ADVANCED AERODYNAMICS AND MULTIPHYSICS

CFD e-Learning

Mesh and discretization

G. Puigt and H. Deniau

Release 1.1
January 2011

Centre Européen de Recherche et de Formation Avancée en Calcul Scientifique
42 avenue Coriolis – 31057 Toulouse Cedex 1 – France
Tél : +33 5 61 19 31 31 – Fax : +33 5 61 19 30 00
http://www.cerfacs.fr – e-mail: secretar@cerfacs.fr

Contents

Chapter 1

Navier-Stokes equations

1.1 Introduction

In this chapter, the Navier-Stokes equations are derived from physical assumptions. The main goal is to give explanations on the Navier-Stokes closure and these explanations are justified by physical assumptions. Moreover, the Navier-Stokes equations include formally the conservation principle – this is why it is generally said "Let solve the Navier-Stokes equations in conservative form" – and the conservation principle will be justified.

Remark 1.1.1 *Notations and definitions of this chapter will be valid for the whole document.*

Remark 1.1.2 *The Navier-Stokes equations presented in this document are only valid for the continuous regime. The flow is in continuous regime if the mean free path*[1] *x is sufficiently short. There exist a microscopic theory which derives the Navier-Stokes equations from rarefied flow equations (see [3] in French). This method is out of purpose and will not be addressed in this document.*

Remark 1.1.3 *The Navier-Stokes equations can also be derived from mathematical considerations [5].*

1.2 Notations

Let Ω be an open space of $\mathbb{R}^k$ ($k = 2$ or $k = 3$). Consider a fluid during a time interval $[0,T']$. Let's define the following symbols:

- $x \in \mathbb{R}^k$: a point in Ω,
- $t \in [0, T']$: a time instant,
- $\rho(x, t) \in \mathbb{R}_+^* \times [0, T']$: the density field,
- $\vec{u}(x, t) \in \mathbb{R}^k \times [0, T']$: the velocity vector field,
- $p(x, t) \in \mathbb{R}_+^* \times [0, T']$: the pressure field,
- $e(x, t) \in \mathbb{R}_+^* \times [0, T']$: the internal energy associated with molecules movement inside an elementary volume,

[1] average distance covered by a moving particle between two successive impacts.

- $T(x,t) \in \mathbb{R}_+^* \times [0,T']$: the temperature field associated with internal energy.

Let us define mathematical objects based on the scalar variable q, on the vector variables v and w, and on the matrices A and B:

$$\begin{aligned}
\partial_t q &= \frac{\partial q}{\partial t}, \\
\partial_j q &= \frac{\partial q}{\partial x_j}, \\
\nabla q &= \text{gradient of } q \text{ (vector)}, \\
\nabla v &= \text{ second order tensor} (\nabla v)_{ij} = \partial_i v_j, \\
\nabla . v &= \text{divergence of } v : \nabla . v = \sum_i \partial_i v_i \\
\nabla . A &= \text{vector which } j\text{-th component is} \sum_i \partial_i A_{ij}, \\
\vec{u}.\vec{v} &= \text{ scalar product}, \\
v \nabla v &= v_i \partial_i v, \\
A : B &= \sum_{ij} A_{ij} B_{ij}, \\
\vec{v} \otimes \vec{w} &= \text{second order tensor } (\vec{v} \otimes \vec{w})_{ij} = v_i \, w_j \, , \\
\Delta \vec{v} &= \text{Laplacian of } \vec{v} : \Delta v = \nabla . (\nabla v) \, .
\end{aligned}$$

Remark 1.2.1 *Einstein summation convention must be applied to all formulae: repeating indices means summation with respect to this subscript.*

1.3 Mass conservation

Let A be a regular sub-domain of Ω. The conservation principle for the mass is:

The mass variation in A is equal to the mass flux across the boundary ∂A of A.

It results:

$$\partial_t \int_A \rho dx \; = \; - \int_{\partial A} \rho \vec{u}.\vec{n} \; ds \, ,$$

where $\vec{n}$ is the unit outward vector, normal to the boundary and defined at each point of the boundary. Since A is regular, Stokes formula leads to:

$$\int_A \nabla . (\rho \vec{u}) dx \; = \; \int_{\partial A} \rho \vec{u}.\vec{n} \; ds \, ,$$

and

$$\partial_t \int_A \rho dx + \int_A \nabla . (\rho \vec{u}) dx \; = \; 0.$$

Since A is defined arbitrary, the mass conservation principle leads to:

$$\partial_t \rho + \nabla . (\rho \vec{u}) \; = \; 0. \tag{1.1}$$

Remark 1.3.1 *This equations is also called continuity equation.*

1.4 Momentum conservation

The momentum conservation equation comes from Newton's law:

$$\sum \textit{ Forces } = \textit{ acceleration given by the forces}$$

Let a fluid particle be in x at time t. The particle will be in $x + u(x,t)\delta t$ at time $t + \delta t$ and its acceleration is:

$$\lim_{\delta t \to 0} \frac{1}{\delta t} \left(\vec{u}(x + \vec{u}(x,t)\, \delta t, t + \delta t) - \vec{u}(x,t) \right) = \partial_t \vec{u} + \vec{u}.\nabla \vec{u}\,.$$

Now, let us define the forces which act on A:

- external forces $\displaystyle\int_A f\, dx$ with f external force per volume unit.
- pressure and viscous forces:

$$\int_{\partial A} (\tau - p\mathbf{I})\vec{n} ds \;=\; \int_A (\nabla p - \nabla.\tau)\, dx\,, \tag{1.2}$$

  where τ is the shear stress tensor, $\mathbf{I}$ is the unit diagonal tensor, $\vec{n}$ is the unit outward normal vector on ∂A. The equivalence of left and right hand sides of Eq. 1.2 is due to Stokes formula.

Therefore, it comes:

$$\int_A \rho(\partial_t \vec{u} + \vec{u}.\nabla \vec{u}) dx \;=\; \int_A (f - \nabla p + \nabla.\tau) dx\,,$$

and therefore

$$\rho(\partial_t \vec{u} + \vec{u}.\nabla \vec{u}) + \nabla p - \nabla.\tau = f\,.$$

Since

$$\rho \partial_t \vec{u} = \partial_t(\rho \vec{u}) - (\partial_t \rho)\, \vec{u} = \partial_t(\rho \vec{u}) + \nabla.(\rho \vec{u})\, \vec{u} = \partial_t(\rho \vec{u}) + \nabla.(\rho \vec{u} \otimes \vec{u}) - \rho \vec{u}\, \nabla \vec{u}\,,$$

one finally obtains:

$$\partial_t(\rho \vec{u}) + \nabla.(\rho \vec{u} \otimes \vec{u}) + \nabla.(p\mathbf{I} - \tau) = f\,. \tag{1.3}$$

1.5 Energy conservation and state equation

Let A be a volume moving with the fluid. The specific total energy E in A is the sum of:

- the specific internal energy e,
- the kinetic energy $\|\vec{u}\|^2/2$.

Therefore, $E = e + \|\vec{u}\|^2/2$ and the energy in A is:

$$\int_A \rho E \, dx \, .$$

The energy E is the sum of mechanical work and heat. The force work is

$$\int_{\partial A} \vec{u}.(f + \tau - p\mathbf{I})\vec{n} ds \, .$$

Let q be the energy flux density transported by thermal conduction; it comes:

$$\frac{d}{dt}\int_A \rho E \, dx = \int_A \Big(\partial_t \rho E + \nabla.(\rho \vec{u} E)\Big) dx = \int_A \vec{u}.f \, dx + \int_{\partial A} \Big(\vec{u}(\tau - p\mathbf{I}) - q\Big)\vec{n} ds \, .$$

With Stokes formula, and since the relation is true for all A, the energy conservation equation writes:

$$\partial_t(\rho E) + \nabla.\Big(\vec{u}(\rho E + p)\Big) = \nabla.\,(\vec{u}\tau - q) + \vec{u}.f \, . \tag{1.4}$$

1.6 Navier-Stokes equations

Finally, the Navier-Stokes equations have been obtained. One can write it without external force ($f = 0$):

$$\begin{cases} \partial_t \rho + \nabla.(\rho \vec{u}) = 0 \\ \\ \partial_t(\rho \vec{u}) + \nabla.(\rho \vec{u} \otimes \vec{u}) = \nabla.\sigma \\ \\ \partial_t(\rho E) + \nabla.(\rho \vec{u} E) = \nabla.\,(\vec{u}\sigma) - \nabla.q \, , \end{cases} \tag{1.5}$$

with $\sigma = -p\mathbf{I} + \tau$ and τ is the shear stress tensor.

The Navier-Stokes system of equations is **open** with **more unknown variables than equations**. Additional relations are necessary to **close** it and link intermediate variables σ (or τ) and q to the main variables ρ, $\vec{u}$ and E. The closure is done by providing two behavior laws for τ et q and the state law which links the pressure with the main variables.

Remark 1.6.1 *In Computational Fluid Dynamics tools, gravity is not taken into account, its effects being several order of magnitude lower than that of pressure and viscous forces.*

1.6.1 Conservative form of the equations

Deriving the Navier-Stokes equations from physical considerations is based on an integral formulation paradigm. This approach shows the principle of conservation: for a stationary flow, the integration of the divergence term on a control volume leads to boundary fluxes such that inward and outward fluxes are equal. It is the principle of conservation form of equations.

On the computational side, this conservative formulation has several advantages:

- guaranty of no mass build during the computational process,
- guaranty of no momentum build during the computational process,
- guaranty of no energy built during the computational process.

1.7 Behavior laws

1.7.1 Law for τ (newtonian fluid)

One needs to understand that the shear stress tensor τ depends by nature on the fluid viscosity. The viscosity measures the resistance of a flow and induces constraints inside the flow.

Planar Couette flow

The fluid viscosity can be measured experimentally with the planar Couette's process represented in Fig. 1.1.

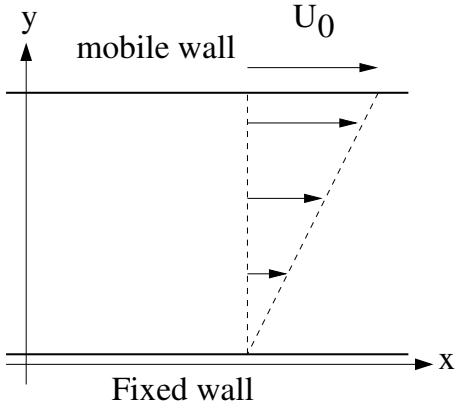

Figure 1.1: *Principle of the planar Couette's flow.*

In this experiment, the fluid between two parallel planar surfaces, with a distance of h meters between them and with one in a steady translation at constant velocity U_0, is put in movement thanks to the viscous effects. For the sake of clarity, let the upper wall be moving and let the lower wall be fixed (Fig. 1.1). The flow movement only results from the movement of the upper wall if there is no extern force.

For the permanent regime of some fluids, the experiment shows that the velocity profile between both planar surface is linear. Moreover, this solution is maintained if and only if a force F applied on an area A is such that:

$$\frac{F}{A}\frac{h}{U_0} = Cst\,.$$

The dynamic viscosity of the fluid μ is the positive or null constant:

$$\frac{F}{A} = \mu\frac{U_0}{h}\,. \tag{1.6}$$

Eq. 1.6 can be applied to fluids such as air or water and is the origin of the rheologic behavior of *newtonian* fluids.

Consequence

Eq. 1.6 introduces the notion of *constraint* or force per surface unit F/A and since the force direction is parallel with the surface, the constraint is defined tangential:

$$\tau_{xy} = \frac{F}{A}\,. \tag{1.7}$$

The generalization of Eq. 1.7 to 3D configurations gives an expression for the shear stress tensor.

