Agenzia nazionale per le nuove tecnologie, l'energia e lo sviluppo economico sostenibile

MINISTERO DELLO SVILUPPO ECONOMICO

Sviluppo di un solutore "multi-zona" a diversa risoluzione spaziale

G.Rossi, B.Favini

SAPIENZA
UNIVERSITÀ DI ROMA

Report RdS/PAR2013/293

SVILUPPO DI UN SOLUTORE “MULTI-ZONA” A DIVERSA RISOLUZIONE SPAZIALE

G. Rossi, B.Favini (Dipartimento di Ingegneria meccanica e Aeronautica, Università di Roma "Sapienza")

Settembre 2014

Report Ricerca di Sistema Elettrico
Accordo di Programma Ministero dello Sviluppo Economico – ENEA
Piano Annuale di Realizzazione 2013
Area: Produzione di energia elettrica e protezione dell’ambiente
Progetto: Cattura e sequestro della CO_2 prodotta dall’utilizzo di combustibili fossili
Obiettivo: Parte A - b - Te cnol ogi e pe r l ’otti mi zza zi one de i proces s i di combus tione
Task b.1 – Metodologie numeriche avanzate per la simulazione dei processi di combustione e la progettazione di componenti
Responsabile del Progetto: Ing. Stefano Giammartini, ENEA

Il presente documento descrive le attività di ricerca svolte all’interno dell’Accordo di collaborazione “*Sviluppo di tecniche e modelli numerici per LES nel codice HeaRT per la simulazione di processi di combustione”*
Responsabile scientifico ENEA: Franca Rita Picchia
Responsabile scientifico Università di Roma “Sapienza”: Bernardo Favini

Contents

Summary **4**

1 Introduction **5**

2 HeaRT code overview **8**

2.1 Conservation Laws . . . 8
2.2 Multifluids Model . . . 9
2.3 Constitutive Equations . . . 9
2.3.1 The Diffusive Momentum Flux . . . 9
2.3.2 The Diffusive Species Mass Flux . . . 10
2.3.3 The Diffusive Heat Flux . . . 11
2.4 LES Model . . . 12
2.4.1 Filtered Conservation Equations . . . 13
2.4.2 Subgrid-scale Modelling . . . 14
2.5 Numerical Schemes . . . 15
2.5.1 AUSM Numerical Scheme . . . 15
2.5.2 Finite Difference Numerical Scheme . . . 18
2.5.3 Newton-Raphson Method . . . 19
2.5.4 Runge-Kutta Method . . . 20
2.6 HeaRT Code . . . 21

3 Mesh Refinement **25**

3.0.1 Solution Algorithm . . . 25
3.0.2 Communication Procedures . . . 26

4 Validation **32**

4.1 Parabolic Flux Test Case . . . 32
4.2 LES of a Lean Premixed CH_4/H_2-Air Slot Flame . . . 32

Bibliography **38**

Sommario

Nell'ambito delle simulazioni numeriche di flussi turbolenti, la tecnica Large Eddy Simulation (LES) sta acquisendo sempre più importanza, data la grande accuratezza dei risultati ottenuti con questo approccio e soprattutto considerata la sua abilità di predirre, con elevata affidabilità, il comportamento di campi di moto molto complessi.

Per ottenere risultati ancora più precisi senza dover aumentare in maniera considerevole i tempi di calcolo richiesti, è stata sviluppata, all'interno del codice HeaRT, una tecnica di mesh refinement che prevede la presenza di griglie a diversa risoluzione nel dominio di calcolo: queste griglie integrano in maniera indipendente le equazioni di conservazione e comunicano fra loro, quando hanno raggiunto lo stesso istante temporale, sui nodi di "esterni" (le celle "ghost", necessarie per l'integrazione delle equazioni di conservazione nei punti di zona prossimi al contorno della zona stessa).

Per trasferire i valori da una griglia più lasca ad una più fitta è usata un'interpolazione mediante la tecnica dei minimi quadrati, al terzo ordine, che garantisce un'adeguata accuratezza della soluzione sulla griglia raffinata: note le coordinate del punto da interpolare, si costruisce l'interpolatore selezionando tutti i punti della griglia rada che "circondano" il punto fitto nelle tre direzioni; l'interpolazione, inoltre, è effettuata imponendo la conservazione "globale" della grandezza, garantendo cioè che la somma dei valori fitti interpolati (pesati per la frazione di volume della cella fitta rispetto a quella rada) sia pari al valore della cella rada alla quale le celle fitte considerate appartengono.

Per comunicare i valori nel verso opposto (dalla griglia più fitta ad una più lasca), vengono sommati i valori corrispondenti alle celle fitte che sono contenute all'interno di una cella rada, opportunamente pesati mediante la frazione di volume fitto rispetto al volume rado.

La tecnica, dopo un'accurata validazione, è utilizzata nella simulazione di una fiamma premiscelata H_2/CH_4 – Aria, in cui la zona centrale, dove avvengono il mescolamento e le reazioni chimiche, è a risoluzione più alta, mentre le zone ove il campo presenta gradienti meno marcati sono a risoluzione più bassa.

1 Introduction

Turbulent flow field is very common in energy production burners: some particular phenomena that occur in this plants are heavily conditioned by interaction between turbulent structures and burner walls, injectors, or the flame front.

In particular, to avoid combustion instabilities and blow out phenomena, swirlers and bluff bodies are widely used: in this way oxidizer and fuel (or mixture) flows have all the three components of velocity vector different from zero; for this reason is obvious that turbulence of the flow field has a fundamental role in energy production burners behaviour.

Usually in energy production burners, inlet flow (premiscelated or not) in combustion chamber has high velocity and injectors have very little dimensions: so a numerical simulation based on DNS (Direct Numerical Solution) approach is impossible; a numerical simulation based on RANS (Reynolds Averaged Navier-Stokes) approach is very difficult to obtain because of the complexity to obtain a turbulence model suitable to the solution of such flow field (thermal and chemical non equilibrium, with complex geometries).

So numerical simulation based on LES (Large Eddy Simulation) approach is growing in importance because when a sufficient portion of the energy spectrum is resolved, is possible to obtain very good numerical results, also if computational cost is anyway high (but no so high as in DNS approach), because grid cell dimension is necessarily little to obtain a good solution of significant turbulent scales.

Modern energy production burners have usually big dimensions and a cylindrical shape, with multi-phase flow: also if a LES numerical simulation for such burners is the ideal choice, there are some problems to obtain a good numerical result: in particular, is necessary a large number of grid points to guarantee sufficient resolution to solve little turbulent scales next to the little injectors; multi-phase flow is another constraint on the dimension of grid cell, because solid particles or droplets cannot be bigger than a single grid cell.

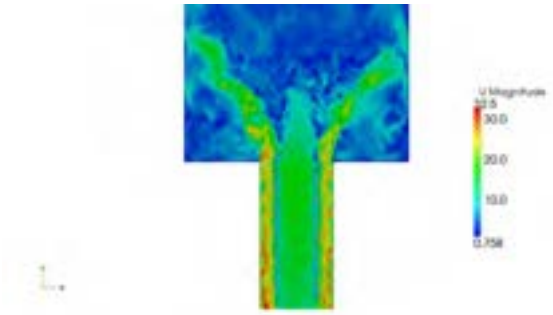
For example, in figure 1.1 are illustrated numerical results of a LES simulation of a particle laden flow[1], based on Sommerfeld and Qiu experiment: the injection system is composed of a cylindrical duct and of an annular duct coaxial to the first one. Air and glass particles (diameter between $20\mu m$ and $80\mu m$) flow through the cylindrical duct to the test cylindrical chamber, while from the annular duct a swirled air flow is introduced in the test section.

Also if the numerical solution globally has a good agreement with experimental data, and the presence of many small flow structures is captured as the position of the stagnation point, typical for swirled combustors, in proximity of the test chamber inlet zone radial velocity component and its RMS are not well predicted (figure 1.2): this can be dued to the poor resolution adopted in front of the bluff body separating the inner and the outer duct; in fact, since $\Delta z = 3mm$, only 3 grid points are present before the plane where the measures are taken and therefore it will not be possible to reconstruct smaller structures in that zone.

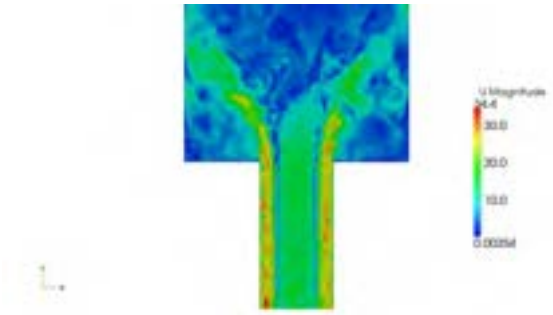
Total number of grid points for this numerical simulation (more than 4 millions) makes impossible to decrease Δz minimum and increase resolution, also because of the use of structured grids.

So a mesh refinement technique is mandatory to obtain a good numerical solution (with a relatively low computational cost) based on LES approach: numerical grid is divided in several structured grids with different spatial resolution: next to fuel and oxidizer injectors, where turbulent scales have a very little characteristic dimension, grid cells have very little dimensions, while in the outlet zone of the burner or the test section, where flow velocities and flow turbulence are low, grid cells have bigger dimensions.

Multigrid approach is used for the solution of hyperbolic or elliptic systems of equations: with this technique is possible to obtain a remarkable increase in the convergence speed of the numerical simulation; in a multigrid approach, iterative procedures succeed to shrink the *residuals* obtained from the approximate solution of

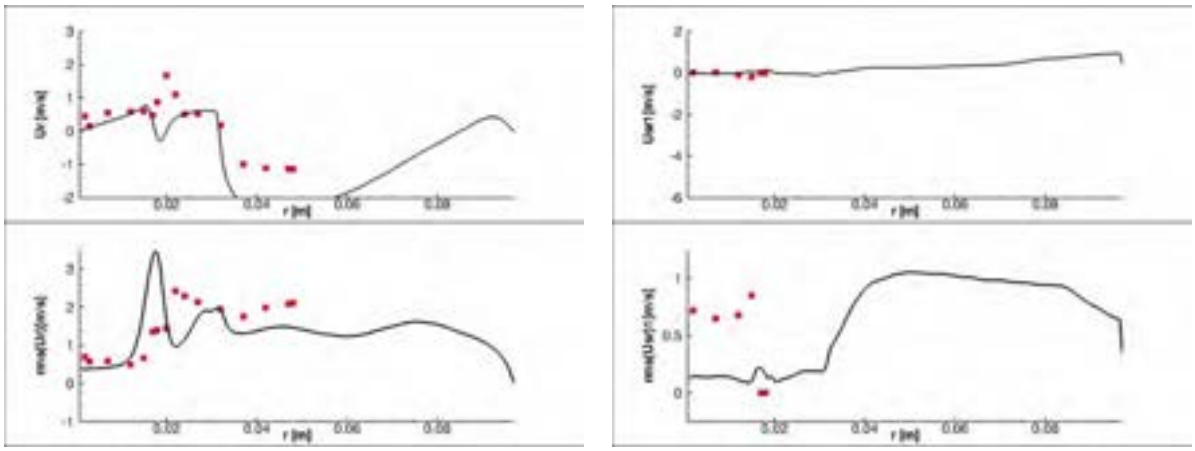


(a)



(b)

Figure 1.1: Velocity Magnitude [m/s]



(a) (b)

Figure 1.2: Velocity Component [m/s] and RMS along r direction

governing equations ([2]): this method isn't suitable for HeaRT code, because in this numerical code time integration is performed by the mean of an explicit scheme.

When there aren't chemical reactions, multi-level techniques are frequently used in LES simulations ([3], [4]): the main part of the simulation is performed on the coarse grid levels, while the smallest resolved scales are kept frozen (quasi-static approximation); this allows to reduce significantly the CPU times in comparison with classical LES, while the accuracy of the simulation is preserved by the use of a fine discretization level, where the coarse flow field is interpolated.

When non-uniform grids are used, the best achievable result is a better representation of the mean velocity profiles: to improve the numerical solution quality, overlapping finest grids are necessary ([5]). Gird interaction is obtained by the mean of boundary conditions enforcement from the coarse grid to the finest grid, that has to be repeated every time step.

For incompressible flows, grid coupling can be obtained by means of Poisson equation for the pressure ([6], [7]): every grid solves momentum conservation equations and the additional "information" due to the better spatial resolution of the finest grid is included in the source term of the Poisson equation. Fourier polynomials are used to transfer informations, every time step, form different gird levels.

In an hyperbolic system of equations is possible to implement an adaptive mesh refinement (AMR) technique ([8]): by the means of an error evaluation procedure (where the error is calculated as the difference between the numerical solution obtained in a time step $2t$ and the numerical solution obtained in two time steps t), zones where mesh refinement is necessary are located. Grid level communication is carried out with bilinear interpolations (from coarse to fine) and with weighted sums (from fine to coarse).

2 HeaRT code overview

2.1 Conservation Laws

The governing equations are written in the form of partial differential equations. This form of the governing equations derives directly from the physical principles of flow dynamics:

- conservation of mass;
- conservation of Linear Momentum;
- conservation of energy.

The conservation laws written in this form relate the flow field variables at a point in the flow, as opposed to governing equations written in integral form that deal with a finite space, the control volume.

Conservation of Mass

The conservation of mass principle states that, in absence of mass adduction, the variation in time of the density of the flow field in a point is equal to the variation in space of the product between the density and the velocity vector, that is:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \tag{2.1}$$

where ρ is the density of the fluid.

Conservation of Momentum

The conservation of momentum principle states that in absence of external forces the rate of change of linear momentum in a point in the space is equal to the resultant forces on that point, that is:

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) = -\nabla \cdot \underline{\mathbf{S}} + \rho \sum_{i=1}^{N_s} Y_i \mathbf{f_i} \tag{2.2}$$

where $\underline{\mathbf{S}}$ is the stress tensor, N_s is the number of chemical species, Y_i is the mass fraction of the ith species and f_i is the body force per mass unit acting on the ith species.

Conservation of Energy

The conservation of energy principle states that in absence of energy adduction the rate of change of energy in a point in the space is equal to the heat transfer rate and the total work made by the forces acting on that point, that is:

$$\frac{\partial E}{\partial t} + \nabla \cdot (\rho \mathbf{u} E) = \nabla \cdot (\underline{\mathbf{S}}\mathbf{u}) - \nabla \cdot \mathbf{q} + \rho \sum_{i=1}^{N_s} Y_i \mathbf{f_i} \cdot (\mathbf{u} + \mathbf{V}_i) \tag{2.3}$$

where E is the total energy (internal and kinetic), $\mathbf{q}$ is the heat transfer rate and $\mathbf{V}_i$ is the diffusion velocity of the ith chemical species.

2.2 Multifluids Model

In a chemical reacting and multispecies flow, an additional equation is necessary to complete the flow description: the conservation of species mass fraction; this principle states that the rate of accumulus of the ith species depends on the convective flux, the diffusive flux and the production (or destruction) rate of the species due to the chemical reactions, that is:

$$\frac{\partial \rho Y_i}{\partial t} + \nabla \cdot (\rho \mathbf{u} Y_i) = -\nabla \cdot \mathbf{J}_i + \rho \dot{\omega}_i \tag{2.4}$$

where $\mathbf{J}_i$ is the diffusive mass flux and $\dot{\omega}_i$ is the production – destruction rate of the ith species.

