same operating conditions:

$$\theta_F = tanh\left(\beta^{'}\frac{\Omega}{\Omega_0}\right), \tag{4.1}$$

where $\beta^{'}$ is a constant and always set to 50. This value was chosen via experience from all reactive AMR simulations using YALES2 and Ω is defined as,

$$\Omega = Y_F^{n_F} Y_O^{n_O} exp\left(-\Gamma\frac{E_a}{RT},\right) \tag{4.2}$$

with Y_F and Y_O, the fuel and oxidizer mass fractions, E_a, the activation energy and T, the temperature. θ_F is called the flame sensor in the TFLES model [134] and is used to locate the flame and determines where thickening should be applied.

The flame location being found using θ_F, it can also be used to create a mask following:

$$M_{flame} = MAX(0.0, MIN(1.0, (\theta_F * 10.0))) \tag{4.3}$$

This mask takes by definition a value of 1 inside the flame and 0 outside with a smooth transition thanks to the hyperbolic tangent of Eq. (4.1) on either side of the flame front. The factor 10 of Eq. (4.3) is here used to sharpen the mask around the transition region of the flame sensor.

Thanks to this mask, a fixed thickening factor (corresponding to a fixed target mesh resolution) can be introduced to indicate the target and desired grid resolution of the simulated flame. This user defined parameter can then be easily adapted so the grid resolution can be locally obtained only at the flame front location.

Turbulence detection and Turbulence mask

The issue of turbulence detection is a broad topic and many criteria already exist. In AVBP, the detection of the turbulent zones for explosions is proposed to be based on a vorticity criterion. Mathematically, the vorticity of a three-dimensional flow is a vector field, usually denoted by $\vec{\omega}$, defined as the curl of the velocity field $\vec{v}$.

where, $\vec{\omega} = (\omega_x, \omega_y, \omega_z)$. Using these vorticity components, the magnitude of vorticity ($|\vec{\omega}|$) is a quantity that is closely linked to the zones where turbulence appears. In all of our simulations, the turbulence mask is activated using a threshold value i.e. a reference simulation is first used to pinpoint regions of vorticity and based on these results a threshold for vorticity is chosen to trigger an adapted simulation and a mask can easily be defined as,

$$M_{turb} = MAX(MIN(\frac{|\vec{\omega}| - \omega_{thresh}}{\omega_{thresh}}, 1.0), 0.0) \tag{4.4}$$

With such a method, the mask M_{turb} is dependent on this chosen threshold of vorticity magnitude and is always between 0 and 1. The value of the threshold is however dependent on the configuration and needs to be adjusted on a case-by-case basis. In most cases, a preliminary simulation, possibly without AMR is required to obtain an appropriate estimate of the threshold to be applied subsequently to all AMR based simulations.

When the mask is unity as in the case of the flame based sensor the mesh is refined to match the minimum edge length and the mask is zero in regions where the mesh will remain at the original edge length.

Having defined both masks, M_{flame} and M_{turb} can be used simultaneously in some cases to obtain a final mask (M) that is passed to MMG for mesh adaptation.

$$M = M_{flame} \cup M_{turb} \tag{4.5}$$

Wherever either of the masks are unity, the final mask is set to unity and only when both masks are zero, the final mask value is also set to zero. Just like these respective scalar masks, M_{flame} and M_{turb}, when M unity, the mesh is refined to the minimum edge length and where it is 0, the mesh is left unchanged. This mask is calculated at every time step of our simulation. However, to make sure that the cost of remeshing does not exceed the one induced by performing a simulation on a uniform mesh, a remeshing criteria is introduced next to have control on the remeshing frequency: a remeshing threshold, M_{thres}.

4.3.2 Mesh adaptation triggering

Mesh adaptation can theoretically be applied at every time step during the simulation, however it is not economically viable to do so. As a result, an important parameter usually introduced to limit the cost of triggering an adaptation at every time step attempts to control the frequency at which the mesh is adapted. This remeshing threshold (M_{thres}) needs to be chosen optimally so that the simulation is efficient while not compromising accuracy. To do so, the following measure at the n^{th} time step is proposed,

$$M^{*(n)} = \frac{\int_V |M_t^{(n)} - M_0| dV}{\int_V M_0 dV}. \tag{4.6}$$

Here, the initial mask field noted M_0, is fixed and does not change in time while the current mask field at time t (n^{th} time step) is noted $M_t^{(n)}$. M_t is updated at each iteration and follows the evolution of the quantities of interest.

From these two quantities, M^* can be calculated at each iteration to quantify the deviation between the two quantities. The variation of M^* can then be compared to a user input value M_{thresh}, the remeshing threshold (acceptable deviation of mask), With the objective that when M^* is larger than M_{thresh}, mesh adaptation has to be triggered. Note that after each adaptation, the mask M_0 is updated and set equal to M_t so the check can go on for the mesh at the next iteration of the solver as shown in Fig. 4.2. Based on experience and test cases, acceptable remeshing threshold values range from 2.5% to 5.0%. This corresponds that the current mask and the initial mask should stay within the

remeshing threshold value, if this is not true, mesh adaptation loop needs to be called.

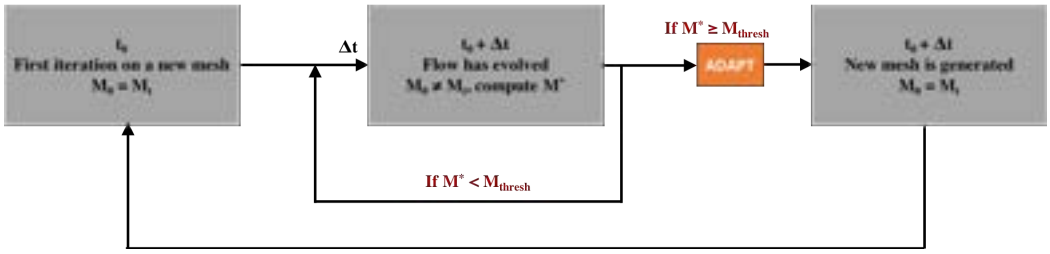


Figure 4.2: Schematic showing the triggering of mesh adaptation process.

It is clear that the mesh adaptation process as described previously is dependent on user defined parameters such as M_{thres}, creating one of the following conditions if not set appropriately:

1. For a value of M_{thresh} that is too high, the current mask may deviate significantly from the initial mask which means that the phenomena of interest will potentially no longer be located in the refined parts of the mesh and yield inaccurate results.

2. For a value of M_{thresh} that is too low, the mesh adaptation will be too frequent resulting in increased computational costs.

4.4 The Remeshing algorithm

At this point, we have reached the stage where a mesh adaptation cycle can be triggered as shown by the dashed black lines of Fig. 4.1. In the proposed algorithm, MMG is a key tool which is used to perform the adaptation. Static mesh refinement using MMG [126] along with the use of AVBP has been performed and validated at CERFACS [135] successfully before. The results demonstrated that sensor and adaptation approaches produce the desired local mesh refinement to match target losses measured experimentally [135]. In such a process, MMG produces isotropic meshes using a provided mask field. This software is based on local mesh modifications and a Delaunay kernel [136] to insert vertices [126]. However, MMG has a limit as of today: it is not a parallel library. There is no parallel management of the remeshing process nor load balancing management. In its basic version, MMG, is a single process shared memory tool. A parallel version has been developed since but it was not present at the beginning of this work. In terms of algorithms investigated here, there are two pipelines by which

the entire mesh adaptation algorithm can be performed using MMG. The objective of the present discussion is hence to evaluate both methods in terms of performance and accuracy.

First, the parallel implementation of MMG already carried out with the YALES2 code is coupled to AVBP [125]. To do so, people at CERFACS (Dr. Olivier Vermorel and Dr. Antony Misdariis) have coupled the solver YALES2 to manage the remeshing library MMG for massively parallel computations. This YALES2-coupled version is indeed able to use multiple processors with distributed memory. The coupling of AVBP and YALES2 using MMG is hereafter named **YALESadapt** and is one of the two techniques used to perform parallel mesh adaptation for a variety of reacting and non-reacting flows in the present and following chapters. However, due to copyright limits rights to YALES2 as well as the need to support YALES2 on top of AVBP, a second in-house methodology has been designed and implemented. **Treeadapt** is the second mesh adaptation technique tested here. Note that Treeadapt is a novel topology-aware online parallel mesh adaptation framework. This feature of Treeadapt is made possible using the in-house open-source parallel unstructured mesh partitioning and load-balancing tool TreePart [137]. This adaptation framework uses an iterative algorithm which leverages any given serial adaptation tool to obtain a massively parallel mesh adaptation while taking care of the load-balancing and global mesh data.

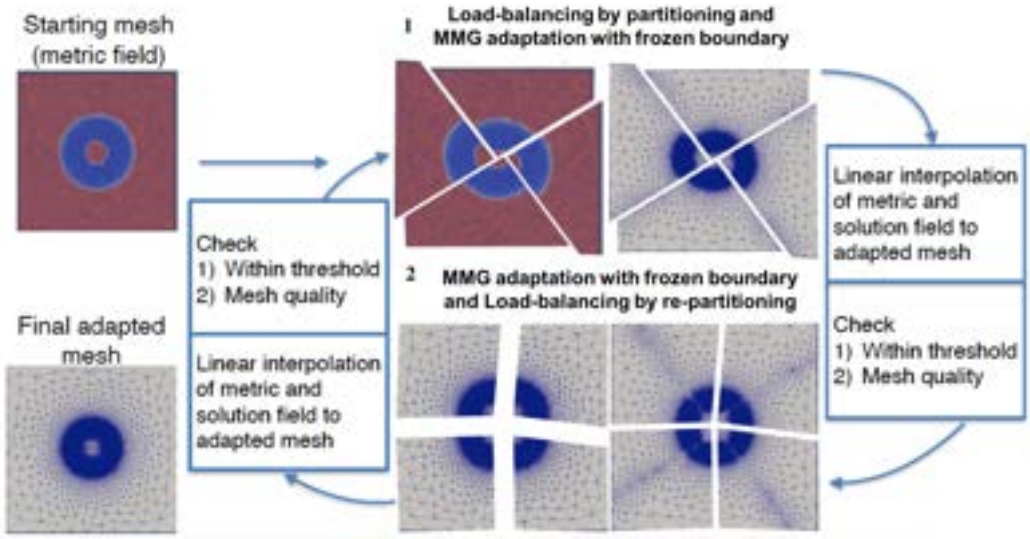


Figure 4.3: Overview of the iterative parallel mesh adaptation algorithm used in Treeadapt/YALESadapt.

Although the general algorithm of both pipelines is quite similar, small differences are present and listed below and described in Fig. 4.3 for Treeadapt as

well as described by Fig. 4.4. For the sake of the two algorithm descriptions, one presumes that the mask M is available and the remeshing threshold described previously are met so that mesh adaptation calls to YALESadapt or to Treeadapt are triggered. In such cases:

1. Each process of the simulation calls the MMG [126] library while freezing the subdomain/processor boundary nodes to produce a new processor subdomain mesh.

2. The CFD solution and masks are interpolated on the newly obtained mesh i.e. corresponding to the union of all new subdomain node redistribution/adaptation with fixed boundary nodes. If the target mask value is not reached, one moves to the step 3 otherwise, moves to step 4.

3. Load balancing and re-partitioning is called so that frozen subdomain boundary nodes become subdomain interior nodes. This step is needed to ensure that frozen mesh nodes can be adapted by moving back to step 1.

4. The desired adapted mesh has been obtained. The next time step of the CFD solver can proceed and the process repeats until adaptation is triggered again.

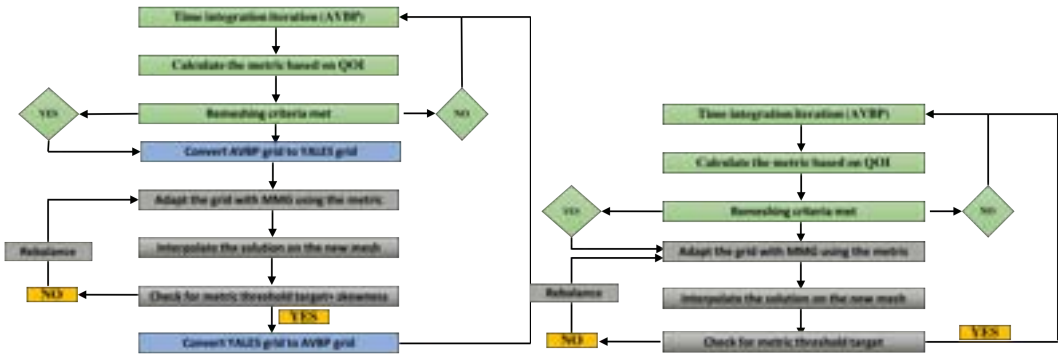


Figure 4.4: Adaptation interface using AVBP, MMG and (a) YALES2 and (b) Treeadapt.

It is important to underline that the complete parallel mesh adaptation algorithm procedure is an iterative procedure, which includes successive calls to the MMG library by each processor till a certain target or conditions are met. The previously described process is common to both adaptation tools i.e. YALESadapt and Treeadapt, although, differences exist. The small differences in the way YALES2 and Treeadapt are coupled with AVBP and MMG is shown in

frames of Fig. 4.4. There is clearly an extra step involved in YALESadapt, every time MMG library is called where the grid is converted from a AVPB format to YALES2 format and vice-versa and is marked in blue in Fig. 4.4 (a). In both the frames all the "Precursor steps to the remeshing algorithm" are marked in green and are identical for both adaptation techniques. While all the steps involved in "The Remeshing algorithm" are marked in grey and are performed by using MMG. Note that, in YALESadapt, an additional quality check of the mesh is performed: the skewness of mesh cells are verified to comply with YALES2 requirements.

The key differences when using either adaptation technique lie in the way partitioning, load balancing and interpolation are performed (Fig. 4.3). For partitioning and load balancing, Treeadapt uses a parallel hierarchical graph partitioner, while YALESadapt performs a serial partitioning using METIS [138]. For interpolation, Treeadapt uses the limited weighted least square interpolation method [139, 140, 141] while YALESadapt uses a simple linear interpolation. The data partitioning approach in both Treeadapt and YALESadapt are detailed next.

4.4.1 Data Partitioning

Hierarchical Partitioning: Treeadapt

As mentioned earlier, Treeadapt uses a topology-aware hierarchical unstructured mesh partitioning and load-balancing algorithm. To do so, it interfaces existing partitioning libraries like Zoltan [142], ParMetis [143] and provides a scalable infrastructure for use with high MPI rank counts. This method has been successfully used to partition up to $128K$ domains. With this new approach, the hwloc [144] library is used to query about the hardware hierarchy to build internal hierarchical MPI communicators for each level at run-time. The unstructured mesh is hence used as input via distributed chunks which can then be partitioned in any of the hierarchical levels by simply selecting the appropriate one.

There are two types of hierarchical communications in Treeadapt : (a) aggregation and (b) cascade. In the aggregation process, the data in a bottom level hierarchy is aggregated and moved to a top level hierarchy while the cascade process does exactly the opposite. Both operations are highly scalable because it does not involve network traffic across a node. The data used for the aggregation or cascade is called the payload. Any operation across a node is only performed at the top-most level. This way, the number of ranks that participate can be controlled or restricted in global communications to reduce network traffic.

In Treeadapt, domain decomposition is used for partitioning based on the hardware. The structure of the MPI tree topology is as follows. Nodes are the individual computing units connected by a common network to create a computing pool or cluster: each node being made up of multiple computing units (CPU) called sockets sharing a common NUMA memory pool. A socket comprises multiple cores that can simultaneously execute instructions. An example of the architecture and its topology is shown in Fig. 4.5. For the three level hierarchy shown in Fig. 4.5, one creates three MPI communicators, namely node, socket, and core communicators. Except for the node communicator, the socket and core are local to the node.

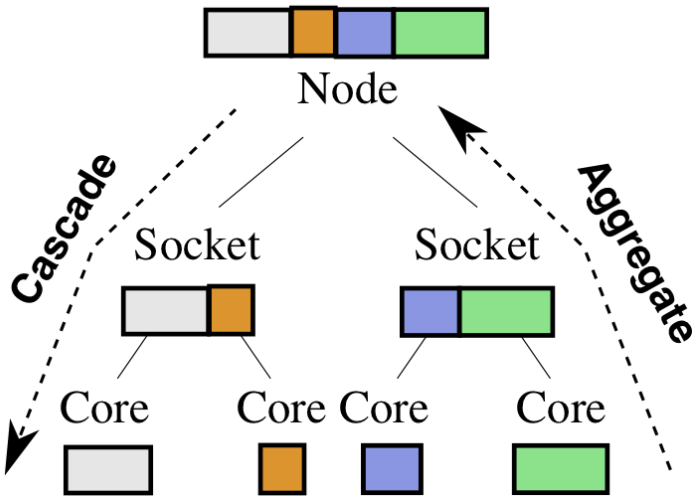


Figure 4.5: Structure of hardware topology for MPI Tree.

The partitioning then starts at the top most level of the topology referred to as the bootstrap partition level (BPL). To start the process at BPL, one requires an approximately partitioned mesh to be available as shown in Fig. 4.6. The payload can hence be positioned in any arbitrary level (usually the bottom-most level) and one has to aggregate the payload to the BPL. Based on the user input, a particular partitioning scheme is chosen and the mesh is redistributed based on the new partitioning. An illustration of this procedure is shown at the top of Fig. 4.6. After partitioning at the bootstrap level, there are two ways to proceed, (1) cascade equal chunks of mesh to the next level and perform distributed parallel partitioning or (2) perform a shared-memory parallel partitioning in the current level and cascade partitions to the next level. An illustration of both approaches is also shown in Fig. 4.6. Approach (2) is definitely more attractive as MPI communication is completely avoided. The partitioner terminates upon reaching the last level.

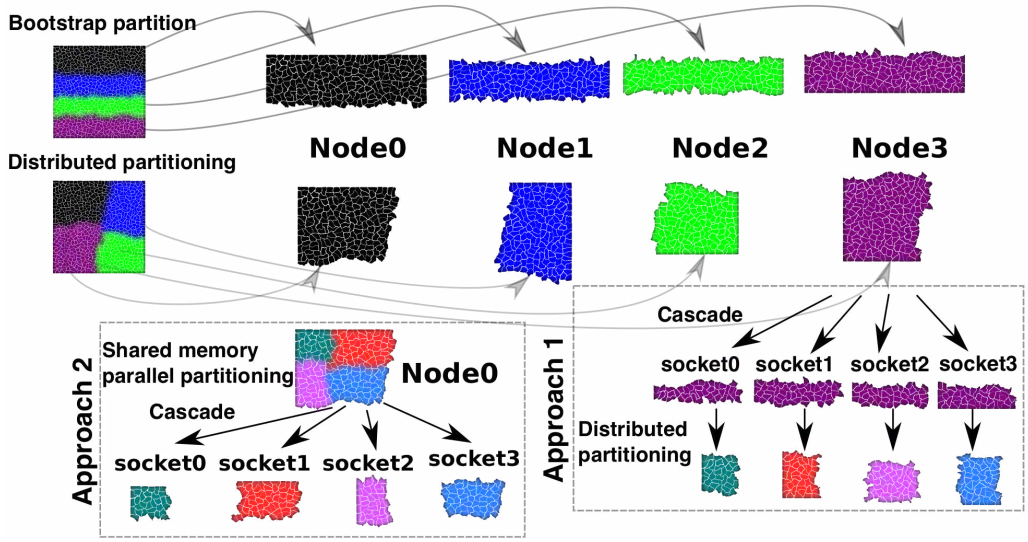


Figure 4.6: Bootstrap partition and the two approaches (1) and (2) to hierarchical partitioning considering two level node-socket hierarchy.

Two ways are available to interface the unstructured mesh data. In the library approach, the mesh is assumed to be already divided among all the participating ranks (lowest level) and is supplied to Treeadapt using a ParMETIS style function call to generate a refined partition. ParMETIS extends the functionality provided by METIS and includes routines that are especially suited for parallel AMR computations and large scale numerical simulations. Then the mesh data is aggregated to BPL to start the partitioning process. In the second approach, a standalone tool is used which can read two popular mesh/grid formats CGNS [145] meshes or the in-house HiP HDF5 unstructured mesh format [146]. For the following analysis, the library approach is used to interface with the AVBP solver.

Partitioning using METIS: YALESadapt

In YALESadapt, the partitioning is performed in a serial way using METIS [138]. Note that, it is critical to design algorithms that are able to produce good partitioning of highly unstructured graphs for efficient solvers and applications on parallel computers, especially when using solvers based on finite element methods. The distribution must be done in such a way, that one can have the same number of elements assigned to each processor, thus minimizing the number of adjacent elements assigned to different processors. Graph partitioning in that case is used to successfully satisfy the above two conditions by partitioning the finite element mesh with a graph, and then dividing it into equal parts [147].

METIS is a package used for partitioning large irregular graphs, large meshes, and computing fill-reducing orderings of sparse matrices. The algorithms in METIS are based on a multilevel graph partitioning and are described in detail in [138, 148]. In traditional graph partitioning algorithms, partition of the graph is computed by operating directly on the original graph. However, these algorithms are often too slow and produce poor quality partitions. Multilevel partitioning algorithms take a completely different approach [138, 148, 149], which reduces the size of the graph by collapsing vertices and edges, partition the smaller graph, and finally refine/coarsen it to construct a partition for the original graph.

Various studies have shown that the partitioning done by METIS is consistently better than those produced by other algorithms. These studies were performed on a large number of graphs arising in various domains including finite element methods, linear programming, VLSI. The partitions produced by METIS are usually 10% to 50% better than those produced by spectral partitioning algorithms [149, 150]. Studies on a wide range of graphs have shown that METIS is also one to two orders of magnitude faster than other widely used partitioning algorithms and the run time of METIS is comparable or even smaller than the run time of some geometric partitioning algorithms that usually produce much worse partitions. Also, the fill-reducing orderings produced by METIS are substantially better than those produced by other commonly used algorithms. For different classes of problems arising in scientific computations, METIS is able to reduce the storage and computational requirements of sparse matrix factorization methods by up to an order of magnitude. In YALESadapt, partitioning is done at 2 levels (cell groups and cells) and not on a single level. The load distribution is done at cell group level.

4.4.2 Load-balancing

Hierarchical load-balancing: Treeadapt

Hierarchical load-balancing is one of the most distinguishing and novel feature of Treeadapt. Mesh rebalancing using nodal and element weights in the tool has been implemented so it can be used at any given hierarchical level. This allows the user the advantage of using fast and scalable partitioning at the lowest level more frequently. Treeadapt uses shared-memory parallel partitioning at lower levels, as a result it totally avoids MPI communications as shown by Fig. 4.7. The rebalancing cost is thereby drastically reduced since the movement of data within the node is restricted. The expensive node level rebalancing of the mesh can be performed less frequently to considerably reduce the rebalancing cost.

Note that users can optimize the rebalancing frequency across levels for their specific application to achieve significant reductions in run-time.

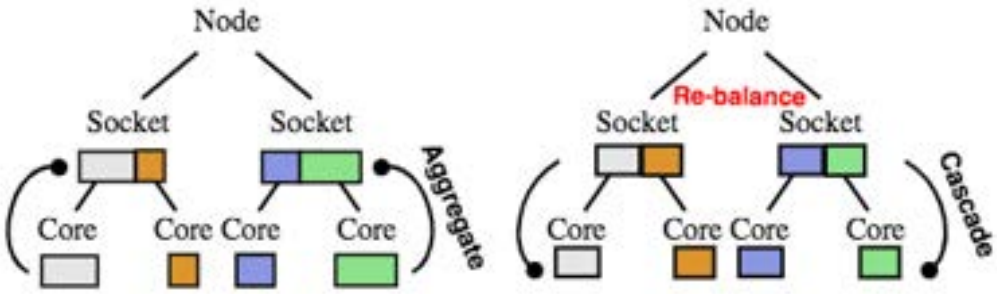


Figure 4.7: Rebalancing at a shared memory hierarchical level is three step process, aggregation, rebalance and cascade.

Load balancing using METIS: YALESadapt

The load balancing in YALESadapt is performed using METIS. This is required to ensure that the grid remains properly balanced from one rank to another: i.e. that all ranks have approximately the same number of tetrahedra. Another thing that needs to be done at this step is to enforce that the interface boundary nodes which were frozen are moved. To achieve this, one solution is to impose strong weights at the connections between the elements at processor interfaces. The load balancing algorithm will thus be strongly forced to place new interfaces away from the previous interfaces. Thanks to this logic, the process can be repeated till convergence is obtained. This method of ensuring that the frozen nodes are adapted in the next sub-iteration. In YALESadapt, load balancing is done at the cell group level to make sure that connectivity graphs of all elements is not required.

4.4.3 Interpolation

Interpolation is an important step of AMR. The solution and the mask are indeed interpolated every time an adaptation loop is produced. This can be furthermore memory consuming as well as CPU intensive and thus needs to be handles with care.

Limited weighted least-square interpolation: Treeadapt

In Treeadapt, a KDTree search [139, 140, 141] is applied to form the stencil to which the interpolation technique is applied. A k-d tree (short for k-dimensional tree) is a space-partitioning data structure for organizing points in

a k-dimensional space. K-d trees are a useful data structure for several applications: searches involving a multidimensional search key and creating point clouds for example. The k-d tree is a binary tree in which every node is a k-dimensional point. Every node that is not a leaf can thus be considered as a splitting hyperplane that divides the space into two parts known as half spaces. Points on either sides of this hyperplane are represented by a left subtree and a right subtree. The hyperplane is chosen so that every node of the tree is associated to one of the k-dimensions and the hyperplane is perpendicular to the dimension's axis. For example, if in a particular split the x-axis is chosen, all points in the subtree with a smaller x value than the node will appear in the left subtree and all the points with larger x value are placed in the right subtree. The hyperplane is set by the x-value of the point and its normal is the unit x-axis. The multidimensional binary search tree is a data structure for storage of information to be retrieved by associative searches and is quite efficient in its storage requirements. Indeed, a nearest neighbor query can be performed such that, given a distance function D, a collection of points B (in k-dimensional space), a point P (in that space), the nearest neighbor Q of point P in B can be found. More details about the algorithm can be found in [139]. Using this methodology the stencil to perform limited weighted least-square interpolation is formed.

With Treeadapt, the limited weighted least-square interpolation technique is used to interpolate the mask and solution on the adapted mesh. Weighted least-squares (WLS) estimation is an intuitive solution, which multiplies a weighting matrix to residuals, making minimum modification to the original solution. The weighted least-squares procedure is used to evaluate the derivatives of the fluxes and the convention as shown in Fig. 4.8. The collection of neighboring points j of point i are called the support stencil or connectivity of point i and is represented as C(i). The polynomial least-squares for gradient approximation and interpolation is used as follows:

a) Lets assume a first order polynomial variation of function f over the neighborhood point i, have the form,

$$f = a(x - x_i) + b(y - y_i) + c(z - z_i) + d. \tag{4.7}$$

There are 4 values required to determine the coefficients a, b, c and d. However in most cases, the number of neighbors i chosen are more than 4, which leads to an over-determined system. One then uses the weighted least-squares approximation and obtains the values of the coefficients of the polynomial as shown below,

$$\begin{bmatrix} \Sigma w_{io} & \Sigma w_{io}\Delta x_{io} & \Sigma w_{io}\Delta y_{io} & \Sigma w_{io}\Delta z_{io} \\ \Sigma w_{io}\Delta x_{io} & \Sigma w_{io}\Delta x_{io}^2 & \Sigma w_{io}\Delta x_{io}\Delta y_{io} & \Sigma w_{io}\Delta x_{io}\Delta z_{io} \\ \Sigma w_{io}\Delta y_{io} & \Sigma w_{io}\Delta y_{io}\Delta x_{io} & \Sigma w_{io}\Delta y_{io}^2 & \Sigma w_{io}\Delta y_{io}\Delta z_{io} \\ \Sigma w_{io}\Delta z_{io} & \Sigma w_{io}\Delta z_{io}\Delta x_{io} & \Sigma w_{io}\Delta z_{io}\Delta y_{io} & \Sigma w_{io}\Delta z_{io}^2 \end{bmatrix} \begin{bmatrix} d \\ a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \Sigma w_{io} f_o \\ \Sigma w_{io}\Delta x_{io} f_o \\ \Sigma w_{io}\Delta y_{io} f_o \\ \Sigma w_{io}\Delta z_{io} f_o \end{bmatrix}. \tag{4.8}$$

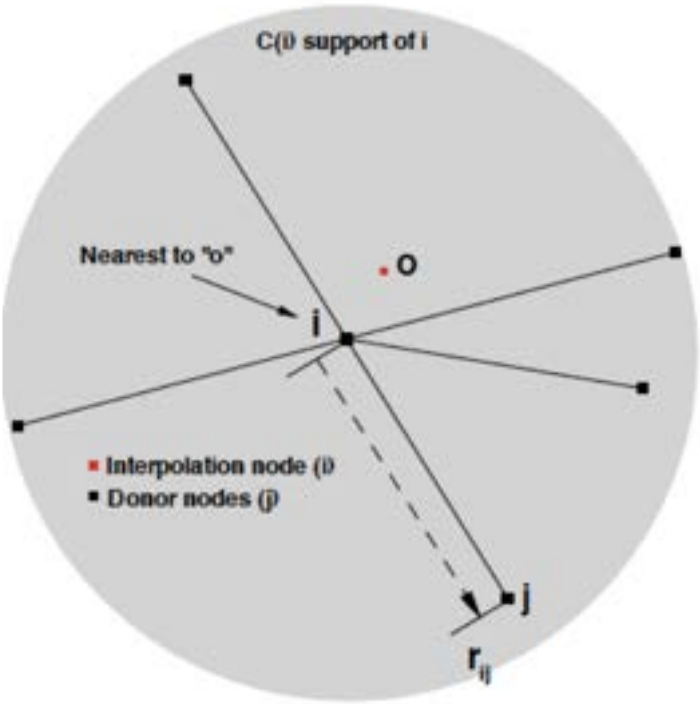


Figure 4.8: Least square procedure for flux derivative evaluation.

Note that Eq. 4.8 can be rewritten in the coefficient form,

$$\begin{bmatrix} d \\ a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \Sigma \zeta_{io} f_o \\ \Sigma \alpha_{io} f_o \\ \Sigma \beta_{io} f_o \\ \Sigma \gamma_{io} f_o \end{bmatrix}, \tag{4.9}$$

where, α, β, γ and ζ are the least-square coefficients for a given edge i_0. The interpolation values are then limited to be between the min/max of the stencil while interpolating. Note that, when the stencil is deficient and don't have 4 neighboring points to perform the interpolation, one can fall back to the inverse-distance weighted average approach to perform the interpolation.