Generalization and general expression

In air, the internal friction only appears when several parts of the flow move at different velocities. As a consequence [7], τ must depend on the velocity gradient with respect to the three directions in space. If the gradient is not too high, it is admitted that the movement due to viscosity only depends on first order derivatives of the velocity and τ depends linearly on $\nabla\vec{u}$. Since τ is zero for $\vec{u} = Cst$, τ expression does not contain terms independent of $\nabla\vec{u}$. Moreover, $\tau = 0$ if the fluid does a full uniform rotation since there is no friction in the flow. For a uniform rotation at angular velocity ω and at radius r, the velocity is $\omega \wedge r$. The linear combinations of $\nabla\vec{u}$ which are zero for $\vec{u} = \omega \wedge r$ are based on:

$$\nabla\vec{u} + \nabla\vec{u}^T\,.$$

τ depends on symmetric combinations of $\nabla\vec{u}$ and using the most general tensor notation, one finds finally:

$$\tau = \mu(\nabla\vec{u} + \nabla\vec{u}^T) + \xi\nabla.\vec{u}\mathbf{I} \text{ with } \mu \geq 0\,, \tag{1.8}$$

where μ and ξ represent two scalar values with a definition close to Lamé's coefficients for linear elasticity. The first real scalar is called dynamic viscosity and the other one is the second viscosity coefficient. The dynamic viscosity is a positive coefficient. Eq. 1.8 is known as *Newton's law for the viscosity* and the fluid which respects Eq. 1.8 is said *newtonian*. The kinetic theory [9] gives a validity relation for Eq. 1.8:

$$\frac{\tau_m}{c}\left|\nabla\vec{u}\right| << 1 \tag{1.9}$$

with τ_m the mean free path and c a characteristic molecular velocity such as the speed of sound.

Eq. 1.8 can be written in a different way, introducing spheric and deviator contributions [4]:

$$\tau = \mu\left(\nabla\vec{u} + \nabla\vec{u}^T - \frac{2}{3}\nabla.\vec{u}\mathbf{I}\right) + \left(\xi + \frac{2}{3}\mu\right)\nabla.\vec{u}\mathbf{I} \text{ with } \mu \geq 0 \text{ and } \xi + \frac{2}{3}\mu \geq 0$$

which shows that

$$\eta = \xi + \frac{2}{3}\mu$$

plays the role of a volume viscosity, in the sense that it is associated with volume variations.

1.7.2 Stokes' hypothesis

Stokes' hypothesis introduces a new level in the modeling for τ. It comes from thermodynamic assumptions at equilibrium: the mechanic pressure $p_m = p + \eta \nabla . \vec{u}$ is strictly equal to the dynamic pressure p, leading to:

$$\eta = 0 \,,$$

and so

$$3\xi + 2\mu = 0 \,.$$

Stokes' hypothesis means that the relaxation time for dynamic and mechanic pressures to be equal is infinitely small. With Stokes' relation, Eq. 1.8 becomes the *Newton-Stokes law*:

$$\tau = \mu \left(\nabla \vec{u} + \nabla \vec{u}^T - \frac{2}{3} \nabla . \vec{u} \mathbf{I} \right) \,. \tag{1.10}$$

At this level of modeling, one can compute the sear stress if a law is given for the viscosity.

1.7.3 Thermal flux and Fourier's law for the heat flux q

Remark 1.7.1 *We only consider in this document the thermal conduction as mode for heat transfer.*

Let q be the energy flux density transported by thermal conduction. If the temperature gradient is low, q can be expressed as a power of the temperature gradient. The first order term [7] is written:

$$q = -\lambda \nabla T \,. \tag{1.11}$$

where λ is called thermal conductivity. This relation is known as *Fourier's law* and is valid under the following relation [9] :

$$\frac{\tau_m}{T} \left| \nabla T \right| << 1 \tag{1.12}$$

where τ_m is the mean free path.

The thermal conduction coefficient λ is always positive since the energy flux comes from high temperature regions to low temperature regions. Therefore, q and ∇T must have opposite signs. λ is classically related to μ with:

$$\lambda = \frac{C_p \, \mu}{Pr}, \tag{1.13}$$

where C_p is the heat capacity at constant pressure (it depends on the gas and on the state equation) and Pr is a non-dimensional number, called the Prandtl number. Pr represents the ratio of the thermal diffusion time over the dynamic diffusion time for a fixed reference length.

1.7.4 Law for the viscosity

The viscosity depends on the temperature in general. For air in non extreme conditions of temperature and pressure, μ follows Sutherland's law:

$$\mu(T) = \mu_{ref} \left(\frac{T}{T_{ref}} \right)^{1.5} \left(\frac{T_{ref} + 110.4}{T + 110.4} \right) \,, \tag{1.14}$$

where $T_{ref} = 273.15$ K and $\mu_{ref} = 1.711\ 10^{-5}\ Kg.m^{-1}.s^{-1}$. For a temperature lower than $1500K$, Eq. 1.14 is a good approximation of μ. For aircrafts or turbomachinery flows, it is the classical (preferred) relation to define μ.

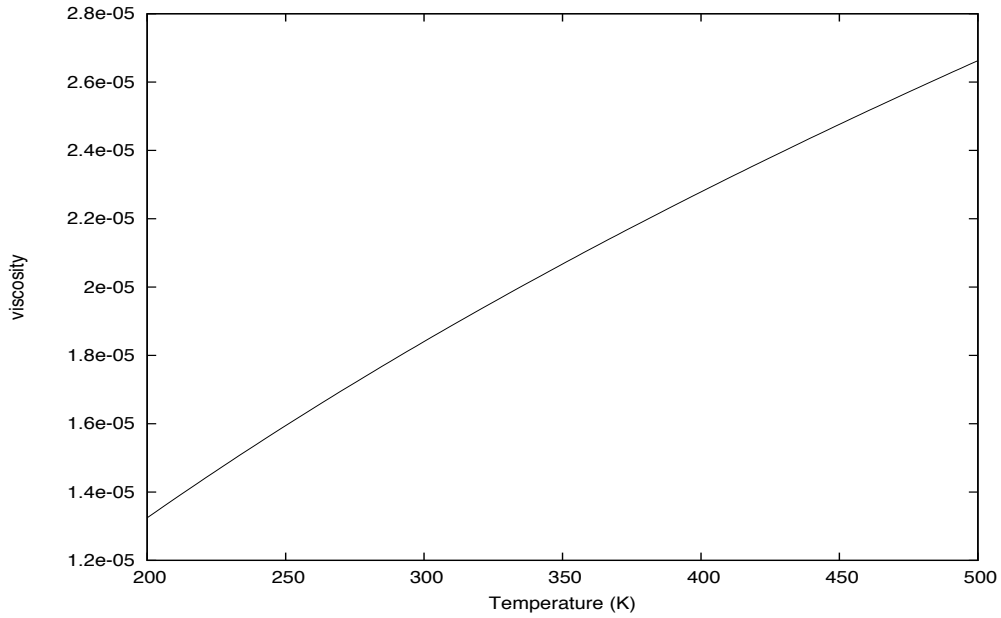

Figure 1.2: *Sutherland's law: evolution of the viscosity with respect to the temperature.*

1.7.5 Some remarks on closure

Stokes' hypothesis leads to a simplification of the shear stress tensor with a zero Lamé's coefficient. The volume viscosity is neglected, which is only valid for a pure monoatomic gas and some studies have shown that Stokes' hypothesis was false for air. The volume viscosity can be computed with the following law [6]:

$$\eta = \mu\left(7.821\exp(-16.8T^{-1/3})\right).$$

For most of the CFD solvers that compute air flows, the Prandtl number, which measures the ratio of the thermal conductivity on the diffusion one, is assumed constant, equal to 0.72. An analysis of some measures lead Papin [8] to propose a function of the temperature [8] :

$$Pr(T) = 0.66 + 0.1\exp\left(-\frac{T-123.15}{300}\right)$$

computed from a sampling [2] (valid between $120K$ and $670K$). One deduces from this relation that the Prandtl number varies between 0.686 and 0.735 with a value near 0.72 for 273K approximately (Fig. 1.3).

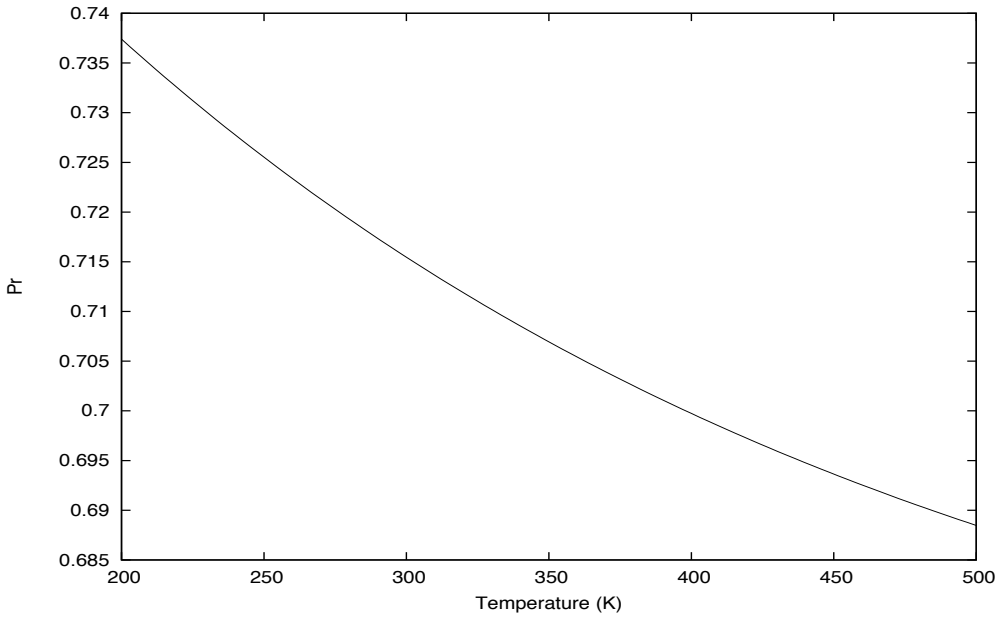

Figure 1.3: *Evolution of the Prandtl number with the temperature (K).*

In [6], a law for the thermal conductivity independent of the kinetic viscosity is proposed for air:

$$\lambda = \frac{2.64638\ 10^{-3} T^{3/2}}{T + 254.4\ 10^{-12/T}}\,.$$

1.8 Perfect state equation

1.8.1 Definition

A thermodynamic state is characterized by two independent variables ρ and T, or ρ and S where S represents the entropy. The state law consists in giving the functions P or g such that:

$$p = P(\rho, T) \text{ or } p = g(\rho, S)\,.$$

1.8.2 Reminder on thermodynamic

The first principle

In thermodynamics, the internal energy per mass unit e varying between two equilibrium states with an infinitesimal process follows:

$$de = \delta w + \delta q$$

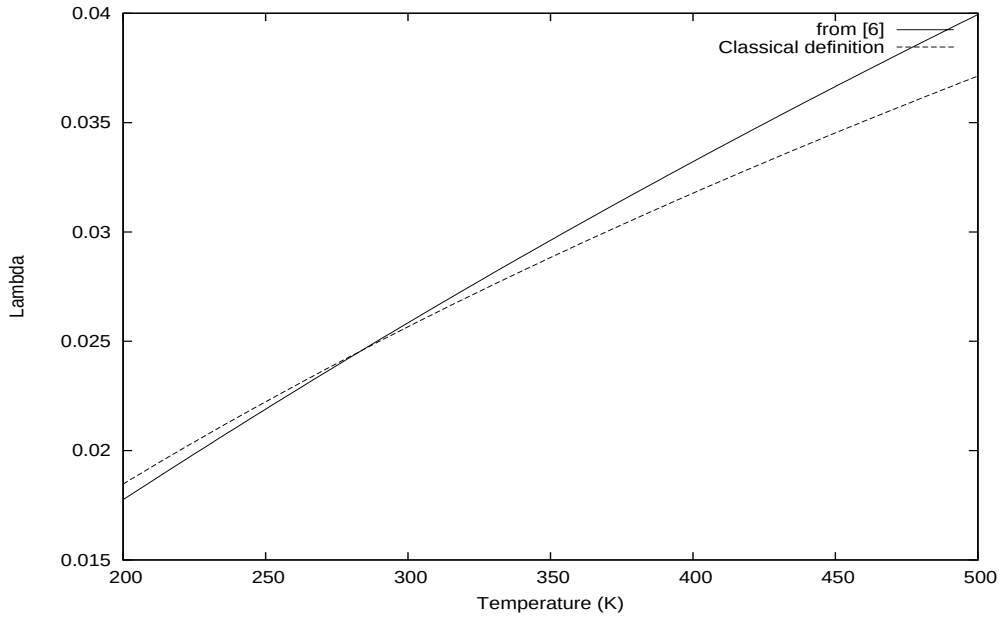

Figure 1.4: *Thermal conductivity: comparison of expression from [6] with Eq. 1.13.*

where δw and δq represent the specific work and the specific heat given to the system. It is also possible to introduce the specific enthalpy h:

$$h = e + \frac{p}{\rho},$$

and the heat capacities at constant pressure or constant volume, respectively:

$$C_p = \left(\frac{\partial h}{\partial T}\right)_p \text{ et } C_v = \left(\frac{\partial e}{\partial T}\right)_v .$$

The second principle

In thermodynamics, the second principle is written in two parts:

1. there exists a temperature T and a state variable S called specific entropy such that, for all infinitesimal evolution (reversible or not):

$$de = TdS - pd\left(\frac{1}{\rho}\right),$$

2. The evolution principle says that, for a closed system (no mass exchange with "the rest of the world"), the following relation is true:

$$dS \geq \sum_{i \in J} \left(\frac{\delta q_i}{T_i}\right),$$

where J represents all thermal sources i at temperature T_i which bring the heat transfer δq_i. Equality is only obtained for reversible evolutions.

