1

CFD Numerical Models

This chapter defines two fluid mechanics models and two particular numerical approximations for their discretization, which will be used in the examples of mesh adaptation algorithms that constitute the main part of this book. We first consider compressible fluid flows and introduce an approximation method, referred in the sequel as mixed element volume (MEV), which relies mainly on a standard continuous P_1-Galerkin approximation. Its stabilization is obtained by introducing high-order Godunov upwinding. Second, we consider a multifluid model based on the incompressible Navier–Stokes equations and introduce an approximation method based on the continuous P_1-Galerkin approximation. Pressure stabilization is obtained by projection. Advective stabilization is obtained by introducing high-order upwinding.

1.1. Compressible flow

1.1.1. *Introduction*

The simulation of compressible flows experienced, in the 1990s, a small revolution with the development of new algorithms that are able to compute flows through (or around) any kind of shape. This was due to new numerical algorithms and mesh generation algorithms. For both types, the main innovation was related to unstructured meshes, and the way to do it was first to rely on tetrahedrizations. Unstructured mesh generation and in particular tetrahedrization has been the object of many research and advances, and we refer, for example, to previous studies (Frey and George 2008; Borouchaki and George 2017; George et al. 2019, 2020) for monographies presenting these methods. The "any kind of shape" slogan has been progressively completed by the adaptation to any kind of flow, including flows in moving meshes, and by automatic mesh adaptation, appearing as an important issue to address in order to improve the expected benefits from a numerical simulation.

Several numerical methods have been developed for computing compressible flows on unstructured meshes. First, low-order methods (typically second-order) were developed. Let us mention central-differenced cell-centered and vertex-centered finite-volume methods (Jameson 1987; Mavriplis 1997), Taylor–Galerkin methods (Donea et al. 1987; Löhner et al. 1984), least–square Galerkin methods (Hughes and Mallet 1986) and the distributive schemes (Deconinck et al. 1993) for second-order accurate methods. Higher order accurate schemes have then been developed; let us mention unstructured ENO methods (Abgrall 1994) and discontinuous Galerkin methods (Cockburn 2003; Cockburn and Shu 1989).

In the first part of this section, we present and discuss a mixed finite-element/finite-volume low-order discretization for the Euler models of aerodynamics applicable to a very general class of tetrahedrizations, and we consider a few crucial numerical issues for the application of an Euler scheme:

– mastering numerical dissipation;

– mastering positiveness;

– evaluating the synergy between such kind of numerics and high-performance mesh adaptation methods.

In the second part, the extension of the MEV method to Navier–Stokes and Reynolds-averaged Navier–Stokes is considered.

The MEV discretization method is a combination of a finite-element method (FEM) with a vertex-centered finite-volume method (FVM). Like any vertex-centered approximation, it enjoys the property of handling the smallest number of unknowns for a given mesh and the possibility to assemble the fluxes on an edge-based mode. The underlying FEM is the standard Galerkin method with continuous piecewise linear approximation on triangles or tetrahedra. The FEM is applied directly for discretizing second-order derivatives (diffusion or viscosity terms). For hyperbolic terms, the FEM needs extra stabilization terms that are derived from an upwind FVM. The underlying FVM is a vertex-centered edge-based method. The finite-volume cell is built around each vertex, generally by using medians (2D) or median planes (3D); advection terms are stabilized with upwinding or artificial dissipation, and second-order "viscous" terms are discretized with finite elements. Among the different ways of constructing second-order accurate upwind schemes, the MUSCL formulation introduced by van Leer (1979) for finite-volume methods is particularly attractive and has been generally chosen.

The family of upwind MEV schemes was initiated by Baba and Tabata (1981) for first-order upwind diffusion-convection models and by Fezoui et al. for Euler flows (see Fezoui 1985; Dervieux 1985, 1987; Fezoui and Stoufflet 1989; Fezoui and Dervieux 1989; Stoufflet et al. 1996). It has been studied by many CFD teams (see, in

particular, Whitaker et al. (1989); Anderson and Bonhaus (1994); Venkatakrishnan (1996); Barth (1994); Catalano (2002)). The framework proposed in Selmin and Formaggia (1998) can also be considered as an extension of MEV. Many developments and results relying on this family of schemes are regularly reported by Farhat and co-workers (Farhat and Lesoinne 2000). A particular advantage of MEV is its ability to perform well in combination with very irregular meshes. As a result, this scheme was identified as particularly convenient for developing methods for shape design (Farhat 1995; Nielsen and Anderson 2002; Vàzquez et al. 2004), for fluid–structure interaction with moving meshes (Farhat and Lesoinne 2000) and of course for anisotropic mesh adaptation (Loseille et al. 2007).

Several theoretical or methodological questions concerning MEV are addressed in this chapter:

– *Accuracy*. The basic scheme is introduced in section 1.1.2. In case of meshes with a bounded aspect ratio, the second-order accuracy of the underlying Galerkin method holds for steady-state problems, even for rather irregular meshes. For the unsteady case, since the mass matrix diagonalization is applied, the constraint on mesh regularity is somewhat stronger. But the behavior of the upwind versions of the MEV for highly stretched structured meshes is the main drawback of this class of schemes. Barth (1994) suggests a modification in the shape of finite-volume cells, which we describe in section 1.4.

– *Higher order*. Extension to second-order upwinding is based on a MUSCL formulation and is presented in section 1.1.3.

– *Superconvergent low dissipation versions*. In the case where the flow field under study is smooth, the numerical dissipation can be importantly reduced while not allowing Gibbs oscillations. In the linear theory, oscillations arise when high-frequency components of the solution are dispersed, that is, propagated with large phase velocity error, without enough dissipation to damp them. For reducing overall dissipation while avoiding oscillation, we follow the lines of higher order upwinding. This is obtained by introducing a new type of MUSCL reconstruction. Dissipation appears as relying on higher order even derivatives. Some versions of the new family show higher order convergence on regular or very smooth meshes. We call this property superconvergence. This method is presented in section 1.1.4.

– *Robustness and positivity*. In the 1980s, robustness of numerical schemes for hyperbolics was put in relation with positiveness, monotony and total variation diminishing properties. In section 1.1.6, we state several positivity results for MEV schemes.

– *Viscous flows*. The compressible study is completed in section 1.2 with the description of the numerical scheme for viscous and turbulent (statistical closure) flows.

1.1.2. *Spatial representation*

1.1.2.1. *Mathematical model*

We write the unsteady Euler equations as follows in the computational domain $\Omega \subset \mathbb{R}^3$:

$$\Psi(W) = \frac{\partial W}{\partial t} + \nabla \cdot \mathcal{F}(W) = 0 \quad \text{in } \Omega, \tag{1.1}$$

where $W = {}^t(\rho, \rho u, \rho v, \rho w, \rho E)$ is the vector of conservative variables. $\mathcal{F}(W) = (\mathcal{F}_1(W), \mathcal{F}_2(W), \mathcal{F}_3(W))$ is the convective flux:

$$\mathcal{F}_1(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho E + p)u \end{pmatrix}, \mathcal{F}_2(W) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (\rho E + p)v \end{pmatrix},$$

$$\mathcal{F}_3(W) = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (\rho E + p)w \end{pmatrix} \tag{1.2}$$

so that the state equation becomes

$$\frac{\partial W}{\partial t} + \frac{\partial \mathcal{F}_1(W)}{\partial x} + \frac{\partial \mathcal{F}_2(W)}{\partial y} + \frac{\partial \mathcal{F}_3(W)}{\partial z} = 0.$$

Here, ρ, p and E represent, respectively, density, thermodynamical pressure and total energy per mass unit. Symbols u, v and w stand for the Cartesian components of velocity vector $\mathbf{u} = (u, v, w)$. For a calorically perfect gas, we have

$$p = (\gamma - 1)\big(\rho E - \frac{1}{2}\rho|\mathbf{u}|^2\big), \tag{1.3}$$

where γ is constant. A weak formulation including boundary conditions of this system writes for $W \in V = \big[H^1(\Omega)\big]^5$[1] as follows:

1 Space H^1 (respectively, H^k) consists of measurable functions with square integrable derivative up to order one (respectively, k).

$$\forall \phi \in V, \quad (\Psi(W),\, \phi) \;=\; \int_{\Omega} \left(\phi\, \frac{\partial W}{\partial t} + \phi \nabla \cdot \mathcal{F}(W) \right) \, \mathrm{d}\Omega$$
$$- \int_{\Gamma} \phi\, \hat{\mathcal{F}}(W) \cdot \mathbf{n} \, \mathrm{d}\Gamma \;=\; 0, \qquad [1.4]$$

where Γ is the boundary of the computational domain Ω (Figure 1.1), $\mathbf{n}$ is the outward normal to Γ and the boundary flux $\hat{\mathcal{F}}$ contains the boundary conditions detailed in viscous case in section 1.2.2. We are interested by this unsteady formulation together with the steady one, in which the time derivative is not introduced.

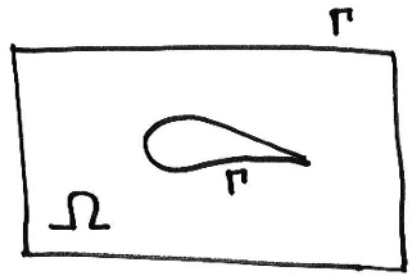

Figure 1.1. *A typical computational domain Ω limited by the two-component boundary $\Gamma = \partial\Omega$*

1.1.2.2. *Discrete variational representation*

We consider here the *steady* case, which is written as

$$\nabla \cdot \mathcal{F}(W) = 0$$

or in variational formulation:

$$\forall \phi \in V, \quad (\Psi(W),\, \phi) = \int_{\Omega} (\phi \nabla \cdot \mathcal{F}(W)) \; \mathrm{d}\Omega - \int_{\Gamma} \phi\, \hat{\mathcal{F}}(W) \cdot \mathbf{n} \, \mathrm{d}\Gamma \;=\; 0. \qquad [1.5]$$

The discretization chosen relies on two main choices. First, we consider a *tetrahedrization* as the discretization of the computational domain. This choice is made in connection with the progresses made for automatically generating and adapting meshes of this kind. Second, once the mesh is chosen, we have to put on it a set of nodes that are the geometrical supports of the degrees of freedom. The option chosen is the *set of vertices*. It is the option of the usual continuous P^1 FEM

approximation. It corresponds to the smallest number of nodes for a given mesh. Let $\mathcal{T}_h$ be a tetrahedrization of Ω, which is admissible for finite elements, that is, Ω is partitioned in tetrahedra, and the intersection of two different tetrahedra is either empty, or a vertex, or an edge, or a face. The test functions are taken into the approximation space V_h made up of continuous piecewise linear functions included in $V = [H^1(\Omega)]^5$:

$$V_h \;=\; \left\{ \phi_h \;\middle|\; \phi_h \text{ is continuous and } \phi_{h|T} \text{ is linear } \forall T \in \mathcal{T}_h \right\}^5.$$

In order to avoid the management of projectors applicable in the whole H^1 space, we work inside the following spaces:

$$\bar{V} = ([H^2(\Omega)]^5) \text{ and } \bar{V}_h = \bar{V} \cap V_h.$$

It is useful to introduce Π_h the corresponding P^1 interpolation operator:

$$\begin{aligned} &\Pi_h : \bar{V}_h \longrightarrow V_h \\ &\qquad \phi \longmapsto \Pi_h \phi \quad \text{with } \Pi_h \phi(i) = \phi(i) \;\; \forall i \text{ vertex of } \mathcal{T}_h. \end{aligned}$$

Then the discrete steady formulation of problem [1.5] is written as

$$\forall \phi_h \in V_h, \quad \int_\Omega \phi_h \nabla \cdot \mathcal{F}_h(W_h)\, d\Omega - \int_\Gamma \phi_h \hat{\mathcal{F}}_h(W_h) \cdot \mathbf{n}\, d\Gamma \;=\; 0, \qquad [1.6]$$

where $\mathcal{F}_h$ is by definition the P^1 interpolate of $\mathcal{F}$ in the sense that

$$\mathcal{F}_h(W) = \Pi_h \mathcal{F}(W) \;\text{ and }\; \mathcal{F}_h(W_h) = \Pi_h \mathcal{F}(W_h), \qquad [1.7]$$

and, as the operator $\mathcal{F}_h$ applies to the values of W at the mesh vertices, we have

$$\mathcal{F}_h(W) = \mathcal{F}_h(\Pi_h W) = \Pi_h \mathcal{F}(\Pi_h W). \qquad [1.8]$$

We get the same relations for $\hat{\mathcal{F}}_h(W)$:

$$\hat{\mathcal{F}}_h(W) = \Pi_h \hat{\mathcal{F}}(\Pi_h W) \;\text{ and }\; \hat{\mathcal{F}}_h(W_h) = \Pi_h \hat{\mathcal{F}}(W_h). \qquad [1.9]$$

Practically, this definition means that nodal flux values are written as $\mathcal{F}_h(\mathbf{x}_i) = \mathcal{F}_h(W(\mathbf{x}_i))$, where fluxes $\mathcal{F}_h$ are evaluated at the mesh vertices i. Discrete fluxes functions $\mathbf{x} \mapsto \mathcal{F}_h(\mathbf{x})$ are derived from the nodal values by P^1 intrapolation inside every element. In contrast to the standard Galerkin approach, this definition emphasizes that the discrete fluxes are in V_h.

1.1.2.3. *MEV basic equivalence*

The discrete formulation [1.6] can be transformed into a *vertex-centered finite-volume scheme* applied to tetrahedral unstructured meshes. This assumes a particular partition in control cells C_i of the discretized domain Ω_h:

$$\Omega_h = \bigcup_{i=1}^{n_c} C_i \ , \qquad [1.10]$$

each control cell being associated with a vertex i of the mesh. The corresponding test functions are the piecewise constant characteristic functions of cells:

$$\chi^i(\boldsymbol{x}) = \begin{cases} 1 \text{ if } \boldsymbol{x} \in C_i, \\ 0 \text{ otherwise.} \end{cases}$$

Then, using the Stokes formula, the finite-volume weak formulation (steady case) becomes for each vertex i, that is, for each cell C_i,

$$\int_{\partial C_i} \mathbf{n}_i \mathcal{F}(W)\, d\sigma = 0, \qquad [1.11]$$

where $\mathbf{n}_i$ holds for the unit normal to ∂C_i outpointing from C_i.

DEFINITION 1.1.– *Median cell*: The *dual finite-volume cell* is built by the rule of medians. In 2D, the median cell is limited by segments of medians between centroids and mid-edge (Figure 1.2). In 3D, each tetrahedron T of the mesh is split into four hexahedra[2] constructed around each of its four vertices. For a vertex i, the hexahedron $C_i \cap T$ is defined by the following points (Figure 1.3):

i) the three middle points of the edges issued from i;

ii) the three gravity centers of the faces containing i;

iii) the center of gravity of the tetrahedron;

iv) the vertex i.

