**EDF R&D**

Fluid Dynamics, Power Generation and Environment Department
Single Phase Thermal-Hydraulics Group

6, quai Watier
F-78401 Chatou Cedex

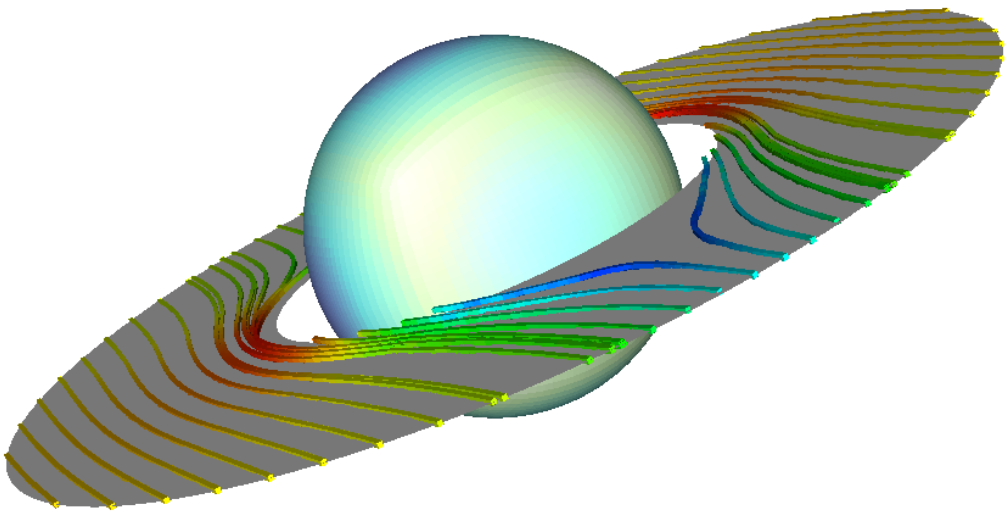
Tel: 33 1 30 87 75 40
Fax: 33 1 30 87 79 16

MARCH 2010

*Code_Saturne* documentation

# *Code_Saturne* 1.3.3 Theory and Programmer's Guide

contact: saturne-support@edf.fr



*ABSTRACT*

*Code_Saturne* solves the Navier-Stokes equations for 2D, 2D axisymmetric, or 3D, steady or unsteady, laminar or turbulent, incompressible or dilatable flows, with or without heat transfer, and with possible scalar fluctuations. The code also includes a Lagrangian module, a semi-transparent radiation module, a gas combustion module, a coal combustion module, an electric module (Joule effect and electric arc) and a compressible module. In the present document, the "gas combustion", "coal combustion", "electric" and "compressible" capabilities of the code will be referred to as "particular physics". The code uses a finite volume discretization. A wide range of unstructured meshes, either hybrid (containing elements of different types) and/or non-conform, can be used.

This document constitutes the theory and developer's guide associated with the kernel of *Code_Saturne*. Combustion, electric and compressible modules are also presented.

To make the documentation immediately suitable to the developers' needs, it has been organized into sub-sections corresponding to the major steps of the algorithm and to some important subroutines of the code. In each sub-section (for each subroutine), the **function** of the piece of code considered is indicated. Then, a description of the **discretisation** follows. Finally, and more oriented towards the developers, details of the **implementation** are provided and a list of open problems is given (improvements, limitations...).

To make it easier for the developers to keep the documentation up to date during the development process, the files have been associated "physically" with the release of the code (each release of the code includes a directory containing the whole documentation). In practice, the users of *Code_Saturne* can access the documentation at the following location `$CS_HOME/doc/NOYAU/`. The general command `info_cs [theory]` also provides this information.

**SOMMAIRE**


I Introduction 5

II Module de base 21

1- Sous-programme bilsc2 22

2- Sous-programme clptur 32

3- Sous-programme clptrg 52

4- Sous-programme clsyvt 64

5- Sous-programme codits 72

6- Sous-programme condli 78

7- Sous-programme covofi 94

8- Sous-programme gradmc 112

9- Sous-programme gradrc 118

10- Sous-programme inimas 128

11- Sous-programme itrmas/itrgrp 132

12- Sous-programme matrix 134

13- Sous-programme navsto 140

14- Sous-programme preduv 150

15- Sous-programme recvmc 162

16- Sous-programme resolp 164

17- Sous-programme turbke 174

18- Sous-programme turrij 180

19- Sous-programme viscfa 194

20- Sous-programme visort 196

21- Sous-programme vissec 200

22- Sous-programme vortex 204

III Module compressible 210

1- Sous-programme cfbl** 212

2- Sous-programme cfener 230

3- Sous-programme cfmsvl 236

4- Sous-programme cfqdmv 242

5- Sous-programme cfxtcl 246

IV Module électrique 266

1- Sous-programme elec** 268

V Module combustion 286

1- Sous-programme co**, cp**, fu** and so ... d3p*, ebu*, lwc*, pdf* 288

# Part I

# Introduction

# Introduction

## 1.1 Disclaimer

*Code_Saturne* is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

*Code_Saturne* is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.[1]

## 1.2 Aims of the document

This chapter constitutes an introduction to the theory and developer's guide associated with the kernel of *Code_Saturne*. The system of equations considered consists of the Navier-Stokes equations, with turbulence and passive scalars. First, the continuous equations for mass, momentum, turbulence and passive scalars are presented. Then, information related to the time scheme is supplied. Finally, the spatial discretisation is detailed: it is based on a co-located[2] finite volume scheme for unstructured meshes.

To make the documentation immediately suitable to the developers' needs, it has been organized into sub-sections corresponding to the major steps of the algorithm and to some important subroutines of the code.

In each sub-section (for each subroutine), the **function** of the piece of code considered is indicated. Then, a description of the **discretisation** follows. Finally, and more oriented towards the developers, details of the **implementation** are provided and a list of open problems is given (improvements, limitations...).

Several accessibility levels have been defined for the documentation, so that it is possible to choose the information level best suited to the intended use. At the present time, on UNIX and Linux platforms, free access is granted to all information (level called "`Complet`"). The most restrictive level provides only the function of the subroutines ("`Fonction`" level). The intermediate level provides the function of the subroutines and the associated discretisation ("`Discret`" level).

During the development process of the code, the documentation is naturally updated as and when required by the evolution of the source code itself. Suggestions for improvement are welcome. In particular, it will be necessary to deal with some transverse subjects (such as parallelism, periodicity or memory management) which were voluntarily left out of the first versions, to focus on the algorithms and their implementation.

To make it easier for the developers to keep the documentation up to date during the development process, the files have been associated "physically" with the release of the code (each release of the code includes a directory containing the whole documentation). In practice, the users of *Code_Saturne* will find the documentation at the following location (UNIX and Linux server at MFEE):

`$CS_HOME/doc/NOYAU/Postscripts/Base`,

The general command `info_cs [noyau]` also provides this information.

[1] You should have received a copy of the GNU General Public License along with *Code_Saturne*; if not, write to the Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA

[2] All the variables are located at the centres of the cells.

## 1.3 Continuous equations

This section presents the continuous equations. It is no substitutes for the specific sub-sections of this documentation: the purpose here is mainly to provide an overview before more detailed reading.

In the following, $\rho$ stands for the density, $\underline{u}$ for the velocity field. A mass source term, $\Gamma$ , may present, but in most cases the right-hand side of the Masss equation is $\Gamma = 0$

**Mass**

$$\mathrm{div}(\rho\underline{u}) = \Gamma \tag{I.1.1}$$

In fact, to compute a given unknown $\phi$ (and in particular for the velocity prediction), the equation $\dfrac{\partial\rho}{\partial t} + \mathrm{div}(\rho\underline{u}) = \Gamma$ is used to re-write the term $\dfrac{\partial\rho\,\phi}{\partial t}$ as follows: $\rho\dfrac{\partial\phi}{\partial t} - \phi\,\mathrm{div}(\rho\underline{u}) + \Gamma\,\phi$. The possible variations in time of the density are however not taken into account in the pressure correction step.

**Momentum**

$$\begin{cases} \dfrac{\partial}{\partial t}(\rho \underline{u}) + \mathrm{div}(\rho\, \underline{u} \otimes \underline{u}) = \mathrm{div}(\underline{\underline{\sigma}}) + \underline{ST} - \underline{\underline{K}}\, \underline{u} \\ \mathrm{div}(\rho \underline{u}) = 0 \end{cases} \tag{I.1.2}$$

$[\,\underline{ST} - \underline{\underline{K}}\,\underline{u}\,]$ stands for the additional momentum Source Terms which may be prescribed by the user (head loss, $\Gamma \underline{u}^i$ contribution associated with a user-prescribed mass source term...).

$\underline{\underline{K}}$ is a positive diagonal tensor, by definition (derived, for example, from the diagonal source term of the head loss tensor).

• For laminar flow, $\underline{\underline{\sigma}}$ is the stress tensor:

$$\underline{\underline{\sigma}} = \underline{\underline{\tau}} - P\underline{\underline{Id}} \tag{I.1.3}$$

where $\underline{\underline{\tau}}$ is the viscous stress tensor, defined from $\mu = \mu_l$ (dynamic molecular viscosity) and $\underline{\underline{S}}$ (strain rate tensor) as:

$$\underline{\underline{\tau}} = 2\,\mu\, \underline{\underline{S}} - \frac{2}{3}\mu\, tr(\underline{\underline{S}})\, \underline{\underline{Id}} \qquad \text{with} \qquad \underline{\underline{S}} = \frac{1}{2}(\underline{\underline{\mathrm{grad}}}\, \underline{u} + {}^t\underline{\underline{\mathrm{grad}}}\, \underline{u}) \tag{I.1.4}$$

• For turbulent flow, $\underline{\underline{\sigma}}$ also accounts for the turbulent Reynolds stress tensor (correlations of the velocity fluctuations arising from the non linear convective term). The modelling of the latter depends upon the turbulence model adopted:

- with an eddy-viscosity model (EVM) such as the $k-\varepsilon$ model, the closure requires a turbulent viscosity $\mu_t$. With formally the same definition for $\underline{\underline{\tau}}$ as in equation (I.1.4), but with $\mu = \mu_l + \mu_t$, $\underline{\underline{\sigma}}$ reads:

$$\underline{\underline{\sigma}} = \underline{\underline{\tau}} - (P + \frac{2}{3}\rho k)\underline{\underline{Id}} \tag{I.1.5}$$

- with the *LES* approach, the definition for $\underline{\underline{\sigma}}$ remains the same as for the EVM, above, but the turbulent viscosity $\mu_t$ now accounts only for the sub-grid effects.

- with a Differential Reynolds Stress Model (DRSM), the components of the Reynolds stress tensor $\underline{\underline{R}}$ are solved as extra variables during the simulation, and are readily available for the momentum equation, so one obtains, with $\mu = \mu_l$ in the definition of $\underline{\underline{\tau}}$ (equation (I.1.4)) :

$$\underline{\underline{\sigma}} = \underline{\underline{\tau}} - P\underline{\underline{Id}} - \rho\underline{\underline{R}} \tag{I.1.6}$$

In the following, only three standard types of turbulence models are described, as representatives of the types of equations that need to be dicretised. A more detailed description of available turbulence models is described in section (?? Turbulence Models??)

**Equations for the variables $k$ and $\varepsilon$ (standard $k-\varepsilon$ model)**

$$\left\{\begin{array}{l} \dfrac{\partial}{\partial t}(\rho k) + \mathrm{div}\left[\rho \underline{u}\, k - (\mu + \dfrac{\mu_t}{\sigma_k})\underline{\mathrm{grad}}\ k\right] = \mathcal{P} + \mathcal{G} - \rho\varepsilon + \Gamma k^i + ST_k \\ \dfrac{\partial}{\partial t}(\rho \varepsilon) + \mathrm{div}\left[\rho \underline{u}\, \varepsilon - (\mu + \dfrac{\mu_t}{\sigma_\varepsilon})\underline{\mathrm{grad}}\ \varepsilon\right] = C_{\varepsilon 1}\dfrac{\varepsilon}{k}\left[\mathcal{P} + (1-C_{\varepsilon 3})\mathcal{G}\right] - \rho C_{\varepsilon 2}\dfrac{\varepsilon^2}{k} + \Gamma\varepsilon^i + TS_\varepsilon \end{array}\right. \quad \text{(I.1.7)}$$

$\mathcal{P}$ is the production term created by mean shear:

$$\begin{aligned} \mathcal{P} &= -\rho R_{ij}\frac{\partial u_i}{\partial x_j} = -\left[-\mu_t\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)+\frac{2}{3}\mu_t\frac{\partial u_k}{\partial x_k}\delta_{ij}+\frac{2}{3}\rho k\delta_{ij}\right]\frac{\partial u_i}{\partial x_j} \\ &= \mu_t\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)\frac{\partial u_i}{\partial x_j} - \frac{2}{3}\mu_t(\mathrm{div}\underline{u})^2 - \frac{2}{3}\rho k\mathrm{div}(\underline{u}) \\ &= \mu_t\left[2\left(\frac{\partial u}{\partial x}\right)^2+2\left(\frac{\partial v}{\partial y}\right)^2+2\left(\frac{\partial w}{\partial z}\right)^2+\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)^2+\left(\frac{\partial u}{\partial z}+\frac{\partial w}{\partial x}\right)^2+\left(\frac{\partial v}{\partial z}+\frac{\partial w}{\partial y}\right)^2\right] \\ &\qquad -\frac{2}{3}\mu_t(\mathrm{div}\underline{u})^2 - \frac{2}{3}\rho k\mathrm{div}(\underline{u}) \end{aligned}$$

$\mathcal{G}$ is the production term created by gravity effects: $\mathcal{G} = -\dfrac{1}{\rho}\dfrac{\mu_t}{\sigma_t}\dfrac{\partial \rho}{\partial x_i}g_i$

The turbulent viscosity is: $\mu_t = \rho C_\mu \dfrac{k^2}{\varepsilon}$.

$ST_\varphi$ ($\varphi = k$ or $\varepsilon$) stands for the additional source terms prescribed by the user (in rare cases only).

The constants of the model are given below:

| $C_\mu$ | $C_{\varepsilon 1}$ | $C_{\varepsilon 2}$ | $\sigma_k$ | $\sigma_\varepsilon$ |
|---|---|---|---|---|
| $0,09$ | $1,44$ | $1,92$ | $1$ | $1,3$ |

$C_{\varepsilon 3} = 0$ if $\mathcal{G} \geqslant 0$ (unstable stratification) and $C_{\varepsilon 3} = 1$ if $\mathcal{G} \leqslant 0$ (stable stratification).

**Equations for the Reynolds stress tensor components $R_{ij}$ and $\varepsilon$ (LRR model)**

$$\left\{\begin{array}{ll} \dfrac{\partial}{\partial t}(\rho R_{ij}) + \mathrm{div}(\rho \underline{u}\, R_{ij} - \mu \underline{\mathrm{grad}}\ R_{ij}) & = \mathcal{P}_{ij} + G_{ij} + \Phi_{ij} + d_{ij} - \rho\varepsilon_{ij} \quad +\Gamma R^i_{ij} + ST_{R_{ij}} \\ \dfrac{\partial}{\partial t}(\rho\varepsilon) + \mathrm{div}\,[\rho \underline{u}\,\varepsilon - (\mu \underline{\mathrm{grad}}\ \varepsilon)] & = d + C_{\varepsilon_1}\dfrac{\varepsilon}{k}\left[\mathcal{P} + G_\varepsilon\right] - \rho C_{\varepsilon_2}\dfrac{\varepsilon^2}{k} \quad +\Gamma\varepsilon^i + ST_\varepsilon \end{array}\right. \tag{I.1.8}$$

$\underline{\underline{\mathcal{P}}}$ stands for the turbulence production tensor associated with mean flow strain-rate and $\underline{\underline{\mathcal{G}}}$ is stands for the production- tensor associated with buoyancy effects:

$$P_{ij} = -\rho\left[R_{ik}\frac{\partial u_j}{\partial x_k} + R_{jk}\frac{\partial u_i}{\partial x_k}\right] \qquad \text{and} \qquad G_{ij} = -\frac{3}{2}\frac{C_\mu}{\sigma_t}\frac{k}{\varepsilon}(r_i g_j + r_j g_i) \tag{I.1.9}$$

$$\mathrm{with} k \quad = \quad \frac{1}{2}R_{ll} \mathrm{and} r_i = R_{ik}\frac{\partial \rho}{\partial x_k} \tag{I.1.10}$$

$$\mathrm{moreover},\ \mathcal{P} \quad = \quad \frac{1}{2}\mathcal{P}_{kk} \qquad \text{and} \qquad G_\varepsilon = max(0, \frac{1}{2}G_{kk}) \tag{I.1.11}$$

$\underline{\underline{\Phi}}$ is the term representing pressure-velocity correlations:

$$\Phi_{ij} = \phi_{ij,1} + \phi_{ij,2} + \phi_{ij,3} + \phi_{ij,w} \tag{I.1.12}$$

$$\mathrm{with} \phi_{ij,1} = -\rho\, C_1 \frac{\varepsilon}{k}(R_{ij} - \frac{2}{3}k\delta_{ij}) \ \ , \ \ \phi_{ij,2} = -\rho\, C_2(\mathcal{P}_{ij} - \frac{2}{3}\mathcal{P})\delta_{ij} \quad \text{and} \quad \phi_{ij,3} = -C_3(G_{ij} - \frac{1}{3}\delta_{ij}G_{kk}) \tag{I.1.13}$$

The term $\phi_{ij,w}$ is called "wall echo term" (by default, it is not accounted for : see `turrij`).

The dissipation term, $\varepsilon_{ij}$ , is considered isotropic:

$$\varepsilon_{ij} = \frac{2}{3}\ \delta_{ij}\varepsilon \tag{I.1.14}$$

The turbulent diffusion terms are :

$$d_{ij} = C_S\frac{\partial}{\partial x_k}(\rho\frac{k}{\varepsilon}R_{km}\frac{\partial R_{ij}}{\partial x_m}) \qquad \text{and} \qquad d = C_\varepsilon\frac{\partial}{\partial x_k}\left(\rho\frac{k}{\varepsilon}R_{km}\frac{\partial \varepsilon}{\partial x_m}\right) \tag{I.1.15}$$

In the rare event of masse sources, $\Gamma R^i_{ij}$ and $\Gamma\varepsilon^i$ are the corresponding injection terms. $ST_{R_{ij}}$ and $ST_\varepsilon$ are also rarely used additional source terms that can prescribed by the user.

| $C_\mu$ | $C_\varepsilon$ | $C_{\varepsilon_1}$ | $C_{\varepsilon_2}$ | $C_1$ | $C_2$ | $C_3$ | $C_S$ | $C'_1$ | $C'_2$ |
|---|---|---|---|---|---|---|---|---|---|
| $0,09$ | $0,18$ | $1,44$ | $1,92$ | $1,8$ | $0,6$ | $0,55$ | $0.22$ | $0,5$ | $0,3$ |

**Definition of $\mu_t$ for $LES$**

⋆ Smagorinsky model

$$\mu_t = \rho\,(C_s\,\overline{\Delta})^2\sqrt{2\overline{S_{ij}}\,\overline{S_{ij}}} \tag{I.1.16}$$

With $\overline{S_{ij}}$ the filtered strain rate tensor components:

$$\overline{S_{ij}} = \frac{1}{2}\left(\frac{\partial\,\overline{u_i}}{\partial\,x_j} + \frac{\partial\,\overline{u_j}}{\partial\,x_i}\right) \tag{I.1.17}$$

Here, $\overline{u_i}$ stands for the $i^{th}$ resolved velocity component[3].

$C$ is the Smagorinsky constant. Its theoretical value is 0.18 for homogenous isotropic turbulence, but the value 0.065 is classic for channel flow.

$\overline{\Delta}$ is the filter width associated with the finite volume formulation (implicit filtering which corresponds to the integration over a cell). The value recommended for hexahedral cells is: $\overline{\Delta} = 2\,|\Omega|^{\frac{1}{3}}$ where $|\Omega|$ is the volume of the cell.

⋆ Classic dynamic model (??this should be moved to chapter turbulence models??)

A second filter is introduced: it is an explicit filter with a characteristic width $\widetilde{\Delta}$ superior to that of the implicit filter ($\overline{\Delta}$). If $a$ is a discrete variable defined over the computational domain, the variable obtained after applying the explicit filter to $a$ is noted $\tilde{a}$. Moreover, with:

$$L_{ij} = \widetilde{\overline{u_i}\,\overline{u_j}} - \widetilde{\overline{u_i}}\,\widetilde{\overline{u_j}}$$

$$\tau_{ij} = \overline{u_i u_j} - \overline{u_i}\,\overline{u_j}$$

$$T_{ij} = \widetilde{\overline{u_i u_j}} - \widetilde{\overline{u_i}}\,\widetilde{\overline{u_j}}$$

Germano identity reads:

$$L_{ij} = T_{ij} - \widetilde{\tau_{ij}}$$

Both dynamic models described herafter rely on the estimation of the tensors $\underline{\underline{T}}$ and $\underline{\underline{\tau}}$ as functions of the filter widths and of the strain rate tensor (Smagorinsky model). The following modelling is adopted[4]:

$$T_{ij} - \frac{1}{3}\delta_{ij}T_{kk} = -2C\widetilde{\Delta}^2|\widetilde{\overline{D_{ij}}}|\widetilde{\overline{D_{ij}}}$$

$$\tau_{ij} - \frac{1}{3}\delta_{ij}\tau_{kk} = -2C^*\overline{\Delta}^2|\overline{D_{ij}}|\overline{D_{ij}}$$

$\overline{u}$ stands for the "implicit-filtered" value of a variable $u$ defined at the centres of the cells and $\overline{\overline{u}}$ represents the "explicit-filtered" value associated with the variable $u$. It follows that the numerical computation of $L_{ij}$ is possible, since it requires the explicit filtering to be applied to implicitly filtered variables only (*i.e.* to the variables explicitly computed).

On the following assumption:

$$C = C^*$$

[3] In the case of implicit filtering, the discretisation in space introduces a spectral low pass filter: only the structures larger that twice the size of the cells are accounted for. Those structures are called "the resolved scales", whereas the rest, $u_i - \overline{u_i}$, is referred to as "unresolved scales" or "sub-grid scales". The influence of the unresolved scales on the resolved scales have to be modelled.

[4] $\delta_{ij}$ stands for the Kroeneker symbol.

and assuming that $C^*$ is only slightly non-uniform, so that it can be taken out of the explicit filtering operator, the following equation is obtained:

$$L_{ij} - \frac{1}{3}\delta_{ij}L_{kk} = C(\alpha_{ij} - \widetilde{\beta}_{ij})$$

with:

$$\alpha_{ij} = -2\widetilde{\Delta}^2|\widetilde{\overline{D_{ij}}}|\widetilde{\overline{D_{ij}}}$$
$$\beta_{ij} = -2\overline{\Delta}^2|\overline{D_{ij}}|\overline{D_{ij}}$$

Since we are left with six equations to determine one single variable, the least square method is used[5]. With:

$$E_{ij} = L_{ij} - C(\alpha_{ij} - \widetilde{\beta}_{ij})$$

the value for $C$ is obtained by solving the following equation:

$$\frac{\partial E_{ij}E_{ij}}{\partial C} = 0$$

Finally:

$$C = \frac{M_{ij}L_{ij}}{M_{kl}M_{kl}}$$

with

$$M_{ij} = \alpha_{ij} - \widetilde{\beta}_{ij}$$

This method allows to calculate the Smagorinsky "constant" dynamically at each time step and at each cell. However, the value obtained for $C$ can be subjected to strong variations. Hence, this approach is often restricted to flows presenting one or more homogeneous directions (Homogeneous Isotropic Turbulence, 2D flows presenting an homogeneous spanwise direction...): indeed, in such cases, the model can be (and is) stabilized by replacing $C$ by an average value of $C$ computed over the homogeneous direction(s).

For a general case (without any homogeneous direction), a specific averaging is introduced to stabilize the model: for any given cell of the mesh, the averaged Smagorinsky constant is obtained as an average of $C$ over the "extended neighbourhood" of the cell (the set of cells that share at least one vertex with the cell considered). More precisely, the average value (also denoted $C$ hereafter) is calculated as indicated below:

$$C = \frac{\widetilde{M_{ij}L_{ij}}}{\widetilde{M_{kl}M_{kl}}}$$

[5] $L_{kk}$ has no effect for incompressible flows.

**Equations for the scalars**

Two types of transport equations are considered:
⋆ advection of a scalar with additional source terms:

$$\frac{\partial(\rho a)}{\partial t} + \underbrace{\mathrm{div}\left((\rho \underline{u})\, a\right)}_{\text{convection}} - \underbrace{\mathrm{div}\left(K \underline{\mathrm{grad}}\ a\right)}_{\text{diffusion}} = TS_a + \Gamma\, a^i \tag{I.1.18}$$

⋆ advection of the variance $\widetilde{a''^2}$ with additional source terms :

$$\begin{aligned}\frac{\partial(\rho\widetilde{a''^2})}{\partial t} + \underbrace{\mathrm{div}\left((\rho\, \underline{u})\ \widetilde{a''^2}\right)}_{\text{convection}} - \underbrace{\mathrm{div}\left(K\ \underline{\mathrm{grad}}\ \widetilde{a''^2}\right)}_{\text{diffusion}} &= TS_{\widetilde{a''^2}} + \ \Gamma\, \widetilde{a''^2}^i \\ &\underbrace{+2\,\frac{\mu_t}{\sigma_t}(\underline{\mathrm{grad}}\ \widetilde{a})^2 - \frac{\rho\,\varepsilon}{R_f k}\ \widetilde{a''^2}}_{\text{production and dissipation}}\end{aligned} \tag{I.1.19}$$

The two previous equations can be unified formally as:

$$\frac{\partial(\rho f)}{\partial t} + \mathrm{div}\left((\rho\, \underline{u}) f\right) - \mathrm{div}\left(K \underline{\mathrm{grad}}\ f\right) = TS_f + \Gamma\, f^i + T_s^{pd} \tag{I.1.20}$$

with:

$$T_s^{pd} = \begin{cases} 0 & \text{for } f = a, \\ 2\ \dfrac{\mu_t}{\sigma_t}(\underline{\mathrm{grad}}\ \widetilde{a})^2 - \dfrac{\rho\,\varepsilon}{R_f k}\ \widetilde{a''^2} & \text{for } f = \widetilde{a''^2} \end{cases} \tag{I.1.21}$$

$TS_f$ represents the additional source terms that may be prescribed by the user.

## 1.4 Discretisation in time

At first, the physical properties of the flow are computed (density, viscosity, specific heat...): indeed, they may depend upon the variables (such as the temperature for example).

The time scheme is a $\theta$-scheme:

$$\begin{cases} \theta = 1 & \text{for an implicit first order Euler scheme} \\ \theta = 1/2 & \text{for second order Crank-Nicolson scheme} \end{cases} \tag{I.1.22}$$

For the second order scheme, the time step is assumed to be constant.

A fractional step scheme is used to solve the mass and momentum equations (Chorin). The first step (predictor step) provides predicted velocity components: they are determined sequentially and without coupling between each other. The mass equation is taken into account during the second step (corrector step): a pressure Poisson equation is solved and the mass fluxes at the cell faces are updated.

If required, the equations for the turbulent variables are solved (turbulent kinetic energy and dissipation or Reynolds stresses and dissipation), using a $\theta$-scheme again. For the $k-\varepsilon$ model, an additional step is carried out to couple the source terms. For the Reynolds stress model, the variables (turbulent stresses and dissipation) are solved sequentially, without coupling.

Next, the equations for the "scalars" (enthalpy, temperature, tracers, concentrations, mass fractions...) are solved, also with a $\theta$-scheme.

Finally, all the variables are updated and another time step may start.

The general equation for advection (valid for the velocity components, the turbulent variables and the scalars) is re-written as follows in a condensed form; the mass equation ($\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho \underline{u}) = \Gamma$) has been used to split the time derivative:

$$\rho \frac{\partial f}{\partial t} + \mathrm{div}\left((\rho\, \underline{u}) f\right) - \mathrm{div}\left(K \underline{\mathrm{grad}}\ f\right) = S_i(\Phi,\varphi)\, f + S_e(\Phi,\varphi) + \mathrm{div}\left(\rho\, \underline{u}\right) f \tag{I.1.23}$$

In this equation:

| | | |
|---|---|---|
| $\Phi$ | : | represents the physical properties $(\rho, K, \mu_t, ...)$ |
| $\varphi$ | : | represents the variables of the problem $(\underline{u}, k, \epsilon, ...)$ |
| $S_i(\Phi,\varphi)\, f$ | : | represents the linear part of the source terms |
| $S_e(\Phi,\varphi)$ | : | includes all other source terms |
| $\mathrm{div}\left(\rho\, \underline{u}\right) f$ | : | is the term associated with "mass accumulation" |

The time at which the different quantities are evaluated is indicated below:
- $\Phi$: the time considered is defined by the time scheme applied to the physical properties.
- $(\rho\, \underline{u})$: the time considered is defined by the time scheme applied to the mass flux.
- $S_e(\Phi,\varphi)$: the time considered is defined by the time scheme applied to the explicit source terms.

If $\theta = 1/2$, or if an extrapolation is used, the time step $\Delta t$ is constant in time and uniform in space.

### 1.4.1 Physical properties

The physical properties of the flow (density, viscosity, specific heat...) are:

- either explicit, defined at the time step $n$.
- or extrapolated at $n + \theta_\Phi$ using the Adam-Bashforth time scheme (in this case, the time step is assumed to be constant).

Under a more general form, this reads:

$$\Phi \equiv \Phi^{n+\theta_\Phi} = (1+\theta_\Phi)\, \Phi^n - \theta_\Phi\, \Phi^{n-1} \tag{I.1.24}$$

$$\begin{cases} \theta_\Phi = 0 & \text{standard explicit formulation} \\ \theta_\Phi = 1/2 & \text{second order extrapolation at } n+1/2 \\ \theta_\Phi = 1 & \text{first order extrapolation at } n+1 \end{cases} \tag{I.1.25}$$

### 1.4.2 Mass flux

For the mass flux, three time schemes are available. The mass flux may be:

- explicit, taken at time step $n$ for the momentum equations and updated with its value at time step $n+1$ for the equations for turbulence and scalars (standard scheme).

- explicit, taken at time step $n$ for the momentum equations and also for the equations for turbulence and scalars.

- taken at $n+\theta_F$ (second order if $\theta_F = 1/2$). To solve the momentum equations, $(\rho\,\underline{u})^{n-2+\theta_F}$ and $(\rho\,\underline{u})^{n-1+\theta_F}$ are known. Hence, the value at $n+\theta_F$ is obtained as a result of the following extrapolation:

$$(\rho\,\underline{u})^{n+\theta_F} = 2\,(\rho\,\underline{u})^{n-1+\theta_F} - \,(\rho\,\underline{u})^{n-2+\theta_F} \tag{I.1.26}$$

  At the end of this phase (after the pressure correction step), $(\rho\,\underline{u})^{n+1}$ is known and the following interpolation is used to determine the mass flux at $n+\theta_F$ that will be adopted for the equations for turbulence and scalars:

$$(\rho\,\underline{u})^{n+\theta_F} = \frac{1}{2-\theta_F}\,(\rho\,\underline{u})^{n+1} + \frac{1-\theta_F}{2-\theta_F}\,(\rho\,\underline{u})^{n-1+\theta_F} \tag{I.1.27}$$

### 1.4.3 Source terms

As for the physical properties, the **explicit** source terms are:

- explicit:

$$[S_e(\Phi,\varphi)]^n = S_e(\Phi^{n+\theta_\Phi},\varphi^n) \tag{I.1.28}$$

- extrapolated at $n+\theta_S$ using the Adams-Bashforth scheme:

$$[S_e(\Phi,\varphi)]^{n+\theta_S} = (1+\theta_S)\,S_e(\Phi^n,\varphi^n) - \theta_S\,S_e(\Phi^{n-1},\varphi^{n-1}) \tag{I.1.29}$$

By default, to be consistent and preserve the order of convergence in time, the implicit source terms are discretized with the same scheme as that used for convection-diffusion of the unknown considered, *i.e.* taken at $n+\theta$:

$$[S_i(\Phi,\varphi)\,f]^{n+\theta} = S_i(\Phi^{n+\theta_\Phi},\varphi^n)\,[\theta\,f^{n+1} + (1-\theta)\,f^n] \tag{I.1.30}$$

Note:
The **implicit** source terms taken also at $n+\theta$ for $\theta_S \neq 0$, while for $\theta_S = 0$, the implicit source terms are taken at $n+1$ , this to enhance stability.

### 1.4.4 General discrete form

For the sake of clarity, it is assumed hereafter that, unless otherwise explicitly stated, the mass flux is taken at $n+\theta_F$ and the physical properties are taken at $n+\theta_\Phi$, with $\theta_F$ and $\theta_\Phi$ dependant upon the specific schemes selected for the mass flux and the physical properties respectively.

Under a general form, the discrete counterpart of equation (I.1.23) at $n+\theta$ reads:

$$\begin{array}{c}\dfrac{\rho}{\Delta t}(f^{n+1}-f^n)+\mathrm{div}\left((\rho\,\underline{u})f^{n+\theta}\right)-\mathrm{div}\left(K\underline{\mathrm{grad}}\ f^{n+\theta}\right)= \\ \left[S_i(\Phi,\varphi)\,f\right]^{n+\theta}+\left[S_e(\Phi,\varphi)\right]^{n+\theta_S}+\mathrm{div}\left(\rho\,\underline{u}\right)f^{n+\theta}\end{array} \tag{I.1.31}$$

Using the standard $\theta$-scheme $f^{n+\theta}=\theta\,f^{n+1}+(1-\theta)\,f^n$, the equation reads:

$$\begin{array}{c}\dfrac{\rho}{\Delta t}(f^{n+1}-f^n)+\theta\,\mathrm{div}\left((\rho\,\underline{u})f^{n+1}\right)-\theta\,\mathrm{div}\left(K\underline{\mathrm{grad}}\ f^{n+1}\right)= \\ -(1-\theta)\,\mathrm{div}\left((\rho\,\underline{u})f^n\right)+(1-\theta)\,\mathrm{div}\left(K\underline{\mathrm{grad}}\ f^n\right)+ \\ S_i(\Phi,\varphi^n)\left[\theta\,f^{n+1}+(1-\theta)\,f^n\right]+\left[S_e(\Phi,\varphi)\right]^{n+\theta_S}+\mathrm{div}\left(\rho\,\underline{u}\right)\left(\theta\,f^{n+1}+(1-\theta)\,f^n\right)\end{array} \tag{I.1.32}$$

For numerical reasons, the system is solved in an iterative and incremental manner, with the help of the series $\delta f^{n+1,k+1}=f^{n+1,k+1}-f^{n+1,k}$ (with, by definition, $f^{n+1,0}=f^n$). In particular, this method allows to treat implicitly a portion of the advection-diffusion terms associated correction terms for non orthogonal meshes. Hence, the system actually solved reads:

$$\begin{array}{c}\underbrace{\left[\dfrac{\rho}{\Delta t}-\theta\,S_i(\Phi,\varphi^n)-\theta\,\mathrm{div}\left(\rho\,\underline{u}\right)\right]}_{\texttt{ROVSDT}}\left(f^{n+1,k+1}-f^{n+1,k}\right) \\ +\theta\,\mathrm{div}\left((\rho\,\underline{u})\left(f^{n+1,k+1}-f^{n+1,k}\right)\right)-\theta\,\mathrm{div}\left(K\underline{\mathrm{grad}}\ \left(f^{n+1,k+1}-f^{n+1,k}\right)\right)= \\ \texttt{SMBRS}\left\{\begin{array}{c}-\theta\,\mathrm{div}\left((\rho\,\underline{u})\,f^{n+1,k}\right)+\theta\,\mathrm{div}\left(K\underline{\mathrm{grad}}\ f^{n+1,k}\right) \\ -(1-\theta)\,\mathrm{div}\left((\rho\,\underline{u})\,f^n\right)+(1-\theta)\,\mathrm{div}\left(K\underline{\mathrm{grad}}\ f^n\right) \\ -\dfrac{\rho}{\Delta t}(f^{n+1,k}-f^n)+S_i(\Phi,\varphi^n)\left[\theta\,f^{n+1,k}+(1-\theta)\,f^n\right] \\ +\mathrm{div}\left(\rho\,\underline{u}\right)\left(\theta\,f^{n+1,k}+(1-\theta)\,f^n\right)+\left[S_e(\Phi,\varphi)\right]^{n+\theta_S}\end{array}\right.\end{array} \tag{I.1.33}$$

Whatever the equation considered (momentum equation, equations for turbulence or scalars,...) the system representation is always the same: a right-hand side (stored in the vector-array `SMBRS`) and a vector-array `ROVSDT` for the part linear with respect to $\delta f^{n+1,k+1}$.

The vector-array `ROVSDT` is computed by the subroutine `covofi` for the scalars, by `preduv` for the velocity and by `turbke` or `turrij` for the turbulence.

The vector-array `SMBRS` is not computed at one go, but in two successive steps.

The first part is calculated by the subroutines `covofi`, `preduv`, `turbke` and `turrij` (respectively for the scalars, the velocity and the turbulence). At that point, the vector `SMBRS` is defined as follows:

$$\texttt{SMBRS}=S_i(\Phi,\varphi^n)\,f^n+\mathrm{div}\left(\rho\,\underline{u}\right)f^n+\left[S_e(\Phi,\varphi)\right]^{n+\theta_S} \tag{I.1.34}$$

then, the calculation of `SMBRS` is complemented at each sub-iteration during the resolution of the equation by the subroutine `codits` as follows:

$$\begin{array}{c}\texttt{SMBRS}=\texttt{SMBRS}-\texttt{ROVSDT}\left(f^{n+1,k}-f^n\right) \\ -\theta\,\mathrm{div}\left((\rho\,\underline{u})\,f^{n+1,k}\right)+\theta\,\mathrm{div}\left(K\underline{\mathrm{grad}}\ f^{n+1,k}\right) \\ -(1-\theta)\,\mathrm{div}\left((\rho\,\underline{u})\,f^n\right)+(1-\theta)\,\mathrm{div}\left(K\underline{\mathrm{grad}}\ f^n\right)\end{array} \tag{I.1.35}$$

## 1.5 Discretisation in space

Within the framework of the finite volume approach, the equations are integrated over each cell of the mesh (or "control volume" $\Omega_i$). This section is limited to a brief description of the way the terms appearing in the equations are integrated. Specific attention is devoted to the calculation of gradients, since it is a major characteristic of the co-located finite volume method (all the variables are associated with the same point, namely the cell centre[6]).

The terms of **order 0** (*i.e.* the terms that are not a space derivative) are integrated to introduce their average over the cell. For example, $\rho g$ becomes $|\Omega_i|\overline{\rho}_i\, g$. In this expression, $|\Omega_i|$ is the volume of cell $\Omega_i$ and $\overline{\rho}_i$ denotes the average of $\rho$ over the control volume (the cell) $\Omega_i$. For less clumsy notations, $|\Omega_i|\overline{\rho}_i\, g$ is simply written as $|\Omega_i|\rho_i\, g$. When "reconstructions" (in fact Taylor series) are required to reach a higher order in space, the average value $\rho_i$ is assumed to be associated with the centre I of $\Omega_i$.

The "divergence" terms (or "flux" terms, or again "conservative" terms) are integrated using the Green relation to introduce cell faces values (and so, "fluxes" appear naturally). For example, a term such as $div(p)$ becomes[7] $\sum_{j\in Neibr(i)} \overline{p}_{ij} S_{ij}$. In this expression, $\overline{p}_{ij}$ is the average of $p$ on the interface between the neighbouring cells $i$ and $j$. The summation is carried out for $j \in Neibr(i)$, that is, all cells in the neighbourhood of $\Omega_i$ thus sharing a cell face with it. As for the cell-average above, the overbar is omitted: the sum is simply written $\sum_{j\in Neibr(i)} p_{ij} S_{ij}$. The value $p_{ij}$ is assumed to be associated with the centre F of the face $ij$ when "reconstructions" are needed to reach a higher order in space.

The precision of the value $p_{ij}$ determines the precision of the calculation of $div(p)$. For $p_{ij}$, it is possible to take a non-centred and non-interpolated value (upwind scheme for convection) or a linear interpolation between the values at the centres I and J of the neighbouring cells. Both methods are relatively straighforward but may lack consistence and precision for arbitrary meshes (and in particular on non-orthogonal meshes). A higher order in space may be reached if reconstruction techniques are used. The idea is to compute the value for $p_{ij}$ more precisely: to do so, $p$ is interpolated at $F_{ij}$ (the centre of the face) using the values for $p$ at I and J and... the gradients of $p$ calculated at I and J. The reader will notice that it is precisely the calculation of the space derivatives[8] of $p$ that motivated the need for a good interpolation of $p$ at $F_{ij}$. Hence, the computation of the space derivatives of $p$ requires to solve a system: this is done in an iterative manner.