State variables and exact linearization

The internal energy e and the enthalpy h are state variables which can be differentiated exactly, and

$$de = \left(\frac{\partial e}{\partial T}\right)_\rho dT + \left(\frac{\partial e}{\partial \rho}\right)_T d\rho = C_v dT + \left(\frac{\partial e}{\partial \rho}\right)_T d\rho\,,$$

$$dh = \left(\frac{\partial h}{\partial T}\right)_p dT + \left(\frac{\partial h}{\partial p}\right)_T dp = C_p dT + \left(\frac{\partial h}{\partial p}\right)_T dpd$$

1.8.3 Perfect gas model

The kinetic theory for perfect gas has been essentially written by Maxwell in 1859. It is based on the molecular representation of gas suggested by Avogadro in 1811 and on some statistical considerations. At human being scale, the number of molecules is simply enormous (remember the meaning of Avogrado's number $\mathcal{N}_A = 6.02253\ 10^{23}$). Following the kinetic theory, the state law for a perfect monoatomic gas is written:

$$p = nkT\,, \tag{1.15}$$

where k is the Boltzmann's constant ($k = 1.3806581\ 10^{-23} J.K^{-1}$) and n is the number of molecules per volume unit.

Let $\mathcal{M}$ be the molar mass. The density ρ is

$$\rho = \frac{n\mathcal{M}}{\mathcal{N}_\mathcal{A}}\,.$$

and it comes from Eq. 1.15 :

$$p = \rho \frac{k\mathcal{N}_A}{\mathcal{M}} T\,.$$

The product $\mathcal{R} = k\mathcal{N}_A$ represents the perfect gas constant $\mathcal{R} = 8.3144\, J.K^{-1}.mol^{-1}$ and $R = \mathcal{R}/\mathcal{M}$ is the perfect gas constant for the considered gas. To conclude, a perfect gas is characterized by:

$$p = \rho RT\,, \tag{1.16}$$

with $R = 287$ for air.

One can also prove that **enthalpy h and internal energy e are functions of the temperature only for a perfect gas**, leading to

$$de = C_v(T)dT \text{ et } dh = C_p(T)dT\,,$$

with $C_p(T) - C_v(T) = R$. For a monoatomic gas, C_p and C_v are truly constant numbers, while they vary for polyatomic gases.

For transonic flows around civil aircraft and for turbomachinery, one assumes air to be a perfect gas. This means that air is *perfect following the thermodynamic theory* and also that

it is *a perfect polytropic gas characterized by constant C_p and C_v coefficients.* For this gas, the polytropic coefficient γ is

$$\gamma = \frac{C_p}{C_v}\,.$$

Now, let us compute e. Following perfect gas relations, it comes easily that:

$$e = C_v T \text{ and } h = C_p T\,,$$

and finally, denoting E to total energy per mass unit, one has:

$$E = C_v T + \frac{\|\vec{u}\|^2}{2}\,,$$

with $\|\vec{u}\|$ euclidian norm of the velocity vector.

1.9 The Navier-Stokes system of equations

The Navier-Stokes system of equations has finally been closed. Its conservative form is:

$$\left\{\begin{array}{lcl} \partial_t \rho + \nabla.(\rho\vec{u}) & = & 0 \\ \partial_t(\rho\vec{u}) + \nabla.(\rho\vec{u}\otimes\vec{u}) + \nabla p - \nabla.\tau & = & 0 \\ \partial_t(\rho E) + \nabla.\Big(\vec{u}(\rho E + p)\Big) & = & \nabla.\left(\vec{u}\tau + \lambda\nabla T\right) \end{array}\right. \tag{1.17}$$

where $\tau = \mu(\nabla\vec{u} + \nabla\vec{u}^T) - \dfrac{2\mu}{3}\mathbf{I}\nabla.\vec{u}$ and $\lambda = \dfrac{C_p\mu}{Pr}$.
To solve it, one needs boundary conditions. There exist a lot of possibilities for boundary conditions, depending on the flow physics and on the application.

Two numbers (without dimension) characterize the flow.

- the Mach number:
$$M = \frac{\|\vec{u}\|}{c}\,,\ \text{where } c \text{ is the sound velocity,}$$
- the Reynolds number:
$$Re = \frac{\rho\|\vec{u}\|L}{\mu}\,,\ \text{where } L \text{ is a characteristic length of the object in movement.}$$

The mach number gives the importance of the flow movement with respect to sound velocity. **Flows at Mach number lower than 0.1 are assumed incompressible and are solutions of Eq. 1.17 with the hypothesis $\rho = Cst$.**

The Reynolds' number measures the importance of viscosity in the flow relatively to momentum forces. For high Reynolds number flows, the viscous force is lower than the kinetic force on the object in movement. Low Reynolds number flows are generally organized, easily reproducible (laminar flow). For high Reynolds flows, the importance of the viscosity if lower and its regularization effects on the flow are negligible. In this case, variables are varying in time and space and this kind of flow is said turbulent.

There is no criterium to decide if the flow is laminar or turbulent a priori, except for some very simple (academic) cases. Moreover, the mechanisms for a flow to turn from the laminar regime to the turbulent one, which is called transition, are well-understood but can not be estimated nor located *a priori*.

Chapter 2

Towards the numerical simulation of the Navier-Stokes equations

2.1 Introduction

This chapter is devoted to the introduction of numerical techniques for discretizing the Navier-Stokes equations. The analysis is done on a model problem, namely the linear heat equation, in a one-dimension context. The choice of the technique for discretizing is justified.

Finally, some key points around mesh definition are introduced. Attention is focused on structured, unstructured and hybrid grids discretization.

2.2 Discretization of partial differential equations

2.2.1 Mathematical analysis on a model problem

Formally, the Navier-Stokes equations represent evolutionary equations for conservative aerodynamic quantities. There exist nowadays several techniques for their discretization. The three classical approaches – finite element, finite volume, finite differences – are presented on a model problem. There are also new discretization techniques, out of purpose of this document. Among them, one can consider as an example the Discontinuous Galerkin discretization technique.

For the sake of clarity, we consider the linear equation for heat diffusion in a one dimension:

$$\partial_t T = \alpha \Delta T, \tag{2.1}$$

where T is the temperature, α the (constant) diffusion coefficient, and Δ is the symbol for the Laplacian. In 1D and if x denotes the discretization space, the Δ operator applied to a function f is written:

$$\Delta f = \frac{\partial^2 f}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial f}{\partial x}\right). \tag{2.2}$$

The model problem completeness is achieved by defining a segment on which the solution is searched, namely $[0, 1]$ here, and by providing boundary values. In this case, we consider Dirichlet boundary conditions: $T(x = 0) = T_0$ and $T(x = 1) = T_1$ independant of time t.

For the spatial discretization, the segment $[0, 1]$ is divided into N segments of equal length. This point enables simplifications in explanations and it is not a prerequisite in general. The $N + 1$ segment limits are denoted x_j and we have $x_j = j/N$ for $0 \leq i \leq N$.

For the temporal discretization, we assume that time t is in $[0, +\infty)$ and that $[0, +\infty)$ is divided in intervals of same duration, namely δt. Time instants are denoted $t^n = n(\delta t)$.

In the following, temperature T depends on space and time positions and let $T_j^n = T(t = n\delta t, x = x_j)$ be the temperature at node j and at time n. Three discretization techniques will be studied, namely

1. finite differences in section 2.2.2,
2. finite elements in section 2.2.3,
3. finite volumes in section 2.2.4.

2.2.2 Finite differences approach

The finite difference approach is based on Taylor expansions for the different terms of the equations.

Temporal derivative

Let n be the instant at which the temporal derivative is considered and let i be the spatial position. One has:

$$\left(\partial_t T\right)_j^n \simeq \frac{T(n\delta t + \delta t, x_j) - T(n\delta_t, x_j)}{\delta t} = \frac{T_j^{n+1} - T_j^n}{\delta t}. \tag{2.3}$$

Eq. 2.3 is a rewriting of the temporal derivative, issued from the following relations:

$$\begin{aligned} T_j^{n+1} &= T_j^n + \delta t \left(\frac{\partial T}{\partial t}\right)_j^n + o((\delta t)^2) \\ \frac{T_j^{n+1} - T_j^n}{\delta t} &= (\partial_t T)_j^n + o(\delta t). \end{aligned}$$

The term $o(\delta t)$ contains all terms associated with powers of δt greater (strictly) than one: the formula is said to be precise at order one in time.

Spatial derivative

Let's consider the spatial derivative at discrete time n and spatial position j. The spatial derivative term is discretize:

$$\Delta T_j^n \simeq \frac{T_{j-1}^n - 2T_j^n + T_{j+1}^n}{(\delta x)^2}, \tag{2.4}$$

which is called a second order centered formulation around T_j^n. Eq. 2.4 is easily justified; since

$$T_{j+1}^n = T_j^n + \delta x \left(\frac{\partial T}{\partial x}\right)_j^n + \frac{(\delta x)^2}{2}\left(\frac{\partial^2 T}{\partial x^2}\right)_j^n + \frac{(\delta x)^3}{6}\left(\frac{\partial^3 T}{\partial x^3}\right)_j^n + o((\delta x)^4)\,,$$

$$T_{j-1}^n = T_j^n - \delta x \left(\frac{\partial T}{\partial x}\right)_j^n + \frac{(\delta x)^2}{2}\left(\frac{\partial^2 T}{\partial x^2}\right)_j^n - \frac{(\delta x)^3}{6}\left(\frac{\partial^3 T}{\partial x^3}\right)_j^n + o((\delta x)^4),$$

it comes, by a summation of the latest two relations:

$$\begin{aligned} T_{j+1}^n + T_{j-1}^n &= 2T_j^n + (\delta x)^2 \left(\frac{\partial^2 T}{\partial x^2}\right)_j^n + o((\delta x)^4) \\ \frac{T_{j-1}^n - 2T_j^n + T_{j+1}^n}{(\delta x)^2} &= \left(\frac{\partial^2 T}{\partial x^2}\right)_j^n + o((\delta x)^2) \end{aligned}$$

The centered aspect of the discretization is justified by the use of same weights on the temperature at positions around the considered location for the discretization. The second order of precision is issued from the $o((\delta x)^2)$ term.