2 The quadrilateral between a mid edge, the two neighboring face centroids and tetrahedron's centroid being on a plane.

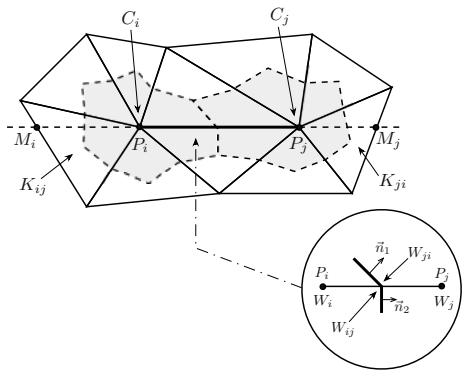

Figure 1.2. *Illustration of finite-volume cell construction in two dimensions with two neighboring cells, C_i and C_j, around i (P_i in the figure) and j (P_j in the figure), respectively, and of the upwind triangles K_{ij} and K_{ji} associated with the edge ij. Representation of the common boundary ∂C_{ij} with the solution extrapolated values for the MUSCL type approach. Dash lines are segments of medians of the triangles*

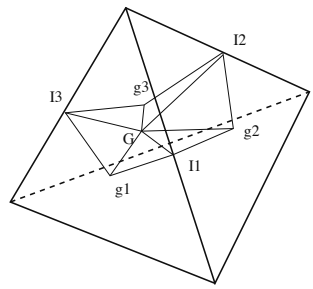

Figure 1.3. *The planes which delimit the finite-volume cell (related to upper vertex) inside a tetrahedron (3D case). G is the tetrahedron centroid, g_ks are face centroids and I_ks are edge centers*

The cell C_i of vertex i is the collection of all hexahedra linked to i. The common boundary $\partial C_{ij} = \partial C_i \cap \partial C_j$ between two neighboring cell C_i and C_j is divided into several triangular interface facets. □

An illustration of this construction is shown in Figure 1.4 for the 3D case.

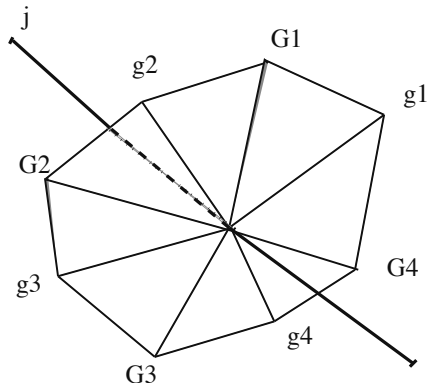

Figure 1.4. *Illustration of finite-volume cell interface ∂C_{ij} between two neighboring cells C_i and C_j (3D case)*

The finite-volume fluxes between cells around vertices i and j are integrated through the common boundary ∂C_{ij} with a value of $\mathcal{F}_h$ equal to the half-sum of $\mathcal{F}_h(W_i)$ and $\mathcal{F}_h(W_j)$:

$$\Phi_{ij}^{MEV} = \frac{\mathcal{F}_h(W_i) + \mathcal{F}_h(W_j)}{2} . \nu_{ij}, \qquad [1.12]$$

where ν_{ij} denotes the integral of the normal $\mathbf{n}_i$ to common boundary between cells C_i and C_j,

$$\nu_{ij} = \int_{\partial C_{ij}} \mathbf{n}_i \, d\sigma$$

and $W_i = W(i)$. The finite-volume formulation for an internal vertex i writes as the sum of all the fluxes evaluated from the vertices j belonging to $V(i)$ where $V(i)$ is the set of all neighboring vertices of i. Taking into account the boundary fluxes, the discrete scheme [1.6] then writes:

$$\sum_{j \in V(i)} \Phi_{ij}^{MEV} \; - \; \int_{\Gamma \cap \partial C_i} \bar{\mathcal{F}}_h(W_h) \cdot \mathbf{n} \, d\Gamma \; = \; 0. \qquad [1.13]$$

We obtain a vertex-centered finite-volume approximation, which is P^1-exact with respect to the flux function $\mathcal{F}_h$. This scheme enjoys most of the accuracy properties

of the Galerkin method (Mer 1998), such as the second-order accuracy on any mesh for diffusion-convection models. However, it lacks stability and cannot be applied to purely hyperbolic models such as the Euler equations.

1.1.2.4. *Flux integration*

Once the cells are defined, the spatial divergence $div\mathcal{F}$ is transformed via the Stokes formula into integrals of normal fluxes $\mathcal{F}.\mathbf{n}$ at cell boundaries. In the proposed family of schemes, the accuracy of the integration quadrature on cell boundaries is not as crucial: we choose a very simple option, the *edge-based integration*. On the contrary, flux integration sets the important problem of scheme stabilization. The variables are assumed to be constant by cell, and therefore, they are discontinuous from a cell to its neighbor. Upwind integration will rely on the Godunov method based on the two different values at each side of the discontinuity.

1.1.2.4.1. Central differencing

Let us write a vertex-centered central differenced finite-volume scheme for the steady Euler equations applied to an unstructured mesh as follows:

$$\begin{aligned} \mathbf{\Psi}_h(\Gamma, W)_i &= 0, \quad \text{with} \\ \mathbf{\Psi}_h(\Gamma, W)_i &= \sum_{j \in V(i)} \Phi^{central}(W_i, W_j, \nu_{ij}) + \mathbf{B}_h(\Gamma, W)_i, \end{aligned} \qquad [1.14]$$

where $V(i)$ is the set of vertices that are neighbors of i, and ν_{jk} is the integral on interface between j and k of the normal vector. Symbol $\mathbf{B}_h(\Gamma, W)_i$ represents boundary fluxes in which Euler fluxes take into account the available boundary information. The centered integration for elementary flux Φ is written as follows:

$$\Phi^{central}(W_i, W_j, \nu_{ij}) = 0.5(\mathcal{F}_i + \mathcal{F}_j) \cdot \nu_{ij}, \qquad [1.15]$$

where $\mathcal{F}_i = \mathcal{F}(W_i)$ are the Euler fluxes computed at W_i. This is equivalent to introduce the following discrete space operator ∇_h^*:

$$\nabla_h^*(\mathcal{F})_i = \sum_{j \in V(i)} 0.5(\mathcal{F}_i + \mathcal{F}_j) \cdot \nu_{ij} \; / \; a(i), \qquad [1.16]$$

where $a(i)$ is the measure of cell C_i.

1.1.2.4.2. Godunov differencing

Godunov-type methods rely on the discontinuous representation of the unknowns at cell interfaces and on the computation of the fluxes at these discontinuities in

function of both "left" and "right" values through the application of an approximate or an exact Riemann solver. This process introduces numerical viscosity terms that are very useful for stabilizing most flow calculations. First, we consider that W is constant by cell equal to W_i in C_i[3]. We write a vertex-centered first-order Godunov scheme for the Euler equations applied to an unstructured mesh as follows:

$$\mathbf{\Psi}_h(\Gamma, W)_i \;=\; \sum_{j \in V(i)} \Phi^{ARS}(W_i, W_j, \nu_{ij}) \;+\; \mathbf{B}_h(\Gamma, W)_i. \qquad [1.17]$$

Here, $\Phi^{ARS}(W_i, W_j, \nu_{ij})$ is evaluated by an approximate Riemann solver.

Roe approximate Riemann solver. A standard option is the Roe flux difference splitting (Roe 1981):

$$\Phi^{Roe}(W_i, W_j, \nu_{ij}) \;=\; 0.5(\mathcal{F}_i + \mathcal{F}_j) \cdot \nu_{ij} \;\;+\;\; 0.5|\mathcal{A}|(W_i - W_j) \qquad [1.18]$$

where $|\mathcal{A}|$ is the absolute value of the Jacobian flux along ν_{ij}:

$$\begin{aligned} &\mathcal{A} = \left(\frac{\partial \mathcal{F}}{\partial W}\right)_1 (\nu_{ij})_1 \;+\; \left(\frac{\partial \mathcal{F}}{\partial W}\right)_2 (\nu_{ij})_2 \;+\; \left(\frac{\partial \mathcal{F}}{\partial W}\right)_3 (\nu_{ij})_3 \quad \text{(3D case)} \\ &\mathcal{A} = T \Lambda T^{-1}\,, \;\; \Lambda \;=\; \text{diagonal eigenvalues matrix}, \\ &|\mathcal{A}| = T|\Lambda|T^{-1}. \end{aligned} \qquad [1.19]$$

These matrices are computed at an intermediate value $\overline{W}_{ij}$ of W_i and W_j; in short, we have:

$$\overline{W}_{ij} \;=\; (\rho_i^{\frac{1}{2}} W_i + \rho_j^{\frac{1}{2}} W_j)/(\rho_i^{\frac{1}{2}} + \rho_j^{\frac{1}{2}})$$

which enjoys the following property:

$$\mathcal{F}(W_i) - \mathcal{F}(W_j) \;=\; \mathcal{A}(\overline{W}_{ij})(W_i - W_j).$$

In the fully supersonic cases where $\mathcal{A}(\overline{W}_{ij}) \;=\; |\mathcal{A}(\overline{W}_{ij})|$ or $\mathcal{A}(\overline{W}_{ij}) \;=\; -|\mathcal{A}(\overline{W}_{ij})|$, Roe's splitting is fully upwind. By the hyperbolicity

3 In the MUSCL method, we consider that the mean value in cell C_i is identical to the value at vertex i, an approximation bringing simplification but not permitting the extension to very high order.

assumption, matrix $\mathcal{A}(\overline{W}_{ij})$ can be diagonalized. The absolute value $|\mathcal{A}(\overline{W}_{ij})|$ is given as:

$$|\mathcal{A}(\overline{W}_{ij})| \;=\; TDiag(|\lambda_1|,|\lambda_2|,|\lambda_3|,|\lambda_4|,|\lambda_5|)T^{-1} \;=\; sign(\mathcal{A}(\overline{W}_{ij}))\mathcal{A}(\overline{W}_{ij})\;,$$

where $sign(\mathcal{A}) = TDiag(sign(\lambda_1), sign(\lambda_2), sign(\lambda_3), sign(\lambda_4), sign(\lambda_5))T^{-1}$. Thus, this averaging also permits the following equivalent formulation:

$$\begin{aligned}\Phi^{Roe}(W_i,W_j,\nu_{ij}) \;=\; & 0.5(\mathcal{F}_i+\mathcal{F}_j).\nu_{ij} \\ & +0.5sign(\mathcal{A}(\overline{W}_{ij}))(\mathcal{F}(W_j)-\mathcal{F}(W_i)).\end{aligned} \qquad [1.20]$$

HLLC approximate Riemann solver. The idea of the HLLC[4] solver (following Toro 1999) is to consider locally a simplified Riemann problem with two intermediate states depending on the local left and right states. The simplified solution to the Riemann problem consists of a contact wave with a velocity S_M and two acoustic waves, which may be either shocks or expansion fans. The acoustic waves have the smallest and the largest velocities (S_i and S_j, respectively) of all the waves present in the exact solution. If $S_i > 0$, then the flow is supersonic from left to right and the upwind flux is simply defined from $F(W_i)$ where W_i is the state to the left of the discontinuity. Similarly, if $S_j < 0$, then the flow is supersonic from right to left and the flux is defined from $F(W_j)$ where W_j is the state to the right of the discontinuity. In the more difficult subsonic case when $S_i < 0 < S_j$, we have to calculate $F(W_i^\star)$ or $F(W_j^\star)$. Consequently, the HLLC flux is given by:

$$\Phi_{ij}^{HLLC}(W_i,W_j,\mathbf{n}_{ij}) = \begin{cases} F(W_i)\cdot\mathbf{n}_{ij} & \text{if } S_i>0 \\ F(W_i^\star)\cdot\mathbf{n}_{ij} & \text{if } S_i\le 0<S_M \\ F(W_j^\star)\cdot\mathbf{n}_{ij} & \text{if } S_M\le 0\le S_j \\ F(W_j)\cdot\mathbf{n}_{ij} & \text{if } S_j<0 \end{cases}. \qquad [1.21]$$

where $W_i^\star$ and $W_j^\star$ are evaluated as follows. Let us denote $\eta = \mathbf{u}\cdot\mathbf{n}$. Assuming that $\eta^\star = \eta_i^\star = \eta_j^\star = S_M$, the following evaluations are proposed (Batten et al. 1997) (the subscripts $_i$ and $_j$ are omitted for clarity):

$$W^\star = \frac{1}{S-S_M}\begin{pmatrix}\rho\,(S-\eta)\\ \rho\mathbf{u}\,(S-\eta)+(p^\star-p)\mathbf{n}\\ \rho E\,(S-\eta)+p^\star S_M-p\eta\end{pmatrix} \text{ where } p^\star=\rho\,(S-\eta)(S_M-\eta)+p.$$

4 From Harten et al. (1983) for contact discontinuities.

A key feature of this solver is in the definition of the three waves velocity. For the contact wave, we consider:

$$S_M = \frac{\rho_j \eta_j (S_j - \eta_j) - \rho_i \eta_i (S_i - \eta_i) + p_i - p_j}{\rho_j (S_j - \eta_j) - \rho_i (S_i - \eta_i)},$$

and the acoustic wave speeds based on the Roe average (denoted by $^-$):

$$S_i = \min(\eta_i - c_i, \overline{\eta} - \overline{c}) \quad \text{and} \quad S_j = \max(\eta_j + c_j, \overline{\eta} + \overline{c}).$$

With such waves velocities, the approximate HLLC Riemann solver has the following properties (Toro 1999): it automatically (i) satisfies the entropy inequality, (ii) resolves isolated contacts exactly, (iii) resolves isolated shocks exactly and (iv) preserves positivity of ρ, T, p.

1.1.3. *Spatial second-order accuracy: MUSCL*

The above schemes with Roe or HLLC are spatially first-order accurate. First-order upwind schemes of Godunov type enjoy a lot of interesting qualities, and in particular HLLC enjoys formally monotonicity or, in the case of the Euler model ρ-, T- and p-positivity. They can be extended to second order by applying the MUSCL method. Indeed, the fact that the Godunov method builds fluxes between cells with unknown variables constant by cells implies first-order accuracy. van Leer (1979, 1977) proposed to reconstruct a linear interpolation of the variables inside each cell and then to introduce in the Riemann solver the boundary values of these interpolations. Further, the slopes used for linear reconstruction can be limited in order to represent the variable without introducing new extrema. The resulting MUSCL method produces positive second-order schemes. We describe now an extension of MUSCL to unstructured triangulations with dual cells. The MUSCL ideas also apply to reconstructions which are different on each interface between cells, or equivalently on each edge. Several slopes of a dependant variable F are defined on the two vertices i and j of an edge ij as follows:

1) *Gradients*. First, the *centered gradient* $(\nabla F)^c_{ij}$ is defined as $(\nabla F)^c_{ij} \cdot \vec{ij} = F_j - F_i$.