The last equation used is the thermodynamic state equation that states a relation between the pressure, the density and the temperature of an ideal gas, that is:

$$p = \rho \sum_{i=1}^{N_s} \frac{Y_i}{W_i} \mathcal{R}_u T \tag{2.5}$$

where W_i is the molecular weight of the ith species, $\mathcal{R}_u$ is the universal gas constant, T is the temperature and p is the pressure of the fluid.

The summation of all species transport equations (2.4) yields the conservation of mass equation (2.1): therefore the N_s species mass fraction conservation and mass transport equation are linearly dependent and one of them is redundant. Furthermore, to be consistent with mass conservation, the diffusion fluxes ($\mathbf{J}_i = \rho Y_i \mathbf{V}_i$) and the chemical source terms must satisfy:

$$\sum_{i=1}^{N_s} \mathbf{J}_i = 0 \quad \text{and} \quad \sum_{i=1}^{N_s} \dot{\omega}_i = 0 \tag{2.6}$$

2.3 Constitutive Equations

Each material has a different response to an external force, depending on the properties of the material itself: the constitutive equations describe this behaviour.

In particular, they express with simple mathematical models the microscopic molecular diffusion of momentum, energy and mass. For a gas mixture, they should model the stress – strain relation between $\mathbf{S}$ and $\mathbf{E}$, the heat flux $\mathbf{q}$ and the species mass flux $\mathbf{J}_i$.

2.3.1 The Diffusive Momentum Flux

For all gases that can be treated as continuum it has been observed a linear proportion between viscous stresses that arise from the flow and the local strain rate: that is equivalent to saying that those forces are proportional to the rates of change of the fluid's velocity vector as one moves away from the point in question in various directions; a fluid that behaves in this manner is called Newtonian fluid. With this assumption is possible to derive an expression that relates the stress tensor $\mathbf{S}$ to the pressure p, the strain rate $\mathbf{E}$ and the velocity vector $\mathbf{u}$, that is:

$$\mathbf{S} = (-\mathbf{p} + \boldsymbol{\lambda} \boldsymbol{\nabla} \cdot \mathbf{u})\, \mathbf{I} + 2\boldsymbol{\mu} \mathbf{E} = -\mathbf{p}\mathbf{I} + \boldsymbol{\tau} \tag{2.7}$$

where τ is the viscous part of stress tensor, μ is the coefficient of viscosity and λ is the second coefficient of viscosity. These two coefficient of viscosity are related to the coefficient of bulk viscosity μ_b by the $\mu_b = 2/3\mu + \lambda$; with the assumption of the Stoke hypothesis (the sum of the normal stresses is zero), $\mu_b = 0$, so $\lambda = 2/3\mu$ and the viscous stress tensor becomes:

$$\tau_{ij} = \lambda \frac{\partial u_i}{\partial x_i} + 2\mu \left[\frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right] \tag{2.8}$$

Pressure at the macroscopic level corresponds to the microscopic transport of momentum by means of molecular collisions in the direction of molecules motion. Instead, molecular momentum transport in other directions is what at macroscopic level is called viscosity. They are of different nature, because in terms of work done, when continuous distribution are considered, pressure produces reversible transformations (changes of volume), while viscous stresses produce irreversible transformations (dissipation of energy into heat).

2.3.2 The Diffusive Species Mass Flux

In equation (2.4) the knowledge of diffusive species mass flux $\mathbf{J}_i$ is required: this flux expresses the relative motion of chemical species with respect to the motion of their (moving) center of mass; by the means of a constitutive law this motion can be expressed without an additional momentum equation for chemical species. Both modelling and calculation of individual species diffusive mass flux is not easy; the distribution of N_s chemical species in a multicomponent gaseous mixture is rigorously obtained by the means of kinetic theory [9]:

$$\nabla X_i = \sum_{i=1, i\neq j}^{N_s} \underbrace{\frac{X_i X_j}{D_{ij}} \left(\mathbf{V}_j - \mathbf{V}_i\right)}_{\text{DV}} + \underbrace{\left(Y_i - X_i\right) \frac{\nabla p}{p}}_{\text{PG}} + \underbrace{\frac{\rho}{p} \sum_{j=1}^{N_s} Y_i Y_j \left(\mathbf{f}_i - \mathbf{f}_j\right)}_{\text{BF}} + \underbrace{\sum_{j=1}^{N_s} \frac{X_i X_j}{\rho D_{ij}} \left(\frac{\alpha_j}{Y_j} - \frac{\alpha_i}{Y_i}\right) \frac{\nabla T}{T}}_{\text{SG}} \tag{2.9}$$

where D_{ij} is the binary diffusion coefficient of species i into the species j, X_j and Y_j are the molar and the mass fraction of the jth species respectively, $\mathbf{f}_j$ the body force per unit mass, acting on species j, α_j the thermodiffusion coefficient of species j.

Equations (2.9) are referred to as the Maxwell-Stefan equations, since Maxwell [10, 11] suggested them for binary mixtures on the basis of kinetic theory, and Stefan [12] generalized them to describe the diffusion in a gas mixture with N_s species.

The main feature of equations (2.9) is that they couple inextricably all diffusion velocities $\mathbf{V}_j$ (and thus all fluxes), to all concentrations X_j and Y_j and their gradients. According to equations 2.9, concentrations gradients can be physically created by:

- differences in Diffusion Velocities (DV);
- Pressure Gradients (PG);
- differences in Body Forces (BF) per unit mass acting on molecules of different species;
- thermo-diffusion, or Soret Effect (SE), i.e., mass diffusion due to temperature gradients, driving light species towards hot regions of the flow; this last effect, often neglected, is nevertheless known to be important, in particular for hydrogen combustion, and in general when very light species play an important role.

The linear system of equations (2.9) for the variable $\mathbf{V}_j$ has size $N_s \times N_s$ and requires knowledge of $Ns(Ns-1)/2$ diffusivities. Only $N_s - 1$ equations are independent, since the sum of all diffusion fluxes must be zero. This system must be solved in each direction of the frame of reference (coordinate system), at every computational node and, for unsteady flows, at each time step of numerical integration.

Extracting the diffusion velocities is a very difficult task, therefore in many CFD computations simplified models are preferred; in the present dissertation, Hirschfelder and Curtiss law is used to calculate diffusion velocities in an approximate way:

$$\mathbf{V}_i = -D_i \frac{\nabla X_i}{X_i} \tag{2.10}$$

where

$$D_i = \frac{1 - Y_i}{\sum_{j=1, j\neq i}^{N_s} \frac{X_j}{D_{ji}}} \tag{2.11}$$

The coefficient D_i is an equivalent diffusion coefficient of species i into the whole remaining mixture of $N_s - 1$ species.

When approximated expressions (like Hirschfelder and Curtiss law) are used to obtain diffusion velocities, mass conservation problems can occur: in fact the diffusion velocities do not necessarily satisfy the constrain

$$\sum_{i=1}^{N_s} \mathbf{J}_i = \sum_{i=1}^{N_s} \rho Y_i \mathbf{V}_i = 0$$

A simple empirical remedy to impose global mass conservation consists in subtracting any residual artificial diffusional velocity from the flow velocity in the species transport equations. In fact, summing all species transport equations, the mass conservation equation must be obtained, while it is found:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = -\nabla \cdot (\rho \sum_{i=1}^{N_s} Y_i \mathbf{V}_i) \tag{2.12}$$

Thus, in order for the conservation of mass to be respected, a term $\rho_f \mathbf{V}^c$ involving a correction velocity $\mathbf{V}^c$ must be introduced. $\mathbf{V}^c$ is defined as

$$\mathbf{V}^c = -\sum_{i=1}^{N_s} Y_i \mathbf{V}_i$$

and assuming Hirschfelder's law holds, it becomes

$$\mathbf{V}^c = \sum_{i=1}^{N_s} \frac{W_i}{W_{mix}} D_i \nabla X_i \tag{2.13}$$

The correction velocity must be computed at each time step and added to the flow velocity in the species convective term. The corrected convective term of species transport equations must then become

$$\nabla(\rho_f \mathbf{u}_f Y_i) \rightarrow \nabla(\rho_f (\mathbf{u}_f + \mathbf{V}^c) Y_i) \tag{2.14}$$

With this "trick", any artificial flow due to the nonzero diffusional mass flux is thereby cancelled. Furthermore, the constrain $\sum_{i=1}^{N_s} Y_i = 1$ is strictly ensured by the

$$Y_i = \frac{Y_i}{\sum_{i=1}^{N_s} Y_i} (1 - Y_i) + Y_i \tag{2.15}$$

2.3.3 The Diffusive Heat Flux

The heat flux $\mathbf{q}$ for a gaseous mixture of N_s chemical species can be splitted in three contributions:

1. the heat transfered by conduction, modeled by the Fourier's law; at a microscopic level, this flux is dued to molecular collision: molecules with an higher level of kinetic energy (with higher temperature) transfer energy by molecular collision to the ones with a lower level of kinetic energy (with a lower temperature); in the continuum view, the heat flux arises from temperature gradients;

2. the heat transfered by molecular diffusion in a multi–species mixture, dued to concentration gradients; when $\nabla Y_i \neq 0$, each species diffuses with its own velocity V_i; this means that can be energy transfer even in a gas at uniform temperature or in a rarefied gas (where conduction is negligible);

3. the heat transfered by Dufour effect: the Onsager principle of microscopic reversibility in the thermodynamics of irreversible processes implies that if temperature gradients cause species diffusion (thermodiffusive or Soret effect), concentration gradients must cause a reciprocal (Dufour effect) heat flux; the Dufour effect is neglected [9].

The total heat flux $\mathbf{q} = \mathbf{q}_F + \mathbf{q}_{V_i} + \mathbf{q}_D$, where $\mathbf{q}_D$ is the thermal flux dued to Dufour effect (neglected) is:

$$\mathbf{q} = -k\nabla T + \rho \sum_{i=1}^{N_s} \mathcal{H}_i Y_i \mathbf{V}_i \tag{2.16}$$

where k is the thermal conductivity and $\mathcal{H}_i$ is the enthalpy of the ith species.

2.4 LES Model

Large eddy simulations are based on the assumption that small-scale turbulent eddies are more isotropic than the large ones, and are responsible mostly for energy dissipation in the mean. Modelling the small scales, while resolving the large eddies, may be very beneficial: first, since most of the momentum transport is due to the large eddies, model inaccuracies are less critical; secondly, the modelling of the unresolved scales is easier, since they tend to be more homogeneous and isotropic than the large ones, which depend on the boundary conditions.

Thus, LES is based on the use of a filtering operation: a filtered (or resolved) large-scale variable is defined by:

$$\bar{f}(\mathbf{x}) = \int_D f(\mathbf{x}') \, G(\mathbf{x}, \mathbf{x}', \overline{\Delta}) \, d\mathbf{x}' \tag{2.17}$$

where D is the entire domain and G is the filter function. The size of the smallest eddies that are resolved in LES is clearly related to the filter width, denoted by $\overline{\Delta}$: the grid spacing h should be sufficiently fine to represent accurately eddies of size $\overline{\Delta}$. There has been considerable discussion on the appropriate value of filter width; in most cases, the filter width is chosen proportional to the grid size ($\overline{\Delta} = nh$), but if the mesh is anisotropic, an appropriate average is used to determine h:

$$h = (h_x h_y h_z)^{1/3} \quad \text{or} \quad h = \left(h_x^2 + h_y^2 + h_z^2\right) \tag{2.18}$$

When complex geometries are studied, is preferable relating the filter width to physical quantities, rather than the grid. The most commonly used filter function are the sharp Fourier cutoff filter, defined in the wave space, that is:

$$\widehat{G}(k) = \begin{cases} 1, & \text{if } k \leqslant \pi/\overline{\Delta} \\ 0, & \text{otherwise} \end{cases} \tag{2.19}$$

the Gaussian filter in the physical space, that is:

$$G(\mathbf{x}) = \sqrt{\frac{6}{\pi \overline{\Delta}^2}} \exp\left(-\frac{6\mathbf{x}^2}{\overline{\Delta}^2}\right) \tag{2.20}$$

and the tophat filter in physical space, that is

$$G(\mathbf{x}) = \begin{cases} 1/\overline{\Delta}, & \text{if } |\mathbf{x}| \leqslant \overline{\Delta}/2 \\ 0, & \text{otherwise} \end{cases} \tag{2.21}$$

In this work, the filter operation is implicitly defined by the mesh size. The uncertainties related to the procedure of exchanging the order of the filter and differential operators (commutation errors), are neglected and assumed to be incorporated in the sub-grid scale modeling.

2.4.1 Filtered Conservation Equations

In compressible flows, it is convenient to use Favre-filtering to avoid the introduction of subgrid-scale terms in the equation of conservation of mass: when the mass balance equation is averaged (with this operation a generic variable ϕ is splitted into a mean value $\overline{\phi}$ and a deviation from the mean denoted by ϕ'), indeed, velocity/density fluctuation correlation appears; to avoid the explicit modelling of such correlation, a Favre average is introduced: a Favre-filtered variable is defined by the

$$\widetilde{f} = \frac{\overline{\rho f}}{\overline{\rho}} \tag{2.22}$$

The Favre-filtered equations of motion can be written in the form:

$$\frac{\partial \overline{\rho}}{\partial t} + \frac{\partial \left(\overline{\rho}\widetilde{u}_j\right)}{\partial x_j} = 0 \tag{2.23}$$

$$\frac{\partial \left(\overline{\rho}\widetilde{u}_j\right)}{\partial t} + \frac{\partial \left(\overline{\rho}\widetilde{u}_i\widetilde{u}_j\right)}{\partial x_j} = -\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial \overline{\tau}_{ji}}{\partial x_j} - \frac{\partial \tau_{ji}^{SGS}}{\partial x_j} + \overline{\rho}\sum_{i=1}^{N_s}\widetilde{Y_i \mathbf{f}_i} \tag{2.24}$$

$$\frac{\partial \left(\overline{\rho}\widetilde{\mathcal{H}}\right)}{\partial t} + \frac{\partial \overline{\rho}\widetilde{\mathcal{H}}\widetilde{u}}{\partial x_j} = \frac{D\overline{p}}{Dt} - \frac{\partial \left(\overline{q}_j + q_j^{SGS}\right)}{\partial x_j} + \overline{\rho}\sum_{i=1}^{N_s}\widetilde{Y}_i\widetilde{f}_{ij}\widetilde{V}_{ij} \tag{2.25}$$

where

$$\overline{\tau_{ij}} = \overline{\mu}\left(\frac{\partial \widetilde{u}_i}{\partial x_j} + \frac{\partial \widetilde{u}_j}{\partial x_i}\right) - \frac{2}{3}\overline{\mu}\delta_{ij}\frac{\partial \widetilde{u}_k}{\partial k}$$

and

$$\tau_{ij}^{SGS} = \overline{\rho}\widetilde{u_i u_j} - \overline{\rho}\widetilde{u}_i\widetilde{u}_j \tag{2.26}$$

$$q_j^{SGS} = \overline{\rho}\widetilde{\mathcal{H}u_j} - \overline{\rho}\widetilde{\mathcal{H}}\widetilde{u}_j \tag{2.27}$$

The subgrid-scale heat flux due to diffusion effects has been considered negligible with respect to SGS heat flux q_{ij}^{SGS}.