Inverse distance weighting (IDW) is a type of deterministic method for interpolation with a known scattered set of points. The assigned values to the

unknown points are calculated with a weighted average of the values available at the known points. The name given to this type of method was motivated by the weighted average applied, since it resorts to the inverse of the distance to each known point when assigning weights. A general form of finding the interpolated value u at a given point x based on samples $u_i = u(x_i)$ for $i = 1, 2, ..., N$ using IDW reads,

$$u(x) = \begin{cases} \frac{\Sigma_{i=1}^{N} w_i(x) u_i}{\Sigma_{i=1}^{N} w_i(x)}, & \text{if } d(x, x_i) \neq 0 \text{ for all i} \\ u_i, & \text{if } d(x, x_i) = 0 \text{ for some i} \end{cases},$$

where, $w_i(x) = \frac{1}{d(x,x_i)^p}$ is a simple IDW weighting function [151], x denotes the arbitrary interpolated point, x_i is the known point and d is a given distance from the known point x_i to the unknown point x. N is the total number of known points used in the interpolation technique and p is a positive real number, represented by the power parameter. In the above expression, weight decreases as the distance increases from the interpolated points. The greater the value of p is the greater is the effect of closer points to the interpolated point. The average number of nodes considered for interpolation is 4.

Linear Interpolation: YALESadapt

In YALESadapt, interpolation of all numerical quantities is done from the initial mesh to the adapted mesh at each sub-step, benefiting from the 2-level domain decomposition to easily locate the nearest source tetrahedron/triangle from a given destination element. Here, basic linear interpolation is used as the mesh doesn't usually change at a front or interface location when using dynamic mesh adaptation, as a result linear interpolation has indeed been found to be sufficient [125].

4.4.4 Other Parameters

Apart from the differences explained above there are additional parameters that need to be explained and understood to fully understand the mesh adaptation algorithms. Some of these parameters are common to both Treeadapt and YALESadapt while some are only used in one or the other.

Element growth rate

The element growth rate noted h_{grad}, is a parameter that is used in both Treeadapt and YALESadapt. It represents the maximum size ratio between two adjacent edges and allows a smooth transition between the refined and coarse mesh regions. The values of h_{grad} can be set between 0 and 1: 0 leading to a homogeneous refined mesh and 1 leading to a transition from the refined to the coarse mesh using a factor of 2 in size along an edge. From a numerical point of view, this parameter needs to be appropriate to avoid sharp transitions which could lead to a loss of precision or could create numerical issues. For simulations with AVBP, the maximum value recommended is 0.5 to avoid numerical issues. This is a user defined threshold and for the simulations performed in this manuscript the value was chosen to be 0.3. This parameter is used in both adaptation techniques and is imposed every adaptation step but is not a thresholding criteria. The mask is adapted a priori internally in MMG so that the resulting mesh conforms to this value.

Target Mask deviation

A mask M defines the zones where the mesh needs to be refined, i.e where the mesh size Δx should reach the target mesh size Δ_x^{ref}. The value Δ_x^{ref} is a user defined parameter and is also known as the min edge length. The target deviation is the deviation between the target and the current edge size and is the check that is used in both adaptation techniques before moving to the next time integration step. The deviation has been set to 60 % by default through experience using YALES2, although this value has no specific basis. As a result, the same value is used for all simulations using Treeadapt and YALESadapt.

Check for skewness

In YALESadapt, there is an additional check that is performed and used to modify the mesh on top of the check for the target deviation which is not done in Treeadapt. A mesh optimization technique has indeed been developed and used [125] based on the improvement of the cell skewness [152]. The skewness k, is defined as the ratio between the difference of volume of an equilateral tetrahedron of circumsphere k and the actual volume of k and volume of an equilateral tetrahedron in the circumsphere of $k = (\frac{V_{ref}-V}{V_{ref}})$. This method uses edge swaps and node re-allocations. In this approach, the new position of the vertex is found so that skewness is improved of all the elements associated with the node. In

YALESadapt, convergence is only achieved when the difference between the target and resulting masks (explained in detail later) is below a threshold value and when the element skewness does not exceed a limit value. This is a user defined parameter and the target value is usually set to 0.3 % to obtain accurate results. All the key parameters and differences between both adaptation techniques having been discussed, simple test cases are now produced and detailed thereafter to validate the two techniques.

4.5 Validation of the adaptation tools

The goal of both chosen test cases is to validate the mesh adaptation algorithm explained so far and to replicate results obtained elsewhere. These test cases are furthermore chosen since they are ideal for testing both non-reacting and reacting flows in a simple yet realistic environment.

The first test case addresses the 2D Karman vortex street problem while the second test focuses on the propagation problem of a planar flame in a 3D domain. The objectives are first to ensure that the physics of interest is well captured meaning that the appropriate mask is selected and triggers correctly mesh adaptation. At the same occasion, one can verify that simulations are more effective when AMR is activated and if compared to reference results obtained with a given high resolution mesh.

4.5.1 2D Karman vortex street

Multiple test cases have been studied using YALESadapt. These include a turbulent plane jet, the flow in a meso-scale combustor [125] and fuel injection problems in aeronautical combustion chambers [153]. As a result, one focuses here on Treeadapt. In this section, a 2D Karman vortex street problem is studied first to validate the Treeadapt adaptation algorithm by comparing the prediction to a reference simulation. The objective of this test case is to study the flow around a cylinder at a Reynolds number of $Re = 100$.

In the methodology of dynamic mesh adaptation algorithm explained so far, two quantities of interest are mainly used to obtain masks to be used for the AMR procedure. The 2D Karman vortex street problem is an ideal candidate to test non-reacting flows where the vorticity based sensor can be tested for a simple yet realistic problem. In the Karman vortex street problem, the shedding frequency of vortices released downstream of the cylinder can be used as a quantitative measure to verify AMR against static-reference simulations.

Numerical Setup

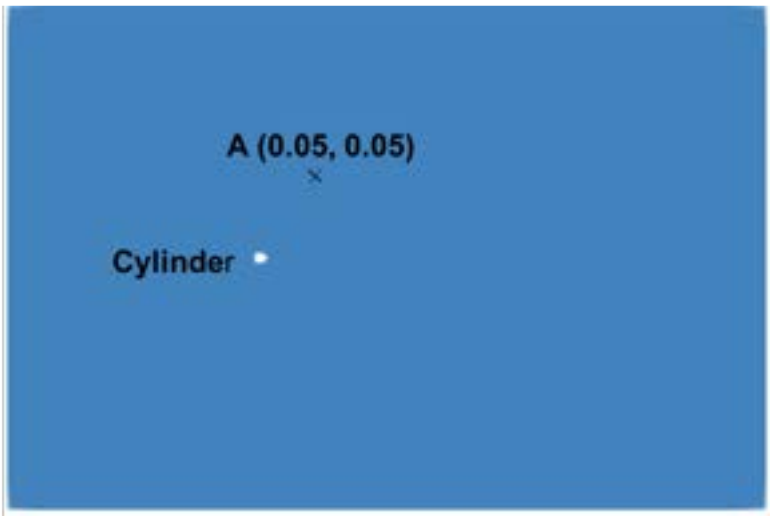


Figure 4.9: Schematic of the numerical domain for the 2D Karman vortex street problem. The probe (point A) is located at the coordinates mentioned.

Figure 4.9 shows a schematic of the computational domain used for the current study. The flow direction is from left to right. A cylinder of diameter, $d = 0.03$ m, is present in the center of the domain. The center of the computational domain coincides with the center of this solid cylinder. The length of the domain is 1.8 m x 1.2 m in the x and y directions respectively. A probe (point A - marked in Fig. 4.9) is located in the domain at the given coordinates to collect data. A reason for selecting this location is that in the mesh adaptation cases, the region lies in an area of the mesh that undergoes adaptation. Since it is important to ensure that obtained flow properties are not affected by the mesh adaptation methodology. Results obtained here should be independent of the method used to produce the simulation results.

The mesh used for the reference simulation as well as the initial mesh used for the adapted case with Treeadapt are shown in Fig. 4.10. The reference mesh has approximately $110,000$ elements, while in the case of the adapted mesh the number of elements in the initial mesh is approximately $22,000$ cells. The difference in resolution between the two meshes is shown in Fig. 4.10. The reference mesh has the smallest cell corresponding to the min volume, $vol_{min} = 0.1395E-07m^3$. While in the case of the adapted simulation, the smallest cell volume, $vol_{min} = 0.8828E-07m^3$, yielding a volume of the smallest cell that is approximately 6 times that of the reference case.

As shown later, the adapted simulation proceeds using the mask based on vorticity so the mesh adapts as the flow field evolves. The number of elements

in the adapted mesh therefore keeps on increasing as the vortex train sheds and propagates downstream the cylinder. The number of elements in the final adapted mesh: i.e. at $t = 0.25s$ is approximately $109,000$ which is close to the number of elements of the reference simulation.

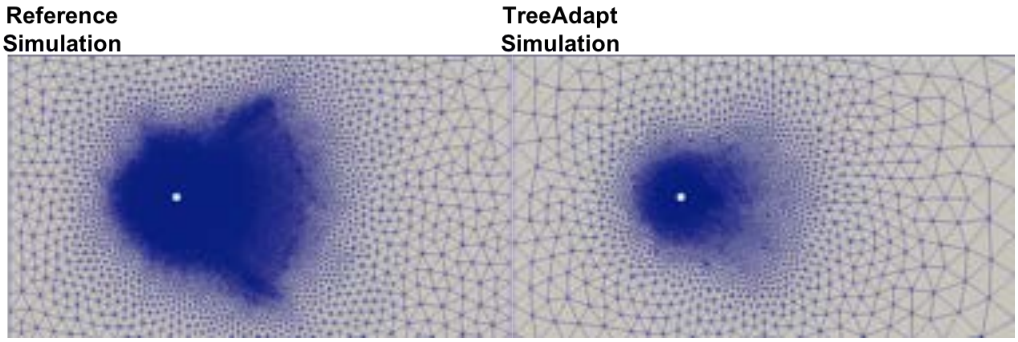


Figure 4.10: Comparison of initial meshes used in the (a) uniform reference simulation and (b) adaptation simulation using Treeadapt.

For the specific problem simulated here, an inflow velocity of $U_0 = 50$ m/s is imposed at the inlet. In the computational domain, ambient conditions of pressure and temperature are applied: $P = 1bar$, $T = 300K$. Air ($Y_{air} = 1.0$) fills the entire domain at all time.

In terms of boundary condition specification, the flow going from left to right, inlet non-reflecting condition with a prescribed velocity is set on the left and an outlet is set the right using a non-reflecting condition with a relaxation on pressure. The top and bottom sides of the computational domain are set as outlet boundary conditions with relaxation on pressure set at ambient conditions. The solid cylinder is prescribed to be no-slip adiabatic wall condition.

For numerical integration, a centered continuous Taylor-Galerkin scheme that is third-order in space and time [68] is used. Note finally that no LES model is used the problem being highly resolved and with a low Reynolds number. We are here in the context of DNS and turbulence modeling is not required.

Quantity of interest

The test case is non-reacting, so the area of interest is based on the flow evolution around and downstream the cylinder. In the early phase, the wake of the cylinder is expected to oscillate with a very small amplitude. With time, the oscillation amplitude increases until vortices shed periodically in the wake of the cylinder. To adapt the mesh to such a physics, a QOI is needed and chosen here to be based on the vorticity to capture the expected periodic shedding of vortices in the wake. For the case of interest, an appropriate vorticity threshold of is chosen

so all the important features of the flow are captured as seen in the reference simulation without mesh adaptation. A vorticity threshold of $\omega_{thresh} = 250s^{-1}$ is here used. This threshold is needed to normalize the vorticity magnitude calculated during the simulation in the domain to obtain a mask between 0 and 1 as explained in Section 4.3.1. An illustration of the mask field around and in the wake of the cylinder is shown in Fig. 4.11 . The minimum edge length needed to limit adaptation is set to $5e-4$ m which corresponds to the mesh resolution ($\Delta x_{ref} = 0.0005$ m) in the refined region of the reference simulation. Other parameters needed for AMR are set such that: the element growth rate $h_{grad} = 0.3$ and the target deviation is default value of 60%. The mask field shown on Fig. 4.11 corresponds to a view taken at an early instant of the adapted simulation showing that selected threshold values used are sufficient to encompass the vortical structures of importance which eventually result in the periodic vortex shedding.

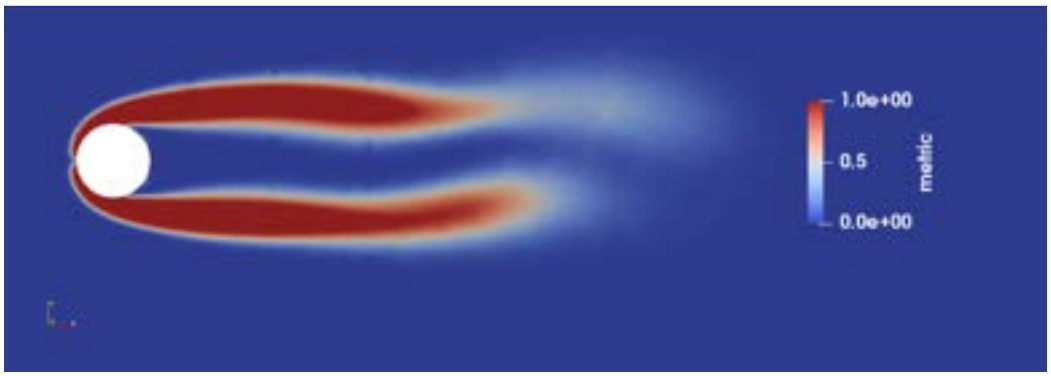


Figure 4.11: Mask based on vorticity around a cylinder of diameter, d, capturing the periodic shedding of vortices from the wake of the cylinder.

Results

The following discussion describes in more details obtained results. Prior to detailed flow snapshot comparisons, more quantitative results are probed. Typically, Fig. 4.12 shows the temporal evolution of the stream-wise velocity value taken at the probe location A [154]. This point is chosen as it is slightly outside of the wake and it maximizes the influence of the upper vortex street. As anticipated, after some time, the amplitude of the oscillation has reached a periodic regime and vortices are shed into the downstream flow on either sides of the cylinder. As seen from Fig. 4.12, the trends of the registered stream-wise velocity signal at probe A for both the simulations are similar confirming the good agreement between both simulations at this location. Similarly to the axial velocity signal,

the temporal evolution of the pressure signal at probe location A is shown in Fig. 4.13. In agreement with Fig. 4.12, both simulations reach a steady state and a periodic oscillation clearly settles.

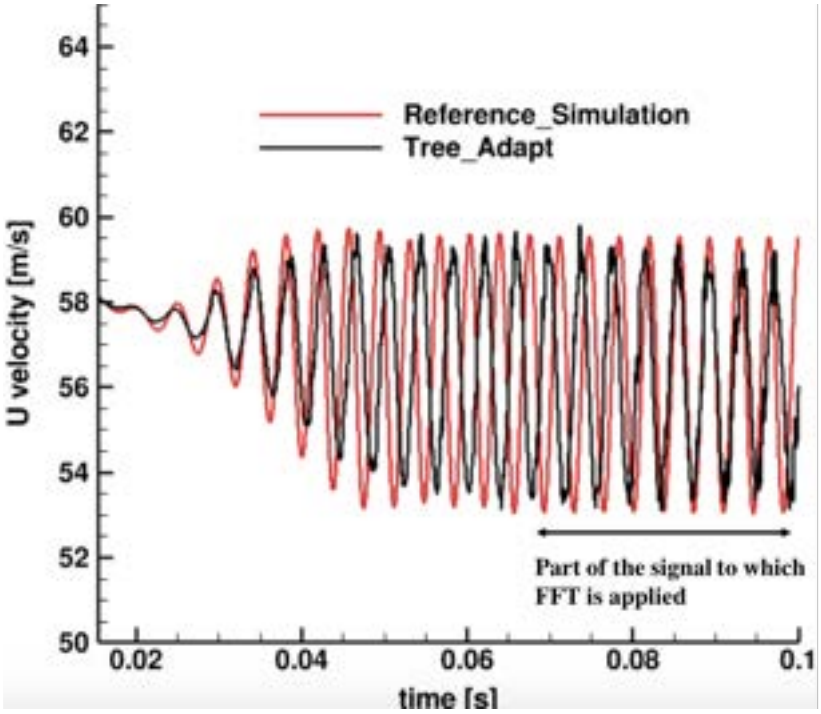


Figure 4.12: Temporal recording of the stream-wise velocity component at the probe (point A) location for the reference and adapted simulations.

Using the stream-wise velocity, U_0, the non-dimensional Strouhal number can be calculated as:

$$St = \frac{fd}{U_0}. \tag{4.11}$$

Here, f is the shedding frequency, d is the cylinder diameter and U_0 is the inflow velocity. The fundamental shedding frequency, f, can be easily obtained by performing the power spectral density (PSD) of the stream-wise velocity temporal recording. FFT is applied only to the periodic part of the signal as marked in Fig. 4.12. For the reference simulation, a peak is observed at f = 274 Hz. In the adapted case, where the initial mesh has only $20K$ elements, the retrieved shedding frequency is around f = 257 Hz, yielding respectively $St = 0.1644$ and $St = 0.1542$. Note that although the grid spacing for both cases is different results appear very similar. This allows to conclude that we are able to obtain the shedding frequency adequately with mesh adaptation.

Although the first analysis indicates a proper behavior of the AMR approach, the analysis remains highly local and a more thorough analysis of the entire field

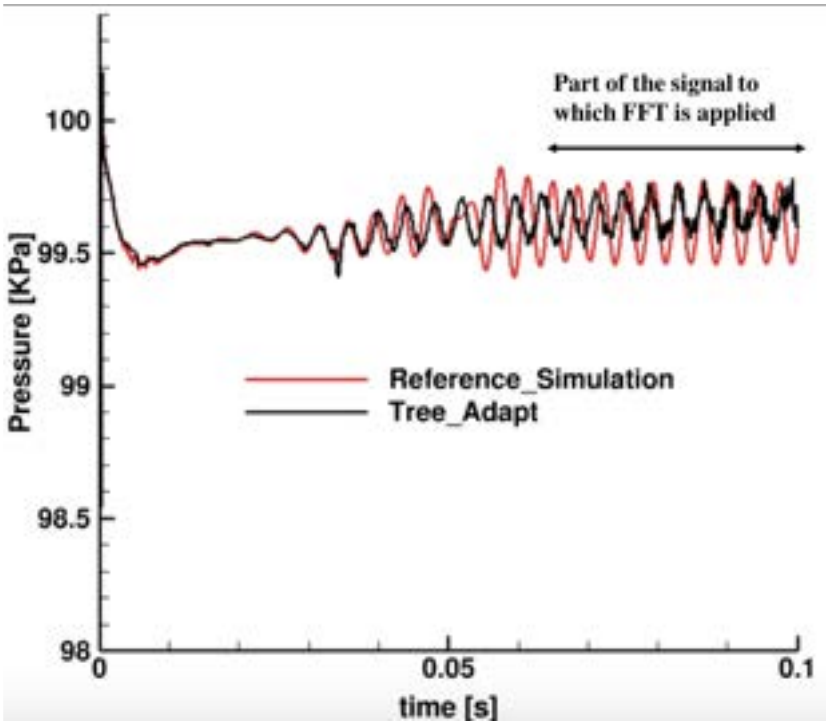


Figure 4.13: Temporal recording of the pressure signal at the probe (point A) location for the reference and adapted simulations.

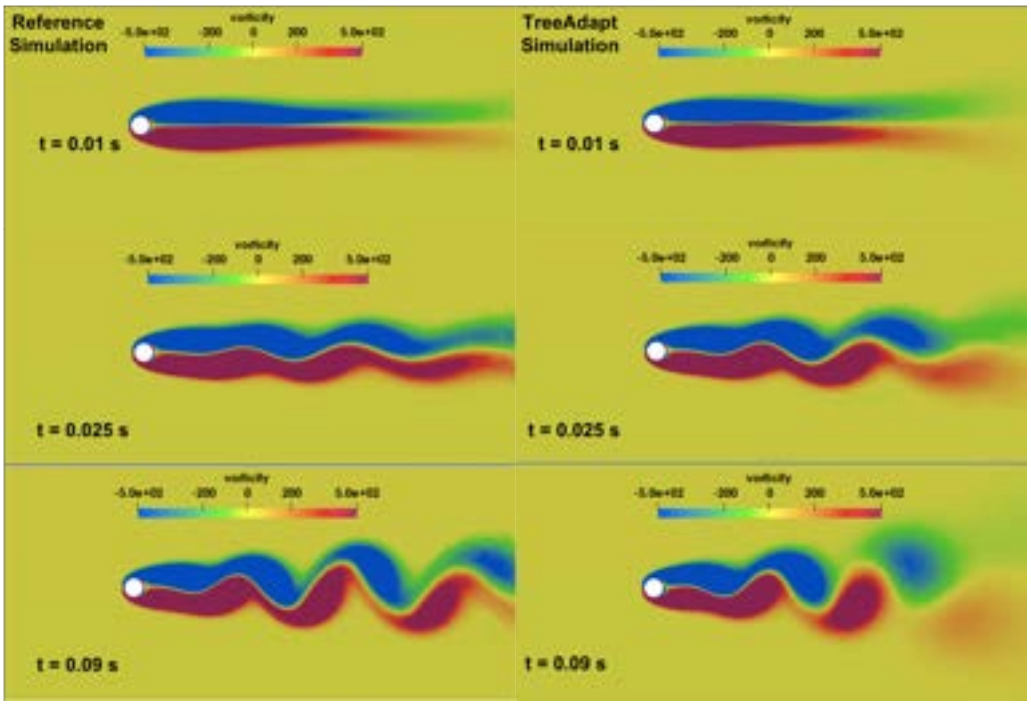


Figure 4.14: Snapshots of the vorticity at the indicated times for the reference and adapted simulations.

and simulation is needed. To do so, snapshots of the vorticity field for both the reference and the adapted simulations at the indicated times are shown in

Fig. 4.14. In the left column, the results for the reference simulation are shown while on the right, the snapshots at the same time instants are obtained using Treeadapt. In the early phase of each simulation, the wake starts to oscillate with a small amplitude generating no clear vortex shedding. It is only later that the amplitude of the oscillation increases and vortices are shed for both cases. Note that it can be clearly seen that downstream of the cylinder in the case of the adapted mesh, structures are dissipated throughout the duration of the simulation. This is explained by the fact that the adapted mesh remains coarser in this region at all instants. This potential deficiency is expected to disappear for longer duration simulations so that the algorithm has time to adapt to this convective process. Note however that despite this limitation, at $t = 0.09$ s, the periodic shedding is present in both simulations, the vortical structures being slightly more dissipated with AMR because of threshold value used potentially induces a relatively coarse mesh resolution in that case in comparison to the reference one.

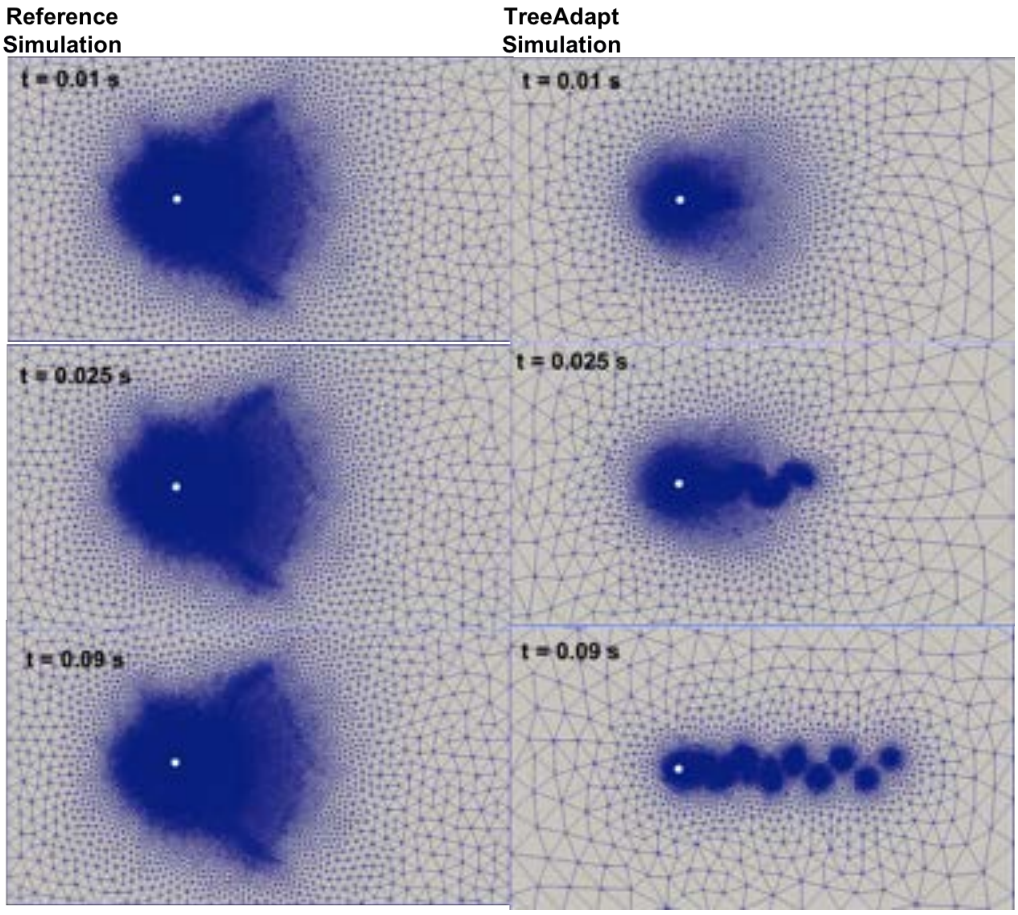
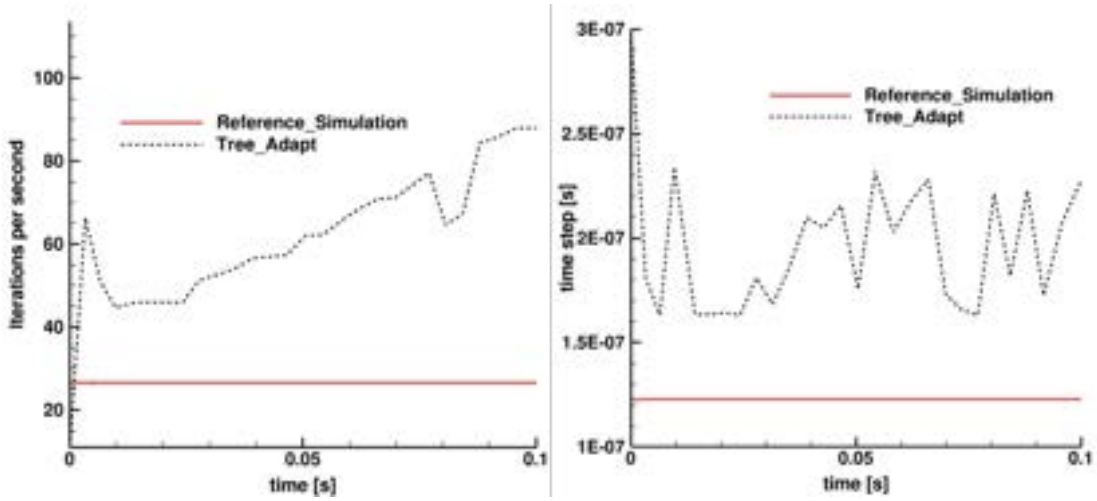


Figure 4.15: Snapshots of the mesh at the same indicated times for the reference and adapted simulations as shown in Fig. 4.14.

All above comments are confirmed by looking at the mesh evolution for both the reference and the adapted simulations at the indicated times as discussed above and shown in Fig. 4.15. Again, on the left, one observes the reference mesh which is static in time. Note that this mesh is created so that there is enough resolution downstream the cylinder to capture the shedding of the vortex and is the result of a mesh dependency study. On the right, the evolving mesh issued by the use of Treeadapt is shown. In the initial time, the mesh is seen to be adequately refined only near the cylinder and as the flow evolves the mesh quality is improved as the vorticity based sensor captures all the features of this test case.

Aside from the capacity to capture the proper local mesh resolution as a simulation proceeds, a primary interest behind the use of AMR is the gain in CPU cost of a simulation. To analyze such a feature, the iterations per second for both the reference simulation and the adapted simulation are shown in frame (a) of Fig. 4.16. The iterations per second of the reference simulation remains constant throughout the simulation as expected. For the adapted case, it stays above the reference case confirming the gain in overall cost by a factor of nearly 4 by the end of the simulation. Note that as the simulation proceeds, the number of iterations per second with AMR keeps increasing. This is due to the fact that the number of elements of the mesh decreases due to the adaptation process. As said before, the vortical structures downstream of the cylinder are however not fully captured by the sensor by the end of the simulation. As a result, vortices are dissipated as observed previously. Reducing the vorticity threshold value could help alleviating this deficiency, thereby increasing the area of adaptation and the cost of the simulation. Likewise, having a longer simulation time keeping the provided value will help attenuating this effect. It is hence clear that with AMR there is a fine balance between minimizing the cost and maintaining the accuracy of a simulation.

Another consequence of interest behind the use of mesh adaptation is shown in frame (b) of Fig. 4.16. This figure provides the time step used for both the reference and the adapted simulations. For the reference simulation, it is small and constant at around $t = 1.21e{-}7$ s. For the adapted case, it is always larger by nearly a factor of 2 and only varies slightly during the simulation confirming the adequacy of the approach. It results that the total time taken for the simulation with AMR is effectively 3.75 times faster than the reference simulation, the share of time taken by the AMR procedure in comparison to the total compute time being of the order of 12.0%.



(a) (b)

Figure 4.16: (a) Iterations per second and (b) Time step for the reference and adapted simulation using 5 processors.

To conclude, using an initially coarse mesh, one is able to obtain a reasonably accurate solution with Treeadapt and using a vorticity based sensor while obtaining performance gains.

4.5.2 Propagation of 3D planar flame

The second test case addresses the propagation of a premixed 3D planar Methane/Air flame and aims at validating the mesh adaptation tools for reacting flows. To do so, a premixed planar flame is initialized in the domain at time $t = 0$. This laminar planar flame will then advance in the domain at a known speed. In our study, a reference uniform mesh and two AMR cases are used to track this moving flamefront.

This simple case is chosen as a first step since it mimics well a realistic flame behavior. Indeed the propagation of a laminar flame usually corresponds to the first phase of an explosion test case. Another reason to use such a test case is that although the flame is unsteady it should remain planar and move at a known speed in the domain: This speed is effectively the difference between the upstream fresh gas velocity and the laminar flame speed of the reacting mixture. Finally, at any time instant the flame only occupies a small portion of the domain separating the fresh from the burnt gases. It is hence beneficial to start with a coarse initial mesh everywhere in the domain and use AMR only to track the flame and ensure that a sufficient resolution is obtained thereby resulting in performance and cost gains as well as a proper propagation of the front against the fresh stream.