Doing so allows to calculate the **"cell gradient"** of the variables. It is important to keep in mind that the "cell gradient" of a given variable represents the gradient of the variable in the cell and that it is obtained from the values of the variable interpolated at the cell faces.

Similarly, the terms written as **"divergence of gradient"** are integrated to introduce face values. One has to calculate the gradient at each face (or "face gradient") in the direction normal to the face. This concept of "face gradient" is extremely important. The "face gradient" normal to a face can be easily calculated with just the values at the centres of the two cells that share the face (points I and J on figure I.1.1), but this method is limited to orthogonal meshes. For consistence and to reach a higher order in space, the values of the variables at points I' and J' have to be used. These are calculated by a Taylor series from the values at I and J and from the "cell gradient" previously determined.

[6] The centre of a cell is a geometric point associated with the cell and located preferably inside the cell. Nevertheless, the word "centre" shall not be taken literally, especially in the case of polyhedral cells that do not have a regular shape.

[7] If the cell $i$ is at the domain boundary, the sum becomes $\sum_{j\in Vois(i)} \overline{p}_{ij} S_{ij} + \sum_{k\in\gamma(i)} \overline{p}_{b_{ik}} S_{b_{ik}}$, with $b_{ik}$ referring to the faces of the cell $i$ which are at the domain boundary.

[8] The first derivatives in space are required to obtain $div(p)$ in each cell.

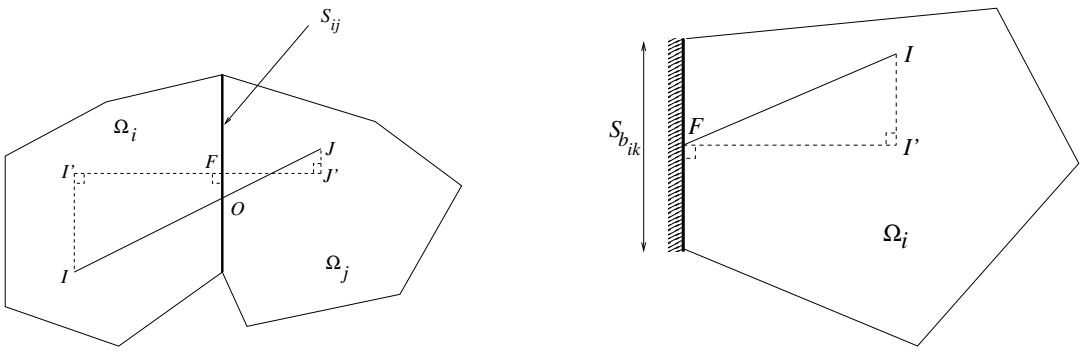



Figure I.1.1: Identification of the geometric entities for internal (left) and boundary faces (right).

## 1.6 Calling tree

Each sub-section of this document is associated with an important subroutine. The full list of the subroutines described here is the following: `bilsc2` `clptur` `clsyvt` `codits` `condli` `covofi` `gradmc` `gradrc` `inimas` `itrmas` `matrix` `navsto` `preduv` `recvmc` `resolp` `turbke` `turrij` `viscfa` `visort` `vissec`.

The table 1 presents their sequence within a time step. This calling tree is only partial. In particular, it does not account for the number of calls to each subroutine. Also, for the sake of clarity, no reference has been made to the subroutines dedicated to the gradient calculation (`gradmc`, `gradrc`), which are called very often. For the same reason, the calls to `bilsc2` (advection fluxes) and `matrix` (matrix calculation) which are made from within `codits` (treatment of an advection equation with source terms) have not been reported.

The sub-sections where important information can be found are indicated below:

**Calculation of gradients**
- `gradrc`
- `gradmc`

**Least square method**
- `recvmc`
- `gradmc`

**Convective schemes**
- `bilsc2`

**Wall-laws** (for velocity and temperature)
- `clptur`
- `condli`

**System solve** (incremental method)
- `codits`

**Calculation of the values at the faces** (not exhaustive)
- `viscfa`
- `visort`

Finally, for the reader wishing to become more familiar with the methods implemented in *Code_Saturne*, it is recommended to begin with the study of the advection equation for a scalar (`covofi`) which is solved iteratively using an incremental method (`codits`). It will then be useful to look at `navsto` which briefly presents the solution of the system made up of the mass and momentum equations.

**Calculation of the physical properties**
**Boundary Conditions**
`condli`
`clptur` "turbulent" conditions at the wall
`clsyvt` symmetry conditions for the vectors and the tensors
**Navier-Stokes solution**
`navsto`
**Velocity prediction**
`preduv`
`vissec` momentum source terms related to the transposed gradient of the velocity
`viscfa` calculation of the viscosity at the faces
`codits` iterative solution of the system using an incremental method
**Pressure correction**
`resolp`
`viscfa` calculation of the time step at the faces...
`visort` ...according to the selected options
`matrix` calculation of the Poisson equation matrix
`inimas` initialisation of the mass flow rate
`itrmas` update of the mass flow rate
**Velocity correction**
standard method or...
`recvmc` least square method
$k-\varepsilon$ **model**
`turbke`
`viscfa` preliminary steps before...
`bilsc2` ...source terms coupling
`viscfa` calculation of the viscosity at the faces
`codits` iterative solution of the systems using an incremental method
**Reynolds stress model**
`turrij`
`visort` calculation of the viscosity at the faces
`codits` iterative solution of the systems using an incremental method
**Equations for the scalars**
`covofi`
`viscfa` calculation of the viscosity at the faces
`codits` iterative solution of the systems using an incremental method

Table 1: Partial and simplified calling tree associated with the successive stages within a time step.

# Part II

# Module de base

# 1- Sous-programme bilsc2

## 1.1 Fonction

In this subroutine, called by `codits` and `turbke`, the contributions to the explicit budget of the reconstructed (on non-orthogonal meshes and if the user chooses to) convective and diffusive terms of the right-hand side of a convection/diffusion equation for a scalar $a$ are computed. These terms write [1]:

$$\mathcal{B}_\beta((\rho\,\underline{u})^n, a) \quad = \underbrace{-\mathrm{div}(\,(\rho\underline{u})^n a)}_{\text{convective part}} + \underbrace{\mathrm{div}(\,\beta\,\underline{\mathrm{grad}}\;a)}_{\text{diffusive part}} \tag{II.1.1}$$

with $\rho$, $\underline{u}$, $\beta$ and $a$ the variables at time $t^n$.

## 1.2 Discretization

### 1.2.1 Convective Part

Using the notations adopted in the subroutine `navsto`, the explicit budget corresponding to the integration over a cell $\Omega_i$ of the convective part $-\mathrm{div}(\,(\rho\,\underline{u})^n a)$ of $\mathcal{B}_\beta$ can be written as a sum of the numerical fluxes $F_{ij}$ calculated at the faces of the internal cells, and the numerical fluxes $F_{b_{ik}}$ calculated at the boundary faces of the computational domain $\Omega$. Let's take $Vois(i)$ the set of the centres of the neighbouring cells of $\Omega_i$ and $\gamma_b(i)$ the set of the centres of the boundary faces of $\Omega_i$ (if they exist). Thus we can write

$$\int_{\Omega_i} \mathrm{div}((\rho\underline{u})^n a)\,d\Omega = \sum_{j\in Vois(i)} F_{ij}((\rho\underline{u})^n, a) + \sum_{k\in\gamma_b(i)} F_{b_{ik}}((\rho\underline{u})^n, a)$$

with :

$$F_{ij}((\rho\underline{u})^n, a) = \left[(\rho\underline{u})^n_{ij}.\,\underline{S}_{ij}\right]\; a_{f,ij} \tag{II.1.2}$$

$$F_{b_{ik}}((\rho\underline{u})^n, a) = \left[(\rho\underline{u})^n_{b_{ik}}.\,\underline{S}_{b_{ik}}\right]\; a_{f\,b_{ik}} \tag{II.1.3}$$

where $a_{f,ij}$ and $a_{f\,b_{ik}}$ represent the values of $a$ at the internal and boundary faces of $\Omega_i$, respectively. représentent les valeurs aux faces internes et de bord respectivement.

Before presenting the different convection schemes available in *Code_Saturne*, we define:

$$\alpha_{ij} = \frac{\overline{FJ'}}{\overline{I'J'}} \text{ defined at the internal faces only and}$$

$$\underline{u}_{K'} = \underline{u}_K + (\underline{\underline{\mathrm{grad}}}\;\underline{u})_K.\,\underline{KK'} \text{ at the first order in space, for } K = I \text{ or } J$$

The value of the convective flux $F_{ij}$ depends on the numerical scheme. Three different types of convection schemes are available in this subroutine:

[1] They appear on the right-hand side of the incremental system for cell $I$ of the momentum prediction step: $\mathcal{EM}(\delta\underline{u}^{k+1}, I) = \mathcal{E}(\underline{u}^{n+1/2,k}, I)$ (see `navsto` for more details)

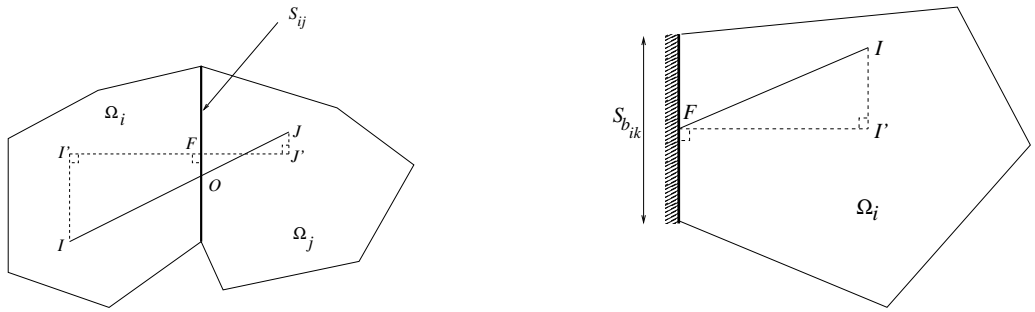



Figure II.1.1: Definition of the geometric entities for internal (left) and a boundary faces (right).

- a $1^{st}$ order upwind scheme:

$$F_{ij}((\rho\underline{u})^n, a) = F_{ij}^{upstream}((\rho\underline{u})^n, a)$$

où : $$a_{f,ij} = \begin{cases} a_I \text{ si } (\rho\underline{u})^n_{ij} \cdot \underline{S}_{ij} \geqslant 0 \\ a_J \text{ si } (\rho\underline{u})^n_{ij} \cdot \underline{S}_{ij} < 0 \end{cases}$$

- a centered scheme:

$$F_{ij}((\rho\underline{u})^n, a) = F_{ij}^{centered}((\rho\underline{u})^n, a)$$

with : $a_{f,ij} = \alpha_{ij} a_{I'} + (1 - \alpha_{ij}) a_{J'}$

- a Second Order Linear Upwind scheme (SOLU):

$$F_{ij}((\rho\underline{u})^n, a) = F_{ij}^{SOLU}((\rho\underline{u})^n, a)$$

with : $$a_{f,ij} = \begin{cases} a_I + \underline{IF} \cdot (\underline{\text{grad}}\ a)_I \text{ si } (\rho\underline{u})^n_{ij} \cdot \underline{S}_{ij} \geqslant 0 \\ a_J + \underline{JF} \cdot (\underline{\text{grad}}\ a)_J \text{ si } (\rho\underline{u})^n_{ij} \cdot \underline{S}_{ij} < 0 \end{cases}$$

The value of $F_{b_{ik}}$ is calculated as :

$$a_{f\,b_{ik}} = \begin{cases} a_I & \text{if } (\rho\underline{u})^n_{b_{ik}} \cdot \underline{S}_{b_{ik}} \geqslant 0 \\ a_{b_{ik}} & \text{if } (\rho\underline{u})^n_{b_{ik}} \cdot \underline{S}_{b_{ik}} < 0 \end{cases}$$

$a_{b_{ik}}$ is the boundary value directly computed from the prescribed boundary conditions. REMARK 1

When a centered scheme is used, we actually write (to ensure first order discretization in space for $a$)

$$a_{f,ij} = \alpha_{ij} a_I + (1 - \alpha_{ij}) a_J + \frac{1}{2} \left[ (\underline{\text{grad}}\ a)_I + (\underline{\text{grad}}\ a)_J \right] \cdot \underline{OF}$$

A factor $\frac{1}{2}$ is used for numerical stability reasons.

REMARK 2
A slope test (which may introduce non-linearities in the convection operator) allows to switch from

[1] Extrapolation of the upwind value at the faces centre.

the centered or SOLU scheme to the first order upwind scheme (without blending). Additionally, in standard mode $a_{f,ij}$ is computed as a weighted average between the upstream value and the centered value (blending), according to users' choice (variable `BLENCV` in the subroutine `usini1`).

### 1.2.2 Diffusive Part

Similarly, the diffusive part writes :

$$\int_{\Omega_i} \mathrm{div}(\beta \ \underline{\mathrm{grad}}\ a)\ d\Omega = \sum_{j \in Vois(i)} D_{ij}(\beta, a) + \sum_{k \in \gamma_b(i)} D_{b_{ik}}(\beta, a)$$

with:

$$D_{ij}(\beta, a) = \beta_{ij} \frac{a_{J'} - a_{I'}}{\overline{I'J'}} S_{ij} \tag{II.1.4}$$

and :

$$D_{b_{ik}}(\beta, a) = \beta_{b_{ik}} \frac{a_{b_{ik}} - a_{I'}}{\overline{I'F}} S_{b_{ik}} \tag{II.1.5}$$

using the same notations as before, and with $S_{ij}$ and $S_{b_{ik}}$ being the norms of vectors $\underline{S}_{ij}$, $S_{b_{ik}}$ and $\underline{S}_{b_{ik}}$, $a_{b_{ik}}$, respectively.

## 1.3 Implementation

In the following, the reader is reminded of the role of the variables used in the different tests: • `IRCFLP`, from array `IRCFLU` ; indicates for the considered variables wether or not the convective and diffusive fluxes are reconstructed

= 0 : no reconstruction

= 1 : reconstruction

• `ICONVP`, from array `ICONV` ; indicates if the considered variables is convected or not.

= 0 : no convection

= 1 : convection

• `IDIFFP`, from array `IDIFF` ; indicates if the diffusion of the considered variables is taken into account or not.

= 0 : no diffusion

= 1 : diffusion

• `IUPWIN` indicates locally, in `bilsc2` (to avoid unnecessary calculations) whether a pure upwind scheme is chosen or not for the considered variables to be convected.

= 0 : no pure upwind

= 1 : pure upwind is used

• `ISCHCP`, from array `ISCHCV` ; indicates which type of second order convection scheme is used on orthogonal meshes for the considered variable to convect (only useful if `BLENCP` $> 0$ ).

= 0 : we use the SOLU scheme (Second Order Linear Upwind )

= 1 : we use a centered scheme

In both cases the blending coefficient `BLENCP` needs to be given in `usini1`.

• `BLENCP`, from array `BLENCV` ; indicates the percentage of centered or SOLU convection scheme that one wants to use. This weighting coefficient is between 0 and 1.

• `ISSTPP`, from array `ISSTPC` ; indicates if one wants to remove the slope test that switches the convection scheme from second order to upwind if the test is positive.

= 0 : a slope test is systematically used

= 1 : no slope test

### 1.3.1 Computation of the gradient $\underline{G}_{c,i}$ of variable $a$

The computation of the gradient of variable $a$ is necessary for the computation of the explicit budget. `grdcel` is called everytime this gradient is needed, and it is stored in the array (`DPDX`, `DPDY`, `DPDZ`). The computation of the gradient is necessary in the following situations:

- if the convection is activated with a non pure upwind scheme (`ICONVP` $\neq 0$ and `IUPWIN` $= 0$) **and**,
  if we want to reconstruct the fluxes (`IRCFLP` $= 1$),
  **or** if we want to use the SOLU scheme (`ISCHCP` $= 0$),
  **or** if we use the slope test (`ISSTPP` $= 0$),

or :

- if there is diffusion and we want to reconstruct the fluxes (`IDIFFP` $\neq 0$ and `IRCFLP` $= 1$).

In all other cases, the array (`DPDX`, `DPDY`, `DPDZ`) is set to zero.

### 1.3.2 Computation of the upwind gradient $\underline{G}_{c,i}^{amont}$ of variable $a$

$\underline{G}_{c,i}^{amont}$ refers to the upwind gradient of variable $a$, for cell $\Omega_i$. It is stored in the array (`DPDXA`, `DPDYA`, `DPDZA`). We also define the scalars $a_{ij}^{amont}$ and $a_{b_{ik}}^{amont}$ as:

$$|\Omega_i|\,\underline{G}_{c,i}^{upwind} \overset{def}{=} \sum_{j\in Vois(i)} a_{ij}^{upwind}\,\underline{S}_{ij} + \sum_{k\in\gamma_b(i)} a_{b_{ik}}^{upwind}\,\underline{S}_{b_{ik}} \qquad \text{(II.1.6)}$$

After initializing it to zero, $\underline{G}_{c,i}^{amont}$ **is only computed** when the user wishes to compute **a convection term with a centered or SOLU method, and a slope test**.

- For each cell $\Omega_i$, the face values $a_{IF}$ (variable `PIF`) and $a_{JF}$ (variable `PJF`), are computed as:

$$\begin{aligned} a_{IF} &= a_I + \underline{IF}\,.\,(\underline{\text{grad}}\ a)_I \\ a_{JF} &= a_J + \underline{JF}\,.\,(\underline{\text{grad}}\ a)_J \end{aligned}$$

Depending on the sign $s_{ij}^n$ of the mass flux $(\rho\underline{u})_{ij}^n$. $\underline{S}_{ij}$, we give $a_{IF}$ or $a_{JF}$ the value $a_{ij}^{upwind}$ of the expression $\sum_{j\in Vois(i)} a_{ij}^{upwind}\,\underline{S}_{ij}$.

$$a_{ij}^{upwind} = \begin{cases} a_I + \underline{IF}\,.\,(\underline{\text{grad}}\ a)_I & \text{si } s_{ij}^n = 1 \\ a_J + \underline{JF}\,.\,(\underline{\text{grad}}\ a)_J & \text{si } s_{ij}^n = -1 \end{cases}$$

• The boundary terms are computed in a classic manner as follows (keeping the same notations as in the other subroutines):

$$\begin{aligned} \sum_{k\in\gamma_b(i)} a_{b_{ik}}^{upwind}\,\underline{S}_{b_{ik}} &= \sum_{k\in\gamma_b(i)} (\texttt{INC}\,A_{b,ik} + B_{b,ik}\,a_{I'})\,\underline{S}_{b_{ik}} \\ &= \sum_{k\in\gamma_b(i)} \left[\texttt{INC}\,A_{b,ik} + B_{b,ik}\,a_I + B_{b,ik}\,\underline{II'}\,.\,\underline{G}_{c,i}\right]\,\underline{S}_{b_{ik}} \end{aligned}$$

$(A_{b,ik}, B_{b,ik})_{k\in\gamma_b(i)}$ are stored in the arrays (`COEFAP`, `COEFBP`). The vector $\underline{II'}$ is stored in the array (`DIIPBX`, `DIIPBY`, `DIIPBZ`). The surfaces $(\underline{S}_{b_{ik}})_{k\in\gamma_b(i)}$ are stored in the array `SURFBO` .

### 1.3.3 Summation of the numerical convective and diffusive fluxes

The contributions to the explicit budget $[-\text{div}(\,(\rho\underline{u})^n a) + \text{div}(\,\beta\,\underline{\text{grad}}\ a)]$ are computed and added to the right-hand side array `SMBR`, which has already been initialized before the call to `BILSC2` (with the explicit source terms for instance, etc.).
The variable `FLUX` gathers the convective and diffusive parts of the numerical fluxes. It is computed in a classic manner, first on the internal faces, and then on the boundary faces. The indices $i$ and $j$

are represented by II and JJ, respectively.
In order to take into account (when necessary) the sign $s_{ij}^n$ of the mass flux $(\rho\underline{u})_{ij}^n.\ \underline{S}_{ij}$, the following equations are used :

For any real $b$, we have :

$$\left\{\begin{array}{ll} b & = b^+ + b^- \text{ with } b^+ = max\ (b,0),\quad b^- = min\ (b,0) \\ |\,b| & = b^+ - b^- \\ b^+ & = \dfrac{1}{2}\,[\ b + |\,b|\ ] \\ b^- & = \dfrac{1}{2}\,[\ b - |\,b\ |\ ] \end{array}\right.$$

In this subroutine, $b$ represents the mass flux FLUMAS(IFAC) on an internal face IFAC (FLUMAB(IFAC) for a boundary face IFAC) ; $b^+$ is stored in FLUI and $b^-$ in FLUJ.

- for an internal face $ij$ (IFAC)

We calculate :

$$\sum_{j\in Vois(i)} F_{ij}((\rho\underline{u})^n,a) - \sum_{j\in Vois(i)} D_{ij}(\beta,a) = \sum_{j\in Vois(i)} \left( \left[(\rho\underline{u})_{ij}^n.\ \underline{S}_{ij}\right]\ a_{f,ij} - \beta_{ij}\frac{a_{J'}-a_{I'}}{\overline{I'J'}}S_{ij}\right)$$

The above sum corresponds to the numerical operation:

$$\begin{array}{rl} \texttt{FLUX} & = \texttt{ICONVP}\,.\,[\ \texttt{FLUI}\,.\,\texttt{PIF} + \texttt{FLUJ}\,.\,\texttt{PJF}\ ] \\ & + \texttt{IDIFFP}\,.\,\texttt{VISCF(IFAC)}\,.\,[\ \texttt{PIP} - \texttt{PJP}\ ] \end{array} \qquad \text{(II.1.7)}$$

The above equation does not depend on the chosen convective scheme, since the latter only affects the quantities PIF (face value of $a$ used when $b$ is positive) and PJF (face value of $a$ used when $b$ is négative). PIP represents $a_{I'}$, PJP $a_{J'}$ and VISCF(IFAC) $\beta_{ij}\dfrac{S_{ij}}{\overline{I'J'}}$ .
The treatment of diffusive part is identical (either with or without reconstruction). Consequently, only the numerical scheme relative to the convection differs.

- for a boundary face $ik$ (IFAC)

We compute the terms :

$$\sum_{k\in\gamma_b(i)} F_{b_{ik}}((\rho\underline{u})^n,a) - \sum_{k\in\gamma_b(i)} D_{b_{ik}}(\beta,a) = \sum_{k\in\gamma_b(i)} \left( \left[(\rho\underline{u})_{b_{ik}}^n.\ \underline{S}_{b_{ik}}\right]\ a_{f\,b_{ik}} - \beta_{b_{ik}}\frac{a_{b_{ik}}-a_{I'}}{\overline{I'F}}S_{b_{ik}}\right)$$

with:

$$\begin{array}{ll} a_{I'} & = a_I + \underline{II'}.\underline{G}_{c,i} \\ a_{b1_{ik}} & = \texttt{INC}\,A_{b,ik} + B_{b,ik}\,a_{I'} \\ a_{b_{ik}} & = \texttt{INC}\,A_{b,ik}^{diff} + B_{b,ik}^{diff}\,a_{I'} \end{array}$$

The coefficients $(A_{b,ik}, B_{b,ik})_{k\in\gamma_b(i)}$ $\left(\text{ resp.}(A_{b,ik}^{diff}, B_{b,ik}^{diff})_{k\in\gamma_b(i)}\right)$ represent the boundary conditions associated with $a$ (resp. the diffusive fluxes [2] of $a$).
The above sum corresponds to the numerical operation::

$$\begin{array}{rl} \texttt{FLUX} & = \texttt{ICONVP}\,.\,[\ \texttt{FLUI}\,.\,\texttt{PVAR(II)} + \texttt{FLUJ}\,.\,\texttt{PFAC}\ ] \\ & + \texttt{IDIFFP}\,.\,\texttt{VISCB(IFAC)}\,.\,[\ \texttt{PIP} - \texttt{PFACD}\ ] \end{array} \qquad \text{(II.1.8)}$$

where PFAC represents $a_{b1_{ik}}$, PIP $a_{I'}$, PFACD $a_{b_{ik}}$ and VISCB(IFAC) $\beta_{b_{ik}}\dfrac{S_{b_{ik}}}{\overline{I'F}}$.
This treatment is common to all schemes, because boundary values only depend on boundary conditions, and because a very simplified expression of $F_{b_{ik}}$ is used (upwind) [3].

We still have to compute, when the convection option is activated (ICONVP = 1), the values of variables PIF and PJF, for any internal face IFAC between cell $\Omega_i$ and $\Omega_j$.

[2] see clptur for more details. The difference is actually only effective when the $k-\epsilon$ model is used, and for the velocity only.
[3] Actually, $a_{f\,b_{ik}}$ is $a_I$ if $(\rho\underline{u})_{b_{ik}}^n.\,\underline{S}_{b_{ik}} \geqslant 0$, $a_{b1_{ik}}$ otherwise.

## Calculation of the flux in pure upwind IUPWIN = 1

In this case, there is no reconstruction since only the values PVAR(II) and PVAR(JJ) at the cell centres are needed.

$$\begin{aligned} \texttt{PIF} &= \texttt{PVAR(II)} \\ \texttt{PJF} &= \texttt{PVAR(JJ)} \end{aligned} \tag{II.1.9}$$

The variable INFAC counts the number of calculations in pure upwind, in order to be printed in the listing file. In order to obtain the global numerical flux FLUX (convective + diffusive) associated, the following operations are performed :
• calculation of vectors II' and JJ',
• calculation of the face gradient (DPXF, DPYF, DPZF) with the half-sum of the cell gradients $\underline{G}_{c,i}$ et $\underline{G}_{c,j}$,
• calculation of the reconstructed (if necessary) values $a_{I'}$ and $a_{J'}$ (variables PIP and PJP, respectively) given by :

$$a_{K'} = a_K + \texttt{IRCFLP} \,.\, \underline{KK'} \,.\, \frac{1}{2} \left( \underline{G}_{c,i} + \underline{G}_{c,j} \right) \quad \text{K = I et J} \tag{II.1.10}$$

• calculation of the quantities FLUI and FLUJ,
• calculation of the flux FLUX using (II.1.7).
The computation of the sum in SMBR is straight-forward, following (II.1.1) [4] .

## Calculation of the flux with a centered or SOLU scheme (IUPWIN = 0)

The two available second order schemes on orthogonal meshes are the centered scheme and the SOLU scheme.
In both cases, the following operations are performed:
• calculation of the vector II', the array (DIIPFX, DIIPFY, DIIPFZ) and the vector JJ', the array (DJJPFX, DJJPFY, DJJPFZ)
• calculation of the face gradient (DPXF, DPYF, DPZF) haff-sum of the cell gradients $\underline{G}_{c,i}$ and $\underline{G}_{c,j}$,
• calculation of the possibly reconstructed (if IRCFLP = 1) values $a_{I'}$ and $a_{J'}$ (variables PIP and PJP, respectively) given by :

$$a_{K'} = a_K + \texttt{IRCFLP} \,.\, \underline{KK'} \,.\, \frac{1}{2} \left( \underline{G}_{c,i} + \underline{G}_{c,j} \right) \quad \text{K = I and J} \tag{II.1.11}$$

• calculation of FLUI and FLUJ.

- without slope test (ISSTPP = 1)

  - ⋆ with a centered scheme (ISCHCP = 1)

The values of the variables PIF and PJF are equal, and calculated using the weighting coefficient $\alpha_{ij}$ as follows:

$$\begin{aligned} P_{IF} &= \alpha_{ij} \,.\, P_{I'} + (1 - \alpha_{ij}) \,.\, P_{J'} \\ P_{JF} &= P_{IF} \end{aligned} \tag{II.1.12}$$

  - ⋆ with a SOLU scheme (ISCHCP = 0)

After calculating the vectors $\underline{IF}$ and $\underline{JF}$, the values of the variables PIF and PJF are computed as follows:

$$\begin{aligned} P_{IF} &= P_I + \underline{IF} \,.\, \underline{G}_{c,i} \\ P_{JF} &= P_J + \underline{JF} \,.\, \underline{G}_{c,j} \end{aligned} \tag{II.1.13}$$

PIF and PJF are systematically reconstructed in order to avoid using pure upwind, *i.e.* this formulae is applied even when the user chooses not to reconstruct (IRCFLP = 0).

- with slope test (ISSTPP = 0)

[4] taking into account the negative sign of $\mathcal{B}_\beta$.

The procedure is quite similar to the one described in the previous paragraph. There is, in addition to hte previous procedure, a slope test that makes under certain conditions the scheme switch locally (but systematically) from the chosen centered or SOLU scheme to a pure upwind scheme.

$\rightsquigarrow$ calculation of the slope test

Equation (II.1.6) writes on an internal cell $\Omega_i$, with $s_{ij}^n = sgn\left[(\rho \underline{u})_{ij}^n \,.\, \underline{S}_{ij}\right]$ :

$$\begin{aligned}
|\Omega_i|\,\underline{G}_{c,i}^{upwind} &= \sum_{j\in Vois(i)} a_{ij}^{upwind}\,\underline{S}_{ij} \\
&= \sum_{j\in Vois(i)} \left[\frac{1}{2}(\ s_{ij}^n + 1\ ) \quad a_{IF} + \frac{1}{2}(\ s_{ij}^n - 1\ )\, a_{JF}\right]\ \underline{S}_{ij} \\
&= \sum_{j\in Vois(i)} \left[\frac{1}{2}(\ s_{ij}^n + 1\ ) \quad (\ a_I + \underline{IF}\,.\,(\underline{\mathrm{grad}}\ a)_I\ )\right. \\
&\qquad\qquad \left. +\frac{1}{2}\ (\ s_{ij}^n - 1\ )(\ a_I + \underline{JF}\,.\,(\underline{\mathrm{grad}}\ a)_J\ )\right]\ \underline{S}_{ij}
\end{aligned}$$

On a cell $\Omega_i$ with neighbours $(\Omega_j)_{j\in Vois(i)}$, the classic slope test consists in locating where a variable $a$ is non-monotonic by studying the sign of the scalar product of the cell gradients of $\underline{G}_{c,i}$ and $\underline{G}_{c,j}$. If this product is negative, we switch to an upwind scheme, if it is positive, we use a centered or SOLU scheme.

Another technique which also ensures the monotonicity of the solution is to apply this criterion to the upwind gradients $\underline{G}_{c,k}^{amont}$ or to their normal projection on face $(\underline{G}_{c,k}^{amont} \,.\, \underline{S}_{kl}\,)$.

We then study the sign of the product $\underline{G}_{c,i}^{amont} \,.\, \underline{G}_{c,j}^{amont}$ or of the product $(\underline{G}_{c,i}^{amont} \,.\, \underline{S}_{ij}\,) \,.\, (\underline{G}_{c,j}^{amont} \,.\, \underline{S}_{ij}\,)$.

The slope test implemented is based on the first quantity, $\underline{G}_{c,i}^{amont} \,.\, \underline{G}_{c,j}^{amont}$ (the second one was abandonned because it was found to be less general). The choice of a slope test based on $\underline{G}_{c,i}^{amont} \,.\, \underline{G}_{c,j}^{amont}$ comes from the following line of argument in one-dimension [5]:

Let's take $p$ a second order in $x$ polynomial function. Its value at points $I-1$, $I$, $I+1$ of coordinates $x_{I-1}$, $x_I$ and $x_{I+1}$ are $p_{I-1}$, $p_I$, and $p_{I+1}$, respectively. To simplify, we suppose that $I$ is the origin $O$ ( $x_I = 0$ ), and that the grid spacing $h$ is constant, which results in $x_{I+1} = -x_{I-1} = h$. Additionally, we suppose that the velocity is orientated from point $I$ towards point $I+1$, *i.e.* $s_{ij}^n = 1$. Therefore we consider the points $I-1$, $I$ and $I+1$ for the face $ij$ which is located between $I$ and $I+1$.

The sign of the product $p'(x_{I-1})\,.\,p'(x_{I+1})$ inidicates the monotonicity of function $p$. If this product it positive, the function is monotonic and we use a centered or a SOLU scheme, otherwise, we switch to an upwind scheme. By identifying the polynomial coefficients using the equations $p\,(x_{I-1}) = p_{I-1}$ , $p\,(x_I) = p_I$ , $p\,(x_{I+1}) = p_{I+1}$ , we obtain :

$$\begin{aligned}
p'(x_{I-1}) &= +\frac{p_{I+1} - p_{I-1}}{2h} \quad + \left[\frac{p_I - p_{I-1}}{h} - \frac{p_{I+1} - p_I}{h}\right] \\
p'(x_{I+1}) &= +\frac{p_{I+1} - p_{I-1}}{2h} \quad - \left[\frac{p_I - p_{I-1}}{h} - \frac{p_{I+1} - p_I}{h}\right]
\end{aligned} \tag{II.1.14}$$

or after simplification :

$$\begin{aligned}
p'(x_{I-1}) &= G_{c,i} + \left(G_{c,i}^{amont} - \frac{p_{I+1} - p_I}{h}\right) \\
p'(x_{I+1}) &= G_{c,i} - \left(G_{c,i}^{amont} - \frac{p_{I+1} - p_I}{h}\right)
\end{aligned} \tag{II.1.15}$$

We know that :

♣ $\dfrac{p_{I+1} - p_I}{h}$ represente the upwind derivative at point $I+1$, directly accessible by the values of $p$ in the neighbouring cells of face $ij$,

♣ $\dfrac{p_{I+1} - p_{I-1}}{2h}$ represents the centered derivative (in finite volume) at point $I$, namely $G_{c,i}$,

[5] Information on the second derivative would permit to study more finely the behaviour and the strong variations of $a$.

♣ $\dfrac{p_I - p_{I-1}}{h}$ represents the value of the upwind derivative (in finite volume) at point $I$, namely $G_{c,i}^{amont}$.
The slope test relative to $p'(x_{I-1})\,.\,p'(x_{I+1})$ reduces to studying the sign of $\mathcal{TP}_{1d}$ :

$$\begin{aligned}\mathcal{TP}_{1d} &= \left(G_{c,i} \;+\; [\, G_{c,i}^{amont} - \frac{p_{I+1}-p_I}{h}]\right).\left(G_{c,i} \;-\; [\, G_{c,i}^{amont} - \frac{p_{I+1}-p_I}{h}]\right) \\ &= |G_{c,i}|^2 - (\, G_{c,i}^{amont} - \frac{p_{I+1}-p_I}{h})^2\end{aligned} \tag{II.1.16}$$

Using a similar line of argument, a possible extension to higher dimensions consists in replacing the values $G_{c,k}$ and $G_{c,k}^{amont}$ by $(\underline{G}_{c,k}\,.\,\underline{S}_{kl}\,)$ eand $(\underline{G}_{c,k}^{amont}\,.\,\underline{S}_{kl}\,)$ respectively. After simplifications, this leads us to the formulae $\mathcal{TP}_{3d}^{+}$ :

$$\mathcal{TP}_{3d}^{+} = (\underline{G}_{c,i}\,.\,\underline{S}_{ij})^2 - (\underline{G}_{c,i}^{amont}\,.\,\underline{S}_{ij} - \frac{a_J - a_I}{\overline{I'J'}}S_{ij})^2 \tag{II.1.17}$$

for $(\rho\underline{u})_{ij}^n.\ \underline{S}_{ij} > 0$.
Similarly, we can deduce a $\mathcal{TP}_{3d}^{-}$ associated with $(\rho\underline{u})_{ij}^n.\ \underline{S}_{ij} < 0$, defined by :

$$\mathcal{TP}_{3d}^{-} = (\underline{G}_{c,j}\,.\,\underline{S}_{ij})^2 - (\underline{G}_{c,j}^{amont}\,.\,\underline{S}_{ij} - \frac{a_J - a_I}{\overline{I'J'}}S_{ij})^2 \tag{II.1.18}$$

We introduce the variables `TESTI`, `TESTJ` and `TESTIJ` computed as:

$$\begin{aligned}\texttt{TESTI} &= \underline{G}_{c,i}^{amont}\,.\,\underline{S}_{ij} \\ \texttt{TESTJ} &= \underline{G}_{c,j}^{amont}\,.\,\underline{S}_{ij} \\ \texttt{TESTIJ} &= \underline{G}_{c,i}^{amont}\,.\,\underline{G}_{c,j}^{amont}\end{aligned} \tag{II.1.19}$$

The quantity `TESQCK` corresponding to $\mathcal{TP}_{3d}$, is computed dynamically, depending on the sign of the mass flux $s_{ij}^n$.

$\rightsquigarrow$ consequently :

$\diamond$ if $(\rho\underline{u})_{ij}^n.\ \underline{S}_{ij} > 0$ and

if $\underbrace{(\underline{G}_{c,i}\,.\,\underline{S}_{ij})^2 - (\underline{G}_{c,i}^{amont}\,.\,\underline{S}_{ij} - \frac{a_J - a_I}{\overline{I'J'}}S_{ij})^2}_{\texttt{TESQCK}} < 0$ or $(\underline{G}_{c,i}^{amont}\,.\,\underline{G}_{c,j}^{amont}) < 0$,

or :
if $(\rho\underline{u})_{ij}^n.\ \underline{S}_{ij} < 0$ and

if $\underbrace{(\underline{G}_{c,j}\,.\,\underline{S}_{ij})^2 - (\underline{G}_{c,j}^{amont}\,.\,\underline{S}_{ij} - \frac{a_J - a_I}{\overline{I'J'}}S_{ij})^2}_{\texttt{TESQCK}} < 0$ or $(\underline{G}_{c,i}^{amont}\,.\,\underline{G}_{c,j}^{amont}) < 0$,

then we switch to a pure upwind scheme:

$$\begin{aligned}\texttt{PIF} &= \texttt{PVAR(II)} \\ \texttt{PJF} &= \texttt{PVAR(JJ)}\end{aligned} \tag{II.1.20}$$

and `INFAC` is incremented.

$\diamond$ otherwise :
the centered or the SOLU scheme values values are used as before :

$\star$ with a centered scheme (`ISCHCP` $= 1$)
The values of the variables `PIF` and `PJF` are equal and calculated using the weighting coefficient $\alpha_{ij}$ :

$$\begin{aligned}P_{IF} &= \alpha_{ij}\,.\,P_{I'} + (1-\alpha_{ij})\,.\,P_{J'} \\ P_{JF} &= P_{IF}\end{aligned} \tag{II.1.21}$$

⋆ with a SOLU scheme (ISCHCP = 0)

After calculating the vectors $\underline{IF}$ and $\underline{JF}$, the values of the variables PIF and PJF are computed as follows:

$$\begin{aligned} P_{IF} &= P_I + \underline{IF}\,.\,\underline{G}_{c,i} \\ P_{JF} &= P_J + \underline{JF}\,.\,\underline{G}_{c,j} \end{aligned} \tag{II.1.22}$$

PIF and PJF are systematically reconstructed in order to avoid using pure upwind, *i.e.* this formulae is applied even when the user chooses not to reconstruct (IRCFLP = 0).

Wether the slope test is activated or not, when the centered or the SOLU schemes are activated, a blending coefficient (BLENCP) between 0 and 1, provided by the user, enables to blend, if desired, the chosen scheme and the pure upwind scheme following the formulae:

$$\begin{aligned} P_{IF} &= \texttt{BLENCP} P_{IF}^{(centre\ ou\ SOLU)} + (1 - \texttt{BLENCP})\ P_{II} \\ P_{JF} &= \texttt{BLENCP} P_{JF}^{(centre\ ou\ SOLU)} + (1 - \texttt{BLENCP})\ P_{JJ} \end{aligned} \tag{II.1.23}$$

• calculation of FLUI and FLUJ,
• calculation of the flux FLUX using equation (II.1.7).
The computation of the sum in SMBR is straight-forward, following (II.1.1)[6]

REMARK

For more information on the convection schemes and the slope test in *Code_Saturne* (version 1.1), the reader is referred to EDF internal report EDF HI-83/04/020 (F. Archambeau, 2004).

[6] taking into account the negative sign of $\mathcal{B}_\beta$.

## 1.4 Points to treat

• **Convection scheme**

⇝ Upwind scheme

As all first-order schemes, it is robust, but introduces severe numerical diffusion.

⇝ Centered or SOLU scheme

This type of schemes can generate numerical oscillations, that can cause the calculation to blow up. It can also lead to physical scalars taking unphysical values.

Considering these limitations, other schemes are currently being tested and implemented in order to improve the quality of the schemes available to the users.

• **Diffusion scheme**

The formulae :

$$D_{ij}(\beta, a) = \beta_{ij} \frac{a_{J'} - a_{I'}}{\overline{I'J'}} S_{ij} \tag{II.1.24}$$

is second-order accurate only for $\alpha_{ij} = \frac{1}{2}$. A possible correction may be to write :

$$\underline{G}_{f,ij} \cdot \underline{S}_{ij} = (\underline{\text{grad}}\ a)_{ij} = \frac{a_{J'} - a_{I'}}{\overline{I'J'}} \cdot \underline{S}_{ij} + (\frac{1}{2} - \alpha_{ij}) [(\underline{\text{grad}}\ a)_{I'} - (\underline{\text{grad}}\ a)_{J'}] \cdot \underline{S}_{ij} \tag{II.1.25}$$

with a gradient limiter and a computation of $\beta_{ij}$ which does not alter the order of accuracy.