The Favre-filtered equations of species and state can be written in the form:

$$\frac{\partial \left(\overline{rho}\widetilde{Y}_i\right)}{\partial t} + \frac{\partial \left(\overline{\rho\widetilde{Y}_i\widetilde{V}_{ij}}\right)}{\partial x_j} = -\frac{\partial \left(\overline{\rho}\widetilde{Y}_i\widetilde{V}_{ij}\right)}{\partial +}\overline{\rho}\widetilde{\omega} - \frac{\partial J_{ij}^{SGS}}{\partial x_j} \tag{2.28}$$

$$\overline{p} = \overline{\rho}\sum_{i=1}^{N_s}\mathcal{R}_u\frac{\widetilde{Y_i}}{W_i}T \tag{2.29}$$

where

$$J_{ij}^{SGS} = \overline{\rho}\widetilde{Y_i u_j} - \overline{\rho\widetilde{Y}_i\widetilde{u}_j} \tag{2.30}$$

Also in this case, the subgrid-scale effects due to diffusion, arising from $\widetilde{J}_{ij}$, may be neglected with respect to SGS species transport $\mathbf{J}_i^{SGS}$.

2.4.2 Subgrid-scale Modelling

The main role of subgrid-scale model must be to remove energy from the resolved scales, mimicking the drain that is usually associated with the energy cascade. It may not be necessary for a model to represent the "exact" SGS stresses accurately at each point in space and time, but only to account for their global effect; the resolved scales exchange energy with the unresolved scales and the surroundings through several mechanisms.

The Smagorinsky model is the progenitor if most subgrid-scale stress models: the eddy viscosity (that has to dissipate energy from the resolved scales) is assumed to be proportional to the subgrid-scale characteristic length scale $\overline{\Delta}$ and to a characteristic turbulent velocity $v_{\overline{\Delta}} = \overline{\Delta}\left|\overline{S}\right|$, that is:

$$\mu_T = \left(C_s\overline{\Delta}\right)^2 \left|\overline{S}\right| \tag{2.31}$$

where the Smagorinsky constant C_s is real, so the model is absolutely dissipative: $\epsilon_{SGS} = -\left(C_s\overline{\Delta}\right)^2 \left|\overline{S}\right|^3 \leqslant 0$, and $\left|\overline{S}\right| = \sqrt{2\overline{S}_{ij}\overline{S}_{ij}}$ is the magnitude of the strain-rate tensor. The value of C_s can be evaluated by the:

$$C_s \simeq \frac{1}{\pi}\left(\frac{2}{3C_K}\right)^{3/4} \tag{2.32}$$

that for a Kolmogorov constant $C_K \simeq 1.4$ yields $C_s \simeq 0.18$. For a lower value of Smagorinsky constant ($C_s = 0.1$) this model behaves very well for isotropic turbulence, free-shear flows and channel flow (with a damping function at the wall): near solid boundaries, indeed, Smagorinsky model is too dissipative and therefore the lenghtscale needs to be modified by the introduction of Van Driest damping, to account the reduced growth of the small scales near the wall; the eddy viscosity can be rewritten as:

$$\mu_T = \left[C_s\overline{\Delta}\left(1 - e^{-y^+/25}\right)\right]\left|\overline{S}\right| \tag{2.33}$$

Smagorinsky model is often used as a "base" for dynamic models, where coefficients are computed while the computation progress, rather than input a priori. This is accomplished by defining a test filter (for the computed field) whose width $\widehat{\Delta}$ is larger than the grid filter $\overline{\Delta}$: $\widehat{\Delta} = \alpha\overline{\Delta}$, with tipically $\alpha = 2$.

The subgrid-scale tensor of the double filtered field $\widetilde{\overline{u}}$ is obtained from equation (2.26)

$$\widehat{\tau}_{ij}^{SGS} = \widetilde{\overline{u_i u_j}} - \widetilde{\overline{u}}_i\widetilde{\overline{u}}_j \tag{2.34}$$

The resolved turbulent stress corresponding to the test filter applied to the field $\overline{u}$ is:

$$\mathcal{L}_{ij} = \widetilde{\overline{u}_i\overline{u}_j} - \widetilde{\overline{u}}_i\widetilde{\overline{u}}_j \tag{2.35}$$

Finally, applying the filter "tilde" to equation (2.26):

$$\widetilde{\tau}_{ij}^{SGS} = \widetilde{\overline{u_i u_j}} - \widetilde{\overline{u}_i\overline{u}_j} \tag{2.36}$$

Adding equations (2.34) and (2.36) Germano's identity can be obtained:

$$\mathcal{L}_{ij} = \widehat{\tau}_{ij}^{SGS} - \widetilde{\tau}_{ij}^{SGS} \tag{2.37}$$

In this equation $\widehat{\tau}_{ij}^{SGS}$ and $\widetilde{\tau}_{ij}^{SGS}$ have to be modelled while $\mathcal{L}_{ij}$ can be explicitly calculated by applying the test filter to the LES results.

Assuming that an eddy-viscosity model is used to parametrize both subgrid and subtest stresses:

$$\widetilde{\tau}_{ij}^{SGS} - \frac{1}{3}\widetilde{\tau}_{ll}^{SGS}\delta_{ij} = 2\widetilde{\mathcal{A}_{ij}C} \quad \text{and} \quad \widehat{\tau}_{ij}^{SGS} - \frac{1}{3}\widehat{\tau}_{ll}^{SGS}\delta_{ij} = 2\mathcal{B}_{ij}C \tag{2.38}$$

Substituing equation (2.38) in equation (2.37):

$$\mathcal{L}_{ij} - \frac{1}{3}\mathcal{L}_{ll}\delta ij = 2\mathcal{B}_{ij}C - 2\widetilde{\mathcal{A}_{ij}C} \tag{2.39}$$

In order to obtain C, it can be removed from the filtering as if it were constant, leading to:

$$\mathcal{L}_{ij} - \frac{1}{3}\mathcal{L}_{ll}\delta ij = 2CM_{ij} \tag{2.40}$$

Now all terms of equation (2.40) can be determined with the aid of $\overline{u}$: there are however five independent equations for only one variable C, so the problem is overdetermined. The coefficient can be determined via a least squares approach to calculate the value that "best satisfies" the system (2.40):

$$C = \frac{1}{2}\frac{\mathcal{L}_{ij}M_{ij}}{M_{ij}^2} \tag{2.41}$$

However, the analysis of DNS and experimental data revealed that C field predicted by the equation (2.41) varies strongly in space and contains a significant fraction of negative values, with a variance which may be ten times higher than the square mean: so the removal of C from the filtering operation is not really justified and very large negative values of the eddy viscosity are a destabilizing process in a numerical simulation.

The cure often adopted consists in averaging both numerators and denominators of equation (2.41) over space and/or time: averaging over direction of flow homogeneity has been a popular choice with can obtain good results.

2.5 Numerical Schemes

In the HeaRT code, described in detail in section 2.6 on page 21 are used the following numerical schemes and methods:

1. AUSM numerical scheme combined with quadratic upstream interpolation developed for the QUICK numerical scheme;
2. finite difference numerical scheme;
3. Newton-Raphson method;
4. Runge-Kutta method.

2.5.1 AUSM Numerical Scheme

The develop of this numerical scheme was motivated by the desire to combine the efficiency of FVS and the accuracy of FDS [13]. Considering the two-dimensional system of Euler equation for a perfect gas

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = 0 \tag{2.42}$$

where the inviscid flux vector $\mathbf{F}$ consists of two physically distinct parts, namely convective and pressure terms:

$$\mathbf{F} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho H \end{pmatrix} u + \begin{pmatrix} 0 \\ p \\ 0 \\ 0 \end{pmatrix} = F^{(c)} + \begin{pmatrix} 0 \\ p \\ 0 \\ 0 \end{pmatrix} \tag{2.43}$$

The convective terms can now be considered as a passive scalar quantities convected by a velocity u at the cell interface: the pressure flux terms are governed by the acoustic wave speed. So the two components can be discretized separately: at an interface $L < 1/2 < R$ the convective terms can be written as:

$$\mathbf{F}^{(c)}_{1/2} = u_{1/2} \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho H \end{pmatrix}_{L/R} = M_{1/2} \begin{pmatrix} \rho a \\ \rho a u \\ \rho a v \\ \rho a H \end{pmatrix}_{L/R} \tag{2.44}$$

where

$$(\bullet)_{L/R} = \begin{cases} (\bullet)_L & \text{if} \quad M_{1/2} \geqslant 0 \\ (\bullet)_R & \text{otherwise} \end{cases} \tag{2.45}$$

The advective velocity $M_{1/2}$ can be expressed as a combination of the wave speeds $(M \pm 1)$ traveling towards the interface $\left(\frac{1}{2}\right)$ from the adjacent L and R cells:

$$M_{1/2} = M_L^+ + M_R^- \tag{2.46}$$

where the split Mach number $M^{\pm}$ is defined according Van Leer splitting:

$$M^{\pm} = \begin{cases} \pm\frac{1}{4}(M \pm 1)^2 & \text{if} \quad |M| \leqslant 1 \\ \frac{1}{2}(M \pm |M|) & \text{otherwise} \end{cases} \tag{2.47}$$

The pressure term can be writed, following equation (2.46):

$$p_{1/2} = p_L^+ + p_R^- \tag{2.48}$$

and the pressure splitting is weighted using the polynomial expansion of the characteristic speeds $(M \pm 1)$. The pressure splitting can be expressed in terms of first order polynomials $(M \pm 1)$:

$$p^{\pm} = \begin{cases} \frac{p}{2}(1 \pm M) & \text{if} \quad |M| \leqslant 1 \\ \frac{p}{2}(M \pm |M|)/M & \text{otherwise} \end{cases} \tag{2.49}$$

or in terms of second order polynomials $(M \pm 1)^2$:

$$p^{\pm} = \begin{cases} \frac{p}{4}(M \pm 1)^2(2 \mp M) & \text{if} \quad |M| \leqslant 1 \\ \frac{p}{2}(M \pm |M|)/M & \text{otherwise} \end{cases} \tag{2.50}$$

All the above formulas can be recast in the form:

$$\begin{aligned} \begin{pmatrix} \rho u \\ \rho u u + p \\ \rho u v \\ \rho u H \end{pmatrix}_{1/2} &= M_{1/2}\frac{1}{2}\left[\begin{pmatrix} \rho a \\ \rho a u \\ \rho a v \\ \rho a H \end{pmatrix}_L + \begin{pmatrix} \rho a \\ \rho a u \\ \rho a v \\ \rho a H \end{pmatrix}_R\right] \\ &- \frac{1}{2}\left|M_{1/2}\right| \Delta_{1/2} \begin{pmatrix} \rho a \\ \rho a u \\ \rho a v \\ \rho a H \end{pmatrix} \\ &+ \begin{pmatrix} 0 \\ p_L^+ + p_R^- \\ 0 \\ 0 \end{pmatrix} \end{aligned} \tag{2.51}$$

where $\Delta_{1/2}\{\bullet\} = \{\bullet\}_R - \{\bullet\}_L$. The first term on the right hand side is a Mach-number-weighted average of L and R states; the second term is the numerical dissipation.

Summarizing, this numerical scheme (AUSM - Advection Upstream Splitting Method) treats the convective and the pressure terms separately:

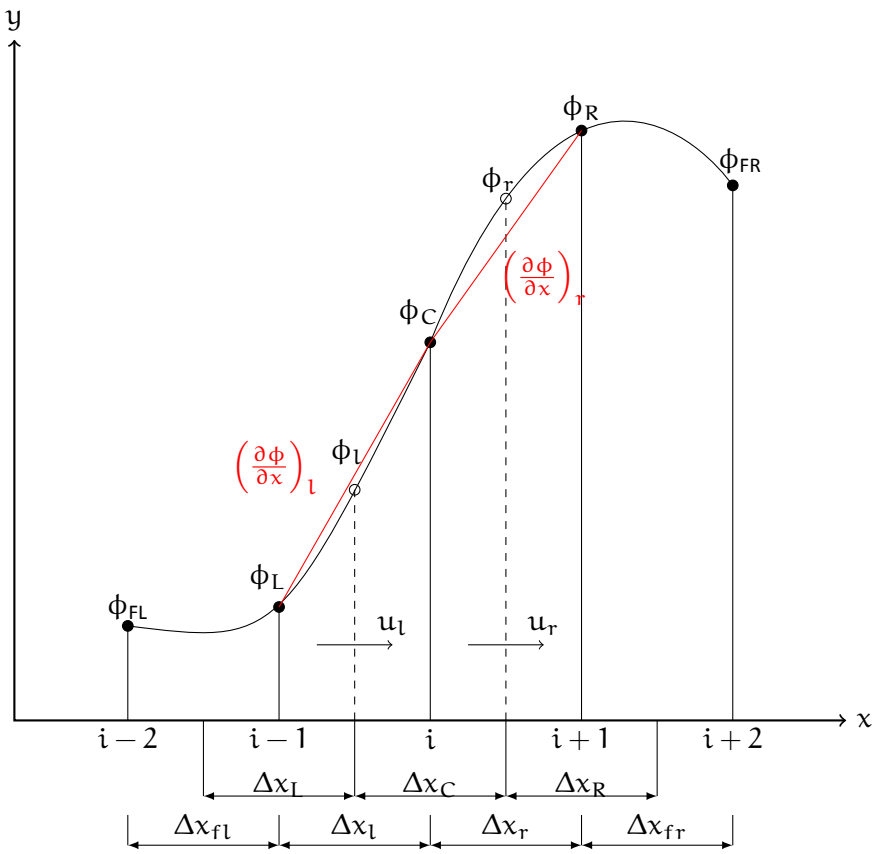


Figure 2.1: Schematic showing a control volume for node i, variables ϕ at upstream and downstream nodes, grid spaces, faces velocities u_* and face gradients $(\partial\phi/\partial x)_*$

- convective terms are biased from the upstream using a properly definited cell-interface velocity;
- pressure term is dealt with convective terms using acoustic waves.

Quick Recontruction

In HeaRT code, "L" and "R" convective terms of equation (2.51) are evaluated by the mean of a quadratic upstream interpolation developed in the QUICK (QUadratic Interpolation for Convective Kinetics) numerical scheme [14]; the method can be expressed as a linear interpolation corrected by a term proportional to the upstream-weighted curvature (see figure 2.1):

$$\begin{aligned}\phi_l^* &= \frac{1}{2}(\phi_L+\phi_C) - \frac{\Delta x_l^2}{8}\frac{1}{\Delta x_L}\left(\frac{\phi_C-\phi_L}{\Delta x_l} - \frac{\phi_L-\phi_{FL}}{\Delta x_{fl}}\right)\\ \phi_r^* &= \frac{1}{2}(\phi_C+\phi_R) - \frac{\Delta x_r^2}{8}\frac{1}{\Delta x_C}\left(\frac{\phi_R-\phi_C}{\Delta x_r} - \frac{\phi_C-\phi_L}{\Delta x_l}\right)\end{aligned} \tag{2.52}$$

where subscripts FL, L, C, R and FR refer to the far left, left, central, right and far right nodes; subscripts l and r refer to the left and right faces.