Final version

Replacing each term of Eq. 2.1 by its discretized counterpart, one obtains:

$$\frac{T_j^{n+1} - T_j^n}{\delta t} = \alpha \frac{T_{j-1}^n - 2T_j^n + T_{j+1}^n}{(\delta x)^2}. \tag{2.5}$$

Eq. 2.5 can finally be written as:

$$T_j^{n+1} = T_j^n + \frac{\alpha \delta t}{(\delta x)^2} \left(T_{j-1}^n - 2T_j^n + T_{j+1}^n\right). \tag{2.6}$$

and Eq. 2.6 means that the temperature at time $n+1$ is computed algebraically from data at time n: it is an **explicit time integration**.

The final equation needs some comments:

- In a steady-state solution context, the computation of numerical solutions of the heat equation or the Navier-Stokes equations is based on a discretization of the temporal derivative, leading to the use of a pseudo-time marching approach. It means that the solutions are computed iteratively at higher and higher times until a stationary solution is found. In this case, the temporal derivative can be neglected and the stationary solution is finally obtained. In practice, the temporal derivative never disappears since it is not possible to do an infinite number of time steps on a computer. The stop criterium is given by the residual, which measures the differences between the solution at time $n+1$ and the one at time n.

- One **never solves the original continuous equations with the finite differences approach**. Never forget that each operator (temporal derivative and Laplacian) are computed from a Taylor expansion for which high order terms are simply dropped. For a steady solution, it means that the convergence to the solution of the continuous problem can only be tackled at the limit, when the spatial step δx goes to 0.

- The finite differences approach needs the solutions to be regular to define successive derivatives of the solution for the Taylor expansion. This hypothesis is valid for the heat equation but can be false for the Navier-Stokes equations, when the solution contains a discontinuity such as a shock (Fig. 2.1)

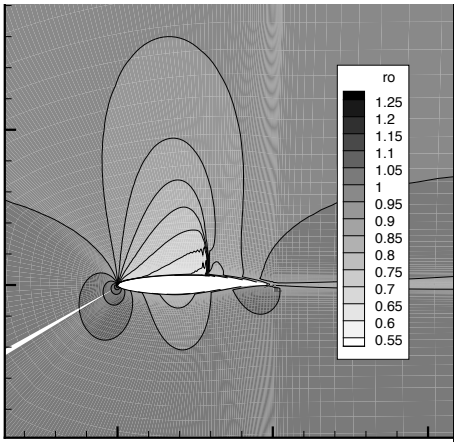

Figure 2.1: *Transonic flow around the RAE2822 profile: a shock appears on the upper side. It is characterized by a discontinuity of the density.*

- The parameter $\alpha\delta t/(\delta x)^2$ influences the update of the solution at time $n+1$. This ratio has a strong influence in numerical simulation since it links time and space steps. The stability[1] of computations is driven by this kind of ratios.

Remark 2.2.1 *For industrial flows that may have shocks, the finite differences approach is not chosen as a candidate for discretizing the continuous equations for fluid dynamics.*

2.2.3 Finite element approach

The finite element approach is mathematically justified by the distribution theory associated with Sobolev spaces. In this document, none of these notions will be recalled.

Principle of finite elements

The finite elements approach is built on a variational formulation associated with the weak form of the equations. The weak form of the continuous equations is defined once the "good" Sobolev space $\mathcal{S}$ is defined and once distribution function are associated with $\mathcal{S}$. In this section, the weak equation corresponding to Eq. 2.1 is derived. The boundary conditions are supposed to be zero values for the temperature[2]: we solve the homogeneous problem. Let $\Omega =]0,1[$ the open space on which the solution is searched.

Let v be a function defined on the "good" Sobolev space (here $\mathcal{S} = H_0^1(\Omega)$). The transformation of Eq. 2.1 is done by first multiplying Eq. 2.1 by v and then by integrating on Ω:

$$\int_\Omega \frac{\partial T(x,t)}{\partial t} v(x)dx = \alpha \int_\Omega \Delta T(x,t)v(x)dx, \quad \forall v \in H_0^1(\Omega). \tag{2.7}$$

[1] not defined here!

[2] A linear change of variables is used to transform the initial problem in the homogeneous one.

Let's study $\int_\Omega \frac{\partial T(x,t)}{\partial t} v(x)dx$. Classical permutation rules for derivation and integration are necessary to obtain:

$$\int_\Omega \frac{\partial T(x,t)}{\partial t} v(x)dx = \frac{d}{dt}\int_\Omega T(x,t)v(x)dx, \quad \forall v \in H_0^1(\Omega). \tag{2.8}$$

Now, consider $\int_\Omega \Delta T(x,t)v(x)dx$. Using Green theorem (in France, it is also called integration by parts), one has:

$$\begin{aligned} \int_\Omega \Delta T(x,t)v(x)dx &= \int_{\partial\Omega} \nabla T(x,t)v\vec{n}ds - \int_\Omega \nabla T(x,t)\nabla v dx \\ &= 0 - \int_\Omega \nabla T(x,t)\nabla v dx \end{aligned} \tag{2.9}$$

since $v = 0$ on $\partial\Omega$. Blending Eq. 2.7, Eq. 2.8 and Eq. 2.9, one finally obtains:

$$\frac{d}{dt}\int_\Omega T(x,t)v(x)dx + \alpha\int_\Omega \nabla T(x,t)\nabla v(x)dx = 0 \quad \forall v \in H_0^1(\Omega). \tag{2.10}$$

Eq. 2.10 is used to define a scalar product in $L^2(\Omega)$:

$$(v,w)_{L^2(\Omega)} = \int_\Omega v(x)w(x)dx \quad \forall v,w \in H_0^1(\Omega)\,, \tag{2.11}$$

and a symmetric bilinear form:

$$a(v,w) = \alpha\int_\Omega \nabla v(x)\nabla w(x)dx \quad \forall v,w \in H_0^1(\Omega)\,. \tag{2.12}$$

The final variational formulation is obtained with Eq. 2.11 and Eq. 2.12:

$$\frac{d}{dt}(T(t),v) + a(T(t),v) = 0 \quad \forall v \in H_0^1(\Omega). \tag{2.13}$$

From now on, it is the work for mathematicians! They will explain, with the "good" theorem, that Eq. 2.13 has almost a solution, that the solution is bounded and has a certain regularity (is the solution continuous?). The mathematical analysis is out of purpose.

The establishment of finite elements is based on the definition of the v test function space. For the numerical solution, one generally chooses v to be polynomial of degree p and the solution u of the problem Eq. 2.13, which one is looking for, is assumed to be also a polynomial of degree p. The extension of the finite element for the compressible Navier-Stokes equations is possible. However, some justifications of the theory remain not demonstrated...

In practice, the finite element methods is applied in structural mechanics, and perhaps less in computational Fluid Dynamics. The CERFACS and IFP[3] code AVBP for Large Eddy Simulation is built on a mixed approach based on both finite element and finite volume formalisms.

2.2.4 Finite volume approach

The finite volume approach has its origins in the building process for the equations (physical point of view). The treatment is more or less the same for the Navier-Stokes or heat transfer equations.

[3]Institut Français du Pétrole

From physics to finite volume

The Navier-Stokes equations are built following an integral formalism (Chapter 1) for guarantying the conservation. It is this conservation principle that characterizes the finite volume approach. The underlying idea is to solve the integral version of the equations.

Principle

For the heat equation Eq. 2.1, it consists in using a space decomposition as for the finite differences approach. The equations are then integrated on each of the segments. One transforms Eq. 2.1 in :

$$\frac{d}{dt}\int_{x_j}^{x_{j+1}} T dx = \alpha \int_{x_j}^{x_{j+1}} \Delta T dx \,. \tag{2.14}$$

Applying Green's theorem on the right hand side of Eq. 2.14, one has:

$$\frac{d}{dt}\int_{x_j}^{x_{j+1}} T dx = \alpha((\nabla T)_{x_{j+1}} - (\nabla T)_{x_j}) \,. \tag{2.15}$$

Eq. 2.15 is a simple form of Green theorem since the analysis is done in one dimension. In a more general framework, if Ω is a control volume in dimension two or three, if $\partial\Omega$ denotes the boundary, if $\vec{n}$ represents the outward unit normal vector, Green formula leads to:

$$\frac{d}{dt}\int_{\Omega} T d\omega = \alpha \int_{\partial\Omega} \nabla T \vec{n} ds \,. \tag{2.16}$$

Eq. 2.16 has two terms: one from the temporal derivative an one boundary integral which will be called **interface flux** in the following.

Flux conservation

Suppose that Ω is decomposed in N_C volumes $(\Omega_i)_{1\leq i\leq N_C}$ such that the intersection between two volumes Ω_i and Ω_j is $\partial\Omega_i \cap \partial\Omega_j$ or is empty. We call mesh the space composed of all volumes $(\Omega_i)_{1\leq i\leq N_C}$ that cover Ω.

Remark 2.2.2 *The most important property of the finite volume approach:*
If the intersection between Ω_i and Ω_j is a mesh face (not an empty space), then the outward flux for Ω_i on $\partial\Omega_i \cap \partial\Omega_j$ is the inward flux for Ω_j on $\partial\Omega_i \cap \partial\Omega_j$.

Without loss of generality, we will show this conservation process with $\Omega = \Omega_1 \cup \Omega_2$ such as shown in Fig. 2.2 and let $C = \Omega_1 \cap \Omega_2$ be the intersection face. The boundary of Ω_1 without C is denoted $\overline{\partial\Omega_1}$ and the one of Ω_2 without C is $\overline{\partial\Omega_2}$. Eq. 2.16 is true on Ω, on Ω_1 and on Ω_2. Since $\Omega = \Omega_1 \cup \Omega_2$, it is clear that:

$$\alpha \int_{\partial\Omega} \nabla T \vec{n} ds = \frac{d}{dt}\int_{\Omega} T d\omega = \frac{d}{dt}\int_{\Omega_1} T d\omega + \frac{d}{dt}\int_{\Omega_2} T d\omega.$$

In the same way, one obtains from Eq. 2.16:

$$\frac{d}{dt}\int_{\Omega_1} T d\omega = \alpha \int_{\partial\Omega_1} \nabla T \vec{n} ds \text{ and } \frac{d}{dt}\int_{\Omega_2} T d\omega = \alpha \int_{\partial\Omega_2} \nabla T \vec{n} ds.$$

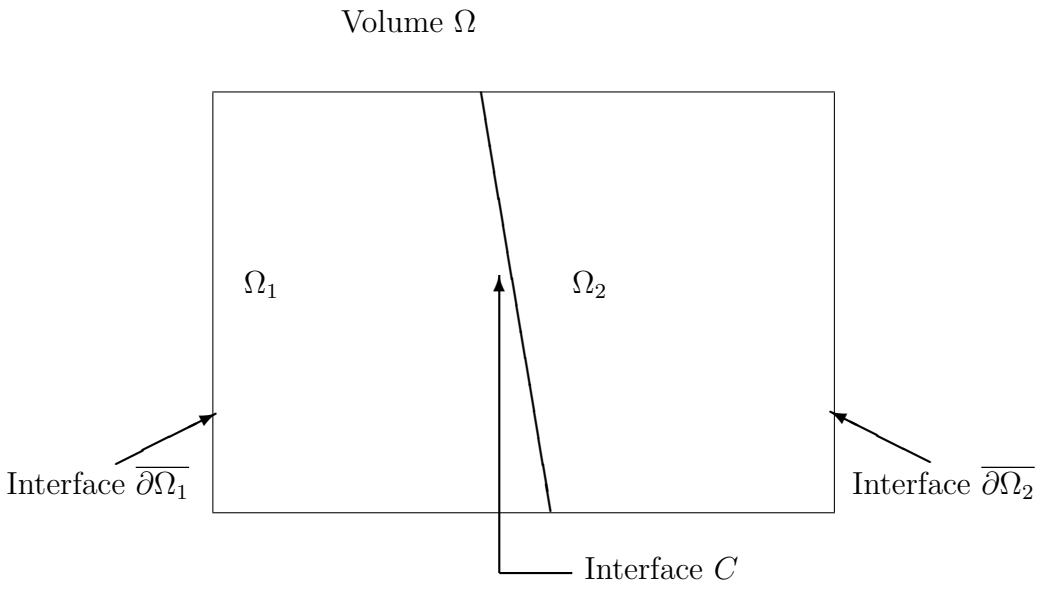

Figure 2.2: *Decomposition of volume* Ω *in* Ω_1 *and* Ω_2 *with a non empty intersection.*