Numerical Setup

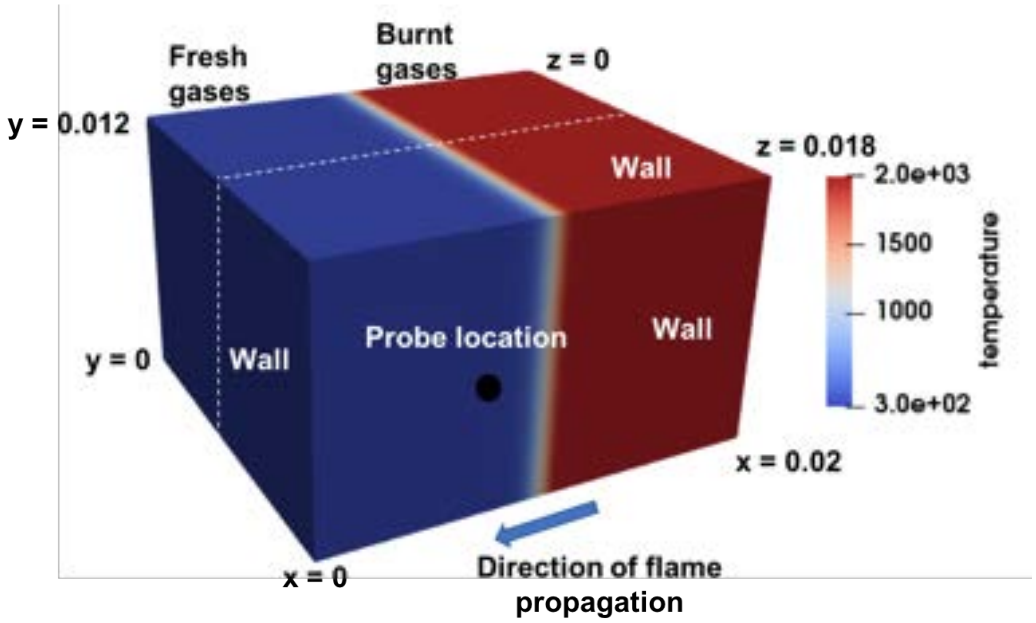


Figure 4.17: Schematic of the numerical domain for the 3D planar flame in a chamber. The probe is located at the coordinates mentioned.

Figure 4.17 shows a schematic of the computational domain used for the current study. The flame propagation direction is from right to left as indicated by the arrow. The planar flame is initialized to be in the middle of the domain at $t = 0$ as shown in Fig. 4.17. The chamber is 0.02 m in the x-direction, 0.012 m in the y-direction and 0.018 m in the z direction. Note that a probe (marked in Fig. 4.17) is added at the coordinates (0.0075 m, 0.006 m, 0.009 m), (lies on the white dashed plane marked at the center of the z-direction) to collect flow data as time proceeds. A reason for selecting this probe location is that in the mesh adaptation cases, this region lies in the area of the mesh that will undergo adaptation due to the coming flame. In terms of dynamics since it is important to ensure that flame properties are well captured, the use of the mesh adaptation process can be sampled here and results should be independent of the resolution of the method adopted.

The flame considered is a lean ($\phi = 0.8$) methane-air premixed flame. The domain is initialized with a planar flame generated from a 1D laminar CANTERA solution obtained for a fresh gas temperature T = 300 K and pressure P = 1 bar. The burnt gases are on the right hand side of the flame front and reach T_{burnt} = 2012.78 K. Fresh gases are on the left hand side of the front. The propagating flame is chosen to be in a slow flow: i.e. with $U_{freshgases} = 0.02$ m/s.

Since the laminar flame speed and laminar flame thickness are $S_L = 0.281$ m/s and $\delta_L = 0.43$ mm respectively, the flame should propagate at 0.261 m/s.

For this problem, a two step reduced BFER [155] chemistry with Pre-Exponential Adjustment (PEA) formalism is used. The PEA functions are used to recover the correct flame speeds in such mixtures [156] and is valid over a wide range of pressures (1-12 bars) and temperatures (300-700 K). It involves five species CH_4, CO_2, CO, O_2, H_2O and N_2 for two reactions.

All boundary conditions correspond to walls. Wall at the left hand side specifies relaxation on velocity, temperature and pressure set at fresh gases conditions. At the right most wall, relaxation with pressure is prescribed at atmospheric pressure. The remaining walls are all set with as symmetry.

For temporal and space integration, the Lax-Wendroff scheme with an explicit single step time integration is used [67]. Note that although the problem is by nature a DNS type simulation, the TFLES model is used and helps thickening the flame front. The value of the thickening factor F depends on the number of cells N_c specified by the user. In the following, Nc = 5 is used and no LES model is used.

A slice along the z-normal direction (marked by the dashed white line in Fig. 4.17) is shown to highlight the resolution of the initial meshes used for the various simulations. The mesh for the reference simulation as well as the initial mesh used for the adapted cases (YALESadapt and Treeadapt) are shown in Fig. 4.18. The reference mesh case which remains fixed in time and is uniform in terms of resolution contains approximately 2.1 million tetrahedral elements. The mesh resolution in the entire domain is approximately 0.25 mm. For the case using an adapted mesh, the number of elements in the initial grid is approximately $685,000$. The difference between the two meshes is clearly visible on Fig. 4.18. In the adapted case, the initial mesh resolution in the flame front is the same as the reference case, the rest of the grid being much coarser. As a result, the reference mesh has a smallest cell volume, $vol_{min} = 3.9301E-13\ m^3$ just like the adapted case for which, $vol_{min} = 3.8589E-13\ m^3$.

Quantity of Interest

In such reacting cases, it is essential to capture the flamefront as the flow evolves in the chamber. As seen in the initial meshes used in these simulations, the resolution is uniform in the case of the reference simulation. For the adapted cases, a flame based sensor should be used so that the front is always identically

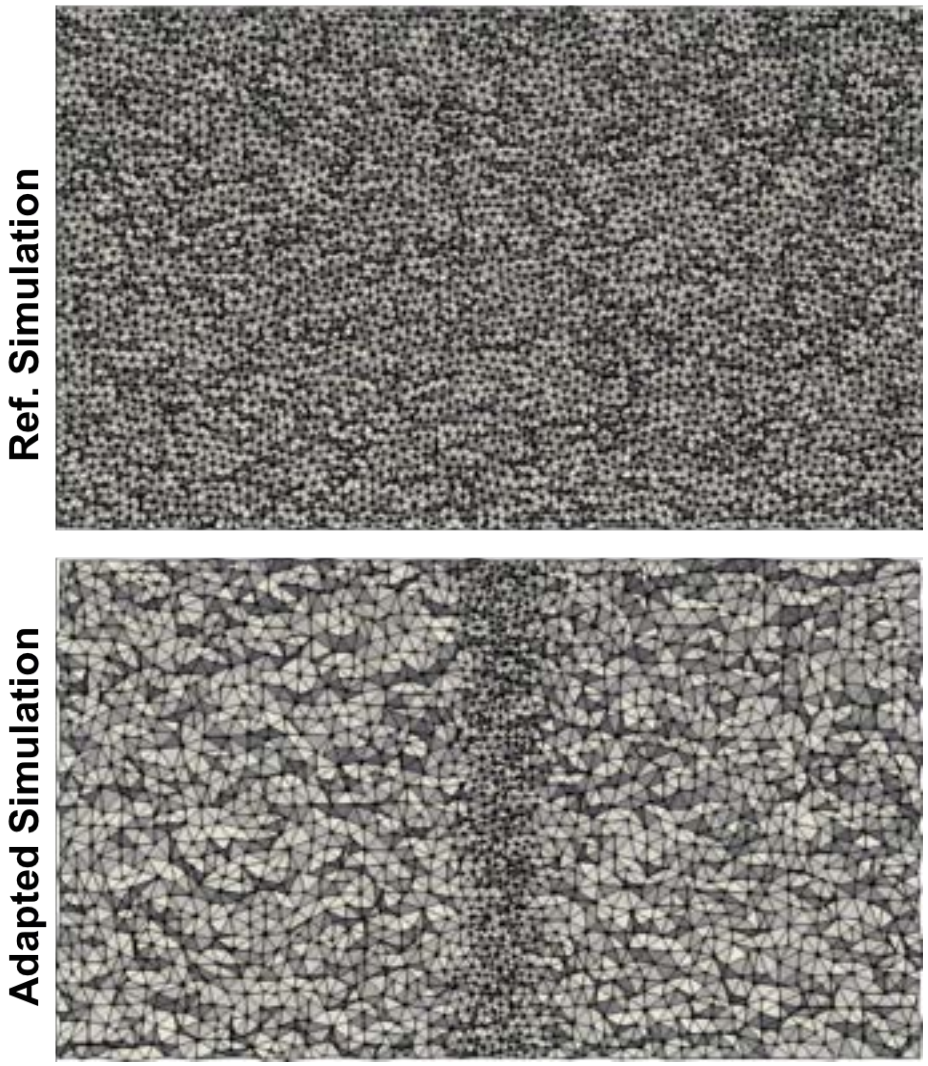


Figure 4.18: Comparison of initial meshes used in the (a) uniform reference simulation and (b) adaptation simulation using YALESadapt and Treeadapt.

resolved to capture its mask and retrieve the flame resolution of the reference case.

As a result, in the adapted simulation, the minimum edge length (0.25 mm) is chosen so that the same flame resolution is obtained as in the case of the reference simulation. Therefore, all simulations are constructed so that 5 points are present within the flame front. The resulting mask at $t = 0$ is shown along a z- normal slice in Fig. 4.19. In the region where the mask is unity, the mesh is considered as resolved and the edge length corresponds to 0.25 mm. Everywhere else, where the mask is zero, the mesh remains coarse as initially constructed for the adapted cases. Finally, the remeshing threshold, M_{thresh}, is set to 5.0%. All the other parameters such as element growth rate and target deviation are fixed to their default values.

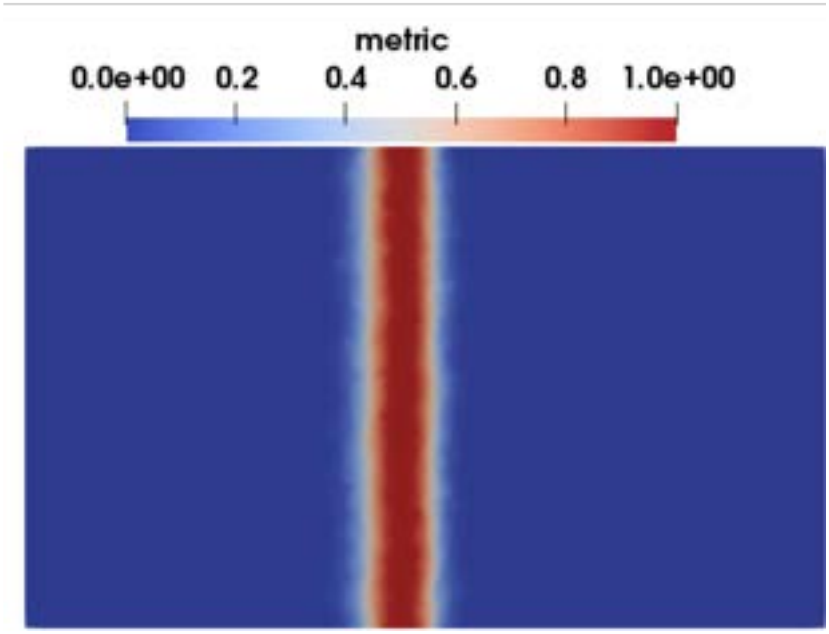


Figure 4.19: Mask based on flame sensor to track the flame front as it evolves in the chamber.

4.5.2.1 Results

To compare the flame front issued by all three simulations, snapshots of the thickening factor (to compare the flame front) fields at three different instants are shown in Fig. 4.20. Note that, in all frames, the approximate center of the flame front is marked by a dashed black line. The mesh resolution is also shown to highlight the behavior of the AMR cases for which the mesh is better resolved only in the flame front while the rest of the domain is coarse. At all times, the location of the flame and the thickening factor are identical for all simulations. As a result, no difference is noticed in the speed at which the front moves in all simulations. Qualitatively, simulations look well matched, indicating the initial accuracy of the AMR cases if compared to the reference simulation. It is also clear that the flame front is moving from the initial location in (a) in the left direction towards the fresh gases as evidenced in frame (c) at $t = 0.03$ s.

Using the location of each front the flame propagation speed is evaluated at all instants yielding for all cases approximately 0.26 m/s which agrees with the theoretical value of 0.281m/s. This confirms that quantitatively the flame front in all simulations indeed moves at the laminar flame speed throughout the simulation and is identical for all cases. However, further analysis of the behavior of all simulations are detailed next.

As this is a reacting case, it is essential to compare other properties of the simulations. Typically, the mean thickening factor as well as the mean heat

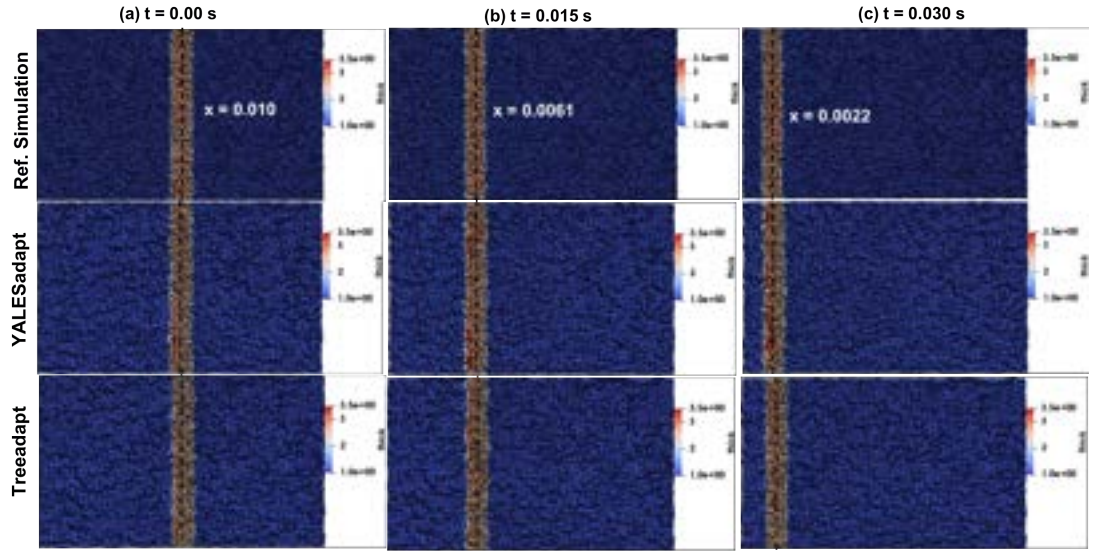
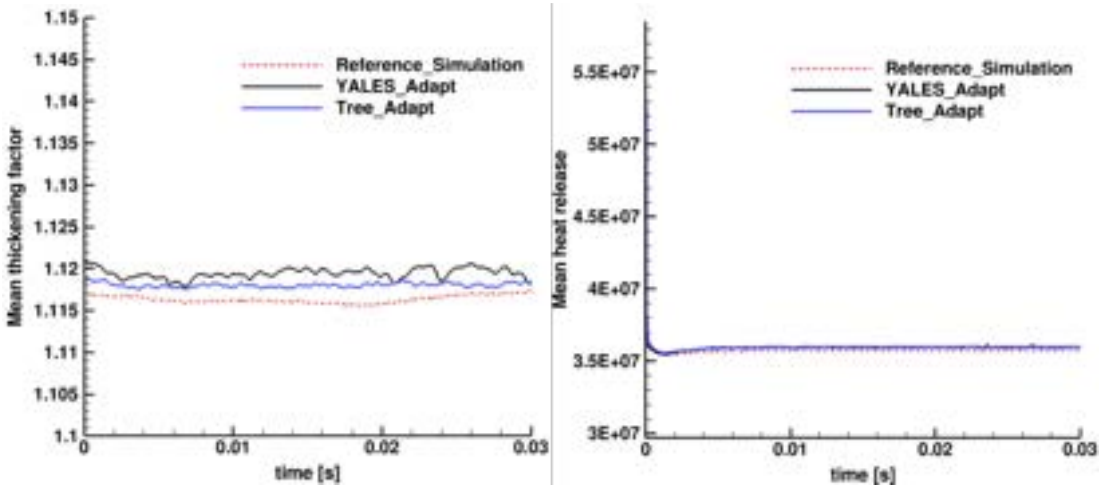


Figure 4.20: Comparison of thickening factor snapshots along with the mesh at indicated times (a) t = 0.005 s, (b) t = 0.010 s and (c) t= 0.015 s for the reference simulation and the two adapted cases (YALESadapt and Treeadapt).

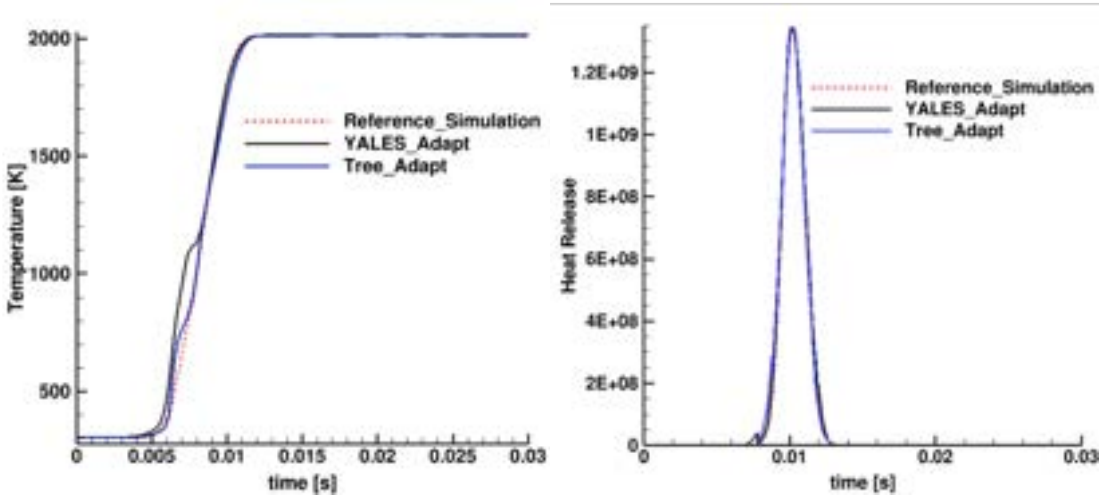
release rate are compared for all simulations in Fig. 4.21. In frame (a), the mean thickening factor in the chamber shows similar trend and values for all simulations. The mean heat release, frame (b), shows also an excellent agreement.



(a) (b)

Figure 4.21: Comparison of (a) the mean thickening factor and (b) the mean heat release in the chamber for all the simulations.

Looking at the temporal evolution at the probe location, the planar flame is seen to pass through this location going from a fresh gas state to a fully burnt state. Figure 4.22 shows the temporal evolution of temperature value at the probe and the heat release for all simulations. As the flame approaches, a rise of



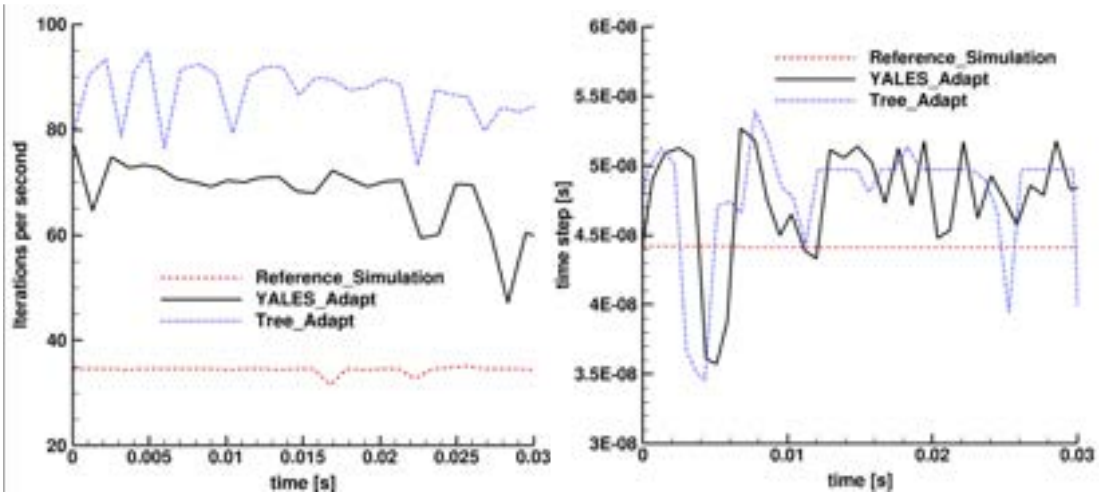
(a) (b)

Figure 4.22: Comparison of temporal recording of (a) temperature and (b)heat release at the probe location for all the simulations.

temperature is noticed at the same time instant for all simulations irrespective of the adopted method. Differences of slope of the temperature curve are noticed for both adaptation methods. This can be attributed to the small algorithm differences which impact the local adaptation process and therefore solution. These changes remain small however and can be considered acceptable. To finish, the target mesh resolution being met in the first state, all three simulations exhibit identical behaviors and lie on top of each other before settling at the same peak temperature which corresponds to the adiabatic temperature ($T_{ad} = 2012.78$K). In frame (b), the temporal evolution of heat release is compared for all simulations. Similar trends are observed with negligible differences between curves in agreement with observed temperature differences.

Ultimately, performance and cost of the AMR methods are compared next. The iterations per second for both reference and adapted simulations are shown in Fig. 4.23(a). The reference simulation as expected remains almost constant throughout. For both adapted cases, the iterations per second are significantly higher than the reference case, confirming the suitability of these approaches. Iterations per second appear greater for Treeadapt if compared to YALESadapt again due to algorithmic differences. Note that also since the flame moves in the chamber and only the flame front is tracked, iterations per second for the adapted cases should not change significantly as the simulation proceeds.

The time steps temporal evolution for the reference and the adapted simulations are shown in Fig. 4.23(b). By construction, the time step for the reference simulation is constant. For both adapted cases, the timesteps used are quite



(a) (b)

Figure 4.23: (a) Iterations per second and (b) Number of elements in the mesh for the reference and adapted simulations using 180 processors.

comparable and it is greater than the reference case. The reason for observed fluctuations of obtained time steps with adapted cases could not be clearly identified. Such behavior was however not observed in all other cases studied in the present manuscript.

Quantity compared	Reference Simulation	YALES adapt	Tree adapt
Total simulation time	27,086 s	16,262 s	16,547 s
Average Memory	123.850 MB	712.728 MB	673.861 MB
Min. Memory	114.520 MB	697.969 MB	627.391 MB
Max. Memory	142.609 MB	731.723 MB	699.000 MB

Table 4.1: Comparison of total time for simulation and memory consumed by the two adaptation techniques with the reference simulation for the 3D case of the planar premixed flame using 180 processors.

Finally, the comparison of the total start to end simulation time as well as the memory consumption for all simulations are provided on Table 4.1. The total time taken for both adapted simulations is clearly significantly reduced compared to the one needed for the reference simulation. The simulation with Treeadapt is here again similar with YALESadapt. However, the average memory consumption is smaller for Treeadapt than for YALESadapt. Note that, for YALESadapt, the share of the time taken by the AMR procedure in comparison to the total compute time is approximately 12.0%, where it is approximately 18.0% with Treeadapt. Although the AMR procedure is more involved for Treeadapt since it performs

more iterations per second, the overall total time remains almost identical for both adaptation procedures.

4.6 Conclusions and Perspectives

When solving complex large-scale flows, it is essential to have sufficient resolution to resolve the problem suitably and ensure accurate solutions. Dealing with such problems is however difficult from a numerical point of view. AMR is a mean to alleviate this challenge. Indeed, thanks to the concept of local adaptation of the mesh resolution, the fine resolution can be maintained only in regions where it is required while keeping the mesh coarser everywhere else. This is especially true for solving reacting explosion cases. Indeed these cases are ideal candidates for dynamic mesh adaptation which can be significantly be sped up due to their highly transient nature. A key in this context is the ability to construct different sensors based on the physics of interest that can then be used to adapt the mesh on the fly resulting in gains.

The steps involved in a complete parallel dynamic mesh refinement algorithm were first explained in the current chapter. The logic behind the triggering of a mesh adaptation was also detailed to understand the requirements behind such algorithms when coupled to a solver. The technical details of the adapted strategies for AVBP are then discussed and two different techniques/pipelines are described namely: YALESadapt and Treeadapt. In particular, all the key parameters such as the physics of interest, the resulting notion of masks as well as the differences between YALESadapt and Treeadapt are explained. Next, two simple test cases are selected to validate the implementations and results obtained with these two adaptation techniques. These cases are chosen to be simple although representative of non-reacting and reacting flows encountered during explosion simulations. Thanks to a 2D Karman vortex street case, a vorticity based sensor is validated. A 3D propagating planar flame is then used to validate both YALESadapt and Treeadapt using a flame sensor based on the thickened flame model. Good agreement is obtained with reference cases for adapted simulations and performance gains are confirmed. The fundamental validation steps of AMR for AVBP being obtained, the next chapter uses both adaptation techniques for more complex large-scale reacting compressible and turbulent explosion cases to investigate methodological requirements and identify if gains can indeed be confirmed for real problems of increased complexity.

Dynamic mesh refinement: Applied to explosion problems

Contents

5.1 Introduction

Large-eddy simulations (LES) are predictive as long as the large scales of a flow are properly simulated while the influence of the small scales are modeled. With the ever-increasing capabilities and resources of computing, LES is now used to simulate realistic industrial systems in complex geometries [129, 157]. In the case of transient flows and most specifically explosions/deflagrations, grid resolution and the evolution of the turbulent scales become of paramount importance. In such cases, performing LES on an adequate mesh to obtain valid LES results becomes difficult because transitions from a laminar regime to a turbulent one appear in various regions of the domain as time proceeds making the exercise very challenging. One of the most successful tools to improve the performance and validity of such large-scale numerical simulations is in this context clearly AMR.

AMR is beginning to be used in the field of computational combustion. For example, a grid sensitivity analysis has been performed in the DNS framework for a turbulent hydrogen jet flame using a structured grid [158]. Parallel AMR has also been applied to the solution of non-reacting and reacting flow cases of the Sydney bluff-body burner configuration [159]. In [160], a detonation initiated with a hot jet in a supersonic H_2-O_2-Ar mixture is investigated with the use of an

adaptive mesh refinement method. In another context, the solver YALES2 and the remeshing library MMG3D [124] have been used to perform simulations of an isothermal flow in a combustor [125]. Such tools have also emerged for primary atomization simulations [161].

The present chapter intents to demonstrate the capacities of a robust and accurate AMR methodology to maintain the accuracy of LES while reducing the solution sensitivity to the initial grid (using an efficient initial coarse grid and if used in the context of explosion simulations). Explosion problems are indeed ideal candidates for dynamic mesh refinement being fully unsteady. Masks to capture the flow dynamics need however to be estimated efficiently. If met all of these factors should result in efficient and cost effective numerical simulations giving one access to problems otherwise inaccessible with present computational capabilities.

The proposed strategy relies on a mesh refinement procedure which can also be applied to complex turbulent reacting flows. The criteria used to adapt the meshes are based on the physics of importance as discussed in the previous chapter. For demonstration, the methodology is applied to two complex reacting and turbulent test cases: (a) the Boeck setup [162] and (b) the MASRI setup (the deflagration in a large venting chamber [163, 164]). In both cases, massive parallelization is needed to obtain performance while maintaining computational accuracy and ensuring good quality results in comparison to the experimental data.

The chapter is organized as follows. The first test case (Case A), which corresponds to the Boeck setup is explained in Section 5.2.1. First, the experimental setup, the computational domain and the numerical setup are detailed. A 2D computational version of the Boeck case is then used to tune the mesh adaptation parameters so these parameters can then be used to simulate the 3D real setup. After validation, further analysis is performed to ensure accuracy of results obtained with AMR. The gains in performance and cost are also highlighted. The second test case (Case B) corresponds to the MASRI setup and follows the same presentation logic as the Boeck setup. Finally, conclusions and perspectives are provided in Section 5.3.

5.2 Description of the test Cases

Throughout the following discussion, the mesh adaptation techniques introduced in the previous chapter are applied to two complex turbulent reacting cases. First, the Boeck case is chosen as it is slightly more complex than the previously

studied 3D planar flame case. However, in terms of explosion test cases it is still considered a relatively simple test case. In this case, although the flame front remains laminar throughout, it only undergoes stretching and the flame evolves from a smooth and laminar stage to a progressively wrinkled flame front. Second, the MASRI case is chosen because it is more relevant to a complex explosion test case. In that case, the flame indeed undergoes all the stages experienced during an explosion, starting from a laminar flame eventually transitioning to a fully turbulent flame.

5.2.1 Case A: Boeck Setup

The Boeck test case [162] is chosen for two reasons: first, this particular configuration allows one to address the behavior of the flame during a deflagration in a channel with a simple central obstacle. Second, it is closer to an industrial application and more realistic than the 3D planar flame done earlier. Note that, the ultimate objective of this test is to demonstrate that accurate results can be obtained beginning with a coarser initial mesh provided that the AMR methodology is properly implemented. Another reason why this test case is useful for AMR is that the 2D and its real 3D versions are not too dissimilar when it comes to flow dynamics and evolution. Indeed, as mentioned before, the mesh adaptation parameters are case dependent and need to be properly tuned to be used efficiently for any simulation. The simpler 2D setup can therefore be used to tune the parameters which can then be used to validate the full 3D Boeck AMR prediction.

Experimental Setup

The experiments are performed for a closed channel as shown in Fig. 5.1. The square cross-section and length of the channel are 0.0762 m X 0.0762 m and 0.4953 m, respectively. A simple obstacle is located at $x = 0.1524$ m. This obstacle consists of an upper and lower wall mounted fence-type obstacle, of height 0.0191m, thickness 0.0127 m and extending across the entire spanwise dimension of the channel. Experimentally, mixtures are prepared in a separate vessel using the method of partial pressures and used to fill the channel. For ignition, a spark plug mounted on the channel closed end is placed at mid-height ($x = 0$ m) and is used to initiate the mixture combustion. More information regarding the experimental setup can be found in [162].

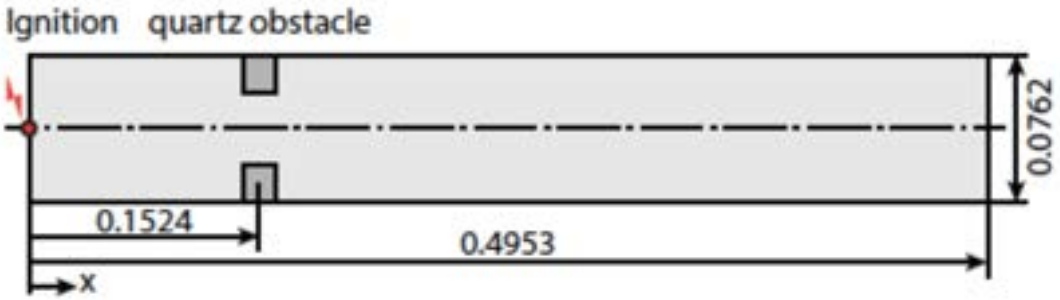


Figure 5.1: Schematic of the experimental setup, side view. All dimensions are marked in meter [162].