The general formula for the QUICK scheme is the following:

$$
\begin{aligned}
\phi_l^* &= \frac{1}{2}\left(\phi_L + \phi_C\right) - \frac{\Delta x_l^2}{8}\mathrm{Curv}_l \\
&= \frac{1}{2}\left(\phi_L + \phi_C\right) - \frac{\Delta x_l^2}{8}\left[\frac{1}{\Delta x_L}\left(\mathrm{Grad}_l - \mathrm{Grad}_{fl}\right)\right] \\
\phi_r^* &= \frac{1}{2}\left(\phi_C + \phi_R\right) - \frac{\Delta x_r^2}{8}\mathrm{Curv}_r \\
&= \frac{1}{2}\left(\phi_C + \phi_R\right) - \frac{\Delta x_r^2}{8}\left[\frac{1}{\Delta x_C}\left(\mathrm{Grad}_r - \mathrm{Grad}_l\right)\right]
\end{aligned}
\tag{2.53}
$$

where "Curv" represents (upstream-weighted) ϕ curvatures and "Grad" ϕ gradients.

2.5.2 Finite Difference Numerical Scheme

The idea of a finite-difference representation for a derivative can be introduced by recalling the definition of the derivative for the function $u(x, y)$ at $x = x_0$ and $y = y_0$, that is:

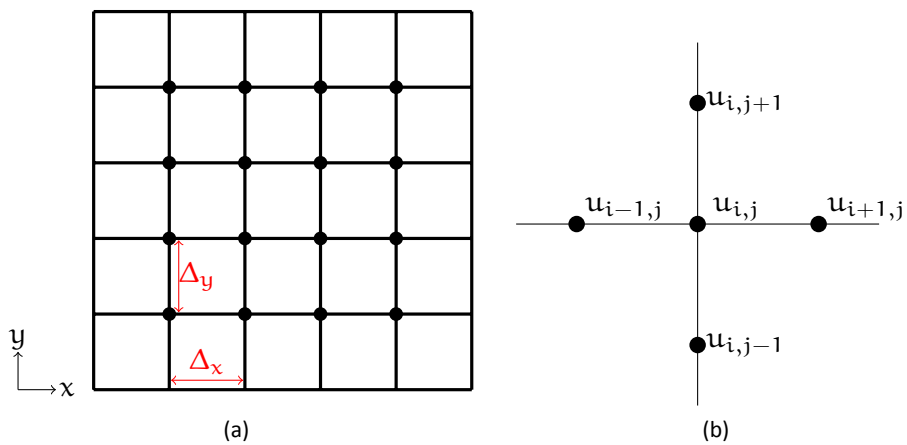


Figure 2.2: A typical finite-difference grid

$$
\frac{\partial u}{\partial x} = \lim_{\Delta x \to 0} \frac{u\left(x_0 + \Delta x, y_0\right) - u\left(x_0, y_0\right)}{\Delta x} \tag{2.54}
$$

If the function u is continuous and Δ_x is "sufficiently" small but finite, is expected that

$$
\frac{\left[u\left(x_0 + \Delta x, y_0\right) - u\left(x_0, y_0\right)\right]}{\Delta x}
$$

will be a good approximation to $\partial u/\partial x$. Developing a Taylor-series expansion for $u\left(x_0 + \Delta x, y_0\right)$ about $\left(x_0, y_0\right)$ gives (see figure 2.2)

$$
\begin{aligned}
u\left(x_0 + \Delta x, y_0\right) = u\left(x_0, y_0\right) &+ \left.\frac{\partial u}{\partial x}\right|_0 \Delta x + \left.\frac{\partial^2 u}{\partial x^2}\right|_0 \frac{(\Delta x)^2}{2!} + \ldots \\
&+ \left.\frac{\partial^{n-1} u}{\partial x^{n-1}}\right|_0 \frac{(\Delta x)^{n-1}}{(n-1)!} + \left.\frac{\partial^n u}{\partial x^n}\right|_\xi \frac{(\Delta x)^n}{n!}
\end{aligned}
\tag{2.55}
$$

where $x_0 \leqslant \xi \leqslant (x_0 + \Delta x)$. Thus we can form the "forward" difference by rearranging equation (2.55), switching to the i, j notation:

$$\left.\frac{\partial u}{\partial x}\right|_{i,j} = \frac{u_{i+1,j} - u_{i,j}}{\Delta x} + \mathcal{O}(\Delta x) \tag{2.56}$$

An infinite number of difference representations can be found for $\partial u/\partial x|_{i,j}$, for example we could expand "backward":

$$\left.\frac{\partial u}{\partial x}\right|_{i,j} = \frac{u_{i,j} - u_{i-1,j}}{\Delta x} + \mathcal{O}(\Delta x) \tag{2.57}$$

or subtracting equations (2.56) and (2.57), to obtain the "central" difference:

$$\left.\frac{\partial u}{\partial x}\right|_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x} + \mathcal{O}(\Delta x)^2 \tag{2.58}$$

or adding equations (2.56) and (2.57), to obtain an approximation of the second derivative:

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{i,j} = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{(\Delta x)^2} + \mathcal{O}(\Delta x)^2 \tag{2.59}$$

The mixed derivative approximation can be obtained from the Taylor-series expansion for two variables:

$$\begin{aligned} u(x_0 + \Delta x, y_0 + \Delta_y) = u(x_0, y_0) &+ \left(\Delta x \frac{\partial}{\partial x} + \Delta y \frac{\partial}{\partial y}\right) u(x_0, y_0) + \\ &+ \frac{1}{2!}\left(\Delta x \frac{\partial}{\partial x} + \Delta y \frac{\partial}{\partial y}\right)^2 u(x_0, y_0) + \dots \\ &+ \frac{1}{n!}\left(\Delta x \frac{\partial}{\partial x} + \Delta y \frac{\partial}{\partial y}\right)^n u(x_0 + \theta\Delta x, y_0 \theta \Delta y) \end{aligned} \tag{2.60}$$

where $0 \leqslant \theta \leqslant 1$; rearranging equation (2.60), switching to the i, j notation:

$$\left.\frac{\partial^2 u}{\partial x \partial y}\right|_{i,j} = \frac{1}{\Delta x}\left(\frac{u_{i+1,j} - u_{i+1,j-1}}{\Delta y} - \frac{u_{i,j} - u_{i,j-1}}{\Delta x}\right) + \mathcal{O}(\Delta x, \Delta y) \tag{2.61}$$

In this manner is possible to obtain all derivatives that appear in Navier-Stokes equations described in section 2.1, for cylindrical and cartesian structured grids.

2.5.3 Newton-Raphson Method

Newton–Raphson method is a method for finding successively better approximations to the roots (or zeroes) of a real-valued function. Assuming that an initial estimate x_0 is known for the desired root α of $f(x) = 0$, Newton-Raphson method will produce a sequence of iterates which will converge to α. Since x_0 is assumed close to α, the function $f(x)$ can be approximated by constructing its tangent line at $(x_0, f(x_0))$, as can be seen in figure 2.3 on the next page.

Then the root of this tangent line can be used to approximate α, calling this approximation x_1. This process can be repeated to obtain a sequence of iterates x_n with the following iteration formula:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad \text{with} \quad n \geqslant 0 \tag{2.62}$$

Newton-Raphson method can be explained also starting with a Taylor series development: expanding $f(x)$ about x_n:

$$f(x) = f(x_n) + (x - x_n) f'(x_n) + \frac{(x - x_n)^2}{2} f''(\xi) \tag{2.63}$$

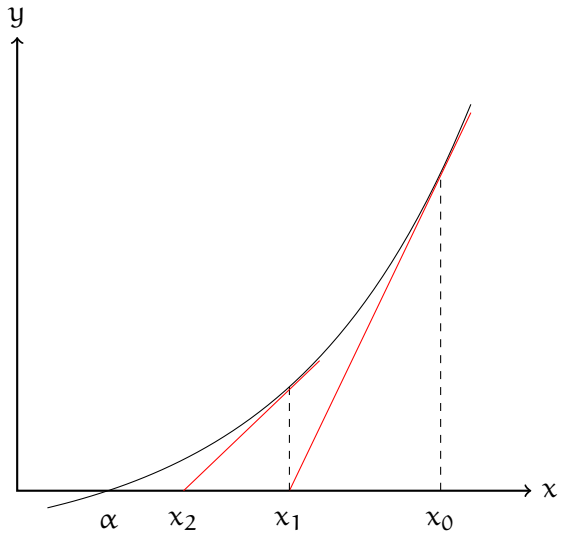


Figure 2.3: Newton's method

with ξ between x and x_n. Letting $x = \alpha$ and using $f(\alpha) = 0$, solving for α we obtain:

$$\alpha = x_n - \frac{f(x_n)}{f'(x_n)} - \frac{(\alpha - x_n)^2}{2} \cdot \frac{f''(\xi_n)}{f'(x_n)} \tag{2.64}$$

with ξ_n between x_n and α. Recalling equation (2.62):

$$\alpha - x_{n+1} = -(\alpha - x_n)^2 \cdot \frac{f''(\xi_n)}{2f'(x_n)} \quad \text{with} \quad n \geqslant 0 \tag{2.65}$$

Using equations (2.64) and (2.65), Newton-Raphson method has a quadratic order of convergence ($p = 2$) in:

$$|\alpha - x_{n+1}| \leqslant c\,|\alpha - x_n|^p \quad \text{with} \quad n \geqslant 0, c > 0 \tag{2.66}$$

2.5.4 Runge-Kutta Method

Runge-Kutta method is a numerical scheme commonly used to solve initial value problems for ODE's. This scheme was developed around 1900 by the german mathematicians C. Runge and M. W. Kutta and essentially utilizes the weighted average of several solutions over the interval Δt in order to improve accuracy of solution. A generic Runge-Kutta scheme can be expressed by the:

$$\begin{cases} u^{n+1} = u^n + h \sum_{i=1}^{s} b_i k_i^n & n = 0, .., N-1 \\ u^0 = u(t_0) \end{cases} \tag{2.67}$$

where

$$h = t^{n+1} - t^n \tag{2.68}$$

$$k_i^n = F\left(t_n + c_i h, u^n + h\sum_{j=1}^{i-1} a_{ij} k_j^n\right) \qquad i = 1, .., s \tag{2.69}$$

$$c_1 = 0 \tag{2.70}$$

$$\sum_{j=1}^{i-1} a_{ij} = c_i \quad \text{for} \quad i = 2, \ldots, s \tag{2.71}$$

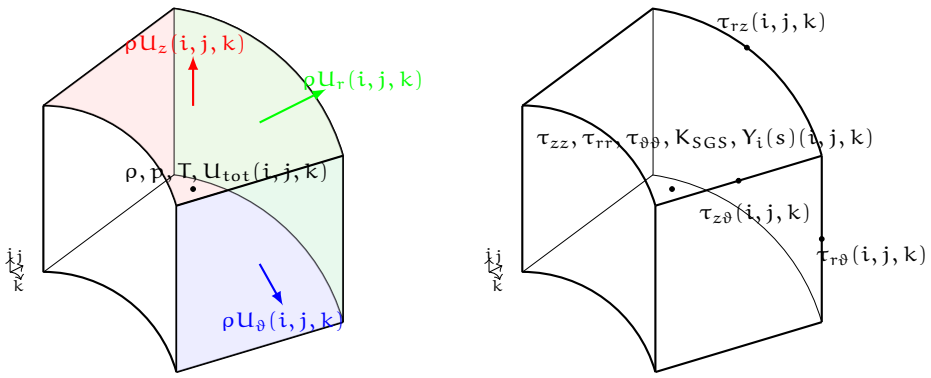
and a_{ij}, c_i, b_i are coefficients to be determined and s are the sub steps of Runge-Kutta scheme.

2.6 HeaRT Code

Into the section 2.1 on page 8 we have pointed out that in order to perform a LES numerical simulation of an energy burner, Navier-Stokes conservation laws are suitable. The Navier-Stokes system is the first component of our CFD study: our aim is to find the solution of the Navier-Stokes system (conveniently modified) when the initial and boundary conditions are imposed to it. In this section the numerical model used to find this solution is presented.

HeaRT (Heat Release and Turbulence) numerical code, developed by UTTEI (Technical Unit for Advanced Technologies for Energy and Industry) of ENEA, in collaboration with DIMA (Mechanical and Aerospace Engineering Department) of Sapienza University of Rome, is an unsteady numerical solver for turbulent reacting and non reacting flows, at low Mach number, in three-dimensional cartesian and cylindrical geometries, discretized by the means of structured grids. Navier-Stokes equation are implemented in the compressible formulation, in order to highlight wave propagation phenomena that are very important for combustion instability analysis.

Governing equations are solved, in HeaRT code, on a staggered grid scheme: scalars (density, temperature, pressure, total energy and mass fractions) are set in the cell center, while the three mass fluxes are staggered in space by half grid width and collocated in the "positive" faces of the cell (see figure 2.4a). Viscous stresses are set in the cell center and in the edges of the cell (see figure 2.4b).



(a) Scalars and Mass Fluxes (b) Viscous Stresses

Figure 2.4: Variables Position in a Cell

Summarizing, HeaRT code is a staggered numerical code that uses:

- AUSM numerical scheme with QUICK reconstruction for convective fluxes of scalar variables (ρ, U_{tot}, Y_i); the numerical algorithm, for evaluation of convective mass flux in a generic direction, is the following:

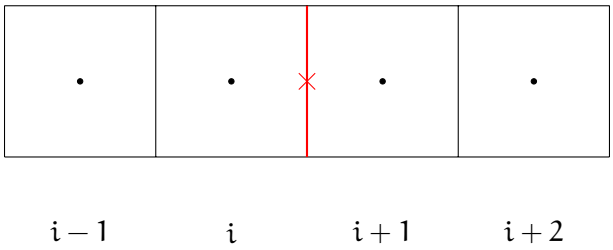


Figure 2.5: AUSM grid

1. left and right sound velocities are evaluated on the red face in figure 2.5 by the mean of a quadratic upstream interpolation

$$\begin{aligned} a_L &= Q_1^+ a\,(i+1) + Q_2^+ a\,(i) + Q_3^+ a\,(i-1) \\ a_R &= Q_1^- a\,(i+2) + Q_2^- a\,(i+1) + Q_3^- a\,(i) \end{aligned} \tag{2.72}$$

   where Q_i^+ and Q_i^-, with $i = 1, \cdots 3$ are grid spacing depending metric terms;

2. Mach number on the red face in figure 2.5 is evaluated from the

$$M = U\,(i)\,\frac{a_L + a_R}{2 a_L a_R} \tag{2.73}$$

   where $U(i)$ is the flow velocity othogonal to the red face, where is collocated due to the staggered formulation;

3. the convective mass flux is evaluated from

$$F_c = \frac{1}{2} M\,(\rho a_L + \rho a_R) + |M|\,(\rho a_L - \rho a_R) \tag{2.74}$$

   where ρa_L and ρa_R are the product of density and sound velocity evaluated by the means of equation (2.72);

the same algorithm is used for the other two directions and for convective energy fluxes $\rho U_{tot} + p$ and convective species fluxes ρY_i.