• **Implementation**

In order to improve the CPU time, an effort on loops can be done. More particularly, there is a test `IF` inside of a loop on variable `IFAC` that needs to be checked.

• **Calculation of the gradient used during the reconstruction of the diffusive fluxes**

Why do we use $\frac{1}{2}(\underline{G}_{c,i} + \underline{G}_{c,j})$ instead of $\underline{G}_{c,k}$, for $k = i$ or for $k = j$ in the reconstructed values $a_{I'}$ or $a_{J'}$ of (II.1.10) and (II.1.11)?

# 2- Sous-programme clptur

## 2.1 Function

This subroutine is dedicated to the calculation of the wall boundary conditions. The notations introduced in CONDLI for the general boundary conditions will be used.

The wall boundary conditions refer to all the boundary conditions for the velocity, the turbulent variables ($k$, $\varepsilon$, $R_{ij}$), the temperature when it has a prescribed value at the wall (or the enthalpy and more generally the *VarScalaires*[1] to treat at the wall by using a similarity law for the associated boundary layer). For the *VarScalaire* in particular, when the boundary conditions at the wall are of Neumann type (homogeneous or not), they are treated in condli and don't present them here. In particular, the boundary conditions of the *VarScalaires* are not treated here because their treatment at the wall is of homogeneous Neumann type.

We present the calculation of the pair of coefficients $A_b$ and $B_b$ which are used during the computation of certain discretized terms of the equations to solve, and which allow in particular to determine a value associated with the boundary faces $f_{b,int}$ (at a point located at the "centre" of the boundary face, the barycentre of its vertices) using the formulae $f_{b,int} = A_b + B_b\, f_{I'}$ ($f_{I'}$ is the value of the variable at point $I'$, the projection of the centre of the boundary cell onto the line normal to the boundary face and passing through its centre : see figure II.2.1).

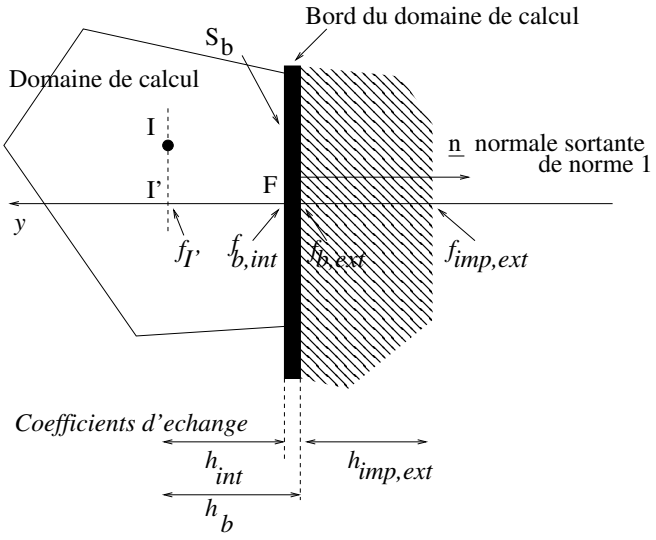



Figure II.2.1: Boundary cell.

## 2.2 Discretisation

• **Notations**

[1] As in condli the *VarScalaire* are any solution of a convection-diffusion equation apart from the velocity, the pressure and the turbulent variables $k$, $\varepsilon$ and $R_{ij}$. More specifically, the name *VarScalaire* can refer to the temperature, the enthalpy or a passive scalar.

The velocity of the wall is noted $\underline{v}_p$. We assume it is projected onto the plane tangent to the wall (if it is not, then the code projects it).

The velocity of the fluid is noted $\underline{u}$. Index $I$, $I'$ or $F$ denotes the point at which the velocity is estimated. The component tangent to the wall writes $u_\tau$. The fluid velocity in the coordinate system attached to the wall ("relative" velocity) writes $\underline{u}^r = \underline{u} - \underline{v}_p$.

The orthonormal coordinate system attached to the wall writes $\hat{\mathcal{R}} = (\underline{\tau}, \underline{\tilde{n}}, \underline{b})$.

- $\underline{\tilde{n}} = -\underline{n}$ is the unit vector orthogonal to the wall and directed towards the interior of the computational domain.
- $\underline{\tau} = \dfrac{1}{\|\underline{u}^r_{I'} - (\underline{u}^r_{I'} \,.\, \underline{\tilde{n}})\|} \left[\underline{u}^r_{I'} - (\underline{u}^r_{I'} \,.\, \underline{\tilde{n}})\right]$ is the unit vector parallel to the projection of the relative velocity at $I'$, $\underline{u}^r_{I'}$, in the plane tangent to the wall (*i.e.* orthogonal to $\underline{\tilde{n}}$) : see figure II.2.1.
- $\underline{b}$ is the unit vector which completes the positively oriented coordinate system.

The dimensionless limit distance which separates the viscous sublayer from the logarithmic region writes $y^+_{lim}$. Its value is $1/\kappa$ (with $\kappa = 0,42$) in general (to ensure the continuity of the velocity gradient) and 10.88 in LES (to ensure the continuity of the velocity).

In the case of the **two velocity scale model**,

- $u_k$ is the friction velocity at the wall obtained from the turbulent kinetic energy. We write $u^*$ the friction velocity at the wall calculated from the equation $\dfrac{u^r_{\tau,I'}}{u^*} = f(y^+_k)$.
- $y^+_k$ represents a dimensionless wall distance, $y^+_k = \dfrac{u_k\, I'F}{\nu}$ ($\nu$ is the molecular kinematic viscosity taken at the centre $I$ of the boundary cell). The function $f$ gives the ideal shape of the velocity profile. It is piecewisely approximated by the logarithmic law $f(z) = f_1(z) = \dfrac{1}{\kappa} ln(z) + 5,2$ for $z > y^+_{lim}$ and by the linear law $f(z) = f_2(z) = z$ otherwise.
- The two velocity scale $u_k$ and $u^*$ are simple to compute but their computation requires the knowledge of the turbulent kinetic energy $k_I$ at the centre of cell adjoint to the boundary face (with the $R_{ij} - \varepsilon$ model, we use half the trace of the Reynolds stress tensor).
- The two velocity scale model is the default model in *Code_Saturne*. It often permits, and in particular in cases with heat transfer, to reduce the effects of certain flaws associated to the $k-\varepsilon$ model.

Later on, we will use $u^*$ and $u_k$ for the boundary conditions of the velocity and scalars (in particular the temperature).

**Two velocity scale model**

$$\left\{ \begin{array}{l} u_k = C_\mu^{\frac{1}{4}} k_I^{\frac{1}{2}} \\ u^* \text{is solution of } \left\{ \begin{array}{lll} \dfrac{u^r_{\tau,I'}}{u^*} & = \dfrac{1}{\kappa} ln(y^+_k) + 5,2 & \text{for } y^+_k > y^+_{lim} \\ \dfrac{u^r_{\tau,I'}}{u^*} & = y^+_k & \text{for } y^+_k \leqslant y^+_{lim} \end{array} \right. \\ \qquad \text{with } C_\mu = 0,09 \qquad y^+_k = \dfrac{u_k\, I'F}{\nu} \text{ and } \kappa = 0,42 \end{array} \right. \tag{II.2.1}$$

In the case of the **one velocity scale model**,

we write $u^*$ the only friction velocity at the wall solution of the equation $\dfrac{u^r_{\tau,I'}}{u^*} = f(y^+)$. $y^+$ represents a dimensionless wall distance $y^+ = \dfrac{u^*\, I'F}{\nu}$ ($\nu$ is the molecular kinematic viscosity taken at the centre $I$

of the boundary cell). The function $f$ gives the ideal shape of the velocity profile, as in the case of the two velocity scale model. One can note that this friction velocity, calculated using a more complicated approach (Newton method), can however be obtained without making any reference to the turbulent variables ($k$, $\varepsilon$, $R_{ij}$). For convenience in the case of the one velocity scale model, we write $u_k = u^*$.

Later on, we will use $u^*$ and $u_k$ for the boundary conditions of the velocity and scalars (in particular the temperature).

**Modèle à une échelle de vitesse**

$$\left\{ \begin{array}{l} u_k = u^* \\ u^* \text{ solution de } \left\{ \begin{array}{ll} \dfrac{u^r_{\tau,I'}}{u^*} & = \dfrac{1}{\kappa} ln(y^+) + 5,2 \quad \text{pour } y^+ > y^+_{lim} \\ \dfrac{u^r_{\tau,I'}}{u^*} & = y^+ \qquad\qquad\quad \text{pour } y^+ \leqslant y^+_{lim} \end{array} \right. \\ \qquad \text{avec } y^+ = \dfrac{u^* \, I'F}{\nu} \text{ et } \kappa = 0,42 \end{array} \right. \tag{II.2.2}$$

**Remark :** Note that the user subroutine `usruet` permits to modify the value of the velocity scales. Hereafter, we provide three exemples based on the two velocity scale model.

- In this way, one can implement a specific wall function :

  $$\frac{u_{\tau,I'}}{u^*} = g(y^+)$$

  by simply imposing $u^* = u_{\tau,I'}/g(y^+)$ (the values of $u_{\tau,I'}$ and $y^+$ are available in the arguments of `usruet`).

- It is also possible to use a rough-wall wall function such as :

  $$\frac{u_{\tau,I'}}{u^*} = \frac{1}{\kappa} \, ln(\frac{y}{\xi}) + 8,5$$

  where $\xi$ is the height of the roughness elements at the wall : one just has to impose $u^* = u_{\tau,I'} / \left[ \dfrac{1}{\kappa} ln(\dfrac{y}{\xi}) + 8,5 \right]$, $y$ being deduced from $y^+$, available as an argument, by the equation $y = y^+ \dfrac{\nu}{u_k}$.

- Even a more general correlation could be used of Colebrook type :

  $$u^* = u_{deb} / \left[ -4\sqrt{2} log_{10} \left( \frac{2,51}{2\sqrt{2} D^+_H} + \frac{\xi}{3,7 \, D_H} \right) \right]$$

  where $D^+_H$ is the hydraulic diameter made dimensionless using $u_k$, $\nu$, $u_{deb}$ the mean streamwise velocity and $\dfrac{\xi}{D_H}$ the relative roughness.

**• Boundary conditions for the velocity in $k - \varepsilon$**

We first consider the boundary conditions used in the case of calculation using the $k - \varepsilon$ model. Indeed these cases are the most complex and general.

The boundary conditions are necessary to prescribe at the boundary the correct tangential stress $\sigma_\tau = \rho_I u^* u_k$ in the momentum equation[2] ($\rho_I$ is the density at the centre of cell $I$). The term which requires boundary conditions is the one containing the velocity derivative in the normal direction to

[2] Proposition de modification des conditions aux limites de paroi turbulente pour le Solveur Commun dans le cadre du modèle $k - \varepsilon$ standard, rapport EDF HI-81/00/019/A, 2000, M. Boucker, J.-D. Mattei.

the wall[3] : $(\mu_I + \mu_{t,I})\underline{\underline{\text{grad}}}\ \underline{u}\,\underline{n}$. It appears on the right-hand side of the usual momentum equation (see `bilsc2` and `preduv`).

In the case where the $k-\varepsilon$ model tends to surestimate the production of turbulent kinetic energy, the length scale of the model, $L_{k-\varepsilon}$, can become significantly larger than the maximum theoretical length scale of the turbulent boundary layer eddies $L_{\text{theo}}$. We write :

$$\left\{\begin{array}{l} L_{k-\varepsilon} = C_\mu \dfrac{k^{\frac{3}{2}}}{\varepsilon} \\ L_{\text{theo}} = \kappa\, I'F \end{array}\right. \tag{II.2.3}$$

In the case where $L_{k-\varepsilon} > L_{\text{théo}}$, we thus have $\mu_{t,I} > \mu_t^{lm}$ with $\mu_{t,I}$ the turbulent viscosity of the $k-\varepsilon$ model at point $I$ and $\mu_t^{lm} = \rho_I L_{\text{theo}} u_k$ the turbulent viscosity of the mixing length model. Additionally, the tangential stress can write by making the turbulent viscosity appear :

$$\sigma_\tau = \rho_I u^* u_k = \frac{u^*}{\kappa\, I'F} \underbrace{\rho_I \kappa\, I'F\, u_k}_{\mu_t^{lm}} \tag{II.2.4}$$

The viscosity scale introduced in the stress thus contradicts the one deduced from the neighbouring turbulence calculated by the model. Consequently we prefer to write the stress, by using the velocity scale of the $k-\varepsilon$ model when it is lower than the limit $L_{\text{théo}}$ :

$$\sigma_\tau = \frac{u^*}{\kappa\, I'F} max(\mu_t^{lm}, \mu_{t,I}) \tag{II.2.5}$$

One can then use this value to calculate the diffusive flux which depends upon it in the Navier-Stokes equations :

$$(\mu_I + \mu_{t,I})\underline{\underline{\text{grad}}}\ \underline{u}\,\underline{n} = -\sigma_\tau \underline{\tau}. \tag{II.2.6}$$

But the velocity gradient (face gradient) is computed in the code as :

$$(\mu_I + \mu_{t,I})\underline{\underline{\text{grad}}}\ \underline{u}\,\underline{n} = \frac{(\mu_I + \mu_{t,I})}{\overline{I'F}} (\underline{u}_F - \underline{u}_{I'}) \tag{II.2.7}$$

Using (II.2.6) and (II.2.7) we obtain the value of $\underline{u}_F$ to be prescribed, referred to as $\underline{u}_{F,flux}$ (conservation of the momentum flux) :

$$\begin{array}{rl} \underline{u}_{F,flux} & = \underline{u}_{I'} - \dfrac{\overline{I'F}}{\mu_I + \mu_{t,I}} \sigma_\tau \underline{\tau} \\ & = \underline{u}_{I'} - \dfrac{u^*}{\kappa\,(\mu_I + \mu_{t,I})} max(\mu_t^{lm}, \mu_{t,I}) \underline{\tau} \end{array} \tag{II.2.8}$$

In reality, an extra approximation is made. It consists in imposing a zero normal velocity at the wall and in using equation (II.2.8) projected on the plane parallel to the wall :

$$\underline{u}_{F,flux} = \left[ u_{\tau,I'} - \frac{u^*}{\kappa\,(\mu_I + \mu_{t,I})} max(\mu_t^{lm}, \mu_{t,I}) \right] \underline{\tau} \tag{II.2.9}$$

Moreover, if the value obtained for $y^+$ is lower than $y^+_{lim}$ a no-slip condition is applied. Finally, one can also make the wall velocity appear in the final expression :

**"Flux" boundary conditions of the velocity** $(k-\varepsilon)$

$$\left\{\begin{array}{lll} \underline{u}_{F,flux} & = \underline{v}_p & \text{if } y^+ \leqslant y^+_{lim} \\ \underline{u}_{F,flux} & = \underline{v}_p + \left[ u^r_{\tau,I'} - \dfrac{u^*}{\kappa\,(\mu_I + \mu_{t,I})} max(\mu_t^{lm}, \mu_{t,I}) \right] \underline{\tau} & \text{otherwise} \end{array}\right. \tag{II.2.10}$$

[3] The transpose gradient term is treated in `vissec` and thus will not be considered here.

A first pair of coefficients $A_{flux}$ and $B_{flux}$ can then be deduced (for each component of the velocity separately) and it is used only to compute the tangential stress dependent term (see `bilsc2`) :

**Coefficients associated with the "flux" boundary conditions of the velocity** $(k-\varepsilon)$

$$\left\{ \begin{array}{l} \left\{ \begin{array}{ll} \underline{A}_{flux} & = \underline{v}_p \qquad \text{if } y^+ \leqslant y^+_{lim} \\ \underline{A}_{flux} & = \underline{v}_p + \left[ u^r_{\tau,I'} - \dfrac{u^*}{\kappa\,(\mu_I+\mu_{t,I})} max(\mu_t^{lm}, \mu_{t,I}) \right] \underline{\tau} \text{ otherwise} \end{array} \right. \\ \underline{B}_{flux} = \underline{0} \end{array} \right. \tag{II.2.11}$$

We saw above how to impose a boundary condition to compute directly the stress term. Further analysis is necessary to calculate correctly the velocity gradients. We want to find a boundary face value which permits to obtain, with the chosen expression for the gradient, the value the turbulent production as close as possible to its theoretical value (determined by using the logarithmic law), in order to evaluate the normal derivative the tangential velocity. Thus, we define (at point $I$) :

$$P_{\text{théo}} = \rho_I u^* u_k \| \frac{\partial u_\tau}{\partial \underline{n}} \|_I = \rho_I \frac{u_k (u^*)^2}{\kappa\, I'F} \tag{II.2.12}$$

Morevoer, the dominant term of the production computed in cell $I$ is, in classical situations ($y$ is the coordinate on the axis whose direction vector is $\underline{\tilde{n}}$),

$$P_{\text{calc}} = \mu_{t,I} \left( \frac{\partial u_\tau}{\partial y} \right)^2_I \tag{II.2.13}$$

The normal gradient of the tangential velocity (cell gradient) is calculated in the code using finite volume, and its expression on regular orthogonal meshes is (see the notations on figure II.2.2) :

$$P_{\text{calc}} = \mu_{t,I} \left( \frac{u_{\tau,G} - u_{\tau,F}}{2d} \right)^2 = \mu_{t,I} \left( \frac{u_{\tau,I} + u_{\tau,J} - 2u_{\tau,F}}{4d} \right)^2 \tag{II.2.14}$$

We then assume that $u_{\tau,J}$ can be obtained from $u_{\tau,I}$ and from the normal gradient of $u_\tau$ calculated in G from the logarithmic law :

$$u_{\tau,J} = u_{\tau,I} + IJ\,.\,(\partial_y u_\tau)_G + \mathcal{O}(IJ^2) \approx u_{\tau,I} + IJ\,.\,\left[ \partial_y \left( \frac{u^*}{\kappa}\, ln(y^+) + 5,2 \right) \right]_G = u_{\tau,I} + 2d \frac{u^*}{\kappa\, 2d} \tag{II.2.15}$$

and thus we obtain :

$$\begin{array}{rcl} P_{\text{calc}} & = & \mu_{t,I} \left( \dfrac{u_{\tau,I} + u_{\tau,I} + 2d \frac{u^*}{\kappa\, 2d} - 2u_{\tau,F}}{4d} \right)^2 \\ & = & \mu_{t,I} \left( \dfrac{2u_{\tau,I} + 2\frac{u^*}{2\kappa} - 2u_{\tau,F}}{4d} \right)^2 = \mu_{t,I} \left( \dfrac{u_{\tau,I} + \frac{u^*}{2\kappa} - u_{\tau,F}}{2d} \right)^2 \end{array} \tag{II.2.16}$$

We then use (II.2.12) and (II.2.16) to impose that the calculated production is equal to the theoretical production. The preceeding formulae are extended with no precaution to non-orthogonal meshes (the velocity at $I$ is then simply computed at $I'$). The following expression for $u_{\tau,F}$ is then obtained :

$$u_{\tau,F,grad} = u_{\tau,I'} - \frac{u^*}{\kappa} \left( 2 \sqrt{\frac{\rho_I \kappa\, u_k I'F}{\mu_{t,I}}} - \frac{1}{2} \right) \tag{II.2.17}$$

Additionally, we force the gradient to remain as stiff as the one given by the normal derivative of the theoretical velocity profile (logarithmic) at $I'$ :
$\partial_y u_\tau = \partial_y (\frac{u^*}{\kappa}\, ln(y^+) + 5,2) = \frac{u^*}{\kappa\, \overline{I'F}}$, thus :

$$u_{\tau,F,grad} = u_{\tau,I'} - \frac{u^*}{\kappa} max \left( 1, 2 \sqrt{\frac{\rho_I \kappa\, u_k I'F}{\mu_{t,I}}} - \frac{1}{2} \right) \tag{II.2.18}$$

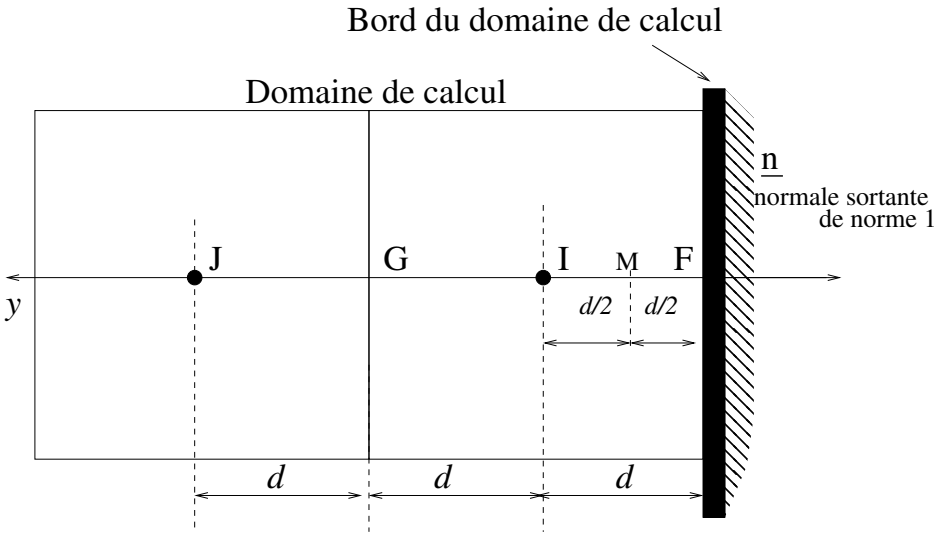



Figure II.2.2: Cellule de bord - Maillage orthogonal.

Finally, we clip the velocity at the wall with a minimum value calculated assuming that we are in the logarithmic layer :

$$u_{\tau,F,grad} = max\left(u^*\left(\frac{1}{\kappa}ln(y^+_{lim})+5,2\right), u_{\tau,I'} - \frac{u^*}{\kappa}\left[max\left(1, 2\sqrt{\frac{\rho_I \kappa\, u_k I'F}{\mu_{t,I}}} - \frac{1}{2}\right)\right]\right) \qquad \text{(II.2.19)}$$

The normal derivative at the wall is prescribed to zero. If the $y^+$ value at the wall is lower than $y^+_{lim}$, a no-slip condition is prescribed. Finally, one can also make explicit the velocity of the wall in the final expression :

**"Gradient" boundary conditions of the velocity**$(k-\varepsilon)$

$$\begin{cases} \underline{u}_{F,grad} = \underline{v}_p & \text{if } y^+ \leqslant y^+_{lim} \\ \underline{u}_{F,grad} = \underline{v}_p + & \\ \left\{ max\left(u^*\left(\frac{1}{\kappa}ln(y^+_{lim})+5,2\right), u^r_{\tau,I'} - \frac{u^*}{\kappa}\left[max\left(1, 2\sqrt{\frac{\rho_I \kappa\, u_k I'F}{\mu_{t,I}}} - \frac{1}{2}\right)\right]\right) \right\} \underline{\tau} & \text{otherwise} \end{cases} \qquad \text{(II.2.20)}$$

A second pair of coefficients $A_{grad}$ and $B_{grad}$ can then be deduced (for each velocity component separately). It is used when the velocity gradient is necessary (except for the terms depending on the tangential shear, those being treated in `bilsc2` using $A_{flux}$ and $B_{flux}$) :

**Coefficients associated to the "gradient" boundary conditions of the velocity**$(k-\varepsilon)$

$$\begin{cases} \begin{cases} \underline{A}_{grad} = \underline{v}_p & \text{if } y^+ \leqslant y^+_{lim} \\ \underline{A}_{grad} = \underline{v}_p + & \\ \left\{ max\left(u^*\left(\frac{1}{\kappa}ln(y^+_{lim})+5,2\right), u^r_{\tau,I'} - \frac{u^*}{\kappa}\left[max\left(1, 2\sqrt{\frac{\rho_I \kappa\, u_k I'F}{\mu_{t,I}}} - \frac{1}{2}\right)\right]\right) \right\} \underline{\tau} & \text{otherwise} \end{cases} \\ \underline{B}_{grad} = \underline{0} \end{cases} \qquad \text{(II.2.21)}$$

• **Boundary conditions of the velocity in** $R_{ij}-\varepsilon$

The boundary conditions of the velocity with the $R_{ij}-\varepsilon$ model are more simple, since there are only of one type. Keeping the same notations as above, we want the tangential velocity gradient (calculated

at $I$, and to be used to evaluate the turbulent production) to be consistent with the logarithmic law giving the ideal tangential velocity profile. The theoretical gradient is :

$$G_{\text{theo}} = \left(\frac{\partial u_\tau}{\partial y}\right)_{I'} = \frac{u^*}{\kappa\, I'F} \tag{II.2.22}$$

The normal gradient of the tangential velocity (cell gradient) is calculated in the code using finite volumes, and its expression in the case of regular orthogonal meshes is (see notations in figure II.2.2) :

$$G_{\text{calc}} = \frac{u_{\tau,G} - u_{\tau,F}}{2d} = \frac{u_{\tau,I} + u_{\tau,J} - 2u_{\tau,F}}{4d} \tag{II.2.23}$$

We then assume that $u_{\tau,J}$ can be obtained from $u_{\tau,I}$ and from the normal gradient of $u_\tau$ calculated in G from the logarithmic law (see equation (II.2.15)) $u_{\tau,J} = u_{\tau,I} + 2d\dfrac{u^*}{\kappa\, 2d}$ and we thus obtain :

$$G_{\text{calc}} = \frac{u_{\tau,I} + u_{\tau,I} + 2d\dfrac{u^*}{\kappa\, 2d} - 2u_{\tau,F}}{4d} = \frac{2u_{\tau,I} + 2\dfrac{u^*}{2\kappa} - 2u_{\tau,F}}{4d} = \frac{u_{\tau,I} + \dfrac{u^*}{2\kappa} - u_{\tau,F}}{2d} \tag{II.2.24}$$

We then use the equations (II.2.22) and (II.2.24) to derive an expression for $u_{\tau,F}$ (the preceeding formulae are extended with no precaution to the case non-orthogonal meshes, the velocity at $I$ being simply computed at $I'$) :

$$u_{\tau,F} = u_{\tau,I'} - \frac{3u^*}{2\kappa} \tag{II.2.25}$$

The normal derivative at the wall is prescribed to zero. If the value obtained for $y^+$ at the wall is lower than $y^+_{lim}$, a no-slip condition is prescribed. Finally, one can also make explicit the velocity of the wall in the final expression :

**Boundary conditions of the velocity** $(R_{ij} - \varepsilon)$

$$\begin{cases} \underline{u}_F = \underline{v}_p & \text{if } y^+ \leqslant y^+_{lim} \\ \underline{u}_F = \left[u^r_{\tau,I'} - \dfrac{3u^*}{2\kappa}\right]\underline{\tau} + \underline{v}_p & \text{otherwise} \end{cases} \tag{II.2.26}$$

Un couple de coefficients $A$ et $B$ s'en déduit (pour chaque composante de vitesse séparément) :

**Coefficients associés aux conditions aux limites sur la vitesse** $(R_{ij} - \varepsilon)$

$$\begin{cases} \begin{cases} \underline{A} = \underline{v}_p & \text{si } y^+ \leqslant y^+_{lim} \\ \underline{A} = \left[u^r_{\tau,I'} - \dfrac{3u^*}{2\kappa}\right]\underline{\tau} + \underline{v}_p & \text{sinon} \end{cases} \\ \underline{B} = \underline{0} \end{cases} \tag{II.2.27}$$

A pair of coefficients $A_{grad}$ and $B_{grad}$ can be deduced from the above equation (for each velocity component separately).

### • Boundary conditions of the velocity in laminar

When no turbulence model is activated, we implicitly use a one velocity scale model (there is no turbulent variables to compute $u_k$), and the same conditions [4] as in $R_{ij} - \varepsilon$ are used : the model degenerates automatically.

[4] In other words; the boundary conditions are given by (II.2.26) and (II.2.27).

• **Boundary conditions for $k$ and $\varepsilon$ (standard $k-\varepsilon$ model)**

We impose $k$ with a Dirichlet condition using the friction velocity $u_k$ (see equation (II.2.1)) :

$$k = \frac{u_k^2}{C_\mu^{\frac{1}{2}}} \tag{II.2.28}$$

We want to impose the normal derivative of $\varepsilon$ from of the following theoretical law (see the notations in figure II.2.2) :

$$G_{\text{theo},\varepsilon} = \frac{\partial \left(u_k^3/(\kappa\, y)\right)}{\partial y} \tag{II.2.29}$$

We use point $M$ to impose a boundary condition with a higher order of accuracy in space. Indeed, using the simple expression $\varepsilon_F = \varepsilon_I + d\partial_y\varepsilon_I + O(d^2)$ leads to first order accuracy. A second order accuracy can be reached using the following Taylor series expansion:

$$\left\{ \begin{array}{ll} \varepsilon_M & = \varepsilon_I - \dfrac{d}{2}\partial_y\varepsilon_I + \dfrac{d^2}{8}\partial_y^2\varepsilon_I + O(d^3) \\ \varepsilon_M & = \varepsilon_F + \dfrac{d}{2}\partial_y\varepsilon_F + \dfrac{d^2}{8}\partial_y^2\varepsilon_F + O(d^3) \end{array} \right. \tag{II.2.30}$$

By substracting these twxo expression, we obtain

$$\varepsilon_F = \varepsilon_I - \frac{d}{2}(\partial_y\varepsilon_I + \partial_y\varepsilon_F) + O(d^3) \tag{II.2.31}$$

Additionally, we have

$$\left\{ \begin{array}{ll} \partial_y\varepsilon_I & = \partial_y\varepsilon_M + d\partial_y^2\varepsilon_M + O(d^2) \\ \partial_y\varepsilon_F & = \partial_y\varepsilon_M - d\partial_y^2\varepsilon_M + O(d^2) \end{array} \right. \tag{II.2.32}$$

The sum of these last two expressions gives $\partial_y\varepsilon_I + \partial_y\varepsilon_F = 2\partial_y\varepsilon_M + O(d^2)$ and, using equation (II.2.31), we finally obtain a second order approximation for $\varepsilon_F$ :

$$\varepsilon_F = \varepsilon_I - d\partial_y\varepsilon_M + O(d^3) \tag{II.2.33}$$

The theoretical value (see equation II.2.29) is then used in order to evaluate $\partial_y\varepsilon_M$ and thus the value to prescribe at the boundary is obtained ($d = I'F$) :

$$\varepsilon_F = \varepsilon_I + d\frac{u_k^3}{\kappa\,(d/2)^2} \tag{II.2.34}$$

This expression is extended to non-orthogonal mesh without any precautions (which is bound to deteriorate the spatial accuracy ot the method).

Additionally, the velocity $u_k$ is set to zero for $y^+ \leqslant y^+_{lim}$. Consequently, the value of $k$ and the flux of $\varepsilon$ are both zero.

Finally we have :

**Boundary conditions for $k$ and $\varepsilon$**

$$\left\{ \begin{array}{ll} k_F & = \dfrac{u_k^2}{C_\mu^{\frac{1}{2}}} \\ \varepsilon_F & = \varepsilon_{I'} + I'F\dfrac{u_k^3}{\kappa\,(I'F/2)^2} \end{array} \right. \tag{II.2.35}$$

with $u_k = 0$ if $y^+ \leqslant y^+_{lim}$

and the associated pair of coefficients

$$\textbf{Coefficients associated to the boundary conditions of } k \textbf{ et } \varepsilon$$
$$\begin{cases} A_k = \dfrac{u_k^2}{C_\mu^{\frac{1}{2}}} & \text{and } B_k = 0 \\ A_\varepsilon = I'F\dfrac{u_k^3}{\kappa\,(I'F/2)^2} & \text{and } B_\varepsilon = 1 \end{cases} \quad \text{with } u_k = 0 \text{ if } y^+ \leqslant y^+_{lim} \tag{II.2.36}$$

• **Boundary conditions for $R_{ij}$ and $\varepsilon$ (standard $R_{ij} - \varepsilon$ model)**

The boundary conditions for the Reynolds stresses in the coordinate system attached to the wall write ($\hat{R}$ refers to the local coordinate system) :

$$\partial_{\tilde{n}}\hat{R}_{\tau\tau} = \partial_{\tilde{n}}\hat{R}_{\tilde{n}\tilde{n}} = \partial_{\tilde{n}}\hat{R}_{bb} = 0 \quad \text{et } \hat{R}_{\tau\tilde{n}} = -u^* u_k \quad \text{et } \hat{R}_{\tau b} = \hat{R}_{\tilde{n}b} = 0 \tag{II.2.37}$$

Additionally, if the value obtained for $y^+$ is lower than $y^+_{lim}$, all Reynolds stresses are set to zero (we assume that the turbulent stresses are negligible compared to the viscous stresses).

Although it is done systematically in the code, expressing the above boundary conditions in the computation coordinate system is relatively complex (rotation of a tensor): the reader is referred to the documentation of `clsyvt` where more details are provided. In what follows, the boundary conditions will only be presented in the local coordinate system.

Thus we want to impose the boundary values :

$$\textbf{Boundary conditions of } R_{ij}$$
$$\begin{cases} \text{if } y^+ \leqslant y^+_{lim} & \hat{R}_{\alpha\alpha,F} = \hat{R}_{\alpha\beta,F} = 0 \\ \text{otherwise} & \begin{cases} \hat{R}_{\alpha\alpha,F} = \hat{R}_{\alpha\alpha,I'} \text{ with } \alpha \in \{\tau, \tilde{n}, b\} \text{ (without summation)} \\ \hat{R}_{\tau\tilde{n}} = -u^* u_k \text{ and } \hat{R}_{\tau b} = \hat{R}_{\tilde{n}b} = 0 \end{cases} \end{cases} \tag{II.2.38}$$

For the dissipation, the boundary condition applied is identical to the one applied with the $k - \varepsilon$ model :

$$\textbf{Boundary conditions of } \varepsilon \ (R_{ij} - \varepsilon)$$
$$\begin{cases} \varepsilon_F = \varepsilon_{I'} + I'F\dfrac{u_k^3}{\kappa\,(I'F/2)^2} \\ \text{with } u_k = 0 \text{ if } y^+ \leqslant y^+_{lim} \end{cases} \tag{II.2.39}$$

These boundary conditions can be imposed explicitly (by default, ICLPTR=0) or (semi-)implicitly ( ICLPTR=1). The standard option (explicit) leads to the following values[5] of the coefficients $A$ and $B$ :

[5]It can be noticed that the value of $\varepsilon$ is not reconstructed at $I'$. We thus wish to improve the "stability" since $\varepsilon$ has a very steep gradient at the wall ($\varepsilon \approx \dfrac{1}{y}$), and thus only weak recontruction errors at $I'$ could lead to important deterioration of the results. However, it would be necessary to check if stability is altered with the gradient reconstruction of `gradrc`.

**Coefficients associated to the explicit boundary conditions of $R_{ij}$et $\varepsilon$**

$$\left\{\begin{array}{l} \text{If } y^+ \leqslant y^+_{lim} : \\ \qquad A_{\hat{R}_{\alpha\alpha}} = A_{\hat{R}_{\alpha\beta}} = 0 \quad \text{and } B_{\hat{R}_{\alpha\alpha}} = B_{\hat{R}_{\alpha\beta}} = 0 \\ \text{Otherwise :} \\ \qquad \left\{\begin{array}{lll} A_{\hat{R}_{\alpha\alpha}} = (R_{\alpha\alpha})_I & \text{and } B_{\hat{R}_{\alpha\alpha}} = 0 & \text{with } \alpha \in \{\tau, \tilde{n}, b\} \text{ (without summation)} \\ A_{\hat{R}_{\tau\tilde{n}}} = -u^* u_k & \text{and } B_{\hat{R}_{\tau\tilde{n}}} = 0 & \\ A_{\hat{R}_{\tau b}} = A_{\hat{R}_{\tilde{n}b}} = 0 & \text{and } B_{\hat{R}_{\tau b}} = B_{\hat{R}_{\tilde{n}b}} = 0 & \end{array}\right. \\ \text{And for all cases :} \\ \qquad A_\varepsilon = \varepsilon_I + I'F \dfrac{u_k^3}{\kappa\,(I'F/2)^2} \text{ and } B_\varepsilon = 0 \end{array}\right.$$

with $u_k = 0$ if $y^+ \leqslant y^+_{lim}$

(II.2.40)

The semi-implicit option leads to the following values for the coefficients $A$ and $B$. They differ from the preceeding ones, only as regards as the diagonal Reynolds stresses and dissipation. In the general case, impliciting of some components of the tensor in the local coordinate system leads to partially impliciting all the components in the global computation coordinate system :

**Coefficients associated to the semi-implicit boundary conditions of sur les variables $R_{ij}$et $\varepsilon$**

$$\left\{\begin{array}{l} \text{If } y^+ \leqslant y^+_{lim} : \\ \qquad A_{\hat{R}_{\alpha\alpha}} = A_{\hat{R}_{\alpha\beta}} = 0 \quad \text{and } B_{\hat{R}_{\alpha\alpha}} = B_{\hat{R}_{\alpha\beta}} = 0 \\ \text{Sinon :} \\ \qquad \left\{\begin{array}{lll} A_{\hat{R}_{\alpha\alpha}} = 0 & \text{and } B_{\hat{R}_{\alpha\alpha}} = 1 & \text{with } \alpha \in \{\tau, \tilde{n}, b\} \text{ (without summation)} \\ A_{\hat{R}_{\tau\tilde{n}}} = -u^* u_k & \text{and } B_{\hat{R}_{\tau\tilde{n}}} = 0 & \\ A_{\hat{R}_{\tau b}} = A_{\hat{R}_{\tilde{n}b}} = 0 & \text{and } B_{\hat{R}_{\tau b}} = B_{\hat{R}_{\tilde{n}b}} = 0 & \end{array}\right. \\ \text{And for all cases :} \\ \qquad A_\varepsilon = I'F \dfrac{u_k^3}{\kappa\,(I'F/2)^2} \text{ and } B_\varepsilon = 1 \end{array}\right.$$

with $u_k = 0$ if $y^+ \leqslant y^+_{lim}$

(II.2.41)

• **Boundary conditions of the** *VarScalaires*

Only the boundary conditions when a boundary value is imposed (at the wall or away from the wall with a possible external exchange coefficient) are treated here. The reader is referred to the notations in figure II.2.1 and to the general presentation provided in `condli` (in what follows only the most essential part of the presentation is repeated).

The conservation of the normal flux at the boundary for variable $f$ writes :

$$\underbrace{h_{int}(f_{b,int}-f_{I'})}_{\phi_{int}} = \underbrace{h_b(f_{b,ext}-f_{I'})}_{\phi_b} = \begin{cases} \underbrace{h_{imp,ext}(f_{imp,ext}-f_{b,ext})}_{\phi_{real\ imposed}} & \text{(Dirichlet condition)} \\ \underbrace{\phi_{imp,ext}}_{\phi_{real\ imposed}} & \text{(Neumann condition)} \end{cases} \tag{II.2.42}$$

The above two equation are rearranged in order to obtain the value of the numerical flux $f_{b,int} = f_F$ to impose at the wall boundary face, according to the values of $f_{imp,ext}$ and $h_{imp,ext}$ set by the user, and to the value of $h_b$ set by the similarity laws detailed hereafter. The coefficients $A$ and $B$ can then be readily derived, and are presented here.

**Boundary conditions of the** *VarScalaires*

$$f_{b,int} = \underbrace{\frac{h_{imp,ext}}{h_{int}+h_r h_{imp,ext}} f_{imp,ext}}_{A} + \underbrace{\frac{h_{int}+h_{imp,ext}(h_r-1)}{h_{int}+h_r h_{imp,ext}} f_{I'}}_{B} \text{ with } h_r = \frac{h_{int}}{h_b} \tag{II.2.43}$$

**Similarity principle : calculation of $h_b$.**

The only remaining unknown in expression (II.2.43) is the value of $h_b$, since $h_{int}$ has a numerical value which is coherent with the face gradient computation options detailed in `condli` ($h_{int} = \frac{\alpha}{\overline{I'F}}$). The value of $h_b$ must relate the flux to the values $f_{I'}$ and $f_{b,ext}$ by taking into account the boundary layer (the profile of $f$ is not always linear) :

$$\phi_b = h_b \left( f_{b,ext} - f_{I'} \right) \tag{II.2.44}$$

The following considerations are presented using the general notations. In particular, the Prandtl-Schmidt number writes $\sigma = \frac{\nu \rho C}{\alpha}$. When the considered scalar $f$ is the temperature, we have (see `condli`)

- $C = C_p$ (specific heat capacity),
- $\alpha = \lambda$ (molecular conductivity),
- $\sigma = \frac{\nu \rho C_p}{\lambda} = Pr$ (Prandtl number),
- $\sigma_t = Pr_t$ (turbulent Prandtl number),
- $\phi = \left( \lambda + \frac{C_p \mu_t}{\sigma_t} \right) \frac{\partial T}{\partial y}$ (flux in $W m^{-2}$).