We consider a couple of two triangles inm and jrs (2D) or two tetrahedra $inmo$ and $jrst$ (3D), one having i as a vertex and the second having j as a vertex. With reference to Figure 1.5, we define ϵ_{in}, ϵ_{im} and ϵ_{io} (respectively, ϵ_{jr}, ϵ_{js} and ϵ_{jt}) as the

components of vector $\vec{ji}$ (respectively, $\vec{ij}$) in the oblique system of axes $(\vec{in}, \vec{im}, \vec{io})$ (respectively, $(\vec{jr}, \vec{js}, \vec{jt})$):

$$\vec{ji} = \epsilon_{in}\,\vec{in} + \epsilon_{im}\,\vec{im}, + \epsilon_{io}\,\vec{io},$$

$$\vec{ij} = \epsilon_{jr}\,\vec{jr} + \epsilon_{js}\,\vec{js} + \epsilon_{js}\,\vec{js}.$$

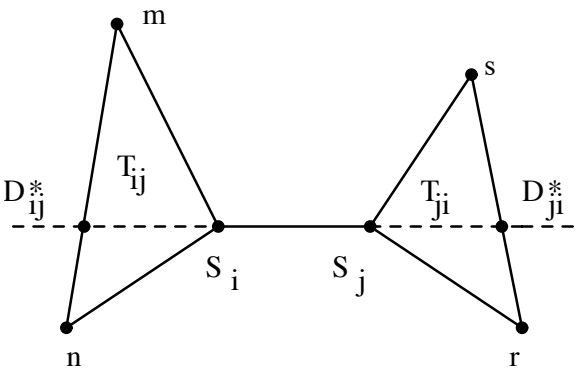

Figure 1.5. *Butterfly molecule in 2D: localization of the extra interpolation points D^*_{ij} and D^*_{ji} of nodal gradients. This allows to evaluate three derivatives along direction $ij = S_iS_j$, namely with D^*_{ij} and S_i, or S_i and S_j, or S_j and D^*_{ji}*

We say that T_{ij} and T_{ji} are upwind and downwind elements with respect to edge ij if the components $\epsilon_{in}, \epsilon_{im}, \epsilon_{io}, \epsilon_{jr}, \epsilon_{js}, \epsilon_{jt}$, are all non-negative:

$$T_{ij} \text{ upstream and } T_{ji} \text{ downstream } \Leftrightarrow Min(\epsilon_{in}, \epsilon_{im}, \epsilon_{io}, \epsilon_{jr}, \epsilon_{js}, \epsilon_{jt}) \geq 0. \quad [1.22]$$

The *upwind gradient* $(\nabla W)^u_{ij}$ is computed as the usual finite-element gradient on T_{ij} and the *downwind gradient* $(\nabla W)^d_{ij}$ on T_{ji}. This is written as:

$$(\nabla W)^u_{ij} = \nabla W\,|_{T_{ij}} \text{ and } (\nabla W)^d_{ij} = \nabla W\,|_{T_{ji}} \text{ where } \nabla W\,|_T = \sum_{k\in T} W_k \nabla \Phi_k|_T$$

are the P1-Galerkin gradients on triangle T.

2) *Interpolation at cell interface*. We now specify our method for computing the *interpolation slopes* $(\nabla W)_{ij}$ and $(\nabla W)_{ji}$:

$$(\nabla W)_{ij} \cdot \vec{ij} = (1-\beta)(\nabla W)^c_{ij} \cdot \vec{ij} + \beta(\nabla W)^u_{ij} \cdot \vec{ij}. \quad [1.23]$$

The computation of W_{ji} is analogous:

$$(\nabla W)_{ji} \cdot \vec{ij} = (1-\beta)(\nabla W)^c_{ij} \cdot \vec{ij} + \beta(\nabla W)^d_{ij} \cdot \vec{ij}. \quad [1.24]$$

The coefficient β is an upwinding parameter that controls the combination of fully upwind and centered slopes and that is generally taken equal to $1/3$, according to the error analysis in Table 1.1.

Scheme	β	ξ^c	ξ^d	Spatial order	CFL_{max}
RK4(0.11, 0.2766, 0.5, 1)	1/3	0	0	3	1.9
RK6 (Abalakin et al. 2002a)	1/3	- 1/30	- 2/15	5	1.867
RK3-SSP	1/3	0	0	3	2

Table 1.1. *Superconvergent orders and maximal Courant numbers for MUSCL-third-order ($\xi^c = \xi^d = 0$) and V6 spatial schemes (1D analysis). The RK4 first and second coefficients are optimized for higher CFL with MUSCL*

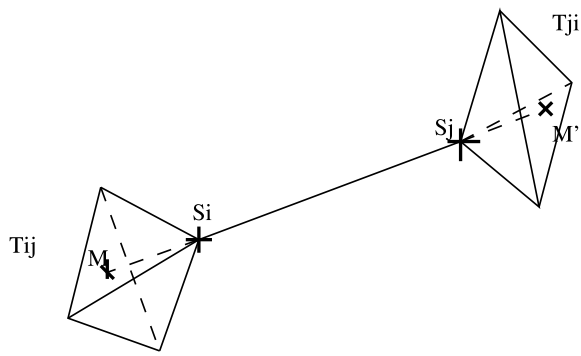

Figure 1.6. *Butterfly molecule in 3D: downwind and upwind tetrahedra are tetrahedra having, respectively, S_i and S_j as a vertex and such that line S_iS_j intersects the opposite face*

3) *Flux balance*. The scheme description is completed by replacing the first-order formulation [1.17] by the following flux balance:

$$\begin{array}{l} \mathbf{\Psi}_h(\Gamma, W)_i = \sum_{j \in V(i)} \Phi^{ARS}(W_{ij}, W_{ji}, \nu_{ij}) \; + \; \mathbf{B}_h(\Gamma, W)_i \\ \text{with :} \\ W_{ij} = W_i + \frac{1}{2}(\nabla W)_{ij}.\vec{ij} \quad , \quad W_{ji} = W_j + \frac{1}{2}(\nabla W)_{ji}.\vec{ji}. \end{array} \quad [1.25]$$

REMARK 1.1.– The numerical viscosity introduced by the MUSCL method is equivalent to a fourth-order derivative, weighted by a $O(h^3)$ coefficient, where h is local mesh size. When using certain very stretched meshes, the combination between MUSCL and median cells can produce a severe loss in accuracy. A cure for it is discussed in Annex 1 (section 1.4).

For this second-order version, the amount of dissipation, which is introduced, is the dominant term of the numerical error and may seem larger than needed in many applications.

1.1.4. *Low dissipation advection schemes*

The approximation described in the previous section is spatially second-order accurate. Note that the method combines finite differences in the local reconstruction and finite volume for fluxes. In contrast to reconstruction based on mean values (e.g. ENO schemes), a higher-order accurate interpolation does not bring a higher accuracy for the scheme. More precisely, for a nonlinear flux function, the accuracy of MUSCL schemes is limited to second order, as remarked by Wu and Wang (1995).

We now examine how to moderately change the reconstruction in order to improve the scheme. We get inspired by a direct simulation technique in which non-dissipative high-order approximations are stabilized in good accuracy conditions because of filters, which rely on very-high even order derivatives. In order to do this, we have to further extend the discretization stencil. Then it can be also interesting to choose a stencil extension, which also improves dispersion properties, since a less dispersive scheme needs less dissipation to avoid Gibbs-like oscillations. This leads to the idea of *superconvergent advection schemes* in the sense that they are of higher order (than two) on Cartesian mesh regions and for simpler models, like linear hyperbolic ones. More sophisticated versions giving high-order accuracy for nonlinear advection are described in Magoules (2011).

1.1.4.1. *Spatial scheme*

This section defines a low dissipation/low dispersion scheme introduced in Debiez (1996) and Debiez and Dervieux (1999). This scheme can be called *superconvergent* because it is designed in such a way that it is a higher-order scheme when applied to a Cartesian mesh. The scheme is built as follows:

1) A background flow $W = (\rho, \rho u, \rho v, E)$ on each vertex of the mesh is given.

2) Compute the primitive variable $\tilde{U} = (\rho, u, v, p)$ on each vertex (vertexwise loop).

3) Compute the nodal gradients $\nabla\tilde{U}$.

$$(\nabla\tilde{U})_i \;=\; \frac{1}{meas(C_i)} \sum_{T\,\in\, C_i} \frac{meas(T)}{3}\Big(\sum_{k\,\in\, T} (\tilde{U})_k\, \nabla\, \Phi_k|_T\Big) \quad \text{(2D case).} \qquad [1.26]$$

4) Start *edgewise assembly loop* by computing the extrapolated slopes:

$$\begin{aligned}(\nabla\tilde{U})_{ij}\cdot\vec{ij} = {} & (1-\beta)(\nabla\tilde{U})^c_{ij}\cdot\vec{ij} + \beta(\nabla\tilde{U})^d_{ij}\cdot\vec{ij} \\ & +\xi_c\left[(\nabla\tilde{U})^u_{ij}\cdot\vec{ij} - 2(\nabla\tilde{U})^c_{ij}\cdot\vec{ij} + (\nabla\tilde{U})^d_{ij}\cdot\vec{ij}\right] \\ & +\xi_d\left[(\nabla\tilde{U})_M\cdot\vec{ij} - 2(\nabla\tilde{U})_i\cdot\vec{ij} + (\nabla\tilde{U})_j\cdot\vec{ij}\right] \;,\end{aligned} \qquad [1.27]$$

where $(\nabla\tilde{U})_M$ is the gradient at the point M, intersection of line ij with the face of T_{ij} which does not contain i as a vertex, as shown in Figure 1.6. The expression is analog for $\nabla(\tilde{U})_{ji}$. Let us define left and right variable interpolations:

$$\tilde{U}_{ij} \;=\; \tilde{U}_i + \nabla\tilde{U}_{ij} \;\;;\;\; \tilde{U}_{ji} \;=\; \tilde{U}_j - \nabla\tilde{U}_{ji},$$

and recover the left and right values of conservative variables $W_{ij} = W(\tilde{U}_{ij}), W_{ji} = W(\tilde{U}_{ji})$. The upwind differenced flux is then written as follows (the index V6 holds for the numerical viscosity by sixth-order derivatives):

$$\mathbf{\Psi}^{V6}_h(\Gamma, W)_i \;=\; \sum_{j\in V(i)} \Phi^{ARS}(W_{ij}, W_{ji}, \nu_{ij}) \;+\; \mathbf{B}_h(\Gamma, W)_j. \qquad [1.28]$$

1.1.5. *Time advancing*

After semi-discretization of the unsteady Euler equation using the above scheme, by applying mass lumping we obtain

$$meas(C_i)\frac{dW_i}{dt} + \mathbf{\Psi}_h(\Gamma, W)_i = 0. \qquad [1.29]$$

The standard Runge–Kutta scheme can be used for time advancing the solution:

$$\begin{aligned} V_1 &= \Delta t\ \Psi(W^n)_i \\ V_2 &= \Delta t\ \Psi(W^n + V_1/2)_i \\ V_3 &= \Delta t\ \Psi(W^n + V_2/2)_i \\ V_4 &= \Delta t\ \Psi(W^n + V_3)_i \\ W_i^{n+1} &= W_i^n + V_1/6 + V_2/3 + V_3/3 + V_4/6, \end{aligned} \quad [1.30]$$

where $\Psi(W)_i = -\mathbf{\Psi}_h(\Gamma, W)_i / meas(C_i)$.

In many case, a linearized version of the time stepping can be used. Jameson (1993) writes as follows (N-stage version):

$$\begin{aligned} W^{(0)} &= W^n \\ W^{(k)} &= W^{(0)} + \frac{\Delta t}{N-k+1} \Psi\left(W^{(k-1)}\right), \quad k = 1 \ldots N \\ W^{n+1} &= W^{(N)}. \end{aligned} \quad [1.31]$$

An A-stability analysis as in Hirsch (1988) can be applied. We give in Table 1.1 some typical maximal CFL numbers for the six-stage RK scheme, which ensure a global accuracy order of five for the two best schemes of the proposed family. The RK3-SSP is written as

$$U^{(1)} = U^n + \Delta t\, L(U^n), \quad [1.32]$$

$$U^{(2)} = \frac{3}{4}U^n + \frac{1}{4}U^{(1)} + \frac{1}{4}\Delta t\, L(U^{(1)}), \quad [1.33]$$

$$U^{n+1} = \frac{1}{3}U^n + \frac{2}{3}U^{(2)} + \frac{2}{3}\Delta t\, L(U^{(2)}) \quad [1.34]$$

(see Shu and Osher 1988; Spiteri and Ruuth 2002). Table 1.1 illustrates that the above schemes can be used with CFL number of the order of unity. In the sequel, we consider spatial improvements for monotony and positivity. When using these improvements, the explicit time advancing to be used is the RK3-SSP.

All of the above spatial schemes can be advanced in time with implicit schemes such as BDF1 and BDF2. Details of implementation for the complete viscous model are given in section 1.2.11. A spatially first-order accurate simplified Jacobian is systematically used. In the unsteady case, that option is used inside a two-step

Newton-like process, referred as unsteady defect correction (Martin and Guillard 1996). Linear stability is unconditional in all cases. In the case where we seek a steady solution or a slowly evoluting solution during a long time, the efficiency of an explicit scheme applying on unstructured mesh is severely limited by the Courant condition on the time step. The implicit time advancing is mandatory and needs to be combined with an efficient linear solver.

REMARK 1.2.– It can be efficient to apply a multigrid iteration in combination with pseudo-time advancing (steady case) or (for both cases) an efficient implicit time advancing. Designing a multigrid scheme for unstructured meshes rises the problem of defining a series of coarser grids. In other words, we have to define several new meshes or to find an alternative strategy. In Francescatto and Dervieux (1998) and Lallemand et al. (1992), this is done in a transparent manner from the fine mesh by using the so-called cell agglomeration. Parallel multigrid extensions are proposed in Mavriplis (1997) and Fournier et al. (1998). □

Another option which is well adapted to message passing parallelism is a Krylov–Newton–Schwarz (KNS) algorithm, as in Knoll and Keyes (2004). A first version of KNS, under the form of the restrictive additive Schwarz (RAS), was developed in Cai and Sarkis (1999) and Sarkis and Koobus (2000). A deflation-based coarse grid extension of RAS is studied in Alcin et al. (2013).

The *Code Aironum* of INRIA and Montpellier University, derived from the AERO software developed in a collaboration between University of Colorado at Boulder and INRIA, involves all the above features (in particular RAS) in a massively parallel implementation.

1.1.5.1. *Conclusion on the superconvergent scheme*

Based on MUSCL schemes, the V6 schemes involve sophisticated primitive variable reconstruction designed in order to enjoy low dissipation properties because of a model of sixth derivative. In the case of linear advection with uniform velocity, the scheme presents superconvergence properties on Cartesian subregions of mesh where convergence is close to the fifth order. On unstructured non-Cartesian subregions (where limiters do not apply, see the next section), convergence reduces to an order between the second order and fifth order. Interested readers are referred to some previous studies (Debiez 1996; Debiez and Dervieux 1999; Debiez et al. 1998; Abalakin et al. 2001, 2002a,b,c, 2004; Camarri et al. 2001, 2002, 2004; Gourvitch et al. 2004; Abalakin and Kozubskaya 2014).