- second order centered finite difference numerical scheme for momentum (ρU_z, ρU_r, ρU_θ) and for diffusive fluxes of scalar variables; due to the staggered formulation adopted, some interpolations are necessary to obtain all the required variables in the right place; for "axial" momentum flux, mass flow and velocity have to be re-collocated in the cell center by the means of a linear interpolation from the two cell faces: as can be seen in figure 2.6a on the next page, staggered velocties and mass fluxes (the black crosses) are interpolated in the momentum cell faces (the red dotted cell) to compute momentum flux balance, with the following equation:

$$\phi(i) = c_i \phi(i + 1/2) + (1 - c_i)\phi(i - 1/2) \tag{2.75}$$

  where due to the staggered formulation $c_i = 1/2$; after that, axial convective momentum flux $p + \rho u u$ can be computed, because the pressure is collocated on the cell center; for "orthogonal" momentum flux, mass flux and velocity have to be re-collocated on the corners of the scalar cell (the red dots in figure 2.6b on the facing page); orthogonal velocities (green crosses) are interpolated in the momentum cell face with the equation:

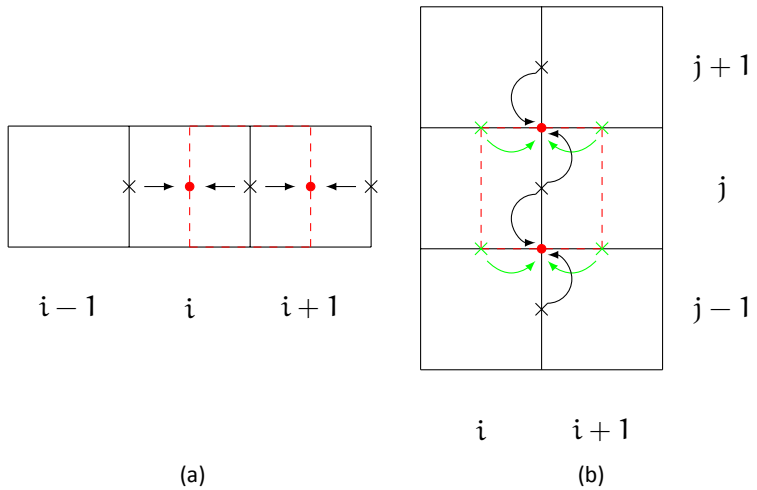


Figure 2.6: Interpolation scheme for “axial” and “orthogonal” mass fluxes and velocities

$$\phi(i+1/2) = c_i\phi(i+1) + (1-c_i)\phi(i) \tag{2.76}$$

but otherwise from “axial” flux, $c_i = \frac{x_{i+1/2}-x_i}{x_{i+1}-x_i}$; axial mass fluxes (black crosses) are interpolated in the momentum cell face with the equation:

$$\phi(j+1/2) = c_j\phi(j+1) + (1-c_j)\phi(j) \tag{2.77}$$

where $c_j = \frac{y_{j+1/2}-y_j}{y_{j+1}-y_j}$. after these simple interpolations, the other convective flux ($\rho u_a u_o$, where subscript “a” is for “axial” component and subscript “o” is for “orthogonal” component of flow velocity); viscous terms in the momentum equation, due to the staggered formulation adopted (see figure 2.4b on page 21), don’t need any interpolation and the derivatives of these terms can be easily calculated by the equation (2.56).

For energy equation, viscous work terms and heat fluxes are evaluated from equation (2.56) as for diffusive terms in species equation.

- Newton-Raphson procedure for temperature calculation; the temperature is evaluated, starting from its old value (at the previous time-step), as the value for which the equation $nRT + E_{tot} - H_{tot} = 0$, where R is the universal gas constant, E_{tot} is the internal energy and H_{tot} is the standard formation enthalpy of the mixture.

- Runge-Kutta third order numerical scheme for time integration; in order to obtain a III order accurate scheme $s = 3$ must be chosen in equation (2.71); in the scheme here adopted [15] the coefficients are set to:

$$\begin{aligned} &c_2 = c_3 = 0 \\ &b_1 = b_2 = \tfrac{1}{6} \quad b_3 = \tfrac{2}{3} \\ &a_{21} = 1 \quad a_{31} = a_{32} = \tfrac{1}{4}. \end{aligned}$$

In order to ensure the calculation stability two conditions must be respected: the condition on the *Courant-Friedricks-Lewy* (CFL) condition and the condition on the *Von Neumann* number VNN for the stability of transport-diffusion systems. The CFL (or *Courant*) number is defined as

$$\mathrm{CFL} = \lambda_j^{\mathrm{MAX}} \frac{\Delta t}{\Delta x_j} \tag{2.78}$$

where λ_j^{MAX} is the maximum local eigenvalue in the jth direction. The physical meaning of this condition can be explained with reference to Figure 2.7. $\mathrm{CFL} = 1$ implies that Δt is the time by which the signal entering the control volume crosses it reaching the opposite interface. Δt is therefore the time by which the flux estimate should be updated, unless the initial estimate already takes into account the outcoming flux variation.

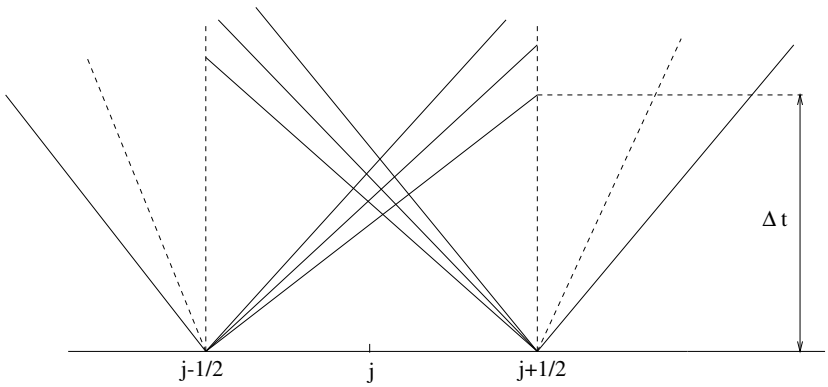


Figure 2.7: Physical interpretation of the CFL condition

The non-linear stability of this scheme, in conjunction with a ENO reconstruction is reported in [15] to be given by the condition $\mathrm{CFL} < 1$. Nevertheless, the application of III order R-K schemes, in conjunction with the staggered discretization described in section (2.5 on page 15), to compressible flow LES is reported in [16] to be performed using $\mathrm{CFL} = 0.29$ in order to limit the effects of the truncation error.

The condition on the VNN defined as

$$\mathrm{VNN} = \nu \frac{\Delta t}{(\Delta x)^2} \tag{2.79}$$

for a Forward-Time-Centered-Space (FTCS) it is reported to be $\mathrm{VNN} < 0.5$ proven that a condition for the cell Reynolds number Re_Δ is satisfied. The complete analysis of the FTCS scheme stability is reported in [17].

3 Mesh Refinement

Dall'analisi precedente appare evidente come sia necessario introdurre una tecnica di mesh refinement "ad hoc" per il codice HeaRT, considerate le caratteristiche dei fluidi simulati: siamo infatti in presenza di flussi comprimibili, reagenti e ad elevata turbolenza. Considering the features of the flow field (compressible, with chemical reactions and with an high level of turbulence) simulated by the mean of HeaRT code, is been developed a mesh refinement technique founded on joined grids: on the entire computational domain there's only one grid, that can have different zones with different spatial resolution. The refinement is constant and the same in the three direction:

$$\begin{aligned} \frac{\Delta x_1}{\Delta x_2} &= \frac{\Delta x_2}{\Delta x_3} = \dots \frac{\Delta x_{l-1}}{\Delta x_l} = r \\ \frac{\Delta y_1}{\Delta y_2} &= \frac{\Delta y_2}{\Delta y_3} = \dots \frac{\Delta y_{l-1}}{\Delta y_l} = r \\ \frac{\Delta z_1}{\Delta z_2} &= \frac{\Delta z_2}{\Delta z_3} = \dots \frac{\Delta z_{l-1}}{\Delta z_l} = r \end{aligned} \tag{3.1}$$

Level data structure is udes for an easier identification of the number of fine cells contained in a single coarse cell. For example, if a level l_1 zone shares a border with a finest level l_2 zone, and the refinement ratio (a fixed parameter that identifies the number of fine cells contained in the immediately previous coarse level - i. e. between level 4 and level 3 - and is the same for all the entire domain, see equation (3.1)) is set to p, there are $n = p^{(l_2 - l_1)}$ fine cells contained in a coarse cell.

With this approach is necessary to develop two different communication procedures, one that transfer variable values from coarse to fine grid (named *prolongation*) and the other that makes the opposite way (named *restriction*), but in both cases variable values are transferred only on "ghost" cells.

3.0.1 Solution Algorithm

In the solution algorithm, each grid block solves conservation equations indipentently from other grid blocks, but the time step choosen in order to satisfy CFL condition is the same for all grids: minimum value is selected, as can be seen in figure 3.1 on the next page (where, for the sake of simplicity, only 3 zones of 3 different spatial resolutions - or levels - are illustrated).

Time integration procedure is here described:

1. *First RK Substep* of all zones, from t_0 to t_{in_1};
2. *Restriction* of boundary conditions at time t_{fin_1} from zone 2 to zone 1;
3. *Prolongation* of boundary conditions at time t_{fin_1} from zone 1 to zone 2;
4. *Restriction* of boundary conditions at time t_{fin_1} from zone 1 to zone 0;
5. *Prolongation* of boundary conditions at time t_{fin_1} from zone 0 to zone 1;
6. *Second RK Substep* of all zones, from t_{in_1} to t_{in_2};
7. *Restriction* of boundary conditions at time t_{fin_2} from zone 2 to zone 1;
8. *Prolongation* of boundary conditions at time t_{fin_2} from zone 1 to zone 2;

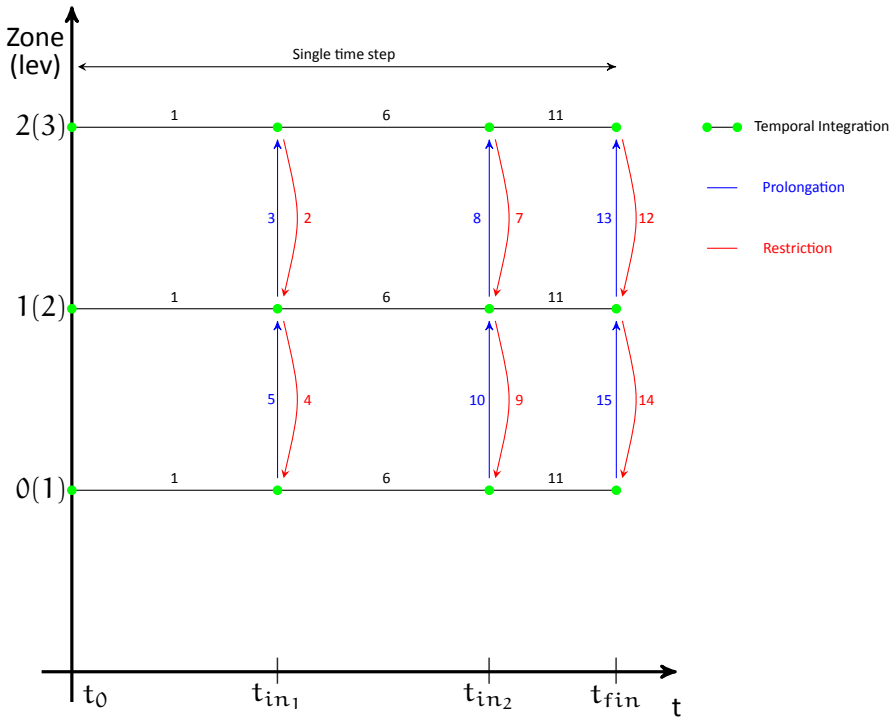


Figure 3.1: Scheme of temporal advance on all grid levels

9. *Restriction* of boundary conditions at time t_{fin_2} from zone 1 to zone 0;

10. *Prolongation* of boundary conditions at time t_{fin_2} from zone 0 to zone 1;

11. *Third RK Substep* of all zones, from t_{in_2} to t_{fin};

12. *Restriction* of boundary conditions at time t_{fin_3} from zone 2 to zone 1;

13. *Prolongation* of boundary conditions at time t_{fin_3} from zone 1 to zone 2;

14. *Restriction* of boundary conditions at time t_{fin_3} from zone 1 to zone 0;

15. *Prolongation* of boundary conditions at time t_{fin_3} from zone 0 to zone 1;

So in HeaRT solver routine, at the end of any Runge-Kutta substep, finest zones start to transfer variables values on ghost cells of neighboring coarser zones, that subsequently transfer variables values on ghost cells of the finest zones; this couple of operations (prolongation and restriction) is repeated until coarsest zones are reached, and all ghost cells of all zones have updated their values. At this point, a new Runge-Kutta substep begins for all the zones of the domain.

Also in this case communications between zones of different spatial resolution are very important, because a correct variables transfer on ghost cells is essential to obtain a continuous solution on the entire domain.

3.0.2 Communication Procedures

For the sake if simplicity, communication procedures hereafter illustrated are related to a cartesian uniform bi-dimensional grid: their form and structure can be easily extended to a three-dimensional cylindrical or cartesian non-uniform grid.

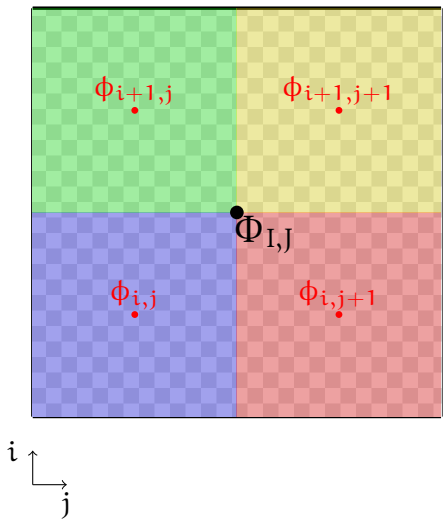


Figure 3.2: Scalar positions in a bidimensional grid

Restriction

Restriction procedure is used for communication of numerical solution from level l to level $l-1$. For scalar variables (collocated at the centre of the cell, see figure 3.2), coarse value is obtained from fine values by the means of:

$$\Phi_{I,J} = \frac{\left(\phi_{i,j}A_{i,j} + \phi_{i,j+1}A_{i,j+1} + \phi_{i+1,j}A_{i+1,j} + \phi_{i+1,j+1}A_{i+1,j+1}\right)}{A_{i,j} + A_{i,j+1} + A_{i+1,j} + A_{i+1,j+1}} \tag{3.2}$$

where $A_{i,j} + A_{i,j+1} + A_{i+1,j} + A_{i+1,j+1} = A_{I,J}$; variables $A_{i,j}$, $A_{i,j+1}$, $A_{i+1,j}$, $A_{i+1,j+1}$ are fine cell areas (colored areas in figure 3.2), contained in coarse cell $A_{I,J}$ (checkboard zone in figure 3.2). Essentially a scalar value is a weighted sum of fine value, where the weight is the area fraction of each fine cell contained in the coarse one. Is straightforward that for a three-dimensional case coarse value is calculated with:

$$\Phi_{I,J} = \frac{\sum_{i=1}^{n} \phi_i V_i}{\sum_{i=1}^{n} V_i} \tag{3.3}$$

where n is the number of fine scalar cells contained in a scalar coarse cell.