Numerical Setup

Figure 5.2 shows a schematic of the 3D computational domain used for the current study. The x, y and the z axes are oriented in the streamwise, wall normal and transverse directions to the bottom wall of the channel respectively. The venting direction is marked with a solid red arrow. The dimensions of the computational domain are chosen to be identical to the experimental configuration shown in Fig. 5.1. A zoomed view of the initial mesh for the region enclosed by the dashed line of Fig. 5.2 is shown in Fig. 5.3. At initial time, t = 0, the flame is ignited as in the experimental setup.

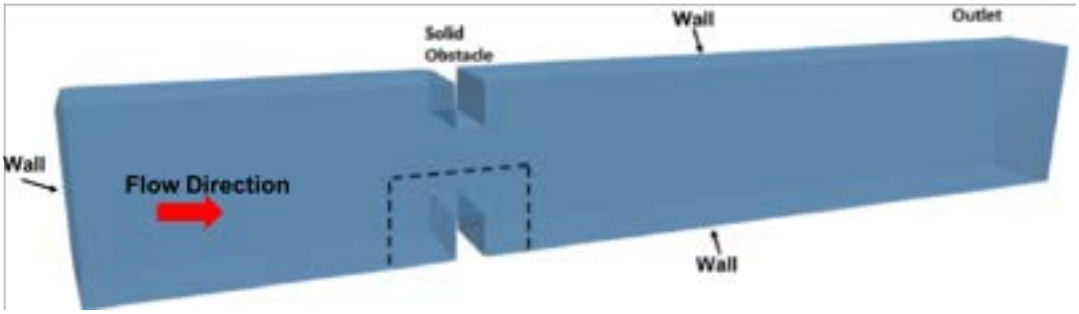


Figure 5.2: Computational domain used for the present Study.

Figure 5.3 depicts a Z-normal sliced view of the numerical grid used for the static reference mesh case. In this case, a uniform well resolved mesh is used to produce results to be compared to the experimental findings. To do so, a high quality unstructured grid using tetrahedral cells is generated. The mesh has a total number of cells of approximately 37 million. This specific mesh was chosen after performing various simulations with different meshes of variable resolutions. The mesh shown and discussed here ensured the accuracy of the results as well as reasonable costs which are achievable with available computational resources. The minimum volume of the cell in this reference mesh is $vol_{min} = 0.4537E -$

$11m^3$. Another advantage of having access to such a reference simulation is that it provides data to be compared to results with the AMR.

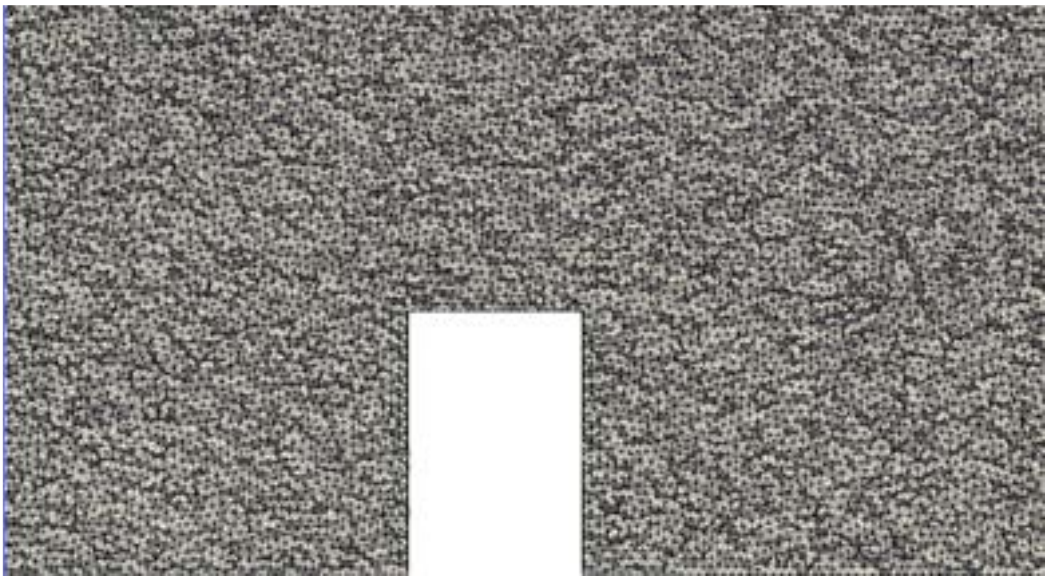


Figure 5.3: Computational grid: Z-normal slice at the center line for (a) initial 3D reference mesh.

The initial conditions being known from the experimental setup [162], the same conditions are applied to the numerical simulations at $t = 0$: i.e. no initial velocity in the entire domain and the case is ignited using a premixed spherical flame at the right wall of Fig. 5.2 . The mixture used consists of H_2/O_2 at stoichiometry. Temperature and pressure of 300K, 15 kPa as in the experiment and an exit pressure NSCBC condition [165] is used at the outlet section.

Note that, modeling the ignition phase can be complex and is beyond the scope of the present study. In our case, the deflagration is initialized with a spherical flame. The radius of the flame kernel is 8 mm and the center is located at $(0.0m, 0.0381m, 0.0m)$. Volpiani et al., [166]) has indeed shown that for relatively low values (typically lower than 10 mm), the initial flame radius does not impact the flame propagation characteristics. The internal structure of the initial flame is that of an artificially thickened laminar 1D flame at the operating conditions (the thickening being chosen so that 5 cells are present inside the flame thermal thickness).

Several studies [128, 167, 168] have shown the capacity of single-step mechanisms to correctly predict the main features of a flame acceleration as soon as key ingredients are reproduced. These chemistries are usually fitted against the mixture laminar flame speed s_L and the adiabatic burnt gas temperature T_{ad} at the initial conditions of the explosion. Based on this common method, for the LES of

the Boeck setup presented in this chapter a single-step approach is adopted. The chemical scheme used involves three species (H_2, O_2 and H_2O) and one reaction, the laminar flame speed and laminar flame thickness being $S_L = 7.32$ m/s and $\delta_L = 2.30$ mm respectively.

Since the channel is mainly closed, most boundary conditions correspond to walls for simulation considerations. Since search for efficiency is paramount here, the mesh resolution employed does not allow to consider a wall resolved LES foundation (high y^+). Instead a law-of-the-wall (Schmidt et al. [169]) approach is preferred and adiabatic slip wall conditions are applied.

When it comes to the numerics, a centered continuous Taylor-Galerkin scheme that is third-order in space and time [68] is used. The DTFLES model, described in the previous chapter helps thickening the flame front. The value of the thickening factor F_{max} in the flame front region is set to depend on the number of cells N_c specified by the user to remain inside the thermal thickness of the thickened flame. In the following, Nc = 5 is used. An efficiency function E must also be introduced to retrieve the flame wrinkling effect. The model of Charlette et al. [131] is used with $\beta = 0.5$. Finally, the small scale turbulence effects is modeled using dynamic Smagorinsky model [170].

Comparison of the 3D reference simulation with the experiment

Figure 5.4 shows OH-PLIF images and the predicted temperature fields obtained for the reference case. The experimental sequence is reconstructed from eight different experiments with a difference in trigger delay time between images. The description of the flame evolution is divided into two phases according to the observed flame topology of the OH-PLIF images.

(a) frames 1-4 : the flame propagates upstream and past the obstacle showing a smooth and laminar flame surface
(b) frames 5-8 : flame front locates further downstream of the obstacle and, transforms into a progressively wrinkled flame front.

In (a) (frames 1-4), the absence of perturbations in the images suggests that no significant turbulent motion is present immediately downstream of the flame front. Also one can note that, the flame tip starts to stretch as it approaches the central obstacle as seen in frame 2. The flame finally elongates as it passes through the obstacle opening (frames 3 and 4). Excellent match is observed between the experimental images and the numerical simulation for these steps. The flame evolution as well as the flamefront speed are almost identical for both the experiment and the numerical simulation.

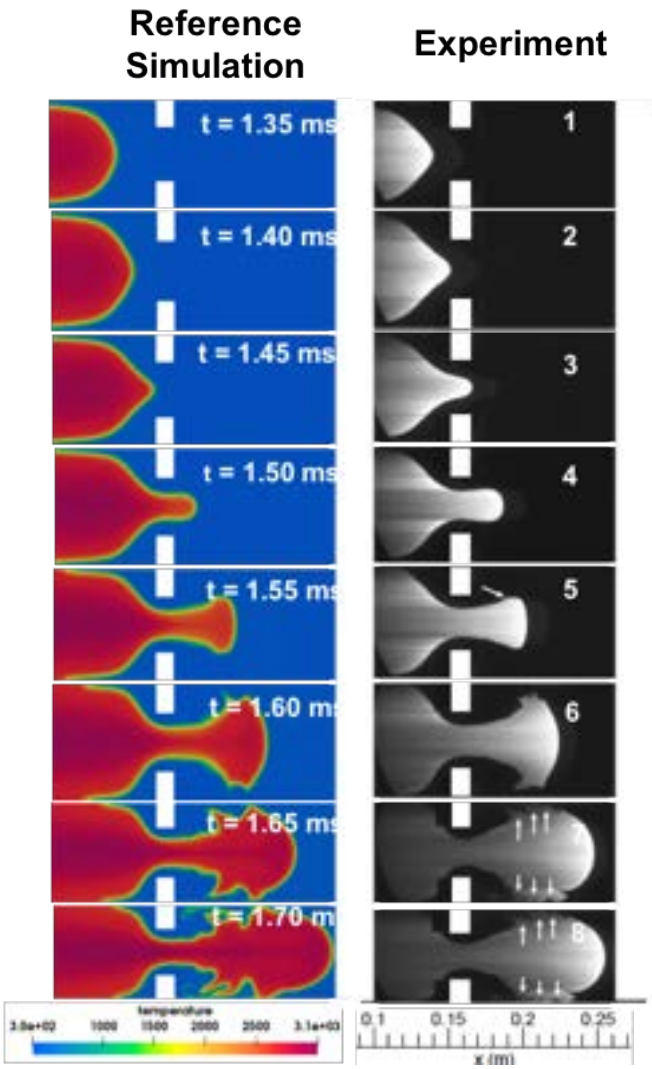


Figure 5.4: Comparison of Reference simulation with the experimental OH-PLIF images [162] for the 3D Boeck setup.

In zone (b), frame 5 of the OH-PLIF sequence suggests first signs of flame interaction with vortices shed from the obstacle edges. This results in a local deformation of the flame (see arrow in frame 5). Flame perturbations in the OH-PLIF images can be unambiguously attributed to this interaction and the effect of the vortices. This vortex-flame interaction is dictated by the convection of the vortices as well as the lateral expansion of the flame tip as observed in frame 5. Frame 6 coincides with the moment where the flame enters and interacts with the recirculation zones created downstream of the obstacle. The convection of the flame into the recirculation zone is here asymmetric, with it being larger near the bottom wall. In frames 7 and 8, small-scale wrinkling is observed as the flame approaches the upper and lower channel walls as marked by the white arrows. Note that even in this phase (b), the flame tip remains smooth and laminar at

all times. The numerical simulation follows the experimental images throughout the evolution in the chamber, both in phase (a) and (b),

It is clearly evident that the static reference simulation is successfully able to capture the flow dynamics of the Boeck setup. However, the goal of using AMR for such an unsteady setup is to obtain similarly accurate results but at a cheaper computation cost. This is attempted next for the 3D Boeck setup. To do so, the two adaptation procedures and tools detailed in the previous chapter are used. As mentioned before, one of the hurdles with these techniques is that at this stage, the parameters are case dependent and need to be tuned for each case. Since, it is not economically feasible to tune the parameters for the full 3D setup directly, an AMR calibration process is first obtained with the 2D numerical setup of Boeck. One of the future goals of this work would be to remove this calibration constraint.

AMR calibration process

As said before, a simpler 2D setup of the Boeck configuration is used for this process. It indeed allows one to get an idea of the parameters that need to be used for the full 3D validation without much CPU effort. For this specific calibration process, only YALESadapt is used.

To check the accuracy of the 2D AMR simulations, a highly refined uniform (reference) simulation is performed first. This reference simulation was found to be sufficient since a simulation with a further refined mesh although not discussed here showed no difference in results. The reference simulation corresponds to a 2D mesh with 1.5 million triangular elements. Based on these results, adaptation simulations are chosen so that the number of elements in the initial mesh are significantly less than this 2D reference mesh. Then using the appropriate adaptation parameters, the mesh should adapt on its own to obtain accurate results. The initial mesh used for the 2D AMR YALESadapt simulation contains $350,000$ triangular elements. Note that initial conditions, boundary conditions, numerical models and chemistry mentioned above are used for all 2D simulations. It was found that for both the explosion cases, the parameters tuned in the 2D setup could be applied to get correct results for the 3D setup.

The Boeck problem deals with fluid dynamics and combustion, it thus requires the combination of a vorticity-based sensor as well as a flame sensor for AMR cases. A threshold value for vorticity and minimum acceptable edge length-(corresponding to the same flame resolution as the reference simulation ($F = 1.1$) introduced before) is imposed. For the 2D simulation, due to lack of experimental data, the resolution of the reference case is chosen so that the thickening factor

is close to unity. A mask is then calculated based on these thresholds and are used during the AMR simulations as explained in the previous chapter. Note that for this case, when the flame is approaching the central obstacle, a large portion of the combustion chamber upstream of the central obstacle is expected to be occupied by the evolving flame. It is hence essential to use the flame-based sensor that tracks this evolution. In the regions downstream of the obstacle, along the walls and the edges of the obstacle, vortical structures will form. These also need to be captured accurately by the simulation. Recall that a remeshing threshold is also used to determine how often mesh adaptation is performed and the appropriate value (5%) used here was obtained through numerical tests. The minimum edge length used for the flame sensor is fixed at 1.5 x 10^{-4} m so the flame local mesh resolution is always equivalent to the one prescribed for the uniform reference case. In the adaptation algorithms used in this thesis, no coarsening is applied. If the mask is zero, the mesh remains unchanged at the initial mesh resolution. All these key parameters are listed in Table 5.1, the comparison between the adapted simulation (performed via YALESadapt) and the 2D reference simulation are shown next.

Keyword for adaptation	Value
Min. edge size	1.5 x $10^{-4}m$
Remeshing metric threshold	0.05
Use vorticity metric	Yes
Use dtfles sensor metric	Yes
Vorticity threshold	6 x$10^{3}s^{-1}$

Table 5.1: List of important keywords for mesh adaptation in the 2D Boeck configuration.

Snapshots of the temperature fields at indicated times are compared for the 2D reference and 2D YALESadapt simulations in Fig. 5.5. From the figure, it is evident that the propagation speed of the front is only marginally different in both simulations. Differences are only noted when the flame front is approaching the central obstacle. Apart from this subtle difference, all phases of the flame evolution for this Boeck setup appear to be captured adequately using the 2D assumption.

As shown on Fig. 5.5, a probe is selected to compare the results more quantitatively in a region just downstream of the central obstacle and along the center line of the chamber. This is an interesting zone as here the flame accelerates and undergoes perturbation due to flame-vortex interactions as illustrated by Fig. 5.5. This region is also selected since it will undergo mesh adaptation as the flame

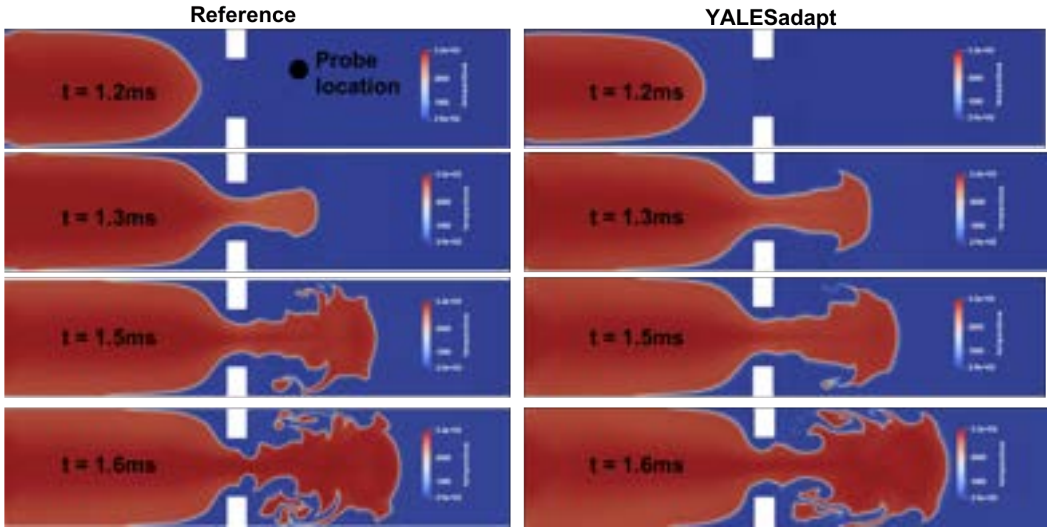
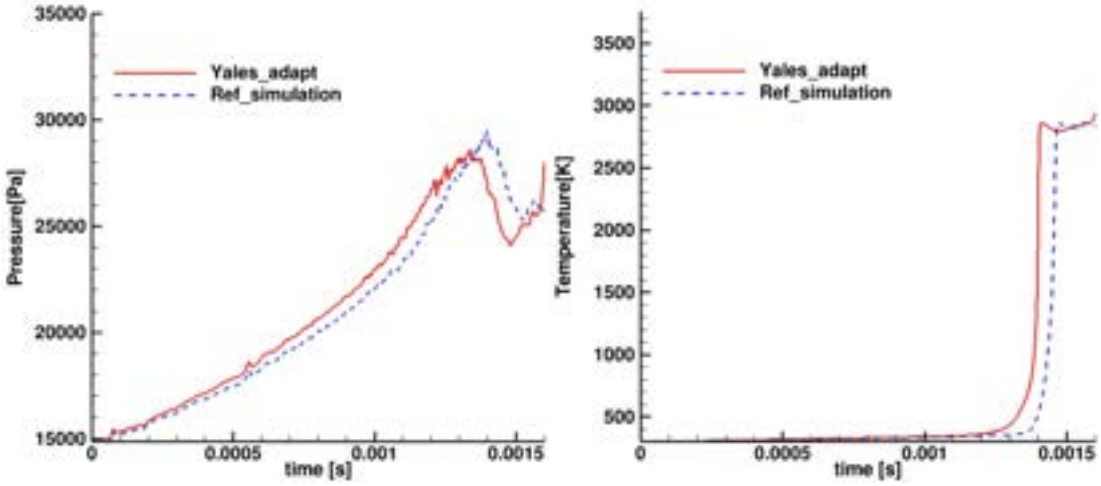


Figure 5.5: Comparison of temperature field for the 2D Boeck simulations performed for the reference and YALESadapt meshes.



(a) (b)

Figure 5.6: Comparison of (a) pressure and (b) temperature at the probe location for the 2D Boeck setup for the Reference simulation with the YALESadapt simulation.

evolves through the chamber. It is hence a quantitative way of ensuring that no dynamics of the flow is lost because of the mesh adaptation process.

The comparison of the results at the indicated probe is shown via the temperature and pressure signals registered in time, Fig. 5.6. The peak and trends of both quantities are consistent in both simulations. There is however a slight difference in speed with which the flame front travels. These specific values of the adaptation parameters shows good match between the simulations. Other values do not provide satisfactory result. From all test cases conducted and not detailed

here, one can conclude that:

(a) if the flame resolution (min. edge length) is not sufficient, the dynamics of the flame is not captured accurately resulting in a slightly dispersed flame front. (b) if the vorticity threshold is not sufficient, the pressure signal produced within the simulation is always underestimated at various probe locations.// The values of Table 5.1 will thus be used for both the YALESadapt and the Treeadapt 3D simulations.

Comparison of the 3D adaptation simulations with the experiment

As mentioned before, the 3D reference mesh chosen for the Boeck setup contains 37 million tetrahedral elements as it was most economical way to address this problem. This mesh corresponds to a flame resolution which coincides to a thickening factor, $F = 3.0$. Effectively, It is not cost effective to use a 3D mesh with a thickening factor close to unity as in 2D. The minimum edge length has therefore to be adjusted (5.5 x $10^{-4}m$) to match the flame resolution of the 3D reference case. All other parameters obtained from the calibration process are retained and used for the 3D adaptation simulations as listed in Table 5.2. Note however that for the vorticity threshold, since the threshold value is obtained from the 2D setup it might not be applicable to be the full 3D setup (as the cell characteristic lengths are not the same and this vector involves much more directions). It needs to be therefore addressed carefully. Despite this in this case, the results obtained with the same vorticity threshold were found to be satisfactory.

Keyword for adaptation	Value
Min. edge size	5.5 x $10^{-4}m$
Remeshing metric threshold	0.05
Use vorticity metric	Yes
Use dtfles sensor metric	Yes
Vorticity threshold	6 x $10^{3}s^{-1}$

Table 5.2: List of important keywords for mesh adaptation in the 3D Boeck configuration.

One key of the adaptation algorithm as built here relies on the initial mesh resolution. Since coarsening is not taken into account, the coarsest region will remain untouched and the minimum cell number achievable is thus set by this initial grid. In the following, an initial mesh is chosen for the 3D adaptation simulations to derive from the reference 3D mesh. It contains approximately 9.5 million tetrahedral cell elements. This was chosen as the initial mesh as it contains

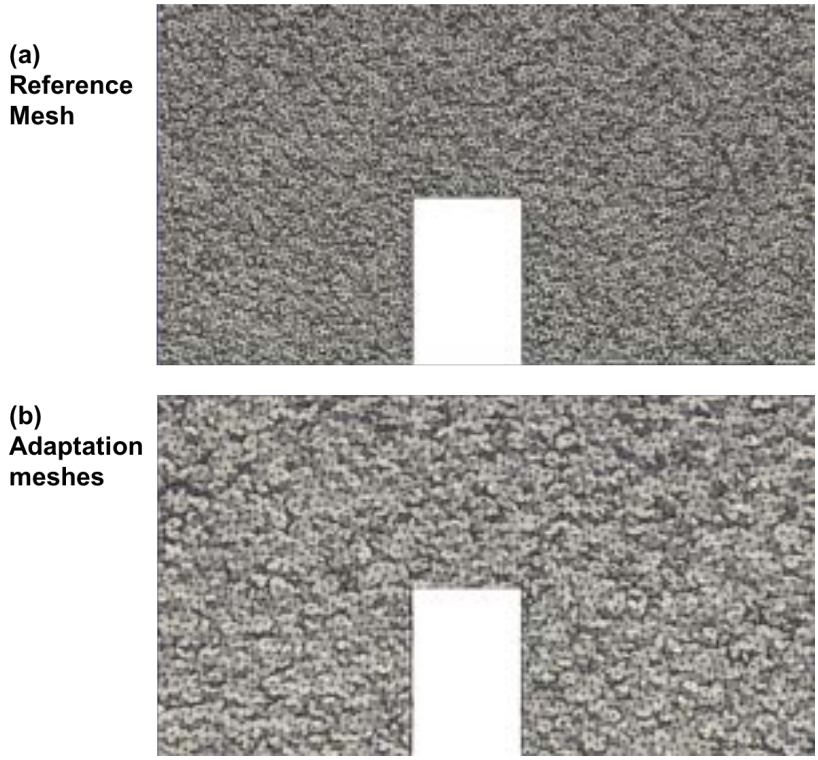


Figure 5.7: Computational grid: Z-normal slice at the center line for (a) initial 3D reference mesh and (b) initial mesh used for the 3D AMR simulations.

approximately a quarter of the number of elements compared to the reference case. The two meshes are compared with each other in Fig. 5.7. In frame (b), visually it is quite evident that the initial mesh used for the adaptation runs is much coarser. It is indeed expected that as the simulation proceeds, the mesh will adapt based on the sensor and the number of elements in the mesh will adjust itself. The minimum volume in the mesh is of the order of $vol_{min} = 0.8511E - 11m^3$ compared to $vol_{min} = 0.4537E - 11m^3$ for the reference case. The smallest initial element for the AMR cases has approximately twice the volume of the smallest element of the finer 3D reference mesh.

The LES results of the 3D Boeck AMR cases are compared with the experimental images available in Fig. 5.8. All the results displayed here to compare the two AMR methodologies with the OH-PLIF images. Similarly to the initial discussion about this test case, frames 1-4 correspond to the smooth and laminar part of the flame dynamics. In this part of the sequence, flame evolution is captured identically by both adaptation techniques and show excellent match with the experiment just like the reference 3D prediction.

Once the flame has passed the central obstacle frames 5-8, it is perturbed due to interactions with the developing flow eventually leading to an interaction of

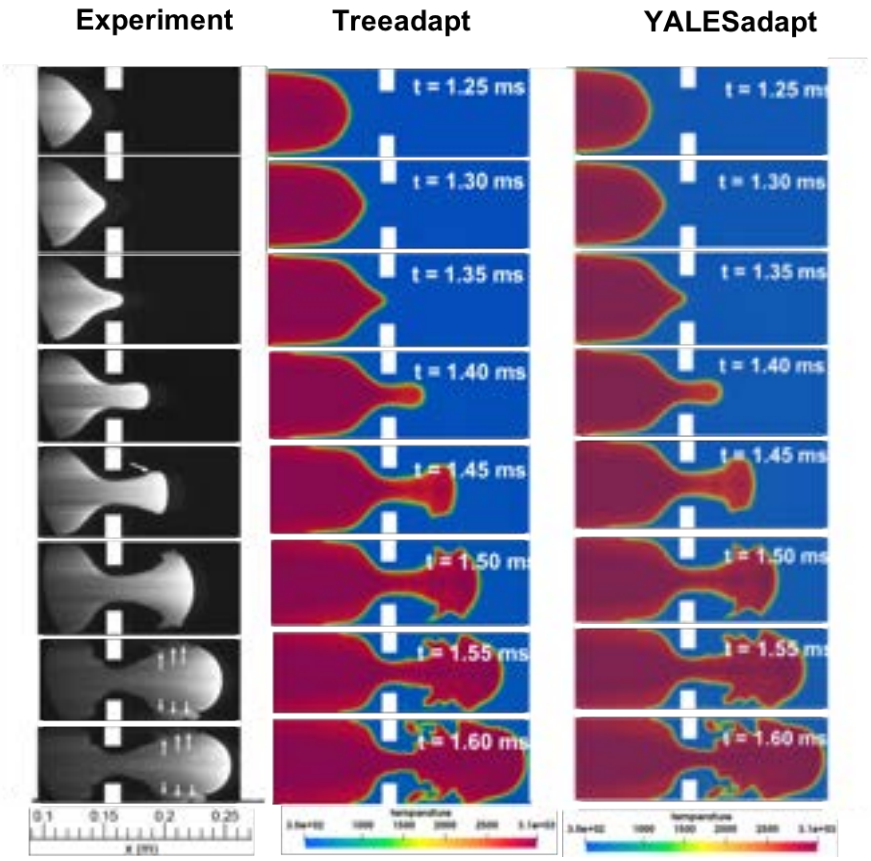


Figure 5.8: Comparison of Treeadapt and YALESadapt simulations with the experimental OH-PLIF images [162] for the 3D Boeck setup.

parts of the flame with the generated recirculation zones. Small scale wrinkling of the flame is also observed at this occasion and subsequent stages. All these phenomena (frames 5-8) are clearly captured accurately by both adaptation techniques. There is only marginal differences due to the flame front speed recovered with adaptation techniques used.

To conclude, the comparison of the evolution of the flame obtained by LES with the experiments confirms that the proposed numerical simulations, especially AMR, are all able to capture the dynamics of the chosen problem qualitatively.

Results and discussion

A more thorough and quantitative comparison is done for the reference, YALESadapt and Treeadapt simulations in the following. Focus is first put on the overall quantitative dynamics of the simulations. Then, the local response of the flow is compared in an important region of the problem via data collected at a probe location during the simulations. Finally, diagnostics dealing with performance and cost associated with these simulations are compared.

Accessing quantitative measures of explosions is a rather complex task. For closed vessels, the resulting increase in pressure within the chamber can be linked to local flow behaviors by integrating in space and over the whole combustion chamber the governing equations [171]. Doing so, the increase in pressure inside the chamber can be considered as the result between two competitive phenomena: the combustion rate and the venting rate. Indeed, the burnt gases inside the chamber have a lower density than the fresh gases which they replace and hence, take up a bigger volume, inducing a pressure increase in the chamber. At the same time, while expanding gases are pushed out through the open end of the vessel by the generated flow. Thanks to a volume balance, the mechanism by which the pressure inside the chamber increases is highlighted by directly comparing these two terms. Both quantities are effectively very important as they calibrate the strength of the explosion and provide a indication of accuracy of simulations.

For the combustion rate, denoting the fresh gas density by ρ_f and the burned gases density by ρ_b. Thanks to mass conservation, the fresh gas volume V_f and the burnt gas volume V_b can be linked through $V_f = \frac{\rho_b}{\rho_f * V_b}$. The combustion rate then can be written as the variation of the burnt gas volume minus the variation of the consumed fresh gases, so that,

$$\dot{V}_{comb} = \frac{\partial V_b}{\partial t} - \frac{\partial V_f}{\partial t} = \frac{\partial V_b}{\partial t}(1 - \frac{\rho_b}{\rho_f}). \tag{5.1}$$

Note also that the variation of the burnt gas volume can be linked to the combustion process or any rate of production or consumption of combustion product. For example, considering H_2O, one has,

$$\frac{\partial V_b}{\partial t} = \frac{\dot{\omega}_{H_2O}}{\rho_b Y^b_{H_2O}}, \tag{5.2}$$

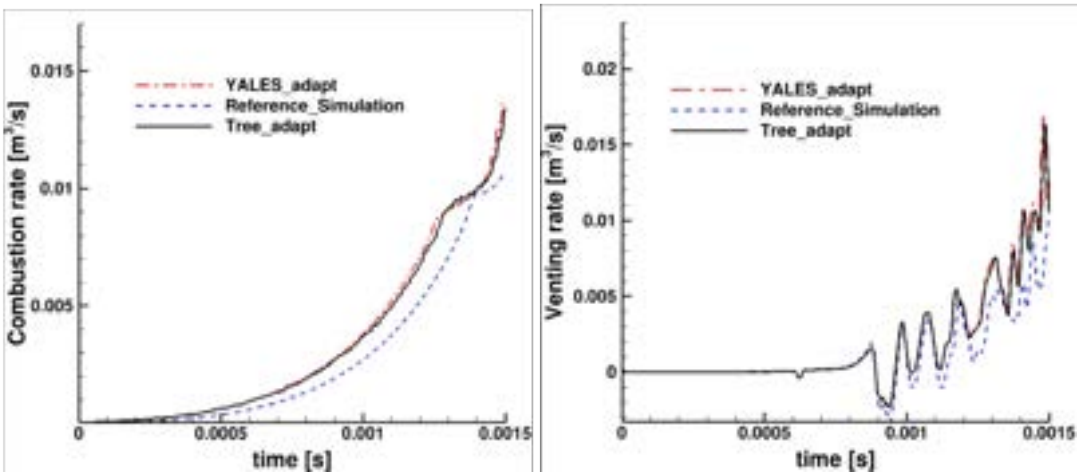
where, $Y^b_{H_2O}$ is the H_2O mass fraction in the burnt gases and $\dot{\omega}_{H_2O}$ is the rate of production of species H_2O integrated over the entire chamber.