The reference "Convection Heat Transfer", Vedat S. Arpaci and Poul S. Larsen, Prentice-Hall, Inc was used.

The flux at the wall writes for the scalar $f$ (the flux is positive if it enters the fluid domain, as shown by the orientation of the $y$ axis) :

$$\phi = - \left( \alpha + C \frac{\mu_t}{\sigma_t} \right) \frac{\partial f}{\partial y} = -\rho C \left( \frac{\alpha}{\rho C} + \frac{\mu_t}{\rho \sigma_t} \right) \frac{\partial f}{\partial y} \tag{II.2.45}$$

Similarly for the temperature, with $a = \frac{\lambda}{\rho C_p}$ and $a_t = \frac{\mu_t}{\rho \sigma_t}$, we have :

$$\phi = -\rho C_p (a + a_t) \frac{\partial T}{\partial y} \tag{II.2.46}$$

In order to make $f$ dimensionless, we introduce $f^*$ defined using the flux at the boundary $\phi_b$ :

$$f^* = -\frac{\phi_b}{\rho C u_k} \tag{II.2.47}$$

For the temperature, we thus have :

$$T^* = -\frac{\phi_b}{\rho C_p u_k} \tag{II.2.48}$$

We then divide both sides of equation (II.2.45) by $\phi_b$. For the left-hand side, we simplify using the conservation of the flux (and thus the fact that $\phi = \phi_b$). For the right-hand side, we replace $\phi_b$ by its value $-\rho C u_k f^*$. With the notations :

$$\nu = \frac{\mu}{\rho} \qquad \nu_t = \frac{\mu_t}{\rho} \qquad y^+ = \frac{y u_k}{\nu} \qquad f^+ = \frac{f - f_{b,ext}}{f^*} \tag{II.2.49}$$

we have :

$$1 = \left( \frac{1}{\sigma} + \frac{1}{\sigma_t} \frac{\nu_t}{\nu} \right) \frac{\partial f^+}{\partial y^+} \tag{II.2.50}$$

One can remark at this stage that with the notations used in the preceeding, $h_b$ can be expressed as a function of $f_{I'}^+$ :

$$h_b = \frac{\phi_b}{f_{b,ext} - f_{I'}} = \frac{\rho\, C\, u_k}{f_{I'}^+} \tag{II.2.51}$$

In order to compute $h_b$, we integrate equation (II.2.50) to obtain $f_{I'}^+$. The only difficulty then consists in prescribing a variation law $\mathcal{K} = \frac{1}{\sigma} + \frac{1}{\sigma_t}\frac{\nu_t}{\nu}$.

In the fully developed turbulent region (far enough from the wall, for $y^+ \geqslant y_2^+$), a mixing length hypothesis models the variations of $\nu_t$ :

$$\nu_t = l^2 \left|\frac{\partial U}{\partial y}\right| = \kappa\, y\, u^* \tag{II.2.52}$$

Additionally, the molecular diffusion of $f$ (or the conduction when $f$ represents the temperature) is negligible compared to its turbulent diffusion : therefore we neglect $\frac{1}{\sigma}$ compared to $\frac{1}{\sigma_t}\frac{\nu_t}{\nu}$. Finally we have [6] :

$$\mathcal{K} = \frac{\kappa\, y^+}{\sigma_t} \tag{II.2.53}$$

On the contrary, in the near-wall region (for $y^+ < y_1^+$) the turbulent contribution becomes negligible compared to the molecular contribution and we thus neglect $\frac{1}{\sigma_t}\frac{\nu_t}{\nu}$ compared to $\frac{1}{\sigma}$.

It would be possible to restrict ourselves to these two regions, but Arpaci and Larsen suggest the model can be improved by introducing an intermediate region ($y_1^+ \leqslant y^+ < y_2^+$) in which the following hypothesis is made :

$$\frac{\nu_t}{\nu} = a_1 (y^+)^3 \tag{II.2.54}$$

where $a_1$ is a constant whose value is obtained from experimental correlations :

$$a_1 = \frac{\sigma_t}{1000} \tag{II.2.55}$$

Thus the following model is used for $\mathcal{K}$ (see a sketch in figure 2.2) :

$$\mathcal{K} = \begin{cases} \dfrac{1}{\sigma} & \text{if } y^+ < y_1^+ \\ \dfrac{1}{\sigma} + \dfrac{a_1 (y^+)^3}{\sigma_t} & \text{if } y_1^+ \leqslant y^+ < y_2^+ \\ \dfrac{\kappa\, y^+}{\sigma_t} & \text{if } y_2^+ \leqslant y^+ \end{cases} \tag{II.2.56}$$

The values of $y_1^+$ and $y_2^+$ are obtained by calculating the intersection points of the variations laws used for $\mathcal{K}$.

The existence of an intermediate region depends upon the values of $\sigma$. Let's first consider the case where $\sigma$ cannot be neglected compared to 1. In practise we consider $\sigma > 0,1$ (this is the common case when scalar $f$ represents the air or the water temperature in normal temperature and pressure conditions). It is assumed that $\frac{1}{\sigma}$ can be neglected compared to $\frac{a_1 (y^+)^3}{\sigma_t}$ in the intermediate region. We thus obtain :

$$y_1^+ = \left(\frac{1000}{\sigma}\right)^{\frac{1}{3}} \qquad y_2^+ = \sqrt{\frac{1000\kappa}{\sigma_t}} \tag{II.2.57}$$

[6] We make the approximation that the definitions of $y^+$ from $u^*$ and $u_k$ are equivalent.

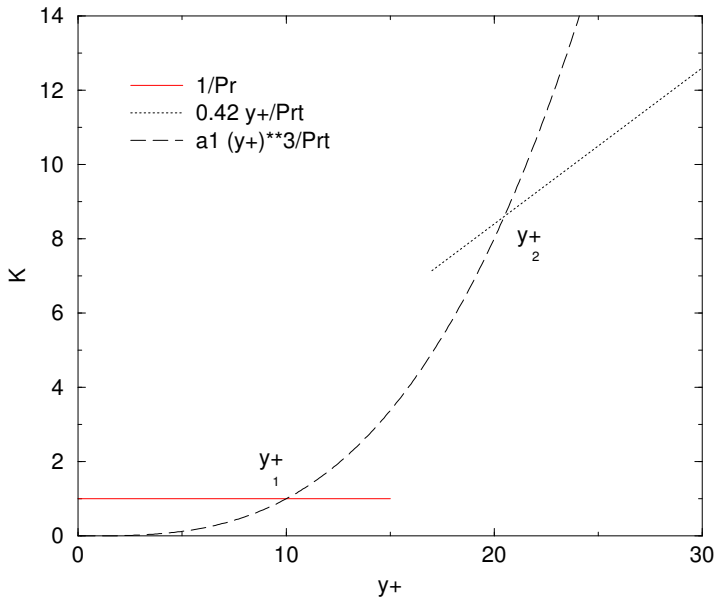



Figure II.2.3: $(a + a_t)/\nu$ as a function of $y^+$ obtained for $\sigma = 1$ and $\sigma_t = 1$.

The dimensionless equation (II.2.50) is integrated under the same hypothesis and we obtain the law of $f^+$ :

$$\begin{cases} f^+ = \sigma\, y^+ & \text{if } y^+ < y_1^+ \\ f^+ = a_2 - \dfrac{\sigma_t}{2\, a_1\, (y^+)^2} & \text{if } y_1^+ \leqslant y^+ < y_2^+ \\ f^+ = \dfrac{\sigma_t}{\kappa}\, ln(y^+) + a_3 & \text{if } y_2^+ \leqslant y^+ \end{cases} \tag{II.2.58}$$

where $a_2$ and $a_3$ are integration constants, which have been chosen to obtain a continuous profile of $f^+$ :

$$a_2 = 15\sigma^{\frac{2}{3}} \qquad a_3 = 15\sigma^{\frac{2}{3}} - \frac{\sigma_t}{2\kappa}\left(1 + ln\left(\frac{1000\kappa}{\sigma_t}\right)\right) \tag{II.2.59}$$

Let's now study the case where $\sigma$ is much smaller than 1. In practise it is assumed that $\sigma \leqslant 0,1$ (this is for instance the case for liquid metals whose thermal conductivity is very large, and who have Prandtl number of values of the order of 0.01). The intermediate region then disappears and the coordinate of the interface between the law used in the near-wall region and the one used away from the wall is given by :

$$y_0^+ = \frac{\sigma_t}{\kappa\sigma} \tag{II.2.60}$$

The dimensionless equation (II.2.50) is then integrated under the same hypothesis, and the law of $f^+$ is obtained :

$$\begin{cases} f^+ = \sigma\, y^+ & \text{if } y^+ \leqslant y_0^+ \\ f^+ = \dfrac{\sigma_t}{\kappa}\, ln\left(\dfrac{y^+}{y_0^+}\right) + \sigma\, y_0^+ & \text{if } y_0^+ < y^+ \end{cases} \tag{II.2.61}$$

To summarize, the computation of $h_b$

$$h_b = \frac{\phi_b}{f_{b,ext} - f_{I'}} = \frac{\rho\, C\, u_k}{f_{I'}^+} \tag{II.2.62}$$

is performed by calculating $f_{I'}^+$ from $y^+ = y_{I'}^+$ using the following laws.

If $\sigma \leqslant 0,1$, a two-layer model is used :

$$\begin{cases} f^+ = \sigma\, y^+ & \text{if } y^+ \leqslant y_0^+ \\ f^+ = \dfrac{\sigma_t}{\kappa}\, ln\left(\dfrac{y^+}{y_0^+}\right) + \sigma\, y_0^+ & \text{if } y_0^+ < y^+ \end{cases} \tag{II.2.63}$$

with

$$y_0^+ = \frac{\sigma_t}{\kappa\sigma} \tag{II.2.64}$$

If $\sigma > 0,1$, a three-layer model is used :

$$\begin{cases} f^+ = \sigma\, y^+ & \text{if } y^+ < y_1^+ \\ f^+ = a_2 - \dfrac{\sigma_t}{2\, a_1\, (y^+)^2} & \text{if } y_1^+ \leqslant y^+ < y_2^+ \\ f^+ = \dfrac{\sigma_t}{\kappa}\, ln(y^+) + a_3 & \text{if } y_2^+ \leqslant y^+ \end{cases} \tag{II.2.65}$$

with

$$y_1^+ = \left(\frac{1000}{\sigma}\right)^{\frac{1}{3}} \qquad y_2^+ = \sqrt{\frac{1000\kappa}{\sigma_t}} \tag{II.2.66}$$

and

$$a_2 = 15\sigma^{\frac{2}{3}} \qquad a_3 = 15\sigma^{\frac{2}{3}} - \frac{\sigma_t}{2\kappa}\left(1 + ln\left(\frac{1000\kappa}{\sigma_t}\right)\right) \tag{II.2.67}$$

## 2.3 Implementation

This subroutines treats the variables `IVAR` on the faces `IFAC` for which `ICODCL(IFAC,IVAR)`=5.

First the velocity of the wall (possibly zero) is projected onto the plane parallel to the wall. Its three components in the computation coordinate system are stored in the arrays `RCODCL(IFAC,IUIPH,1)`,`RCODCL(IFAC,IVIPH,1)``RCODCL(IFAC,IWIPH,1)`.

We then compute the local coordinate system $\hat{\mathcal{R}}$. For each face, it is stored in the vectors $\underline{\tau}$ `(TX, TY, TZ)`, $\underline{\tilde{n}}$ `(-RNX, -RNY,-RNZ)`, and $\underline{b}$ `(T2X, -T2Y,-T2Z)`. It should be noticed that the third vector is only needed (and thus only calculated) when the $R_{ij} - \varepsilon$ model is used (for the second order tensor projection). Additionally, if the norm of the tangential velocity is lower than the arbitrary value `EPZERO` ($10^{-12}$), the indicator `TXNO` is set to 0 (it is 0 otherwise) and the vector $\underline{\tau}$ is

- chosen arbitrarily, in the plane orthogonal to $\underline{\tilde{n}}$ in $R_{ij} - \varepsilon$ (the components of $\underline{\tilde{n}}$ are used to construct $\underline{\tau}$ ; if $\underline{\tilde{n}}$ is the zero vector, the code stops) ;
- zero, otherwise.

Once the local coordinate system has been calculated, the subroutine `clca66` is called with [7] `CLSYME` =0 (if the $R_{ij} - \varepsilon$ model is activated) in order to compute the matrix `ALPHA`. The latter will be used to compute the values to prescribe at the boundary, using the the values of the Reynolds stress tensor at the points $I'$.

[7] `CLSYME` = 0 indicates that we consider wall boundary conditions and not symmetry boundary conditions. One is referred to `CLSYVT` for more details on `clca66`.

The friction velocities are then computed and stored in `UET` ($= u^*$) and `UK` ($= u_k$). The subroutine `causta` computes the friction velocity for the one velocity scale model (`IDEUCH`=0). Fot the two velocity scale model (`IDEUCH`=1), the (more simple) computation of `UET` and `UK` is performed directly in `clptur`. The user can then modify the velocity scales (to take into account a rough wall, or variants of the logarithmic law for instance) in the user subroutine `usruet`.

If one velocity scale model is activated, we impose `UK=UET` to keep the coherence of the coding in the following of the subroutine. However, if we are in the viscous sublayer, we force `UK=0` to obtain a zero value for $k$ and a zero flux for $\varepsilon$ (the value `UET` is not used, because the condition of the velocity is a no-slip condition and the Reynolds stresses are zero).

If we are in the viscous sublayer, we compute `UET` again (and `YPLUS` if we use the one velocity scale model) since the previous derivation has been made under the assumption that we were in the logarithmic region.

The boundary conditions for the velocity are then completed.

- In $k-\varepsilon$, the coefficients $A_{flux}$ (coming from the analysis of the tangential velocity) for the three components of the velocity are stored in

  `COEFA(IFAC,ICLUF)`, `COEFA(IFAC,ICLVF)` et `COEFA(IFAC,ICLWF)`

  , respectively.
  Similarly, the coefficients $B_{flux}$ are stored in

  `COEFB(IFAC,ICLUF)`, `COEFB(IFAC,ICLVF)` et `COEFB(IFAC,ICLWF)`.

  The boundary conditions coming from the analysis of the velocity gradient $A_{grad}$ and $B_{grad}$ are stored in

  `COEFA(IFAC,ICLU)`, `COEFA(IFAC,ICLV)`, `COEFA(IFAC,ICLW)`

  and

  `COEFB(IFAC,ICLU)`, `COEFB(IFAC,ICLV)`, `COEFB(IFAC,ICLW)`.

- When the tangential velocity at $I'$ is lower than `EPZERO`, the indicator `TXN0` is set to zero (otherwise, `TXN0` is 1). Likewise, when the value of $y^+$ is lower or equal to $10,88$, the indicator `UNTURB` is set to 0 (otherwise, it is 1). Both these indicators are used to set to zero the coefficients $A$, and thus to prescribe no-slip boundary conditions.

- the velocity of the wall is taken into account in the boundary conditions (coefficients `COEFA`).

The conditions of the turbulent variables are then filled in. Additional details are provided for the Reynolds tensor in $R_{ij}-\varepsilon$. Using tensorial notation, we want to obtain $\underline{\underline{R}}_F = E_{loglo}\,\underline{\underline{\hat{R}}}_F\,E^t_{loglo}$, where $\underline{\underline{\hat{R}}}_F$ is the stress tensor to impose in the local coordinate system and $E_{loglo}$ is the transformation matrix. The tensor in the local coordinate system is defined by the boundary conditions detailed above, $\underline{\underline{\hat{R}}}_F = \underline{\underline{\hat{R}}}_F^{B=1}$, with [8] :

$$\underline{\underline{\hat{R}}}_F^B = \begin{bmatrix} \hat{R}_{\tau\tau,I'} & -Bu^*u_k & 0 \\ -Bu^*u_k & \hat{R}_{\tilde{n}\tilde{n},I'} & 0 \\ 0 & 0 & \hat{R}_{bb,I'} \end{bmatrix} \tag{II.2.68}$$

The Reynolds tensor is stored in a vector with 6 components (at points $I'$ of the boundary faces, these values are stored in `RIJIPB(IFAC,II)`, with `IFAC` the index of the face and `II` the index of the stress, from 1 to 6 `II` refers to $R_{11}, R_{22}, R_{33}, R_{12}, R_{13}$ and $R_{23}$, respectively. The array `IFAC` of dimension $6\times 6$ (calculated by `clca66`) is used to perform the necessary change of coordinate system computations, and to obtain the values of $\underline{\underline{\hat{R}}}_F^{B=0}$ from the values stored in `RIJIPB(IFAC,II)`. Thus,

[8] The parameter $B$ permits to set to zero two terms if needed.

- for each Reynolds stress ISOU, in case of explicit boundary conditions (ICLPTR=0), we set COEFA(IFAC,ISOU) to the corresponding value of $\underline{\underline{R}}_F^{B=1}$ obtained by first computing $\underline{\underline{R}}_F^{B=0}$ by $\sum_{II=1,6}$ALPHA(ISOU,II)*RIJIPB(IFAC,II), and then by adding the quantity factorised by B in the previous sum, in order to obtain the complete value of $\underline{\underline{R}}_F^{B=1}$. Since the boundary conditions are explicit, COEFB is set to 0.

- in case of semi-implicit boundary conditions (ICLPTR=1), for each Reynolds stress ISOU, we store in the array COEFA(IFAC,ISOU) the corresponding value $\underline{\underline{R}}_F^{B=1}$ minus the terms which depend upon RIJIPB(IFAC,ISOU) (which will be implicited) This value is obtained by first calculating $\sum_{II=1,6|II\neq ISOU}$ALPHA(ISOU,II)*RIJIPB(IFAC,II), and then by adding the quantity factorised by B in order to obtain the value of $\underline{\underline{R}}_F^{B=1}$ minus the term ALPHA(ISOU,ISOU) RIJIPB(IFAC,ISOU). ALPHA(ISOU,ISOU) is then stored in COEFB (implicit part of the boundary conditions)[9].

The boundary conditions of the scalars are then completed. The coefficients COEFA and COEFB are simply calculated using the boundary conditions previously described. The only difficulty consists in dealing adequatly with the various values necessary to compute the exchange coefficient $h_b$ without any mistake.

The indicator ISCSTH gives for each *VarScalaires* the value of $C$ to use in the treatment of the boundary conditions. Thus, for ISCSTH=1, the variable should be treated as a température, with $C = C_p$. For ISCSTH=0 or 2, the variable should be treated as a passive scalar or as an enthalpy, respectively, with $C = 1$ (dimensionless constant) in both cases. For ISCSTH=3, the variables is the resolved energy in the framework of the compressible module (see cfxtcl). We then have $C = 1$.

Additionally, a strictly positive value of the integer IPCCP indicates that $C_p$ varies in space and that it is available in the array PROPCE(IEL,IPCCP ) (filled in USPHYV). When IPCCP is zero, $C_p$ is constant and it is available as a real number CP0(IPHAS).

The indicator IHCP gathers these informations :

- IHCP = 0 : CPP = $C = 1$
- IHCP = 1 : CPP = $C = C_p$ uniform in space
- IHCP = 2 : CPP = $C = C_p$ variable in space

For *VarScalaire* LL, the indicator IVISLS(LL) indicates if $\frac{\alpha_m}{C}$ is variable in space and available in the array PROPCE(IEL,IPCVSL) (IVISLS $> 0$) or if is uniform in space and available as a real number VISLS0(LL) (IVISLS $= 0$). We take RKL= $\frac{\alpha_m}{C}$.

The local Prandtl number is then calculated as PRDTL = $\frac{\mu}{\alpha_m/C}$ ($\mu$ is the molecular dynamic viscosity available in VISCLC).

The coefficient $h_{int} = \frac{\alpha}{\overline{\overline{I'F}}}$ is then calculated and stored in HINT.

When a turbulence model is activated, the subroutine HTURBP computes HFLUI = $Pr\frac{y^+}{T^+}$ (or HFLUI = 1 in the viscous sublayer), which is then immediately multiplied by CPP*RKL/DISTBF= $\frac{\alpha_m}{\overline{\overline{I'F}}}$ to obtain HFLUI = $h_b = \frac{\rho C u_k}{T^+}$ (or HFLUI = $h_b = \frac{\alpha_m}{\overline{\overline{I'F}}}$ in the viscous sublayer). If the calculation is performed in laminar, we simply have HFLUI = HINT ($= h_b = h_{int} = \frac{\alpha}{\overline{\overline{I'F}}}$)

[9] the implicit part does not include the tangential stress due to the friction velocities and thus the boundary conditions are not fully implicit.

In case we wish to store the exchange coefficient (coupling with SYRTHES), `HFLUI` is kept in the array `HBORD`. In case the raditation module is used, `HFLUI` is stored in the work array `RA(IHCONV)`.

We then have all the information necessary to compute the coefficients $A$ and $B$ (`COEFA` and `COEFB`) relative to the treated variable. Finally one can give the following correspondances to relate the source code implementation to expression (II.2.43) : `HEXT` $= h_{imp,ext}$, `PIMP` $= f_{imp,ext}$, `HREDUI` $= h_r$, `HINT` $= h_{int}$, `HFLUI` $= h_b$.

## 2.4 Points to treat

The use of `HFLUI/CPP` when `ISCSTH` is 2 (case with radiation) needs to be checked (`CPP` is actually 1 in this case).

The boundary conditions of the velocity are based on derivations focusing on only one term of the tangential stress $(\mu_I + \mu_{t,I})(\underline{\underline{\text{grad}}}\ \underline{u})\,\underline{n}$ without taking into account the tranpose gradient.

In order to establish the boundary conditions of the velocity in $k-\varepsilon$ based on the constraint , a projection onto the plane tangent to the wall and an arbitrary zero normal velocity are introduced.

The hypothesis made in order to establish formulae for the different types of boundary conditions (dissipation, velocities) are based on the assumption that the mesh is orthogonal at the wall. This assumption is extended without any caution to the case of non-orthogonal meshes.

The one velocity scale (`causta`) wall function requires solving an equation using a Newton algorithm. The computational cost of the latter is low. One can also used a 1/7 power law (Werner et Wengle) which yields results which are as accurate as the logarithmic law in the logarithmic region, and which permits analytical resolutions (chosen option in LES mode). Be careful however, since with this law, the intersection with the linear law is slightly different, which thus requires some adaptations (intersection around 11.81 instead of 10.88 for the law adopted here $U^+ = 8,3\,(y^+)^{\frac{1}{7}}$).

The values of all the physical properties are taken at the cell centres, without any reconstruction. Without modifying this approach, it would be useful to keep this in mind.

For the thermal law with very small Prandtl numbers compared to 1, Arpaci and Larsen suggest $y_0^+ \simeq 5/Pr$ (with proof from experimental data) rather than $Pr_t/(Pr\,\kappa)$ (current value, computed as the analytical intersection of the linear and logarithmic laws considered). One should address this question.

# 3- Sous-programme `clptrg`

## 3.1 Fonction

Ce sous-programme est dédié au calcul des conditions aux limites en paroi rugueuse. On utilise le formalisme introduit dans `CONDLI` pour les conditions aux limites générales.

Par conditions aux limites en paroi, on entend ici l'ensemble des conditions aux limites pour la vitesse, les grandeurs turbulentes ($k$, $\varepsilon$), la température lorsqu'elle a une valeur de paroi imposée (ou l'enthalpie et plus généralement les *VarScalaires*[1] à traiter en paroi en prenant en compte une loi de similitude pour la couche limite associée). Pour les *VarScalaires*, en particulier, lorsque les conditions aux limites de paroi sont du type Neumann (homogène ou non), elles sont traitées dans `condli` et on ne s'y intéresse donc pas ici. En particulier, les conditions aux limites des *VarScalaires* représentant la variance de fluctuations d'autres *VarScalaires* ne sont pas traitées ici car leur traitement en paroi est de type Neumann homogène.

On indique comment sont calculés les couples de coefficients $A_b$ et $B_b$ qui sont utilisés pour le calcul de certains termes discrets des équations à résoudre et qui permettent en particulier de déterminer une valeur associée aux faces de bord $f_{b,int}$ (en un point localisé au "centre" de la face de bord, barycentre de ses sommets) par la relation $f_{b,int} = A_b + B_b\, f_{I'}$ ($f_{I'}$ est la valeur de la variable au point $I'$, projeté du centre de la cellule jouxtant le bord sur la droite normale à la face de bord et passant par son centre : voir la figure II.3.1).

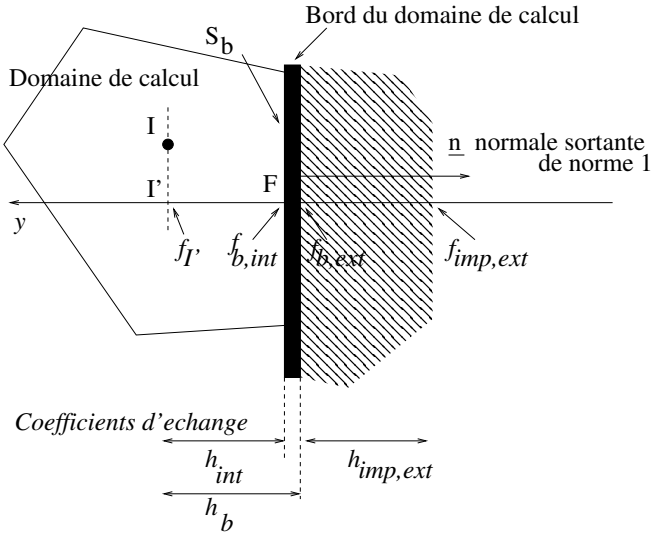



Figure II.3.1: Cellule de bord.

## 3.2 Discrétisation

• **Notations**

[1] Comme dans `condli`, on désignera ici par *VarScalaire* toute variable solution d'une équation de convection-diffusion autre que la vitesse, la pression et les grandeurs turbulentes $k$, $\varepsilon$. La dénomination *VarScalaire* pourra en particulier se rapporter à la température, à l'enthalpie ou à un scalaire passif.

La vitesse de la paroi est notée $\underline{v}_p$. On la suppose projetée dans le plan tangent à la paroi (si elle ne l'est pas, le code la projette).

La vitesse du fluide est notée $\underline{u}$. L'indice $I$, $I'$ ou $F$ désigne le point auquel elle est estimée. La composante tangentielle par rapport à la paroi est notée $u_\tau$. La vitesse du fluide dans le repère lié à la paroi (vitesse "relative") est notée $\underline{u}^r = \underline{u} - \underline{v}_p$.

Le repère orthonormé lié à la paroi est noté $\hat{\mathcal{R}} = (\underline{\tau}, \underline{\tilde{n}}, \underline{b})$.

- $\underline{\tilde{n}} = -\underline{n}$ est le vecteur normé orthogonal à la paroi et dirigé vers l'intérieur du domaine de calcul.
- $\underline{\tau} = \dfrac{1}{\|\underline{u}^r_{I'} - (\underline{u}^r_{I'} \,.\, \underline{\tilde{n}})\|} \left[\underline{u}^r_{I'} - (\underline{u}^r_{I'} \,.\, \underline{\tilde{n}})\right]$ est le vecteur normé porté par la projection de la vitesse relative en $I'$, $\underline{u}^r_{I'}$, dans le plan tangent à la paroi (*i.e.* orthogonal à $\underline{\tilde{n}}$) : voir la figure II.3.1.
- $\underline{b}$ est le vecteur normé complétant le repère direct.

Dans le cas du **modèle à deux échelles de vitesse**, on note :

- $u_k$ la vitesse de frottement en paroi obtenue à partir de l'énergie turbulente.
- $u^*$ la vitesse de frottement en paroi déterminée à partir de la relation $\dfrac{u^r_{\tau,I'}}{u^*} = f(z_p)$.
- $z_p$ représente une distance à la paroi (c'est à dire la distance depuis le bord du domaine de calcul), soit $z_p = I'F$ (voir la figure II.3.1). La fonction $f$ traduit la forme idéale du profil de vitesse. Dans l'atmosphère, cette fonction est donnée par une loi de type logarithmique faisant intervenir la rugosité dynamique de la paroi $z_0$ :
  $f(z_p) = \dfrac{1}{\kappa} ln \left( \dfrac{z_p + z_0}{z_0} \right)$
- Les deux échelles de vitesse $u_k$ et $u^*$ sont simples à calculer mais leur obtention nécessite la connaissance de l'énergie turbulente $k_I$ au centre de la maille jouxtant la face de bord.
- Le modèle à deux échelles est le modèle par défaut dans *Code_Saturne*. Il permet souvent, et en particulier dans les cas avec transfert thermique, de diminuer les effets de certains défaut liés au modèle $k - \varepsilon$ (exemple classique du jet impactant).

On se sert plus bas de $u^*$ et $u_k$ pour les conditions aux limites portant sur la vitesse et les scalaires (température en particulier).

**Modèle à deux échelles de vitesse**

$$\left\{ \begin{array}{l} u_k = C_\mu^{\frac{1}{4}} k_I^{\frac{1}{2}} \\ u^* \text{solution de } \dfrac{u^r_{\tau,I'}}{u^*} = \dfrac{1}{\kappa} ln(z_k) \\ z_k = \dfrac{I'F + z_0}{z_0} = \dfrac{z_p + z_0}{z_0} \\ \text{avec } C_\mu = 0,09 \text{ et } \kappa = 0,42 \end{array} \right. \tag{II.3.1}$$

Dans le cas du **modèle à une échelle de vitesse**, on note $u^*$ l'unique vitesse de frottement en paroi solution de l'équation $\dfrac{u^r_{\tau,I'}}{u^*} = f(z_p)$. La grandeur $z_p$ représente une distance à la paroi, soit $z_p = I'F$. La fonction $f$ traduit la forme idéale du profil de vitesse comme pour le modèle à deux échelles de vitesses. On peut noter que cette vitesse de frottement, d'un calcul plus délicat (point fixe), s'obtient cependant sans faire référence aux variables turbulentes ($k$, $\varepsilon$). Par commodité, on posera dans le cas du modèle à une échelle $u_k = u^*$.

On se sert plus bas de $u^*$ et $u_k$ pour les conditions aux limites portant sur la vitesse et les scalaires (température en particulier).

**Modèle à une échelle de vitesse**

$$\left\{\begin{array}{l} u_k = u^* \\ u^* \text{solution de } \dfrac{u^r_{\tau,I'}}{u^*} = \dfrac{1}{\kappa} ln(z_k) \\ z_k = \dfrac{I'F + z_0}{z_0} = \dfrac{z_p + z_0}{z_0} \\ \text{avec } C_\mu = 0,09 \text{ et } \kappa = 0,42 \end{array}\right. \tag{II.3.2}$$

• **Conditions aux limites pour la vitesse en $k-\varepsilon$**

On considère tout d'abord les conditions utilisées dans le cas d'un calcul réalisé avec le modèle $k-\varepsilon$. Ce sont en effet les plus complexes et les plus générales.

Les conditions aux limites sont nécessaires pour imposer au bord la contrainte tangentielle $\sigma_\tau = \rho_I u^* u_k$ adéquate dans l'équation de quantité de mouvement[2] ($\rho_I$ est la masse volumique au centre de la cellule $I$). Le terme qui nécessite des conditions aux limites est celui qui contient la dérivée de la vitesse dans la direction normale à la paroi, soit[3] : $(\mu_I + \mu_{t,I})\underline{\underline{\text{grad}}}\ \underline{u}\,\underline{n}$. Il apparaît au second membre de l'équation de quantité de mouvement usuelle (voir `bilsc2` et `preduv`).

Dans le cas où le modèle $k-\varepsilon$ a tendance à surestimer la production de l'énergie turbulente, l'échelle de longueur du modèle, $L_{k-\varepsilon}$, peut devenir significativement plus grande que l'échelle théorique maximale des tourbillons de la couche limite turbulente $L_{\text{théo}}$. On note :

$$\left\{\begin{array}{l} L_{k-\varepsilon} = C_\mu \dfrac{k^{\frac{3}{2}}}{\varepsilon} \\ L_{\text{théo}} = \kappa\ (I'F + z_0) = \kappa\ (z_p + z_0) \end{array}\right. \tag{II.3.3}$$

Dans le cas où $L_{k-\varepsilon} > L_{\text{théo}}$, on a donc $\mu_{t,I} > \mu_t^{lm}$ avec $\mu_{t,I}$ la viscosité turbulente du modèle $k-\varepsilon$ au point $I$ et $\mu_t^{lm} = \rho_I L_{\text{théo}} u_k$ la viscosité turbulente du modèle de longueur de mélange. En outre, la contrainte tangentielle peut s'écrire en faisant apparaître la viscosité turbulente, soit :

$$\sigma_\tau = \rho_I u^* u_k = \frac{u^*}{\kappa\ (I'F + z_0)} \underbrace{\rho_I \kappa\ (I'F + z_0)\, u_k}_{\mu_t^{lm}} \tag{II.3.4}$$

L'échelle de viscosité introduite dans la contrainte est alors en contradiction avec celle déduite de la turbulence calculée alentour par le modèle. On préfère dès lors écrire, en utilisant l'échelle de longueur du $k-\varepsilon$ chaque fois qu'elle est inférieure à la limite $L_{\text{théo}}$ :

$$\sigma_\tau = \frac{u^*}{\kappa\ (I'F + z_0)} max(\mu_t^{lm}, \mu_{t,I}) \tag{II.3.5}$$

On peut alors utiliser cette valeur pour le calcul du flux diffusif qui en dépend dans l'équation de Navier-Stokes :

$$(\mu_I + \mu_{t,I})\underline{\underline{\text{grad}}}\ \underline{u}\,\underline{n} = -\sigma_\tau \underline{\tau} \tag{II.3.6}$$

Or, le gradient de vitesse (gradient à la face de bord) est calculé dans le code sous la forme suivante :

$$(\mu_I + \mu_{t,I})\underline{\underline{\text{grad}}}\ \underline{u}\,\underline{n} = \frac{(\mu_I + \mu_{t,I})}{\overline{I'F}} (\underline{u}_F - \underline{u}_{I'}) \tag{II.3.7}$$

[2] Proposition de modification des conditions aux limites de paroi turbulente pour le Solveur Commun dans le cadre du modèle $k-\varepsilon$ standard, rapport EDF HI-81/00/019/A, 2000, M. Boucker, J.-D. Mattei.

[3] Le terme en gradient transposé est traité dans `vissec` et ne sera pas considéré ici.

Du rapprochement de (II.3.6) et de (II.3.7) on tire alors la valeur de $\underline{u}_F$ à imposer, soit $\underline{u}_{F,flux}$ (respect du flux de quantité de mouvement) :

$$\begin{aligned} \underline{u}_{F,flux} &= \underline{u}_{I'} - \frac{\overline{I'F}}{\mu_I + \mu_{t,I}} \sigma_\tau \underline{\tau} \\ &= \underline{u}_{I'} - \frac{u^*}{\kappa\,(\mu_I + \mu_{t,I})} max(\mu_t^{lm}, \mu_{t,I})\, \frac{I'F}{(I'F + z_0)} \underline{\tau} \end{aligned} \tag{II.3.8}$$

En réalité, une approximation supplémentaire est réalisée, qui consiste à imposer la vitesse normale nulle à la paroi et à utiliser l'équation (II.3.8) projetée sur le plan tangent à la paroi, soit :

$$\underline{u}_{F,flux} = \left[ u_{\tau,I'} - \frac{u^*}{\kappa\,(\mu_I + \mu_{t,I})} max(\mu_t^{lm}, \mu_{t,I})\, \frac{I'F}{(I'F + z_0)} \right] \underline{\tau} \tag{II.3.9}$$

Enfin, on peut également faire apparaître la vitesse de la paroi dans l'expression finale :

**Conditions aux limites sur la vitesse de type "flux"** $(k-\varepsilon)$

$$\left\{ \underline{u}_{F,flux} = \underline{v}_p + \left[ u^r_{\tau,I'} - \frac{u^*}{\kappa\,(\mu_I + \mu_{t,I})} max(\mu_t^{lm}, \mu_{t,I})\, \frac{I'F}{(I'F + z_0)} \right] \underline{\tau} \right. \tag{II.3.10}$$

Un premier couple de coefficients $A_{flux}$ et $B_{flux}$ s'en déduit (pour chaque composante de vitesse séparément) et il n'est utilisé que pour le calcul du terme dépendant de la contrainte tangentielle (voir `bilsc2`) :

**Coefficients associés aux conditions aux limites sur la vitesse de type "flux"** $(k-\varepsilon)$

$$\left\{ \begin{array}{l} \underline{A}_{flux} = \underline{v}_p + \left[ u^r_{\tau,I'} - \dfrac{u^*}{\kappa\,(\mu_I + \mu_{t,I})} max(\mu_t^{lm}, \mu_{t,I})\, \dfrac{I'F}{(I'F + z_0)} \right] \underline{\tau} \\ \underline{B}_{flux} = \underline{0} \end{array} \right. \tag{II.3.11}$$

On a vu ci-dessus comment imposer une condition à la limite permettant de calculer correctement le terme en contrainte. Une analyse supplémentaire est nécessaire pour le calcul des gradients de vitesse. On cherche à trouver une valeur en face de bord qui permette d'obtenir, avec la formulation adoptée pour le gradient, la valeur de la production turbulente la plus proche possible de la valeur théorique, elle-même déterminée en utilisant la loi logarithmique, pour évaluer la dérivée normale de la vitesse tangentielle. Ainsi, on définit (au point $I$) :

$$P_{\text{théo}} = \rho_I u^* u_k \| \frac{\partial u_\tau}{\partial \underline{n}} \|_I = \rho_I \frac{u_k (u^*)^2}{\kappa\ (I'F + z_0)} \tag{II.3.12}$$

Par ailleurs, le terme prépondérant de la production calculée dans la cellule $I$ est, pour les situations classiques ($z$ est l'ordonnée sur l'axe de vecteur directeur $\underline{\tilde{n}}$),

$$P_{\text{calc}} = \mu_{t,I} \left( \frac{\partial u_\tau}{\partial z} \right)^2_I \tag{II.3.13}$$

Or, le gradient normal de la vitesse tangentielle (gradient cellule) est calculé dans le code en volumes finis et son expression dans le cas d'un maillage orthogonal et régulier est la suivante (voir les notations sur la figure II.3.2) :

$$P_{\text{calc}} = \mu_{t,I} \left( \frac{u_{\tau,G} - u_{\tau,F}}{2d} \right)^2 = \mu_{t,I} \left( \frac{u_{\tau,I} + u_{\tau,J} - 2u_{\tau,F}}{4d} \right)^2 \tag{II.3.14}$$

On suppose alors que $u_{\tau,J}$ peut être obtenu à partir de $u_{\tau,I}$ et du gradient normal de $u_\tau$ évalué en G à partir de la loi logarithmique, soit :

$$u_{\tau,J} = u_{\tau,I} + IJ\,.\,(\partial_z u_\tau)_G + \mathcal{O}(IJ^2) \approx u_{\tau,I} + IJ\,.\,\left[ \partial_z \left( \frac{u^*}{\kappa}\, ln(z) \right) \right]_G = u_{\tau,I} + 2d\, \frac{u^*}{\kappa\,(2d + z_0)} \tag{II.3.15}$$

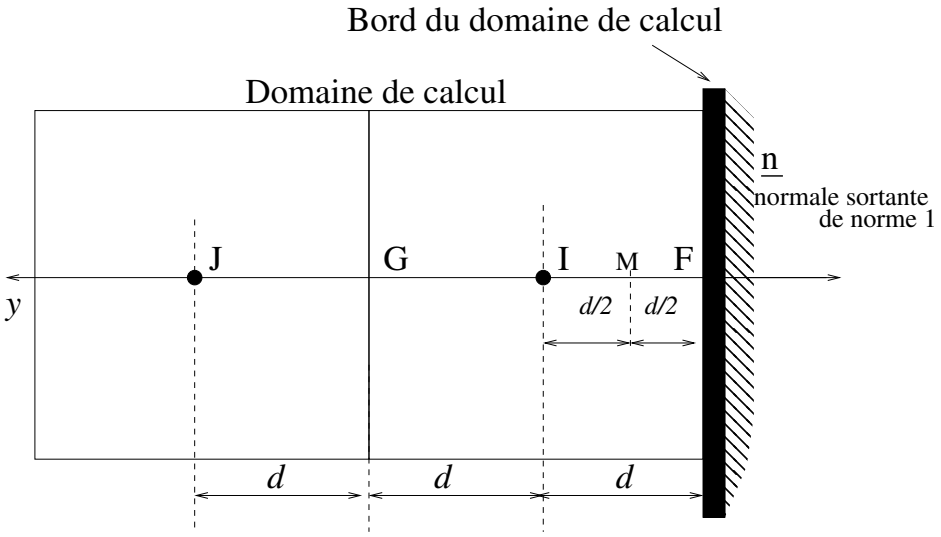



Figure II.3.2: Cellule de bord - Maillage orthogonal.