1.1.6. *Positivity of mixed element-volume formulations*

1.1.6.1. *Introduction*

Some of the most useful properties of upwind approximation schemes for hyperbolic equations are their monotonicity and positivity properties. For example, for the Euler equations, and by combining flux splitting and limiters, Perthame and co-workers (Perthame and Khobalate 1992; Perthame and Shu 1996) proposed second-order accurate schemes that maintain a density and a temperature positive. We refer also to Linde and Roe (1998) and the workshop by Venkatakrisnan (1998). Non-oscillating schemes (Harten and Osher 1987; Cockburn and Shu 1989) propose high accuracy approximations applicable to many stiff problems. However, they do not enjoy a strict satisfaction of positivity or monotony, which remains an important issue for stiff simulations, particularly in relation with highly heterogeneous fluid flows (see, for example, Abgrall 1996 or Murrone and Guillard 2005).

The present section defines positive/monotone versions of the MEV scheme. We first examine the maximum principle for a scalar conservation law, then we introduce a flux splitting that preserves density positivity for the Euler equations in 1D. This provides a basis to construct *multidimensional* schemes, applying to unstructured meshes and ensuring density positivity and maximum principle for convected species. This is of paramount importance for most flows of industrial interest for two reasons. The first one is that many industrial flows are at medium Mach number and are rather stable when negative densities are avoided. The second reason is that in many Reynolds-averaged Navier–Stokes flows, limiters are not necessary for the mean flow itself, but robustness problems arise in the computation of turbulence closure variables such as k and ε.

1.1.6.2. *Positive schemes and LED schemes for nonlinear scalar conservation laws*

This section recalls some basic properties. We keep the usual TVD/LED (TVD: total variation diminishing, LED: local extremum diminishing) criterion preferably to weaker positivity criteria in order to deal also with the maximum principle. We consider a nonlinear *scalar* conservation law form for the unknown $U(\mathbf{x}, t)$:

$$\frac{\partial U}{\partial t}(\mathbf{x}, t) + \nabla_{\mathbf{x}} \cdot \mathcal{F}\Big(U(\mathbf{x}, t)\Big) = 0, \qquad [1.35]$$

where t denotes the time and $\mathbf{x}$ denotes the space coordinate. For the 1D (nonlinear) case and under some classical assumptions (Godlewski and Raviart 1996), the solution

U satisfies a *maximum principle* that, for simplicity, we write in the case of the whole space:

$$\min_{\mathbf{x}} U(\mathbf{x},0) \leq U(\mathbf{x},t) \leq \max_{\mathbf{x}} U(\mathbf{x},0). \qquad [1.36]$$

In 2D and 3D, but for the advection model, a positivity principle also holds. In the general multidimensional case, no TVD property applies. But the TVD numerical method can still be introduced in order to build non-oscillatory schemes. Note, however, that for compressible Euler and Navier–Stokes, the positivity of density, pressure, temperature, entropy, etc., holds in the continuous case and can be partly recovered in the discrete case by using TVD/LED numerics. From a positivity standpoint, it is enough to adopt the weaker LED criterion, which we do in next section.

Assuming that the mesh nodes are numbered, we call U_i the value at mesh node i. We recall now the classical positivity statement for an explicit time-integration (the proof is immediate).

LEMMA 1.1.– *A positivity criterion*. Suppose that an explicit first-order time-integration of equation [1.35] can be expressed under the form

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} = b_{ii} U_i^n + \sum_{j \neq i} b_{ij} U_j^n \ , \qquad [1.37]$$

where all the b_{ij}, $j \neq i$, are non-negative and $b_{ii} \in I\!R$. Then it can be shown that the above explicit scheme preserves positivity under the following condition on the time-step Δt: $b_{ii} + \dfrac{1}{\Delta t} \geq 0$.

The maximum principle can be evaluated with the incremental form introduced by Harten (1983) and used by Jameson (1993) as follows for defining multidimensional LED schemes (see also Godlewski and Raviart 1996):

LEMMA 1.2.– *A LED criterion (Jameson 1993)*. Suppose that an explicit first-order time-integration of equation [1.35] can be written in the form

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} = \sum_{k \in V(i)} c_{ik}(U^n)\,(U_k^n - U_i^n), \qquad [1.38]$$

with all $c_{ik}(U^n) \geq 0$, and where $V(i)$ denotes the set of the neighbors of node i. Then the previous scheme verifies that a local maximum cannot increase and a local

minimum cannot decrease, and under an appropriate condition on the time-step the positivity and the maximum principle are preserved[5].

1.1.6.2.1. First-order space-accurate MEV schemes

We integrate [1.35] over a cell C_i, integrating by parts the resulting convective fluxes and using a conservative approximation leads to the following semi-discretization of [1.35]:

$$a_i \frac{dU_i}{dt} + \sum_{j \in V(i)} \Phi(U_i, U_j, \nu_{ij}) = 0, \qquad [1.39]$$

where a_i is the measure of cell C_i. In the above semi-discretization, the values U_i and U_j correspond to a constant per cell *interpolation* of the variable U, and Φ is a *numerical flux function* so that $\Phi(U_i, U_j, \nu_{ij})$ approximates $\displaystyle\int_{\partial C_{ij}} \mathcal{F}(U) \cdot \mathbf{n}_i(\sigma) d\sigma$. In general, the numerical flux function $\Phi : (u, v, \nu) \mapsto \Phi(u, v, \nu)$ is assumed to be Lipschitz continuous, monotone increasing with respect to u, monotone decreasing with respect to v, and consistent:

$$\Phi(u, u, \nu) = \mathcal{F}(u) \cdot \nu. \qquad [1.40]$$

As already noted, taking $\Phi = \Phi^{HLLC}$ [1.21] makes this discretization a ρ, T, p-positive one under a usual time step condition.

1.1.6.2.2. Three-entry limited high-order space-accurate MEV scheme

We replace the values U_i and U_j by "better" interpolations U_{ij} and U_{ji} at the interface ∂C_{ij}. More precisely, the first-order MEV scheme becomes:

$$a_i \frac{dU_i}{dt} + \sum_{j \in V(i)} \Phi(U_{ij}, U_{ji}, \nu_{ij}) = 0, \qquad [1.41]$$

where U_{ij} and U_{ji} are left and right values of U at the interface ∂C_{ij}. Our purpose is to build a scheme that is non-oscillatory and positive. It has been early proved

5 PROOF.– Given U_i^n a local maximum, we deduce that $(U_k^n - U_i^n) \leq 0$ for all $k \in V(i)$. Therefore, [1.38] implies that $\dfrac{U_i^{n+1} - U_i^n}{\Delta t} \leq 0$, and the local maximum cannot increase. Likewise, we can prove that a local minimum cannot decrease. On the other hand, the reader can easily check that the positivity and the maximum principle are preserved when the time-step satisfies $\dfrac{1}{\Delta t} - \displaystyle\sum_{k \in V(i)} c_{ik}(U^n) \geq 0$. □

that high-order positive schemes must be necessarily built with a nonlinear process, the *limiter*. A limiter locally chooses between the first-order monotone version of the schemes and a higher-order one. In the case of unstructured meshes and scalar models, second-order positive schemes were derived using a two-entry symmetric limiter by Jameson (1993). Here, instead of the Jameson symmetric limiter, we work with the MUSCL formulation, involving two limiters per edge. The adaptation to triangulations (or tetrahedrizations) is close to the one proposed in Fezoui and Dervieux (1989) and Stoufflet et al. (1996). Following Debiez (1996), we extend it to a three-entry limiter that allows to design a positive scheme of third- (or even fifth-)order far from extrema when U varies smoothly. Then [1.41] becomes

$$a_i \frac{dU_i}{dt} + \sum_{j \in V(i)} \Phi(U_i + \frac{1}{2} L_{ij}(U), U_j - \frac{1}{2} L_{ji}(U), \nu_{ij}) \; = \; 0. \qquad [1.42]$$

In order to define $L_{ij}(U)$ and $L_{ji}(U)$, we use the upstream and downstream triangles (or tetrahedra) T_{ij} and T_{ji} (see Figures 1.6 and 1.7), as introduced in Fezoui and Dervieux (1989). Element T_{ij} is *upstream* to vertex i with respect to edge ij if for any sufficiently small real number η, the vector $-\eta \vec{ij}$ is inside element T_{ij}. Symmetrically, element T_{ji} is *downstream* to vertex i with respect to edge ij if for any small enough real number η the vector -$\eta \vec{ji}$ is inside element T_{ji}. If the 3D case is now considered, let i, n, m, o (respectively, j, r, s, t) be the four vertices of tetrahedron T_{ij} (respectively, T_{ji}), and let ϵ_{in}, ϵ_{im}, ϵ_{io} (respectively, ϵ_{jr}, ϵ_{js}, ϵ_{jt}) be the components of vector $\vec{ji}$ (respectively, $\vec{ij}$) in the oblique system of axes $(\vec{in}, \vec{im}, \vec{io})$ (respectively, $(\vec{jr}, \vec{js}, \vec{jt})$), we have

$$\vec{ji} \; = \; \epsilon_{in} \, \vec{in} \; + \; \epsilon_{im} \, \vec{im} \; + \; \epsilon_{io} \, \vec{io}, \quad ; \quad \vec{ij} \; = \; \epsilon_{jr} \, \vec{jr} \; + \; \epsilon_{js} \, \vec{js} \; + \; \epsilon_{jt} \, \vec{jt}.$$

Then T_{ij} and T_{ji} are upstream and downstrean elements, which mean that they have been chosen in such a way that the components ϵ_{in}, etc., are all non-negative, according to [1.22]. Let us introduce the following notations:

$$\Delta^- U_{ij} \; = \; \nabla U \, |_{T_{ij}} . \, \vec{ij}, \;\; \Delta^0 U_{ij} = U_j - U_i \;\; \text{and} \;\; \Delta^- U_{ji} \; = \; \nabla U \, |_{T_{ji}} \cdot \vec{ij},$$

where the gradients are those of the P1 (continuous and linear) interpolation of U. Jameson (1993) has noted that (for the 3D case):

$$\Delta^- U_{ij} \; = \; \epsilon_{in} \, (U_i \; - \; U_n) \; + \; \epsilon_{im} \, (U_i \; - \; U_m) \; + \; \epsilon_{io} \, (U_i \; - \; U_o),$$

and

$$\Delta^- U_{ji} \; = \; \epsilon_{jr} \, (U_r \; - \; U_j) \; + \; \epsilon_{js} \, (U_s \; - \; U_j) \; + \; \epsilon_{jt} \, (U_t \; - \; U_j),$$

with the same non-negative ϵ_{in}, ϵ_{im}, ϵ_{io}, ϵ_{jr}, ϵ_{js} and ϵ_{jt}.

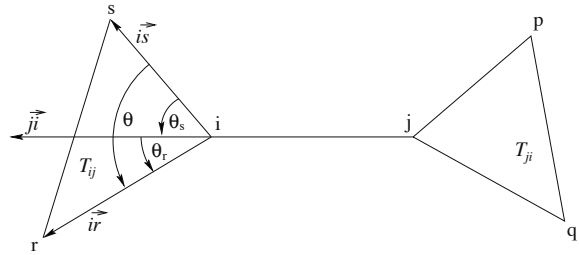

Figure 1.7. *Two-dimensional case: splitting of vector $\vec{ji}$ in the directions of downstream and upstream triangles edges*

Now, we introduce a family of continuous limiters with three entries, satisfying:

– $(P1)\ L(u,v,w) = L(v,u,w)$

– $(P2)\ L(\alpha\, u, \alpha\, v, \alpha\, w) = \alpha\, L(u,v,w)$

– $(P3)\ L(u,u,u) = u$

– $(P4)\ L(u,v,w) = 0$ if $uv \leq 0$

– $(P5)\ 0 \leq \frac{L(u,v,w)}{v} \leq 2$ if $v \neq 0$.

Note that there exists K^- and K^0 depending on (u,v,w) such that

$$L(u,v,w) = K^- u = K^0 v\ , \quad \text{with} \quad 0 \leq K^- \leq 2\ ,\ 0 \leq K^0 \leq 2. \qquad [1.43]$$

A function verifying (P1) to (P5) exists, and in the numerical examples, we shall use the following version of the Superbee method of Roe:

$$\begin{cases} L_{SB}(u,v,w) = 0 & \text{if} \quad uv \leq 0 \\ \qquad\qquad = Sign(u)\ \min(\,2\,|u|,\ 2\,|v|,\ |w|) & \text{otherwise.} \end{cases} \qquad [1.44]$$

We define

$$L_{ij}(U) = L(\Delta^- U_{ij},\ \Delta^0 U_{ij},\ \Delta^{HO} U_{ij}) \qquad [1.45]$$

$$L_{ji}(U) = L(\Delta^- U_{ji},\ \Delta^0 U_{ij},\ \Delta^{HO} U_{ji}), \qquad [1.46]$$

where $\Delta^{HO}U_{ji}$ is a third way of evaluating the variation of U which we can introduce for increasing the accuracy of the resulting scheme (see the following remark). Under the above assumptions, the LED property can be established and in particular, the positivity of density for Euler discretizations:

LEMMA 1.3.– The scheme defined by conditions P1-P4 and equations [1.45] and [1.46], based on a ρ-positive flux splitting is ρ-positive under a CFL-like condition.

PROOF.– See Cournède et al. (2006). □

REMARK 1.3.– Let assume that the option $L_{ij}(U) = \Delta^{HO}U_{ij}$ gives a high-order approximation with order ω. For U to be sufficiently smooth and assuming that the mesh size is smaller than α (small), there exists $\epsilon(\alpha)$ such that if $|\nabla U.\vec{ij}|/||\vec{ij}|| > \epsilon(\alpha)$ for an edge ij, the limiter L_{ij} is not active, that is,$L_{ij}(U) = \Delta^{HO}U_{ij}$. Then, the scheme is locally of order ω. □

REMARK 1.4.– In the case of a viscous flow, a particular attention has to be paid to the discrete diffusion operators that should also preserve positivity of variables; for finite-element discretization, this is related to standard acute angle condition (see Baba and Tabata 1981).

The proposed analysis does not need any assumption concerning the terms $\Delta^{HO}U_{ij}$. One option is the *superconvergent high order*, which allows computing a flow with, at the same time, lower dissipation and shock capturing. For $\Delta^{HO}U_{ij}$ we recommend using the extrapolated slopes defined in [1.27]. This option is useful for unsteady flows. It should be noted that dissipation can be even further decreased by adding sensors dedicated to the inhibition of limiters in regions where the flow is regular. Limiters should be avoided or locally well controlled in calculations, which need a very low level of dissipation, such as large eddy simulation and aeroacoustics. A second option is the usual *third-order,* which relies on the following flux:

$$\Delta^{HO3}U_{ij} = \frac{1}{3}\Delta^{-}U_{ij} + \frac{2}{3}\Delta^{0}U_{ij}, \qquad [1.47]$$

which gives us a third-order space-accurate scheme for linear advection on Cartesian triangular meshes. Since $\Delta^{HO3}U_{ij}$ is expressed from the two first entries, it produces in fact a "two-entry" limiter.