For momentum ρU, collocated at the positive edges (in a bi-dimensional grid) of a cell (see figure 3.3 on the next page, coarse grid value is obtained from fine values by the means of:

$$\Phi_{I,J} = \frac{\phi_{i+1,j}l_1 + \phi_{i+1,j+1}l_2}{l_1 + l_2} \tag{3.4}$$

where $L = l_1 + l_2$. For a three-dimensional grid, coarse scalar value is calculated from:

$$\Phi_{I,J} = \frac{\sum_{i=1}^{n} \phi_i A_i}{\sum_{i=1}^{n} A_i} \tag{3.5}$$

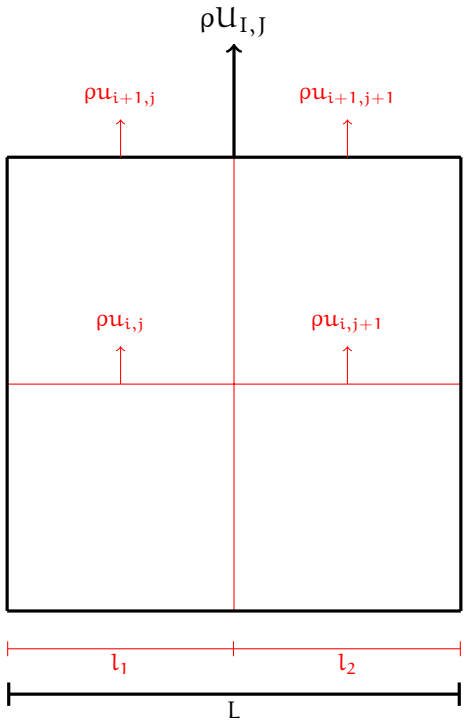


Figure 3.3: Momentum restriction

where A_i is the face area of fine cell where is collocated ϕ_i momentum and $\mathcal{A} = \sum_{i=1}^{n} A_i$ is the face area of coarse cell where $\Phi_{I,J}$ is collocated.

Prolongation

For prolongation is choosed an accurated interpolation algorithm, and therefore the whole communication "operator" is really complex.

The selected algorithm is the same for both scalar and momentum prolongation: least squares interpolation, that conserves mean value in the control volumes. To compute the fine cell interpolated value a Taylor series expansion about the coarse value c_i that belongs to the corresponding coarse cell (the "centroid"):

$$\begin{aligned}\phi_{c_i}^{INT}(\mathbf{x}) = \phi_{c_i} &+ \left.\frac{\partial\phi}{\partial x}\right|_{c_i}\Delta x + \left.\frac{\partial\phi}{\partial y}\right|_{c_i}\Delta y + \left.\frac{\partial\phi}{\partial z}\right|_{c_i}\Delta z + \\ &+ \left.\frac{\partial^2\phi}{\partial x^2}\right|_{c_i}\frac{\Delta x^2}{2} + \left.\frac{\partial^2\phi}{\partial y^2}\right|_{c_i}\frac{\Delta y^2}{2} + \left.\frac{\partial^2\phi}{\partial z^2}\right|_{c_i}\frac{\Delta z^2}{2} + \\ &+ \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_{c_i}\Delta x\Delta y + \left.\frac{\partial^2\phi}{\partial x\partial z}\right|_{c_i}\Delta x\Delta z + \left.\frac{\partial^2\phi}{\partial y\partial z}\right|_{c_i}\Delta y\Delta z\end{aligned} \tag{3.6}$$

where $\Delta x = x - x_{c_i}, \Delta y = y - y_{c_i}, \Delta z = z - z_{c_i}$ being the distances, along the three cartesian coordinates, between the reconstruction point and the centroid c_i where derivatives in equation (3.6) are calculated. If first order interpolation is choosed, only the first 4 must be mantained in equation (3.6). In a three-dimensional case, for a third order interpolation, there are 9 unknowns (the derivatives in the equation (3.6)), so for the sake of simplicity of the numerical algorithm, a total number of 26 neighboring points to the centroid are used to construct the stencil of least squares interpolation: indeed, if the centroid has coordinates (I, J, K), the stencil extends from $(I-1, J-1, K-1)$ to $(I+1, J+1, K+1)$ (see figure 3.4 on page 31).

To conserve the mean value in the control volume of the interpolating function $\phi_{c_i}^{INT}(\mathbf{x})$, the following equation must be forced:

$$\overline{\phi}_i = \frac{1}{V_i}\int_{V_i} \phi_{c_i}^{INT}(\mathbf{x})dV \tag{3.7}$$

Substituting equation (3.6) in (3.7) and computing the mean value in a volume V_j of the interpolation stencil gives:

$$\begin{aligned}
\phi_j = \overline{\phi}_i &+ \left.\frac{\partial\phi}{\partial x}\right|_{c_i}\frac{1}{V_j}\int_{V_j}\Delta x dV + \left.\frac{\partial\phi}{\partial y}\right|_{c_i}\frac{1}{V_j}\int_{V_j}\Delta y dV + \\
&+ \left.\frac{\partial\phi}{\partial z}\right|_{c_i}\frac{1}{V_j}\int_{V_j}\Delta z dV + \\
&+ \left.\frac{\partial^2\phi}{\partial x^2}\right|_{c_i}\frac{1}{2V_j}\int_{V_j}\Delta x^2 dV + \left.\frac{\partial^2\phi}{\partial y^2}\right|_{c_i}\frac{1}{2V_j}\int_{V_j}\Delta y^2 dV + \\
&+ \left.\frac{\partial^2\phi}{\partial z^2}\right|_{c_i}\frac{1}{2V_j}\int_{V_j}\Delta z^2 dV + \\
&+ \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_{c_i}\frac{1}{V_j}\int_{V_j}\Delta x\Delta y dV + \left.\frac{\partial^2\phi}{\partial x\partial z}\right|_{c_i}\frac{1}{V_j}\int_{V_j}\Delta x\Delta z dV \\
&+ \left.\frac{\partial^2\phi}{\partial y\partial z}\right|_{c_i}\frac{1}{V_j}\int_{V_j}\Delta y\Delta z dV
\end{aligned} \tag{3.8}$$

To avoiding calculation of $x - x_{c_i}$, ecc..., integrals for each control volume V_j with respect to c_i [18], values $x - x_{c_i}$, $y - y_{c_i}$, $z - z_{c_i}$ in equation (3.8) are replaced with $(x - x_{c_j})(x_{c_j} - x_{c_i})$, $(y - y_{c_j})(y_{c_j} - y_{c_i})$, $(z - z_{c_j})(z_{c_j} - z_{c_i})$:

$$\begin{aligned}
\overline{\phi}_j = \overline{\phi}_i &+ \left.\frac{\partial\phi}{\partial x}\right|_{c_i}\widehat{x} + \left.\frac{\partial\phi}{\partial y}\right|_{c_i}\widehat{y} + \left.\frac{\partial\phi}{\partial z}\right|_{c_i}\widehat{z} + \left.\frac{\partial^2\phi}{\partial x^2}\right|_{c_i}\widehat{x^2} + \left.\frac{\partial^2\phi}{\partial y^2}\right|_{c_i}\widehat{y^2} + \left.\frac{\partial^2\phi}{\partial z^2}\right|_{c_i}\widehat{z^2} \\
&+ \left.\frac{\partial^2\phi}{\partial xy}\right|_{c_i}\widehat{xy} + \left.\frac{\partial^2\phi}{\partial xz}\right|_{c_i}\widehat{xz} + \left.\frac{\partial^2\phi}{\partial yz}\right|_{c_i}\widehat{yz}
\end{aligned} \tag{3.9}$$

with

$$\begin{aligned}
\widehat{x^n y^m z^p} &= \frac{1}{V_i}\int_{V_i}\left[(x - x_{c_j})(x_{c_j} - x_{c_i})\right]^n\left[(y - y_{c_j})(y_{c_j} - y_{c_i})\right]^m \\
&\left[(z - z_{c_j})(z_{c_j} - z_{c_i})\right]^p dV \\
&= \sum_{k=0}^{n}\frac{n!}{k!(n-k)!}(x_{c_j} - x_{c_i})^k\sum_{l=0}^{m}\frac{m!}{l!(m-l)!}(y_{c_j} - y_{c_i})^l \\
&\sum_{r=0}^{p}\frac{p!}{r!(p-r)!}(y_{c_j} - y_{c_i})^r\,\overline{x^{n-k}y^{m-l}z^{p-r}}
\end{aligned} \tag{3.10}$$

and

$$\overline{x^m y^n z^p} = \frac{1}{V_i}\int_{V_i}(x - x_{c_i})^m(y - y_{c_i})^n(z - z_{c_i})^p dV \tag{3.11}$$

Writing equation (3.9) for any coarse point that belongs to the interpolation stencil, an overdetermined system of equations can be obtained:

$$\mathbf{\Delta}\phi = \mathbf{S}d\phi \tag{3.12}$$

where

$$\mathbf{\Delta}\phi = \begin{bmatrix} w_{i1}\left(\overline{\phi}_1 - \overline{\phi}_{c_i}\right) \\ w_{i2}\left(\overline{\phi}_2 - \overline{\phi}_{c_i}\right) \\ \cdots \\ \cdots \\ w_{iN}\left(\overline{\phi}_{N_i} - \overline{\phi}_{c_i}\right) \end{bmatrix} \tag{3.13}$$

$$\mathbf{S} = \begin{bmatrix} \widehat{x}_1 & \widehat{y}_1 & \widehat{z}_1 & \widehat{x^2}_1 & \widehat{y^2}_1 & \widehat{z^2}_1 & \widehat{xy}_1 & \widehat{yz}_1 & \widehat{xz}_1 \\ \widehat{x}_2 & \widehat{y}_2 & \widehat{z}_2 & \widehat{x^2}_2 & \widehat{y^2}_2 & \widehat{z^2}_2 & \widehat{xy}_2 & \widehat{yz}_2 & \widehat{xz}_2 \\ & & & & \cdots & & & & \\ & & & & \cdots & & & & \\ \widehat{x}_{N_i} & \widehat{y}_{N_i} & \widehat{z}_{N_i} & \widehat{x^2}_{N_i} & \widehat{y^2}_{N_i} & \widehat{z^2}_{N_i} & \widehat{xy}_{N_i} & \widehat{yz}_{N_i} & \widehat{xz}_{N_i} \end{bmatrix} \tag{3.14}$$

$$d\phi = \begin{bmatrix} \frac{\partial\phi}{\partial x} & \frac{\partial\phi}{\partial y} & \frac{\partial\phi}{\partial z} & \frac{\partial^2\phi}{\partial x^2} & \frac{\partial^2\phi}{\partial y^2} & \frac{\partial^2\phi}{\partial z^2} & \frac{\partial\phi}{\partial xy} & \frac{\partial\phi}{\partial yz} & \frac{\partial\phi}{\partial xz} \end{bmatrix} \tag{3.15}$$

and

$$w_{ij} = \frac{1}{\left|\overrightarrow{x}_j - \overrightarrow{x}_i\right|^2} \tag{3.16}$$

The system of equations (3.12) can be rearranged in the form:

$$\left(\mathbf{S}^T\mathbf{S}\right)^{-1}\mathbf{S}^T\mathbf{\Delta}\phi = \mathbf{C}\mathbf{\Delta}\phi = d\phi \tag{3.17}$$

where $\mathbf{C}$ matrix contains only geometrical constants and so can be calculated once.

In figure 3.4 on the next page interpolation domain for scalar value in a bidimensional cartesian grid is illustrated: the green dot is the centroid, the red cross is the fine value that has to be calculated and the black dots are all coarse points that belong to the interpolation stencil.

The values in (3.10) can be analytically calculated; referring to figure 3.4 on the facing page, the following values are obtained:

$$\begin{aligned} \widehat{x} &= \Delta x \\ \widehat{y} &= \Delta y \\ \widehat{x^2} &= \frac{dx_x^2 - dx_{cen}^2}{12} + \Delta x^2 \\ \widehat{y^2} &= \frac{dy_y^2 - dj_{cen}^2}{12} + \Delta y^2 \\ \widehat{xy} &= \Delta x \Delta y \end{aligned} \tag{3.18}$$

where with the subscript cen are indicated cell dimensions for the centroid cell while with subscripts x and y are indicated cell lenghts for all the points that belong to the interpolation stencil. For momentum ρU the algorithm and the distances scheme is the same.

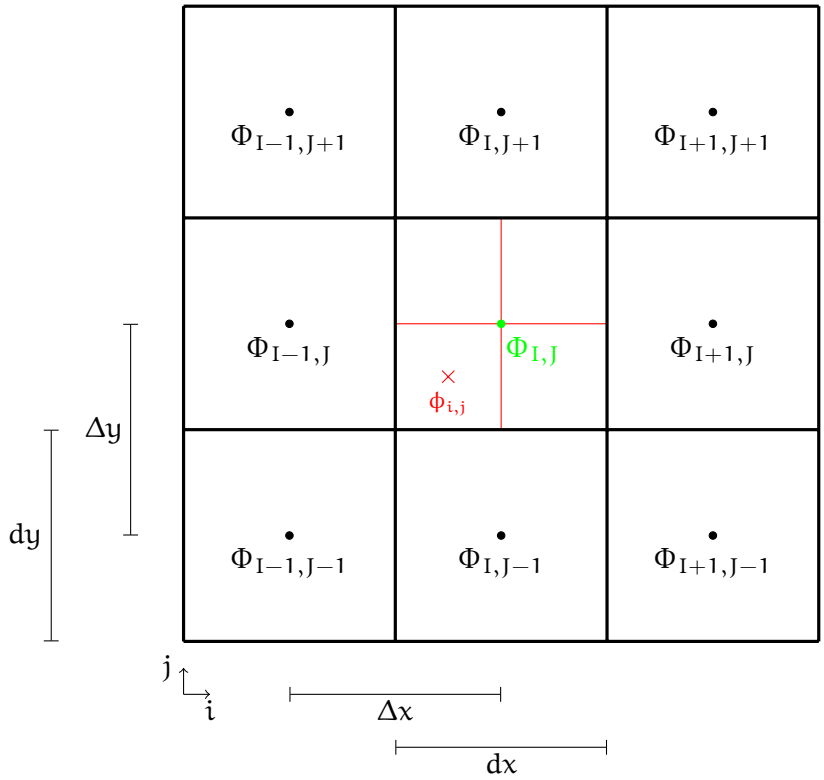

$\Phi_{I-1,J+1}$
$\Phi_{I,J+1}$
$\Phi_{I+1,J+1}$
$\Phi_{I-1,J}$
$\Phi_{I,J}$
$\Phi_{I+1,J}$
$\phi_{i,j}$
Δy
dy
$\Phi_{I-1,J-1}$
$\Phi_{I,J-1}$
$\Phi_{I+1,J-1}$
j
i
Δx
dx

Figure 3.4: LS

4 Validation

The validation is a fundamental phase for the development of a numerical code. In this chapter a simple three-dimensional test case and a more severe numerical simulation are presented.

4.1 Parabolic Flux Test Case

A three-dimensional test case is been performed: the test isn't very complex, so the simulation time and computational resources necessary aren't so high; furthermore, with a simple (but efficient) test case, simulation setup is very easy and the debuggin process is fast.

This test case selected is a simple tube, with an inlet at the right far end and and outlet at the other one, adiabatic walls and as initial condition the flow is fixed; at the inlet, a velocity profile (represented in figure 4.1c) is forced. The maximum velocity in z direction is $10\ \mathrm{m\,s^{-1}}$.