Therefore:

$$\int_{\partial\Omega} \nabla T \vec{n} ds - \int_{\partial\Omega_1} \nabla T \vec{n} ds - \int_{\partial\Omega_2} \nabla T \vec{n} ds = 0. \tag{2.17}$$

Due to linear relations with integration, the following relations are true:

$$\int_{\partial\Omega} \nabla T \vec{n} ds = \int_{\overline{\partial\Omega_1}} \nabla T \vec{n} ds + \int_{\overline{\partial\Omega_2}} \nabla T \vec{n} ds, \tag{2.18}$$

$$\int_{\partial\Omega_1} \nabla T \vec{n} ds = \int_{\overline{\partial\Omega_1}} \nabla T \vec{n} ds + \int_{C} \nabla T \vec{n} ds, \tag{2.19}$$

$$\int_{\partial\Omega_2} \nabla T \vec{n} ds = \int_{\overline{\partial\Omega_2}} \nabla T \vec{n} ds + \int_{C} \nabla T \vec{n} ds. \tag{2.20}$$

Introducing Eq. 2.18, Eq. 2.19 and Eq. 2.20 in Eq. 2.17, it comes:

$$\left(\int_C \alpha \nabla T \vec{n} ds\right)_{\text{side of } \Omega_1} + \left(\int_C \alpha \nabla T \vec{n} ds\right)_{\text{side of } \Omega_2} = 0 \tag{2.21}$$

In Eq. 2.21, normal unit vectors are outward and therefore, the unit outward vector $\vec{n_1}$ on C for volume Ω_1 is entering in Ω_2. Denoting $\vec{n_1}$ the unit normal on C oriented from Ω_1 to Ω_2, one has:

$$\left(\int_C \alpha (\nabla T) \vec{n_1} ds\right)_{\text{side of } \Omega_1} = \left(\int_C \alpha (\nabla T) \vec{n_1} ds\right)_{\text{side of } \Omega_2}, \tag{2.22}$$

which means that the outward flux for Ω_1 through C is entering in Ω_2 through C.

A conclusion on finite volumes

Several notions for finite volumes appear in Eq. 2.16:

1. the notion of control volume, the famous finite volume,
2. the principle of conservation: through an interface denoted ij between two volumes referenced i and j, the outward flux for i is entering j.
3. the notion of numerical scheme, which is the technique to compute unknown quantities such as gradients at interface...
4. The notion of mean value T_Ω of T over a control volume, and precisely

$$Vol(\Omega)T_\Omega = \int_\Omega T d\omega \text{ and for a one-dimension space } \delta x T_{[x_j, x_{j+1}]} = \int_{x_j}^{x_{j+1}} T dx,$$

   If it is needed (by numerical discretization schemes), the mean value is assumed stored in the cell center.
5. the notion of metrics, which represent all tables to access geometrical quantities such as volume, surface normal vectors, volume centers...

Now, we need to introduce and define the mesh. A mesh is a division of space on non overlapping elements on which the continuous equations are discretized. There exists two kinds of mesh and they are introduced in section 2.3.

2.3 Unstructured and structured meshes

Computational Fluid Dynamics numerical tools are divided into two main branches, following the kind of mesh used for the simulations, either structured or unstructured solvers. Both these meshes have pro and cons.

2.3.1 Structured mesh

A structured mesh is a mesh for which there exists one privileged direction per space dimension, which enables to associate mesh nodes to a couple of integers (i, j) in dimension two, or a triplet of integers (i, j, k) in three-dimension space.

Remark 2.3.1 *In the following, all considerations will be presented in a two-dimension framework and their extensions to three dimensions will be obvious.*

Definition of a block in a mesh

A structured mesh is composed of several blocks which are defined with:

- two integers $im + 1$ and $jm + 1$,
- $(im + 1) \times (jm + 1)$ mesh coordinates following space directions x and y,

- an implicit definition of the $im \times jm$ control volumes. A volume whose reference is (i, j) uses mesh nodes referenced (i, j), $(i+1, j)$, $(i, j+1)$, $(i+1, j+1)$ as boundary points in directions i and j. Directions i and j are in general not aligned with the space directions x and y.

Fig. 2.3 shows a schematic view of a structured mesh block of size $(5, 6)$, *I.E.* with $im = 4$, $jm = 5$ and therefore with 20 control volumes and 30 mesh nodes.

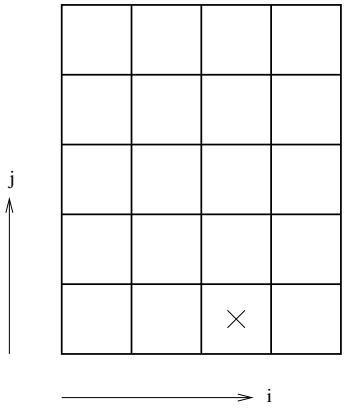

Figure 2.3: *Example of a structured mesh with* $im = 4$ *and* $jm = 5$. *The mesh cell with a cross is referenced by* $(3, 1)$.

Towards a multi blocs approach

It is generally impossible to discretize a Computer-Aided Design (CAD) with a single block. As an example, one can consider the case of a T pipe junction for which it is impossible to define properly a mesh with a single block Fig. 2.4. In this case, one needs at least two blocks.

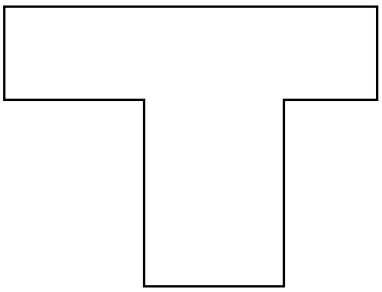

Figure 2.4: *Example of a T-pipe junction configuration for which it is not possible to define a mesh composed of one block.*

Moreover, the need of a multi-block approach coupled with the finite volume approach leads the connectivity between blocks to have one-to-one abutting nodes. Therefore, mesh lines of a given block go across the block interface to the neighboring block. An example is given Fig. 2.5: the number of lines in i for block 1 fixes the number of lines in j for block 2.

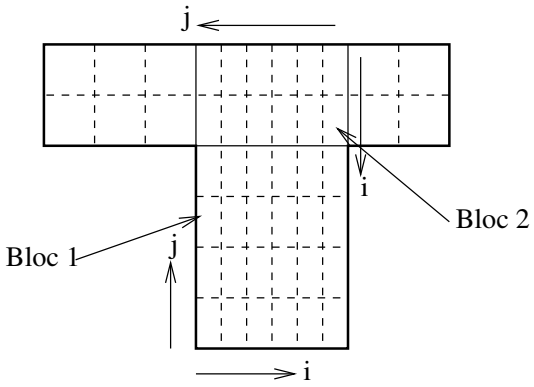

Figure 2.5: *Interface connectivity with one-to-one abutting nodes. Block 1 has* $(6, 4)$ *nodes while block 2 is composed of* $(2, 6)$ *nodes. The line number in i for block 1 is the number of lines in j for block 2.*

Basic mesh topologies

Structured multi-blocks meshes are composed of elementary mesh topologies, namely H, O and C-grid topologies. We will see that naming and mesh shape are linked.

• H-grid topology.
It is the simpler topology. It has been shown at the first time in Fig. 2.3.

• O-grid topology.
As suggested by its name, the O grid shape is used to mesh a circular element. In fact, covering a disk with a single block is possible with a H-grid (Fig. 2.6) but some cells are flattened, with a shape generally not adapted to numerical schemes (weak precision, lack of robustness...). The solution consists in splitting the mesh in 5 blocks following the representation on Fig. 2.6.

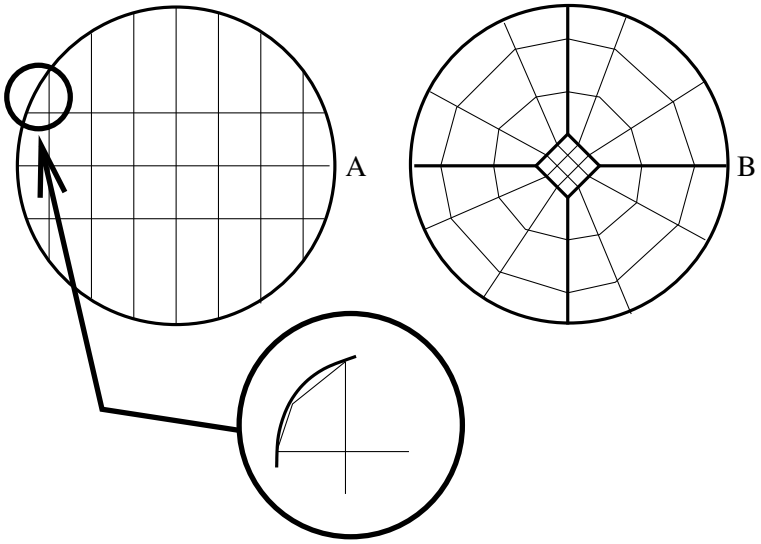

Figure 2.6: *Example of a mesh on a disk with a single block (A) and with a multi-block approach (B). For (A), a focus shows the flattening of mesh cells. One can also remark generally that the mesh is not completely defined on the CAD: the distance between mesh and CAD depends on mesh refinement.*

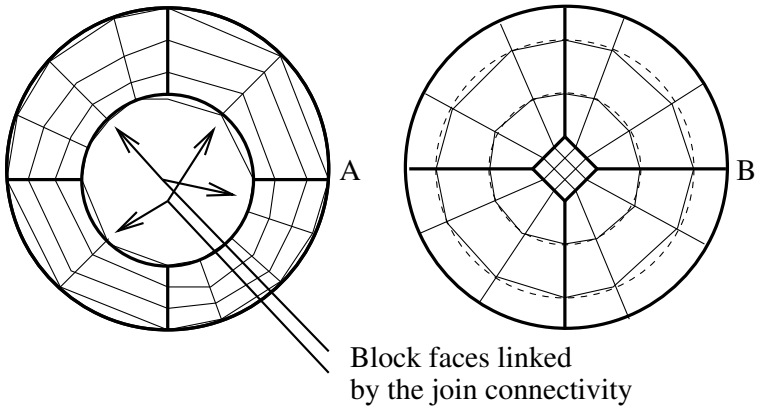

Figure 2.7: *Example of an incomplete O-grid (left) and of a complete O-grid (right).*

Even if the classical form of a 2D O-grid contains 5 blocks, one can find incomplete mesh shapes for which finding the true mesh topology is not so obvious. This typical situation is encountered when a block inside the topology must not be discretized (Fig. 2.7).

• C-grid topology.
As suggested by its naming, the C-grid topology is simply half an O-grid one. It is the basic topology to mesh half a disk and this technique leads to 4 mesh blocks, as shown on Fig. 2.8.