We propose now the analysis of these two-entry quasi-third order limited schemes with the important purpose of obtaining converged and accurate steady solutions. Let us define

$$R = \Delta^{0}U_{ij}/\Delta^{-}U_{ij}.$$

The extrapolation to mid-edge $U_{ij} = U_i + \frac{1}{2}L_{ij}(U)$ (see [1.42]) is written as

$$U_{ij} = U_i + \frac{1}{2}\psi(R)\Delta^- U_{ij} \quad \text{or} \quad U_{ij} = U_i + \frac{1}{2}\phi(\Delta^0 U_{ij}, \Delta^- U_{ij}).$$

The above limiter reduces to the one introduced by Koren (1993):

$$\psi_{Ko}(R) = min(1, 2R, \frac{1}{3} + \frac{2}{3}R) \quad \text{or} \quad \phi_{Ko}(a, b) = min(2a, 2b, \frac{1}{3}b + \frac{2}{3}a),$$

where ψ and ϕ are zero for $R \leq 0$. It is well known that this limiter shows a compressive behavior, with stiffer shocks. Also, it presents discontinuous derivatives. These properties have been identified as the cause of difficulties in convergence to steady state.

1.1.6.2.3. Piperno-limited MEV scheme

Piperno (1996) and Piperno and Depeyre (1998) have defined five design criteria for a *smooth limiter* based on the above third-order formulation:

i) the higher order gradient is taken for $R = 1$. Indeed, this is in the smooth part and we want to maximize the accuracy;

ii) function ψ has to be $\mathcal{C}^1$;

iii) since the upwind scheme is fully second-order and does not oscillate for $R \to \infty$, let us have $\psi(R) \to 1$ when $R \to \infty$;

iv) moreover, the curve should be flat when $R \to \infty$, that is, $\psi'(R) \to 0$ when $R \to \infty$;

v) at $R = 1$, we should have at least $\mathcal{C}^2$, and possibly the curve should coincide with the order three, that is, $\psi(R) = \frac{1}{3} + \frac{2}{3}R$ in a large interval around $R = 1$, in order to preserve third-order in this region.

Taking into account these criteria, Piperno defined the following limiter:

$$\psi_{Piperno}(R) = (\frac{1}{3} + \frac{2}{3}R) \begin{cases} \dfrac{3\frac{1}{R^2} - 6\frac{1}{R} + 19}{\frac{1}{R^3} - 3\frac{1}{R} + 18} & \text{if } R < \frac{1}{2} \\ 1 + (\dfrac{3}{2}\dfrac{1}{R} + 1)(\dfrac{1}{R} - 1)^3 & \text{if } R \geq 1. \end{cases}$$

This limiter is smoother than the previous one and is preferred for computing steady flows.

1.1.6.2.4. Gamma limiter

We follow an idea of Koren (1993) and enlarge the order three interval (criterion (v)) to have it between $R = 1/k$ and $R = k$, for $k \geq 2$. The novel limiter is written as

$$\psi_\gamma(R) = (\frac{1}{3} + \frac{2}{3}R) \begin{cases} \dfrac{3\frac{1}{R^2} - 6\frac{1}{R} + 19}{(\frac{1}{R} - k)^3 + 3\frac{1}{R^2} - 6\frac{1}{R} + 19} & \text{if } R < \frac{1}{k} \\ 1 & \text{if } \frac{1}{k} \leq R \leq k \\ 1 + (\dfrac{3}{2}\dfrac{1}{R-k+1} + 1)(\dfrac{1}{R-k+1} - 1)^3 & \text{if } R \geq k. \end{cases}$$

Our choices. In steady computations, the Gamma limiter will be chosen. For unsteady computations, and in particular in combination with the low-dissipation superconvergent scheme, the three-entry limiter will be applied.

1.2. Viscous compressible flows

1.2.1. *Model for laminar flows*

1.2.1.1. *Continuous state system*

The compressible Navier–Stokes equations for mass, momentum and energy conservation read

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \\ \dfrac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p = \nabla \cdot \mathcal{T}, \\ \dfrac{\partial (\rho E)}{\partial t} + \nabla \cdot ((\rho E + p)\mathbf{u}) = \nabla \cdot (\mathcal{T} \cdot \mathbf{u}) + \nabla \cdot (\lambda \nabla T), \end{cases} \quad [1.48]$$

where ρ denotes the density ($\mathrm{kg/m^3}$), $\mathbf{u}$ is the velocity ($\mathrm{m/s}$), E is the total energy per mass ($\mathrm{m^2.s^{-2}}$), p is the pressure ($\mathrm{N/m^2}$), given by [1.3], T is the temperature (K) such that $\rho C_v T = E - \frac{1}{2}\rho(u^2 + v^2 + w^2)$ (C_v being the specific heat at constant volume), μ is the laminar dynamic viscosity ($\mathrm{kg/(m.s)}$) and λ is the laminar conductivity. $\mathcal{T}$ is the laminar stress tensor:

$$\mathcal{T} = \mu \left[(\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathbf{T}}) - \frac{2}{3} \nabla . \mathbf{u}\, I \right],$$

where (in 3D) $\mathbf{u} = (u, v, w)$ and

$$\nabla.\,\mathbf{u}\,I = \begin{pmatrix} u_x + v_y + w_z & 0 & 0 \\ 0 & u_x + v_y + w_z & 0 \\ 0 & 0 & u_x + v_y + w_z \end{pmatrix},$$

where $u_x = \frac{\partial u}{\partial x}$, $u_y = \frac{\partial u}{\partial y}$, $u_z = \frac{\partial u}{\partial z}$ (idem for v and w). The variation of non-dimensionalized laminar dynamic viscosity and conductivity coefficients μ and λ as functions of the dimensional temperature T are defined by Sutherland's law:

$$\mu = \mu_\infty \left(\frac{T}{T_\infty}\right)^{\frac{3}{2}} \left(\frac{T_\infty + \mathrm{Su}}{T + \mathrm{Su}}\right) \quad \text{and} \quad \lambda = \lambda_\infty \left(\frac{T}{T_\infty}\right)^{\frac{3}{2}} \left(\frac{T_\infty + \mathrm{Su}}{T + \mathrm{Su}}\right),$$

where $\mathrm{Su} = 110^\circ K$ is the Sutherland temperature and the index ∞ denotes reference quantities. The relation linking μ and λ is expressed from the Prandtl laminar number:

$$\mathrm{Pr} = \frac{\mu C_p}{\lambda} \quad \text{with} \quad \mathrm{Pr} = 0.72 \quad \text{for (dry) air,}$$

where C_p is the specific heat at constant pressure. In some case, we shall be interested by the steady compressible Navier–Stokes system for a perfect gas, which can be written under a compact form as

$$\nabla \cdot \mathcal{F}^E(W) - \nabla \cdot \mathcal{F}^V(W) = 0 \ \text{ in}\, \Omega \subset \mathbb{R}^3, \qquad [1.49]$$

where $W = {}^t(\rho,\, \rho\mathbf{u},\, \rho E)$ is the conservative flow variables vector and vector $\mathcal{F}^E$ represents the Euler fluxes:

$$\mathcal{F}^E(W) = {}^t\left(\rho\mathbf{u},\, \rho u_1\mathbf{u} + p\mathbf{e}_1,\, \rho u_2\mathbf{u} + p\mathbf{e}_2,\, \rho u_3\mathbf{u} + p\mathbf{e}_3,\, \rho\mathbf{u}H\right),$$

where $H = E + p/\rho$ is the total enthalpy. We write in short the viscous fluxes and viscous stress tensor σ as follows:

$$\mathcal{F}^V(W) = [0, \sigma, -(\mathbf{q} - \mathbf{u} \cdot \sigma)]^T \quad ; \quad \sigma = \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}) - \frac{2}{3}\mu\nabla \cdot \mathbf{u}\mathbf{I},$$

with μ representing the constant viscosity. The heat flux $\mathbf{q}$ is given by Fourier's law $\mathbf{q} = -\lambda\nabla T$, where λ is the heat conduction. With the standpoint of mesh adaptation,

we consider for theoretical developments regular C^3 solution fields. The five unknowns are in functional product space:

$$\mathcal{V} \;=\; \left[H^1(\Omega)\cap C^3(\bar{\Omega})\right] \times \left[H^1_0(\Omega)\cap C^3(\bar{\Omega})\right]^3 \times \left[H^1(\Omega)\cap C^3(\bar{\Omega})\right], \quad [1.50]$$

assuming adiabatic conditions on walls. We formulate the Navier–Stokes model in a compact variational formulation:

$$\text{Find } W \in \mathcal{V} \text{ such that } \forall \boldsymbol{\psi} \in \mathcal{V},\ (\Psi(W),\, \boldsymbol{\psi}) = 0 \text{ with } \Psi = \Psi^E + \Psi^\Gamma + \Psi^V, \quad [1.51]$$

where $\boldsymbol{\psi} = (\psi_\rho, \psi_{\rho u}, \psi_{\rho v}, \psi_{\rho w}, \psi_{\rho E})^T$. The Euler term Ψ^E relies on the usual Euler fluxes $\mathcal{F}^E$ denoted $\mathcal{F}$ in [1.1]:

$$\left(\Psi^E(W),\, \boldsymbol{\psi}\right) \;=\; \int_\Omega \boldsymbol{\psi} \cdot\, \nabla \cdot \mathcal{F}^E(W)\, \mathrm{d}\Omega. \quad [1.52]$$

Term Ψ^Γ holds for boundary fluxes contribution, which we denote shortly (see equation [1.4]):

$$\left(\Psi^\Gamma(W),\, \boldsymbol{\psi}\right) \;=\; \int_\Gamma \boldsymbol{\psi} \cdot\, \hat{\mathcal{F}}(W) \cdot \mathbf{n}\, \mathrm{d}\Gamma.$$

Viscous fluxes $\mathcal{F}^V$ provide seven terms:

$$(\Psi^V(W), \boldsymbol{\psi}) = \int_\Omega \boldsymbol{\psi} \cdot\, \nabla \cdot \mathcal{F}^V(W)\, \mathrm{d}\Omega = \sum_{k=1}^{7} T_k^V. \quad [1.53]$$

The first three terms come from moment equations and depend only on $\boldsymbol{\psi}_{\rho\mathbf{u}} = (\psi_{\rho u}, \psi_{\rho v}, \psi_{\rho w})^T$:

$$\begin{aligned}
T_1^V &= \int_\Omega \boldsymbol{\psi}_{\rho\mathbf{u}} \cdot \nabla \cdot (\mu \nabla \mathbf{u})\ \mathrm{d}\Omega \\
T_2^V &= \int_\Omega \boldsymbol{\psi}_{\rho\mathbf{u}} \cdot \nabla \cdot \left(\mu (\nabla \mathbf{u})^T\right) \mathrm{d}\Omega \\
T_3^V &= -\frac{2}{3} \int_\Omega \boldsymbol{\psi}_{\rho\mathbf{u}} \cdot \nabla \cdot (\mu\, (\nabla \cdot \mathbf{u})\, \mathbf{I})\ \mathrm{d}\Omega.
\end{aligned}$$

The last four terms are derived from the energy equation:

$$
\begin{aligned}
T_4^V &= \int_\Omega \psi_{\rho E} \nabla \cdot (\lambda \nabla T)\, \mathrm{d}\Omega \\
T_5^V &= \int_\Omega \psi_{\rho E} \nabla \cdot \left(\mu\, \mathbf{u} \cdot (\nabla \mathbf{u})^T\right) \mathrm{d}\Omega \\
T_6^V &= \int_\Omega \psi_{\rho E} \nabla \cdot (\mu\, \mathbf{u} \cdot \nabla \mathbf{u})\, \mathrm{d}\Omega \\
T_7^V &= -\frac{2}{3} \int_\Omega \psi_{\rho E} \nabla \cdot (\mu\, \mathbf{u} \cdot ((\nabla \cdot \mathbf{u})\, \mathbf{I}))\, \mathrm{d}\Omega.
\end{aligned}
$$

1.2.1.2. *Variational discrete formulation*

For the spatial semi-discrete model, we consider the extension of the mixed finite element-finite volume (previous section) to viscous case. We can formulate it under the form of a finite element variational formulation. We assume that Ω is covered by a finite element mesh $\mathcal{H}$ composed of simplicial elements, denoted as K. Let us introduce the following approximation space:

$$
\mathcal{V}_h = \left\{ \boldsymbol{\psi}_h \in \mathcal{V} \;\middle|\; \boldsymbol{\psi}_{h|K} \text{ is affine } \forall K \in \mathcal{H} \right\}, \qquad [1.54]
$$

where $\mathcal{V}$ is defined by [1.50]. The weak discrete formulation is written as follows:

Find $W_h \in \mathcal{V}_h$ such that $\forall \boldsymbol{\psi}_h \in \mathcal{V}_h$, $(\Psi_h(W_h),\, \boldsymbol{\psi}_h) = 0$,

with

$$
\begin{aligned}
(\Psi_h(W_h),\, \boldsymbol{\psi}_h) = \int_\Omega \boldsymbol{\psi}_h \cdot \nabla \cdot \mathcal{F}_h^E(W_h)\, \mathrm{d}\Omega + \int_\Gamma \boldsymbol{\psi}_h \cdot \hat{\mathcal{F}}_h(W_h).\mathbf{n}\, \mathrm{d}\Gamma \quad [1.55] \\
+ \int_\Omega \boldsymbol{\psi}_h \cdot \nabla \cdot \mathcal{F}_h^V(W_h)\, \mathrm{d}\Omega + \int_\Omega \boldsymbol{\psi}_h \cdot D_h(W_h)\, \mathrm{d}\Omega. \quad [1.56]
\end{aligned}
$$

The fourth RHS term in the previous formulation is an added numerical diffusion dedicated to numerical stability. Indeed, the D_h term holds for the difference between the Galerkin central-differences approximation and the second-order

Godunov approximation. For smooth fields, the D_h term is a third-order term with respect to the mesh size parameter h.

Taking in system [1.55] the P^1 interpolation of the fluxes $\mathcal{F}$ as discretization principle produces a finite-element scheme, which is identical to the central-differenced finite-volume scheme built on the median dual cells, thus

$$\forall\, W \in \mathcal{V}_h \cup \mathcal{V}\,,\quad \mathcal{F}_h^E(W) = \Pi_h \mathcal{F}^E(W) \text{ and } \hat{\mathcal{F}}_h(W) = \Pi_h \hat{\mathcal{F}}(W). \quad [1.57]$$

However, for the viscous term, we cannot move the interpolation outside the flux because viscous fluxes are expressed in terms of primitive variables. Let f be the transformation function from primitive variables $\tilde{U} = (\rho, \mathbf{u}, T)$ into conservatives ones $W = (\rho, \rho\mathbf{u}, \rho E)$, we set

$$\mathcal{F}_h^V(W) = \mathcal{F}_h^V(f(\tilde{U})) = \mathcal{F}^V(f(\Pi_h \tilde{U})) = \mathcal{F}^V(f(\Pi_h f^{-1}(W))).$$

In other words, our discretization consists of P^1 interpolating the primitive variables in the above viscous terms $\{T_i^V\}_{i=1..7}$ for our discretization principle. This completes the definition of the discrete system under study.