et l'on obtient alors :

$$\begin{aligned} P_{\text{calc}} &= \mu_{t,I}\left(\frac{2u_{\tau,I}+2d\,\dfrac{u^*}{\kappa\,(2d+z_0)}-2u_{\tau,F}}{4d}\right)^2 \\ &= \mu_{t,I}\left(\frac{u_{\tau,I}+d\,\dfrac{u^*}{\kappa\,(2d+z_0)}-u_{\tau,F}}{2d}\right)^2 \end{aligned} \tag{II.3.16}$$

On rapproche alors les équations (II.3.12) et (II.3.16) pour imposer que la production calculée soit égale à la la production théorique. On étend sans précaution les formules précédentes aux maillages non orthogonaux (la vitesse en $I$ est alors simplement prise en $I'$). On obtient alors l'expression suivante pour $u_{\tau,F}$ :

$$u_{\tau,F,grad} = u_{\tau,I'} - \frac{u^*}{\kappa}\left(2d\sqrt{\frac{\rho_I\kappa\,u_k}{\mu_{t,I}\,(I'F+z_0)}}-\frac{1}{2+z_0/I'F}\right) \tag{II.3.17}$$

On impose d'autre part que le gradient reste au moins aussi raide que celui donné par la dérivée normale du profil de vitesse théorique (logarithmique) en $I'$ :
$\partial_z u_\tau = \partial_z(\dfrac{u^*}{\kappa}\,ln(z)) = \dfrac{u^*}{\kappa\,(I'F+z_0)}$, soit donc :

$$u_{\tau,F,grad} = u_{\tau,I'} - \frac{u^*}{\kappa}max\left(1,2d\sqrt{\frac{\rho_I\kappa\,u_k}{\mu_{t,I}\,(I'F+z_0)}}-\frac{1}{2+z_0/I'F}\right) \tag{II.3.18}$$

La vitesse normale à la paroi est imposée nulle. De plus, si la vitesse tangentielle en $I'$ est nulle (de valeur absolue inférieure à une limite numérique arbitraire de $10^{-12}$) une condition d'adhérence est appliquée. Enfin, on peut également faire apparaître la vitesse de la paroi dans l'expression finale :

**Conditions aux limites sur la vitesse de type "gradient"**$(k-\varepsilon)$

$$\begin{cases} \underline{u}_{F,grad} = \underline{v}_p & \text{si } u^r_{\tau,I'} < 10^{-12} \\ \underline{u}_{F,grad} = \underline{v}_p + u^r_{\tau,I'} - \dfrac{u^*}{\kappa}\left[max\left(1,2d\sqrt{\dfrac{\rho_I\kappa\,u_k}{\mu_{t,I}\,(I'F+z_0)}}-\dfrac{1}{2+z_0/I'F}\right)\right]\underline{\tau} \end{cases} \tag{II.3.19}$$

Un second couple de coefficients $A_{grad}$ et $B_{grad}$ s'en déduit (pour chaque composante de vitesse séparément) et est utilisé chaque fois que le gradient de la vitesse est nécessaire (hormis pour les termes dépendant de la contrainte tangentielle, traités dans `bilsc2` au moyen des coefficients $A_{flux}$ et $B_{flux}$) :

**Coefficients associés aux conditions aux limites sur la vitesse de type "gradient"** $(k-\varepsilon)$

$$\begin{cases} \underline{A}_{grad} = \underline{v}_p & \text{si } u^r_{\tau,I'} < 10^{-12} \\ \underline{A}_{grad} = \underline{v}_p + u^r_{\tau,I'} - \dfrac{u^*}{\kappa}\left[max\left(1, 2d\sqrt{\dfrac{\rho_I \kappa\, u_k}{\mu_{t,I}\,(I'F + z_0)}} - \dfrac{1}{2 + z_0/I'F}\right)\right]\underline{\tau} & \\ \underline{B}_{grad} = \underline{0} & \end{cases} \tag{II.3.20}$$

• **Conditions aux limites pour les variables $k$ et $\varepsilon$ (modèle $k-\varepsilon$ standard)**

On impose sur $k$ une condition de Dirichlet tirée de la vitesse de frottement $u_k$ (se reporter à l'équation (II.3.1)), soit :

$$k = \frac{u_k^2}{C_\mu^{\frac{1}{2}}} \tag{II.3.21}$$

On cherche à imposer la dérivée normale de $\varepsilon$ à partir de la loi théorique suivante (voir les notations sur la figure II.3.2) :

$$G_{\text{théo},\varepsilon} = \frac{\partial}{\partial z}\left[\frac{u_k^3}{\kappa\,(z+z_0)}\right] \tag{II.3.22}$$

On utilise le point $M$ pour imposer une condition à la limite avec un ordre plus élevé en espace. En effet, la simple utilisation de la relation $\varepsilon_F = \varepsilon_I + d\partial_z\varepsilon_I + O(d^2)$ conduit à une précision d'ordre 1. En utilisant les développements limités suivants, on peut obtenir une précision à l'ordre 2 :

$$\left\{\begin{array}{ll} \varepsilon_M & = \varepsilon_I - \dfrac{d}{2}\partial_z\varepsilon_I + \dfrac{d^2}{8}\partial_z^2\varepsilon_I + O(d^3) \\ \varepsilon_M & = \varepsilon_F + \dfrac{d}{2}\partial_z\varepsilon_F + \dfrac{d^2}{8}\partial_z^2\varepsilon_F + O(d^3) \end{array}\right. \tag{II.3.23}$$

Par différence, ces relations conduisent à

$$\varepsilon_F = \varepsilon_I - \frac{d}{2}(\partial_z\varepsilon_I + \partial_z\varepsilon_F) + O(d^3) \tag{II.3.24}$$

De plus, on a

$$\left\{\begin{array}{ll} \partial_z\varepsilon_I & = \partial_z\varepsilon_M + d\partial_z^2\varepsilon_M + O(d^2) \\ \partial_z\varepsilon_F & = \partial_z\varepsilon_M - d\partial_z^2\varepsilon_M + O(d^2) \end{array}\right. \tag{II.3.25}$$

La somme de ces deux dernières relations permet d'établir $\partial_z\varepsilon_I + \partial_z\varepsilon_F = 2\partial_z\varepsilon_M + O(d^2)$ et, en reportant dans (II.3.24), on obtient alors une expression de $\varepsilon_F$ à l'ordre 2, comme souhaité :

$$\varepsilon_F = \varepsilon_I - d\partial_z\varepsilon_M + O(d^3) \tag{II.3.26}$$

On utilise alors la valeur théorique (II.3.22) pour évaluer $\partial_z\varepsilon_M$ et on obtient alors la valeur à imposer au bord ($d = I'F$) :

$$\varepsilon_F = \varepsilon_I + d\frac{u_k^3}{\kappa\,(d/2+z_0)^2} \tag{II.3.27}$$

Cette relation est étendue au cas de maillages non orthogonaux sans précaution (ce qui doit dégrader l'ordre en espace).

On a finalement :

**Conditions aux limites sur les variables $k$ et $\varepsilon$**

$$\left\{\begin{array}{ll} k_F & = \dfrac{u_k^2}{C_\mu^{\frac{1}{2}}} \\ \varepsilon_F & = \varepsilon_{I'} + I'F\dfrac{u_k^3}{\kappa\,(I'F/2+z_0)^2} \end{array}\right. \tag{II.3.28}$$

et les coefficients associés

**Coefficients associés aux conditions aux limites sur les variables $k$ et $\varepsilon$**

$$\left\{\begin{array}{lll} A_k & = \dfrac{u_k^2}{C_\mu^{\frac{1}{2}}} & \text{et } B_k = 0 \\ A_\varepsilon & = I'F\dfrac{u_k^3}{\kappa\,(I'F/2+z_0)^2} & \text{et } B_\varepsilon = 1 \end{array}\right. \tag{II.3.29}$$

• **Conditions aux limites pour les** *VarScalaires*

On ne traite ici que les conditions se présentant sous la forme d'une valeur imposée (à la paroi ou en retrait de celle-ci avec un coefficient d'échange externe éventuel). On se reporte aux notations de la figure II.3.1 et à la présentation générale disponible dans `condli` dont on ne reprend que la partie essentielle ci-dessous.

La conservation du flux normal au bord pour la variable $f$ s'écrit sous la forme :

$$\underbrace{h_{int}(f_{b,int}-f_{I'})}_{\phi_{int}} = \underbrace{h_b(f_{b,ext}-f_{I'})}_{\phi_b} = \begin{cases} \underbrace{h_{imp,ext}(f_{imp,ext}-f_{b,ext})}_{\phi_{réel\ imposé}} & \text{(condition de Dirichlet)} \\ \underbrace{\phi_{imp,ext}}_{\phi_{réel\ imposé}} & \text{(condition de Neumann)} \end{cases} \tag{II.3.30}$$

On réarrange ces deux équations afin d'obtenir la valeur numérique $f_{b,int} = f_F$ à imposer en face de paroi, étant données les valeurs de $f_{imp,ext}$ et de $h_{imp,ext}$ fixées par l'utilisateur et la valeur $h_b$ dictée par les lois de similitude qui seront détaillées plus bas. On précise les coefficients $A$ et $B$ qui s'en déduisent naturellement.

**Conditions aux limites sur les** *VarScalaires*

$$f_{b,int} = \underbrace{\frac{h_{imp,ext}}{h_{int}+h_r h_{imp,ext}} f_{imp,ext}}_{A} + \underbrace{\frac{h_{int}+h_{imp,ext}(h_r-1)}{h_{int}+h_r h_{imp,ext}}}_{B} f_{I'} \text{ avec } h_r = \frac{h_{int}}{h_b} \tag{II.3.31}$$

**Principe de similitude : calcul de $h_b$.**

Dans l'expression (II.3.31), seule reste à déterminer la valeur de $h_b$, celle de $h_{int}$ étant une valeur numérique cohérente avec le mode de calcul des gradients aux faces et précisée dans `condli` ($h_{int} = \frac{\alpha}{\overline{I'F}}$). La valeur de $h_b$ doit permettre de relier le flux à l'écart entre les valeurs $f_{I'}$ et $f_{b,ext}$ en prenant en compte la couche limite (le profil de $f$ n'est pas toujours linéaire) :

$$\phi_b = h_b \left( f_{b,ext} - f_{I'} \right) \tag{II.3.32}$$

Les considérations suivantes sont présentées en adoptant des notations générales. En particulier, le nombre de Prandtl-Schmidt est noté $\sigma = \dfrac{\nu \, \rho \, C}{\alpha}$. Lorsque le scalaire $f$ considéré est la température, on a (voir `condli`) :

- $C = C_p$ (chaleur massique),
- $\alpha = \lambda$ (conductivité moléculaire),
- $\sigma = \dfrac{\nu \, \rho \, C_p}{\lambda} = Pr$ (nombre de Prandtl),
- $\sigma_t = Pr_t$ (nombre de Prandtl turbulent),
- $\phi = \left( \lambda + \dfrac{C_p \mu_t}{\sigma_t} \right) \dfrac{\partial T}{\partial z}$ (flux en $W m^{-2}$).

On s'est appuyé sur la référence "The atmospheric boundary layer", J. R. Garratt, Cambridge University Press.

Le flux en paroi s'écrit, pour le scalaire $f$ (le flux est positif s'il est entrant dans le domaine fluide, comme l'indique l'orientation de l'axe $z$) :

$$\phi = - \left( \alpha + C \, \frac{\mu_t}{\sigma_t} \right) \frac{\partial f}{\partial z} = - \rho \, C \left( \frac{\alpha}{\rho \, C} + \frac{\mu_t}{\rho \sigma_t} \right) \frac{\partial f}{\partial z} \tag{II.3.33}$$

Pour la température, avec $a = \dfrac{\lambda}{\rho \, C_p}$ et $a_t = \dfrac{\mu_t}{\rho \, \sigma_t}$, on a donc, de manière équivalente :

$$\phi = - \rho \, C_p (a + a_t) \frac{\partial T}{\partial z} \tag{II.3.34}$$

On introduit $f^*$ afin d'adimensionner $f$, en utilisant la valeur du flux au bord $\phi_b$ :

$$f^* = - \frac{\phi_b}{\rho \, C \, u_k} \tag{II.3.35}$$

Pour la température, on a donc :

$$T^* = - \frac{\phi_b}{\rho \, C_p \, u_k} \tag{II.3.36}$$

On rappelle que dans le cas du modèle à deux échelles de vitesse, $u_k$ est la vitesse de frottement en paroi obtenue à partir de l'énergie cinétique moyenne du mouvement turbulent[4]. Dans le cas du modèle à une échelle de vitesse, on pose $u_k = u^*$ avec $u^*$ la vitesse de frottement en paroi déterminée à partir de la loi logarithmique.

[4] $u_k = C_\mu^{\frac{1}{4}} k_I^{\frac{1}{2}}$

On divise alors les membres de l'équation (II.3.33) par $\phi_b$. Pour le membre de gauche, on simplifie en utilisant le fait que le flux se conserve et donc que $\phi = \phi_b$. Pour le membre de droite, on remplace $\phi_b$ par sa valeur $-\rho\, C\, u_k\, f^*$. Avec les notations :

$$\nu = \frac{\mu}{\rho} \qquad \nu_t = \frac{\mu_t}{\rho} \qquad f^+ = \frac{f - f_{b,ext}}{f^*} \tag{II.3.37}$$

on a :

$$1 = \left(\frac{\nu}{\sigma} + \frac{\nu_t}{\sigma_t}\right) \frac{\partial f^+}{\partial z} \frac{1}{u_k} \tag{II.3.38}$$

Remarquons dès à présent qu'avec les notations précédentes, $h_b$ s'exprime en fonction de $f^+_{I'}$ :

$$h_b = \frac{\phi_b}{f_{b,ext} - f_{I'}} = \frac{\rho\, C\, u_k}{f^+_{I'}} \tag{II.3.39}$$

Pour déterminer $h_b$, on intègre alors l'équation (II.3.38) afin de disposer de $f^+_{I'}$. L'unique difficulté consiste alors à prescrire une loi de variation de $\mathcal{K} = \dfrac{\nu}{\sigma} + \dfrac{\nu_t}{\sigma_t}$

Dans la zone turbulente pleinement développée, une hypothèse de longueur de mélange permet de modéliser les variations de $\nu_t$ :

$$\nu_t = l^2 \left|\frac{\partial U}{\partial z}\right| = \kappa\, u^*\, (z + z_0) \tag{II.3.40}$$

De plus, les effets de diffusion de $f$ (ou effets "conductifs" lorsque $f$ représente la température) sont négligeables devant les effets turbulents : on néglige alors $\dfrac{\nu}{\sigma}$ devant $\dfrac{\nu_t}{\sigma_t}$. On a donc finalement :

$$\mathcal{K} = \frac{\kappa\, u_k}{\sigma_t}\, (z + z_0) \tag{II.3.41}$$

On intègre l'équation adimensionnelle (II.3.38) sous la même hypothèse et on obtient alors la loi donnant $f^+$ :

$$f^+ = \frac{\sigma_t}{\kappa}\, ln\left(\frac{z + z_0}{z_{o_T}}\right) \tag{II.3.42}$$

où $z_{o_T}$ est la longueur de rugosité thermique. Son ordre de grandeur comparé à la rugosité dynamique est donné par la relation $ln\left(\dfrac{z_0}{z_{o_T}}\right) \approx 2$ (référence J. R. Garratt).

Pour résumer, le calcul de $h_b$ est réalisé en déterminant :

$$f^+_{I'} = \frac{\sigma_t}{\kappa}\, ln\left(\frac{I'F + z_0}{z_{o_T}}\right) \tag{II.3.43}$$

$$h_b = \frac{\phi_b}{f_{b,ext} - f_{I'}} = \frac{\rho\, C\, u_k}{f^+_{I'}} \tag{II.3.44}$$

## 3.3 Mise en œuvre

On traite ici les variables `IVAR` sur les faces `IFAC` telles que `ICODCL(IFAC,IVAR)`=5.

La vitesse de défilement (éventuellement nulle) de la paroi est tout d'abord projetée dans le plan tangent à la paroi. Ses trois composantes dans le repère de calcul sont stockées dans les tableaux `RCODCL(IFAC,IUIPH,1)`,`RCODCL(IFAC,IVIPH,1)RCODCL(IFAC,IWIPH,1)`.

On détermine ensuite le repère local $\hat{\mathcal{R}}$. Pour chaque face, il est disponible dans les vecteurs $\underline{\tau}$ `(TX, TY, TZ)`, $\underline{\tilde{n}}$ `(-RNX, -RNY, -RNZ)`, et $\underline{b}$ `(T2X, -T2Y,-T2Z)`. Par ailleurs, si la norme de la vitesse tangentielle est inférieure à la valeur arbitraire `EPZERO` ($10^{-12}$), l'indicateur `TXNO` est positionné à 0 (il vaut 1 sinon) et le vecteur $\underline{\tau}$ est pris

- arbitraire, dans le plan perpendiculaire à $\underline{\tilde{n}}$ en $R_{ij} - \varepsilon$ (on utilise les composantes de $\underline{\tilde{n}}$ pour construire $\underline{\tau}$ ; si $\underline{\tilde{n}}$ est identiquement nul, le code s'arrête) ;

- identiquement nul sinon.

On calcule ensuite les vitesses de frottement qui sont stockées dans `UET` ($= u^*$) et dans `UK` ($= u_k$). Le sous-programme `causta` permet de calculer la vitesse de frottement pour le modèle à une échelle de vitesse (`IDEUCH`=0). Pour le modèle à deux échelles (`IDEUCH`=1), le calcul (plus simple) de `UET` et de `UK` est fait directement dans `clptur`. Le sous-programme utilisateur `usruet` permet alors de donner la main à l'utilisateur qui souhaiterait modifier les échelles de vitesses (prise en compte d'une paroi rugueuse, ou de variantes de la loi logarithmique par exemple).

Dans le cas ou le modèle à une échelle de vitesse est actif, on impose `UK=UET` afin de conserver la cohérence du codage dans la suite du sous-programme.

Les conditions aux limites pour la vitesse sont ensuite complétées.

- En $k - \varepsilon$, on affecte, pour les trois composantes de vitesse respectivement, à

  `COEFA(IFAC,ICLU)`, `COEFA(IFAC,ICLV)` et `COEFA(IFAC,ICLW)`

  les coefficients $A_{flux}$ issus de l'analyse relative à la contrainte tangentielle.
  De même, les coefficients $B_{flux}$ sont affectés à

  `COEFB(IFAC,ICLU)`, `COEFB(IFAC,ICLV)` et `COEFB(IFAC,ICLW)`.

  Les conditions aux limites issues de l'analyse portant directement sur le gradient de vitesse $A_{grad}$ et $B_{grad}$ sont affectées à

  `COEFA(IFAC,ICLUF)`, `COEFA(IFAC,ICLVF)`, `COEFA(IFAC,ICLWF)`

  et

  `COEFB(IFAC,ICLUF)`, `COEFB(IFAC,ICLVF)`, `COEFB(IFAC,ICLWF)`.

- Lorsque la vitesse tangentielle en $I'$ est inférieure à `EPZERO`, l'indicateur `TXN0` est annulé (sinon, `TXN0` vaut 1). Cet indicateur est utilisé pour annuler les coefficients $A$ et imposer des conditions d'adhérence.

- la vitesse de défilement de la paroi est prise en compte dans les conditions aux limites (coefficients `COEFA`).

Les conditions sur les grandeurs turbulentes sont ensuite complétées.

Les conditions aux limites pour les scalaires sont ensuite complétées. Les coefficients `COEFA` et `COEFB` sont simplement renseignés en utilisant les conditions aux limites décrites précédemment. La seule difficulté consiste à gérer correctement les différentes grandeurs permettant de calculer le coefficient d'échange $h_b$ sans erreur.

L'indicateur `ISCSTH` sert, pour chaque *VarScalaires* à indiquer quelle valeur de $C$ utiliser au moment du traitement des conditions aux limites. Ainsi, pour `ISCSTH`=1, la variable doit être traitée comme une température, avec $C = C_p$. Pour `ISCSTH`=0 ou 2, la variable doit être traitée comme un scalaire passif ou une enthalpie respectivement, avec $C = 1$ (constante sans dimension) dans les deux cas.

Par ailleurs, une valeur strictement positive de l'entier `IPCCP` indique que $C_p$ est variable en espace et disponible dans le tableau `PROPCE(IEL,IPCCP )` (renseigné dans `USPHYV`). Lorsque `IPCCP` est nul, $C_p$ est constant et disponible sous forme du réel `CP0(IPHAS)`.

L'indicateur `IHCP` permet de rassembler ces informations :

- `IHCP` $= 0$ : `CPP` $= C = 1$

- IHCP = 1 : CPP = $C = C_p$ uniforme en espace
- IHCP = 2 : CPP = $C = C_p$ variable en espace

Pour la *VarScalaire* LL, l'indicateur IVISLS(LL) permet également de repérer si $\frac{\alpha_m}{C}$ est variable en espace et disponible dans le tableau PROPCE(IEL,IPCVSL) (IVISLS $> 0$) ou uniforme en espace et disponible sous forme du réel VISLS0(LL) (IVISLS $= 0$). On pose RKL= $\frac{\alpha_m}{C}$

Le nombre de Prandtl local est alors calculé PRDTL = $\frac{\mu}{\alpha_m/C}$ ($\mu$ est la viscosité dynamique moléculaire disponible dans VISCLC).

Le coefficient $h_{int} = \frac{\alpha}{\overline{I'F}}$ est ensuite déterminé et conservé dans HINT.

Lorsqu'un modèle de turbulence est activé, on calcule HFLUI $= h_b = \frac{\rho C u_k}{T^+}$. Si le calcul est réalisé en laminaire, on a simplement HFLUI = HINT ($= h_b = h_{int} = \frac{\alpha}{\overline{I'F}}$)

Dans les cas où l'on souhaite stocker le coefficient d'échange (couplage avec SYRTHES), HFLUI est conservé dans le tableau HBORD. Dans les cas où l'on utilise le module de rayonnement, HFLUI est stocké dans le tableau de travail RA(IHCONV).

On dispose alors de tous les élements pour calculer les coefficients $A$ et $B$ (COEFA et COEFB) relatif à la variable traitée. Noter pour terminer les correspondances suivantes qui permettent de rapprocher le code source de la relation (II.3.31) :

$$\begin{aligned} \texttt{HEXT} &= h_{imp,ext} \\ \texttt{PIMP} &= f_{imp,ext} \\ \texttt{HREDUI} &= h_r \\ \texttt{HINT} &= h_{int} \\ \texttt{HFLUI} &= h_b \end{aligned}$$

# 4- Sous-programme clsyvt

## 4.1 Function

The aim of this subroutine is to fill the arrays of boundary conditions (COEFA and COEFB) of the velocity and of the Reynolds stress tensor, for the symmetry boundary faces. These conditions are express relatively naturally in the local coordinate system of the boundary face. The function of clsyvt is then to transform these natural boundary conditions (expressed in the local coordinate sytem) in the general coordinate sytem, and then to possibly partly implicit them.

It should be noted that the part of the subroutine clptur (for the wall boundary conditions) relative to the writing in the local coordinate system and to the rotation is totally identical.

## 4.2 Discretisation

Figure II.4.1 presents the notations used at the face. The local coordinate system is defined from the normal at the face and the velocity at $I'$ :

- $\underline{t} = \dfrac{1}{|\underline{u}_{I',\tau}|}\underline{u}_{I',\tau}$ is the first vector of the local coordinate system.
- $\underline{\tilde{n}} = -\underline{n}$ is the second vector of the local coordinate system.
- $\underline{b} = \underline{t} \wedge \underline{\tilde{n}} = \underline{n} \wedge \underline{t}$ is the third vector of the local coordinate system.

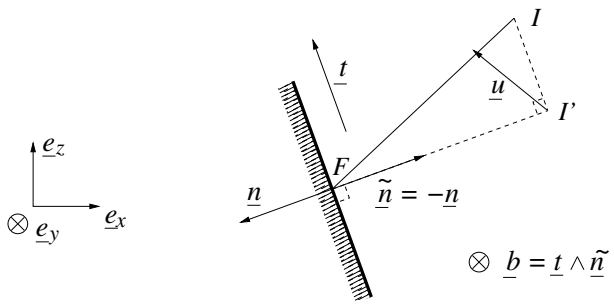



Figure II.4.1: Definition of the vectors forming the local coordinate system.

Here, $\underline{n}$ is the normalized normal vector to the boundary face in the sense of *Code_Saturne* (*i.e.* directed towards the outside of the computational domain) and $\underline{u}_{I',\tau}$ is the projection of the velocity at $I'$ in the plane of the face : $\underline{u}_{I',\tau} = \underline{u}_{I'} - (\underline{u}_{I'}.\underline{n})\underline{n}$.
If $\underline{u}_{I',\tau} = \underline{0}$, the direction of $\underline{t}$ in the plane normal to $\underline{n}$ is irrelevant. thus it is defined as : $\underline{t} = \dfrac{1}{\sqrt{n_y^2 + n_z^2}}(n_z\underline{e}_y - n_y\underline{e}_z)$ where $\underline{t} = \dfrac{1}{\sqrt{n_x^2 + n_z^2}}(n_z\underline{e}_x - n_x\underline{e}_z)$ along the non-zero components of $\underline{n}$ (components in the global coordinate system $(\underline{e}_x, \underline{e}_y, \underline{e}_z)$).

For sake of clarity, the following notations will be used :

- The general coordinate system will write $\mathcal{R} = (\underline{e}_x, \underline{e}_y, \underline{e}_z)$.
- The local coordinate system will write $\hat{\mathcal{R}} = (\underline{t}, -\underline{n}, \underline{b}) = (\underline{t}, \underline{\tilde{n}}, \underline{b})$.
- The matrices of the components of a vector $\underline{u}$ in the coordinate systems $\mathcal{R}$ and $\hat{\mathcal{R}}$ will write $\underline{\mathrm{U}}$ and $\underline{\hat{\mathrm{U}}}$, respectively.
- The matrices of the components of a tensor $\underline{\underline{R}}$ ($2^{nd}$ order) in the coordinate systems $\mathcal{R}$ and $\hat{\mathcal{R}}$ will

write $\underline{\underline{\mathrm{R}}}$ and $\hat{\underline{\underline{\mathrm{R}}}}$, respectively.

- $\underline{\underline{\mathrm{P}}}$ refers to the (orthogonal) matrix transforming $\mathcal{R}$ into $\hat{\mathcal{R}}$.

$$\underline{\underline{\mathrm{P}}} = \begin{bmatrix} t_x & -n_x & b_x \\ t_y & -n_y & b_y \\ t_z & -n_z & b_z \end{bmatrix} \tag{II.4.1}$$

($\underline{\underline{\mathrm{P}}}$ being orthogonal, $\underline{\underline{\mathrm{P}}}^{-1} = {}^t\underline{\underline{\mathrm{P}}}$).

In particular, we have for any vector $\underline{u}$ and for any second order tensor $\underline{\underline{R}}$ :

$$\begin{cases} \underline{\mathrm{U}} = \underline{\underline{\mathrm{P}}} \,.\, \hat{\underline{\mathrm{U}}} \\ \underline{\underline{\mathrm{R}}} = \underline{\underline{\mathrm{P}}} \,.\, \hat{\underline{\underline{\mathrm{R}}}} \,.\, {}^t\underline{\underline{\mathrm{P}}} \end{cases} \tag{II.4.2}$$

Treatment of the velocity

In the local coordinate system, the boundary conditions for $\underline{u}$ naturally write :

$$\begin{cases} u_{F,t} &= u_{I',t} \\ u_{F,\tilde{n}} &= 0 \\ u_{F,b} &= u_{I',b} \end{cases} \tag{II.4.3}$$

or

$$\underline{\mathrm{U}}_F = \underline{\underline{\mathrm{P}}} \,.\, \hat{\underline{\mathrm{U}}}_F = \underline{\underline{\mathrm{P}}} \,.\, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \,.\, \hat{\underline{\mathrm{U}}}_{I'} = \underline{\underline{\mathrm{P}}} \,.\, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \,.\, {}^t\underline{\underline{\mathrm{P}}} \,.\, \underline{\mathrm{U}}_{I'} \tag{II.4.4}$$

Let's take $\underline{\underline{\mathrm{A}}} = \underline{\underline{\mathrm{P}}} \,.\, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \,.\, {}^t\underline{\underline{\mathrm{P}}}$ (matrix in the coordinate system $\mathcal{R}$ of the projector orthogonal to the face).

The boundary conditions for $\underline{u}$ thus write :

$$\underline{\mathrm{U}}_F = \underline{\underline{\mathrm{A}}} \,.\, \underline{\mathrm{U}}_{I'} \tag{II.4.5}$$

Since the matrix $\underline{\underline{\mathrm{P}}}$ is orthogonal, it can be shown that

$$\underline{\underline{\mathrm{A}}} = \begin{bmatrix} 1-\tilde{n}_x^2 & -\tilde{n}_x\tilde{n}_y & -\tilde{n}_x\tilde{n}_z \\ -\tilde{n}_x\tilde{n}_y & 1-\tilde{n}_y^2 & -\tilde{n}_y\tilde{n}_z \\ -\tilde{n}_x\tilde{n}_z & -\tilde{n}_y\tilde{n}_z & 1-\tilde{n}_z^2 \end{bmatrix} \tag{II.4.6}$$

The boundary conditions can then be partially implicited:

$$u_{F,x}^{(n+1)} = \underbrace{1-\tilde{n}_x^2}_{\texttt{COEFB}} u_{I',x}^{(n+1)} \underbrace{-\tilde{n}_x\tilde{n}_y u_{I',y}^{(n)} - \tilde{n}_x\tilde{n}_z u_{I',z}^{(n)}}_{\texttt{COEFA}} \tag{II.4.7}$$

The other components have a similar treatment. Since only the coordinates of $\underline{n}$ are useful, we do not need (for $\underline{u}$) to define explicitly the vectors $\underline{t}$ and $\underline{b}$.

Treatment of the Reynolds stress tensor

We saw that we have the following relation:

$$\underline{\underline{\mathrm{R}}} = \underline{\underline{\mathrm{P}}} \,.\, \hat{\underline{\underline{\mathrm{R}}}} \,.\, {}^t\underline{\underline{\mathrm{P}}} \tag{II.4.8}$$

The boundary conditions we want to write are relations of the type:

$$\mathrm{R}_{F,ij} = \sum_{k,l} \alpha_{ijkl} \mathrm{R}_{I',kl} \tag{II.4.9}$$

We are then naturally brought to introduce the column matrices of the components of $\underline{\underline{R}}$ in the different coordinate systems.

We write

$$\underline{\mathrm{S}} = {}^t[\mathrm{R}_{11}, \mathrm{R}_{12}, \mathrm{R}_{13}, \mathrm{R}_{21}, \mathrm{R}_{22}, \mathrm{R}_{23}, \mathrm{R}_{31}, \mathrm{R}_{32}, \mathrm{R}_{33}] \tag{II.4.10}$$

and

$$\hat{\underline{\mathrm{S}}} = {}^t[\hat{\mathrm{R}}_{11}, \hat{\mathrm{R}}_{12}, \hat{\mathrm{R}}_{13}, \hat{\mathrm{R}}_{21}, \hat{\mathrm{R}}_{22}, \hat{\mathrm{R}}_{23}, \hat{\mathrm{R}}_{31}, \hat{\mathrm{R}}_{32}, \hat{\mathrm{R}}_{33}] \tag{II.4.11}$$

Two functions $q$ and $r$ from $\{1,2,3,4,5,6,7,8,9\}$ to $\{1,2,3\}$ are defined. Their values are given in the following table :

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $q(i)$ | 1 | 1 | 1 | 2 | 2 | 2 | 3 | 3 | 3 |
| $r(i)$ | 1 | 2 | 3 | 1 | 2 | 3 | 1 | 2 | 3 |

$i \longmapsto (q(i), r(i))$ is then a bijection from $\{1,2,3,4,5,6,7,8,9\}$ to $\{1,2,3\}^2$, and we have :

$$\left\{ \begin{array}{l} \mathrm{R}_{ij} = \mathrm{S}_{3(i-1)+j} \\ \mathrm{S}_i = \mathrm{R}_{q(i)r(i)} \end{array} \right. \tag{II.4.12}$$

Using equation II.4.8, we thus have :

$$\begin{aligned} \mathrm{S}_{F,i} &= \mathrm{R}_{F,q(i)r(i)} = \sum_{(m,n)\in\{1,2,3\}^2} \mathrm{P}_{q(i)m} \hat{\mathrm{R}}_{F,mn} \mathrm{P}_{r(i)n} \\ &= \sum_{j=1}^{9} \mathrm{P}_{q(i)q(j)} \hat{\mathrm{R}}_{F,q(j)r(j)} \mathrm{P}_{r(i)r(j)} \quad \text{(d'après la bijectivité de } (q,r)) \\ &= \sum_{j=1}^{9} \mathrm{P}_{q(i)q(j)} \mathrm{P}_{r(i)r(j)} \hat{\mathrm{S}}_{F,j} \end{aligned} \tag{II.4.13}$$

Or

$$\underline{\mathrm{S}}_F = \underline{\underline{\mathrm{A}}} \,.\, \hat{\underline{\mathrm{S}}}_F \quad \text{avec } \mathrm{A}_{ij} = \mathrm{P}_{q(i)q(j)} \mathrm{P}_{r(i)r(j)} \tag{II.4.14}$$

It can be shown that $\underline{\underline{\mathrm{A}}}$ is an orthogonal matrix (see Annexe A).

In the local coordinate system, the boundary conditions of $\underline{\underline{R}}$ write naturally[1].

$$\left\{ \begin{array}{lll} \hat{\mathrm{R}}_{F,11} = \hat{\mathrm{R}}_{I',11} & \hat{\mathrm{R}}_{F,21} = 0 & \hat{\mathrm{R}}_{F,31} = B\hat{\mathrm{R}}_{I',31} \\ \hat{\mathrm{R}}_{F,12} = 0 & \hat{\mathrm{R}}_{F,22} = \hat{\mathrm{R}}_{I',22} & \hat{\mathrm{R}}_{F,32} = 0 \\ \hat{\mathrm{R}}_{F,13} = B\hat{\mathrm{R}}_{I',13} & \hat{\mathrm{R}}_{F,23} = 0 & \hat{\mathrm{R}}_{F,33} = \hat{\mathrm{R}}_{I',33} \end{array} \right. \tag{II.4.15}$$

[1] cf. Davroux A., Archambeau F., *Le $R_{ij} - \varepsilon$ dans Code_Saturne (version $\beta$)*, HI-83/00/030/A

or

$$\hat{\underline{S}}_F = \underline{\underline{B}} \,.\, \hat{\underline{S}}_{I'} \qquad \text{avec } \underline{\underline{B}} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & 0 & \ddots & & & & & & \vdots \\ \vdots & \ddots & B & \ddots & & & & & \vdots \\ \vdots & & \ddots & 0 & \ddots & & & & \vdots \\ \vdots & & & \ddots & 1 & \ddots & & & \vdots \\ \vdots & & & & \ddots & 0 & \ddots & & \vdots \\ \vdots & & & & & \ddots & B & \ddots & \vdots \\ \vdots & & & & & & \ddots & 0 & 0 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & 1 \end{bmatrix} \tag{II.4.16}$$

For the symmetry faces which are treated by `clsyvt`, the coefficient $B$ is 1. However a similar treatment is applied in `clptur` for the wall faces, and in this $B$ is zero. This parameter has to be specified when `clca66` is called (see §4.3).

Back in the global coordinate system, the following formulae is finally obtained :

$$\underline{S}_F = \underline{\underline{C}} \,.\, \underline{S}_{I'} \qquad \text{avec } \underline{\underline{C}} = \underline{\underline{A}} \,.\, \underline{\underline{B}} \,.\, {}^t\underline{\underline{A}} \tag{II.4.17}$$

It can be shown that the components of the matrix $\underline{\underline{C}}$ are :

$$C_{ij} = \sum_{k=1}^{9} P_{q(i)q(k)} P_{r(i)r(k)} P_{q(j)q(k)} P_{r(j)r(k)} (\delta_{k1} + B\delta_{k3} + \delta_{k5} + B\delta_{k7} + \delta_{k9}) \tag{II.4.18}$$

To conclude, it can be noted that, due to the symmetries of the tensor $\underline{\underline{R}}$, the matrix $\underline{S}$. Thus only the simplified matrices $\underline{S}'$ and $\hat{\underline{S}}'$ will be used:

$$\underline{S}' = {}^t[R_{11}, R_{22}, R_{33}, R_{12}, R_{13}, R_{23}] \tag{II.4.19}$$

$$\hat{\underline{S}}' = {}^t[\hat{R}_{11}, \hat{R}_{22}, \hat{R}_{33}, \hat{R}_{12}, \hat{R}_{13}, \hat{R}_{23}] \tag{II.4.20}$$

By gathering different lines of matrix $\underline{\underline{C}}$, equation II.4.17 is transformed into the final equation :

$$\underline{S}'_F = \underline{\underline{D}} \,.\, \underline{S}'_{I'} \tag{II.4.21}$$

The computation of the matrix $\underline{\underline{D}}$ is performed in the subroutine `clca66`. The methodology is described in annexe B.

From $\underline{\underline{D}}$, the coefficients of the boundary conditions can be partially implicited (`ICLSYR` = 1) or totally explicited (`ICLSYR` = 0).

• Partial implicitation

$$S'_{F,i}{}^{(n+1)} = \underbrace{D_{ii}}_{\texttt{COEFB}} S'_{I',i}{}^{(n+1)} + \underbrace{\sum_{j \neq i} D_{ij} S'_{I',j}{}^{(n)}}_{\texttt{COEFA}} \tag{II.4.22}$$

• Total explicitation

$$S'_{F,i}{}^{(n+1)} = \underbrace{\sum_{j} D_{ij} S'_{I',j}{}^{(n)}}_{\texttt{COEFA}} \qquad (\texttt{COEFB} = 0) \tag{II.4.23}$$

## 4.3 Implementation

**• Beginning of the loop**

Beginnning of the loop on all the boundary faces `IFAC` with symmetry conditions. A face is considered as a symmetry face if `ICODCL(IFAC,IU(IPHAS))` is 4. The tests in `vericl` are designed for `ICODCL` to be equal to 4 for `IU` if and only if it is equal to 4 for the other components of the velocity and for the components of $\underline{\underline{R}}$ (if necessary)
.

The value 0 is given to `ISYMPA`, which identifies the face as a wall or symmetry face (a face where the mass flux will be set to zero as explained in `inimas`).

**• Calculation of the basis vectors**
The normal vector $\underline{n}$ is stored in `(RNX,RNY,RNZ)`.
$\underline{u}_{I'}$ is calculated in `CONDLI`, passed *via* `COEFU`, and stored in `(UPX,UPY,UPZ)`.

**• Case with $R_{ij} - \varepsilon$**
With the $R_{ij} - \varepsilon$ model (`ITURB`=30 or 31), the vectors $\underline{t}$ and $\underline{b}$ must be calculated explicitly (we use $\underline{\underline{P}}$, not simply $\underline{\underline{A}}$). The are stored in `(TX,TY,TZ)` and `(T2X,T2Y,T2Z)`, respectively.
The transform matrix $\underline{\underline{P}}$ is then calculated and stored in the array `ELOGLO`.
The subroutine `clca66` is then called to calculate the reduced matrix $\underline{\underline{D}}$. It is stored in `ALPHA`. `clca66` is called with a parameter `CLSYME` which is 1, and which corresponds to the parameter $\omega$ of equation II.4.15.

**• Filling the arrays `COEFA` and `COEFB`**
The arrays `COEFA` and `COEFB` are filled following directly equations II.4.7, II.4.22 and II.4.23.
`RIJIPB(IFAC,.)` corresponds to the vector $\underline{S}'_{I'}$, computed in `condli`, and passed as an argument to `clsyvt`.

**• Filling the arrays `COEFAF` and `COEFBF`**
If they are defined, the arrays `COEFAF` and `COEFBF` are filled. They contain the same values as `COEFA` and `COEFB`, respectively.