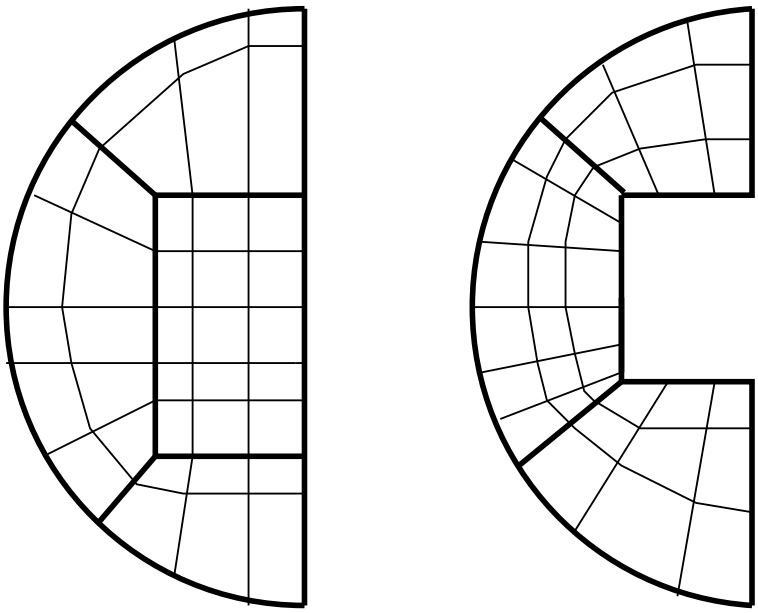

Figure 2.8: *Examples of complete (left) and incomplete (right) C-grids.*

A mesh is finally built by blending the three basic topologies. As an example, consider a two-dimension wing with a planar thick trailing edge: a C-H topology is chosen for the mesh (Fig. 2.9), while it will be a O-grid for a rounded trailing edge (Fig. 2.10). For a 2D wing with a sharp trailing edge, an incomplete C-grid topology is suitable: the block in the wing wake is deleted and block corners are joined (Fig. 2.11).

More comments

Structured CFD codes are designed specifically for structured meshes. One of the main property of the mesh is an easy way to locate volumes, interfaces, surface vectors... with two integers (i, j) for each block. One says that the data addressing is direct. With this technique, data are stored contiguously in memory enabling a quick memory access to the data. Moreover, the Navier-Stokes solutions are not isotropic, especially in the boundary layer. In this case, structured grids are adapted to capture the physics, defining privileged mesh directions to obtain very accurate solutions. In Fig. 2.12 is shown an example of structured mesh around a 3D wing.

The structured multi-blocks technique has some drawbacks. The main drawback is the

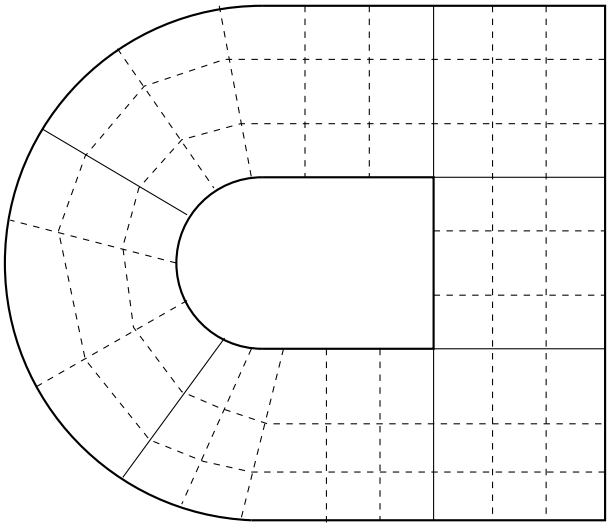

Figure 2.9: *Example of mesh for a 2D wing with a planar thick trailing edge.*

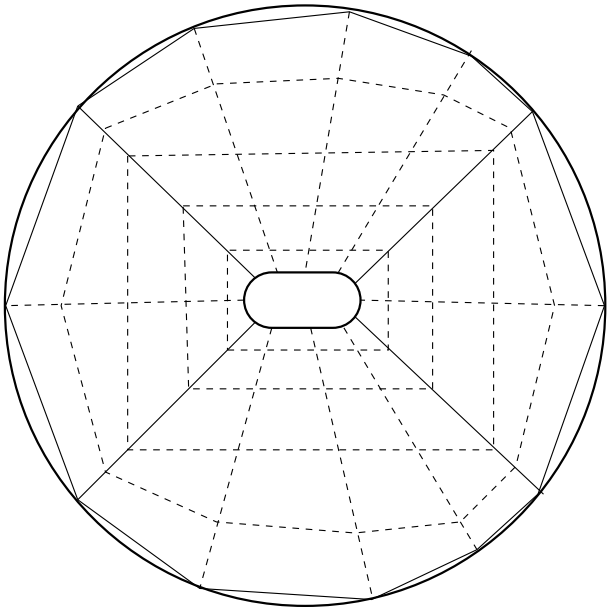

Figure 2.10: *Example of mesh for a 2D wing with a rounded thick trailing edge.*

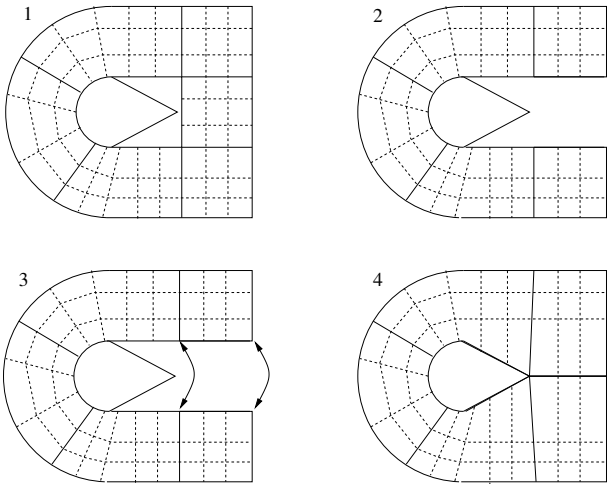

Figure 2.11: *Building process for a C-grid on a 2D wing with a sharp trailing edge. (1) C-grid mesh, (2) delete a block in the wing wake, (3) join of block corners and (4) final mesh.*

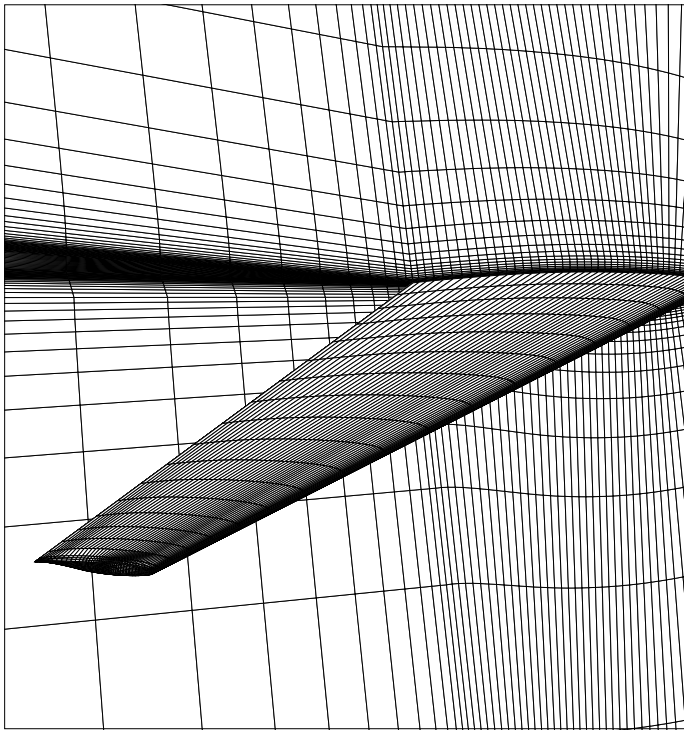

Figure 2.12: *Example of a three dimensional structured grid.*

need to propagate mesh lines up to the end of the domain. For a 2D turbulent flow on a wing with a sharp trailing edge, the mesh lines issued from the boundary layer discretization are propagated in the whole wake. An alternative consists in authorizing non conformal blocks interfaces with non 1-to-1 mesh connectivity. This technique is available in *elsA* (ensemble logiciel pour la simulation en Aérodynamique) [1], a structured CFD software developed by ONERA and CERFACS and used in Airbus, Safran (SNECMA, Turbomeca) or EDF.

2.3.2 Unstructured meshes

An unstructured mesh can be defined by opposition with a structured mesh. Even if this definition is true, it hides most of the properties of unstructured meshes.

An unstructured mesh is a mesh for which the data structure contains (at least) the following elements:

- The number of nodes N_s and the coordinates (x_i, y_i, z_i) for each node i,
- The number of mesh volumes N_V and for each cell *icell*, the list of k nodes which the element is based on: this is called the mesh volume connectivity.

It is clear that addressing the data is indirect. This is particularly true for mesh nodes. For a cell *icell*, the connectivity table gives the list of mesh nodes which define the element *icell*. A second step is necessary to have access to the mesh nodes coordinates stored in a table of size N_s.

In opposition with structured meshes which are difficult to design on complex CAD and which need highly qualified engineers, unstructured meshes are generated quickly by commercial softwares. Fig. 2.13 represents a mesh composed of quadrangles and triangles. In the unstructured community, a mesh composed of several element shapes is called a hybrid mesh.

Element shapes for an unstructured mesh

An unstructured mesh can be composed of any kind of polygon (for dimension 2) or polyhedra (for dimension three). In practice, there are a few codes able to treat general polyhedra. The basis element shapes accepted by all unstructured CFD codes are typically triangles and tetrahedra. Most of the CFD codes can treat hybrid meshes composed of

- triangles et quadrangles,
- tetrahedra, prisms, pyramids and hexahedra.

Cost of indirections

Compared with structured grids, the use of unstructured meshes leads to an increase of the CPU cost due to indirections. For a structured mesh, accessing the mesh nodes that limit a hexahedra (i, j, k) is obvious: the eight nodes are referenced by (i, j, k), $(i+1, j, k)$, $(i, j+1, k)$, $(i+1, j+1, k)$, $(i, j, k+1)$, $(i+1, j, k+1)$, $(i, j+1, k+1)$ et $(i+1, j+1, k+1)$. For an unstructured mesh, one needs to use the mesh connectivity table to access the node numbers. Then, one needs to find these nodes in the list of mesh nodes coordinates.

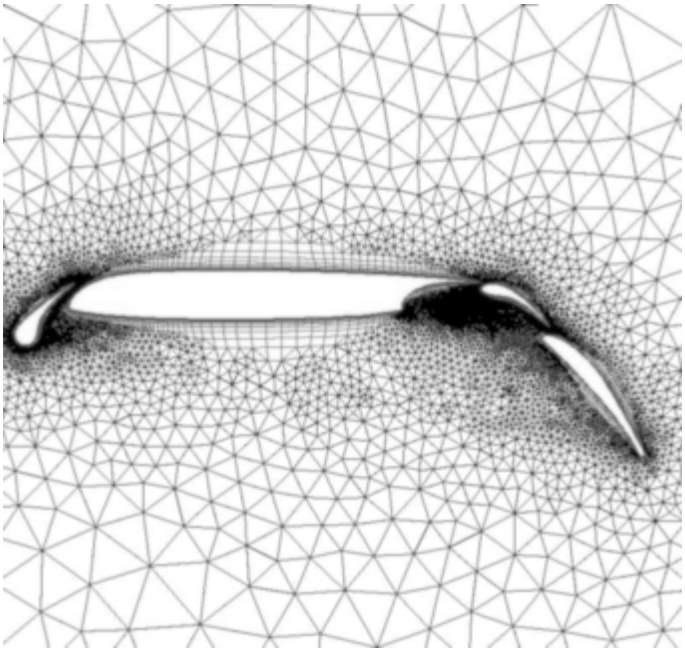

Figure 2.13: *Example of a hybrid unstructured mesh composed of quadrangles and triangles.*

2.3.3 Consequences

We have now defined all the formalism needed for the discretization of the Navier-Stokes equations. Chapter 3 is devoted to the analysis of mesh and solution storage.

Chapter 3

Formulation and location of data

This chapter is devoted to explanations in relation with data storage for finite volumes CFD codes. In particular, we will see that *elsA* and AVBP, two CFD codes which share a finite volume approach, use data not located in the same place.

3.1 Data storage location for structured grids

For a structured grid, there are two possible locations to store the data:

1. the cell centers, as in *elsA*,
2. the mesh nodes, as in NTMIX[1].