1.2.2. *Boundary conditions spatial discretization*

Boundary conditions are imposed element-wise, for example, a boundary flux is computed for each boundary edge/face and added to each of its vertices.

1.2.3. *No-slip boundary condition*

For no-slip boundary conditions, $\mathbf{u} = 0$ is strongly enforced at each iteration. Consistently, we impose $\Phi_\rho = 0$ at the boundary. The energy flux is fixed according to the desired temperature behavior: for an adiabatic wall, it is null, and for an isothermal wall, the energy variable is enforced similarly to the velocity.

1.2.4. *Slip boundary condition*

For this boundary condition, we impose weakly

$$\mathbf{u} \cdot \mathbf{n} = 0. \quad [1.58]$$

To this end, we compute the flux Φ between the state on the boundary W and a mirror state $\overline{W}$:

$$\Phi_{Slip} = \Phi^{HLLC}(W, \overline{W}, \mathbf{n}),$$

where

$$W = \begin{pmatrix} \rho \\ \rho\mathbf{u} \\ \rho E \end{pmatrix} \quad \text{and} \quad \overline{W} = \begin{pmatrix} \rho \\ \rho\mathbf{u} - 2\,\rho\,(\mathbf{u}\cdot\mathbf{n})\,\mathbf{n} \\ \rho E \end{pmatrix}.$$

If condition [1.58] is satisfied, then $W = \overline{W}$ and thus $\Phi^{HLLC}(W, \overline{W}, \mathbf{n}) = F(W)\cdot\mathbf{n}$. Moreover, if W verifies relation [1.58], then $F(W)\cdot\mathbf{n}$ simplifies to

$$\Phi_{Slip} = F(W)\cdot\mathbf{n} = (0, p\,\mathbf{n}, 0)^t.$$

Therefore, if the desired condition is satisfied, then the boundary flux reduces to its well-known commonly used form. Nevertheless, the state W on the boundary does not satisfy this condition unless it is imposed strongly, which is not possible as we will no more conserve the mass. This problem is solved by computing the flux between the state and its mirror state. This flux depends on the considered approximate Riemann solver.

1.2.5. *Influence stencil*

As boundary conditions fluxes are computed element-wise, they involve at most a dependency between a vertex and its neighbors. The total contribution of boundary terms to the flux of a given vertex can be summed up as

$$\mathbf{\Gamma}_i = \sum_{F^{bdy}\ni P_i} \Phi^{bdy}_{i,F^{bdy}}(W_i, W_j, W_k), \qquad [1.59]$$

where F^{bdy} are the boundary elements containing the vertex P_i.

1.2.6. *Spalart–Allmaras one equation turbulence model*

Various forms of the Spalart–Allmaras (SA) turbulence model exist. The original Spalart–Allmaras one equation turbulence model is written as follows (Spalart and Allmaras 1992):

$$\frac{\partial \tilde{\nu}}{\partial t} + \mathbf{u} \cdot \nabla \tilde{\nu} = c_{b1}[1 - f_{t2}]\tilde{S}\tilde{\nu} - \left[c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2}\right] \left(\frac{\tilde{\nu}}{d}\right)^2 + \frac{1}{\sigma}\left[\nabla \cdot ((\nu + \tilde{\nu})\nabla\tilde{\nu}) + c_{b2}\|\nabla\tilde{\nu}\|^2\right] + f_{t1}\Delta\mathbf{u}^2,$$

where $\tilde{\nu}$ is the kinematic eddy turbulent viscosity. Hereafter, we detail some of these variants.

1.2.7. *SA one-equation model without trip and without f_{t2} term*

In the standard SA model, the trip term is being left out, that is, $f_{t1} = 0$. Moreover, some implementations ignore also the f_{t2} term as it is argued that if the trip is not included, then f_{t2} is not necessary (Eça et al. 2007). In this monograph, this simplified version is considered. It reads in pseudo-vector notations:

$$\frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} = c_{b1}\tilde{S}\tilde{\nu} - c_{w1} f_w \left(\frac{\tilde{\nu}}{d}\right)^2 + \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left((\nu + \tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j}\right) + c_{b2}\frac{\partial \tilde{\nu}}{\partial x_i}\frac{\partial \tilde{\nu}}{\partial x_i}\right].$$

However, we prefer to write it under the following form, which is more appropriate for its discretization with the mixed element-volume method:

$$\frac{\partial \rho\tilde{\nu}}{\partial t} + \frac{\partial u_j \rho\tilde{\nu}}{\partial x_j} = \rho c_{b1}\tilde{S}\tilde{\nu} - \rho c_{w1} f_w \left(\frac{\tilde{\nu}}{d}\right)^2 + \frac{\rho}{\sigma}\left[\frac{\partial}{\partial x_j}\left((\nu + \tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j}\right) + c_{b2}\frac{\partial \tilde{\nu}}{\partial x_i}\frac{\partial \tilde{\nu}}{\partial x_i}\right].$$

REMARK 1.5.– This is not a conservative model. If conservative form of the Spalart–Allmaras is foreseen, we have to consider the variation of Catris and Aupoix (2000). □

The turbulent eddy viscosity is computed from

$$\mu_t = \rho\tilde{\nu} f_{v1},$$

where

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \quad \text{and} \quad \chi = \frac{\tilde{\nu}}{\nu} \quad \text{with} \quad \nu = \frac{\mu}{\rho}.$$

Additional definitions are given by the following equations:

$$\tilde{S} = \Omega + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2} \quad \text{where} \quad \Omega = \|\nabla \times \mathbf{u}\|.$$

The magnitude of the vorticity is computed from the vorticity tensor where each component is given by $\omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right)$ and $\Omega = \sqrt{2 \sum_{i,j=1..3} \omega_{ij}\omega_{ij}}$. Symbol d holds for the distance from the field point to the nearest wall and

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}.$$

Note that we have the following relations:

$$\chi f_{v1} = \frac{\tilde{\nu} f_{v1}}{\nu} = \frac{\nu_t}{\nu} \quad \Longrightarrow \quad 1 - \frac{\chi}{1 + \chi f_{v1}} = 1 - \frac{\tilde{\nu}}{\nu + \nu_t}.$$

The constants are

$$\sigma = \frac{2}{3} \qquad c_{b1} = 0.1355 \;\; c_{b2} = 0.622 \;\; \kappa = 0.41$$

$$c_{w1} = \frac{c_{b1}}{\kappa} + \frac{1 + c_{b2}}{\sigma} \qquad c_{w2} = 0.3 \qquad c_{w3} = 2 \qquad c_{v1} = 7.1.$$

Finally, the function f_w is computed as follows:

$$f_w = g\left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right)^{1/6} \text{ with } g = r + c_{w2}\left(r^6 - r\right) \text{ and } r = \min\left(\frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}, 10\right).$$

According to Spalart and Rumsey (2007), for the boundary conditions the following values are chosen:

$$\tilde{\nu}_{wall} = 0 \qquad \tilde{\nu}_{farfield} \in [3\nu_\infty, 5\nu_\infty].$$

1.2.8. *"Standard" SA one-equation model (without trip)*

It reads in pseudo-vector notations:

$$\frac{\partial \tilde{\nu}}{\partial t} + u_j \frac{\partial \tilde{\nu}}{\partial x_j} = c_{b1}[1 - f_{t2}]\tilde{S}\tilde{\nu} - \left[c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2}\right]\left(\frac{\tilde{\nu}}{d}\right)^2 + \frac{1}{\sigma}\left[\frac{\partial}{\partial x_j}\left((\nu + \tilde{\nu})\frac{\partial \tilde{\nu}}{\partial x_j}\right) + c_{b2}\frac{\partial \tilde{\nu}}{\partial x_i}\frac{\partial \tilde{\nu}}{\partial x_i}\right],$$

where $f_{t2} = c_{t3} \exp\left(-c_{t4}\chi^4\right)$ with $c_{t3} = 1.2$ and $c_{t4} = 0.5$.

1.2.9. *"Full" SA one-equation model (with trip)*

For the full SA model, we have to compute the f_{t1} term:

$$f_{t1} = c_{t1} g_t \exp\left(-c_{t2}\frac{\omega_t^2}{\delta \mathbf{u}_t^2}(d^2 + g_t^2 d_t^2)\right) \quad \text{with} \quad g_t = \min\left(0.1, \frac{\delta \mathbf{u}_t}{\omega_t \delta x_t}\right),$$

where $\delta \mathbf{u}_t$ is the difference between the velocity at the field point and that at the trip (on the wall), δx_t is the grid spacing along the wall at the trip, ω_t is the wall vorticity at the trip and d_t is the distance from the field point to the trip and

$$c_{t1} = 1 \quad \text{and} \quad c_{t2} = 2.$$

The farfield boundary condition is given as

$$0 \leq \tilde{\nu}_{farfield} \leq 1/10\nu_\infty.$$

1.2.10. *Mixed element-volume discretization of SA*

To discretize the turbulent model, the mixed element-volume method is considered. The Euler explicit discretization of the above equation in time and in space leads to

$$\frac{V_i}{\Delta t}\delta\rho\tilde{\nu}_i = -\mathbf{F}_i^n + \mathbf{S}_i^n + \mathbf{Q}_i^n,$$

where $\mathbf{F}$, $\mathbf{S}$ and $\mathbf{Q}$ are, respectively, the Euler, viscous and source flux terms, $\delta\rho\tilde{\nu}_i = \rho_i^{n+1}\tilde{\nu}_i^{n+1} - \rho_i^n\tilde{\nu}_i^n$ and $V_i = |C_i|$.

1.2.10.1. *Convective terms*

First, we describe how the convective part is discretized. The flux on the turbulent variable is given by

$$\mathbf{F}_i^n = \sum_{edge_{ij}} \mathbf{\Phi}_{ij}^{\rho\tilde{\nu}}(\mathbf{W}_i, \mathbf{W}_j, \nu_{ij}),$$

where $\mathbf{W}$ is the conservative vector variable $(\rho, \rho\mathbf{u}, \rho E, \rho\tilde{\nu})$, and ν_{ij} is the edge cell interface non-normalized normals from $\mathbf{p}_i$ to $\mathbf{p}_j$. Here, we have two choices.

1.2.10.2. *Linear convection*

The simplest one is to consider a linear advection of $\rho\tilde{\nu}$:

$$\mathbf{\Phi}_{ij}^{\rho\tilde{\nu}}(\mathbf{W}_i, \mathbf{W}_j, \nu_{ij}) = \begin{cases} \eta\ \rho\tilde{\nu}_i & \text{if} \quad \eta > 0 \\ \eta\ \rho\tilde{\nu}_j & \text{otherwise,} \end{cases} \tag{1.60}$$

where $\eta = \dfrac{1}{2}\left(\mathbf{u}_i \cdot \nu_{ij} + \mathbf{u}_j \cdot \nu_{ij}\right)$.

1.2.10.3. *Nonlinear convection*

The second choice is nonlinear and has been proposed in Larrouturou (1989). It has been proven that this scheme preserves the maximum principle for the convected turbulent variable. It reads

$$\mathbf{\Phi}_{ij}^{\rho\tilde{\nu}}(\mathbf{W}_i, \mathbf{W}_j, \nu_{ij}) = \mathbf{\Phi}_{ij}^{\rho}(\mathbf{W}_i, \mathbf{W}_j, \nu_{ij}) \cdot \begin{cases} \tilde{\nu}_i & \text{if} \quad \mathbf{\Phi}_{ij}^{\rho}(\mathbf{W}_i, \mathbf{W}_j, \nu_{ij}) > 0 \\ \\ \tilde{\nu}_j & \text{otherwise,} \end{cases} \tag{1.61}$$

where the density flux $\mathbf{\Phi}_{ij}^{\rho}$ is computed with the appropriate Riemann solver.

1.2.10.4. *Viscous terms*

The viscous terms can been discretized in two different ways.

1.2.10.4.1. First-order method

The classical approach consists of considering the diffusion term as a source term (see source terms paragraph for its discretization), that is, it is discretized under the model form. The dissipation term is discretized with the FEM. Then, we have

$$\nabla \cdot S(\mathbf{W}) = \frac{\rho}{\sigma}\left[\nabla \cdot \left((\nu + \tilde{\nu})\nabla\tilde{\nu}\right)\right].$$

The viscous fluxes are

$$\mathbf{S}_i^n = \sum_{K \ni i} \int_K S_i(\mathbf{W})|_K \cdot \nabla \Phi_i \, d\mathbf{x} = - \sum_{K \ni i} \int_{\partial C_i \cap K} S_i(\mathbf{W})|_K \cdot \mathbf{n} \, d\sigma$$
$$= - \sum_{\mathbf{p}_j \in B(\mathbf{p}_i)} \int_{\partial C_{ij}} S_i(\mathbf{W}) \cdot \mathbf{n}_{ij} \, d\sigma, \qquad [1.62]$$

where Φ_i is the P_1 finite element basis function associated with vertex i $B(\mathbf{p}_i)$ is the set of neighbors of i. The effective computation of the integral then reduces to the computations of integrals of the type

$$\int_K \nabla \Phi_i \, \nabla \Phi_j \, d\mathbf{x} \; = \; |K| \, \nabla \Phi_i|_K \, \nabla \Phi_j|_K .$$

We set

$$\tilde{\nu}|_K = \frac{1}{4} \sum_{\mathbf{p}_j \in K} \tilde{\nu}_j \qquad \text{and} \qquad \nabla \tilde{\nu}|_K = \sum_{\mathbf{p}_j \in K} \tilde{\nu}_j \nabla \Phi_j ,$$

then the dissipation term $\nabla \cdot \left(\frac{\rho_i}{\sigma} (\nu + \tilde{\nu}) \nabla \tilde{\nu} \right)$ is written as

$$\left(S_i^{\text{diss}} \right) |_K = |K| \, \frac{1}{\sigma} \rho_i \left((\nu|_K + \tilde{\nu}|_K) \, \nabla \tilde{\nu}|_K \cdot \nabla \Phi_i|_K \right) .$$

But, in that case, *the discretization is only first-order spatially accurate*. The second order of the method is lost.