## 4.4 Annexe A

PROOF OF THE ORTHOGONALITY OF MATRIX $\underline{\underline{A}}$

All the notations used in paragraphe 2 are kept. We have :

$$\begin{aligned}
({}^t\underline{\underline{A}}\,.\,\underline{\underline{A}})_{ij} &= \sum_{k=1}^{9} A_{ki}A_{kj} \\
&= \sum_{k=1}^{9} P_{q(k)q(i)}P_{r(k)r(i)}P_{q(k)q(j)}P_{r(k)r(j)} \qquad \text{(II.4.24)}
\end{aligned}$$

When $k$ varies from 1 to 3, $q(k)$ remains equal to 1 and $r(k)$ varies from 1 to 3. We thus have :

$$\begin{aligned}\sum_{k=1}^{3} \mathrm{P}_{q(k)q(i)}\mathrm{P}_{r(k)r(i)}\mathrm{P}_{q(k)q(j)}\mathrm{P}_{r(k)r(j)} &= \mathrm{P}_{1q(i)}\mathrm{P}_{1q(j)}\sum_{k=1}^{3}\mathrm{P}_{r(k)r(i)}\mathrm{P}_{r(k)r(j)} \\ &= \mathrm{P}_{1q(i)}\mathrm{P}_{1q(j)}\sum_{k=1}^{3}\mathrm{P}_{kr(i)}\mathrm{P}_{kr(j)} \qquad \text{(II.4.25)} \\ &= \mathrm{P}_{1q(i)}\mathrm{P}_{1q(j)}\delta_{r(i)r(j)} \qquad \text{(by orthogonality of } \underline{\underline{\mathrm{P}}}\text{)}\end{aligned}$$

Likewise for $k$ varying from 4 to 6 or from 7 to 9, $q(k)$ being 2 or 3, respectively, we obtain :

$$\begin{aligned}({}^{t}\underline{\underline{\mathrm{A}}}\cdot\underline{\underline{\mathrm{A}}})_{ij} &= \sum_{k=1}^{9}\mathrm{P}_{q(k)q(i)}\mathrm{P}_{r(k)r(i)}\mathrm{P}_{q(k)q(j)}\mathrm{P}_{r(k)r(j)} \\ &= \sum_{k=1}^{3}\mathrm{P}_{kq(i)}\mathrm{P}_{kq(j)}\delta_{r(i)r(j)} \qquad \text{(II.4.26)} \\ &= \delta_{q(i)q(j)}\delta_{r(i)r(j)} \\ &= \delta_{ij} \qquad \text{(by the bijectivity of } (q,r)\text{)}\end{aligned}$$

Thus ${}^{t}\underline{\underline{\mathrm{A}}}\cdot\underline{\underline{\mathrm{A}}} = \underline{\underline{\mathrm{Id}}}$. Similarly, it can be shown that $\underline{\underline{\mathrm{A}}}\cdot{}^{t}\underline{\underline{\mathrm{A}}} = \underline{\underline{\mathrm{Id}}}$. Thus $\underline{\underline{\mathrm{A}}}$ is an orthogonal matrix.

## 4.5 Annexe B

### Calculation of the matrix $\underline{\underline{\mathrm{D}}}$

The relation between the matrices of dimension $9\times 1$ of the components of $\underline{R}$ in the coordinate system $\mathcal{R}$ at $F$ and at $I'$ (matrices $\underline{\mathrm{S}}_F$ and $\underline{\mathrm{S}}_{I'}$) :

$$\underline{\mathrm{S}}_F = \underline{\underline{\mathrm{C}}}\cdot\underline{\mathrm{S}}_{I'} \qquad \text{(II.4.27)}$$

with

$$\mathrm{C}_{ij} = \sum_{k=1}^{9}\mathrm{P}_{q(i)q(k)}\mathrm{P}_{r(i)r(k)}\mathrm{P}_{q(j)q(k)}\mathrm{P}_{r(j)r(k)}(\delta_{k1}+\omega\delta_{k3}+\delta_{k5}+\omega\delta_{k7}+\delta_{k9}) \qquad \text{(II.4.28)}$$

To transform $\underline{\mathrm{S}}$ into the matrix $6\times 1$ $\underline{\mathrm{S}}'$, the function $s$ from $\{1,2,3,4,5,6,7,8,9\}$ to $\{1,2,3,4,5,6\}$ is defined. It takes the following values :

| $i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| $s(i)$ | 1 | 4 | 5 | 4 | 2 | 6 | 5 | 6 | 3 |

By construction, we have $\mathrm{S}_i = \mathrm{S}'_{s(i)}$ for all $i$ between 1 and 9.

To compute $\mathrm{D}_{ij}$, we can choose a value of $m$ to satisfy $s(m) = i$ and sum all the $\mathrm{C}_{mn}$ to have $s(n) = j$. The choice of $m$ is irrelevent. We can also compute the sum over all $m$ so that $s(m) = i$ and then divide by the number of such values of $m$. We will use this method.

We define $N(i)$ the number of integers between 1 and 9 so that $s(m) = i$. According to the preceeding, we thus have

$$\begin{aligned} \mathrm{D}_{ij} &= \frac{1}{N(i)} \sum_{\substack{s(m)=i \\ s(n)=j}} \mathrm{C}_{mn} \\ &= \frac{1}{N(i)} \sum_{\substack{s(m)=i \\ s(n)=j \\ 1 \leqslant k \leqslant 9}} \mathrm{P}_{q(m)q(k)} \mathrm{P}_{r(m)r(k)} \mathrm{P}_{q(n)q(k)} \mathrm{P}_{r(n)r(k)} (\delta_{k1} + \omega\delta_{k3} + \delta_{k5} + \omega\delta_{k7} + \delta_{k9}) \end{aligned} \quad \text{(II.4.29)}$$

• First case : $i \leqslant 3$ and $j \leqslant 3$
In this case, we have $N(i) = N(j) = 1$. Additionally, if $s(m) = i$ and $s(n) = j$, then $q(m) = r(m) = i$ and $q(n) = r(n) = j$. Thus

$$\mathrm{D}_{ij} = \sum_{k=1}^{9} \mathrm{P}_{iq(k)} \mathrm{P}_{ir(k)} \mathrm{P}_{jq(k)} \mathrm{P}_{jr(k)} (\delta_{k1} + \omega\delta_{k3} + \delta_{k5} + \omega\delta_{k7} + \delta_{k9}) \quad \text{(II.4.30)}$$

When $k$ belongs to $\{1, 5, 9\}$, $q(k) = r(k)$ belongs to $\{1, 2, 3\}$. And for $k = 3$ or $k = 7$, $q(k) = 1$ and $r(k) = 3$, or the inverse (and for $k$ even the the sum of Kronecker symbol is zero). Finally we have :

$$\mathrm{D}_{ij} = \sum_{k=1}^{3} \mathrm{P}_{ik}^2 \mathrm{P}_{jk}^2 + 2\omega \mathrm{P}_{j1} \mathrm{P}_{i3} \mathrm{P}_{i1} \mathrm{P}_{j3} \quad \text{(II.4.31)}$$

• Second case : $i \leqslant 3$ and $j \geqslant 4$
Again we have $N(i) = 1$, and if $s(m) = i$ then $q(m) = r(m) = i$.
On the contrary, we have $N(j) = 2$, the two possibilities being $m_1$ and $m_2$.

- if $j = 4$, then $m_1 = 2$ and $m_2 = 4$, $q(m_1) = r(m_2) = 1$ and $r(m_1) = q(m_2) = 2$. We then have $m = 1$ and $n = 2$.
- if $j = 5$, then $m_1 = 3$ and $m_2 = 7$, $q(m_1) = r(m_2) = 1$ and $r(m_1) = q(m_2) = 3$. We then have $m = 1$ and $n = 3$.
- if $j = 6$, then $m_1 = 6$ and $m_2 = 8$, $q(m_1) = r(m_2) = 2$ and $r(m_1) = q(m_2) = 3$. We then have $m = 2$ and $n = 3$.

And we have :

$$\mathrm{D}_{ij} = \sum_{k=1}^{9} \mathrm{P}_{iq(k)} \mathrm{P}_{ir(k)} \left[ \mathrm{P}_{mq(k)} \mathrm{P}_{nr(k)} + \mathrm{P}_{nq(k)} \mathrm{P}_{mr(k)} \right] (\delta_{k1} + \omega\delta_{k3} + \delta_{k5} + \omega\delta_{k7} + \delta_{k9}) \quad \text{(II.4.32)}$$

But when $k$ is in $\{1, 5, 9\}$, $q(k) = r(k)$ is in $\{1, 2, 3\}$. Thus :

$$\mathrm{D}_{ij} = 2 \sum_{k=1}^{3} \mathrm{P}_{ik}^2 \mathrm{P}_{mk} \mathrm{P}_{nk} + \omega \sum_{k=1}^{9} \mathrm{P}_{iq(k)} \mathrm{P}_{ir(k)} \left[ \mathrm{P}_{mq(k)} \mathrm{P}_{nr(k)} + \mathrm{P}_{nq(k)} \mathrm{P}_{mr(k)} \right] (\delta_{k3} + \delta_{k7}) \quad \text{(II.4.33)}$$

And for $k = 3$ or $k = 7$, $q(k) = 1$ and $r(k) = 3$, or the opposite. We finally have :

$$\mathrm{D}_{ij} = 2 \left[ \sum_{k=1}^{3} \mathrm{P}_{ik}^2 \mathrm{P}_{mk} \mathrm{P}_{nk} + \omega \mathrm{P}_{i1} \mathrm{P}_{i3} \left( \mathrm{P}_{m1} \mathrm{P}_{n3} + \mathrm{P}_{n1} \mathrm{P}_{m3} \right) \right] \quad \text{(II.4.34)}$$

with $(m,n) = (1,2)$ if $j = 4$, $(m,n) = (1,3)$ if $j = 5$ and $(m,n) = (2,3)$ if $j = 6$.

• THIRD CASE : $i \geqslant 4$ and $j \leqslant 3$
By symmetry of $\underline{\underline{\mathrm{C}}}$, we obtain a result which is the symmetric of the second case, except that $N(i)$ is now 2. Thus :

$$\mathrm{D}_{ij} = \sum_{k=1}^{3} \mathrm{P}_{jk}^2 \mathrm{P}_{mk}\mathrm{P}_{nk} + \omega \mathrm{P}_{j1}\mathrm{P}_{j3}\left(\mathrm{P}_{m1}\mathrm{P}_{n3} + \mathrm{P}_{n1}\mathrm{P}_{m3}\right) \tag{II.4.35}$$

with $(m,n) = (1,2)$ if $i = 4$, $(m,n) = (1,3)$ if $i = 5$ and $(m,n) = (2,3)$ if $i = 6$.

• FOURTH CASE : $i \geqslant 4$ and $j \geqslant 4$
Then $N(i) = N(j) = 2$.
We have $(m,n) = (1,2)$ if $i = 4$, $(m,n) = (1,3)$ if $i = 5$ and $(m,n) = (2,3)$ if $i = 6$. Likewise we define $m'$ and $n'$ as a function of $j$. We then obtain :

$$\begin{aligned}
\mathrm{D}_{ij} &= \frac{1}{2}\sum_{k=1}^{9}\left(\mathrm{P}_{mq(k)}\mathrm{P}_{nr(k)} + \mathrm{P}_{nq(k)}\mathrm{P}_{mr(k)}\right)\left(\mathrm{P}_{m'q(k)}\mathrm{P}_{n'r(k)} + \mathrm{P}_{n'q(k)}\mathrm{P}_{m'r(k)}\right) \\
&\qquad \times(\delta_{k1} + \omega\delta_{k3} + \delta_{k5} + \omega\delta_{k7} + \delta_{k9}) \\
&= \frac{1}{2}\left[\sum_{k=1}^{3} 4\mathrm{P}_{mk}\mathrm{P}_{nk}\mathrm{P}_{m'k}\mathrm{P}_{n'k} + 2\omega\left(\mathrm{P}_{m1}\mathrm{P}_{n3} + \mathrm{P}_{n1}\mathrm{P}_{m3}\right)\left(\mathrm{P}_{m'1}\mathrm{P}_{n'3} + \mathrm{P}_{n'1}\mathrm{P}_{m'3}\right)\right]
\end{aligned} \tag{II.4.36}$$

and finally :

$$\mathrm{D}_{ij} = 2\sum_{k=1}^{3} \mathrm{P}_{mk}\mathrm{P}_{nk}\mathrm{P}_{m'k}\mathrm{P}_{n'k} + \omega\left(\mathrm{P}_{m1}\mathrm{P}_{n3} + \mathrm{P}_{n1}\mathrm{P}_{m3}\right)\left(\mathrm{P}_{m'1}\mathrm{P}_{n'3} + \mathrm{P}_{n'1}\mathrm{P}_{m'3}\right) \tag{II.4.37}$$

with $(m,n) = (1,2)$ if $i = 4$, $(m,n) = (1,3)$ if $i = 5$ and $(m,n) = (2,3)$ if $i = 6$
and $(m',n') = (1,2)$ if $j = 4$, $(m',n') = (1,3)$ if $j = 5$ and $(m',n') = (2,3)$ if $j = 6$.

# 5- Sous-programme `codits`

## 5.1 Fonction

Ce sous-programme, appelé entre autre par `preduv`, `turbke`, `covofi`, `resrij`, `reseps`, ..., résout les équations de convection-diffusion d'un scalaire $a$ avec termes sources du type :

$$\begin{aligned} & f_s^{imp}(a^{n+1}-a^n)+\theta \underbrace{\mathrm{div}((\rho\underline{u})\,a^{n+1})}_{\text{convection implicite}} -\theta \underbrace{\mathrm{div}(\mu_{tot}\,\underline{\mathrm{grad}}\ a^{n+1})}_{\text{diffusion implicite}} \\ & = f_s^{exp}-(1-\theta) \underbrace{\mathrm{div}((\rho\underline{u})\,a^{n})}_{\text{convection explicite}} +(1-\theta) \underbrace{\mathrm{div}(\mu_{tot}\,\underline{\mathrm{grad}}\ a^{n})}_{\text{diffusion explicite}} \end{aligned} \tag{II.5.1}$$

où $\rho\underline{u}$, $f_s^{exp}$ et $f_s^{imp}$ désignent respectivement le flux de masse, les termes sources explicites et les termes linéarisés en $a^{n+1}$. $a$ est un scalaire défini sur toutes les cellules[1]. Par souci de clarté on suppose, en l'absence d'indication, les propriétes physiques $\Phi$ (viscosité totale $\mu_{tot}$,...) et le flux de masse $(\rho\underline{u})$ pris respectivement aux instants $n+\theta_\Phi$ et $n+\theta_F$, où $\theta_\Phi$ et $\theta_F$ dépendent des schémas en temps spécifiquement utilisés pour ces grandeurs[2].
L'écriture des termes de convection et diffusion en maillage non orthogonal engendre des difficultés (termes de reconstruction et test de pente) qui sont contournées en utilisant une méthode itérative dont la limite, si elle existe, est la solution de l'équation précédente.

## 5.2 Discrétisation

Afin d'expliquer la procédure utilisée pour traiter les difficultés dues aux termes de reconstruction et de test de pente dans les termes de convection-diffusion, on note, de façon analogue à ce qui est défini dans `navsto` mais sans discrétisation spatiale associée, $\mathcal{E}_n$ l'opérateur :

$$\begin{aligned} \mathcal{E}_n(a) = & f_s^{imp}\,a+\theta\ \mathrm{div}((\rho\underline{u})\,a)-\theta\ \mathrm{div}(\mu_{tot}\,\underline{\mathrm{grad}}\ a) \\ -f_s^{exp}-f_s^{imp}\,a^n & +(1-\theta)\ \mathrm{div}((\rho\underline{u})\,a^n)-(1-\theta)\ \mathrm{div}(\mu_{tot}\,\underline{\mathrm{grad}}\ a^n) \end{aligned} \tag{II.5.2}$$

L'équation (II.5.1) s'écrit donc :

$$\mathcal{E}_n(a^{n+1})=0 \tag{II.5.3}$$

La quantité $\mathcal{E}_n(a^{n+1})$ comprend donc :

$\rightsquigarrow$ $f_s^{imp}\,a^{n+1}$, contribution des termes différentiels d'ordre 0 linéaire en $a^{n+1}$,

$\rightsquigarrow$ $\theta\ \mathrm{div}((\rho\underline{u})\,a^{n+1})-\theta\ \mathrm{div}(\mu_{tot}\,\underline{\mathrm{grad}}\ a^{n+1})$, termes de convection-diffusion implicites complets (termes non reconstruits + termes de reconstruction) linéaires[3] en $a^{n+1}$,

$\rightsquigarrow$ $f_s^{exp}-f_s^{imp}\,a^n$ et $(1-\theta)\ \mathrm{div}((\rho\underline{u})\,a^n)-(1-\theta)\ \mathrm{div}(\mu_{tot}\,\underline{\mathrm{grad}}\ a^n)$ l'ensemble des termes explicites (y compris la partie explicite provenant du schéma en temps appliqué à la convection diffusion).

De même, on introduit un opérateur $\mathcal{EM}_n$ approché de $\mathcal{E}_n$, linéaire et simplement inversible, tel que son expression contient :

[1] $a$, sous forme discrète en espace, correspond à un vecteur dimensionné à `NCELET` de composante $a_I$, I décrivant l'ensemble des cellules.
[2] cf. `introd`
[3] Lors de la discrétisation en espace, le caractère linéaire de ces termes pourra cependant être perdu, notamment à cause du test de pente.

$\rightsquigarrow$ la prise en compte des termes linéaires en $a$,
$\rightsquigarrow$ la convection intégrée par un schéma décentré amont (upwind) du premier ordre en espace,
$\rightsquigarrow$ les flux diffusifs non reconstruits.

$$\mathcal{EM}_n(a) = f_s^{imp}\, a + \theta\ [\mathrm{div}((\rho \underline{u})\, a)]^{amont} - \theta\ [\mathrm{div}(\mu_{tot}\ \underline{\mathrm{grad}}\ a)]^{N\ Rec} \tag{II.5.4}$$

Cet opérateur permet donc de contourner la difficulté induite par la présence d'éventuelles non linéarités introduites par l'activation du test de pente lors du schéma convectif, et par le remplissage important de la structure de la matrice découlant de la présence des gradients propres à la reconstruction.
On a la relation[4], pour toute cellule $\Omega_I$ de centre $I$ :

$$\mathcal{EM}_{disc}(a, I) = \int_{\Omega_i} \mathcal{EM}_n(a)\, d\Omega$$

On cherche à résoudre :

$$0 = \mathcal{E}_n(a^{n+1}) = \mathcal{EM}_n(a^{n+1}) + \mathcal{E}_n(a^{n+1}) - \mathcal{EM}_n(a^{n+1}) \tag{II.5.5}$$

Soit :

$$\mathcal{EM}_n(a^{n+1}) = \mathcal{EM}_n(a^{n+1}) - \mathcal{E}_n(a^{n+1}) \tag{II.5.6}$$

On va pour cela utiliser un algorithme de type point fixe en définissant la suite $(a^{n+1,\,k})_{k\in\mathbb{N}}$[5]:

$$\left\{ \begin{array}{l} a^{n+1,\,0} = a^n \\ a^{n+1,\,k+1} = a^{n+1,\,k} + \delta a^{n+1,\,k+1} \end{array} \right.$$

où $\delta a^{n+1,\,k+1}$ est solution de :

$$\mathcal{EM}_n(a^{n+1,\,k} + \delta a^{n+1,\,k+1}) = \mathcal{EM}_n(a^{n+1,\,k}) - \mathcal{E}_n(a^{n+1,\,k}) \tag{II.5.7}$$

Soit encore, par linéarité de $\mathcal{EM}_n$ :

$$\mathcal{EM}_n(\delta a^{n+1,\,k+1}) = -\mathcal{E}_n(a^{n+1,\,k}) \tag{II.5.8}$$

Cette suite, couplée avec le choix de l'opérateur $\mathcal{E}_n$, permet donc de lever la difficulté induite par la présence de la convection (discrétisée à l'aide de schémas numériques qui peuvent introduire des non linéarités) et les termes de reconstruction. Le schéma réellement choisi par l'utilisateur pour la convection (donc éventuellement non linéaire si le test de pente est activé) ainsi que les termes de reconstruction vont être pris à l'itération $k$ et traités au second membre *via* le sous-programme `bilsc2`, alors que les termes non reconstruits sont pris à l'itération $k+1$ et représentent donc les inconnues du système linéaire résolu par `codits`[6].

On suppose de plus que cette suite $(a^{n+1,\,k})_k$ converge vers la solution $a^{n+1}$ de l'équation (II.5.2), *i.e.* $\lim\limits_{k\to\infty} \delta a^{n+1,\,k} = 0$, ceci pour tout $n$ donné.
(II.5.8) correspond à l'équation résolue par `codits`. La matrice $\underline{\underline{EM}}_n$, matrice associée à $\mathcal{EM}_n$ est à inverser, les termes non linéaires sont mis au second membre mais sous forme explicite (indice $k$ de $a^{n+1,\,k}$) et ne posent donc plus de problème.

Remarque 1
La viscosité $\mu_{tot}$ prise dans $\mathcal{EM}_n$ et dans $\mathcal{E}_n$ dépend du modèle de turbulence utilisé. Ainsi on a $\mu_{tot} = \mu_{laminaire} + \mu_{turbulent}$ dans $\mathcal{EM}_n$ et dans $\mathcal{E}_n$ sauf lorsque l'on utilise un modèle $R_{ij} - \varepsilon$, auquel cas on a $\mu_{tot} = \mu_{laminaire}$.

[4] On pourra se reporter au sous-programme `matrix` pour plus de détails relativement à $\mathcal{EM}_{disc}$, opérateur discret agissant sur un scalaire $a$.
[5] Dans le cas ou le point fixe en vitesse-pression est utilisé (`NTERUP`$>1$) $a^{n+1,0}$ est initialisé par la dernière valeur obtenue de $a^{n+1}$.
[6] cf. le sous-programme `navsto`.

Le choix de $\mathcal{EM}_n$ étant *a priori* arbitraire ($\mathcal{EM}_n$ doit être linéaire et la suite $(a^{n+1,\,k})_{k\in\mathbb{N}}$ doit converger pour tout $n$ donné), une option des modèles $R_{ij}-\varepsilon$ (`IRIJNU` = 1) consiste à forçer $\mu^n_{tot}$ dans l'expression de $\mathcal{EM}_n$ à la valeur $\mu^n_{laminaire}+\mu^n_{turbulent}$ lors de l'appel à `codits` dans le sous-programme `navsto`, pour l'étape de prédiction de la vitesse. Ceci n'a pas de sens physique (seul $\mu^n_{laminaire}$ étant censé intervenir), mais cela peut dans certains cas avoir un effet stabilisateur, sans que cela modifie pour autant les valeurs de la limite de la suite $(a^{n+1,\,k})_k$.

Remarque 2
Quand `codits` est utilisé pour le couplage instationnaire renforcé vitesse-pression (`IPUCOU`=1), on fait une seule itération $k$ en initialisant la suite $(a^{n+1,\,k})_{k\in\mathbb{N}}$ à zéro. Les conditions de type Dirichlet sont annulées (on a `INC` = 0) et le second membre est égal à $\rho|\Omega_i|$. Ce qui permet d'obtenir une approximation de type diagonal de $\underline{\underline{EM}}_n$ nécessaire lors de l'étape de correction de la vitesse[7].

## 5.3 Mise en œuvre

L'algorithme de ce sous-programme est le suivant :
- détermination des propriétés de la matrice $\underline{\underline{EM}}_n$ (symétrique si pas de convection, non symétrique sinon)
- choix automatique de la méthode de résolution pour l'inverser si l'utilisateur ne l'a pas spécifié pour la variable traitée. La méthode de Jacobi est utilisée par défaut pour toute variable scalaire $a$ convectée. Les méthodes disponibles sont la méthode du gradient conjugué, celle de Jacobi, et le bi-gradient conjugué stabilisé (*BICGStab*) pour les matrices non symétriques. Un préconditionnement diagonal est possible et utilisé par défaut pour tous ces solveurs excepté Jacobi.
- prise en compte de la périodicité (translation ou rotation d'un scalaire, vecteur ou tenseur),
- construction de la matrice $\underline{\underline{EM}}_n$ correspondant à l'opérateur linéaire $\mathcal{EM}_n$ par appel au sous-programme `matrix`[8]. Les termes implicites correspondant à la partie diagonale de la matrice et donc aux contributions différentielles d'ordre 0 linéaires en $a^{n+1}$,(*i.e* $f_s^{imp}$), sont stockés dans le tableau `ROVSDT` (réalisé en amont du sous-programme appelant `codits`).
- création de la hiérarchie de maillage si on utilise le multigrille (`IMGRP` $>0$).
- appel à `bilsc2` pour une éventuelle prise en compte de la convection-diffusion explicite lorsque $\theta\neq 0$.
- boucle sur le nombre d'itérations de 1 à `NSWRSM` (appelé `NSWRSP` dans `codits`). Les itérations sont représentées par $k$ appelé `ISWEEP` dans le code et définissent les indices de la suite $(a^{n+1,\,k})_k$ et de $(\delta a^{n+1,\,k})_k$.

Le second membre est scindé en deux parties :

■ un terme, affine en $a^{n+1,\,k-1}$, facile à mettre à jour dans le cadre de la résolution par incrément, et qui s'écrit :

$$-f_s^{imp}\left(a^{n+1,\,k-1}-a^{n+1,0}\right)+f_s^{exp}-(1-\theta)\left[\,\mathrm{div}((\rho\underline{u})\,a^{n+1,0})-\mathrm{div}(\mu_{\,tot}\,\underline{\mathrm{grad}}\;a^{n+1,0})\,\right]$$

■ les termes issus de la convection/diffusion (avec reconstruction) calculée par `bilsc2`.

$$-\theta\left[\,\mathrm{div}\left((\rho\underline{u})\,a^{n+1,\,k-1}\right)-\mathrm{div}\left(\mu_{\,tot}\,\underline{\mathrm{grad}}\;a^{n+1,\,k-1}\right)\right]$$

La boucle en $k$ est alors la suivante :

- Calcul du second membre, hors contribution des termes de convection-diffusion explicite `SMBINI`; le second membre complet correspondant à $\mathcal{E}_n(a^{n+1,\,k-1})$ est, quant à lui, stocké dans le tableau

[7] cf. le sous-programme `resolp`.
[8] On rappelle que dans `matrix`, la convection est traitée, quelque soit le choix de l'utilisateur, avec un schéma décentré amont d'ordre 1 en espace et qu'il n'y a pas de reconstruction pour la diffusion. Le choix de l'utilisateur quant au schéma numérique pour la convection intervient uniquement lors de l'intégration des termes de convection de $\mathcal{E}_n$, au second membre de (II.5.8) dans le sous-programme `bilsc2`.

SMBRP, initialisé par SMBINI et complété par les termes reconstruits de convection-diffusion par appel au sous-programme bilsc2.
À l'itération $k$, SMBINI noté $\texttt{SMBINI}^{\,k}$ vaut :

$$\texttt{SMBINI}^{\,k} = f_s^{exp} - (1-\theta)\,[\,\mathrm{div}((\rho\underline{u})\,a^n) - \mathrm{div}(\mu_{\,tot}\,\underline{\mathrm{grad}}\;a^n)\,] - f_s^{imp}(\,a^{n+1,\,k-1} - a^n\,)$$

• Avant de boucler sur $k$,un premier appel au sous-programme bilsc2 avec THETAP $= 1-\theta$ permet de prendre en compte la partie explicite des termes de convection-diffusion provenant du schéma en temps.

$$\texttt{SMBRP}^{\,0} = f_s^{exp} - (1-\theta)\,[\,\mathrm{div}((\rho\underline{u})\,a^n) - \mathrm{div}(\mu_{\,tot}\,\underline{\mathrm{grad}}\;a^n)\,]$$

Avant de boucler sur $k$, le second membre $\texttt{SMBRP}^{\,0}$ est stocké dans le tableau $\texttt{SMBINI}^{\,0}$ et sert pour l'initialisation du reste du calcul.

$$\texttt{SMBINI}^{\,0} = \texttt{SMBRP}^{\,0}$$

• pour $k=1$,

$$\begin{aligned}\texttt{SMBINI}^{\,1} &= f_s^{exp} - (1-\theta)\,[\,\mathrm{div}((\rho\underline{u})\,a^n) - \mathrm{div}(\mu_{\,tot}\,\underline{\mathrm{grad}}\;a^n)\,] - f_s^{imp}(\,a^{n+1,\,0} - a^n\,)\\ &= f_s^{exp} - (1-\theta)\,\left[\,\mathrm{div}((\rho\underline{u})\,a^{n+1,\,0}) - \mathrm{div}(\mu_{\,tot}\,\underline{\mathrm{grad}}\;a^{n+1,\,0})\,\right] - f_s^{imp}\,\delta a^{n+1,\,0}\end{aligned}$$

On a donc :

$$\texttt{SMBINI}^{\,1} = \texttt{SMBINI}^{\,0} - \texttt{ROVSDT} * (\,\texttt{PVAR} - \texttt{PVARA})$$

et $\texttt{SMBRP}^{\,1}$ est complété par un second appel au sous-programme bilsc2 avec THETAP $= \theta$, de manière à ajouter dans le second membre la partie de la convection-diffusion implicite.

$$\begin{aligned}\texttt{SMBRP}^{\,1} &= \texttt{SMBINI}^{\,1} - \theta\,\left[\,\mathrm{div}((\rho\underline{u})\,a^{n+1,\,0}) - \mathrm{div}(\mu_{\,tot}\,\underline{\mathrm{grad}}\;a^{n+1,\,0})\,\right]\\ &= f_s^{exp} - (1-\theta)\,[\,\mathrm{div}((\rho\underline{u})\,a^n) - \mathrm{div}(\mu_{\,tot}\,\underline{\mathrm{grad}}\;a^n)\,] - f_s^{imp}\,(a^{n+1,\,0} - a^n)\\ &\quad -\theta\,\left[\,\mathrm{div}((\rho\underline{u})\,a^{n+1,\,0}) - \mathrm{div}(\mu_{\,tot}\,\underline{\mathrm{grad}}\;a^{n+1,\,0})\,\right]\end{aligned}$$

• pour $k=2$,
de façon analogue, on obtient :

$$\texttt{SMBINI}^{\,2} = f_s^{exp} - (1-\theta)\,[\,\mathrm{div}((\rho\underline{u})\,a^n) - \mathrm{div}(\mu_{\,tot}\,\underline{\mathrm{grad}}\;a^n)\,] - f_s^{imp}\,(\,a^{n+1,\,1} - a^n\,)$$

Soit :

$$\texttt{SMBINI}^{\,2} = \texttt{SMBINI}^{\,1} - \texttt{ROVSDT} * \texttt{DPVAR}^{\,1}$$

l'appel au sous-programme bilsc2, étant systématiquement fait par la suite avec THETAP $= \theta$, on obtient de même :

$$\texttt{SMBRP}^{\,2} = \texttt{SMBINI}^{\,2} - \theta\left[\mathrm{div}\left((\rho\underline{u})\,a^{n+1,\,1}\right) - \mathrm{div}\left(\mu_{\,tot}\,\underline{\mathrm{grad}}\;\;a^{n+1,\,1}\right)\right]$$

où

$$a^{n+1,\,1} = \texttt{PVAR}^{\,1} = \texttt{PVAR}^{\,0} + \texttt{DPVAR}^{\,1} = a^{n+1,\,0} + \delta a^{n+1,\,1}$$

• pour l'itération $k+1$,
Le tableau $\texttt{SMBINI}^{\,k+1}$ initialise le second membre complet $\texttt{SMBRP}^{\,k+1}$ auquel vont être rajoutées les contributions convectives et diffusives *via* le sous-programme bilsc2.
on a la formule :

$$\texttt{SMBINI}^{\,k+1} = \texttt{SMBINI}^{\,k} - \texttt{ROVSDT} * \texttt{DPVAR}^{\,k}$$

Puis suit le calcul et l'ajout des termes de convection-diffusion reconstruits de $-\,\mathcal{E}_n(a^{n+1,\,k})$, par appel au sous-programme bilsc2. On rappelle que la convection est prise en compte à cette étape par le schéma numérique choisi par l'utilisateur (schéma décentré amont du premier ordre en espace, schéma centré du second ordre en espace, schéma décentré amont du second ordre S.O.L.U. ou une pondération (blending) des schémas dits du second ordre (centré ou S.O.L.U.)

avec le schéma amont du premier ordre, utilisation éventuelle d'un test de pente).
Cette contribution (convection-diffusion) est alors ajoutée dans le second membre $\texttt{SMBRP}^{k+1}$ (initialisé par $\texttt{SMBINI}^{k+1}$).

$$\begin{array}{rl}\texttt{SMBRP}^{k+1} & = \texttt{SMBINI}^{k+1} - \theta \left[ \mathrm{div}\left((\rho \underline{u})\, a^{n+1,\,k}\right) - \mathrm{div}\left(\mu_{tot}\, \underline{\mathrm{grad}}\ a^{n+1,\,k}\right)\right] \\ & = f_s^{exp} - (1-\theta)\left[\mathrm{div}((\rho \underline{u})\, a^n) - \mathrm{div}(\mu_{tot}\, \underline{\mathrm{grad}}\ a^n)\right] - f_s^{imp}\,(a^{n+1,\,k} - a^n) \\ & -\theta \left[\mathrm{div}((\rho \underline{u})\, a^{n+1,k}) - \mathrm{div}(\mu_{tot}\, \underline{\mathrm{grad}}\ a^{n+1,k})\right]\end{array}$$

- Résolution du système linéaire en $\delta a^{n+1,\,k+1}$ correspondant à l'équation (II.5.8) par inversion de la matrice $\underline{\underline{EM}}_n$, en appelant le sous programme `invers`. On calcule $a^{n+1,\,k+1}$ grâce à la formule :

$$a^{n+1,\,k+1} = a^{n+1,\,k} + \delta a^{n+1,\,k+1}$$

  Soit :

$$\texttt{PVAR}^{k+1} = \texttt{PVAR}^{k} + \texttt{DPVAR}^{k+1}$$

- Traitement de la périodicité et du parallélisme.

- Test de convergence :
  Il porte sur la quantité $||\texttt{SMBRP}^{k+1}|| < \varepsilon||\underline{\underline{EM}}_n(a^n) + \texttt{SMBRP}^{1}||$, où $||\,.\,||$ représente la norme euclidienne. Si le test est vérifié, la convergence est atteinte et on sort de la boucle sur les itérations. La solution recherchée est $a^{n+1} = a^{n+1,\,k+1}$.
  Sinon, on continue d'itérer dans la limite des itérations imposées par `NSWRSM` dans `usini1`.
  En "continu" ce test de convergence s'écrit aussi :

$$\begin{array}{rl}||\texttt{SMBRP}^{k+1}|| & < \varepsilon||f_s^{exp} - \mathrm{div}((\rho \underline{u})\, a^n) + \mathrm{div}(\mu_{tot}\, \underline{\mathrm{grad}}\ a^n) \\ & +[\mathrm{div}((\rho \underline{u})\, a^n)]^{amont} + [\mathrm{div}(\mu_{tot}\, \underline{\mathrm{grad}}\ a^n)]^{N\ Rec}||\end{array}$$

  Si bien que sur maillage orthogonal avec schéma de convection upwind et en l'absence de terme source, la suite converge en théorie en une unique itération puisque par construction :

$$||\texttt{SMBRP}^{2}|| = 0 \quad < \varepsilon||f_s^{exp}||$$

Fin de la boucle.

## 5.4 Points à traiter

**• Approximation $\mathcal{EM}_n$ de l'opérateur $\mathcal{E}_n$**
D'autres approches visant soit à modifier la définition de l'approximée, prise en compte du schéma centré sans reconstruction par exemple, soit à abandonner cette voie seraient à étudier.

**• Test de convergence**
La quantité définissant le test de convergence est également à revoir, éventuellement à simplifier.

**• Prise en compte de $T_s^{imp}$**
Lors de la résolution de l'équation par `codits`, le tableau `ROVSDT` a deux fonctions : il sert à calculer la diagonale de la matrice (par appel de `matrix`) et il sert à mettre à jour le second membre à chaque sous-itération de la résolution en incréments. Or, dans le cas où $T_s^{imp}$ est positif, on ne l'intègre pas dans `ROVSDT`, afin de ne pas affaiblir la diagonale de la matrice. De ce fait, on ne l'utilise pas pour mettre à jour le second membre, alors que ce serait tout à fait possible. Au final, on obtient donc un terme source traité totalement en explicite ($T_s^{exp} + T_s^{imp} a^n$), alors que la résolution en incréments nous permettrait justement de l'impliciter quasiment totalement ($T_s^{exp} + T_s^{imp} a^{n+1,k_{fin}-1}$, où $k_{fin}$ est la dernière sous-itération effectuée).
Pour ce faire, il faudrait définir deux tableaux `ROVSDT` dans `codits`.

# 6- Sous-programme `condli`

## 6.1 Function

Boundary conditions are required in at least three principal cases :

- calculation of the convection terms (first order derivative in space) at the boundary : the calculation uses a flux at the boundary and requires the value of the convected variable at the boundary when the latter is entering the domain in the sens of the characteristic curves of the system (in the sens of the velocity, in the case of the single equation of a simple scalar : sufficient interpretation in the current framework of *Code_Saturne* [1]) ;
- calculation of the diffusion terms (second order derivative in space) : a method to determine the value of the first order spatial derivatives at the boundary is then required (more exactly, the terms that depend upon it are required, such as the stresses of the thermal fluxes at the wall) :
- calculation of the cell gradients : the variable at the boundary faces are required (more generally, the discrete terms of the equations which depend upon the gradient inside boundary cells are required, such as the transpose gradient terms in the Navier-Stokes equations).

These considerations only concern the computation variables (velocity, pressure, Reynolds tensor, scalars solution of a convection-diffussion equation). For these variables

Les considérations présentes concernent uniquement les variables de calcul (vitesse, pression, tenseur de Reynolds, scalaires solution d'une équation de convection-diffusion). Pour ces grandeurs [2], the user has to define the boundary conditions at every boundary face (`usclim`).

The subroutine `condli` transforms the data provided by the user (in `usclim`) into an internal format of representation of the boundary conditions. Verifications of the completeness and coherence are also performed (in `vericl`). More particularly, the smooth-wall boundary conditions (`clptur`), the rough-wall boundary conditions (`clptrg`) and the symmetry boundary conditions for the velocities and the Reynolds stress tensor (`clsyvt`) are treated in dedicated subroutines.

The subroutine `condli` provides as an output pairs of coefficients $A_b$ and $B_b$ for each variable $f$ and for each boudary face. These are used for the calculation of the discrete terms in the equations to be solved. More specifically, these coefficients are used to calculate a value at the boundary faces $f_{b,int}$ (localised at the centre of the boundary face, barycentre of its vertices) by the relation $f_{b,int} = A_b + B_b\, f_{I'}$, where $f_{I'}$ is the value of the variable at point $I'$. $I'$ is the projection onto the centre of the cell adjoin to the boundary on the line normal to the boundary face and passing by its centre (see figure II.6.1).

## 6.2 Discretisation

• **Notation**
On désignera dans la suite par *VarScalaire* toute variable

- autre que la vitesse, la pression, les grandeurs turbulentes $k$, $\varepsilon$, $R_{ij}$, $\varphi$, $\bar{f}$ et $\omega$,

[1] except with the compressible module, see. `cfxtcl`

[2] The other variables (the physical properties for instance) have a different treatment which will not be detailed here (for instance, for the density, the user defines directly the values at the boundary. This information is then stored ; one is referred to `usphyv` or `phyvar` for more information).

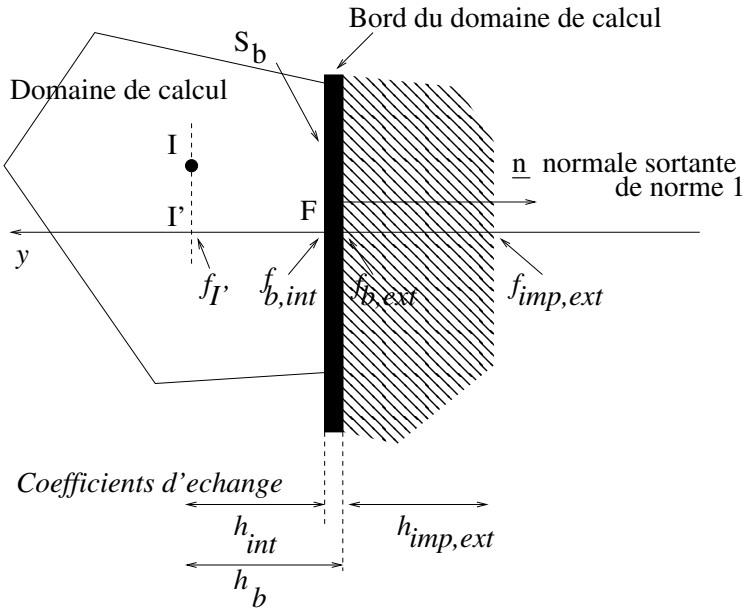



Figure II.6.1: Boundary cell.

- solution d'une équation de convection-diffusion.

La dénomination *VarScalaire* pourra en particulier désigner la température, un scalaire passif, une fraction massique ou (sauf mention contraire explicite) la variance des fluctuations d'une autre *VarScalaire*. Les variables d'état déduites (masse volumique, viscosité...) ne seront pas désignées par *VarScalaire*.