For the Venting rate noted $\dot{V}_{vent}$, it is defined as the volume of flow which exhausts from the chamber. That is,

$$\dot{V}_{vent} = \int_{S_{outlet}} U_n dS, \tag{5.3}$$

where, S_{outlet} is the sectional area of the chamber outlet and U_n is the normal velocity at this section.

To evaluate the pressure temporal evolution in the chamber, the evolution of the rate of combustion and the venting can be compared in time as shown in Fig. 5.9. First, the comparison of the combustion rate issued by all simulations



(a) (b)

Figure 5.9: Comparison of (a) combustion rates and (b) venting rates between Reference, YALESadapt and Treeadapt simulations for the 3D Boeck setup.

in Fig. 5.9 (a), there is only a slight difference between the reference curve and the AMR predictions which can be linked to the differences in speed at which the flame fronts evolve. However, peaks and trends are similar for all simulations. Comparing two AMR simulations, the curves show excellent agreement almost lying on top of each other. Similar trends are observed for the venting rate comparisons and shown in Fig. 5.9 (b).

In terms of contributions, one can decompose the total combustion rate of simulation into a resolved ω_{res} and a sub-grid-scale part ω_{sgs}. Based on such contributions which report the effect of the resolved field (or grid resolution) to the model response, the following ration can be investigated.

$$\eta = \frac{\dot{\omega}_{res}}{\dot{\omega}_{tot}} = 1 - \frac{\dot{\omega}_{sgs}}{\dot{\omega}_{tot}} \tag{5.4}$$

Figure 5.10 shows the evolution of η as a function of time when the flame interacts with the central obstacle. That is when a challenging balance between grid resolution and modeling appears. As anticipated, in areas where the flame is almost laminar (before the central obstacle), the flame wrinkling is fully resolved on the grid and the contribution of the sub-grid-scale combustion model is nearly zero percent. On the other hand, when the flame starts interacting with the solid obstacle, typically downstream of the central obstacle, the resolved part of the combustion rate should be lower due to the use of the turbulence model that should be activated. As evidenced in Fig. 5.10, the two adaptation techniques

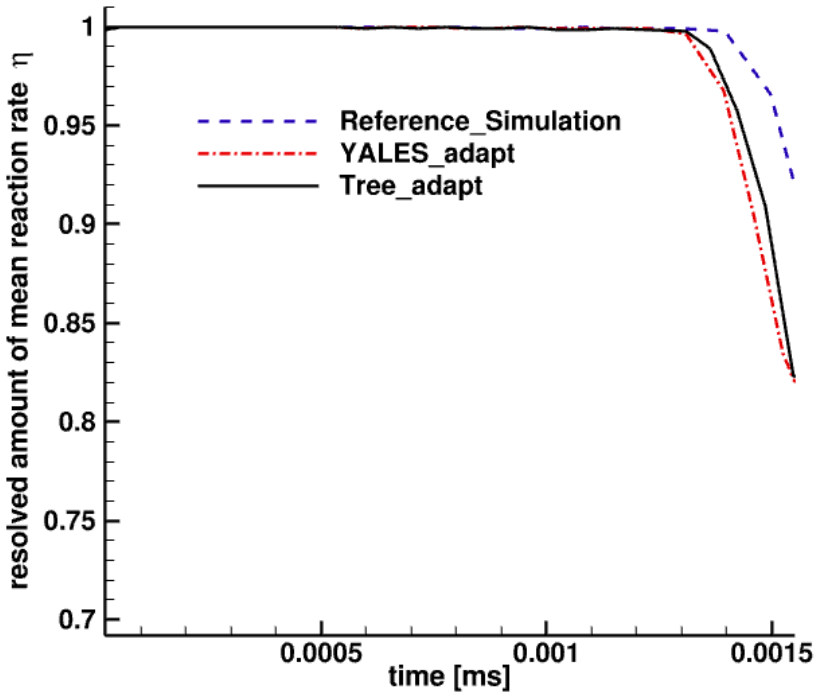


Figure 5.10: Resolved amount of mean reaction rate η for the adapted cases and the reference simulations for the 3D Boeck setup.

(YALESadapt and Treeadapt) almost lie on top of each other. The reference simulation shows a similar trend, only slightly shifted in time.

A more local diagnostic can be obtained by comparing the signal registered at the probe location discussed for the 2D simulations and also present in the 3D predictions. Figure 5.11 shows such a comparison for the pressure signals retrieved by the Reference, the YALESadapt and the Treeadapt simulations. Comparing the reference with the two AMR simulations, a similar peak in pressure at the probe location is reached. In the reference simulation, since the flame front moves slower by about 0.1 ms, as was observed previously, the peak appears later in time if compared to AMR results. Comparing the two adaptation simulations, the simulation starting with the same initial mesh and using the same adaptation parameters, the observed coincidence between the two simulations confirms the consistency between the two tools as observed in Chapter 3.

Similarly, Fig. 5.12 shows the comparison of the temperature signals for the Reference, the YALESadapt and the Treeadapt simulations. Comparing the AMR to the reference simulations, both AMR predictions reach a similar peak value of temperature following similar trends in time. Comparing YALESadapt and Treeadapt evolutions, a small difference in the flame front speed is evidenced, the front passing by the probe at slightly different time instants.

All above comparisons either qualitative, quantitative as well as global and

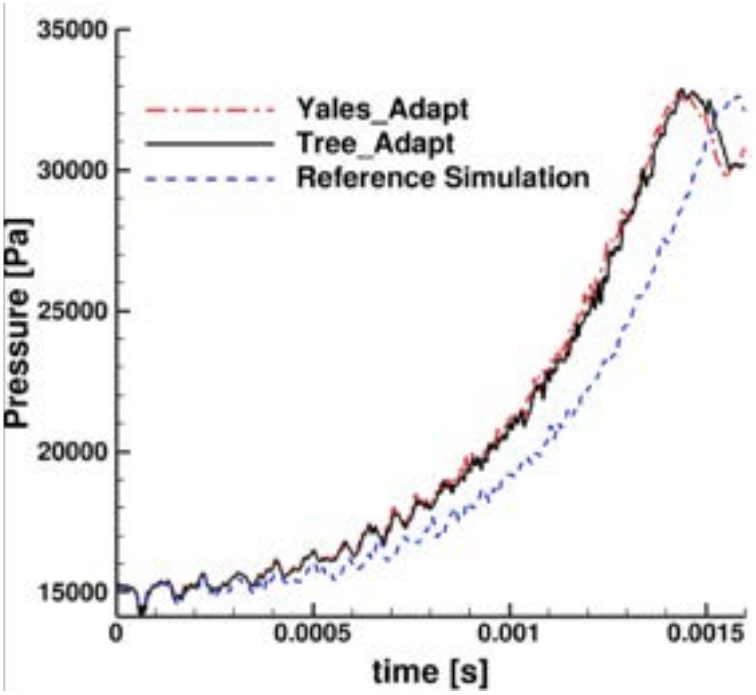


Figure 5.11: Comparison of evolution of pressure probe signal between Reference, YALESadapt and Treeadapt simulations for the 3D Boeck setup.

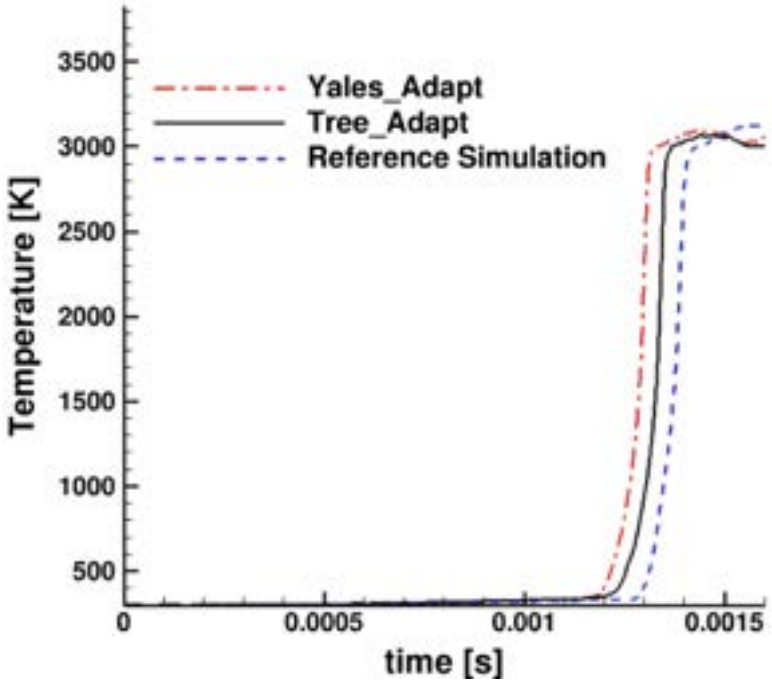


Figure 5.12: Comparison of evolution of temperature probe signal between Reference, YALESadapt and Treeadapt simulations for the 3D Boeck setup.

local indicate that both AMR algorithms implemented successfully meet their purpose for this setup. However, it is imperative that performance and cost associated with these simulations are also meeting expectations. Indeed, apart from the advantage of locally refining the mesh as the flow evolves in time using an appropriate mask based on important physics, a main reason for applying AMR is to obtain gains in terms of CPU cost. It is therefore imperative to look at the performance of the computations i.e. the number of elements in the mesh, the number of iterations performed per second and the timestep used.

As already discussed above, the mesh adaptation technique depends on the initial mesh chosen. As a result an initially coarse mesh is chosen compared to the reference mesh. One element that contributes to the performance evolution of an AMR is clearly the number of elements that are present in a mesh at a given time instant. Figure 5.13 shows that number of elements is changing with time for the two AMR cases. Both the simulations start with an initial mesh of approximately 9.5 million tetrahedral elements. Despite this same starting point, it is evident from the figure that the number of elements in the mesh is slightly greater for the YALESadapt case throughout the simulation. This is due to the additional check for skewness which is performed in YALESadapt as explained previously. As the simulation proceeds, when the flame front occupies the majority of the combustion chamber, the number of elements of both the methods become similar as expected.

The iterations per second performed by the solver depends on the number of elements in the mesh at a time instant. Figure 5.14 shows the retrieved iterations per second for the reference simulation and the AMR simulations. Comparing the reference to the adapted simulations, it is clearly advantageous to use AMR in comparison to a static approach. In the early instants of a AMR run, one is able to attain a gain of about a factor of 3. All throughout the course of the simulation, the gain obtained with AMR is visible. Comparing the two adaptation techniques, for the earlier part of the simulations, Treeadapt performs slightly better or similarly than YALESadapt. Towards the end of the simulation, when the flame occupies most of the combustion chamber volume, the trend switches, YALESadapt showing better performance. Note finally that the retrieved number of iterations per second become almost identical for all cases. This is expected since the flame occupies the entire volume of the combustion chamber so a quite uniform resolution is expected just like in the static reference case.

Figure 5.15 shows the time step evolution during the reference and the AMR simulations. For both, the reference simulation and the AMR simulations, the values are comparable and don't vary much. The reference simulation has a

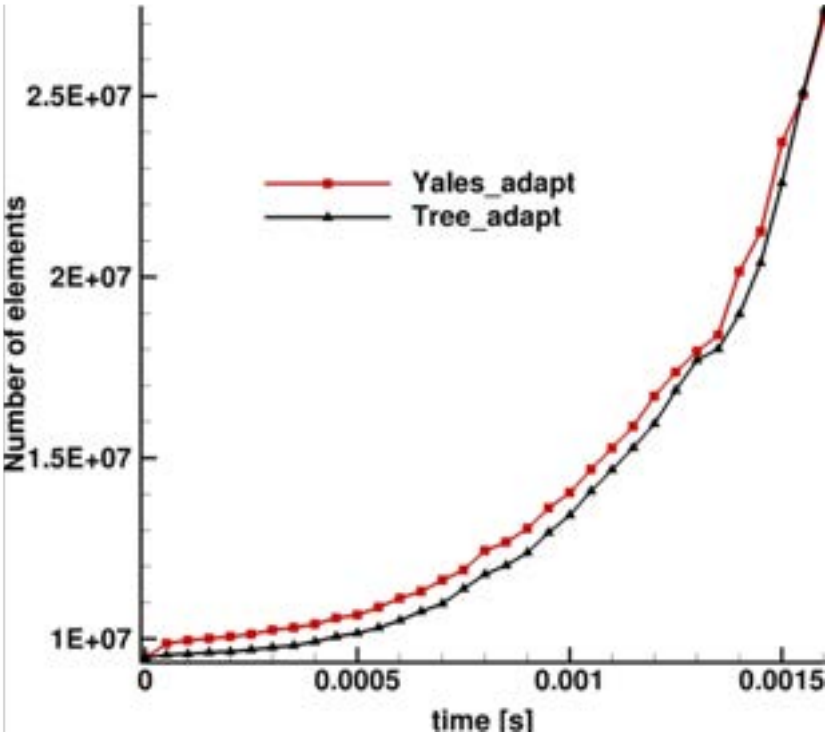


Figure 5.13: Time evolution of the number of elements in the mesh for YALESadapt and Treeadapt cases for the 3D Boeck setup.

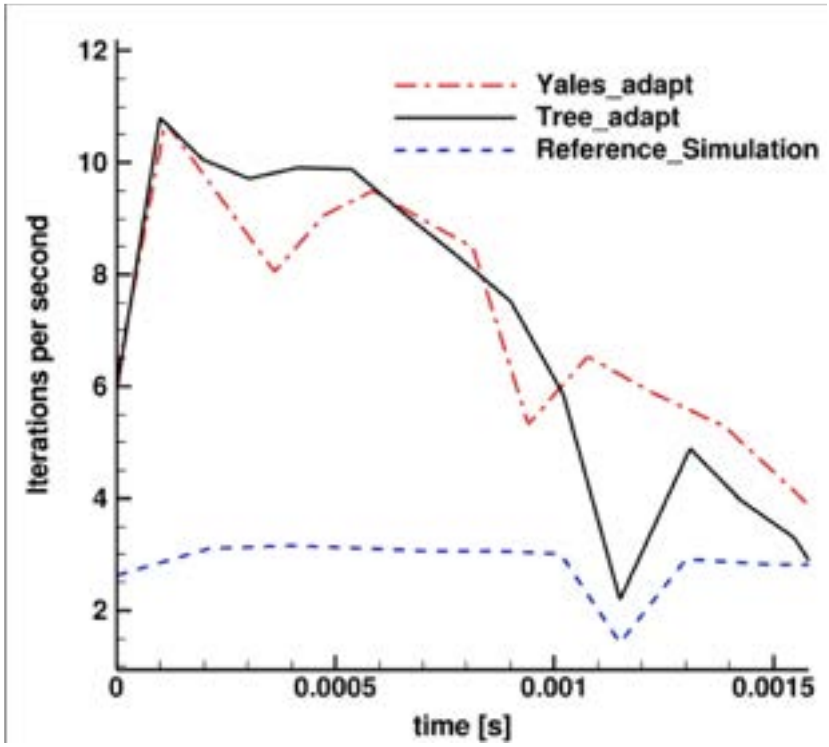


Figure 5.14: Comparison of iterations per second between Reference, YALESadapt and Treeadapt simulations for the 3D Boeck setup.

marginally larger timestep almost throughout the simulation while, the timestep of YALESadapt and Treeadapt remain comparable to each other. Recall, the smallest cell in the mesh is constrained by the flame front resolution in these simulations. As a result, and since the smallest grid size of the AMR is constructed to be equivalent to the reference simulation in the flame front, the time step should be comparable in all the simulations.

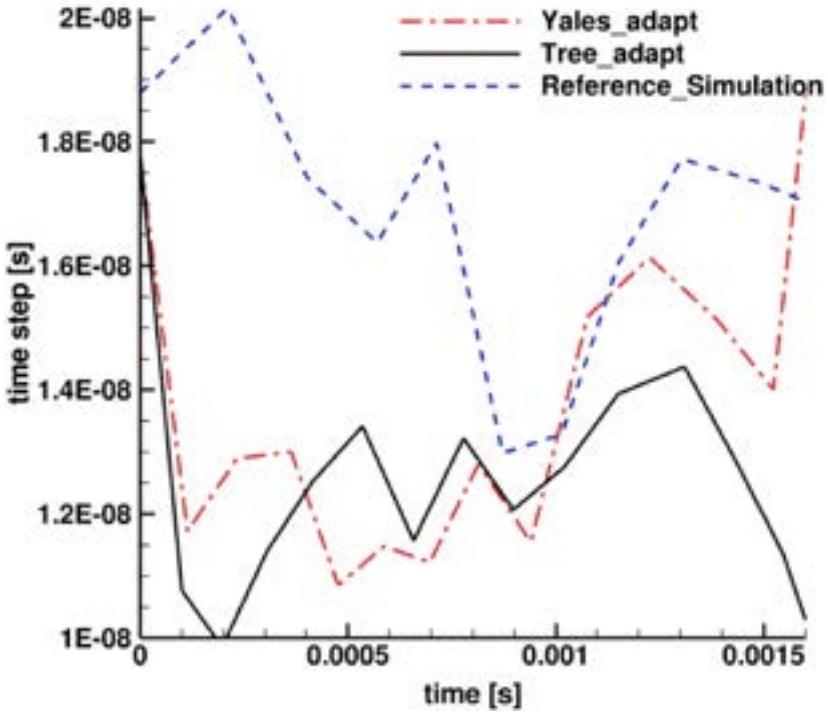
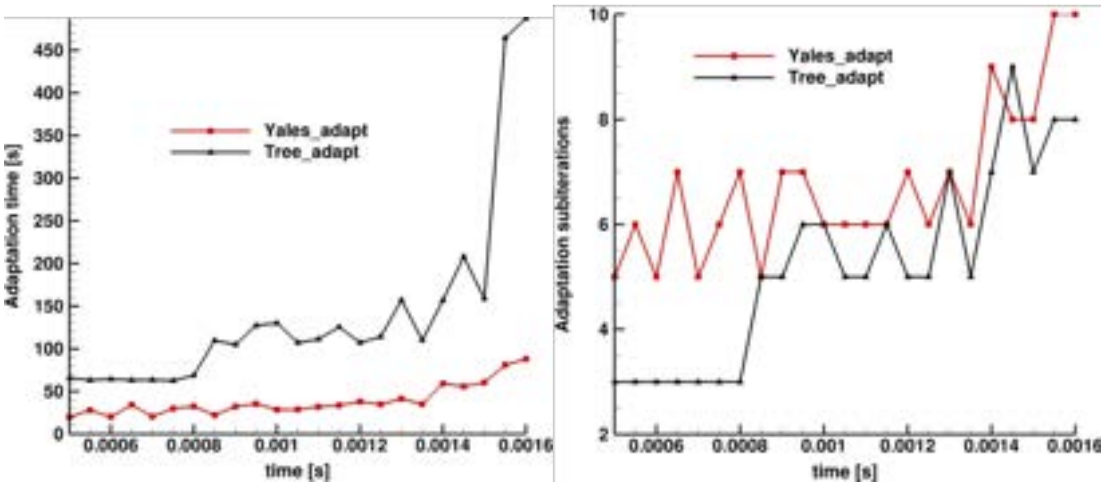


Figure 5.15: Comparison of timestep for reference, YALESadapt and Treeadapt simulations for the 3D Boeck setup.

Finally, the total time taken by the mesh adaptation loop for the two adaptation methods in shown in Fig. 5.16 (a). As a complement, Fig. 5.16 (b) provides the number of sub-iterations needed for the adaptation loop for both methodologies. The time taken to complete the adaptation loop appears longer with Treeadapt even though it takes fewer sub-iterations every time adaptation is called. This behavior is due to interpolation algorithm used, which is about approximately 65% of the total adaptation process needed with Treeadapt per adaptation step. For both the methodologies, as the flame starts approaching the end of the combustion chamber, after passing the large central obstacle, the adaptation time as well as the number of sub-iterations to converge both increase. The differences in the full simulation times for the two techniques, are due to the interpolation time required with Treeadapt. This can clearly be improved and will be addressed in subsequent versions of Treeadapt.



(a) (b)

Figure 5.16: Comparison of (a) total adaptation time and (b) sub-iterations in the AMR loop for the 3D Boeck setup for the adapted (YALESadapt and Treeadapt) simulations .

Feature	Ref. Simulation	YALESadapt	Treeadapt
Total time	32,815 s	20,932 s	32,696 s
Average Memory	156.792 MB	982.526 MB	843.024 MB
Min. Memory	139.742 MB	941.332 MB	754.828 MB
Max. Memory	168.102 MB	1061.152 MB	892.082 MB

Table 5.3: Comparison of total time for simulation and memory consumed by the reference simulation and the two adaptation techniques for Boeck 3D case using 540 processors.

The full comparison of total time for such simulations is compared to the reference case in Table 5.3. Time to solution with the adaptation and for an initial mesh of 9.5 million elements is seen to only result in a marginal improvement (by a factor of approximately 1.56 times using 540 processors) in comparison to the static reference simulation. It can also be seen from Table 5.3 that the full simulation with Treeadapt is approximately 56% slower than the one with YALESadapt. Again the reason for this difference is linked to the interpolation process cost and will be fixed. Faster interpolating techniques as well as a different method for load balancing are indeed already in progress for Treeadapt. The overall memory consumption increases when using AMR is a significant aspect of AMR and good mesh management is clearly required. It can also be observed the memory consumption is reduced with Treeadapt compared to YALESadapt. This is critical as illustrated here if an AMR methodology is to be used with larger meshes.

To conclude, based on this test case mesh adaptation provides only marginal

performance gains without loosing quality in the obtained predictions. To improve gains work is on going at CERFACS in terms of algorithm tuning: i.e. interpolation phase as well as by adding new features such as dynamic coarsening. This matter can be obtained through a redefinition of the adaptation masks. The following test case addresses a more complex problem more relevant to real explosions where transition to turbulence occurs.

5.2.2 Case B: MASRI Setup

The MASRI experiment [164, 172] is chosen next for further assessment of the AMR methods. The primary reason for choosing this setup is that its evolution results from a complex 3D flame interaction with multiple obstacles with an important influence from turbulence. Like in the previous configuration, laminar flame accelerates as it interacts with the obstacles and eventually transition to a fully turbulent flame. A relevant use of AMR algorithm therefore requires to successfully capture this complexity.

Both cases of the present chapter are highly unsteady flows with a flow that changes locally quite rapidly. With a uniform mesh, a high resolution is thus required throughout the domain for accurate simulations. Using AMR to locally modify the mesh as the simulation proceeds, can be very beneficial in terms of cell count. Indeed at ignition, a large portion of the chamber doesn't contain the flame and the flame only evolves through the chamber in time. The use of a coarse mesh in regions where there is no flame appears natural.

Note finally that similarly to the Boeck setup, a 2D setup is used to tune the AMR parameters that are then used for the full 3D MASRI setup.

Experimental setup

The SydGex database also called the MASRI test cases throughout this work contains three different experimental configurations: the so called 'small-scale' (SS) (0.25 m long) configuration as well as two upscaled versions: the 'medium-scale' (MS) (1.5 m long) and 'large-scale' (LS) (6.1 m long). In our study, only the small-scale setup [164] is simulated.

The small-scale MASRI experiment is shown in Fig. 5.17. The configuration consists of a square cross section ($0.05 \times 0.05\ m^2$) chamber, 0.25 m long with solid obstacles. Its overall volume is 0.625 liters for a length (L) and width (W) ratio of $L/W = 5$. Away from the ignition source, three removable baffle plates can be placed along with a central square obstacle fixed. The baffles may be located at one of three locations: 19 mm, 49 mm and 79 mm from the closed

end. Each baffle plate consists of five vertical bars, 4 mm wide, separated by a gap of 5 mm wide. Downstream of the baffle plates locates the central obstacle of square cross section and 96 mm from the closed end of the chamber.

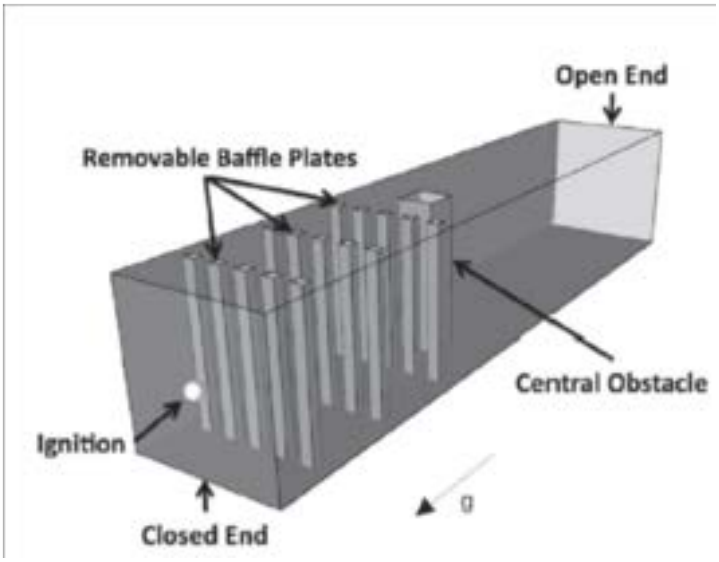


Figure 5.17: Explosion chamber configuration of MASRI.

The chamber is fully enclosed to the exception of an open end to the atmosphere shown in Fig. 5.17. For all the experimental test cases, the chamber is initially filled with a premixed mixture of fuel (propane) and air at atmospheric conditions. From this database it seen that, the arrangement of the baffle plates controls the flame speed, the flame front shape and thus the generated overpressure. For our study, the most complex configuration is chosen: i.e. three rows of baffle plates along with the fixed central obstacle as shown in Fig. 5.18.

Numerical setup

Figure 5.18 shows the 3D computational domain used for this study. The x, y and z axes are the streamwise, wall normal and transverse directions to the bottom wall respectively. The lengths and dimensions of the obstacles are all identical to those mentioned in the experiment. Note that an atmospheric plenum is added to the computational domain and has been removed here to just show the combustion chamber. On Fig. 5.18, the black dashed box corresponds to a zoomed view of the initial mesh shown next for the static reference simulation.

Figure 5.19 depicts a Y-normal sliced view of the numerical grid used for the static reference case. The zoomed region contains the middle baffle plates of the last row and the central obstacle. A high quality unstructured reference mesh with approximately 20 million elements with a typical cell size of 0.5 mm

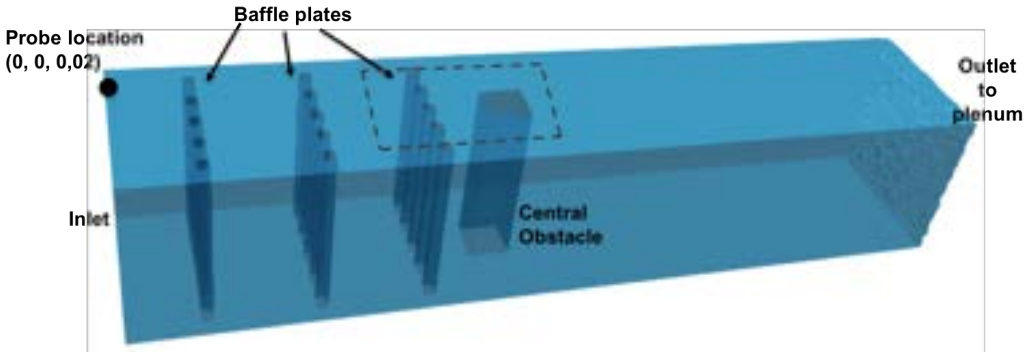


Figure 5.18: Computational domain used for the MASRI setup.

in the chamber is used. This mesh was chosen as it was found to be the most efficient setup for solving this specific problems by the authors of [128, 129] while maintaining accuracy of the results. This resolution indeed ensures that the flame, even thickened, remains thinner than the distance between the columns of the baffle plate. For the reference mesh, a high concentration of cells is observed near all obstacles, Fig. 5.19. The minimum volume of the cell in this reference mesh is $vol_{min} = 0.4598E - 12m^3$. A full view of the reference mesh is shown in Fig. 5.20. The mesh is well resolved around the baffle plates as well as the central obstacle with coarsening appearing downstream to alleviate the overall cost of such a simulation. Note hence that contrary to the Boeck test case, the reference case is not truly uniform.

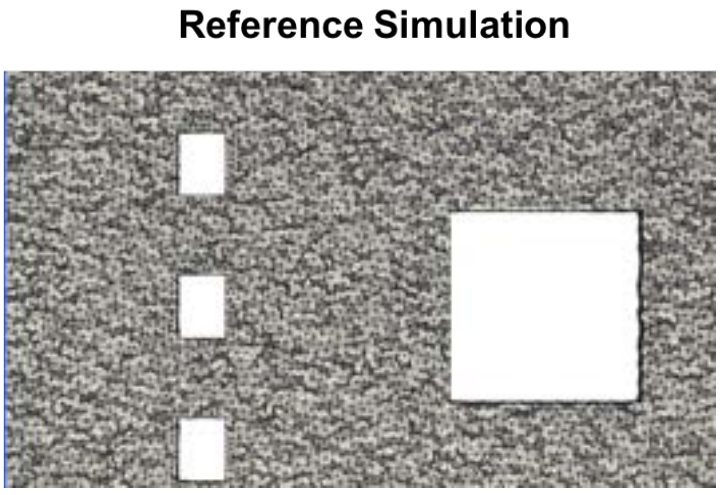


Figure 5.19: Computational grid: Y-normal slice at the center line for initial reference simulation.

All the initial conditions mentioned in the experimental setup are applied to the numerical simulations at time, t = 0. When the flame is ignited as in the case of the experiment no initial velocity is set for the entire domain and the flame is

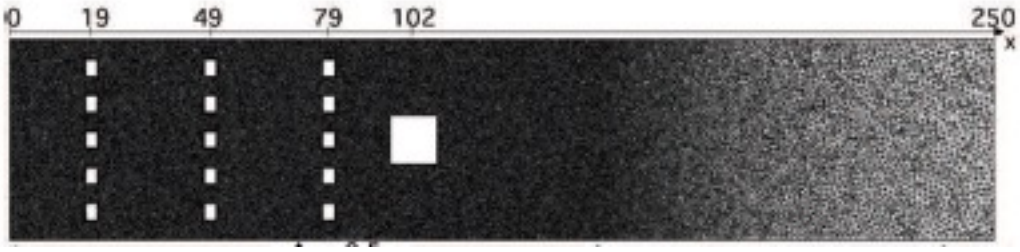


Figure 5.20: Homogeneous mesh used for the reference three-row baffle configuration [129].

ignited through a premixed spherical flame as shown in Fig. 5.17. In order to avoid the modeling of the ignition phase, initialization relies on the addition of a small sphere of burnt gases (radius 1 cm) at the ignition point [129]. The thickening is then chosen so that 5 cells are present inside the thermal thickness for the reference case. The efficiency function E used is the model of Colin [134] with β = 0.5 and the sub-grid-scale turbulence is modeled by the WALE model [173].