1.2.10.5. *Second-order method*

To recover the second order of the method, diffusive and dissipative terms are re-written under the following form:

$$\nabla \cdot S(\mathbf{W}) = \frac{\rho}{\sigma} \left[\nabla \cdot ((\nu + \tilde{\nu}) \nabla \tilde{\nu}) + c_{b2} \nabla \tilde{\nu} \cdot \nabla \tilde{\nu} \right]$$
$$= \frac{\rho}{\sigma} \left[\nabla \cdot ((\nu + (1 + c_{b2}) \tilde{\nu}) \nabla \tilde{\nu}) - c_{b2} \, \tilde{\nu} \, \nabla \cdot \nabla \tilde{\nu} \right]$$

using relation $\nabla \cdot (\tilde{\nu}\nabla\tilde{\nu}) = \nabla\tilde{\nu} \cdot \nabla\tilde{\nu} + \tilde{\nu}\ \nabla \cdot \nabla\tilde{\nu}$. Then, the diffusion and dissipation terms are discretized with the FEM using the viscous flux given by relation [1.62]. The diffusion term $\dfrac{c_{b2}}{\sigma}\ \nabla \cdot (\rho_i\tilde{\nu}_i\ \nabla\tilde{\nu})$ reads

$$\left(S_i^{\text{diff}}\right)|_K = |K|\ \frac{c_{b2}}{\sigma}\rho_i\tilde{\nu}_i\ (\nabla\tilde{\nu}|_K \cdot \nabla\Phi_i|_K)$$

and the dissipation term $\nabla \cdot \left(\dfrac{\rho_i}{\sigma}(\nu + (1+c_{b2})\tilde{\nu})\nabla\tilde{\nu}\right)$ is given as

$$\left(S_i^{\text{diss}}\right)|_K = |K|\ \frac{\rho_i}{\sigma}\Big((\nu|_K + (1+c_{b_2})\tilde{\nu}|_K)\ \nabla\tilde{\nu}|_K \cdot \nabla\Phi_i|_K\Big).$$

1.2.10.6. *Source terms*

Finally, the source terms (diffusion, production and destruction for the order one method and, production and destruction for the order two method) are discretized by simple integration on each vertex cell:

$$\text{First order:}\quad \mathbf{Q}_i^n = |C_i|\left(\rho_i c_{b1}\tilde{S}_i\tilde{\nu}_i - \rho_i c_{w1} f_w\left(\frac{\tilde{\nu}_i}{d_i}\right)^2 + \frac{c_{b2}}{\sigma}\rho_i\|\nabla\tilde{\nu}_i\|\right),$$

$$\text{Second order:}\quad \mathbf{Q}_i^n = |C_i|\left(\rho_i c_{b1}\tilde{S}_i\tilde{\nu}_i - \rho_i c_{w1} f_w\left(\frac{\tilde{\nu}_i}{d_i}\right)^2\right),$$

where $|C_i|$ is the volume of the vertex cell and all variables are point-wise:

$$\tilde{S}_i = \Omega_i + \frac{\tilde{\nu}_i}{\kappa^2 d_i^2} f_{v2}.$$

Influence stencil. As each element produces a viscous flux for each of its vertices, we can deduce the total viscous fluxes for a given cell/vertex i as

$$\mathbf{S}_i = \sum_{K \ni P_i} \Phi_{i,K}^{visc}(W_i, W_j, W_k, W_l), \qquad [1.63]$$

so that the viscous fluxes of a vertex only depend on the ball $\mathcal{V}^1(P_i)$ of first order neighbors of P_i.

1.2.11. *Implicit time integration*

Once the equations have been discretized in space, a set of ordinary differential equations in time is obtained. For an implicit time integration, the semi-discretized RANS system becomes

$$\frac{|C_i|}{\delta t_i^n}\delta W_i \;=\; -\mathbf{F}_i^{n+1} + \mathbf{S}_i^{n+1} + \mathbf{Q}_i^{n+1} + \mathbf{\Gamma}_i^{n+1}, \qquad [1.64]$$

where $\delta W_i = W_i^{n+1} - W_i^n$, $|C_i|$ is the area/volume of the finite volume cell associated with vertex i and δt_i^n is the considered time step. After linearization of the RHS, this becomes:

$$\begin{aligned}
&\left(\frac{|C_i|}{\delta t_i^n} I_d + \frac{\partial \mathbf{F}_i^n}{\partial W_i} - \frac{\partial \mathbf{S}_i^n}{\partial W_i} - \frac{\partial \mathbf{Q}_i^n}{\partial W_i} - \frac{\partial \mathbf{\Gamma}_i^n}{\partial W_i}\right)\delta W_i \\
&+ \sum_{j\in\mathcal{V}^1(i)} \left(\frac{\partial \mathbf{F}_i^n}{\partial W_j} - \frac{\partial \mathbf{S}_i^n}{\partial W_j} - \frac{\partial \mathbf{Q}_i^n}{\partial W_j} - \frac{\partial \mathbf{\Gamma}_i^n}{\partial W_j}\right)\delta W_j \\
&= -\mathbf{F}_i^n + \mathbf{S}_i^n + \mathbf{Q}_i^n + \mathbf{\Gamma}_i^n,
\end{aligned}$$

where $P_j \in \mathcal{V}^1(i)$ is the set of vertices connected to vertex i by an edge. But, the computation of the Jacobian of the second-order inviscid flux $\dfrac{\partial \mathbf{F}_i^n}{\partial W_j}$ is complex and costly (as it involved the second order ball of vertex i, as seen above, while the other terms only involve the first order ball). It is thus approximated by the Jacobian of the first-order inviscid flux $\dfrac{\partial \widetilde{\mathbf{F}}_i^n}{\partial W_j}$, which only depends on the first-order ball (the other Jacobian terms are kept unchanged):

$$\begin{aligned}
&\left(\frac{|C_i|}{\delta t_i^n} I_d + \frac{\partial \widetilde{\mathbf{F}}_i^n}{\partial W_i} - \frac{\partial \mathbf{S}_i^n}{\partial W_i} - \frac{\partial \mathbf{Q}_i^n}{\partial W_i} - \frac{\partial \mathbf{\Gamma}_i^n}{\partial W_i}\right)\delta W_i \\
&+ \sum_{j\in\mathcal{V}^1(i)} \left(\frac{\partial \widetilde{\mathbf{F}}_i^n}{\partial W_j} - \frac{\partial \mathbf{S}_i^n}{\partial W_j} - \frac{\partial \mathbf{Q}_i^n}{\partial W_j} - \frac{\partial \mathbf{\Gamma}_i^n}{\partial W_j}\right)\delta W_j \\
&= \; -\mathbf{F}_i^n + \mathbf{S}_i^n + \mathbf{Q}_i^n + \mathbf{\Gamma}_i^n.
\end{aligned}$$

As the RHS still consider the second-order inviscid flux, this acts as an approximated Jacobian but is independent of the residual and thus does not affect the spatial order of the scheme. The first term of the LHS contributes to the diagonal of

the matrix and the second term of the LHS (i.e. the sum) contributes to extra-diagonal terms on line i of the matrix. This linearized system can be written in vector form:

$$\mathbf{A}^n\,\delta W^n = \mathbf{R}^n \quad \text{with} \quad \mathbf{A}^n = \frac{|C|}{\delta t^n}\mathbf{I} - \frac{\partial \widetilde{\mathbf{R}}^n}{\partial \mathbf{W}} \quad \text{and} \quad \delta \mathbf{W}^n = \mathbf{W}^{n+1} - \mathbf{W}^n,$$

where $\mathbf{R}^n = -\mathbf{F}^n + \mathbf{S}^n + \mathbf{Q}^n + \mathbf{\Gamma}^n$ and $\widetilde{\mathbf{R}}^n = -\widetilde{\mathbf{F}}^n + \mathbf{S}^n + \mathbf{Q}^n + \mathbf{\Gamma}^n$.

This linear system is solved at each flow solver iteration. To solve it, we follow the SGS approach based on lower-upper symmetric Gauss-Seidel (LU-SGS) implicit solver initially introduced by Jameson and Yoon (1987) and fully developed by Luo et al. (1998, 2001), Sharov and Nakahashi (1997) and Sharov et al. (2000). The SGS is very attractive because it uses an edge-based data structure, which can be efficiently parallelized with p-threads (Alauzet and Loseille 2009; Sharov et al. 2000).

The INRIA code `Wolf` involves all the above features, in short the Spalart–Allmaras model, MUSCL, Piperno and Gamma limiters and LU-SGS implicit formulation.

1.3. A multi-fluid incompressible model

1.3.1. *Introduction*

Flows with interfaces have been rather early addressed in the history of computer aided-simulation (see, for example, Harlow and Welch 1965). We give a first example of continuous CFD model for the flow of two non-miscible incompressible flows, using a level set representation of the interface. A discretization based on P^1 interpolation is then defined. Some idea about the resulting accuracy is given.

1.3.2. *Bi-fluid incompressible Navier–Stokes equations*

Let us consider the model problem of two incompressible immiscible fluids moving in a closed vessel Ω under the influence of gravity and without interface tension. Then, the bi-fluid incompressible Navier–Stokes system reads

$$\rho\frac{\partial \boldsymbol{U}}{\partial t} + \rho\nabla\cdot(\boldsymbol{U}\otimes\boldsymbol{U}) = \nabla\cdot(\nu(\rho)\nabla\boldsymbol{U}) - \nabla p + \rho\boldsymbol{g} \text{ in } \Omega \qquad [1.65]$$

$$\nabla\cdot\boldsymbol{U} = 0 \text{ in } \Omega \qquad [1.66]$$

$$\frac{\partial \boldsymbol{\rho}}{\partial t} + \nabla\cdot(\rho\boldsymbol{U}) = 0 \text{ in } \Omega, \qquad [1.67]$$

where $\boldsymbol{U}$ denotes the fluid velocity, p is the pressure, ρ is the density and $\boldsymbol{g}$ is the gravity volumic force. In this formulation, the density takes only two real positive values ρ_l and ρ_g in two subdomains separated by an *free interface* Γ_f. This model is mathematically well-posed when the viscosity $\nu(\rho)$ is not zero (see for example (Simon 1990)). In this chapter, we only consider a two-fluid inviscid model, that is, $\nu(\rho) = 0$.

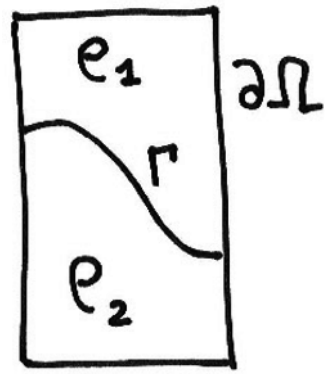

Figure 1.8. *Computational domain*

The interface Γ_f (Figure 1.8) is assumed to be a smooth enough surface, and in particular to have a well-defined normal $\mathbf{n}$:

$$\mathbf{n}\delta_\rho \;=\; \frac{1}{\rho_l - \rho_g}\nabla\rho,$$

where δ_ρ denotes the Dirac delta function on the interface Γ_f,

$$(\mathbf{n}\delta_\rho, \mathbf{f})_{(\mathcal{D}')^2\times(\mathcal{D})^2} = \int_{\Gamma_f} \mathbf{f}\cdot\mathbf{n}\mathrm{d}\Gamma_f.$$

To fix the ideas, ρ_l is the liquid density and ρ_g is the gas density. This system needs to be completed by *boundary conditions*. We assume a slip condition ($\boldsymbol{U}\cdot\mathbf{n}_{\partial\Omega} = 0$ on $\partial\Omega$ where $\mathbf{n}_{\partial\Omega}$ is the boundary normal) on some part of the boundary and a no-slip condition ($\boldsymbol{U} = 0$) on the other part. It is also possible to prescribe the pressure (we do not go into details).

Let us rewrite the Navier–Stokes system [1.65–1.67] in the *Level Set framework* as in Osher and Sethian (1988) (some very preliminary attempts were presented in

Dervieux and Thomasset 1981). We consider a smooth function ϕ, the level set (iso-value) 0 of which always represents the interface. The interface is then defined by the Heaviside step function H applied to ϕ:

$$H(\phi) = \begin{cases} 0 & \text{if } \phi < 0 \\ 1 & \text{if } \phi \geq 0. \end{cases}$$

The density is constant in each fluid and can be rewritten as a function of ϕ: $\rho(\phi) = \rho_g + (\rho_l - \rho_g)H(\phi)$. Equation [1.67] is then replaced by the interface advection equation

$$\phi_t + \nabla \cdot (\boldsymbol{U}\phi) = 0, \qquad [1.68]$$

and the governing equations for the fluid velocity $\boldsymbol{U}$ and pressure p, that is, relations [1.65] and [1.66] become

$$\rho(\phi)\frac{\partial \boldsymbol{U}}{\partial t} + \rho(\phi)\nabla \cdot (\boldsymbol{U} \otimes \boldsymbol{U}) + \nabla p - \rho(\phi)\boldsymbol{g} = 0 \qquad [1.69]$$

$$\nabla \cdot \boldsymbol{U} = 0. \qquad [1.70]$$

1.3.3. *Finite element approximation*

The system of equations [1.69–1.70] is solved explicitly with a standard P_1-P_1 projection finite element method as in Lesage et al. (2007). The interface is advected by means of equation [1.68], which is solved by a vertex-centered finite volume scheme as mentioned in section 1.1. We first introduce some notations. Let Ω_h be a discretization of the domain Ω of $\mathbb{R}^d$ and τ_h a mesh of Ω_h. The discretized domain Ω_h is assumed to be identical to Ω and thus denoted Ω in the following. For the sake of simplicity, the subscript $_h$ associated with the spatial discretization is also omitted for other symbols. $V = \{\psi \in \mathcal{C}^0(\bar{\Omega}) \mid \psi|_T \text{ is affine } \forall T \in \tau_h\}$ is the usual P_1 finite element space. V is spanned by the set of basis functions ψ_i, where ψ_i verifies for any vertex $\mathbf{x}_i$ of τ_h, $\psi_i(\mathbf{x}_i) = 1$ and $\forall j \neq i,\ \psi_i(\mathbf{x}_j) = 0$. Let $\mathbf{V} = V^d$, where d is the space dimension. The discretized multi-fluid variables are

$$\mathbf{U} = \sum_i \mathbf{U}_i\psi_i, \quad p = \sum_i p_i\psi_i \text{ and } \phi = \sum_i \phi_i\psi_i.$$

1.3.3.1. *Transfer operator into* $\mathbf{V}$

For any $u \in \mathbf{L}^2(\Omega)$, we denote by $\mathcal{P}u : \mathbf{L}^2 \mapsto V$ the function such that for any vertex $\mathbf{x}_i$ of τ_h,

$$\mathcal{P}u(\mathbf{x}_i) = \frac{\int_\Omega u\psi_i \,\mathrm{d}\mathbf{x}}{\int_\Omega \psi_i \,\mathrm{d}\mathbf{x}}.$$

And, for all $\mathbf{U} = (u, v) \in (\mathbf{L}^2(\Omega))^2$, we denote by $\mathcal{P}\mathbf{U} = (\mathcal{P}u, \mathcal{P}v)$ the transfer into $\mathbf{V}$. The transfer operator $\mathcal{P}$ transforms a discrete field that is constant by element into a discrete field that is continuous and piecewise linear.

The global algorithm for advancing in time follows.

Stage 1 (*prediction step*). The momentum is explicitly predicted due to equation [1.69] by omitting the pressure term:

$$\bar{\mathbf{U}}_i \;=\; \mathbf{U}_i^n - \frac{\Delta t}{|C_i|} \int_\Omega \psi_i \left(\nabla \cdot (\mathbf{U} \otimes \mathbf{U}) \;-\; \mathbf{g}\right) \mathrm{d}\mathbf{x}\,,$$

where $|C_i|$ is the vertex cell volume given by the following mass-lumped coefficient:

$$|C_i| \;=\; \sum_j \int_\Omega \psi_i \psi_j \,\mathrm{d}\mathbf{x}$$

and where $\mathbf{U}$ in RHS can be taken as $\mathbf{U}^n$ (time-explicit) or $\bar{\mathbf{U}}$ (time-implicit).