3.1.1 The cell center approach

The cell center approach is the widespread technique for the discretization of the Navier-Stokes equations on structured grids. This choice is motivated by the simple definition of elements faces (mesh faces) on which interface fluxes are computed. Addressing data and mesh elements is done in a similar way since there are simple relations between mesh node number and mesh element / cell center solution number.

3.1.2 The node center approach

In this case, data are stored at the mesh nodes. This approach is generally chosen for Direct Numerical Simulation softwares which are used to compute the whole turbulence spectrum. DNS softwares need high order discretization schemes and are not generally applied at high mach number. With both these considerations, one understands that a good solution consists in using finite difference schemes: the error can be measured with the rest of the Taylor expansion.

The node center approach brings some difficulty for general meshes composed of several basic configurations (O and C). In particular, some nodes of a O grid are shared by three blocks (Fig. 3.1), which leads to a complex choice for the direction needed by the finite differences along block interfaces.

[1] Direct Numerical Simulation software developed by CERFACS

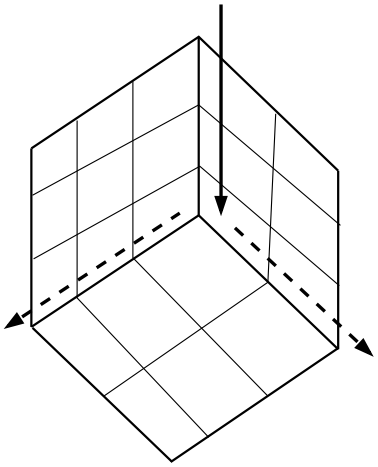

Figure 3.1: *Example of a 3 blocks mesh junction. For the discretization represented by the arrow, one can not choose complementary information from one of the dashed arrows.*

3.2 Data storage location for unstructured grids

In the case of unstructured grid, data are classically stored either in the cell center or at the mesh nodes. As an example, the cell-centered framework has been chosen for unstructured and hybrid structured / unstructured capabilities in the *elsA* CFD code. This choice was essentially motivated by compatibility with structured features of the initial version of the solver. Even if the cell center choice is almost classical and with a motivation comparable with the one for structured meshes, the choice to store data at the mesh nodes is not anymore driven by Taylor expansion considerations. The choice to store data at mesh nodes has been chosen for the AVBP code used by the combustion group at the CERFACS CFD team.

Let us consider a mesh composed of triangles or tetrahedra. Assume that the degrees of freedom for the discretization are associated with mesh nodes. It is possible to derive through a $\mathcal{P}^1$ Finite Element analysis[2] a weak formulation for the Navier-Stokes equations. The use of a Finite Element framework offers the possibility to use a lot of mathematical results. The main theoretical results are summarized in [5] and the most important one is recalled:

On triangles or tetrahedra, the $\mathcal{P}^1$ Finite Element approach and the Finite Volume approach on dual cells are equivalent, provided the fact that the discretization of the time derivative accounts the mass-lumping matrix.

The previous sentence needs some comments. First, each dual cell is built around the corresponding mesh nodes with a simple process. In two dimensions, the volume around a mesh node is limited by 'facets' linking the midpoints of the edges in the primal mesh to the barycentres of the elements obtained by arithmetic averaging of the nodal coordinates. In three dimensions, the dual volume is delimited by quadrangular facets between the edge midpoints, the face barycentres and the element barycentres (Fig. 3.2). This definition is clearly a plus for numerical schemes since the dual cells is in the middle of the mesh edge.

[2]The weak formulation is obtained with test functions defined as polynomials of degree 1. Such polynomials are defined with uniqueness on triangles and tetrahedra with data stored at mesh nodes.

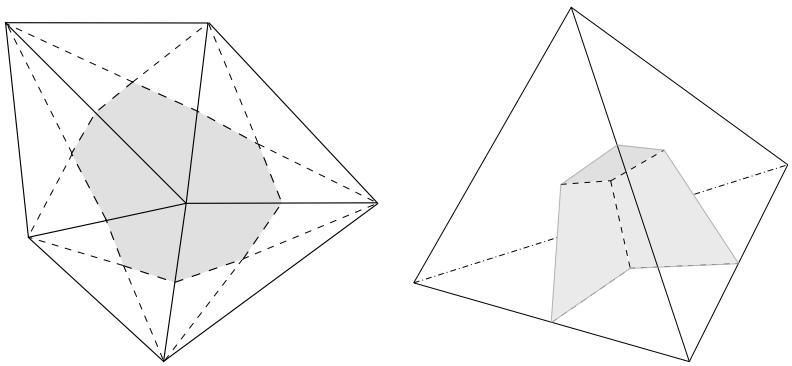

Figure 3.2: *Definition of dual control volume for a mesh node shared by five triangles (left) and dual volume boundary inside a tetrahedron (right).*

Finally, a mass-lumping matrix must be introduced. After applying the $\mathcal{P}^1$ Finite Element approach, one looks for a solution also defined in the $\mathcal{P}^1$ Finite Element space. As a consequence, it appears in the discretization of the time step some integrals of the product of test functions on triangles / tetrahedra. The test function integrals compose the mass-lumping matrix, which is known to be invertible.

The main advantage of this approach is to offer a mathematical background for the construction of gradients: gradients are computed using a classical finite element approach. A lot of mathematical papers have studied the equivalence of both finite-volume and finite element approaches leading to a high level of confidence in the numerical treatment. Moreover, even if upwind convection schemes can be implemented easily in a multi-element context, several authors have proposed diffusion schemes for multi-element approaches and most of the finite-volume / finite-element codes use the formalism of multi-element approach.

Chapter 4

Some extensions to simplify the mesh generation process

We have seen in Chapter 2 that there are mainly two kinds of mesh, according to the mesh generation process. On one hand, unstructured grids are generated very quickly but because of a low level of interaction with the mesh user, they are generally associated with mesh lines not aligned with the flow features. On the other hand, structured meshes are conceived to account for anisotropy of the flow, such as inside a boundary layer for instance. In the latter case, due to the low number of mesh basic element shapes (H, C and O grid shapes), the mesh generation can require a lot of time, typically from one week to several months, which is mostly consumed by human being.

We will see in the following that for structured meshes, there exist some techniques dedicated to simplify the mesh generation process or to decrease the time for the generation process.

4.1 Limiting mesh nodes in C grid wakes

As seen in Chapter 2, C-grid shape is used to mesh (around) a wing with a sharp leading edge. When the mesh is refined in the direction normal to the wall, the number of mesh lines in the wing wake is overestimated to capture the aerodynamic quantities needed for dimensioning during the industrial process (lift, drag coefficients...): the mesh is too much refined in the wake.

Roughly speaking, the principle to diminish the number of mesh lines in the wake is to authorize mesh lines not to cross a block interface. In practice, this is done through the implementation of two kinds of techniques, namely the near-match join and the non matching interface. Both approaches will be described in the following sections.

Remark 4.1.1 *The implementation of near-matching and non-matching mesh interfaces is quite easy in a cell-centred formalism but the situation is much more complex for any other kinds of discretization.*

4.1.1 Near-matching mesh interfaces

The first one is called near-matching block interface in the *elsA* terminology (Fig. 4.1). It consists in defining new mesh lines located each one over x nodes interfaces. Generally, x is issued from possibilities of mesh sequencing or multigrid and is therefore defined as 2^i with i integer, but this choice is not a limitation in general. In the implementation, near-matching

interface is treated as a classical one-to-one block interface and there is a simple link between faces on the refined side and those on the less refined one.

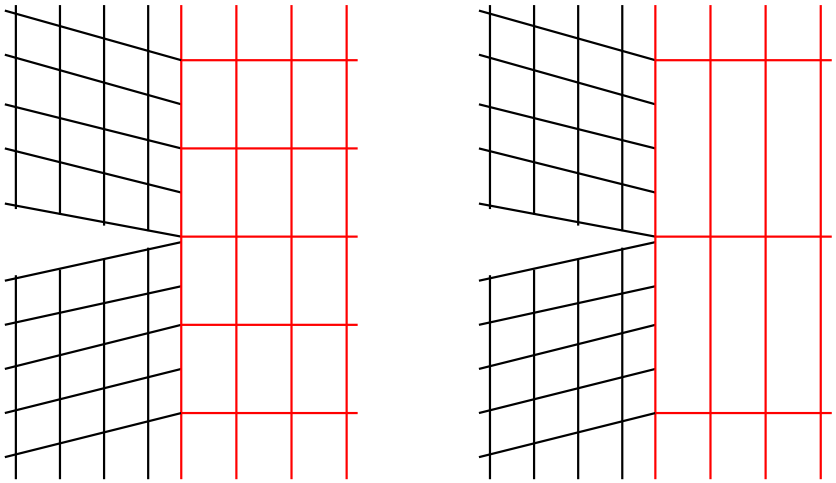

Figure 4.1: *Example of near-matching join interfaces: case of "1 over 2" connectivity (left) and "1 over 4" connectivity (right).*

4.1.2 Non-matching mesh interfaces

The second technique is a generalization of the near-matching interface. The principle is to define a new CAD interface on which blocks limits are projected. As a consequence, mesh nodes are generally not located at the same place on both sides of the interface. As an example, one can consider the case suggested on Fig. 4.2.

Due to the possible definition of CAD interface represented by different blocks interfaces, non-matching interfaces are generally associated with global borders definition. A global border is simply defined as a global container for one or more than one window and a window is a subset or the entire external face of an hexahedral block. A window is typically defined by $(i_1, i_2, j_1, j_2, k_1, k_2)$ and either $i_1 = i_2 = i_{max}$ or i_{min}, or $j_1 = j_2 = j_{max}$ or j_{min}, or $k_1 = k_2 = k_{max}$ or k_{min}. In Fig. 4.2, there is a global border composed of two windows for the upper blocks and a global border composed of a single window for the bottom block.

The numerical treatment of global border is a "tricky" point. In the finite volume framework associated with a cell center formulation, one has to handle boundary integrals. At this level, one has to analyse the surface for integration and the term to integrate. The term to integrate is easily computed using classical schemes, once surface data are provided. The question lying on geometric quantities concerns the definition of the integral. The data integrated on the "less refined" side must account for data located in the refined block on the opposite side. This is possible through the use of a geometric algorithm to compute the general connectivity between both sides of the interface (connectivity between two global borders). The uniqueness of the definition of the intersection surfaces between mesh interfaces is a prerequisite to the numerical treatment. In *elsA*, all geometric quantities are computed by the grid blocks on their own, which means that geometric quantities are not exchanged at the beginning of the time loop of the solver.

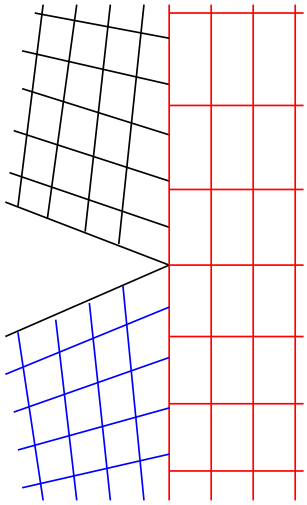

Figure 4.2: *Example of a non-matching join interfaces between two blocks (up) and one block (down). One can remark that only one mesh line is continuous across the interface and this mesh line is not the separation line between the upper blocks.*

The numerical treatment is conservative only for planar interfaces, since for non planar interfaces the global surface size is larger for the refined side than for the less refined one, as shown on Fig. 4.3. For a conservative treatment, it is necessary to change locally the definition of the metrics (surface area, surface normal) according to the refined block in order to share locally and globally the same surface normals and area for the intersection facets. In practice, this kind of modification is not implemented since the lack of conservation is generally small and does not limit convergence.