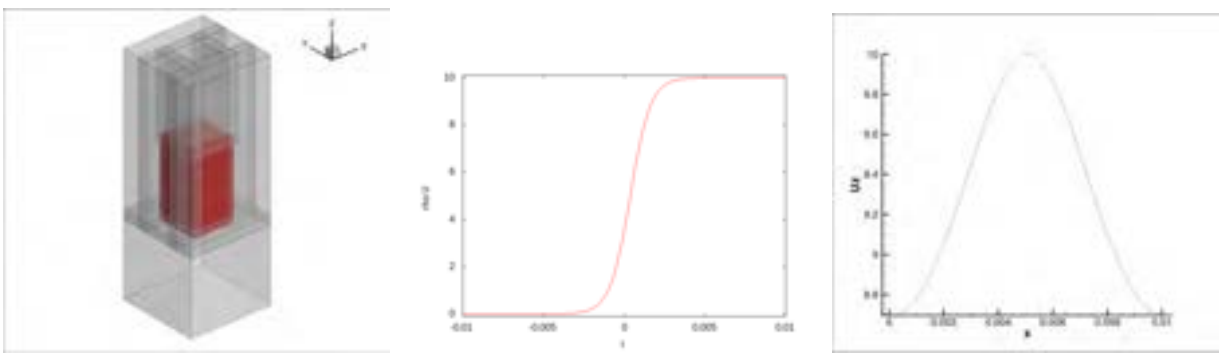
The numerical grid is composed by six different zones: the finest grid (red color in figure 4.1a) is fully surrounded by five coarse zones; in this way the connection operators can be tested in all the three directions.

The numerical test shows that the parabolic flow passes over the fine grid boundaries and doesn't change its shape and its intensity: all variables, in the coarse-fine interface have a continuous profile without any sort of spurioius oscillation.

4.2 LES of a Lean Premixed CH_4/H_2-Air Slot Flame

Instead to simulate the SANDIA Syngas Jet Flame "A" as written in the last year agreement between ENEA-COMSO and DIMA, a different flame is been simulated, because of the availability of large numerical data sets obtained from a DNS and a LES of this particular flame.

The test case consist in an unconfined and athmospheric Bunsen flame developing along z direction; the premixed flame is produced by three adjacent rectangular slot burners (with undefined size in the x direction) separated in the y direction by means of two 0.17 mm walls. From the central slot burner flows a mixture of methane, hydrogen and air (equvalent ratio $\Phi = 0.7$ and with 0.2 mole fraction of hydrogen) with a bulk velocity of $100\ \mathrm{m\,s^{-1}}$ and a temperature of 600 K, while from the lateral slots flows hot combusition product



(a) Computational domain (b) Inlet velocity profile versus time (c) Inlet velocity profile towards x (or y) direction

Figure 4.1: Geometric and inlet setup

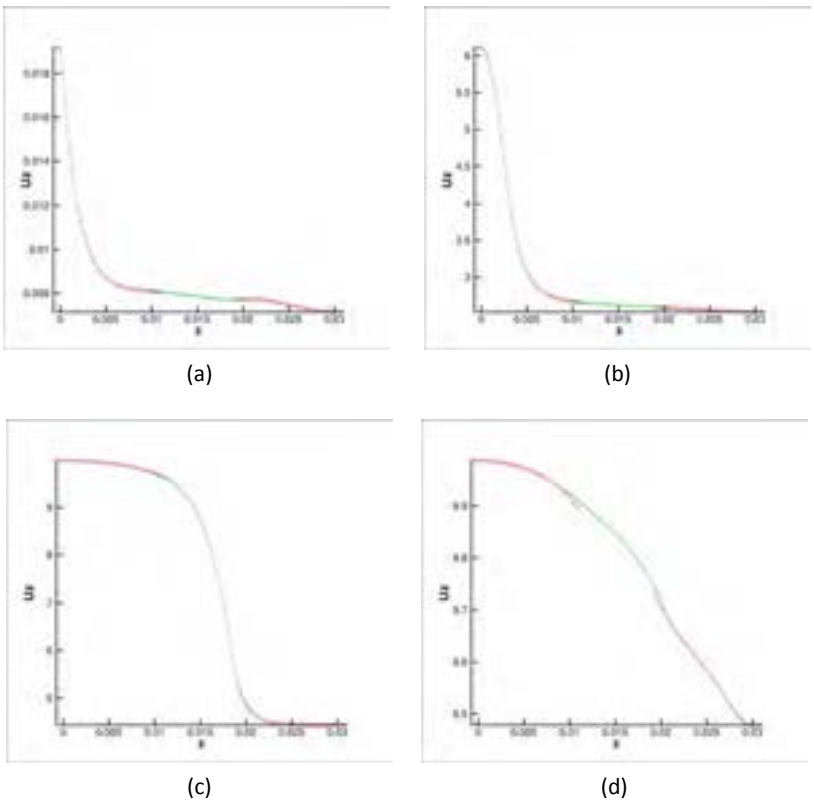


Figure 4.2: Flow fields along z direction

of the central mixture with a velocity of 25 m s^{-1}. The central Reynolds number is 2264, based on the 1.2 mm width of the jet, its bulk velocity, and the kinematic viscosity 5.3×10^{-5} m^2 s^{-1}.

This test was performed on a three-dimensional computational domain: the total lenght in the z direction is 28 mm, 4 mm for the injection zone and 24 mm for the main mixing and reacting zone, the total lenght in the y direction is 36 mm (the injection zone has an y dimension of 1.2 mm) and the total lenght in x direction is 3 mm. The domain is discretized by zones with different spatial resolution: the red ones in figure 4.4 on page 35 are the finest one, while the black ones are the coarsest.

The computational domain has 8 different zones: four fine and four coarse, with a refinement ratio of 2; in x direction the computational domain consist of 60 nodes for the fine zones and 30 nodes for the coarse ones; the central mixing and reacting zone has 216 nodes in z direction and 150 nodes in y direction.

In figure 4.5 on page 36 the temperature of the flow field is presented, with three different isosurfaces: also where there's the join between the two grids of different spatial resolution, the solution is good and there aren't any spurious oscillations; the figure 4.5b shows a particular of the join region, in the upper part of the central mixing and reacting zone, where the flame front takes place and the gradients of all fluidodinamic properties are higher: also here the solution quality is very good.

This correct behaviour is shown also in figure 4.6 on page 36, where the velocity in z direction of the flow field is presented, with three different isosurfaces.

The numerical simulation is still up and running and data collection is in progress: so will be possible to compare all numerical data obtained from this simulation with numerical data provided by the DNS and the LES simulation performed on the same test case, without mesh refinement technique.

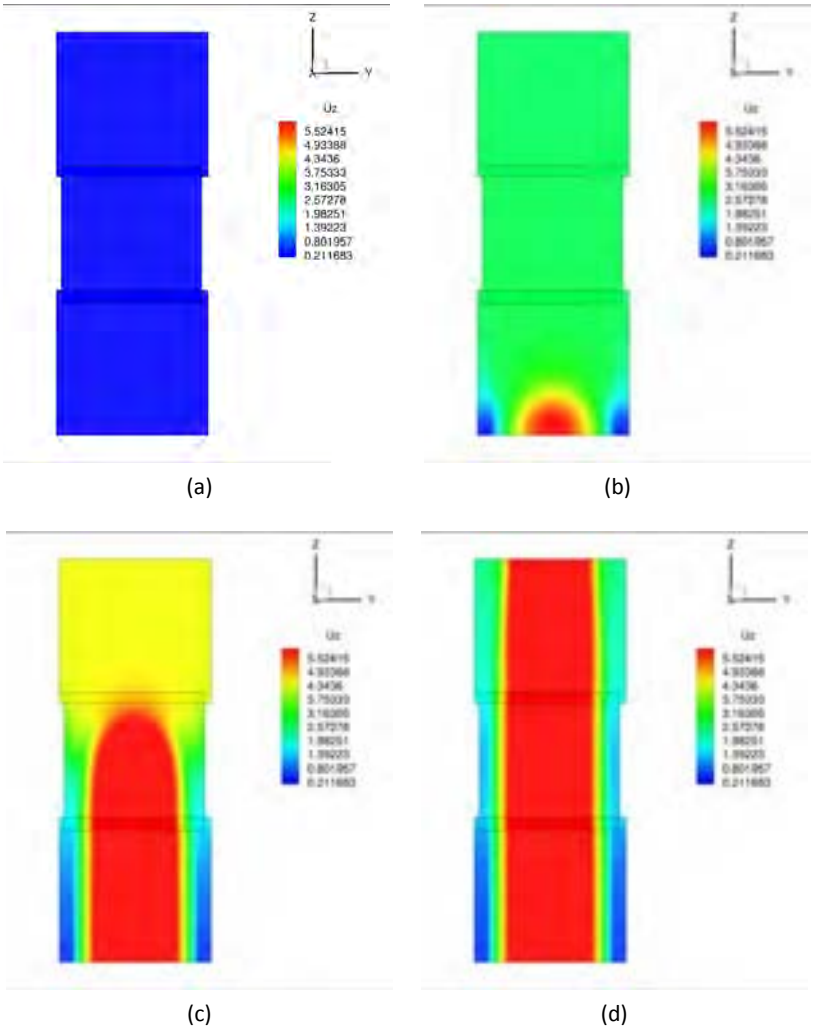


Figure 4.3: Flow fields along z direction (2D)

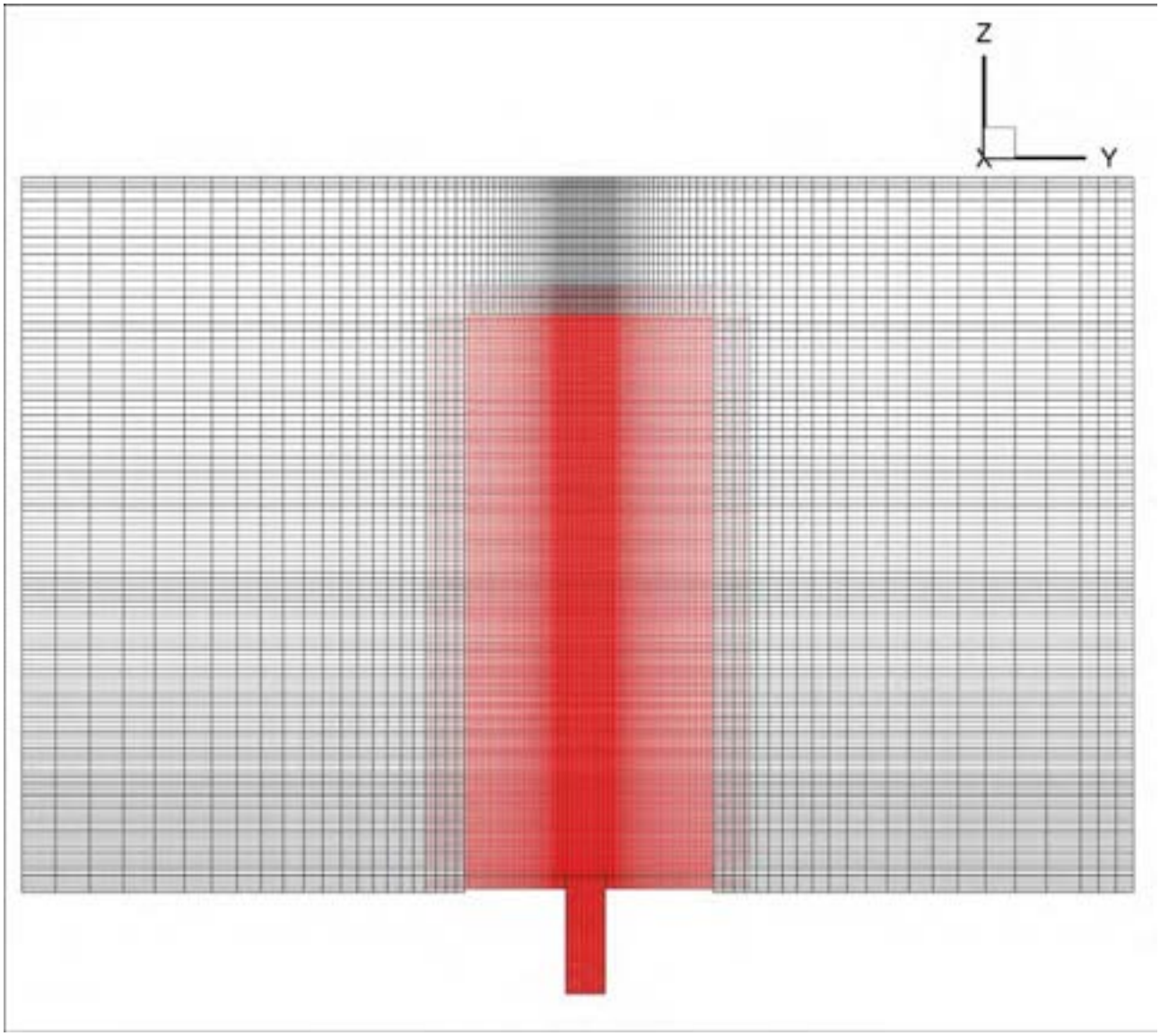
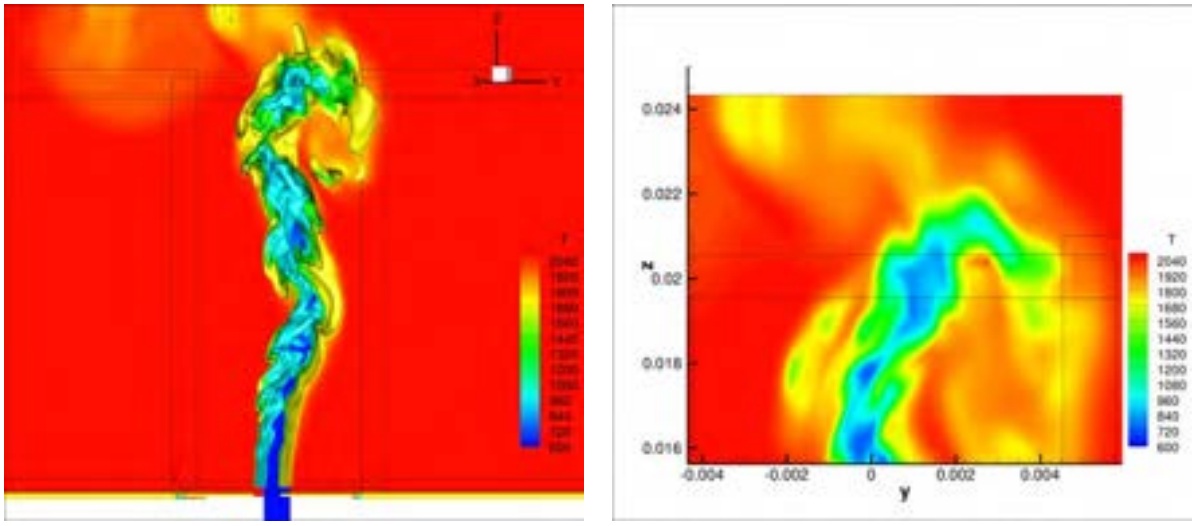
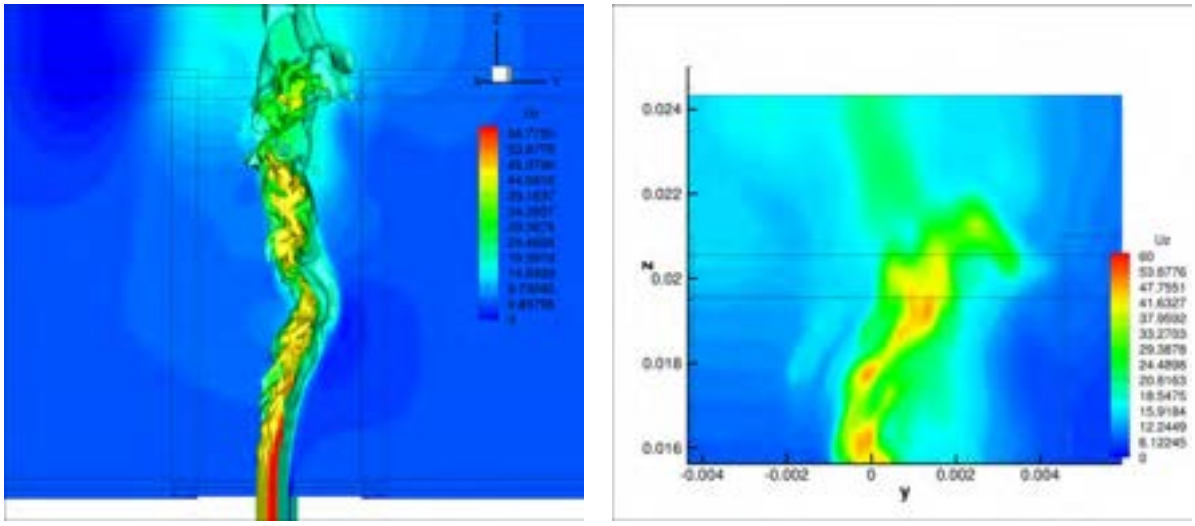


Figure 4.4: Computational Domain with Refined Grids



(a) (b)

Figure 4.5: Temperature Flow Field



(a) (b)

Figure 4.6: Velocity in z direction Flow Field

Bibliography

[1] F. Donato. *Numerical Modeling of Two-Phase Reactive Flows*. PhD thesis, University of Rome "Sapienza", Department of Electric Engineering, 2009.