Stage 2 (*projection step*). The projection step evaluates the pressure using the predicted velocity and updates the velocity by imposing the null divergence constraint (relation [1.70]). To this end, the elliptic system

$$\int \frac{1}{\rho} \nabla p^{n+1} \cdot \nabla \psi \,\mathrm{d}\mathbf{x} \;=\; \frac{1}{\Delta t} \int \nabla \psi \cdot \bar{\mathbf{U}} \,\mathrm{d}\mathbf{x} \quad \forall\, \psi \in V$$

is solved in V by a finite element method. The correction based on the pressure gradient is constant by element. We transfer it into $\mathbf{V}$ and add to the velocity

$$\mathbf{U}^{n+1} = \bar{\mathbf{U}} + \Delta t\, \mathcal{P}\left(\frac{1}{\rho} \nabla p^{n+1}\right) \quad \text{and} \quad \mathbf{U}^{n+1} = 0 \;\text{ on } \partial\Omega.$$

Stage 3 (*level set advection and re-distancing*). Function ϕ is advected by means of equation [1.68] with $\mathbf{U}^n$ from time level n to time level $n+1$. This equation is solved by a second-order vertex-centered finite volume scheme. In a classical manner, the level set function is then re-distanced by replacing it by a distance function $\tilde{\phi}^{n+1}$, which is the signed distance to the set $\{\phi^{n+1} = 0\}$. For re-distancing, several authors recommend building a signed distance by iterating until steady-state a

Hamilton–Jacobi equation (see Sussman et al. 1998; Sethian 1999; Osher and Fedkiw 2002). In the present model, the method of Smolianski (2005) is followed and the adopted option is to rebuild a distance from a geometrical algorithm, defining ϕ_{dist} at each vertex as the minimum distance to discrete zero level of ϕ. At the end of Stage 3, a global volume conservation correction slightly different from that of Smolianski (2005) is applied; a small perturbation C_ϕ is added to ϕ in each point of the domain in order to solve iteratively the following conservation relation:

$$\overline{\phi}_h^{n+1} = \phi_h^{n+1} + C_{\phi_h}, \int H(\overline{\phi}_h^{n+1})dv = \int H(\phi_h^n)dv.$$

1.3.3.2. *Interface thickening*

In equation [1.69], the Heaviside function $H(\phi)$, that defines the density, is replaced by the thickened version H_η defined as follows:

$$H_\eta(\phi) = \begin{cases} 0 & \text{if } \phi < -\eta, \\ \frac{1}{2}(1 + \frac{\phi}{\eta} + \frac{1}{\pi}\sin(\frac{\pi\phi}{\eta})) & \text{if } |\phi| \leq \eta, \\ 1 & \text{if } \phi > \eta. \end{cases}$$

In practice, the parameter η is chosen as $\eta = k\ \Delta x$ with $k \approx 3$. This interface thickening reinforces the scheme stability but it carries an extra feature to tune and reduces accuracy to first order.

Once $\tilde{\phi}^{n+1}$ is obtained, we deduce ρ^{n+1}. We refer to Lesage et al. (2007) for further details.

REMARK 1.6.– Usually, a collocated $P_1 - P_1$ approximation of the incompressible Navier–Stokes equations is not stable. The stabilization is obtained by the introduction of the transfer $\mathcal{P}$, which introduces a stabilization of same type as the Rhie–Chow stabilization (Rhie and Chow 1983).

REMARK 1.7.– The above simplified projection stage induces a strong dependency of the solution to time step size, which can be reduced by introducing an auxiliary time step in the projection stage.

1.3.4. *Error estimate for the level set advection*

This section is devoted to justifying the re-distancing and gives elements to choose the spatial discretization. Let us assume that we know how to numerically advect

the level set ϕ by approximating it with a discrete level set function ϕ_h. Since ϕ is advected with a first-order hyperbolic model, according to variational theory, the usual convergence property for ϕ_h on non-regular meshes is a convergence in L^p, but we can manage a high-order k of accuracy in that norm. Let us examine the consequence for the corresponding $\chi_\phi = H(\phi)$ functions.

PROPOSITION 1.1.– Let ϕ be a $L^p(Q)$ function where $Q = \Omega\times]0,T[$ is the flow integration domain in space and time. Let $(\phi_h)_h$ a sequence of $L^p(Q)$ and we assume that

$$\begin{array}{l} meas(-\zeta \leq \phi \leq \zeta) \leq K_1\zeta \\ \parallel \phi_h - \phi \parallel_{L^p(Q)} \leq K_2 h^k, \end{array} \qquad [1.71]$$

with h, ζ sufficiently small, k the convergence order on ϕ and K_1,K_2 independent of h, ζ then for all real number $q \geq 1$, there is a constant $C(q)$ independent of h such as[6]

$$\parallel H(\phi_h) - H(\phi) \parallel_{L^q(Q)} \leq C(q) h^{\frac{2k}{3q}}. \square \qquad [1.73]$$

---

6 PROOF.– The above integral can be analyzed as follows:

$$\int_Q |H(\phi_h) - H(\phi)| d\Omega dt \;\leq\; meas(|\phi| \leq \zeta) + meas(|\phi_h| \leq \zeta)$$

$$+meas(\phi \geq \zeta, \phi_h \leq -\zeta) + meas(\phi \leq -\zeta, \phi_h \geq \zeta)$$

$$\leq\; meas(|\phi| \leq \zeta) + meas(|\phi_h| \leq \zeta) + meas(|\phi_h - \phi| \geq 2\zeta)$$

but $meas(|\phi| \leq \zeta) \leq K_1\zeta$, according to [1.71] and $meas(|\phi_h| \leq \zeta) \leq meas(|\phi| \leq 2\zeta) + meas(|\phi_h - \phi| \geq \zeta)$, that is $meas(|\phi_h| \leq \zeta) \leq 2K_1\zeta + meas(|\phi_h - \phi| \geq \zeta)$ thus $\int_Q |H(\phi_h) - H(\phi)| d\Omega dt \leq 3K_1\zeta + 2meas(|\phi_h - \phi| \geq \zeta)$. The last sum part can be estimated as follows:

$$meas(|\phi_h - \phi| \geq \zeta) = \int_{|\phi_h - \phi|^p \geq \zeta^p} 1 \; d\Omega dt \leq \frac{1}{\zeta^p} \int_{|\phi_h - \phi|^p \geq \zeta^p} |\phi_h - \phi|^p d\Omega dt$$

$$\leq \frac{1}{\zeta^p} ||\phi_h - \phi||^p_{L^p(Q)} \qquad [1.72]$$

and with [1.72], we get $\int_Q |H(\phi_h) - H(\phi)| d\Omega dt \leq 3K_1\zeta + \frac{2}{\zeta^p} ||\phi_h - \phi||^p_{L^p(Q)}$. By choosing $\zeta = h^\theta$, $\theta = \frac{pk}{p+1}$, we get

$$\int_Q |H(\phi_h) - H(\phi)| d\Omega dt \leq 3K_1 h^\theta + 2K_2^2 h^{-p\theta} h^{pk} \leq (3K_1 + 2K_2^2) h^{\frac{pk}{p+1}},$$

thus $\int_Q |H(\phi_h) - H(\phi)| d\Omega dt \leq (3K_1 + 2K_2^2) h^{\frac{pk}{p+1}}$ or equivalently, for $q \geq 1$: $||H(\phi_h) - H(\phi)||_{L^q} \leq K_5 h^{\frac{pk}{q(p+1)}}$. $\square$

In the global scheme presented, we can consider that $q = 1, k = p = 2$, which gives a convergence of at least $4/3$, better than order one.

REMARK 1.8.– Estimate [1.73] is not optimal. The order of accuracy is between $pk/(q(p+1))$ and k. However, the possible loss of accuracy shown in [1.73] is related to inaccuracy on the interface when ϕ_h is not sufficiently smooth. In practice, ϕ_h is re-initialized by being set to a signed distance to interface, although that extra step introduces other type of errors. In the sequel, we shall refer to the case $p = 2, q = 1$, which gives an order between $\frac{2k}{3}$ and k. □

REMARK 1.9.– The above analysis easily extends to a thickened interface function H_η, giving an estimate for $||H_\eta(\phi_h) - H_\eta(\phi)||_{L^q}$. Now the exact context should rely on H, and the deviation

$$||H_\eta(\phi) - H(\phi)||_{L^q} \leq K_6\eta$$

needs to be taken into account in the global error estimate. □

REMARK 1.10.– The above analysis suggests that in order to get a global second-order convergence for a level-set multifluid calculation, a second-order Navier–Stokes approximation should be combined with a third-order accurate advection of the function ϕ. □

REMARK 1.11.– The above analysis applies only if a second-order convergence of the velocity is true. The latter convergence is generally not reached when the viscosity is discontinuous across the interface. □

1.3.5. *Provisional conclusion on scheme accuracy*

The proposed level-set $P1$ approximation, because of the level set formulation, is not advecting the interface with first-order accuracy, but the analysis shows that the accuracy can be bounded to order $3/2$, as concerns the interface capture. This takes into account neither any thickening of interface, nor the accuracy of interface representation in moment and pressure equations.

The incompressible diphasic component of the multi-physics platform *NiceFlow* of Lemma[7], which will be used in this book, involves all the above features.

7 Available at: https://www.lemma-ing.com/.

1.4. Appendix: circumcenter cells

Median-based cells (Figure 1.9) are the exact counterbalance of the P^1 finite element formulation. They are well adapted to non-stretched unstructured meshes. It was observed (in particular in Barth 1994) that for certain highly stretched meshes, the median tessellation results in an important deterioration of the accuracy. In that case, according to Barth (1994), it is recommended to replace the median cells by the so-called circumcenter cells. Note, however, that in most of our viscous *mesh adaptative* application, the adequation of the mesh allows keeping the median option.

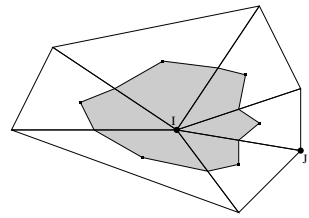

Figure 1.9. *Median cell construction in 2D*

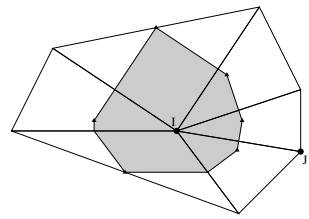

Figure 1.10. *Circumcenter cell construction in 2D*

1.4.1. *Two-dimensional circumcenter cells*

According to an idea of Barth (1994), a cell is built around each vertex by joining the mid-edges with the center of the smallest circle containing the considered triangle. We observe that this center is located at middle of largest edge in a triangle involving an angle larger or equal to 90°. As a result, a mesh made of rectangular triangles has cells, which are rectangles (see Figure 1.12).

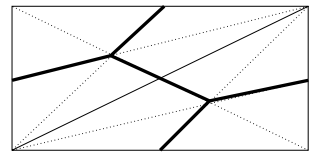

Figure 1.11. *Trace of median division on two elements*

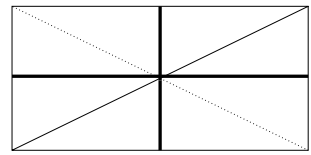

Figure 1.12. *Trace of circumcenter division on the same two elements*

1.4.2. *Three-dimensional circumcenter cells*

A 3D circumcenter cell can be defined as the union of polyhedra such that, for a given element, the common surface between two neighboring cells joins:

– the middle of the edge connecting these two vertices;

– the "surface center" of the faces of the element having this edge in common;

– the "volume center" of the element;

where

– the *"surface center"* of a given face is the center of its circumscript circle if it comprises only acute angles, otherwise, it is the middle of its longest edge;

– the *"volume center"* of an element is the center of its circumscript sphere if this center is located inside the element; otherwise, it is the *"surface center"* of the face which is closest to the center of its circumscript sphere.

In practical applications, we use the circumcenter cells for stretched meshes with aspect ratios larger than 10 and for Cartesian meshes. Indeed, it is proposed in Gourvitch et al. (2004) to combine the Barth cell construction with a particular splitting of cubes into six identical tetrahedra in a Cartesian mesh forming two prisms (Figures 1.4.2 and 1.14). With this combination, the cells are exactly cubes and the scheme is equivalent to the usual vertex-centered-finite volume scheme and the fifth-order accuracy of the V6 scheme holds for advection.

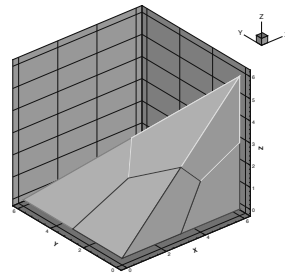

Figure 1.13. *Trace of median cells on a tetrahedron in a particular Cartesian mesh (Gourvitch et al. 2004). For a color version of this figure, see www.iste.co.uk/dervieux/meshadaptation1*

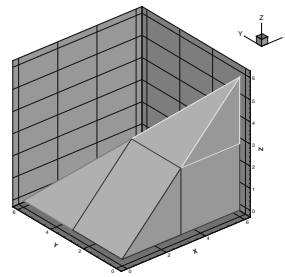

Figure 1.14. *Trace of a circumcenter cell on a tetrahedron in a particular Cartesian mesh (Gourvitch et al. 2004). For a color version of this figure, see www.iste.co.uk/dervieux/meshadaptation1*

1.5. Notes

Both previous models may have solutions fields involving small features challenging for numerical methods. Both flows may involve boundary layers, which are generally much smaller than the mean size of a useable mesh. Compressible flows can have shocks, and multiphase flows show discontinuous interfaces with possibly large change in fluid properties. All these features can be much better addressed when a mesh adaptation is applied, as we shall try to show in this book.

The *upwind-element scheme* described in this chapter was introduced in Stoufflet et al. (1996) for the special treatment of very stiff flows, involving strong bow shocks at large Mach numbers. Positivity statements extend to second-order accuracy the contribution of Farhat et al. (2001).

Further details on the positivity study described in this chapter can found in Cournède et al. (2006).

Interface representation. The numerical representation of an interface in a flow is a difficult problem, which has motivated thousands of publications. In many situation, we need to describe the interface moving in a fixed mesh. Among the methods that can achieve this, we distinguish thick interface techniques (often based on Riemann solvers; see Saurel and Abgrall 1999), interface direct tracking (Unverdi and Tryggvason 1992), volume of fluid (VOF) (Popinet and Zaleski 1999) and level set (LS) (Sethian 1999; Osher and Fedkiw 2002). The issue of the conservation of each phase is well addressed by VOF and not by LS, which has other attracting features (easy extension to high order and better discretization of curvature, among others). The conservative LS of Eça et al. (2007) replaces the distance function by a smooth transition between one phase density and the other one. This transition spreads over a small number of mesh sizes Δx, but this small number needs be sufficiently increased when we try to converge to the exact solution with a reasonable order.