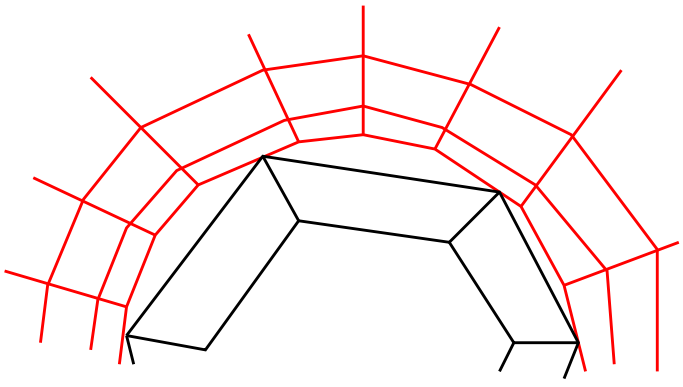

Figure 4.3: *Importance of planar interfaces on geometric quantities such as surface normals.*

The non-matching interface is also used to define global boundaries in movement. As an example, one can consider a 2D airfoil. The airfoil is surrounded by a local circle in movement of rotation and the rest of the computational domain is meshed independently (Fig. 4.4): this is the principle of the sliding mesh.

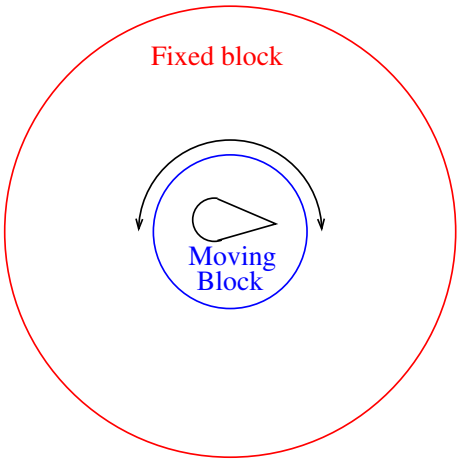

Figure 4.4: *Example of a non-matching interface for a sliding mesh (block blue) inside the outer block (red).*

In practice, the fact of splitting the computational domain into several parts, to mesh them independently and to merge the obtained grids through the definition of global borders enables the process to be done in parallel, each human being working on a part of the domain. This procedure is applied nowadays for more than 80% of the meshes at Airbus France.

4.2 Handling complex CAD with the chimera approach

The lack of the structured approach is its ability to treat highly complex CAD or to deal rapidly and easily with small geometry modifications. Consider the mesh around an aircraft at landing, assuming that one needs to add spoilers needed for brake. Without any automatic procedure, one will need to modify the blocking and the mesh of the aircraft. This procedure can be more or less seen as a redefinition of the blocking since the blocking will be locally modified (around spoilers) and due to one-to-one connectivity on the wing, mesh lines will be propagated in the whole domain. For industrial application, such a process can not be applied routinely.

The solution consists in meshing independently the aircraft and the spoilers. Then, one adds spoiler meshes inside the aircraft mesh and the code must understand that the spoiler has been added. It it the principle of the chimera approach. At the interface between the merged meshes, the solver needs to make interpolations between the background grid and the others.

It is not numerically interesting to have interpolation at the limits of the spoilers mesh since a large region is computed twice. Some algorithms to define the computational mask (implicit hole cutting...) have been developed and implemented in *elsA* to define added zones as close as possible to the added geometry (Fig. 4.5). The chimera approach has been used at Airbus France since 2005.

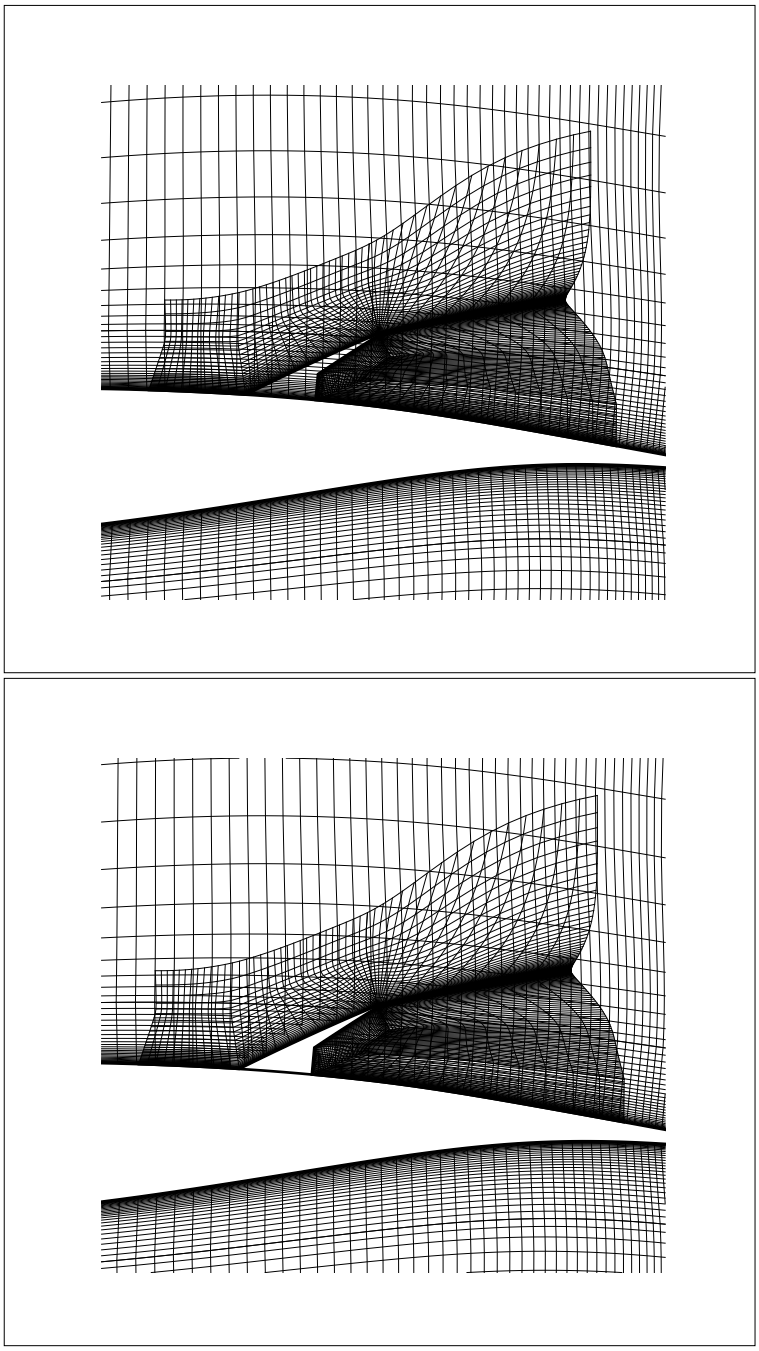

Figure 4.5: *Mesh around a 2D wing completed with a spoiler. Assembling of grids is shown on the upper figure. The meshes restricted to the computed volumes are shown at the bottom: the computed volumes are defined according to the masking technique considered for the computation. These images are extracted from F. Blanc, Méthodes numériques pour l'aéroélasticité des surfaces de contrôle, PhD Thesis, 2009.*

4.3 Prisms in structured grids

We have seen the interest of the O-grid approach for meshing a circle or a cylinder. The O-grid form is complicated to generate and to discretize properly, essentially due to the impact of the discretization of the internal grid on the surrounding ones. The question concerns (in 2D) the definition of a proper decomposition of 2π rad in three parts (it is the corner of the internal grid of the O-grid approach). In 3D, the situation is of course more complex...

The alternative to avoid such a drawback consists in implementing prisms in structured grids. In this case, one builds prisms using radii of the circle, as shown on Fig. 4.6. In practice, the prisms are seen as hexahedra with two pairs of degenerated nodes and the CFD code must have a metric compatible with such a kind of mesh (zero surface and normal vector, correct volume...). *elsA* offers the possibility to treat such a configuration. Practical situations for which prisms are generally encountered concern inflow and outflow of nacelles or the boundary condition of actuator disk (used to mimic a propeller boundary condition).

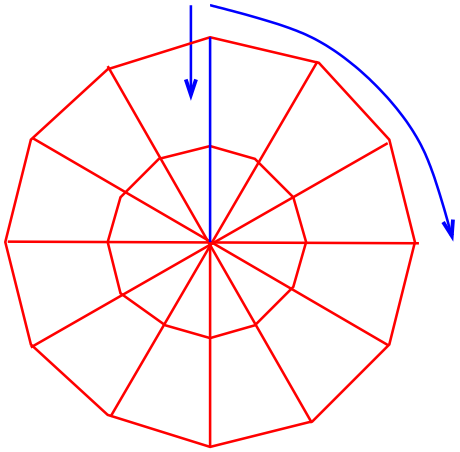

Figure 4.6: *Example of a structured block composed of prisms (degenerated hexa) and hexahedra for meshing a circle. The arrows represent the direction i and j for the discretization.*

4.4 Towards new CFD softwares

4.4.1 Extending unstructured grids to new element shapes

Unstructured grids contain tetrahedra and these elements have a shape not adapted to compute an anisotropic flow. Some unstructured solvers have been extended to treat what is called a hybrid grid in the literature, which is simply a grid with multi-element shapes. Using hexahedra and prisms, it is possible to define with accuracy the direction normal to a wall and therefore to discretize with accuracy the Navier-Stokes equations within the boundary layer. The other basic elements (pyramids and tetrahedra) are needed for backward compatibility (tetrahedra) or enable a bridge from four-nodes faces to three-nodes faces in the mesh.

However, the extrapolation of the simple theoretical analysis to 3D meshes with good quality and anisotropy on complex CAD is not easy! Actually, defining an automatic mesh

generation process for multi-element grid is very complex and needs an increase of the interaction between the mesh generator and its user, something in opposition with automatic capabilities. The unstructrured CFD code AVBP offers the possibility to treat multi-elements grids.

4.4.2 Mixing structured and unstructured capabilities

A lot of numerical techniques are proposed in the literature to define structured meshes with a limited amount of time. However, when the CAD is very complex, it is not possible to build a structured mesh in a duration compatible with industrial constraints. One can consider as an example the flow inside a nacelle (between the engine and the nacelle, there are a lot of pipes with complex shapes). In this case, choosing an unstructured approach seems the best way to proceed. But unstructured grids are only usable by unstructured solvers. In *elsA*, it has been decided to implement inside one (and only one!) CFD kernel structured and unstructured capabilities. This is what we call **a hybrid capability**[1]. The implementation is based on an object oriented framework and the idea is to share common treatments in a mother class and to define typical treatments in sub-classes with inheritance of the mother class. Our current efforts concern the implementation of new capabilities within the hybrid grid context.

Remark 4.4.1 *The CFD code Fluent is assumed to handle hybrid grids issued from the blending of structured and multi-element unstructured zones. However, in practice, the numerical treatment is fully unstructured and there is a conversion of all structured zones in their unstructured equivalent counterparts.*

[1] See section 4.4.1 to understand that this name is not shared by the CFD community

Bibliography

[1] http://elsa.onera.fr.

[2] http://www.engineeringtoolbox.com/air-properties-24_156.html.

[3] R. Brun. *Transport et relaxation dans les écoulements gazeux*. série Physique Fondamentale et Appliquée. Edition Masson, 1986.

[4] P. Chassaing. *Mécanique des fluides*. Editions Cepadues, 1995.

[5] H. Guillard and R. Abgrall. *Modélisation numérique des fluides compressibles*. Series in Applied Mathematics. Edition Gauthier-Villars, 2001.

[6] P.-Y. Lagrée. Transferts thermiques et massiques dans les fluides. module MF204, cours de l'ENSTA accessible sur le web http://www.lmm.jussieu.fr/ lagree/COURS/ENSTA/C0entete.ENSTA.pdf, 2010.

[7] L. Landau and E. Lifchitz. *Physique théorique*. Mir Ed., second edition edition, 1989.

[8] M. Papin. *Contribution à la modélisation d'écoulements particulaires. Etude et validation d'un modèle diphasique discret*. PhD thesis, Université Bordeaux 1, 2005.

[9] D. Vandromme. *Contribution à la modélisation et à la prédiction d'écoulements turbulents à masse volumique variable*. PhD thesis, Université of Lille, 1983.