**• Représentation des conditions aux limites standard dans `usclim`**

Des conditions aux limites standardisées peuvent être fournies par l'utilisateur dans `usclim`. Il est pour cela nécessaire d'affecter un type aux faces de bord des cellules concernées[3]. Les conditions prévues par défaut sont les suivantes :

- **Entrée** : correspond à une condition de Dirichlet sur toutes les variables transportées (vitesse, variables turbulentes, *VarScalaires*...), et à une condition de Neumann homogène (flux nul) sur la pression.
- **Sortie** :
  - lorsque le flux de masse est effectivement dirigé vers l'extérieur du domaine, ce choix correspond à une condition de Neumann homogène sur toutes les variables transportées et à $\dfrac{\partial^2 P}{\partial \underline{n} \partial \underline{\tau}_i} = 0$, pris en compte sous forme de Dirichlet pour la pression ($\underline{n}$ et $(\underline{\tau}_i)_{i\in\{1,2\}}$ désignent respectivement le vecteur normal de la face de sortie considérée et deux vecteurs normés, orthogonaux entre eux et dans le plan de la face de sortie). Cette condition est appliquée de manière explicite en utilisant le champ de pression et son gradient au pas de temps précédent. En outre, la pression étant définie à une constante près, elle est recalée en un point de sortie pour y conserver la valeur `P0` (on évite ainsi toute dérive vers des valeurs très grandes relativement à l'écart maximal de pression sur le domaine)[4].
  - lorsque le flux de masse est dirigé vers l'intérieur du domaine, situation peu souhaitable *a priori*[5], on impose une condition de Dirichlet homogène sur la vitesse (pas sur le flux de

[3] L'affectation d'un type se fait en renseignant le tableau `ITYPFB`.

[4] Lorsqu'il n'y a pas de sortie, le spectre des valeurs propres de la matrice est décalé d'une valeur constante afin de rendre le système inversible : voir `matrix`.

[5] Un message indique à l'utilisateur combien de faces de sortie voient un flux de masse entrer dans le domaine.

masse), à défaut de connaître sa valeur en aval du domaine. La pression est traitée comme dans le cas précédent où le flux de masse est dirigé vers l'extérieur du domaine. Pour les variables autres que la vitesse et la pression, deux cas se présentent :

- on peut imposer une condition de Dirichlet pour représenter la valeur du scalaire introduit dans le domaine par les faces de bord concernées.
- on peut imposer, comme lorsque le flux de masse est sortant, une condition de Neumann homogène (ceci n'est pas une situation souhaitable, puisque l'information portée sur les faces de bord provient alors de *l'aval* de l'écoulement local). C'est le cas par défaut si l'on ne donne pas de valeur pour le Dirichlet.

- **Paroi** : on se reportera à `clptur` (ou à `clptrg` pour les parois rugueuses) pour une description du traitement des conditions aux limites de paroi (supposées imperméables au fluide). Brièvement, on peut dire ici qu'une approche par lois de paroi est utilisée afin d'imposer la contrainte tangentielle sur la vitesse. La paroi peut être défilante[6]. Les *VarScalaires* reçoivent par défaut une condition de Neumann homogène (flux nul). Si l'on souhaite imposer une valeur en paroi pour ces variables (par exemple, dans le cas d'une paroi à température imposée) une loi de similitude est utilisée pour déterminer le flux au bord en tenant compte de la couche limite. Dans le cas des couplages avec SYRTHES, *Code_Saturne* reçoit une température de paroi et fournit un flux thermique. La condition de pression standard est une condition de Neumann homogène[7]

- **Symétrie** : correspond à des conditions de Neumann homogènes pour les grandeurs scalaires et à des conditions de symétrie classiques pour les vecteurs (vitesse) et les tenseurs (tensions de Reynolds) : voir `clsyvt`.

[6] On doit alors fournir les composantes de la vitesse de la paroi.

[7] On pourra se reporter à `gradrc` pour la condition conduisant à l'extrapolation de la pression au bord, condition pilotable par l'utilisation de la variable `EXTRAP`.

• **Représentation des conditions aux limites spécifiques dans `usclim`**

On a vu que l'affectation à une face de bord d'un type standard (entrée, sortie, paroi, symétrie) permettait d'appliquer simplement à l'ensemble des variables un assortiment de conditions aux limites cohérentes entre elles pour les types usuels de frontière physique.

Une solution consiste à définir dans `usclim`, pour chaque face de bord et chaque variable, des conditions aux limites spécifiques[8] (celles-ci, comme les conditions standards, se ramènent finalement à des conditions de type mixte).

Les deux approches ne sont pas nécessairement incompatibles et peuvent même se révéler complémentaires. En effet, les conditions aux limites standards peuvent être surchargées par l'utilisateur pour une ou plusieurs variables données. Il convient cependant de s'assurer que, d'une façon ou d'une autre, une condition à la limite a été définie pour chaque face de bord et chaque variable.

Des conditions de compatibilité existent également entre les différentes variables (voir `vericl`):

- en entrée, paroi, symétrie ou sortie libre, il est important que toutes les composantes de la vitesse aient le même type de condition ;
- lorsque la vitesse reçoit une condition de sortie, il est important que la pression reçoive une condition de type Dirichlet. Pour plus de détails, on se reportera au paragraphe relatif à la condition de sortie pour la pression, page 79 ;
- lorsqu'une des variables de vitesse ou de turbulence reçoit une condition de paroi, il doit en être de même pour toutes ;
- lorsqu'une des composantes $R_{ij}$ reçoit une condition de symétrie, il doit en être de même pour toutes ;
- lorsqu'une *VarScalaire* reçoit une condition de paroi, la vitesse doit avoir le même type de condition.

[8] Les conditions aux limites spécifiques sont codées en renseignant directement les tableaux `ICODCL` et `RCODCL` pour chaque face de bord et chaque variable : des exemples sont fournis dans `usclim`.

• **Représentation interne des conditions aux limites**

**Objectif**

Les conditions fournies par l'utilisateur sont retraduites sous forme de couples de coefficients $A_b$ et $B_b$ pour chaque variable $f$ et chaque face de bord. Ces coefficients sont utilisés pour le calcul des termes discrets intervenant dans les équations à résoudre et permettent en particulier de déterminer une valeur de face de bord $f_{b,int}$. Il est important d'insister dès à présent sur le fait que cette valeur est, de manière générale, une simple valeur numérique qui ne reflète pas nécessairement une réalité physique (en particulier aux parois, pour les grandeurs affectées par la couche limite turbulente). On détaille ci-dessous le calcul de $A_b$, $B_b$ et de $f_{b,int}$.

**Notations**

- On considère l'équation (II.6.1) portant sur le scalaire $f$, dans laquelle $\rho$ représente la masse volumique, $\underline{u}$ la vitesse, $\alpha$ la conductivité et $S$ les termes sources additionnels. $C$ est défini plus bas.

$$\rho\frac{\partial f}{\partial t} + div(\rho\underline{u}f) = div\left(\frac{\alpha}{C}\,\underline{\text{grad}}\ f\right) + S \tag{II.6.1}$$

- Le coefficient $\alpha$ représente la somme des conductivités moléculaire et turbulente (selon les modèles utilisés), soit $\alpha = \alpha_m + \alpha_t$, avec, pour une modélisation de type viscosité turbulente, $\alpha_t = C\,\dfrac{\mu_t}{\sigma_t}$, où $\sigma_t$ est le nombre de Prandtl turbulent[9].

- À titre d'exemple, on précise dans le tableau 6.1 la valeur et l'unité de $\alpha$ pour quelques cas particuliers de $f$ (certains termes de l'équation peuvent alors disparaître : pour la pression, en particulier, l'équation est stationnaire et les termes convectifs sont absents).

- Le coefficient $C_p$ représente la chaleur spécifique, d'unité $m^2\,s^{-2}\,K^{-1} = J\,kg^{-1}\,K^{-1}$.

- On note $\lambda$ la conductivité thermique, d'unité $kg\,m\,s^{-3}\,K^{-1} = W\,m^{-1}\,K^{-1}$.

- Il convient de préciser que $C = 1$ pour toutes les variables hormis pour la température, cas dans lequel on a[10] $C = C_p$. Dans le code, c'est la valeur de $\dfrac{\alpha_m}{C}$ que l'utilisateur doit fournir (si la propriété est constante, les valeurs sont affectées dans `usini1` à `VISCL0` pour la vitesse et à `VISLS0` pour les *VarScalaires* ; si la propriété est variable, ce sont des tableaux équivalents qui doivent être renseignés dans `usphyv`).

- Pour la variance des fluctuations d'une *VarScalaire*, la conductivité $\alpha$ et le coefficient $C$ sont hérités de la *VarScalaire* associée.

**Condition de type Dirichlet simple** : lorsque la condition est une condition de Dirichlet simple, on obtient naturellement (cas particulier de (II.6.6)) :

$$\underbrace{f_{b,int}}_{\text{valeur de bord utilisée par le calcul}} = \underbrace{f_{réel}}_{\text{valeur réelle imposée au bord}} \tag{II.6.2}$$

[9] Le nombre de Prandtl turbulent est sans dimension et, dans certains cas usuels, pris égal à $0,7$.

[10] Plus exactement, on a $C = C_p$ pour toutes les *VarScalaires* $f$ que l'on souhaite traiter comme la température pour les conditions aux limites. Ces *VarScalaires* sont repérables par l'utilisateur au moyen de l'indicateur `ISCSTH=1`. Par défaut cet indicateur est positionné à la valeur 0 pour toutes les *VarScalaires* (qui sont alors traitées comme des scalaires passifs avec $C = 1$) hormis pour la variable thermique éventuelle (`ISCALT`$^{\text{ième}}$ *VarScalaire*), pour laquelle on a `ISCSTH=1` : on suppose par défaut que la variable thermique est la température et non l'enthalpie. Si l'on souhaite résoudre en enthalpie, il faut positionner `ISCSTH` à la valeur 2 pour la variable thermique. Pour le compressible, la variable thermique est l'énergie, identifiée par `ISCSTH=3`. On se reportera à `cfxtcl` pour le traitement des conditions aux limites.

| $f$ | | | $\alpha$ | | |
|---|---|---|---|---|---|
| symbole | nom | unité | symbole | nom | unité |
| $u_i$ | vitesse | $m\,s^{-1}$ | $\mu$ ou $\mu+\mu_t$ | viscosité dynamique | $kg\,m^{-1}\,s^{-1}$ |
| $p$ | pression | $kg\,m^{-1}\,s^{-2}$ | $\Delta t$ | pas de temps | $s$ |
| $T$ | température | $K$ | $\lambda$ | conductivité thermique | $kg\,m\,s^{-3}\,K^{-1}$ $= W\,m^{-1}\,K^{-1}$ |
| $H$ | enthalpie | $m^2\,s^{-2}$ $= J\,kg^{-1}$ | $\lambda/C_p$ | conductivité thermique/$C_p$ | $kg\,m^{-1}\,s^{-1}$ |
| $f$ | *VarScalaire* | unité($f$) | $\alpha$ | conductivité ou diffusivité | $kg\,m^{-1}\,s^{-1}$ |

Table 6.1: Valeurs et unités de $\alpha$ dans quelques cas particuliers usuels.

**Autres cas** : lorsque la condition à la limite porte sur la donnée d'un flux, il s'agit d'un flux diffusif[11]. On a alors :

$$\underbrace{\phi_{int}}_{\text{flux diffusif transmis au domaine interne}} = \underbrace{\phi_{réel}}_{\text{flux diffusif réel imposé au bord}} \tag{II.6.3}$$

Le flux diffusif réel imposé peut être donné

- directement (condition de Neumann), soit $\phi_{réel} = \phi_{imp,ext}$ ou
- déduit implicitement de deux informations imposées : une valeur externe $f_{imp,ext}$ et un coefficient d'échange $h_{imp,ext}$ (condition de Dirichlet généralisée).

Selon le type de condition (Dirichlet ou Neumann) et en prenant pour hypothèse la conservation du flux dans la direction normale au bord, on peut alors écrire (voir figure II.6.1) :

$$\underbrace{h_{int}(f_{b,int}-f_{I'})}_{\phi_{int}} = \underbrace{h_b(f_{b,ext}-f_{I'})}_{\phi_b} = \begin{cases} \underbrace{h_{imp,ext}(f_{imp,ext}-f_{b,ext})}_{\phi_{réel\ imposé}} & \text{(condition de Dirichlet)} \\ \underbrace{\phi_{imp,ext}}_{\phi_{réel\ imposé}} & \text{(condition de Neumann)} \end{cases} \tag{II.6.4}$$

Le rapport entre le coefficient $h_b$ et le coefficient $h_{int}$ rend compte de l'importance de la traversée de la zone proche du bord et revêt une importance particulière dans le cas des parois le long desquelles se développe une couche limite (dont les propriétés sont alors prises en compte par $h_b$ : se reporter à `clptur`). Dans le cadre plus simple considéré ici, on se limitera au cas $h_b = h_{int}$ et $f_{b,ext} = f_{b,int} = f_b$. La relation (II.6.4) s'écrit alors :

$$\underbrace{h_{int}(f_b-f_{I'})}_{\phi_{int}} = \begin{cases} \underbrace{h_{imp,ext}(f_{imp,ext}-f_b)}_{\phi_{réel\ imposé}} & \text{(condition de Dirichlet)} \\ \underbrace{\phi_{imp,ext}}_{\phi_{réel\ imposé}} & \text{(condition de Neumann)} \end{cases} \tag{II.6.5}$$

En réarrangeant, on obtient la valeur de bord :

$$f_b = \begin{cases} \dfrac{h_{imp,ext}}{h_{int}+h_{imp,ext}}\, f_{imp,ext} + \dfrac{h_{int}}{h_{int}+h_{imp,ext}}\, f_{I'} & \text{(condition de Dirichlet)} \\ \dfrac{1}{h_{int}}\, \phi_{imp,ext} + f_{I'} & \text{(condition de Neumann)} \end{cases} \tag{II.6.6}$$

[11] En effet, le flux total sortant du domaine est donné par la somme du flux convectif (si la variable est effectivement convectée) et du flux diffusif. Néanmoins, pour les parois étanches et les symétries, le flux de masse est nul et la condition se réduit à une contrainte sur le flux diffusif. De plus, pour les sorties (flux de masse sortant), la condition à la limite ne porte que sur le flux diffusif (souvent une condition de Neumann homogène), le flux convectif dépendant des conditions amont (il n'a donc pas besoin de condition à la limite). Enfin, aux entrées, c'est le plus souvent une condition de Dirichlet simple qui est appliquée et le flux diffusif s'en déduit.

**Conclusion** : on notera donc les conditions aux limites de manière générale sous la forme :

$$f_b = A_b + B_b\, f_{I'} \tag{II.6.7}$$

avec $A_b$ et $B_b$ définis selon le type des conditions :

$$\text{Dirichlet} \begin{cases} A_b = & \dfrac{h_{imp,ext}}{h_{int}+h_{imp,ext}} f_{imp,ext} \\ B_b = & \dfrac{h_{int}}{h_{int}+h_{imp,ext}} \end{cases} \qquad \text{Neumann} \begin{cases} A_b = & \dfrac{1}{h_{int}} \phi_{imp,ext} \\ B_b = & 1 \end{cases} \tag{II.6.8}$$

**Remarques**

- La valeur $f_{I'}$ est calculée en utilisant le gradient cellule de $f$, soit : $f_{I'} = f_I + \underline{II'}\underline{\text{grad}}\ f_I$.
- Il reste à préciser la valeur de $h_{int}$. Il s'agit d'une valeur *numérique*, n'ayant *a priori* aucun rapport avec un coefficient d'échange physique, et dépendante du mode de calcul du flux diffusif dans la première maille de bord. Ainsi $h_{int} = \dfrac{\alpha}{\overline{I'F}}$ (l'unité s'en déduit naturellement).
- On rappelle que dans le code, c'est la valeur de $\dfrac{\alpha_m}{C}$ que l'utilisateur doit fournir. Si la propriété est constante, les valeurs sont affectées dans `usini1` à `VISCL0` pour la vitesse (viscosité dynamique moléculaire $\mu$ en $kg\,m^{-1}\,s^{-1}$) et à `VISLS0` pour les *VarScalaires* (par exemple, pour la température et l'enthalpie, $\dfrac{\lambda}{C_p}$ en $kg\,m^{-1}\,s^{-1}$). Si la propriété est variable en espace ou en temps, ce sont des tableaux équivalents qui doivent être renseignés dans `usphyv`. En outre, la variance des fluctuations d'une *VarScalaire* hérite automatiquement la valeur de $\dfrac{\alpha_m}{C}$ de la *VarScalaire* associée (*Code_Saturne* 1.1 et suivantes).
- On rappelle également, car ce peut être source d'erreur, que dans le code, on a :
  - pour la température $\alpha_m = \lambda$ et $C = C_p$
  - pour l'enthalpie $\alpha_m = \dfrac{\lambda}{C_p}$ et $C = 1$

**Exemples de cas particuliers**

- Dans le cas d'une condition de Dirichlet, l'utilisateur est donc conduit à fournir deux données : $f_{imp,ext}$ et $h_{imp,ext}$. Pour obtenir une condition de Dirichlet simple (sans coefficient d'échange) il suffit d'imposer $h_{imp,ext} = +\infty$. C'est le cas d'utilisation le plus courant (en pratique, $h_{imp,ext} = 10^{30}$ ).
- Dans le cas d'une condition de Neumann, l'utilisateur fournit une seule valeur $\phi_{imp,ext}$ (nulle pour les conditions de Neumann homogènes).

**Valeur et unité des données à fournir**

Le tableau 6.2 permet d'identifier les unités de $h$ ($h_{imp,ext}$ ou $h_{int}$) et $\phi_{imp,ext}$ pour quelques variables classiques. La variable $d$ représente une longueur (égale, pour $h_{int}$, à $\overline{I'F}$). Il est également important de noter qu'un flux "sortant" doit être positif.

| $f$ | | | $h$ | |
|---|---|---|---|---|
| symbole | nom | unité | homogène à | unité |
| $u_i$ | vitesse | $m\,s^{-1}$ | $(\mu+\mu_t)/d$ | $kg\,m^{-2}\,s^{-1}$ |
| $p$ | pression | $kg\,m^{-1}\,s^{-2}$ | $(\Delta t)/d$ | $s\,m^{-1}$ |
| $T$ | température | $K$ | $(\lambda+C_p\mu_t/\sigma_t)/d$ | $kg\,s^{-3}\,K^{-1}$ $= W\,m^{-2}\,K^{-1}$ |
| $H$ | enthalpie | $m^2\,s^{-2}=J\,kg^{-1}$ | $(\lambda/C_p+\mu_t/\sigma_t)/d$ | $kg\,m^{-2}\,s^{-1}$ |
| $f$ | scalaire | unité($f$) | $\alpha/d$ | $kg\,m^{-2}\,s^{-1}$ |

| $f$ | | | $\phi_{imp,ext}$ | |
|---|---|---|---|---|
| symbole | nom | unité | homogène à | unité |
| $u_i$ | vitesse | $m\,s^{-1}$ | $(\mu+\mu_t)\,\underline{\text{grad}}\ (\underline{u_i}).\underline{n}$ | $kg\,m^{-1}\,s^{-2}$ |
| $p$ | pression | $kg\,m^{-1}\,s^{-2}$ | $(\Delta t)\,\underline{\text{grad}}\ (p).\underline{n}$ | $kg\,m^{-2}\,s^{-1}$ |
| $T$ | température | $K$ | $(\lambda+C_p\mu_t/\sigma_t)\,\underline{\text{grad}}\ (T).\underline{n}$ | $kg\,s^{-3}=W\,m^{-2}$ |
| $H$ | enthalpie | $m^2\,s^{-2}=J\,kg^{-1}$ | $(\lambda/C_p+\mu_t/\sigma_t)\,\underline{\text{grad}}\ (H).\underline{n}$ | $kg\,s^{-3}=W\,m^{-2}$ |
| $f$ | scalaire | unité($f$) | $\alpha\,\underline{\text{grad}}\ (f).\underline{n}$ | $kg\,m^{-2}\,s^{-1}$ unité($f$) |

Table 6.2: Valeur et unité de $h$ et $\phi_{imp,ext}$ dans quelques cas particuliers usuels.

**• Condition de sortie pour la pression**

On précise ici la condition de sortie appliquée à la pression dans le cas des sorties standards. Il est nécessaire d'imposer une condition de type Dirichlet (accompagnée d'une condition de type Neumann homogène sur les composantes de la vitesse). On la calcule à partir des valeurs de la variable au pas de temps précédent.

- En raisonnant sur une configuration simple (de type canal, avec une sortie plane, perpendiculaire à l'écoulement), on peut faire l'hypothèse que la forme des profils de pression pris sur les surfaces parallèles à la sortie est inchangée aux alentours de celle-ci (hypothèse d'un écoulement établi, loin de toute perturbation). Dans cette situation, on peut écrire $\dfrac{\partial^2 P}{\partial \underline{n}\partial \underline{\tau}_i}=0$ ($\underline{n}$ est le vecteur normal à la sortie, $\underline{\tau}_i$ représente une base du plan de sortie).

- Si, de plus, on peut supposer que le gradient de pression pris dans la direction perpendiculaire aux faces de sortie est uniforme au voisinage de celle-ci, le profil à imposer en sortie (valeurs $p_b$) se déduit du profil pris sur un plan amont (valeurs $p_{amont}$) en ajoutant simplement la constante $R=d\,\underline{\text{grad}}\ (p).\underline{n}$ (où $d$ est la distance entre le plan amont et la sortie), soit $p_b=p_{amont}+R$ (le fait que $R$ soit identique pour toutes les faces de sortie est important afin de pouvoir l'éliminer dans l'équation (II.6.9) ci-dessous).

- Avec l'hypothèse supplémentaire que les points $I'$ relatifs aux faces de sortie sont sur un plan parallèle à la sortie, on peut utiliser les valeurs en ces points ($p_{I'}$) pour valeurs amont soit $p_{amont}=p_{I'}=p_I+\underline{II'}.\underline{\text{grad}}\ p$.

- Par ailleurs, la pression étant définie à une constante près (écoulement incompressible) on peut fixer sa valeur arbitrairement en un point $A$ (centre d'une face de sortie choisie arbitrairement[12]) à $p_0$ (valeur fixée par l'utilisateur, égale à `P0` et nulle par défaut), et donc décaler le profil imposé en sortie en ajoutant :
$R_0=p_0-(p_{amont,A}+R)=p_0-(p_{I',A}+R)$.

[12] première face de sortie rencontrée en parcourant les faces de bord dans l'ordre naturel induit par la numérotation interne au code

- On obtient donc finalement :

$$\begin{aligned} p_b &= p_{I'} + R + R_0 \\ &= p_{I'} + R + p_0 - (p_{I',A} + R) \\ &= p_{I'} + \underbrace{p_0 - p_{I',A}}_{\text{valeur constante } R_1} \\ &= p_{I'} + R_1 \end{aligned} \tag{II.6.9}$$

On constate donc que la condition de pression en sortie est une condition de Dirichlet dont les valeurs sont égales aux valeurs de la pression (prises au pas de temps précédent) sur le plan amont des points $I'$ et recalées pour obtenir `PO` en un point de sortie arbitraire.

## 6.3 Mise en œuvre

**• Mode de prescription des conditions aux limites dans `usclim`**

L'utilisateur fournit pour chaque face un moyen de déterminer les conditions aux limites de toutes les variables de calcul. Comme on l'a vu, plusieurs méthodes sont envisageables.

La plus simple consiste à renseigner, pour chaque face, un code désignant une condition à la limite standard (dans le tableau `ITYPFB` de dimension `NFABOR`) et les informations complémentaires éventuelles. Les valeurs de `ITYPFB` suivantes sont prédéfinies[13] : `IENTRE` (entrée), `ISOLIB` (sortie libre), `ISYMET` (symétrie), `IPAROI` (paroi lisse), `IPARUG` (paroi rugueuse). Dans le cas des entrées (et des sorties de type 10), il est nécessaire de fournir une valeur de Dirichlet.

L'utilisateur peut également renseigner un entier (tableau `ICODCL` de dimension `NFABOR`×`NVAR`) désignant le type de condition à appliquer à une variable donnée : les valeurs 1 (pour Dirichlet) et 3 (pour Neumann) sont souvent utilisées, plus rarement 5 (pour les conditions de paroi lisse) ou 6 (pour les conditions de paroi rugueuse) et exceptionnellement 4 (pour la symétrie d'une vitesse ou du tenseur de Reynolds), 9 (pour la sortie libre, uniquement pour les vitesses). Lorsque `ICODCL` est renseigné pour une variable donnée, il prend alors le pas sur `ITYPFB` sur la face considérée. L'utilisateur doit dans ce cas également fournir, suivant les cas, zéro, un ou deux réels (dans le tableau `RCODCL` de dimension `NFABOR`×`NVAR`×3). Pour une face et une variable $f$ données, les 3 valeurs de `RCODCL` désignent respectivement les grandeurs $f_{imp,ext}$, $h_{imp,ext}$ et $\phi_{imp,ext}$.

On indique dans le tableau 6.3 le mode de traitement des variables pour les différents types de conditions aux limites standards. La liste des valeurs complémentaires à fournir est également précisée. Le tableau 6.4 propose le même type d'information pour les conditions aux limites spécifiques plus complexes, définies à partir de la valeur de `ICODCL`. Enfin, le tableau 6.5 synthétise les valeurs admissibles de `ICODCL` pour chaque variable de calcul.

[13] L'utilisateur peut définir d'autres types (*i.e.* affecter à `ITYPFB` d'autres valeurs entières), mais elles ne recouvrent pas de conditions aux limites par défaut. Les faces de bord ainsi repérées sont cependant traitées comme un ensemble particulier lors de l'impression d'informations de type flux de masse par exemple.

| ITYPFB | type | variables traitées | type de condition | données complémentaires |
|---|---|---|---|---|
| IENTRE | entrée | variables transportées | Dirichlet | RCODCL(.,.,1) valeur d'entrée nulle par défaut |
| | | pression | Neumann homogène | |
| ISYMET | symétrie | vitesse | symétrie vecteur | |
| | | tensions de Reynolds | symétrie tenseur | |
| | | autres variables | Neumann homogène | |
| IPAROI | paroi lisse | pression | Neumann homogène[a] | |
| | | vitesses | lois de paroi (Dirichlet)[b] | RCODCL(.,.,1) vitesse de défilement nulle par défaut |
| | | grandeurs turbulentes | lois de paroi[c] | |
| | | autres variables transportées | Neumann homogène[d] | |
| IPARUG | paroi rugueuse | pression | Neumann homogène[a] | |
| | | vitesses | lois de paroi (Dirichlet)[b] | RCODCL(.,.,1) vitesse de défilement nulle par défaut<br>RCODCL(.,.,3)[f] hauteur de rugosité nulle par défaut |
| | | grandeurs turbulentes | lois de paroi[c] | |
| | | autres variables transportées | Neumann homogène[d] | |
| ISOLIB | sortie libre | pression | $\frac{\partial^2 P}{\partial n \partial \tau_i} = 0$ | |
| | | vitesses | Neumann homogène (flux de masse sortant) | |
| | | | Dirichlet homogène (flux de masse entrant) | |
| | | autres variables transportées | Neumann homogène ou Dirichlet[e] (flux de masse entrant) | RCODCL(.,.,1) valeur d'entrée nulle par défaut |

[a] Le gradient peut être extrapolé en paroi selon les valeurs de EXTRAP : voir gradrc

[b] voir clptur pour une paroi lisse ou clptrg pour une paroi rugueuse

[c] voir également clptur ou clptrg

[d] La condition de Neumann homogène est obtenue lorsque l'utilisateur ne précise rien d'autre que le type de frontière IPAROI ou IPARUG ; d'autres conditions sont possibles : le tableau 6.4 les précise.

[e] La valeur est imposée par l'utilisateur. Il n'y a pas d'implicitation temporelle.

[f] Seule la valeur pour IU(IPHAS) est utilisée.

Table 6.3: Conditions aux limites standards.

| ICODCL | variables | type de condition | données complémentaires nécessaires et suffisantes |
|---|---|---|---|
| 1 | toutes | Dirichlet | RCODCL(.,.,1) $f_{imp,ext}$ valeur nulle par défaut<br>RCODCL(.,.,2) $h_{imp,ext}$ coef. d'échange =RINFIN= $10^{30}$ par défaut |
| 3 | toutes | Neumann | RCODCL(.,.,3) $\phi_{imp,ext}$ valeur nulle par défaut |
| 4 | vitesse | Symétrie vecteur | |
| | $R_{ij}$ | Symétrie tenseur | |
| 5 | vitesse | Loi de paroi | RCODCL(.,.,1) vitesse de défilement nulle par défaut |
| | $k$, $R_{ij}$, $\varepsilon$, $\varphi$, $f$, $\omega$ | Loi de paroi | |
| | *VarScalaire* (sauf variance) | Loi de paroi | RCODCL(.,.,1) $f_{imp,ext}$ valeur en paroi nulle par défaut<br>RCODCL(.,.,2) $h_{imp,ext}$ coef. d'échange =RINFIN= $10^{30}$ par défaut |
| 6 | vitesse | Loi de paroi rugueuse | RCODCL(.,.,1) vitesse de défilement nulle par défaut<br>RCODCL(.,.,3) rugosité dynamique nulle par défaut |
| | $k$, $R_{ij}$, $\varepsilon$, $\varphi$, $f$, $\omega$ | Loi de paroi rugueuse | RCODCL(.,.,3) rugosité dynamique nulle par défaut |
| | *VarScalaire* (sauf variance) | Loi de paroi rugueuse | RCODCL(.,.,1) $f_{imp,ext}$ valeur en paroi nulle par défaut<br>RCODCL(.,.,2) $h_{imp,ext}$ coef. d'échange =RINFIN= $10^{30}$ par défaut<br>RCODCL(.,.,3) rugosité thermique nulle par défaut |
| 9 | vitesse | Neumann homogène (flux de masse sortant) | |
| | | Dirichlet homogène (flux de masse entrant) | |

Table 6.4: Conditions aux limites spécifiques.

| Variable | | Valeurs de ICODCL admissibles | | | | | |
|---|---|---|---|---|---|---|---|
| Vitesse | $U$ | 1 | 3 | 4 | 5 | 6 | 9 |
| Pression | $p$ | 1 | 3 | | | | |
| Variable scalaire de turbulence | $k$, $\varepsilon$, $\varphi$, $\bar{f}$, $\omega$ | 1 | 3 | | 5 | 6 | |
| Tenseur de Reynolds | $R_{ij}$ | 1 | 3 | 4 | 5 | 6 | |
| *VarScalaire* (hormis variances) | | 1 | 3 | | 5 | 6 | |
| Variance des fluctuations d'une *VarScalaire* | | 1 | 3 | | | | |

Table 6.5: Valeurs admissibles de ICODCL pour chaque variable.

Il est important de connaître également les compatibilités à assurer entre les valeurs de ICODCL associées aux différentes variables (*Cf.* Points à traiter).

- Si ICODCL vaut 4, 5 ou 9 pour une composante de la vitesse, la même valeur de ICODCL doit être associée à toutes les composantes (symétrie, paroi ou sortie libre).
- Si ICODCL vaut 9 pour une composante de la vitesse, la valeur de ICODCL associée à la pression doit être 1 (sortie libre).
- Si ICODCL vaut 5 pour une composante de la vitesse, pour $k$, $\varepsilon$, $\varphi$, $\bar{f}$, $\omega$ ou pour une des composantes du tenseur de Reynolds $R_{ij}$, la valeur de ICODCL associée à toutes ces variables doit être 5 (paroi). La même remarque vaut pour ICODCL égal à 6.
- Si ICODCL vaut 4 pour une composante de la vitesse ou pour une des composantes du tenseur de Reynolds $R_{ij}$, la valeur de ICODCL associée à toutes ces variables doit être 4 (symétrie).
- Si ICODCL vaut 5 pour une variable *VarScalaire*, la valeur de ICODCL associée à toutes les composantes de la vitesse doit être 5 (paroi lisse), idem si ICODCL vaut 6 (paroi rugueuse).

**• Conversion des données utilisateur**

Une première étape (TYPECL) permet essentiellement de convertir les données des utilisateurs. Plus précisément, les actions suivantes sont réalisées :

- les faces de bord sont triées selon le type de condition à la limite qui leur a éventuellement été affecté (valeurs de ITYPFB) ;
- les types ITYPFB (si l'utilisateur en a défini) sont convertis en quadruplets définis pour chaque face IFAC et chaque variable IVAR (hormis lorsque l'utilisateur a affecté une valeur à ICODCL) : (ICODCL, RCODCL(IFAC,IVAR,1), RCODCL(IFAC,IVAR,2), RCODCL(IFAC,IVAR,3))
- pour la pression, la condition de sortie se traduit par une condition de type Dirichlet : on impose aux faces de bord la valeur de la pression $p$ au point $I'$, calculée à partir des données du pas de temps précédent (et le calcul du gradient de pression est donc réalisé au préalable). Cette valeur $p_{I'}$ est conservée dans le tableau COEFU(.,1), de dimension NFABOR, puis affectée à RCODCL(.,IPR,1) accompagnée d'un recalage (identique pour toutes les faces) destiné à assurer une valeur de Dirichlet de pression fixe P0 sur la première face de sortie rencontrée (*Cf.* Points à traiter).
- L'indicateur IDIRCL (tableau de dimension NVAR) est annulé pour repérer les variables pour lesquelles le système diffusif à résoudre n'est pas nécessairement inversible (le problème sera traité dans matrix). Il s'agit en pratique, dans le cadre actuel, de la variable pression lorsqu'il n'y a aucune face de sortie (cavité fermée par exemple). Plus précisément, IDIRCL est annulé pour les variables qui n'ont aucune condition de Dirichlet[14] et pour lesquelles l'indicateur ISTAT

[14] Le test contient une référence à ICODCL = 2 qui est un héritage d'une version dans laquelle les conditions de Dirichlet avec coefficient d'échange étaient distinguées des conditions de Dirichlet sans coefficient d'échange.

est positionné à zéro. Cet indicateur (de valeur 0 ou 1) est, dans le bilan explicite, le coefficient multiplicatif du terme de dérivée temporelle de la variable résolue ; il est nul pour la pression.

- la valeur du flux de masse est imprimée pour les ensembles de faces de bord de même type `ITYPFB` (il existe un type par défaut appliqué aux faces de bord pour lesquelles l'utilisateur n'a pas renseigné `ITYPFB`).

**• Vérifications**
Une vérification des conditions aux limites est ensuite menée (`vericl`) (l'utilisateur est prévenu et le programme interrompu en cas de problème).

- Le sous-programme s'assure que toutes les faces de bord ont reçu une condition à la limite pour toutes les variables.
- L'admissibilité des conditions aux limites (selon la nature de la variable à laquelle elles sont appliquées) est vérifiée.
- La cohérence des conditions aux limites entre les différentes variables est vérifiée (voir les tableaux 6.4 et 6.5).

**• Calculs préliminaires**
Différentes grandeurs sont ensuite construites pour préparer le traitement ultérieur des conditions aux limites.

- quand la connaissance de la distance à la paroi est nécessaire ($R_{ij} - \varepsilon$ avec termes d'écho de paroi, modèle LES de Smagorinsky avec amortissement de van Driest, modèle $k-\omega$ SST) et que la méthode de calcul "ancienne" a été choisie (`|ICDPAR|=2`, non compatible avec le parallélisme et la périodicité), pour chaque phase `IPHAS`, on détermine, pour chaque cellule `IEL`, le numéro de la face de paroi la plus proche (cette information est stockée dans `IA(IIFAPA(IPHAS)-1+IEL)`) ; cette distance est inutile pour le traitement des conditions aux limites, mais `condli` est le sous-programme dans lequel la réalisation de ce calcul est la plus simple.
- en cas de couplage avec SYRTHES, on détermine, pour le scalaire couplé $f$ (habituellement la température), les valeurs $f_{I'}$ dans les cellules de bord au moyen de la formule $f_{I'} = f_I + \underline{II'}\text{grad}\ f_I$ (ou au moyen de $f_{I'} = f_I$ si on traite le premier pas de temps[15] ou que la reconstruction a été desactivée par `ITBRRB`=0[16]). Ces valeurs sont stockées dans `TBORD` (tableau de dimension `NFABOR`) qui est utilisé en sortie de `condli` pour transmission d'information à SYRTHES.
- si des faces portent des conditions de symétrie ou de paroi, on construit les composantes de la vitesse $u_{j,I'}$ dans les cellules de bord (valeurs stockées dans le tableau local `COEFU` de dimension `NFABOR`×3) au moyen de la formule $u_{j,I'} = u_{j,I} + \underline{II'}\text{grad}\ u_{j\,I}$ (ou au moyen de $u_{j,I'} = u_{j,I}$ au premier pas de temps) ; en outre, si le modèle de turbulence est le $R_{ij} - \varepsilon$, on construit les composantes du tenseur de Reynolds dans les cellules de bord (valeurs stockées dans le tableau local `RIJIPB` de dimension `NFABOR`×6) de la même manière. Les deux tableaux `COEFU` et `RIJIPB` sont utilisés dans `clptur` et `clsyvt`.

**• Conditions aux limites de paroi**
On détermine alors (`clptur`) les conditions aux limites de paroi pour la vitesse et les grandeurs turbulentes. Les *VarScalaires* (excepté les variances) peuvent également recevoir un traitement particulier prenant en compte la couche limite si la condition qui leur est appliquée est de type Dirichlet (`ICODCL`=5). On traite donc en particulier dans `clptur` les conditions portant sur la température

[15] On ne dispose pas encore de conditions aux limites permettant le calcul du gradient lors du passage dans `condli` au premier pas de temps.
[16] ce qui est le cas par défaut

lorsque la paroi est à température imposée. Par contre, les conditions de flux sont traitées plus tard dans `condli`.

Le sous-programme `clptur` remplit les tableaux `COEFA` et `COEFB` pour les faces de paroi et les variables traitées. Ces tableaux représentent les coefficients $A_b$ et $B_b$. Plus précisément, pour une face `IFAC` et une variable `IVAR` donnée, les valeurs renseignées sont `COEFA(IFAC,ICLVAF)` et `COEFB(IFAC,ICLVAF)` avec `ICLVAF=ICLRTP(IVAR,ICOEFF)`. Deux autres valeurs sont également renseignées :
`COEFA(IFAC,ICLVAR)` et `COEFB(IFAC,ICLVAR)` avec `ICLVAR=ICLRTP(IVAR,ICOEF)`.
Elles ne diffèrent des précédentes qu'en $k-\varepsilon$ et représentent des coefficients $A_b$ et $B_b$ particuliers pour le calcul des gradients intervenant dans le terme de production turbulente (il y a donc bien deux types de conditions aux limites pour la vitesse dans ce cas, et chacun est utilisé lors du traitement d'un terme particulier : on se reportera à `clptur`).

**• Conditions aux limites de symétrie**
On détermine également (`clsyvt`) les conditions aux limites de symétrie pour les vecteurs (vitesse) et les tenseurs (de Reynolds). Elles nécessitent des rotations pour s'exprimer dans le repère lié au bord et le traitement lourd qui en résulte a donc été encapsulé. Le sous-programme complète les tableaux `COEFA` et `COEFB` aux faces de symétries pour la vitesse et les tensions de Reynolds (contrairement aux faces de paroi en $k-\varepsilon$, les valeurs `COEFA(IFAC,ICLVAF)` et `COEFB(IFAC,ICLVAF)` sont égales à `COEFA(IFAC,ICLVAR)` et `COEFB(IFAC,ICLVAR)`).

**• Autres conditions aux limites pour la vitesse**
On détermine, pour la vitesse, les conditions aux limites en sortie. On traite en outre les conditions restantes de type Dirichlet et Neumann. Ici également, on fournit une valeur de `COEFA` et de `COEFB` pour chaque face concernée (les valeurs `COEFA(IFAC,ICLVAF)` et `COEFB(IFAC,ICLVAF)` sont égales à `COEFA(IFAC,ICLVAR)` et `COEFB(IFAC,ICLVAR)`). Notons que, si le flux de masse est entrant à travers une face de sortie, on impose un Dirichlet homogène sur la vitesse (`COEFA(IFAC,ICLVAF)`=0, `COEFB(IFAC,ICLVAF)`=0) au lieu de la condition de Neumann homogène (`COEFA(IFAC,ICLVAF)`=0, `COEFB(IFAC,ICLVAF)`=1), lorsque le flux de masse est sortant.

**• Autres conditions aux limites**
On détermine enfin, pour la pression, les grandeurs turbulentes et les autres scalaires, les conditions aux limites de type Dirichlet et Neumann restantes. Ici également, on fournit donc une valeur de `COEFA` et de `COEFB` pour chaque face concernée (les valeurs `COEFA(IFAC,ICLVAF)` et `COEFB(IFAC,ICLVAF)` sont égales à `COEFA(IFAC,ICLVAR)` et `COEFB(IFAC,ICLVAR)`).