As mentioned for the Boeck setup, a single-step mechanisms is able to correctly predict the main features of flame acceleration. Therefore, a typical C_3H_8-air reduced scheme which matches the laminar flame speed, the flame thickness and the burnt gases adiabatic temperature is used here. It has 5 species and the details of the modifications done for this model are given in Ref. [128]. Recall that for the problem of interest, the laminar flame speed and laminar flame thickness are $S_L = 0.384$ m/s and $\delta_L = 0.368$ mm respectively.

The channel is closed so all boundary conditions correspond to walls except at the outlet. The solid walls that represent the obstacles and the explosion chamber are adiabatic nonslip walls.

As previously all simulations are performed with a centered Taylor-Galerkin scheme [68]. It is third order accurate in time and space. The simulation using the reference mesh is first compared with experiment in the following.

Comparison of the 3D reference simulation with the experiment

Figure 5.21 shows LES images of the flame propagation for the reference case compared to LIF-OH images from the experiments [164]. Qualitatively, at all the time frames mentioned, the experimental images and the reaction rate plots of the LES show excellent match. The various phases of the flame evolution are captured perfectly by the LES results. In the early stage of propagation, the flame is laminar and hemispherical as observed in the experiment. At $t - \Delta t_{peak} = 6$ ms, the flame hits the first baffle plate passing first through the two central passages and then the lateral passages. Here, Δt_{peak}, is the time at which the overpressure

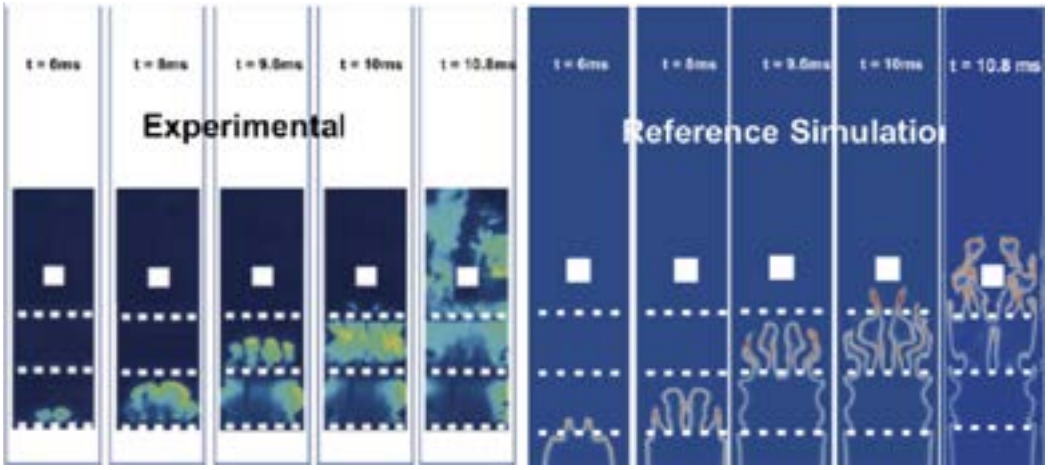


Figure 5.21: Comparison of flame propagation in the MASRI setup experimentally through time sequence of LIF-OH images [164] and time sequence obtained from the reaction rate of the reference simulation. Corresponding time ($t - \Delta t_{peak}$) is given in each frame.

curve reaches the peak value of overpressure. This time is used to shift all curves in time with respect to peak overpressure obtained in the experimental data.

The flame reaches the second baffle plate around time $t - \Delta t_{peak} = 8$ ms. At this point, the flame is still almost laminar as the turbulence generated in the wake of the first obstacle is very low. After passing through the second baffle plate, the flame starts accelerating at $t - \Delta t_{peak} = 9.6$ ms. Afterward, the flame begins to wrinkle due to the higher turbulence level generated by the obstacles around $t - \Delta t_{peak} = 10$ ms. The flame finally encounters the third baffle plate and the central obstacle around $t - \Delta t_{peak} = 10.8$ ms and continues to accelerate.

It is evident that the static reference simulation is here successfully able to capture the complex flow dynamics of the MASRI setup. As mentioned before, the goal of this study remains to perform such simulations using AMR at lower computational costs. But first and as in the case of Boeck setup, the AMR parameters need to be calibrated using a 2D MASRI setup before performing the 3D AMR simulations.

AMR calibration process

Just like for the Boeck setup and since mesh adaptation parameters are case dependent at this stage, a simpler 2D setup of the MASRI configuration is used for this determination process. Note that for this calibration, only YALESadapt is used.

To do so, a highly refined uniform (reference) simulation is performed first. This simulation uses a 2D mesh with 6.5 million triangular elements. Adaptation

simulations are then chosen so that the number of elements in the initial mesh are significantly less than this reference mesh. For these tests, the initial mesh fed to the 2D AMR YALESadapt tool contains $900,000$ triangular elements. One thing to note for the 2D as well as the 3D MASRI simulations, AMR is only applied to the combustion chamber. It is not applied to the attached atmospheric plenum since it is not of interest to us in terms of physics.

Compared to the Boeck case, MASRI is much more complex. Indeed, the reacting flow interacts with solid obstacles causing the laminar flame to eventually transition to a turbulent flame. A combination of a flame based and vorticity based sensor are hence used. To do so, a threshold value for both the vorticity as well as a minimum edge length- (corresponding to the same flame resolution (F = 1.0) as the reference simulation introduced above) are required. Similarly and due to lack of experimental data in 2D, to gain confidence in the reference 2D case, its resolution was chosen that the thickening factor remains close to unity. Complementary, the remeshing threshold of 2.5% is found to be sufficient to obtain AMR results comparable to this reference simulation. Note that, for the Boeck setup, the remeshing threshold of 5% was sufficient. Indeed for MASRI, the 5% remeshing threshold resulted in a discontinuous evolution of the flow quantities especially after the flame passes throughout the baffle plates and the central obstacle. This behavior has been identified as linked to the interpolation sequence which in MASRI generates large errors and a potential unphysical evolution of the simulations. When it comes to the minimum edge length used for the flame sensor with AMR, it is set in accordance with the reference case value that is 1.0 x $10^{-4}m$. Finally, the vorticity threshold is fixed to 6 x $10^{3}s^{-1}$. Note that all values of Table 5.4 were found to be optimal and any smaller value lead to inadequate overpressure curves.

Keyword for adaptation	Value
Min. edge size	1 x $10^{-4}m$
Remeshing metric threshold	0.025
Use vorticity metric	Yes
Use dtfles sensor metric	Yes
Vorticity threshold	6 x $10^{3}s^{-1}$

Table 5.4: List of important keywords for mesh adaptation for the 2D MASRI configuration.

For information, Fig. 5.22 shows a comparison of the temperature fields at $t = 6$ ms after ignition for (a) the reference and (b) the YALESadapt cases. Qualitatively, major features of the flow field are captured accurately. At this

specific instant, the number of elements in the adapted case has increased to 1.2 million triangular elements when the initial mesh had 900,000 elements. The evolution of the flame and its transition while going through the different obstacles is also captured, both simulations exhibiting negligible differences. This agreement between simulations was deemed hence to be sufficient to finalize the adaptation parameter identification. Obtained parameters are now used for both YALESadapt and Treeadapt simulations in the 3D MASRI setup. Detailed about obtained results are explained next.

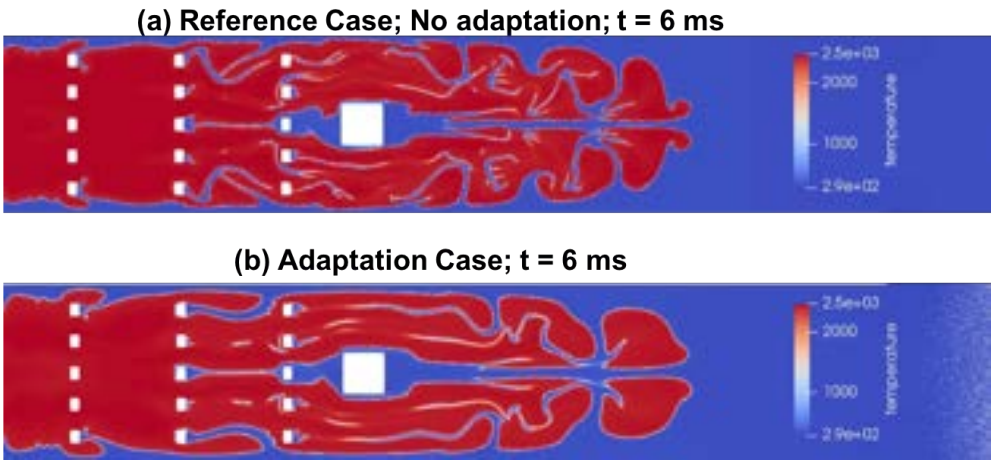


Figure 5.22: Comparison of temperature field between flame-resolved reference case and YALESadapt case for the 2D MASRI configuration at $t = 6$ ms.

Comparison of the 3D MASRI adaptation simulations with the experiment

As mentioned before, the reference 3D mesh chosen for the MASRI setup had 20 million tetrahedral elements and was found to be the most efficient mesh to yield good quality results [129]. This mesh corresponds to a flame resolution for which a thickening factor of $F = 7.2$ is applied. As a result, the minimum edge length is adjusted to 7.0 x $10^{-4}m$ to match the flame resolution for this 3D reference case. Similarly to the Boeck setup, the vorticity threshold obtained from the 2D simulation was sufficient to capture the flow dynamics for the 3D simulation as well and aside from the target thickening factor which changes all other parameters are kept identical to the 2D case in Table 5.5.

An initial arbitrary mesh is chosen for the 3D adaptation simulations. It contains approximately 10 million tetrahedral cell elements. This has approximately a half of the number of elements of the reference case. The two meshes are compared in Fig. 5.23. For the initial mesh used for AMR simulations in frame (b),

Keyword for adaptation	Value
Min. edge size	7 x $10^{-4}m$
Remeshing metric threshold	0.025
Use vorticity metric	Yes
Use dtfles sensor metric	Yes
Vorticity threshold	6 x $10^{3}s^{-1}$

Table 5.5: List of important keywords for mesh adaptation in the 3D MASRI configuration.

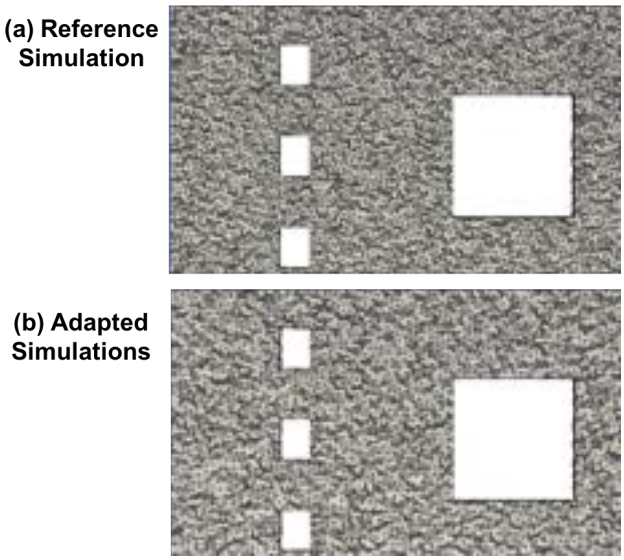


Figure 5.23: Computational grid: Y-normal slice at the center line for (a) initial reference simulation and (b) initial mesh used for the 3D AMR simulations.

the coarser nature of the initial mesh is evident. The minimum volume is of the order of $vol_{min} = 0.7723E-11m^3$ compared to $vol_{min} = 0.4598E-11m^3$ for the reference case.

A global visualization of the flow is provided using experimental images [164] of the flame evolution at different snapshots in the left column of Fig. 5.24. As a compliment, the right column of Fig. 5.24 shows LES images of the flame propagation for both AMR cases. The results for both adaptation cases (YALESadapt and Treeadapt) are shown. Visually, both AMR simulations show excellent match with the experimental images at all time instants. In the laminar phase for all the simulations (reference as well as both AMR simulations), the flame front moves at similar speeds. Note here that all figures are shifted by the time at which the peak overpressure is reached as done by other authors to compare with experiments [129].

After the flame passes through the second baffle plate, the flame starts ac-

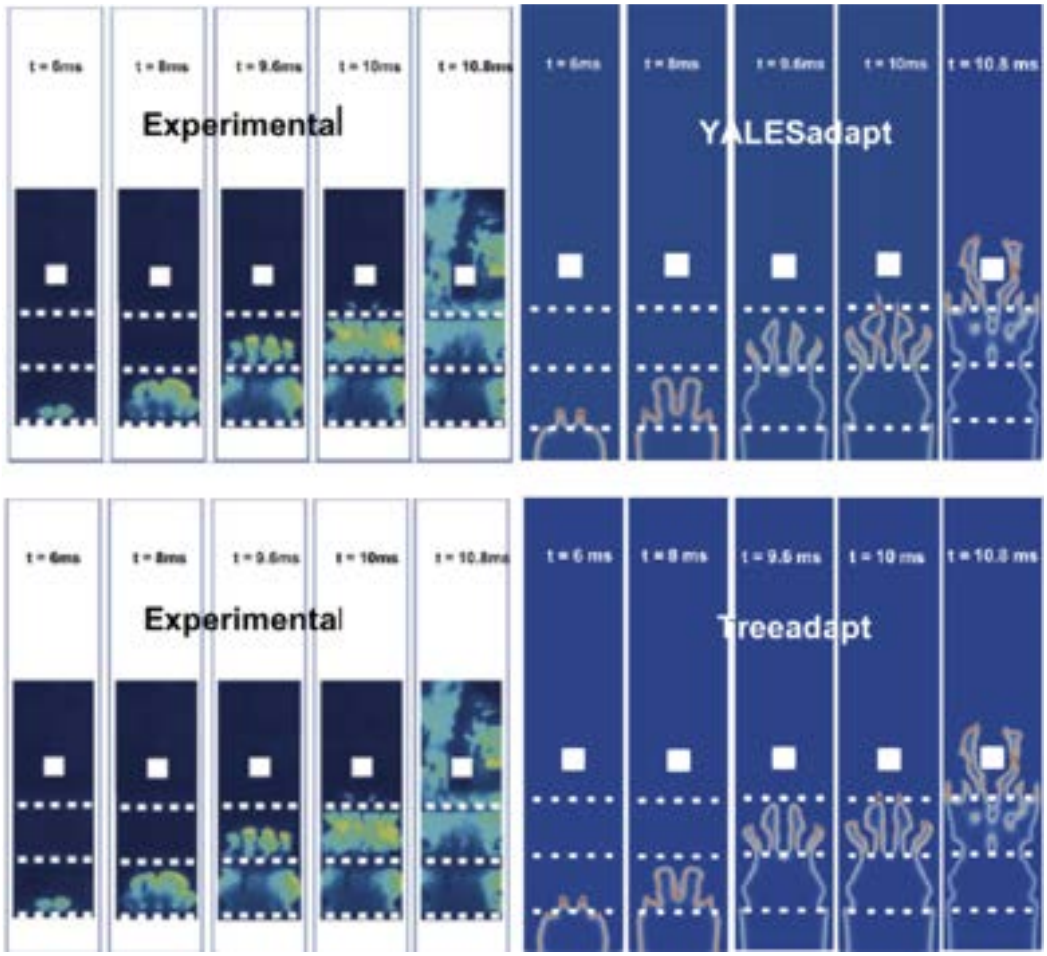


Figure 5.24: Comparison of flame propagation in the MASRI setup experimentally through time sequence of LIF-OH images [164] and time sequence obtained from the reaction rate of the AMR simulations. Corresponding time ($t - \Delta t_{peak}$) is given in each frame.

celerating and wrinkles due to high turbulence levels encountered. Beyond this time, there is a slight difference in the noted speed with which the flame front evolves when comparing the reference to the adapted simulations. However, both for YALESadapt and Treeadapt simulations, the flame fronts travel at identical speeds in all phases.

It is widely known that in such explosion test cases, the calculation of the overpressure is a vital quantitative to check the validity of numerical simulations. As shown in Fig. 5.18, a probe chosen at a specific location in the combustion chamber is used to compare measured data from the experiment [164] to simulations. In the experimental setup, a number of probes are available to collect data and identical probes are used in the simulation. It was found in the experimental setup that the overpressure signal collected at all the probes are identical which was also confirmed in all numerical simulations. As a result, the data at only one

probe location is presented hereafter.

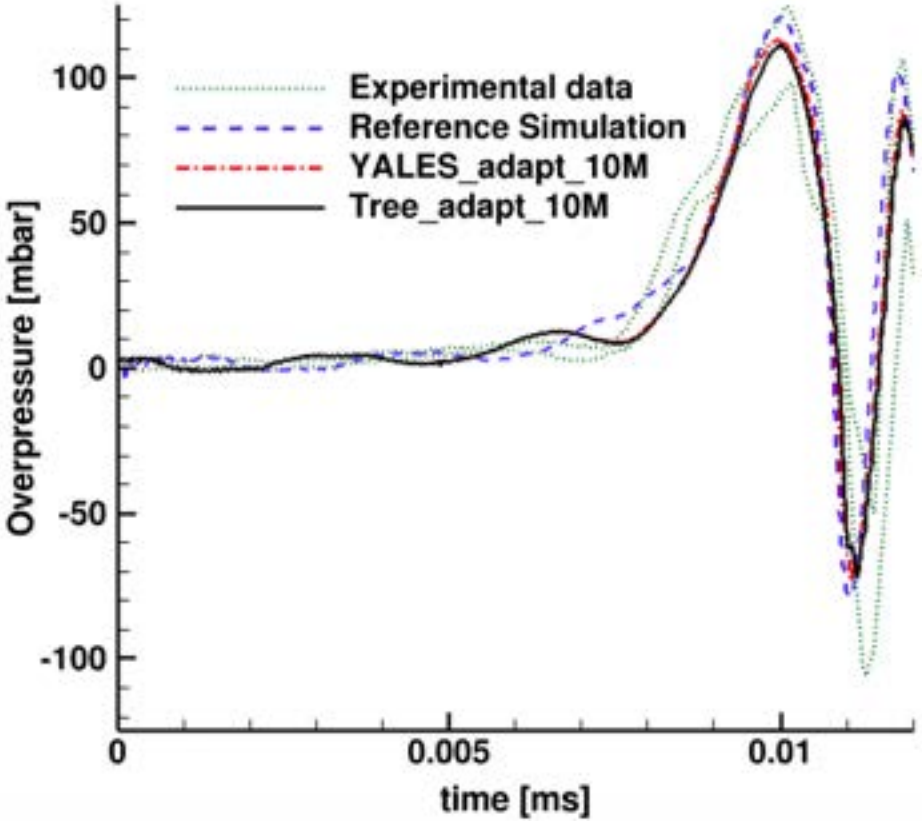


Figure 5.25: Comparison of overpressure between reference case, YALESadapt case and Treeadapt case with the experimental data [164] for the 3D MASRI configuration.

Figure 5.25 shows the comparison of the overpressure with time for the reference case, adapted cases and the experimental data. The minimum and maximum of the experimental data set are shown by the dotted blue curves. First, the reference simulation shows that the peak values of overpressure are within the experimental limits. Next, the AMR simulations indicated by the red and black lines show an excellent match and the curves are almost identical. All the simulations lie within the experimental bounds and follow similar trends for overpressure measured. One important thing to note here is that the curves for two AMR simulations (YALESadapt and Treeadapt) are not shifted in time. The flame front evolves in the combustion chamber with identical speeds. These results are sufficient to validate the AMR simulations, however like in the Boeck case, further comparison is required.

Results and discussion

In this section, similar as the Boeck setup, a quantitative comparison is done to compare the reference, the YALESadapt and the Treeadapt simulations. Results are discussed in the following order, starting with discussion on the overall dynamics of the flow followed by details regarding performance and cost for the 3D MASRI simulations.

Similarly to the Boeck case, the rate of venting and combustion are calculated and compared for all numerical simulations. Note that just like for the case of the overpressure plots, the curves have been shifted in time to match the peaks.

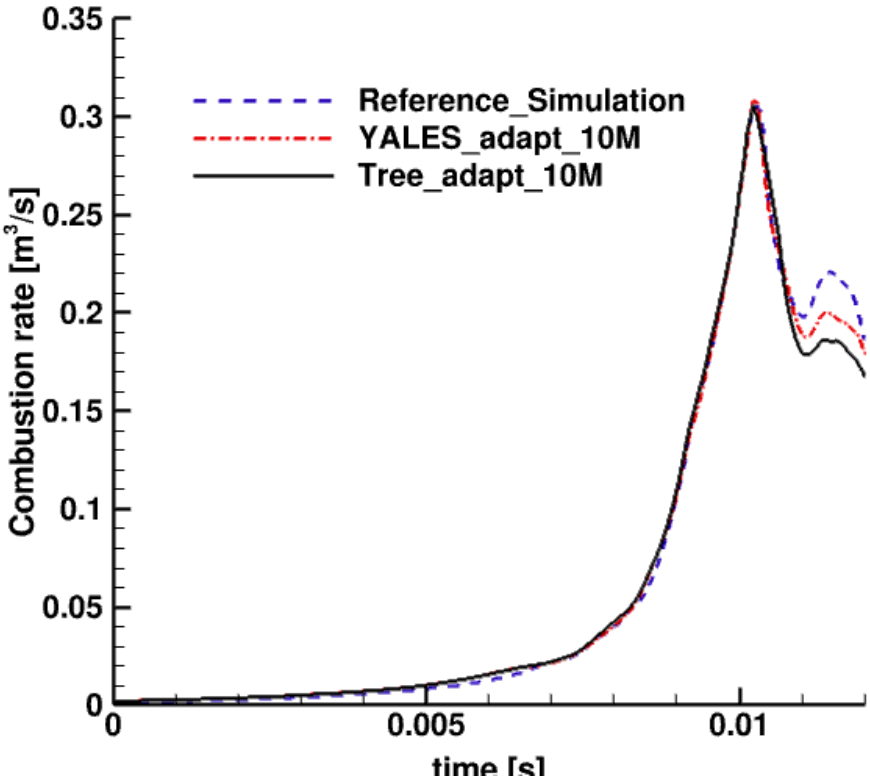


Figure 5.26: Comparison of combustion rates between reference case, YALESadapt case and Treeadapt case for the 3D MASRI configuration.

The rate of combustion are compared between the AMR cases and reference case in Fig. 5.26. The first peak of the rate of combustion is captured perfectly by all the simulations. There is a slight difference in the second peak but a similar trend is observed in all the cases. This difference can be attributed to the fact that when flame approaches the atmospheric plenum, mesh adaptation is no longer used and only applied to the combustion chamber.

Next, the venting rates are compared between the cases and plotted in Fig. 5.27. The first peak is captured very well as was the case for combustion rate. There a

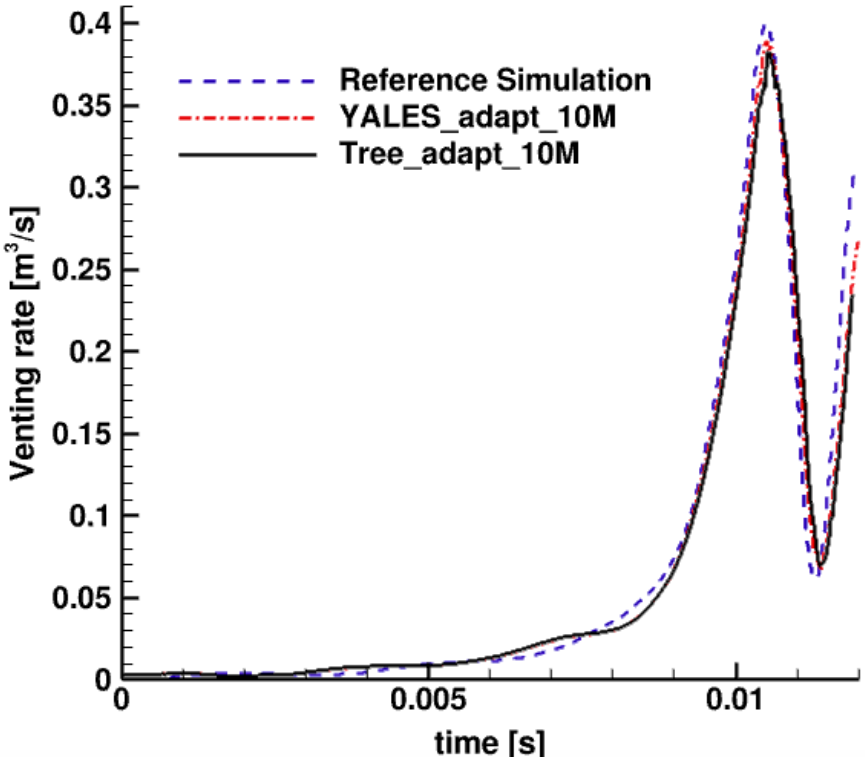


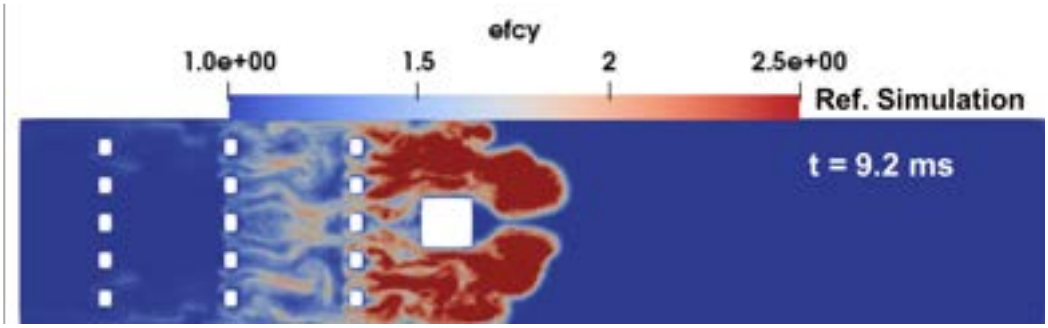
Figure 5.27: Comparison of venting rates between reference case, YALE-Sadapt case and Treeadapt case for the 3D MASRI configuration.

slight difference for the following peaks but again they follow the correct trends as explained above for the combustion rate. The curves of venting rates for YALE-Sadapt and Treeadapt match perfectly. The curves for both combustion and venting rates for the AMR simulations have not been shifted in time with respect to each other.The pressure differential created due to competing combustion and venting rates is captured perfectly by all of the simulations.

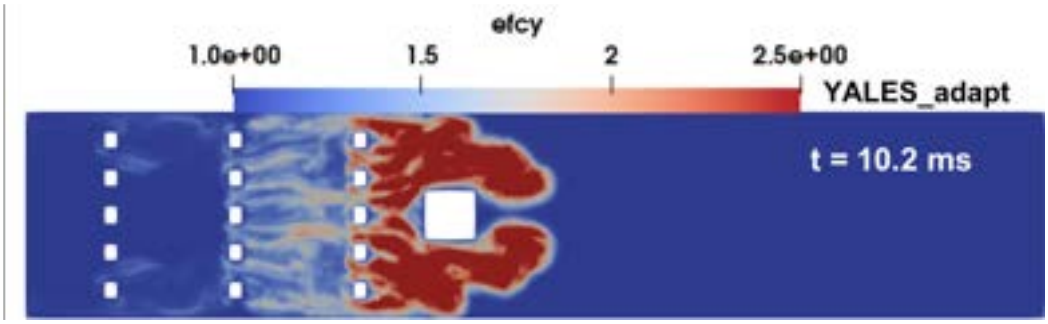
In the MASRI setup, the laminar flame transitions to a turbulent flame so modeling needs to be verified. In that respect, thickening being equivalent it is necessary to compare the efficiency function for all simulations. Indeed, the combustion model associated to the TFLES approach [134, 174] to resolve the flame front on the computational mesh relies on an efficiency function to account for the loss of wrinkling at the sub-grid scale when turbulence occurs. Two different efficiency functions are widely used, namely the model of Colin [134] and Charlette [131]. The efficiency function is defined as the wrinkling ratio Ξ_Δ between the non-thickened reference flame and the thickened flame.

$$\varepsilon = \frac{\Xi_\Delta(\delta_l^o)}{\Xi_\Delta(F\delta_l^o)} \tag{5.5}$$

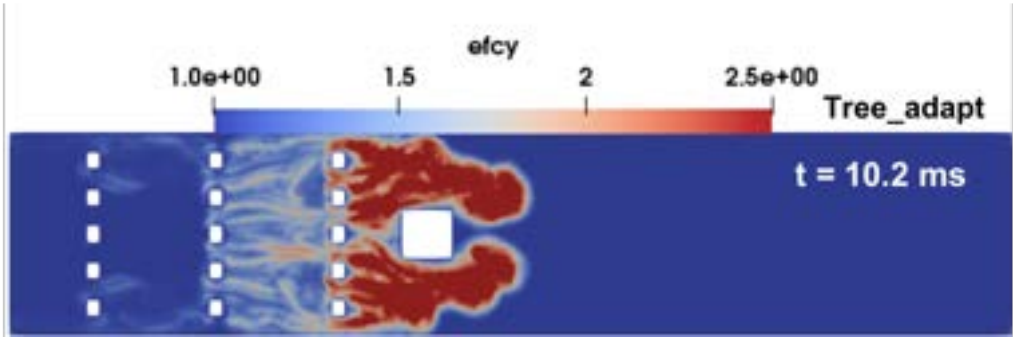
where δ_l^o is laminar flame thickness of the non-thickened flame and F is the



(a)



(b)



(c)

Figure 5.28: The efficiency function of the Colin model is plotted at the similar instants in the combustion chamber with regards to flame front evolution to compare the reference and the adapted simulations.

thickening factor. It is important to check that in the AMR simulations the applied efficiency function values are comparable and accurately triggered at the right places in the simulations since this drives the turbulent phase.

Figure 5.28 displays fields of efficiency functions for (a) the reference, (b) the YALESadapt and (c) the Treeadapt simulations, when the flame interacts with the central obstacle and the baffle plates. When the flame front is present at this location, turbulence levels are high as expected and the range of efficiency values are at their maximum values for the simulations. Clearly, for all simulations, the function activates similarly at the same locations, starting after the second row of

baffle plates and mainly downstream of the central obstacle. All simulations also show similar levels. The fact that the location at which the efficiency function is triggered as well as the values reached coincides with the excellent match obtained for the venting rates, Fig. 5.27. This confirms the ability of AMR to capture complex phenomena such as transition from laminar to turbulent flames due to flame-vortex interactions.