[2] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977.

[3] M. Terracol, P. Sagaut, and C. Basdevant. A multilevel algorithm for large-eddy simulation of turbulent compressible flows. *Journal of Computational Physics*, 167:439–475, 2001.

[4] M. Terracol, P. Sagaut, and C. Basdevant. A time self-adaptive multilevel algorithm for large-eddy simulation. *Journal of Computational Physics*, 184:339–365, 2003.

[5] B. J. Boersma, M. N. Kooper, F. T. M. Nieuwstadt, and P. Wesseling. Local grid refinement in large-eddy simulations. *Journal of Engineering Mathematics*, 32:161–175, 1997.

[6] P. P. Sullivan, J. C. McWilliams, and C. Moeng. A grid nesting methos for large-eddy simulations of planetary boundary-layer flows. *Boundary Layer Metereology*, 80:167–202, 1996.

[7] C. A. Rendlemann, V. E. Beckner, and M. Lijewski. Parallelization of an adaptive mesh refinement method for low mach number combustion, 2001.

[8] M. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics*, 82:64–84, 1989.

[9] K. K. Kuo. *Principles of Combustion*. John Wiley and Sons Inc., 1986.

[10] J. C. Maxwell. Illustration of the dynamical theory of gases. part 1. on the motions and collisions of perfectly elastic spheres. *Phil. Mag.*, 19:19–32, 1860.

[11] J. C. Maxwell. Illustration of the dynamical theory of gases. part 2. on the process of diffusion of two or more kinds of moving particles among one other. *Phil. Mag.*, 20:21–37, 1860.

[12] J. Stefan. über das gleichgewicht und die bewegung, insbesondere die diffusion von gasmengen. *Sitzungsber. Akad. Wiss. Wien*, 63:63–124, 1871.

[13] M. S. Liou and C. J Steffen. A new flux splitting scheme. *Journal of Computational Physics*, 107:23–39, 1993.

[14] B. P. Leonard. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer Methods in Applied Mechanics and Engineering*, 19:59–98, 1979.

[15] C. W. Shu and S. Osher. Efficient implementation on essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 83:32–78, 1989.

[16] S. Nagarajan, S. K. Lele, and J. H. Ferziger. A robust high-order compact method for large eddy simulation. *Journal of Computational Physics*, 191:392–419, 2003.

[17] H. D. Thompson, B. W. Webb, and J. D. Hoffman. The cell reynolds number myth. *International Journal for Numerical Methods in Fluids*, 5(4):305–310, 2005.

[18] C. Ollivier-Gooch. Quasi-eno schemes for unstructured meshes based on unlimited data-depended least-squares recostruction. *Journal of Computational Physics*, 133:6–17, 1997.

[19] M. Sommerfeld and H. H. Qiu. Characterization of particle-laden, confined swirling flows by phase-doppler anemometry and numerical calculation. *Int. J. Multiphase Flow*, 19(6):1093–1127, 1993.

[20] S. M. Mitran. A comparison of adaptive mesh refinement approaches for large eddy simulation, 2001.

[21] M. Manhart. A zonal grid algorithm for dns of turbulent boundary layers. *Computers & Fluids*, 33:435–461, 2004.

[22] C. Safta, J. Ray, and N. N. Najm. A high-order amr algorithm for chemically reacting flows. July 2011. ICDERS 2011, Irvine, USA.

[23] C. Shen, J. Qiu, and A. Christlieb. Adaptive mesh refinement based on high order finite difference weno scheme for multiscale simulations. *Journal of Computational Physics*, 230:3780–3802, 2011.

[24] R. B. Bird, W. E. Stewart, and E. N. Lightfoot. *Transport phenomena*. John Wiley & Sons, Inc., New York, 1960.

[25] M. Sommerfeld and H. H. Qiu. Detailed measurements in a swirling particulate twophase flow by a phase-doppler anemometer. *Int Journal Heat and Fluid Flow*, 12:20–28, 1991.

[26] K. Hishida, K. Takemoto, and M. Maeda. Turbulent characteristics of gas-solid two-phase confined jet. *Japanese Journal of Multiphase Flow*, 1(1):56–69, 1987.

[27] J. Borée, T. Ishima, and I. Flour. The effect of mass loading and inter-particle collisions on the development of polydispersed two-phase flow downstream a confined bluff body. *Journal of Fluid Mechanics*, 443:129–165, 2001.

[28] A. Nejat and C. Ollivier-Gooch. A high-order accurate unstructured finite volume newton-krylov algorithm for inviscid compressible flows. *Journal of Computational Physics*, 227:2582–2609, 2008.

Giacomo Rossi

Curriculum Vitae et Studiorum

GR Vcard

Personal Informations

First name / Surname	Giacomo Rossi
Date of birth	January 29th 1982
Place of birth	Rome, Italy
Nationality	Italian
Gender	Male
Marital status	Married
Fiscal code	RSSGCM82A29H501W

Contact Informations

Address	Via delle Betulle 91, Rome, Italy
ZIP code	00171
Telephone(s)	+39 0692927207 Mobile: +39 3408816643
Email	rossigiacomo82@gmail.com

Education

	January 28th 2010
Degree	Master of Science, 2 years course (Laurea Specialistica), "Pegasus" Certified
Title of qualification awarded	Space Engineer
University	Sapienza, Università degli Studi di Roma
Final Grade	106/110
Scholarship	A.DI.SU. 2008/2009
Thesis title	Supersonic Choking during Solid Rocket Motors Ignition Transient
Tutors	Prof. Bernardo Favini[1]

[1] Associate professor at Dipartimento di Ingegneria Meccanica e Aerospaziale, Sapienza, Università degli Studi di Roma (bernardo.favini@uniroma1.it)

Rome, September 29, 2014

Foundamentals teachings	Gasdynamics, Numerical Fluid dynamics, Rocket Motors, Flight Mechanics, Aerospace Structure
	September 20th 2006
Degree	Bachelor of Science, 3 years course (Laurea)
Title of qualification awarded	Aerospace Engineer
University	Sapienza, Università degli Studi di Roma
Final Grade	97/110
Scholarship	A.DI.SU. 2005/2006
Thesis title	Downforce in Race Cars
Tutors	Prof. Giorgio Graziani[2]
Foundamentals teachings	Aerodynamics, Jet Propulsion, Mechanics, Construction Science, Informatics
	July 2000
Degree	High School Diploma, 5 years course
Title of qualification awarded	General Certificate of Secondary Scientific Education
High School	Tullio Levi Civita, Roma
Final Grade	100/100

Trainings

	From December 3rd to December 6th 2012
Training course	OpenFOAM Foundation & Advanced Course
Training center	The OpenFOAM Foundation
Foundamentals teachings	Fundamental and Advanced Course on OpenFOAM opensource CFD toolbox
	From January 31st to February 2nd 2012
Training course	Introduction to HPC Scientific Programming: tool and techniques
Training center	CASPUR, Consorzio Interuniversitario per le Applicazioni di Supercalcolo per Università e Ricerca
Foundamentals teachings	Computer Architectures, Optimization Techniques, Profiling and Debugging Tools, Scientific Libraries
	From October 25th to October 28th 2011
Training course	Python Scripting
Training center	CASPUR, Consorzio Interuniversitario per le Applicazioni di Supercalcolo per Università e Ricerca
Foundamentals teachings	Python & Shell Scripting, Regular Expressions, Python Steering & Monitoring, Numeric Python

[2]Full professor at Dipartimento di Ingegneria Meccanica e Aerospaziale, Sapienza, Università degli Studi di Roma (g.graziani@uniroma1.it)

Rome, September 29, 2014

	From May 17th to May 19th 2011
Training course	Introduction to Parallel Computing with MPI and OpenMP
Training center	CASPUR, Consorzio Interuniversitario per le Applicazioni di Supercalcolo per Università e Ricerca
Foundamentals teachings	Shared Memory Parallel Programming and Message Passing Paradigm
	From May 3rd to May 6th 2011
Training course	Scientific Computing in Fortran95
Training center	CASPUR, Consorzio Interuniversitario per le Applicazioni di Supercalcolo per Università e Ricerca
Foundamentals teachings	Scientific Computing in Fortran95
	From January 11st to January 13rd 2011
Training course	Introduction to UNIX/Linux
Training center	CASPUR, Consorzio Interuniversitario per le Applicazioni di Supercalcolo per Università e Ricerca
Foundamentals teachings	GNU/Linux fundamentals

Working Experiences

	From August 1st 2011 to July 31st 2015
Employer	Dipartimento di Meccanica e Aeronautica, Sapienza Università di Roma
Working Place	ENEA CRE "Casaccia", S. M. di Galeria, Rome, Italy
Qualification	Research fellow; the aim of the research project is to develop multi-resolution algorithm for Large Eddy Simulation of reacting flows
	From February 1st 2010 to 30th November 2010
Employer	Dipartimento di Ingegneria Meccanica e Aerospaziale (DIMA), Sapienza Università di Roma
Working Place	Rome, Italy
Qualification	Collaboration contractor with DIMA; the aim of the research project is to analize gasdynamic fields in Solid Rocket Motors combustion chambers
	From July 1st 1998 to August 31st 1998
Employer	Cartest s.r.l.
Working Place	Rome, Italy
Qualification	Collaboration contractor with Cartest; the aim of the project is to develop and manage an electronic database

Rome, September 29, 2014

Personal skills and competences

Computer skills and competences	
Languages	Fortran, Python, Bash scripting, C, LaTeX
Platforms	Unix, Linux, MS Windows (98, 2000, XP, Vista, 7, 8)
Concepts	Parallel Programming with OpenMP, MPI
Integrated Environments	Matlab, Matlab-Simulink (Scilab/Octave), MathCad, MS Visual Studio
Scientific Applications	Tecplot, Paraview, VTK framework
Office Applications	MS Office, OpenOffice
Computer Graphics	Autocad, Blender, Inkscape, The GIMP
Hobbies	
Hobbies	Freediving, Spearfishing, Running
Mother tongue	Italian
Other language	

Self-assessment European level(*)

	Understanding		Speaking		Writing
	Listening	Reading	Spoken interaction	Spoken production	
English	B1 Independent user	B2 Independent user	B1 Independent user	B2 Independent user	B2 Independent user

(*)Common European Framework of Reference (CEF) level

Publications

September 2014	About Multi-Resolution Techniques for Large Eddy Simulation of Reactive Multi-Phase Flows, G. Rossi, B. Favini, E. Giacomazzi, F. R. Picchia and N. M. S. Arcidiacono, 69th ATI Congress, Milan, Italy
October 2012	LES Simulation of a Devolatilization Experiment on the IPFR Facility, F. Donato, G. Rossi, B. Favini, E. Giacomazzi, D. Cecere, F. R. Picchia and N. M. S. Arcidiacono, 30th Meeting of the Italian Section of Combustion Institute, Milan, Italy
June 2011	On the NDP Onset in Pre-Ignition Transient of High Performance SRMs: VEGA Z9A Experience, B. Favini, E. Cavallini, G. Rossi, M. Di Giacinto, A. Di Mascio and F. Serraglia, 5th International Conference on Recent Advances in Space Technology, Istanbul, Turkiye
February 2011	Analysis of the Thermal Flow on the Surface of the Launcher VEGA, A. Di Mascio, R. Broglia, R. Muscari, G. Rossi, S. Zaghi, and B. Favini, ESA VG-ESA-SOW-004-IPT Technical Report

Rome, September 29, 2014

Ai sensi del DL 196/03 e della L. 675/96 autorizzo la Società destinataria del presente Curriculum Vitae al trattamento dei miei dati personali.

Giacomo Rossi

Rome, September 29, 2014

Bernardo Favini

Curriculum vitae

1954 nato a Roma il 14 novembre.

1980 laureato in Ingegneria Meccanica con 110/110 e lode, Facoltà di Ingegneria , Università di Roma *la Sapienza.*

1983 ricercatore in Fluidodinamica, ING-IND/06, presso il Dipartimento di Meccanica e Aeronautica della Facoltà di Ingegneria, Università di Roma *la Sapienza.*

1992 professore associato in Fluidodinamica, ING-IND/06, presso il Dipartimento di Meccanica e Aeronautica della Facoltà di Ingegneria, Università di Roma *la Sapienza.*

1998-2001 responsabile settore Applicazioni per il Consorzio Interuniversitario per le Applicazioni del Super Calcolo per l'Università e la Ricerca, CASPUR.

1998-2005 Membro del Collegio dei Docenti del Dottorato in Ingegneria Aerospaziale, Università di Roma *la Sapienza.*

2006 Membro del Collegio dei Docenti del Dottorato in Tecnologia e Scienza Aerospaziale, Università di Roma *la Sapienza.*

2012 Membro del Collegio dei Docenti del Dottorato in Matematica, Università di Roma la Sapienza.

Attività didattica

Docente di diversi corsi del raggruppamento di Fluidodinamica, in particolare: Gasdinamica, Fluidodinamica, Ipersonica, Magnetofluidodinamica, Gas Rarefatti, Aerotermochimica, Modellistica per la Propulsione a Solido, Metodi Numerici per le Equazioni alle Derivate Parziali Non-Lineari.

Docente del Master in Calcolo Scientifico, Università di Roma *la Sapienza,* in*:* Fluidodinamica Numerica e Fluidodinamica Ambientale.

Docente del Master in Sistemi di Trasporto Spaziale, Università di Roma *la Sapienza,* in: Gasdinamica Numerica.

Attività Scientifica

Ricerche svolte nell'ambito della Fluidodinamica con particolare interesse per: Metodi numerici per flussi comprimibili stazionari e non-stazionari in presenza di urti, Flussi ipersonici attorno a corpi di rientro, Flussi turbolenti reattivi, Gasdinamica dei motori a propellente solido.

13 pubblicazioni su riviste internazionali e 49 a congressi internazionali.

Responsabile di finanziamenti Murst, Convenzioni ENEA e AVIO, e partecipante a gruppi di ricerca di Convenzioni ASI, AVIO e ESA-Esrin, ESA-ESTEC.