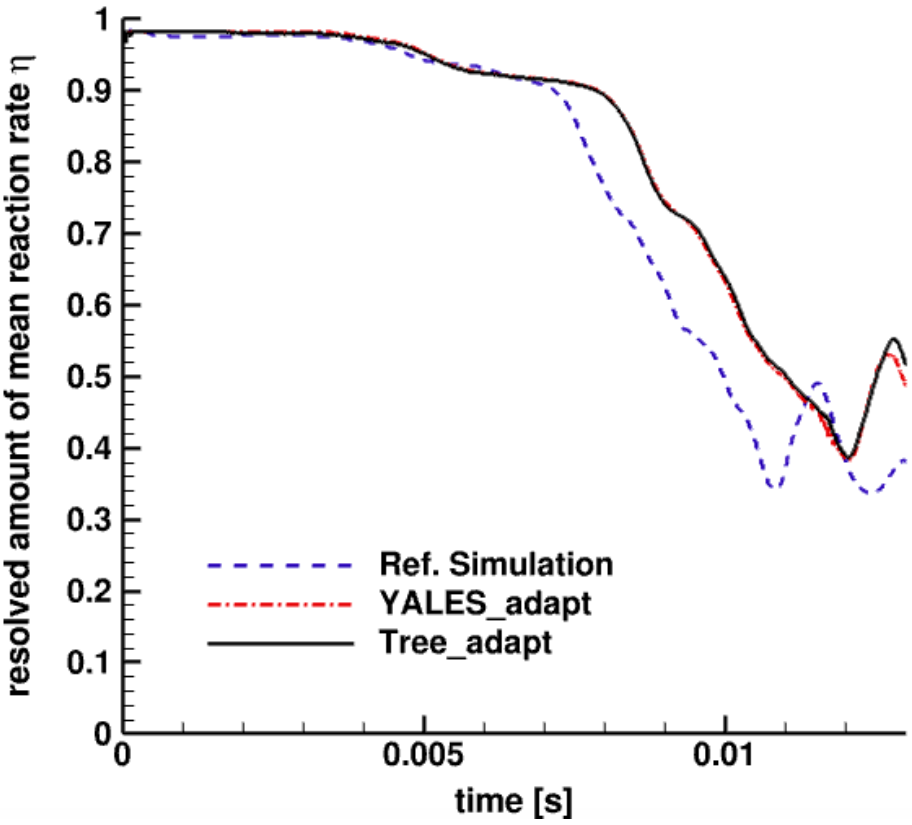


Figure 5.29: Resolved amount of mean reaction rate η for the adapted cases and the reference simulation for the 3D MASRI setup.

The comparison of the respective contributions given by the resolved and sub-grid-scale parts of the combustion rates are provided in Fig. 5.29. From such an evolution, it is seen that in areas where the flame is almost laminar (before the first obstacle), the flame wrinkling is fully resolved and the contribution of the sub-grid-scale combustion model is nearly zero. In this part of evolution, all the simulations show similar behavior. As the flame approaches the first baffle plate, the amount of mean resolved reaction rate reduces slightly due to the initiation of the sub grid model contribution. When the flame becomes fully turbulent, typically downstream the third baffle plate and near the central obstacle, the contribution of resolved part reduces drastically in favor to the model contribution. The drastic fall in the resolved mean reaction rate is seen at the same time

around approximately $t = 9$ for the AMR simulations. The reference simulation also shows similar trends, however, due to a faster moving flame front, the flame approaches the third baffle plate earlier and the curves seem apart. The curves for the two adaptation techniques (YALESadapt and Treeadapt) are identical and the flame front appears to travel at identical speeds. The mean value reaches as low as 35% in its lowest peak, when the flame occupies close to the entire combustion chamber.

All the above analysis performed for the simulations indicate that the AMR methodology has been successfully implemented for this complex test case. The performance based criteria are detailed next to understand the benefits of the AMR simulations. The gains in CPU is clearly one desired objective justifying the use of such a complex dynamic meshing algorithm. As a result, it is first necessary to compare the evolution of the number of elements in the mesh, the number of iterations per second performed by the solver and the timestep used as well as time taken by the AMR loop.

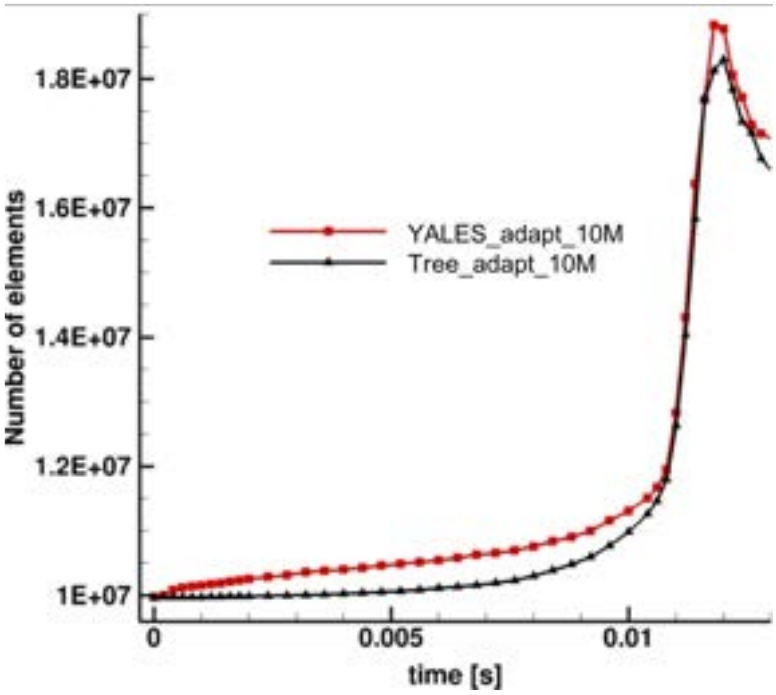


Figure 5.30: Number of elements in the mesh between YALESadapt and Treeadapt simulations for the 3D MASRI configuration.

As mentioned for the Boeck case, the gains observed while using AMR depends on the quality of the initial mesh used. Since, the initial AMR mesh is always coarser than the reference mesh, the major advantage in performance is observed in the early phase of the simulations. Its is therefore important to track the number of elements in the mesh for the AMR simulations as shown

in Fig. 5.30. Throughout the simulation, the number of elements in the case of the Treeadapt simulation is always slightly less than the YALESadapt case. This trend is observed in the manuscript irrespective of the case chosen. This is due to the fact that in YALESadapt as mentioned previously, a mesh optimization technique based on the improvement of the skewness is applied additionally in comparison to YALESadapt. At around $t = 12$ ms, the number of elements are quite similar and reaches the peak value for both cases (similar to the number of elements in the reference mesh) because the flame occupies the entire combustion chamber. Afterwards, when the flame enters the plenum, the number of elements start decreasing for all the cases as the mesh goes back to original resolution in the combustion chamber after the flame enters the atmospheric plenum.

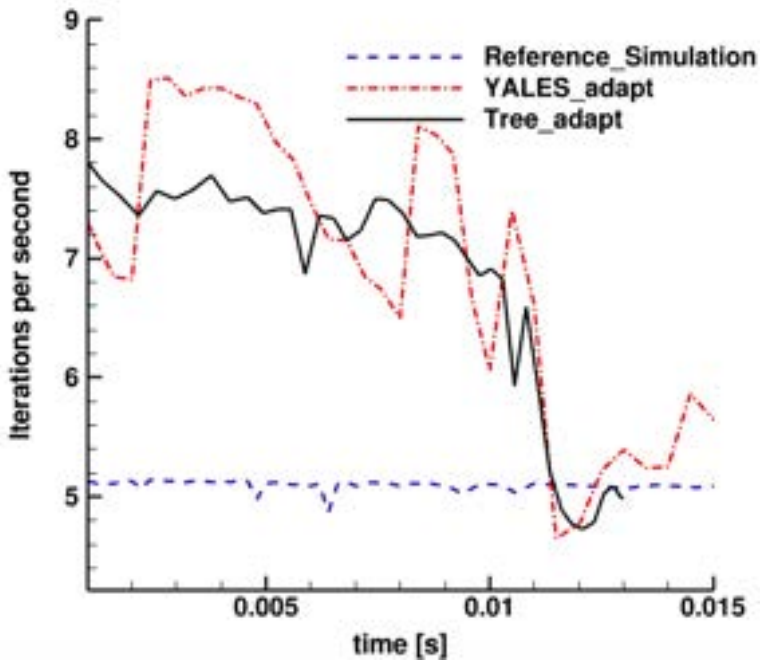


Figure 5.31: Comparison of iterations per second between Reference simulation, YALESadapt and Treeadapt simulations for the 3D MASRI configuration.

The number of elements in the mesh has a direct correlation to the number of iterations per second for the AMR simulations which are plotted in Fig. 5.31. The iterations per second clearly takes advantage of the dynamic mesh refinement with a gain in performance at the early stages for AMR simulations compared to reference simulation. This advantage is mainly due to the coarse mesh and the trends coincide with the number of elements observed in the mesh at these time instants. This increase in performance slowly reduces for both AMR cases with time. At around $t = 12$ ms, the iterations per second for all cases become

similar as this coincides with the fact that the flame occupies the complete chamber. Afterwards, when the flame starts exiting the chamber into the atmospheric plenum, there is an increase in the iterations per second for the adaptation cases as the mesh goes back to the original resolution in the combustion chamber: i.e. when the flame exits to the plenum. The iterations per second are similar for both adaptation techniques and show better performance than the reference case for most of the duration of the simulation.

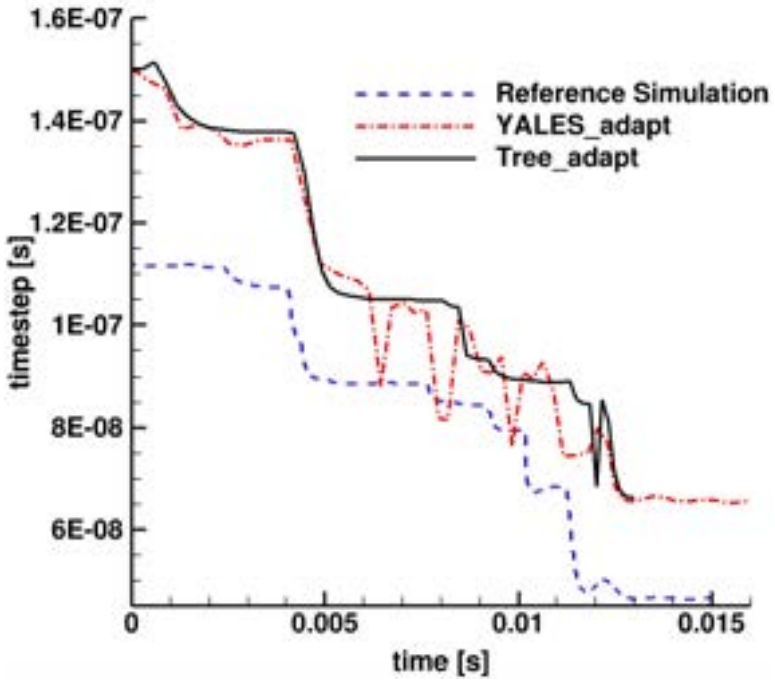


Figure 5.32: Comparison of timestep between Reference simulation, YALESadapt and Treeadapt simulations for the 3D MASRI configuration.

Note that, the time step used during these simulations depends mostly on the volume of the smallest cell present in the mesh. Figure 5.32 displays the timestep evolution for all three simulations. For both AMR cases, the simulation timestep is marginally larger than the reference case throughout the simulation. Since AMR aims at using coarser meshes, this is an expected result. The timestep difference between the YALESadapt and Treeadapt are present throughout the simulations but remain comparable for both methods. This confirms that the larger timestep resulting from the AMR tools provides a benefit especially in terms of CPU cost.

Although both AMR methodologies use the same algorithm and appear to yield similar behaviors, there are differences in the working of both algorithms as already highlighted in chapter 3. It is therefore required to compare the time

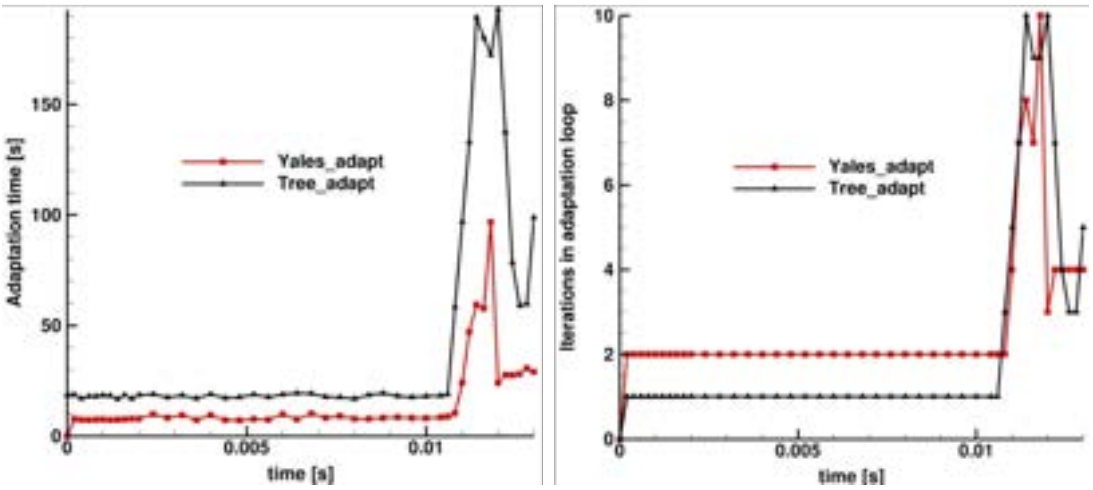


Figure 5.33: Comparison of total AMR time and number of sub-iterations per AMR loop between the two adapted cases with initial mesh of 10 million elements for the 3D MASRI configuration.

taken by the mesh adaptation loop for both methods as shown in left frame of Fig. 5.33. In the right frame, the number of sub-iterations in the adaptation loop per call of MMG are shown for both methodologies. The time taken to complete the adaptation process always takes longer in the case of Treeadapt although it takes fewer sub-iterations in the adaptation loop. This is attributed again to the interpolation time also observed for the Boeck setup and, explained in the previous section. Both adaptation techniques, follow similar trends and the adaptation time as well as the number of sub-iterations to converge increase for both YALESadapt and Treeadapt as expected. Afterwards, in the plenum, the adaptation time and number of sub-iterations decrease for both simulations.

Feature	Ref. simulation	YALESadapt	Treeadapt
Total time	42,235s	29,535 s	37,427 s
Average Memory	181.539 MB	2161.178 MB	1553.93 MB
Min. Memory	158.332 MB	2049.684 MB	1439.352 MB
Max. Memory	193.707 MB	2274.070 MB	1647.824 MB

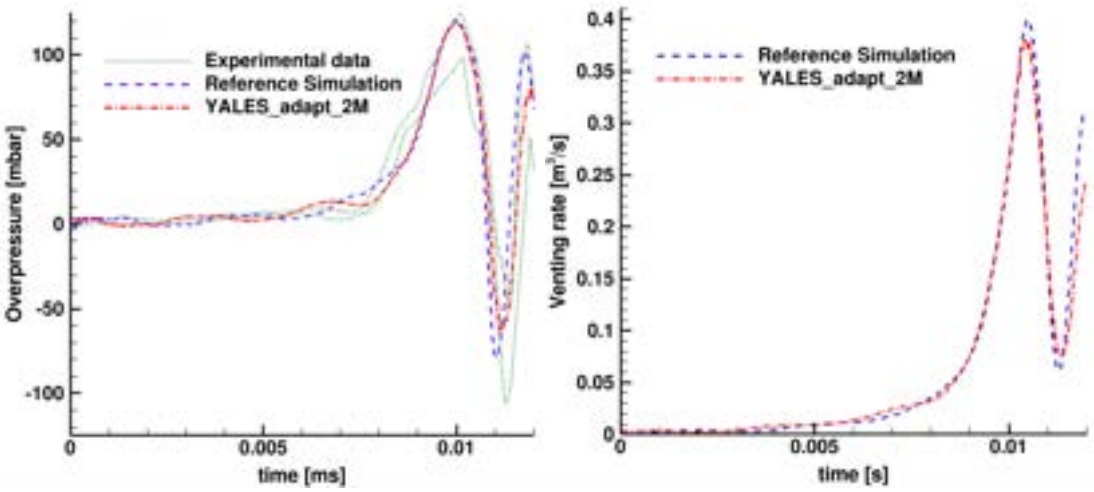
Table 5.6: Comparison of total time for simulation and memory consumed for the three simulations for MASRI 3D case using 360 processors.

Finally, the total time taken (start to end of the simulation) to complete the simulation for the two different adaptation techniques (YALESadapt and Treeadapt) with the same initial mesh of approximately 10 million elements are shown in Table 5.6. It can be observed that the entire simulation in this case with Treeadapt is approximately 26% slower compared to the adaptation by YALESadapt. This is a work in progress and will be improved. It can also be observed

that the memory consumption is significantly reduced in the case of Treeadapt compared to YALESadapt. Finally, a speed up of 1.31 is observed when using an initial mesh of 10 million elements with YALESadapt. For the present findings, all simulations were done using 360 processors on a local cluster available at CERFACS. An additional benefit from the AMR framework, is its capability to correct bad resolution zones at the outlet of the chamber that were responsible of numerical crashes in the static mesh.

Although marginal gains have been observed so far for the MASRI case using AMR, it is important to understand that the gains are all relative to the reference mesh chosen. This reference mesh was in fact optimized and chosen due to its cost effectiveness as well as accuracy of obtained results. Since the initial mesh (10 million) used so far was chosen arbitrarily so it has half the number of elements of the reference mesh, marginal gains can be simply a direct consequence of such a choice. It is also evident that gain in performance depends on the initial mesh used by construction since coarsening is not introduced. The objective of the upcoming test is therefore to see if a coarser initial mesh yields more performance gains while preserving accuracy. Note that, there should be a limit below which the initial mesh will yield inaccurate results. This specific limit is currently out of the scope of the next stage although it will have to be investigated. As a result a new AMR simulation is produced using a very coarse initial mesh containing only 2 million elements. This is a mesh that contains only 10% of the number of elements of the reference simulation. Before discussing the performance gains, it is essential to check first the accuracy of the provided results with such a change in initial grid.

Figure 5.34 compares the overpressure in frame (a) and the venting rates in frame (b) obtained with the 2M mesh in comparison to the reference results and experimental data when available. In frame (a), it can be observed that both simulations show similar trends and peaks in overpressure values lie within the experimental overpressure limits. In frame (b), the venting rates also show great match with only small differences. Effectively, when using this initial mesh, a speed up of approximately 2.1 is obtained (which is significantly better than the 10 million mesh). Note that, it was observed that when using an initial mesh with only one million elements, the overpressure curve did not lie within the experimental limits. This confirms our previous comment highlighting the balance between performance gains obtained using AMR and its parameterization to yield accurate predictions.



(a) (b)

Figure 5.34: Comparison of (a) overpressure and (b) Venting rate between Reference simulation and YALESadapt with an initial mesh of 2 million elements for the 3D MASRI configuration.

5.3 Conclusions and Perspectives

In this chapter, the dynamic mesh algorithm introduced previously is used to solve two complex, compressible, large-scale, reacting, test cases. To do so, a combination of a flame and a vorticity-based sensor is used and found to be adequate to adapt the mesh as the flow evolves in both the Boeck and MASRI setups. For the Boeck configuration, the results with mesh adaptation (both YALESadapt and Treeadapt) show excellent match with the reference simulation as well as the experimental data for this full 3D setup. For this case, only marginal gains are obtained for a simulation using 540 processors compared to the reference simulation at present. This confirms the suitability of the approach that now needs to be optimized to provide effective gains.

Similar, conclusions are reached for the more complex problem that is the 3D MASRI setup. Here again, excellent match for both mesh adaptation techniques are obtained if compared to the reference simulation or the experimental data. A performance gain of up to approximately 2.1 is observed for a coarse initial mesh of 2 million elements using 360 processors while preserving the prediction accuracy compared to the reference case for the full 3D MASRI setup. This confirms that using the correct parameters and selecting an efficient initial mesh, one will in the future avoid using static meshes for complex and large scale simulations while optimizing the overall cost of such simulations.

Note that from the first part of the thesis, it is observed that for reacting flows it is absolutely essential to resolve the flame front. The same conclusion is observed with AMR. It is therefore evident and now technically feasible to produce simulations accurately using a very coarse initial mesh provided a sufficient mesh resolution is captured via adaptation during the AMR simulations while thickening is ensured. The remaining challenge is thus to adequately combine these two elements for optimal use.

When it comes to performance gains observed with Treeadapt or YALESadapt, recall that one in still in progress and major improvement is expected in the future. The memory usage in all adaptation cases using Treeadapt in comparison to YALESadapt is significantly lower. A key advantage of Treeadapt is that it can handle much larger meshes than YALESadapt due to the hierarchical nature of Treeadapt which results in lesser memory usage. This should be extremely useful when performing simulations with meshes in the order of a billion elements.

Conclusions and Perspectives

LES simulations are frequently employed to obtain accurate numerical results in virtually all domains of combustion. The quality and accuracy of such LES predictions are however determined by a number of factors, one of the main being clearly mesh resolution/quality. There are a variety of issues related to a mesh that might occur when employing any numerical simulation. In that respect the understanding related to using any numerical scheme is explained in **Part I** of this thesis.

From a numerical analysis point of view, schemes can be better mastered and understood using the so called Global Spectral Analysis. As detailed in this first part of the document, GSA can be applied to the linear Convection-Diffusion Equation (LCDE) showing its potential over the more standard approach that is the Von-Neumann analysis. As a result of GSA, obtained property charts clearly evidence in 1D, the presence of standing waves as well as q-waves. Likewise, such charts give access to stability, dissipation and diffusivity of the numerical scheme. In particular, it is clearly found that the properties of a given scheme are sensitive to the parameters: the CFL number, N_c, the Peclet number, Pe, and wavenumber, kh. The extension to 2D problems is shown to introduce two new key parameters, the wave propagation angle and the spatial aspect ratio of the grid. Although purely mathematical, GSA derived behaviors are validated numerically solving for multiple 2D flows including the Taylor Green Vortex problem. All tests confirm that the properties of a numerical scheme are sensitive to mesh resolution just like evidenced by GSA.

Extension of GSA for linear Convection-Diffusion-Reaction Equation (LCDRE) provides a relevant model equation for the analysis and resolution of fully premixed combustion problems. This analysis clearly highlights that it is essential to have sufficient grid resolution in a flame front to obtain viable and stable numerical solutions. In that respect, it is of note that with the numerical schemes available within AVBP, at least 5 points are needed in the flame front to ensure proper predictions. This observation, issued by GSA, is fully in line with the standard recommended value when using the TFLES model.

The **scope for future work** associated with GSA would be to further extend and apply such an analysis to cover both non-reacting and reacting flows on equally spaced square grids as detailed here but also for grids with different aspect ratios. The same analysis could furthermore be extended to unstructured grids. In that respect and in the case of premixed flame, new recommendations about the needed mesh resolution in the flame front could be justified. Extending such a process to diffusion flames would also clearly be a major contribution allowing to better understand the choice of a scheme/resolution couple on the prediction and simulation of such flames.

Ideally, while solving full-scale reacting flow problems, it is impractical to use fine scale mesh resolutions as usually introduced in GSA. Considering the computational cost and needed resolution, it is clear that the area around the important flow features should be resolved with finer grids while relatively coarser grids could be used in other areas of the domain. The need for remeshing during a numerical simulation arises naturally for two different reasons: either to reduce the numerical inaccuracy induced by an inadequate mesh or because of a constantly evolving reacting flow field. **In Part II**, the dynamic mesh adaptation technique is introduced with this in mind.

To do so, the adaptive mesh refinement (AMR) methodology and its implementation based on two different techniques (YALESadapt and Treeadapt) are introduced. The general steps involved in such parallel dynamic mesh refinement algorithms are in particular explained and the differences between the two techniques are highlighted and explained. Two simple test cases are then selected to validate both adaptation techniques and ensure the correctness of the algorithms used.

Following these preliminary steps, two explosion cases dedicated to LES of turbulent complex combustion problems are treated with AMR. In the first case, the flow in a chamber with one central obstacle, name the Boeck setup is simulated. This case is relatively challenging, the flame undergoing wrinkling but staying laminar throughout. For the second case, the MASRI setup, has additional baffle plates and a central obstacle. This case mimics the flow evolution expected in an explosion where the flame transitions from a laminar to a turbulent flame with a large acceleration. Thanks to a proper methodology based on flame as well as vorticity based sensors, excellent match is obtained using both AMR techniques and performance gains are observed for each test case. From these simulations, it is furthermore noted that if the correct mesh adaptation parameters are used to obtain sufficient resolution, especially for the flame front,

much coarser meshes can produce accurate results. This corroborates the findings from GSA which indicates that the resolution of the flame front needs to be ensured to obtain accurate and stable results. This is here confirmed if the dual use of the thickened flame model and five points in the reacting front provided a prescribed grid resolution.

In terms of future actions and opportunities around AMR, mesh adaptation was here performed only for a small-scale setup of the MASRI case. There exists much larger scale setups. These should be attempted since most likely it will provide further gains in computational costs and speed up. In these cases since the combustion chamber is longer, a coarse initial mesh can be used to take fully advantage of the AMR technique without the burden of a static mesh approach.

However as underlined in this manuscript, at the present stage, the masks used in the AMR algorithm are case dependent and need to be tuned for proper use. One future goal would clearly to rely on case independent masks for fully and automatic triggering of AMR. Finally and for reacting flows, the link between an optimum grid resolution and the thickening factor, F, of TFLES should be somehow clarified.

To finish, the combined use of GSA, thickening and mesh adaptation could lead better understanding of attainable goals in terms of gain and accuracy. But such strategies also require to be able to coarsen in regions where the flow field is of less importance as time proceeds to truly highlight the advantages of AMR. Using the correct implementation of all of these steps can result in disparition of static mesh simulations which are still the current mean for industrial applications in the future.

List of Figures

List of Tables

Bibliography

[1] Annual Energy Outlook. with projections to 2040. *US Energy Information Administration*, 2011. 2, 181

[2] Mathieu Boileau, Gabriel Staffelbach, Benedicte Cuenot, Thierry Poinsot, and Claude Bérat. Les of an ignition sequence in a gas turbine engine. *Combustion and Flame*, 154(1-2):2–22, 2008. 2

[3] Dilip R Ballal and Arthur H Lefebvre. The influence of flow parameters on minimum ignition energy and quenching distance. In *Symposium (International) on Combustion*, volume 15, pages 1473–1481. Elsevier, 1975. 2

[4] M Champion, B Deshaies, G Joulin, and K Kinoshita. Spherical flame initiation: Theory versus experiments for lean propane- air mixtures. *Combustion and flame*, 65(3):319–337, 1986. 2

[5] Thompson M. Sloane. Energy requirements for spherical ignitions in methane-air mixtures at different equivalence ratios. *Combustion science and technology*, 73(1-3):351–365, 1990. 2

[6] Jean-Luc Beduneau, Bonggyu Kim, Laurent Zimmer, and Yuji Ikeda. Measurements of minimum ignition energy in premixed laminar methane/air flow by using laser induced spark. *Combustion and flame*, 132(4):653–665, 2003. 2

[7] A Alkidas and P Durbetaki. Ignition characteristics of a stagnation point combustible mixture. *Combustion Science and Technology*, 3(4):187–194, 1971. 2

[8] M Baum and T Poinsot. Effects of mean flow on premixed flame ignition. *Combustion science and technology*, 106(1-3):19–39, 1995. 2

[9] Yl Ko, Vedat S Arpaci, and RW Anderson. Spark ignition of propane-air mixtures near the minimum ignition energy: Part ii. a model development. *Combustion and flame*, 83(1-2):88–105, 1991. 2

[10] Thompson M Sloane and Paul D Ronney. A comparison of ignition phenomena modelled with detailed and simplified kinetics. *Combustion science and technology*, 88(1-2):1–13, 1993. 2

[11] Thompson M. Sloane. Ignition and flame propagation modeling with an improved propane oxidation mechanism. *Combustion science and technology*, 83(1-3):77–96, 1992. 2

[12] Thompson M. Sloane. Numerical simulation of electric spark ignition in methane-air mixtures at pressures above one atmosphere. *Combustion science and technology*, 86(1-6):121–133, 1992. 2

[13] Tim Edwards, Cliff Moses, and Fred Dryer. Evaluation of combustion performance of alternative aviation fuels. In *46th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, page 7155, 2010. 3

[14] Stephen Dooley, Sang Hee Won, Saeed Jahangirian, Yiguang Ju, Frederick L Dryer, Haowei Wang, and Matthew A Oehlschlaeger. The combustion kinetics of a synthetic paraffinic jet aviation fuel and a fundamentally formulated, experimentally validated surrogate fuel. *Combustion and flame*, 159(10):3014–3020, 2012. 3

[15] Thierry Poinsot. Prediction and control of combustion instabilities in real engines. *Proceedings of the Combustion Institute*, 36(1):1–28, 2017. 3

[16] Gang Wu, ZhengLi Lu, Yiheng Guan, Yuelin Li, and CZ Ji. Characterizing nonlinear interaction between a premixed swirling flame and acoustics: Heat-driven acoustic mode switching and triggering. *Energy*, 158:546–554, 2018. 3

[17] N Hajialigol and Kiumars Mazaheri. Thermal response of a turbulent premixed flame to the imposed inlet oscillating velocity. *Energy*, 118:209–220, 2017. 3

[18] Sherif S Rashwan, Medhat A Nemitallah, and Mohamed A Habib. Review on premixed combustion technology: stability, emission control, applications, and numerical case study. *Energy & Fuels*, 30(12):9981–10014, 2016. 3

[19] Gautam Kalghatgi. Is it really the end of internal combustion engines and petroleum in transport? *Applied energy*, 225:965–974, 2018. 3

[20] Fuquan Zhao, Thomas N Asmus, Dennis N Assanis, John E Dec, James A Eng, and Paul M Najt. Homogeneous charge compression ignition (hcci) engines. 2003. 3

[21] Haifeng Liu, Qinglong Tang, Zhi Yang, Xingwang Ran, Chao Geng, Beiling Chen, Lei Feng, and Mingfa Yao. A comparative study on partially premixed combustion (ppc) and reactivity controlled compression ignition (rcci) in an optical engine. *Proceedings of the Combustion institute*, 37(4):4759–4766, 2019. 3

[22] Sage L Kokjohn, Reed M Hanson, DA Splitter, and RD Reitz. Fuel reactivity controlled compression ignition (rcci): a pathway to controlled high-efficiency clean combustion. *International Journal of Engine Research*, 12(3):209–226, 2011. 3

[23] MM Rahman, Thamir K Ibrahim, K Kadirgama, R Mamat, and Rosli A Bakar. Influence of operation conditions and ambient temperature on performance of gas turbine power plant. In *Advanced Materials Research*, volume 189, pages 3007–3013. Trans Tech Publ, 2011. 4

[24] Sunday Olayinka Oyedepo, RO Fagbenle, SS Adefila, and SA Adavbiele. Performance evaluation and economic analysis of a gas turbine power plant in nigeria. *Energy Conversion and Management*, 79:431–440, 2014. 4

[25] Thamir K Ibrahim, Firdaus Basrawi, Omar I Awad, Ahmed N Abdullah, G Najafi, Rizlman Mamat, and FY Hagos. Thermal performance of gas turbine power plant based on exergy analysis. *Applied thermal engineering*, 115:977–985, 2017. 4

[26] Madhav Ghanta, Darryl Fahey, and Bala Subramaniam. Environmental impacts of ethylene production from diverse feedstocks and energy sources. *Applied Petrochemical Research*, 4(2):167–179, 2014. 4

[27] Ismaël Amghizar, Laurien A Vandewalle, Kevin M Van Geem, and Guy B Marin. New trends in olefin production. *Engineering*, 3(2):171–178, 2017. 4

[28] LH Hu, R Huo, YZ Li, and HB Wang. Experimental study on the burning characteristics of wood cribs in a confined space. *Journal of fire sciences*, 22(6):473–489, 2004. 5

[29] Guy Marlair, Claude Cwiklinski, and A Tewarson. An analysis of some practical methods for estimating heats of combustion in fire safety studies. In *Interflam 99*, 1999. 6

[30] H Xue, JC Ho, and YM Cheng. Comparison of different combustion models in enclosure fire simulation. *Fire Safety Journal*, 36(1):37–54, 2001. 6

[31] Chung King Law and Gerard M Faeth. Opportunities and challenges of combustion in microgravity. *Progress in Energy and Combustion Science*, 20(1):65–113, 1994. 6

[32] V Novozhilov, Behdad Moghtaderi, DF Fletcher, and JH Kent. Computational fluid dynamics modelling of wood combustion. *Fire safety journal*, 27(1):69–84, 1996. 6

[33] Stephen B Pope. Stochastic lagrangian models of velocity in homogeneous turbulent shear flow. *Physics of fluids*, 14(5):1696–1702, 2002. 7, 8

[34] Philippe Spalart and Steven Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th aerospace sciences meeting and exhibit*, page 439, 1992. 7, 9

[35] Guillaume Boudier, LYM Gicquel, Thierry Poinsot, D Bissieres, and C Bérat. Comparison of les, rans and experiments in an aeronautical gas turbine combustion chamber. *Proceedings of the Combustion Institute*, 31(2):3075–3082, 2007. 7

[36] Philippe R Spalart. Detached-eddy simulation. *Annual review of fluid mechanics*, 41(1):181–202, 2009. 7

[37] Javier Jiménez, Juan C Del Alamo, and Oscar Flores. The large-scale dynamics of near-wall turbulence. *Journal of Fluid Mechanics*, 505:179–199, 2004. 7

[38] Andrey Nikolaevich Kolmogorov. A refinement of previous hypotheses concerning the local structure of turbulence in a viscous incompressible fluid at high reynolds number. *Journal of Fluid Mechanics*, 13(1):82–85, 1962. 7

[39] Juan C del Álamo, Javier Jiménez, Paulo Zandonade, and Robert D Moser. Scaling of the energy spectra of turbulent channels. In *APS Division of Fluid Dynamics Meeting Abstracts*, volume 56, pages DA–002, 2003. 7

[40] Stephen B Pope and Stephen B Pope. *Turbulent flows*. Cambridge university press, 2000. 8, 9

[41] Eric Garnier, Nikolaus Adams, and Pierre Sagaut. *Large eddy simulation for compressible flows*. Springer Science & Business Media, 2009. 9

[42] Edward R Van Driest. On turbulent flow near a wall. *Journal of the aeronautical sciences*, 23(11):1007–1011, 1956. 10

[43] Joseph Smagorinsky. General circulation experiments with the primitive equations: I. the basic experiment. *Monthly weather review*, 91(3):99–164, 1963. 10

[44] Franck Nicoud, Hubert Baya Toda, Olivier Cabrit, Sanjeeb Bose, and Jungil Lee. Using singular values to build a subgrid-scale model for large eddy simulations. *Physics of fluids*, 23(8):085106, 2011. 10

[45] Robert Vichnevetsky and John B Bowles. *Fourier Analysis of Numerical Approximations of Hyperbolic Equations*. SIAM, 1982. 10, 11

[46] Lloyd N Trefethen. Group velocity in finite difference schemes. *SIAM review*, 24(2):113–136, 1982. 10, 11

[47] Tapan K Sengupta, Gaurav Ganeriwal, and S De. Analysis of central and upwind compact schemes. *Journal of Computational Physics*, 192(2):677–694, 2003. 10, 22, 68

[48] Tapan K Sengupta and Anurag Dipankar. A comparative study of time advancement methods for solving navier–stokes equations. *Journal of Scientific Computing*, 21(2):225–250, 2004. 10

[49] Tapan K Sengupta, SK Sircar, and Anurag Dipankar. High accuracy schemes for dns and acoustics. *Journal of Scientific Computing*, 26(2):151–193, 2006. 10

[50] Tapan K Sengupta, Anurag Dipankar, and Pierre Sagaut. Error dynamics: beyond von neumann analysis. *Journal of Computational Physics*, 226(2):1211–1218, 2007. 11, 22, 24, 25, 67, 69, 70

[51] Thierry Poinsot and Denis Veynante. *Theoretical and Numerical Combustion*. 3rd ed. edition, 2011. Available online: www.cerfacs.fr/elearning. 11, 85

[52] Hang Si and A TetGen. A quality tetrahedral mesh generator and three-dimensional delaunay triangulator. *Weierstrass Institute for Applied Analysis and Stochastic, Berlin, Germany*, 81, 2006. 12

[53] Hari Sundar, Rahul S Sampath, and George Biros. Bottom-up construction and 2: 1 balance refinement of linear octrees in parallel. *SIAM Journal on Scientific Computing*, 30(5):2675–2708, 2008. 12

[54] James R Stewart and H Carter Edwards. A framework approach for developing parallel adaptive multiphysics applications. *Finite Elements in Analysis and Design*, 40(12):1599–1617, 2004. 12

[55] Samir Vinchurkar and P Worth Longest. Evaluation of hexahedral, prismatic and hybrid mesh styles for simulating respiratory aerosol dynamics. *Computers & Fluids*, 37(3):317–331, 2008. 12

[56] David Martineau, Sean Stokes, Simon Munday, Andrew Jackson, Brian Gribben, and Niek Verhoeven. Anisotropic hybrid mesh generation for industrial rans applications. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, page 534, 2006. 12

[57] I Babuvška and Werner C Rheinboldt. Error estimates for adaptive finite element computations. *SIAM Journal on Numerical Analysis*, 15(4):736–754, 1978. 13

[58] I Babuska and A Miller. A-posteriori error estimates and adaptive techniques for the finite element method. Technical report, MARYLAND UNIV COLLEGE PARK INST FOR PHYSICAL SCIENCE AND TECHNOLOGY, 1981. 13

[59] AC Hindmarsh. Scientific computing, applications of mathematics and computing to the physical sciences vol 1 ed rs stepleman et al, 1983. 13

[60] Gustavo Adolfo Rios Rodriguez, Norberto Marcelo Nigro, and Mario Alberto Storti. An h-adaptive unstructured mesh refinement strategy for unsteady problems. 2009. 13

[61] TW Nehl and DA Field. Adaptive refinement of first order tetrahedral meshes for magnetostatics using local delaunay subdivisions. *IEEE transactions on magnetics*, 27(5):4193–4196, 1991. 13, 14

[62] Riccardo Hertel and H Kronmuller. Adaptive finite element mesh refinement techniques in three-dimensional micromagnetic modeling. *IEEE transactions on magnetics*, 34(6):3922–3930, 1998. 13

[63] Thomas Grätsch and Klaus-Jürgen Bathe. A posteriori error estimation techniques in practical finite element analysis. *Computers & structures*, 83(4-5):235–265, 2005. 13

[64] HL De Cougny and Mark S Shephard. Parallel refinement and coarsening of tetrahedral meshes. *International Journal for Numerical Methods in Engineering*, 46(7):1101–1125, 1999. 13, 14

[65] N Jansson, J Hoffman, and J Jansson. Parallel adaptive fem cfd. Technical report, Technical Report KTH-CTL-4008, Computational Technology Laboratory, 2010. 13

[66] Lin-Bo Zhang et al. A parallel algorithm for adaptive local refinement of tetrahedral meshes using bisection. *Numer. Math.: Theory, Methods and Applications*, 2:65–89, 2009. 13

[67] Lax PD and B. Wendroff. Systems of conservation laws. *Comm. Pure Appl. Math*, 13, 1960. 14, 17, 129

[68] Olivier Colin and Michael Rudgyard. Development of high-order taylor–galerkin schemes for les. *Journal of Computational Physics*, 162(2):338–371, 2000. 14, 17, 27, 72, 104, 121, 141, 160

[69] J von Neumann and RD Richtmyer. On the numerical solutions of partial differential equations of parabolic type. *John von Neumann Collected Works*, 5:652–663, 1947. 18

[70] Charles Hirsch. *Numerical Computation of Internal and External Flows. Vols. 1 and 2-Computational Methods for Inviscid and Viscous Flows*. 1990. 18

[71] Tapan K Sengupta. *High accuracy computing methods: fluid flows and wave phenomena*. Cambridge University Press, USA, 2013. 18, 22, 24, 25, 68, 69, 70

[72] David J Hill and Dale I Pullin. Hybrid tuned center-difference-weno method for large eddy simulations in the presence of strong shocks. *Journal of Computational Physics*, 194(2):435–450, 2004. 18

[73] Christopher K W Tam and Jay C Webb. Dispersion-relation-preserving finite difference schemes for computational acoustics. *Journal of computational physics*, 107(2):262–281, 1993. 19

[74] VK Suman, Tapan K Sengupta, C Jyothi Durga Prasad, K Surya Mohan, and Deepanshu Sanwalia. Spectral analysis of finite difference schemes for convection diffusion equation. *Computers & Fluids*, 150:95–114, 2017. 19, 20, 23, 24, 67, 70, 71

[75] Soumyo Sengupta, Tapan K Sengupta, Jyothi Kumar Puttam, and VK Suman. Global spectral analysis for convection-diffusion-reaction equation in one and two-dimensions: Effects of numerical anti-diffusion and dispersion. *Journal of Computational Physics*, page 109310, 2020. 19, 20

[76] Sten Rüdiger, Ernesto M Nicola, Jaume Casademunt, and Lorenz Kramer. Theory of pattern forming systems under traveling-wave forcing. *Physics Reports*, 447(3-6):73–111, 2007. 19

[77] Francesc Sagués, David G Míguez, Ernesto M Nicola, Alberto P Muñuzuri, Jaume Casademunt, and Lorenz Kramer. Travelling-stripe forcing of turing patterns. *Physica D: Nonlinear Phenomena*, 199(1-2):235–242, 2004. 19

[78] David B White. The planforms and onset of convection with a temperature-dependent viscosity. *Journal of Fluid Mechanics*, 191:247–286, 1988. 19

[79] Florin Ilinca and Dominique Pelletier. Positivity preservation and adaptive solution for the k-epsilon model of turbulence. *AIAA journal*, 36(1):44–50, 1998. 19, 66

[80] LC Lee, L Zhang, A Otto, GS Choe, and HJ Cai. Entropy antidiffusion instability and formation of a thin current sheet during geomagnetic substorms. *Journal of Geophysical Research: Space Physics*, 103(A12):29419–29428, 1998. 19

[81] Giorgos Kanellopoulos and Ko van der Weele. Critical flow and clustering in a model of granular transport: The interplay between drift and antidiffusion. *Physical Review E*, 85(6):061303, 2012. 20

[82] C Konstantopoulos, L Mittag, G Sandri, and R Beland. Deconvolution of gaussian filters and antidiffusion. *Journal of applied physics*, 68(4):1415–1420, 1990. 20

[83] Vincent Moureau, G Lartigue, Y Sommerer, Christian Angelberger, O Colin, and Thierry Poinsot. Numerical methods for unsteady compressible multi-component reacting flows on fixed and moving grids. *Journal of Computational Physics*, 202(2):710–736, 2005. 20

[84] Tapan K Sengupta and Ashish Bhole. Error dynamics of diffusion equation: Effects of numerical diffusion and dispersive diffusion. *Journal of Computational Physics*, 266:240–251, 2014. 24, 67, 70

[85] Charles Hirsch. *Numerical Computation of Internal External Flows: Fundamentals of Numerical Discretization*. John Wiley Sons, Inc., USA, 1988. 29

[86] VK Suman, Tapan K Sengupta, and JS Mathur. Effects of numerical anti-diffusion in closed unsteady flows governed by two-dimensional navier-stokes equation. *Computers & Fluids*, page 104479, 2020. 33, 56

[87] Tapan K Sengupta, Yogesh G Bhumkar, Manoj K Rajpoot, VK Suman, and Shakti Saurabh. Spurious waves in discrete computation of wave phenomena and flow problems. *Applied Mathematics and Computation*, 218(18):9035–9065, 2012. 37

[88] Tapan K Sengupta, Manoj K Rajpoot, Shakti Saurabh, and VVSN Vijay. Analysis of anisotropy of numerical wave solutions by high accuracy finite difference methods. *Journal of Computational Physics*, 230(1):27–60, 2011. 47

[89] Isaac Harari and Thomas JR Hughes. Finite element methods for the helmholtz equation in an exterior domain: model problems. *Computer methods in applied mechanics and engineering*, 87(1):59–96, 1991. 66

[90] T Nozakura and S Ikeuchi. Formation of dissipative structures in galaxies. *The Astrophysical Journal*, 279:40–52, 1984. 66

[91] Zhen Mei. *Numerical bifurcation analysis for reaction-diffusion equations*, volume 28. Springer Science & Business Media, 2000. 66

[92] Sten Rüdiger, Ernesto M Nicola, Jaume Casademunt, and Lorenz Kramer. Theory of pattern forming systems under traveling-wave forcing. *Physics Reports*, 447(3-6):73–111, 2007. 66

[93] Francesc Sagués, David G Míguez, Ernesto M Nicola, Alberto P Muñuzuri, Jaume Casademunt, and Lorenz Kramer. Travelling-stripe forcing of turing patterns. *Physica D: Nonlinear Phenomena*, 199(1-2):235–242, 2004. 66

[94] David B White. The planforms and onset of convection with a temperature-dependent viscosity. *Journal of Fluid Mechanics*, 191:247–286, 1988. 66

[95] Osamu Hirayama and Ryuji Takaki. Thermal convection of a fluid with temperature-dependent viscosity. *Fluid Dynamics Research*, 12(1):35, 1993. 67

[96] Michael Ardes, FH Busse, and Johannes Wicht. Thermal convection in rotating spherical shells. *Physics of the Earth and Planetary Interiors*, 99(1-2):55–67, 1997. 67

[97] JT Lir and TF Lin. Visualization of roll patterns in rayleigh–bénard convection of air in a rectangular shallow cavity. *International Journal of Heat and Mass Transfer*, 44(15):2889–2902, 2001. 67

[98] Fengqi Yi, Junjie Wei, and Junping Shi. Bifurcation and spatiotemporal patterns in a homogeneous diffusive predator–prey system. *Journal of Differential Equations*, 246(5):1944–1977, 2009. 67

[99] Martin Baurmann, Thilo Gross, and Ulrike Feudel. Instabilities in spatially extended predator–prey systems: Spatio-temporal patterns in the neighborhood of turing–hopf bifurcations. *Journal of Theoretical Biology*, 245(2):220–229, 2007. 67

[100] Brian J Rothschild and Jerald S Ault. Population-dynamic instability as a cause of patch structure. *Ecological Modelling*, 93(1-3):237–249, 1996. 67

[101] Thierry Poinsot and Denis Veynante. *Theoretical and numerical combustion*. RT Edwards, Inc., 2005. 67

[102] Diego A Garzón-Alvarado, CH Galeano, and JM Mantilla. Computational examples of reaction–convection–diffusion equations solution under the influence of fluid flow: First example. *Applied Mathematical Modelling*, 36(10):5029–5045, 2012. 67

[103] Charles D Pierce and Parviz Moin. Progress-variable approach for large-eddy simulation of non-premixed turbulent combustion. *Journal of fluid Mechanics*, 504:73–97, 2004. 67

[104] Anotida Madzvamuse. *A numerical approach to the study of spatial pattern formation*. PhD thesis, University of Oxford, 2000. 67

[105] Tony WH Sheu, SK Wang, and Reui-Kuo Lin. An implicit scheme for solving the convection–diffusion–reaction equation in two dimensions. *Journal of Computational Physics*, 164(1):123–142, 2000. 67

[106] Ramon Codina. Comparison of some finite element methods for solving the diffusion-convection-reaction equation. *Computer methods in applied mechanics and engineering*, 156(1-4):185–210, 1998. 67

[107] L Leboucher. Monotone scheme and boundary conditions for finite volume simulation of magnetohydrodynamic internal flows at high hartmann number. *Journal of Computational Physics*, 150(1):181–198, 1999. 67

[108] Vaibhav Joshi and Rajeev K Jaiman. A positivity preserving variational method for multi-dimensional convection–diffusion–reaction equation. *Journal of Computational Physics*, 339:247–284, 2017. 67

[109] Soumyo Sengupta, Tapan K Sengupta, Jyothi Kumar Puttam, and Keshava Suman Vajjala. Global spectral analysis for convection-diffusion-reaction equation in one and two-dimensions: Effects of numerical anti-diffusion and dispersion. *Journal of Computational Physics*, 408:109310, 2020. 67

[110] Michael Pfitzner. A New Analytic pdf for Simulations of Premixed Turbulent Combustion. *Flow, Turbulence and Combustion*, 106(4):1213–1239, 2021. 67, 85

[111] J. H. Ferziger and Tarek Echekki. A Simplified Reaction Rate Model and its Application to the Analysis of Premixed Flames. *Combustion Science and Technology*, 89(5-6):293–315, 1993. 85

[112] M. Pfitzner and P. Breda. An analytic probability density function for partially premixed flames with detailed chemistry. *Physics of Fluids*, 33(3):1–16, 2021. 85

[113] Thilo Schonfeld and Michael Rudgyard. Steady and unsteady flow simulations using the hybrid flow solver avbp. *AIAA journal*, 37(11):1378–1385, 1999. 89

[114] L. Selle, G. Lartigue, T. Poinsot, R. Koch, K. U. Schildmacher, W. Krebs, B. Prade, P. Kaufmann, and D. Veynante. Compressible large eddy simulation of turbulent combustion in complex geometry on unstructured meshes. *Combustion and Flame*, 137(4):489–505, 2004. 89

[115] T. Poinsot and SK Lele. Boundary conditions for direct simulations of compressible viscous flows. *Journal of computational physics*, 101(1):104–129, 1992. 90

[116] Heinz Pitsch. Large-eddy simulation of turbulent combustion. *Annu. Rev. Fluid Mech.*, 38:453–482, 2006. 99, 104

[117] Sergey B Dorofeev. Flame acceleration and explosion safety applications. *Proceedings of the Combustion Institute*, 33(2):2161–2175, 2011. 100

[118] Stéphane Popinet. A quadtree-adaptive multigrid solver for the Serre-Green-Naghdi equations. *Journal of Computational Physics*, 302:336–358, 2015. 100

[119] J Bell, A Almgren, V Beckner, M Day, M Lijewski, A Nonaka, and W Zhang. Boxlib user's guide. *github. com/BoxLib-Codes/BoxLib*, 2012. 100

[120] Weiqun Zhang, Ann Almgren, Vince Beckner, John Bell, Johannes Blaschke, Cy Chan, Marcus Day, Brian Friesen, Kevin Gott, Daniel Graves, et al. Amrex: a framework for block-structured adaptive mesh refinement. 2019. 100

[121] Benjamin S Kirk, John W Peterson, Roy H Stogner, and Graham F Carey. libmesh: a c++ library for parallel adaptive mesh refinement/coarsening simulations. *Engineering with Computers*, 22(3-4):237–254, 2006. 100

[122] Carsten Burstedde, Lucas C Wilcox, and Omar Ghattas. p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing*, 33(3):1103–1133, 2011. 100

[123] Tobias Weinzierl and Miriam Mehl. Peano—a traversal and storage scheme for octree-like adaptive cartesian multiscale grids. *SIAM Journal on Scientific Computing*, 33(5):2732–2760, 2011. 100

[124] Cecile Dobrzynski. *MMG3D: user guide*. PhD thesis, INRIA, 2012. 100, 137

[125] Pierre Benard, Guillaume Balarac, Vincent Moureau, Cecile Dobrzynski, Ghislain Lartigue, and Yves d'Angelo. Mesh adaptation for large-eddy simulations in complex geometries. *International Journal for Numerical Methods in Fluids*, 81(12):719–740, 2016. 101, 108, 117, 118, 119, 137

[126] Charles Dapogny, Cécile Dobrzynski, and Pascal Frey. Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. *Journal of Computational Physics*, 262:358–378, 2014. 102, 107, 109

[127] Guillaume Daviller, Maxence Brebion, Pradip Xavier, Gabriel Staffelbach, Jens-Dominik Müller, and Thierry Poinsot. A mesh adaptation strategy to predict pressure losses in les of swirled flows. *Flow, Turbulence and Combustion*, 99(1):93–118, 2017. 103

[128] Pierre Quillatre, Olivier Vermorel, Thierry Poinsot, and Philippe Ricoux. Large eddy simulation of vented deflagration. *Industrial & Engineering Chemistry Research*, 52(33):11414–11423, 2013. 103, 140, 159, 160

[129] Olivier Vermorel, Pierre Quillatre, and Thierry Poinsot. Les of explosions in venting chamber: A test case for premixed turbulent combustion models. *Combustion and Flame*, 183:207–223, 2017. 103, 136, 159, 160, 163, 164, 187

[130] Fabrice Charlette, Charles Meneveau, and Denis Veynante. A power-law flame wrinkling model for les of premixed turbulent combustion part ii: dynamic formulation. *Combustion and Flame*, 131(1-2):181–197, 2002. 104

[131] Fabrice Charlette, Charles Meneveau, and Denis Veynante. A power-law flame wrinkling model for les of premixed turbulent combustion part i: non-dynamic formulation and initial tests. *Combustion and Flame*, 131(1-2):159–180, 2002. 104, 141, 168

[132] Jean-Philippe Legier, Thierry Poinsot, and Denis Veynante. Dynamically thickened flame les model for premixed and non-premixed turbulent combustion. In *Proceedings of the Summer Program*, volume 2000, pages 157–168. Center for Turbulence Research Stanford, CA, 2000. 104

[133] Bastien Rochette, Eleonore Riber, Bénédicte Cuenot, and Olivier Vermorel. A generic and self-adapting method for flame detection and thickening in the thickened flame model. *Combustion and Flame*, 212:448–458, 2020. 104

[134] O Colin, Frédéric Ducros, D Veynante, and Thierry Poinsot. A thickened flame model for large eddy simulations of turbulent premixed combustion. *Physics of fluids*, 12(7):1843–1863, 2000. 104, 160, 168

[135] Guillaume Daviller, Maxence Brebion, Pradip Xavier, Gabriel Staffelbach, Jens-Dominik Müller, and Thierry Poinsot. A mesh adaptation strategy to predict pressure losses in les of swirled flows. *Flow, Turbulence and Combustion*, 99(1):93–118, 2017. 107

[136] Boris Delaunay et al. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1–2, 1934. 107

[137] Pavanakumar Mohanamuraly and Gabriel Staffelbach. Hardware locality-aware partitioning and dynamic load-balancing of unstructured meshes for large-scale scientific applications. In *Proceedings of the Platform for Advanced Scientific Computing Conference*, pages 1–10, 2020. 108

[138] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on scientific Computing*, 20(1):359–392, 1998. 110, 112, 113

[139] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975. 110, 114, 115

[140] K Anandhanarayanan, R Krishnamurthy, and Debasis Chakraborty. Development and validation of a grid-free viscous solver. *AIAA Journal*, 54(10):3312–3315, 2016. 110, 114

[141] Pavanakumar Mohanamuraly. Study on robustness of least-sqaures approximation in finite point method for transonic flows. In *Proceedings of Aeronautical Society of India*. AESI, 2012. 110, 114

[142] Erik G Boman, Ümit V Çatalyürek, Cédric Chevalier, and Karen D Devine. The zoltan and isorropia parallel toolkits for combinatorial scientific computing: Partitioning, ordering and coloring. *Scientific Programming*, 20(2):129–150, 2012. 110

[143] George Karypis and Vipin Kumar. Parallel multilevel series k-way partitioning scheme for irregular graphs. *Siam Review*, 41(2):278–300, 1999. 110

[144] François Broquedis, Jérôme Clet-Ortega, Stéphanie Moreaud, Nathalie Furmento, Brice Goglin, Guillaume Mercier, Samuel Thibault, and Raymond Namyst. hwloc: A generic framework for managing hardware affinities in hpc applications. In *2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing*, pages 180–186. IEEE, 2010. 110

[145] Diane Poirier, Steven Allmaras, Douglas McCarthy, Matthew Smith, and Francis Enomoto. The cgns system. In *29th AIAA, Fluid Dynamics Conference*, page 3007, 1998. 112

[146] Jens-Dominik Müller. Coarsening 3-d hybrid meshes for multigrid methods. In *In 9th Copper Mountain Multigrid Conference*. Citeseer, 1999. 112

[147] Vipin Kumar, Ananth Grama, Anshul Gupta, and George Karypis. *Introduction to parallel computing*, volume 110. Benjamin/Cummings Redwood City, CA, 1994. 112

[148] George Karypis and Vipin Kumar. Multilevelk-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed computing*, 48(1):96–129, 1998. 113

[149] Bruce Hendrickson, Robert W Leland, et al. A multi-level algorithm for partitioning graphs. *SC*, 95(28):1–14, 1995. 113

[150] Stephen T Barnard and Horst D Simon. Fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. *Concurrency: Practice and experience*, 6(2):101–117, 1994. 113

[151] Donald Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524, 1968. 117

[152] Frank Ham and Gianluca Iaccarino. Energy conservation in collocated discretization schemes on unstructured meshes. *Annual Research Briefs*, 2004:3–14, 2004. 118

[153] Romain Janodet, Geoffroy Vaudor, Ghislain Lartigue, Pierre Bénard, Vincent Moureau, and Renaud Mercier. An unstructured conservative level-set algorithm coupled with dynamic mesh adaptation for the computation of liquid-gas flows. In *29th European Conference on Liquid Atomization and Spray Systems (ILASS Europe)*, 2019. 119

[154] LSG Kovasznay. Hot-wire investigation of the wake behind cylinders at low reynolds numbers. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 198(1053):174–190, 1949. 122

[155] Benedetta Franzelli, Eleonore Riber, Laurent YM Gicquel, and Thierry Poinsot. Large eddy simulation of combustion instabilities in a lean partially premixed swirled flame. *Combustion and flame*, 159(2):621–637, 2012. 129

[156] B. Franzelli, E. Riber, M. Sanjosé, and T. Poinsot. A two-step chemical scheme for kerosene-air premixed flames. *Combustion and Flame*, 157(7):1364 – 1373, 2010. 129

[157] X. Guo, W. Schröder, and M. Meinke. Large-eddy simulations of film cooling flows. *Computers & Fluids*, 35(6):587–606, 2006. 136

[158] Matthew Emmett, Emmanuel Motheau, Weiqun Zhang, Michael Minion, and John B Bell. A fourth-order adaptive mesh refinement algorithm for the multicomponent, reacting compressible navier–stokes equations. *Combustion Theory and Modelling*, 23(4):592–625, 2019. 136

[159] Xinfeng Gao and Clinton Groth. Parallel adaptive mesh refinement scheme for three-dimensional turbulent non-premixed combustion. In *46th AIAA aerospace sciences meeting and exhibit*, page 1017, 2008. 136

[160] Xiaodong Cai, Jianhan Liang, Ralf Deiterding, Yonggang Che, and Zhiyong Lin. Adaptive mesh refinement based simulations of three-dimensional detonation combustion in supersonic combustible mixtures with a detailed reaction model. *international journal of hydrogen energy*, 41(4):3222–3239, 2016. 136

[161] Julien Leparoux, Renaud Mercier, Vincent Moureau, and Haris Musaefendic. Primary atomization simulation applied to a jet in crossflow aeronautical injector with dynamic mesh adaptation. In *14th Triennal International Conference on Liquid Atomization and Spray Systems*, 2018. 137

[162] LR Boeck, S Lapointe, J Melguizo-Gavilanes, and G Ciccarelli. Flame propagation across an obstacle: Oh-plif and 2-d simulations with detailed chemistry. *Proceedings of the Combustion Institute*, 36(2):2799–2806, 2017. 137, 138, 139, 140, 142, 148, 186

[163] AR Masri and JD Gounder. Turbulent spray flames of acetone and ethanol approaching extinction. *Combustion Science and Technology*, 182(4-6):702–715, 2010. 137

[164] AR Masri, A AlHarbi, S aa Meares, and SS Ibrahim. A comparative study of turbulent premixed flames propagating past repeated obstacles. *Industrial & engineering chemistry research*, 51(22):7690–7703, 2012. 137, 157, 160, 161, 164, 165, 166, 187

[165] T. Poinsot and S. K. Lele. Boundary conditions for direct simulations of compressible viscous flows. *J. Comp. Phys.*, 101(1):104–129, 1992. 140

[166] Pedro S Volpiani, Thomas Schmitt, Olivier Vermorel, Pierre Quillatre, and Denis Veynante. Large eddy simulation of explosion deflagrating flames using a dynamic wrinkling formulation. *Combustion and Flame*, 186:17–31, 2017. 140

[167] Salah S Ibrahim, Sreenivasa Rao Gubba, Assaad R Masri, and Weeratunge Malalasekera. Calculations of explosion deflagrating flames using a dynamic flame surface density model. *Journal of Loss Prevention in the Process Industries*, 22(3):258–264, 2009. 140

[168] Mateusz Zbikowski, Dmitriy Makarov, and Vladimir Molkov. Les model of large scale hydrogen–air planar detonations: Verification by the znd theory. *International Journal of Hydrogen Energy*, 33(18):4884–4892, 2008. 140

[169] Patrick Schmitt, Thierry Poinsot, Bruno Schuermans, and Klaus-Peter Geigle. Large-eddy simulation and experimental study of heat transfer, nitric oxide emissions and combustion instability in a swirled turbulent high-pressure burner. *Journal of fluid mechanics*, 570:17–46, 2007. 141

[170] Massimo Germano, Ugo Piomelli, Parviz Moin, and William H Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3(7):1760–1765, 1991. 141

[171] Valeria Di Sarli, Almerinda Di Benedetto, and Gennaro Russo. Using large eddy simulation for understanding vented gas explosions in the presence of obstacles. *Journal of hazardous materials*, 169(1-3):435–442, 2009. 149

[172] JE Kent, AR Masri, SH Starner, and SS Ibrahim. A new chamber to study premixed flame propagation past repeated obstacles. In *5th Asia-Pacific conference on combustion*, pages 17–20. The University of Adelaide Adelaide, Australia, 2005. 157

[173] Franck Nicoud and Frédéric Ducros. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. *Flow, turbulence and Combustion*, 62(3):183–200, 1999. 160

[174] Denis Veynante and Thierry Poinsot. Effects of pressure gradients on turbulent premixed flames. *Journal of Fluid Mechanics*, 353:83–114, 1997. 168