## 6.4 Points à traiter

**• Représentation des conditions par une valeur de face**
Bien que la méthode utilisée permette une simplicité et une homogénéité de traitement de toutes les conditions aux limites, elle est relativement restrictive au sens où une seule valeur ne suffit pas toujours pour représenter les conditions à appliquer lors du calcul de termes différents.

Ainsi, en $k-\varepsilon$ a-t-il été nécessaire, lors du calcul des conditions aux limites de paroi, de disposer de deux couples ($A_b$, $B_b$) afin de prendre en compte les conditions à appliquer pour le calcul de la contrainte tangentielle et celles à utiliser lors du calcul du terme de production (et un troisième jeu de coefficients serait nécessaire pour permettre le traitement des gradients intervenant dans les termes de gradient transposé, dans `vissec`).

Peut-être pourrait-il être utile de mettre en place une méthode permettant d'utiliser (au moins en certains points stratégiques du code) directement des forces, des contraintes ou des flux, sans passer nécessairement par le calcul d'une valeur de face.

**• Difficultés liées à la donnée d'un flux (entrée des données peu intuitive)**

L'utilisateur est invité à fournir, dans le cas d'une condition de Neumann, une valeur de flux en $W\,m^{-2}$ pour la température ou l'enthalpie. Par souci de cohérence, lorsqu'il souhaite fournir une condition de Neumann sur une autre variable, il est amené à fournir une grandeur similaire (en fait homogène à $\alpha\,\underline{\text{grad}}\ f.\underline{n}$, si $f$ est la variable), ce qui est peu naturel pour la vitesse, la pression ou un scalaire, dans le cas où l'on souhaite imposer la valeur du gradient normal à la face et non pas la valeur de $(\mu+\mu_t)\underline{\text{grad}}\ \underline{u}_j.\underline{n}$, de $\Delta t\underline{\text{grad}}\ (p).\underline{n}$ ou de $\dfrac{\alpha}{C}\underline{\text{grad}}\ (f).\underline{n}$.

**• Condition de sortie en pression**
La condition de pression en sortie se traduit par $p_f = p_{I'} + R1$ et le profil obtenu correspond au profil amont pris aux points $I'$ et recalé pour obtenir $p_0$ en un point $A$ arbitraire. Ce type de condition est appliqué sans précautions, mais n'est pas toujours justifié (une condition de Dirichlet basée sur la valeur calculée directement aux faces de bord pourrait être plus adaptée). Les hypothèses sont en particulier mises en défaut dans les cas suivants :

- la sortie est proche d'une zone où l'écoulement n'est pas établi en espace (ou varie en temps) ;
- la sortie n'est pas une surface perpendiculaire à l'écoulement ;
- le gradient de pression dans la direction normale à la sortie n'est pas le même pour toutes les faces de sortie (dans le cas de sortie multiples, par exemple, le gradient n'est probablement pas le même au travers de toutes les sorties) ;
- les points $I'$ ne sont pas sur une surface parallèle à la sortie (cas des maillage irréguliers par exemple).

On pourrait également tester la méthode d'Orlansky (équation de convection sur la pression).

Par ailleurs, en l'absence de condition de sortie, il pourrait peut-être se révéler utile de fixer une valeur de référence sur une cellule donnée ou de ramener la moyenne de la pression à une valeur de référence (avec le décalage du spectre, on assure l'inversibilité de la matrice à chaque pas de temps, mais il faudrait vérifier si la pression n'est pas susceptible de dériver au cours du calcul).

**• Termes non pris en compte**
Les conditions aux limites actuelles semblent causer des difficultés lors du traitement du terme de gradient transposé de la vitesse dans l'équation de Navier-Stokes (terme traité de manière explicite en temps). Il est possible de "débrancher" ce terme en positionnant le mot clé `IVISSE` à 0. Sa valeur par défaut est 1 (les termes en gradient transposé sont pris en compte).
On remarquera que l'indicateur `IVISSE` active non seulement les termes de gradient transposé en $(\ ^t\underline{\underline{\text{grad}}}\ \underline{v})$, mais également le terme en $-\dfrac{2}{3}\underline{\text{grad}}\ (\,\mu_{\,tot}\,\text{div}\,\underline{v})$ avec :

$$\mu_{\,tot} = \begin{cases} \mu_{\,l} & \text{en laminaire ou en modèle } R_{ij}-\varepsilon \\ \mu_{\,l}+\mu_t & \text{en modèle } k-\varepsilon. \end{cases} \qquad \text{(II.6.10)}$$

# 7- Sous-programme covofi

## 7.1 Fonction

Dans ce sous-programme, on résout :
⋆ soit l'équation de convection-diffusion d'un scalaire en présence de termes sources :

$$\frac{\partial(\rho a)}{\partial t} + \underbrace{\mathrm{div}\left((\rho \underline{u})\, a\right)}_{\text{convection}} - \underbrace{\mathrm{div}\left(K \underline{\mathrm{grad}}\ a\right)}_{\text{diffusion}} = T_s^{imp} a + T_s^{exp} + \Gamma\, a_i \tag{II.7.1}$$

Ici $a$ représente la valeur instantanée du scalaire en approche laminaire ou, en approche RANS, sa moyenne de Reynolds $\widetilde{a}$. Les deux approches étant exclusives et les équations obtenues similaires, on utilisera le plus souvent aussi la notation $a$ pour $\widetilde{a}$.
⋆ soit, dans le cas d'une modélisation RANS, la variance de la fluctuation d'un scalaire en présence de termes sources[1] :

$$\begin{aligned}\frac{\partial(\rho\widetilde{a''^2})}{\partial t} + \underbrace{\mathrm{div}\left((\rho\,\underline{u})\ \widetilde{a''^2}\right)}_{\text{convection}} - \underbrace{\mathrm{div}\left(K\ \underline{\mathrm{grad}}\ \widetilde{a''^2}\right)}_{\text{diffusion}} &= T_s^{imp}\widetilde{a''^2} + T_s^{exp} + \ \Gamma\,\widetilde{a''^2}_i \\ &\underbrace{+2\,\frac{\mu_t}{\sigma_t}(\underline{\mathrm{grad}}\ \widetilde{a})^2 - \frac{\rho\,\varepsilon}{R_f k}\ \widetilde{a''^2}}_{\text{termes de production et de dissipation dus à la turbulence moyenne}}\end{aligned} \tag{II.7.2}$$

$\widetilde{a''^2}$ représente ici la moyenne du carré des fluctuations[2] de $a$. $K$, $\Gamma$, $T_s^{imp}$ et $T_s^{exp}$ représentent respectivement le coefficient de diffusion, la valeur du terme source de masse, les termes sources implicite et explicite du scalaire $a$ ou $\widetilde{a''^2}$. $\mu_t$ et $\sigma_t$ sont respectivement la viscosité turbulente et le nombre de Schmidt ou de Prandtl turbulent, $\varepsilon$ est la dissipation de l'énergie turbulente $k$ et $R_f$ définit le rapport constant entre les échelles dissipatives de $k$ et de $\widetilde{a''^2}$ ($R_f$ est constant selon le modèle assez simple adopté ici).
On écrit les deux équations précédentes sous la forme commune suivante :

$$\frac{\partial(\rho f)}{\partial t} + \mathrm{div}\left((\rho\,\underline{u}) f\right) - \mathrm{div}\left(K \underline{\mathrm{grad}}\ f\right) = T_s^{imp} f + T_s^{exp} + \Gamma\, f_i + T_s^{pd} \tag{II.7.3}$$

avec, pour $f = a$ ou $f = \widetilde{a''^2}$ :

$$T_s^{pd} = \begin{cases} 0 & \text{pour } f = a, \\ 2\,\dfrac{\mu_t}{\sigma_t}(\underline{\mathrm{grad}}\ \widetilde{a})^2 - \dfrac{\rho\,\varepsilon}{R_f k}\ \widetilde{a''^2} & \text{pour } f = \widetilde{a''^2} \end{cases} \tag{II.7.4}$$

Le terme $\dfrac{\partial(\rho f)}{\partial t}$ est décomposé de la sorte :

$$\frac{\partial(\rho f)}{\partial t} = \rho\frac{\partial f}{\partial t} + f\frac{\partial \rho}{\partial t} \tag{II.7.5}$$

[1] Davroux A. et Archambeau F. : Calcul de la variance des fluctuations d'un scalaire dans le solveur commun. Application à l'expérience du CEGB dite "Jet in Pool", HE-41/99/043.

[2] $a$ et $\widetilde{a''^2}$, sous forme discrète en espace, correspondent donc en fait à des vecteurs dimensionnés à NCELET de composantes $a_I$ et $\widetilde{a''^2}_I$ respectivement, I décrivant l'ensemble des cellules.

En utilisant l'équation de conservation de la masse (cf. `preduv`), l'équation précédente s'écrit finalement :

$$\rho\,\frac{\partial f}{\partial t}+\mathrm{div}\left((\rho\,\underline{u})\,f\right)-\mathrm{div}\left(K\ \underline{\mathrm{grad}}\ f\right)=T_s^{imp}f+T_s^{exp}+\Gamma(f_i-f)+T_s^{pd}+f\,\mathrm{div}\left(\rho\,\underline{u}\right) \tag{II.7.6}$$

## 7.2 Discrétisation

Pour intégrer l'équation (II.7.6), une discrétisation temporelle de type $\theta$-schéma est appliquée à la variable résolue[3] :

$$f^{n+\theta}=\theta\ f^{n+1}+(1-\theta)\ f^{n} \tag{II.7.7}$$

L'équation (II.7.6) est discrétisée au temps $n+\theta$ en supposant les termes sources explicites pris au temps $n+\theta_S$, et ceux implicites en $n+\theta$. Par souci de clarté, on suppose, en l'absence d'indication, que les propriétes physiques $\Phi$ ($K$, $\rho$,...) et le flux de masse ($\rho\,\underline{u}$) sont pris respectivement aux instants $n+\theta_\Phi$ et $n+\theta_F$, où $\theta_\Phi$ et $\theta_F$ dépendent des schémas en temps spécifiquement utilisés pour ces grandeurs[4].

$$\begin{aligned}\rho\frac{f^{n+1}-f^{n}}{\Delta t}+\underbrace{\mathrm{div}\left((\rho\,\underline{u})\,f^{n+\theta}\right)}_{\text{convection}}-\underbrace{\mathrm{div}\left(K\ \underline{\mathrm{grad}}\ f^{n+\theta}\right)}_{\text{diffusion}}&=T_s^{imp}\,f^{n+\theta}+T_s^{exp,\,n+\theta_S}\\ +(\Gamma\,f_i)^{n+\theta_S}-\Gamma^n\,f^{n+\theta}+\ T_s^{pd,\,n+\theta_S}&+f^{n+\theta}\,\mathrm{div}\left(\rho\underline{u}\right)\end{aligned} \tag{II.7.8}$$

où :

$$T_s^{pd,\,n+\theta_S}=\begin{cases}0 & \text{pour } f=a,\\ 2\left[\dfrac{\mu_t}{\sigma_t}(\underline{\mathrm{grad}}\ \widetilde{a})^2\right]^{n+\theta_S}-\dfrac{\rho\,\varepsilon^n}{R_f\,k^n}(\widetilde{a^{"2}})^{n+\theta} & \text{pour } f=\widetilde{a^{"2}}\end{cases} \tag{II.7.9}$$

Le terme de production affecté d'un indice $n+\theta_S$ est un terme source explicite et il est donc traité comme tel :

$$\left[\frac{\mu_t}{\sigma_t}(\underline{\mathrm{grad}}\ \widetilde{a})^2\right]^{n+\theta_S}\quad=\quad(1+\theta_S)\ \frac{\mu_t^n}{\sigma_t}(\underline{\mathrm{grad}}\ \widetilde{a}^n)^2-\theta_S\ \frac{\mu_t^{n-1}}{\sigma_t}(\underline{\mathrm{grad}}\ \widetilde{a}^{n-1})^2 \tag{II.7.10}$$

L'équation (II.7.6) s'écrit :

$$\begin{aligned}&\rho\,\frac{f^{n+1}-f^{n}}{\Delta t}+\theta\ \mathrm{div}\left((\rho\,\underline{u})\,f^{n+1}\right)-\theta\ \mathrm{div}\left(K\ \underline{\mathrm{grad}}\ f^{n+1}\right)\\ &\quad-\left[\theta\,T_s^{imp}\ -\theta\,\Gamma^n+\theta\,T_s^{pd,\,imp}+\theta\ \mathrm{div}\left(\rho\ \underline{u}\right)\right]\ f^{n+1}\\ =(1-\theta)\ T_s^{imp}\,f^n+T_s^{exp,\,n+\theta_S}+(\Gamma\,f_i)^{n+\theta_S}&-(1-\theta)\ \Gamma^n\,f^n+T_s^{pd,\,exp}-\theta\,T_s^{pd,\,imp}\,f^n\\ +(1-\theta)\ f^n\,\mathrm{div}\left(\rho\ \underline{u}\right)&-(1-\theta)\ \mathrm{div}\left((\rho\,\underline{u})\,f^n\right)+(1-\theta)\ \mathrm{div}\left(K\ \underline{\mathrm{grad}}\ f^n\right)\end{aligned} \tag{II.7.11}$$

avec :

$$T_s^{pd,\,imp}=\begin{cases}0 & \text{pour } f=a,\\ -\dfrac{\rho\,\varepsilon^n}{R_f\,k^n} & \text{pour } f=\widetilde{a^{"2}}\end{cases} \tag{II.7.12}$$

$$T_s^{pd,\,exp}=\begin{cases}0 & \text{pour } f=a,\\ 2\ \left[\dfrac{\mu_t}{\sigma_t}(\underline{\mathrm{grad}}\ \widetilde{a})^2\right]^{n+\theta_S}-\dfrac{\rho\,\varepsilon^n}{R_f\,k^n}(\widetilde{a^{"2}})^n & \text{pour } f=\widetilde{a^{"2}}\end{cases} \tag{II.7.13}$$

On rappelle que, pour un scalaire $f$, le sous-programme `codits` résout une équation du type suivant

$$\begin{aligned}f_s^{imp}(f^{n+1}-f^n)+\theta\ \mathrm{div}((\rho\,\underline{u})\,f^{n+1})-\theta\ \mathrm{div}\left(K\,\underline{\mathrm{grad}}\ f^{n+1}\right)\\ =f_s^{exp}-\underbrace{(1-\theta)\ \mathrm{div}((\rho\,\underline{u})\,f^n)+(1-\theta)\ \mathrm{div}\left(K\,\underline{\mathrm{grad}}\ f^n\right)}_{\text{convection diffusion explicite}}\end{aligned} \tag{II.7.14}$$

[3] Si $\theta=1/2$, ou qu'une extrapolation est utilisée, le pas de temps $\Delta t$ est supposé uniforme en temps et en espace.
[4] cf. `introd`

$f_s^{exp}$ représente les termes sources discrétisés de manière explicite en temps (hormis contributions de la convection diffusion explicite provenant du $\theta$-schéma) et $f_s^{imp}\, f^{n+1}$ représente les termes linéaires en $f^{n+1}$ dans l'équation discrétisée en temps.
On réécrit l'équation (II.7.11) sous la forme (II.7.15) qui est ensuite résolue par `codits`.

$$\begin{gathered}\underbrace{\left(\frac{\rho}{\Delta t} - \theta\, T_s^{imp} + \theta\, \Gamma^n - \theta\, T_s^{pd,\,imp} - \theta\, \mathrm{div}\,(\rho\,\underline{u})\right)}_{f_s^{imp}}\, \delta f^{n+1} \\ +\theta\ \mathrm{div}(\,(\rho\underline{u})\, f^{n+1}) - \theta\ \mathrm{div}\,(K\,\underline{\mathrm{grad}}\ f^{n+1}) = \\ \underbrace{T_s^{imp}\, f^n + T_s^{exp,\,n+\theta_S} + (\Gamma f_i)^{n+\theta_S} - \Gamma^n\, f^n + T_s^{pd,\,exp} + f^n\,\mathrm{div}(\rho\,\underline{u})}_{f_s^{exp}} \\ -(1-\theta)\,\mathrm{div}(\,(\rho\underline{u})\, f^n\,) + (1-\theta)\,\mathrm{div}\,(K\,\underline{\mathrm{grad}}\ f^n)\end{gathered} \tag{II.7.15}$$

## 7.3 Mise en œuvre

On distingue deux cas suivant le type de schéma en temps choisi pour les termes sources :
• Si les temres sources ne sont pas extrapolés, toutes les contributions du second membre vont directement dans le vecteur `SMBRS`.
• Sinon, un vecteur supplémentaire est nécessaire afin de stocker les contributions du pas de temps précedent (`PROPCE`). Dans ce cas :

- le vecteur `PROPCE` sert à stocker les contributions explicites du second menbre au temps $n-1$ (pour l'extrapolation en $n+\theta_S$).
- le vecteur `SMBRS` est complété au fur et à mesure.

L'algorithme de ce sous-programme est le suivant :

- mise à zéro des vecteurs représentants le second membre (`SMBRS`) et de la diagonale de la matrice (`ROVSDT`).
- calcul des termes sources du scalaire définis par l'utilisateur en appelant le sous-programme `ustssc`.

  ⋆ Si les termes sources sont extrapolés, `SMBRS` reçoit $-\theta_S$ fois la contribution au temps $n-1$ des termes sources qui sont extrapolés (stockés dans `PROPCE`). La contribution des termes sources utisateurs (au pas temps courant) est répartie entre `PROPCE` (pour la partie $T_s^{exp}$ qui est à stocker en vue de l'extrapolation) et `SMBRS` (pour la partie explicite provenant de l'utilisation du $\theta$ schéma pour $T_s^{imp}$). La contribution implicite est alors mise dans `ROVSDT` (après multiplication par $\theta$) quel que soit son signe, afin de ne pas utiliser des discrétisations temporelles différentes entre deux pas de temps successifs, dans le cas par exemple où $T_s^{imp}$ change de signe[5].

  ⋆ Sinon la contibution de $T_s^{exp}$ est directement mise dans `SMBRS`. Celle de $T_s^{imp}$ est ajoutée à `ROVSDT` si elle est positive (de manière à conserver la dominance de la diagonale), ou explicitée et mise dans le second membre sinon.

- prise en compte des physiques particulières : arc électrique, rayonnement, combustion gaz et charbon pulvérisé. Seuls les vecteurs `ROVSDT` et `SMBRS` sont complétés (schéma d'ordre 1 sans extrapolation).

[5] cf. `preduv`

- ajout des termes sources de masse en $\Gamma\,(f_i - f)$ par appel au sous-programme `catsma`.

  $\star$ Si les termes sources sont extrapolés, le terme explicite en $\Gamma\, f_i$ est stocké dans `PROPCE`. Le $\theta$-schéma est appliqué au terme implicite, puis les contributions implicite et explicite réparties entre `ROVSDT` et `SMBRS`.

  $\star$ Sinon, la partie implicte en $-\Gamma\, f$ va dans `ROVSDT`, et tout le reste dans `SMBRS`.

- calcul du terme d'accumulation de masse en $\mathrm{div}(\rho\underline{u})$ par appel à `divmas` et ajout de sa contribution dans `SMBRS`, et dans `ROVSDT` après multiplication par $\theta$[6].

- ajout du terme instationnaire à `ROVSDT`.

- calcul des termes de production $(2\dfrac{\mu_t}{\sigma_t}(\underline{\mathrm{grad}}\ \widetilde{a})^2)$ et de dissipation $(-\dfrac{\rho\varepsilon}{R_f k}\widetilde{a''^2})$ si on étudie la variance des fluctuations d'un scalaire avec un modèle de turbulence de type RANS. Ce calcul s'effectue en calculant préalablement le gradient du scalaire $f$ par appel au sous-programme `grdcel`.

  $\star$ Si les termes sources sont extrapolés, la production est mise dans `PROPCE` puis l'énergie cinétique $k$ et la dissipation turbulentes $\varepsilon$ sont calculées (`XK` et `XE`) en fonction du modèle de turbulence utilisé. `SMBRS` reçoit $-\dfrac{\rho\varepsilon}{R_f k}\widetilde{a''^2}$ au temps $n$ et `ROVSDT` le coefficient d'implicitation $\dfrac{\rho\varepsilon}{R_f k}$ après multiplication par `THETAP` $= \theta$.

  $\star$ Sinon, la production va dans `SMBRS`, et la dissipation est répartie de la même manière que précédemment avec `THETAP` $= 1$.

- une fois la contribution de tous les termes sources calculés, le second membre est assemblé, et le vecteur `PROPCE` ajouté après multiplication par $1 + \theta_S$ à `SMBRS`, dans le cas où les termes sources sont extrapolés.

- calcul du coefficient de diffusion $K$ au centre des cellules, et des valeurs aux faces par appel au sous-programme `viscfa`.

- résolution de l'équation complète (avec les termes de convection diffusion) par un appel au sous-programme `codits` avec $f_s^{exp} =$ `SMBRS` et $f_s^{imp} =$ `ROVSDT`.

- ajustement (clipping) du scalaire ou de la fluctuation du scalaire en appelant le sous-programme `clpsca`.

- impression du bilan explicite d'expression $||\mathcal{E}_n(f^n) - \dfrac{\rho^n}{\Delta t}(\,f^{\,n+1} - f^n\,)||$ , où $||.||$ désigne la norme euclidienne.

On résume dans les tableaux II.7.16 et II.7.17 les différentes contributions (hors convection-diffusion) affectées à chacun des vecteurs `PROPCE`, `SMBRS` et `ROVSDT` suivant le schéma en temps choisi pour les termes sources. En l'absence d'indication, les propriétés physiques $\rho, \mu$, ... sont supposées prises en au temps $n + \theta_\Phi$, et le flux de masse ( $\rho\underline{u}$) pris au temps $n + \theta_F$, les valeurs de $\theta_F$ et de $\theta_\Phi$ dépendant du type de schéma sélectionné spécifiquement pour ces grandeurs[7].

[6] cette opération est faite quel que soit le schéma en temps de façon à rester cohérent avec ce qui est fait dans `bilsc2`

[7] cf. `introd`

Avec extrapolation des termes sources :

| | |
|---|---|
| $\texttt{ROVSDT}^n$ | $\dfrac{\rho}{\Delta t} - \theta\, T_s^{imp} - \theta\, \mathrm{div}(\,\rho \underline{u}) + \theta\, \Gamma^n + \theta\, \dfrac{\rho\, \varepsilon^n}{R_f\, k^n}$ |
| $\texttt{PROPCE}^n$ | $T_s^{exp,\,n} + \Gamma^n\, f_i^n + 2\, \dfrac{\mu_t^n}{\sigma_t} (\underline{\mathrm{grad}}\ f^n)^2$ |
| $\texttt{SMBRS}^n$ | $(1+\theta_S)\, \texttt{PROPCE}^n - \theta_S\, \texttt{PROPCE}^{n-1} + T_s^{imp}\, f^n + \mathrm{div}(\,\rho \underline{u})\, f^n - \Gamma^n\, f^n - \dfrac{\rho\, \varepsilon^n}{R_f\, k^n}\, f^n$ |

(II.7.16)

Sans extrapolation des termes sources :

| | |
|---|---|
| $\texttt{ROVSDT}^n$ | $\dfrac{\rho}{\Delta t} + Max(-T_s^{imp}, 0) - \theta\, \mathrm{div}(\,\rho \underline{u}) + \Gamma^n + \dfrac{\rho\, \varepsilon^n}{R_f\, k^n}$ |
| $\texttt{SMBRS}^n$ | $T_s^{exp} + T_s^{imp}\, f^n + \mathrm{div}(\,\rho \underline{u})\, f^n + \Gamma^n\, (\, f_i^n - f^n) - \dfrac{\rho\, \varepsilon^n}{R_f\, k^n}\, f^n + 2\, \dfrac{\mu_t}{\sigma_t} (\underline{\mathrm{grad}}\ f^n)^2$ |

(II.7.17)

## 7.4 Points à traiter

• **Intégration du terme de convection-diffusion**
Dans ce sous-programme, les points litigieux sont dus à l'intégration du terme de convection-diffusion. On renvoie donc le lecteur au sous-programme `bilsc2` qui les explicite.

## 7.5 Annexe 1 : Inversibilité de la matrice $\underline{\underline{EM}}_n$

Dans cette section, on va étudier plus particulièrement l'inversibilité de la matrice $\underline{\underline{EM}}_n$, matrice du système linéaire à résoudre associée à $\mathcal{EM}_n$ pour le cas d'un schéma en temps de type Euler implicite d'ordre un ($\theta = 1$). Pour toutes les notations, on se reportera à la documentation sur le sous-programme `covofi`. On va montrer que la démarche adoptée permet de s'assurer que la matrice des systèmes de convection-diffusion dans les cas courants est toujours inversible.

### 7.5.1 Introduction

Pour montrer l'inversibilité, on va utiliser le fait que la dominance stricte de la diagonale l'implique[8]. On cherche donc à déterminer sous quelles conditions les matrices de convection diffusion sont à diagonale strictement dominante.

On va montrer qu'en incluant dans la matrice le terme en $\mathrm{div}(\rho\,\underline{u})$ issu de $\dfrac{\partial\,\rho}{\partial\,t}$, on peut établir directement et exactement[9] la propriété. Par contre, si ce terme n'est pas pris en compte dans la matrice, il est nécessaire de faire intervenir le pendant discret de la relation :

$$\int_{\Omega_i} \mathrm{div}(\rho\,\underline{u})\,d\Omega = 0 \tag{II.7.18}$$

Cette relation n'est cependant vérifiée au niveau discret qu'à la précision du solveur de pression près (et, en tous les cas, ne peut être approchée au mieux qu'à la précision machine près). Il paraît donc préférable de s'en affranchir.

Avant d'entrer dans les détails de l'analyse, on rappelle quelques propriétés et définitions.

Soit $\underline{\underline{C}}$ une matrice carrée d'ordre N, d'élément générique $C_{ij}$. On a par définition :

Définition : La matrice $\underline{\underline{C}}$ est à diagonale **strictement dominante** *ssi*

$$\forall i \in [1,N], \quad |C_{ii}| > \sum_{j=1,\,j\neq i}^{j=N} |C_{ij}| \tag{II.7.19}$$

On convient de dire que $\underline{\underline{C}}$ est à diagonale **simplement dominante** *ssi* l'inégalité n'est pas stricte, soit :

$$\forall i \in [1,N], \quad |C_{ii}| \geqslant \sum_{j=1,\,j\neq i}^{j=N} |C_{ij}| \tag{II.7.20}$$

Remarque : Si, sur chaque ligne, la somme des éléments d'une matrice est nulle, que les éléments extradiagonaux sont négatifs et que les éléments diagonaux sont positifs, alors la matrice est à diagonale simplement dominante. Si la somme est strictement positive, la diagonale est strictement dominante.

On a l'implication suivante :

Propriété : Si la matrice $\underline{\underline{C}}$ est à diagonale strictement dominante, elle est inversible.

Cette propriété[10] se démontre simplement si l'on admet le théorème de Gerschgörin ci-dessous :

Théorème : Soit $\underline{\underline{B}}$ une matrice carrée d'ordre N dans $\mathbb{C} \times \mathbb{C}$, d'élément générique $B_{ij}$, les valeurs propres $\lambda_l$ de $B$ sont, dans le plan complexe, telles que $||\lambda_l - B_{ii}||_{\mathbb{C}} \leqslant \sum_{j=1,\,j\neq i}^{j=N} ||B_{ij}||_{\mathbb{C}}$

[8] Ce faisant, on choisit cependant une condition forte et la démonstration n'est probablement pas optimale.
[9] Hormis dans le cas de conditions aux limites mixtes, qu'il conviendrait d'examiner plus en détail.
[10] Lascaux, P. et Théodor, R. : Analyse Numérique Matricielle Appliquée à l'art de l'Ingénieur, Tome 2, Ed. Masson.

Si $B$ est à éléments réels, on écrira $||\lambda_l - B_{ii}||_{\mathbb{C}} \leqslant \sum_{j=1,\, j\neq i}^{j=N} |B_{ij}|$

Démontration de la propriété précédente :

Soit $C$ à diagonale strictement dominante à éléments réels. On montre qu'il est possible d'inverser le système $CX = S$ d'inconnue $X$, quel que soit le second membre $S$. Pour cela, on décompose $C$ en partie diagonale $(D)$ et extradiagonale $(-E)$ soit :

$$C = D - E$$

$C$ étant à diagonale strictement dominante, tous ses éléments diagonaux sont non nuls. $D$ est donc inversible (et les élements de l'inverse sont réels). On considère alors la suite[11] :

$$(X^n)_{n\in\mathbb{N}}, \qquad \text{avec} \qquad X^0 = D^{-1}S \qquad \text{et} \qquad DX^n = S + EX^{n-1}$$

On peut écrire :

$$X^n = \sum_{k=0}^{k=n} \left(D^{-1}E\right)^k D^{-1}S$$

Cette somme converge si le rayon spectral de $B = D^{-1}E$ est strictement inférieur à 1. Or, la matrice $C$ est à diagonale strictement dominante. On a donc pour tout $i \in \mathbb{N}$ (à partir de la relation (II.7.19) et en divisant par $|C_{ii}|$) :

$$\forall i \in [1, N], \quad \frac{|C_{ii}|}{|C_{ii}|} > \sum_{j=1,\, j\neq i}^{j=N} \frac{|C_{ij}|}{|C_{ii}|}$$

ce qui s'écrit encore :

$$\forall i \in [1, N], \quad \frac{|D_{ii}|}{|D_{ii}|} > \sum_{j=1,\, j\neq i}^{j=N} \frac{|E_{ij}|}{|D_{ii}|} = \sum_{j=1,\, j\neq i}^{j=N} |\left[D^{-1}E\right]_{ij}|$$

ou bien :

$$\forall i \in [1, N], \quad 1 > \sum_{j=1,\, j\neq i}^{j=N} |B_{ij}|$$

d'où, avec le théorème de Gerschgörin, une relation sur les valeurs propres $\lambda_l$ de $B$ :

$$\forall i \in [1, N], \quad ||\lambda_l - B_{ii}||_{\mathbb{C}} \leqslant \sum_{j=1,\, j\neq i}^{j=N} |B_{ij}| < 1$$

et comme $B_{ii} = 0$ :

$$||\lambda_l||_{\mathbb{C}} < 1$$

en particulier, la valeur propre dont la norme est la plus grande vérifie également cette équation. Ceci implique que le rayon spectral de $B$ est strictement inférieur à 1. La suite $(X^n)_{n\in\mathbb{N}}$ converge donc (et la méthode de Jacobi converge). Il existe donc une solution à l'équation $CX = S$. Cette solution est unique[12] et la matrice $C$ est donc inversible.

[11] On reconnaît la méthode de Jacobi

[12] On peut le voir "par l'absurde". En effet, supposons qu'il existe deux solutions distinctes $X_1$ et $X_2$ à l'équation $CX = S$. Alors $Y = X_2 - X_1$ vérifie $CY = 0$, soit $DY = -EY$, donc $D^{-1}EY = -Y$. Ceci signifie que $Y$ (qui n'est pas nul, par hypothèse) est vecteur propre de $D^{-1}E$ avec $\lambda = -1$ pour valeur propre associée. Or, le rayon spectral de $D^{-1}E$ est strictement inférieur à 1 et $\lambda = -1$ ne peut donc pas être une valeur propre de $D^{-1}E$. En conséquence, il ne peut exister qu'une seule solution à l'équation $CX = S$.

### 7.5.2 Avec prise en compte des termes issus de $\frac{\partial \rho}{\partial t}$ dans $\underline{\underline{EM}}_n$

#### Introduction

Pour montrer que la matrice $\underline{\underline{EM}}_n$ est inversible, on va montrer qu'elle est à diagonale strictement dominante. Pour cela, on va considérer successivement les contributions :

- des termes différentiels d'ordre 0 linéaires en $\delta f^{n+1,k+1}$,
- des termes issus de la prise en compte de $\frac{\partial \rho}{\partial t}$,
- des termes différentiels d'ordre 1 (convection),
- des termes différentiels d'ordre 2 (diffusion).

Pour chacune de ces contributions, on va examiner la dominance de la diagonale de l'opérateur linéaire associé. Si, pour chaque contribution, la dominance de la diagonale est acquise, on pourra alors conclure à la dominance de la diagonale pour la matrice (somme) complète[13] $\underline{\underline{EM}}_n$ et donc à son inversibilité.

#### Contributions des termes différentiels d'ordre 0 linéaires en $\delta f^{n+1,k+1}$

L'unique contribution est sur la diagonale : il faut donc vérifier qu'elle est strictement positive.

Pour chaque ligne $I$, ${f_s^{imp}}_I$ (cf. (II.7.15)) contient au minimum la quantité strictement positive[14] $\frac{\rho_I^n\ |\Omega_i|}{\Delta t}$. Les autres expressions, $(-\,|\Omega_i|\,(T_s^{imp})_I\ ,\ \ +|\Omega_i|\,\Gamma_I\ ,\ \ -\,|\Omega_i|\,(T_s^{pd,imp})_I)$, lorsqu'elles existent, contribuent toutes positivement[15].

L'opérateur linéaire associé à ces contributions vérifie donc bien la **dominance stricte** de la diagonale (propriété 1). Ce n'est cependant pas vrai si on extrapole les termes source, à cause de $T_s^{imp}$. Il en résulte une contrainte sur la valeur du pas de temps.

#### Contributions des termes différentiels d'ordre 1 et des termes issus de la prise en compte de $\frac{\partial \rho}{\partial t}$

Les termes considérés sont au nombre de deux dans (II.7.11) :

- la contribution issue de la prise en compte de $\frac{\partial \rho}{\partial t}$ se retrouve dans ${f_s^{imp}}_I$ (équation II.7.15),
- la contribution du terme de convection linéarisé.

Après intégration spatiale, la somme de ces deux termes discrets s'écrit :

$$\frac{1}{2}\sum_{j\in Vois(i)}\left[(\,-m_{ij}^n+|\,m_{ij}^n|\,)\,\delta f_I^{n+1,k+1}+(\,m_{ij}^n-|\,m_{ij}^n|)\,\delta f_J^{n+1,k+1}\right] \qquad \text{(II.7.21)}$$

$$+\frac{1}{2}\sum_{k\in\gamma_b(i)}\left[(\,-m_{b_{ik}}^n+|\,m_{b_{ik}}^n|\,)\,\delta f_I^{n+1,k+1}+(\,m_{b_{ik}}^n-|m_{b_{ik}}^n|)\,\delta f_{b_{ik}}^{n+1,k+1}\right] \qquad \text{(II.7.22)}$$

Pour chaque ligne $I$, on va chercher les propriétés de dominance de la diagonale en traitant séparément les faces internes (équation (II.7.21)) et les faces de bord (équation (II.7.22)).

• la contribution des **faces internes** $ij$ (facteur de $\delta f_I^{n+1,k+1}$) à la diagonale est positive ; la contribution aux extradiagonales est négative (facteur de $\delta f_J^{n+1,k+1}$) et la somme de ces contributions

[13] Ce raisonnement n'est pas optimal (la somme de valeurs absolues étant supérieure à la valeur absolue de la somme), mais permet d'obtenir des conclusions dans le cas présent (condition suffisante).

[14] Ceci permettra de conclure à la stricte dominance de la diagonale de la matrice somme complète $\underline{\underline{EM}}_n$.

[15] Le terme de dissipation $\rho\frac{1}{R_f}\frac{\varepsilon}{k}$, spécifique à l'étude de la variance des fluctuations, est positif par définition et ne remet donc pas en cause la conclusion.

est exactement nulle (équation (II.7.21)). Si l'on note $C_{IJ}$ les coefficients de la matrice issus de la contribution de ces termes, on a donc $|C_{II}| \geqslant \sum_{J=1,\, J\neq I}^{J=N} |C_{IJ}|$ qui traduit la **dominance "simple"** (l'inégalité n'est pas "stricte") de la diagonale et règle la question des contributions des faces internes.

• la contribution des **faces de bord** doit être réécrite en utilisant l'expression des conditions aux limites sur $f$ pour préciser la valeur de $\delta f_{b_{ik}}$ (on omet l'exposant $(n+1, k+1)$ pour alléger les notations) :

- pour une condition de Dirichlet : $\delta f_{b_{ik}} = 0$,
- pour une condition de Neumann : $\delta f_{b_{ik}} = \delta f_I$,
- pour une condition mixte ($f_{b_{ik}} = \alpha + \beta f_i$) : $\delta f_{b_{ik}} = \beta\, \delta f_I$.

Pour la contribution des faces de bord, il faut alors considérer deux cas de figure possibles.

- **Le flux de masse au bord est positif ou nul** ( $m^n_{b_{ik}} = (\rho\, \underline{u})^n_{b_{ik}} . \underline{S}_{b_{ik}} \geqslant 0$). Cette situation correspond par exemple aux sorties standards (fluide sortant du domaine), aux symétries ou aux parois étanches (flux de masse nul). Les contributions aux faces de bord sont alors toutes nulles, quelles que soient les conditions aux limites portant sur la variable $f$. En conséquence, la diagonale issue de ces contribution est **simplement dominante**.

- **Le flux de masse au bord est strictement négatif**. Cette situation correspond à une entrée de fluide dans le domaine. Les contributions considérées s'écrivent alors :

$$\sum_{k\in\gamma_b(i)} \left[ (\, -m^n_{b_{ik}}\,)\, \delta f_I^{\,n+1,k+1} + (\, m^n_{b_{ik}}\,)\, \delta f_{b_{ik}}^{\,n+1,k+1} \right] \tag{II.7.23}$$

Il convient alors de distinguer plusieurs situations, selon le type de condition à la limite portant sur $f$ :

⋆ si la condition à la limite de $f$ est de type **Dirichlet**, seule subsiste une contribution positive ou nulle à la diagonale, qui assure donc la **dominance simple** :

$$\sum_{k\in\gamma_b(i)} (\, -m^n_{b_{ik}}\,)\, \delta f_I^{\,n+1,k+1} \tag{II.7.24}$$

⋆ si la condition à la limite de $f$ est de type **Neumann**, la somme des contributions dues aux faces de bord est alors nulle, ce qui assure donc la **dominance simple**.

⋆ si la condition à la limite de $f$ est de type **mixte**, la contribution des faces de bord est sur la diagonale et vaut :

$$\frac{1}{2} \sum_{k\in\gamma_b(i)} (1-\beta)(\, -m^n_{b_{ik}}\,)\, \delta f_I^{\,n+1,k+1} \tag{II.7.25}$$

On ne peut pas se prononcer quand à la dominance de la diagonale, à cause de la présence de $(1-\beta)$ (la valeur de $\beta$ est fixée par l'utilisateur) et la démarche adoptée ici **ne permet donc pas de conclure**. Il faut néanmoins noter que cette situation est rare dans les calculs standards. Elle demande un complément d'analyse et sera pour le moment exclue des considérations exposées dans le présent document.

**On peut conclure**, quand il n'y a pas de condition à la limite de type mixte, que la matrice associée aux contributions des termes différentiels d'ordre 1 (convectifs) et à la prise en compte des termes issus de $\dfrac{\partial \rho}{\partial t}$ et est à diagonale **simplement dominante**.

### Contributions des termes différentiels d'ordre 2

On va considérer enfin les contributions des termes différentiels d'ordre 2 (issus du terme $-\ \mathrm{div}\,(K^n\ \underline{\mathrm{grad}}\ \delta f^{n+1,k+1})$). Pour ces termes, la contribution à la diagonale est positive[16], négative aux extradiagonales[16], compte tenu de :

$$\begin{aligned}&\int_{\Omega_i} -\ \mathrm{div}\,(K^n\ \underline{\mathrm{grad}}\ \delta f^{n+1,k+1})\,d\Omega \\ &= -\sum_{j\in Vois(i)} K^n_{ij}\frac{\delta f_J^{n+1,k+1}-\delta f_I^{n+1,k+1}}{\overline{I'J'}}\,.\,S_{ij} - \sum_{k\in\gamma_b(i)} K^n_{b_{ik}}\frac{\delta f_{b_{ik}}^{n+1,k+1}-\delta f_I^{n+1,k+1}}{\overline{I'F}}\,.\,S_{b_{ik}}\end{aligned} \tag{II.7.26}$$

Considérons deux cas :

- la cellule courante $I$ n'a **que des faces internes** au domaine de calcul (pas de faces de bord). La somme des contributions est nulle. On a donc **dominance simple** de la diagonale.

- la cellule courante $I$ a **des faces de bord**. La somme des contributions diagonales et extra-diagonales est positive quand on a une condition à la limite de type **Dirichlet** ou de type **Neumann** sur $f$. La diagonale est alors **strictement dominante**. Lorsqu'il y a des conditions à la limite de type mixte, il n'est plus possible de conclure (situation écartée précédemment).

**On peut conclure**, quand il n'y a pas de condition à la limite de type mixte, que la matrice associée aux contributions des termes différentiels d'ordre 2 est au moins à diagonale **simplement dominante**.

### Conclusion

En travaillant sur des maillages non pathologiques (à transmittivité positive, voir la note de bas de page numéro 16) et en n'imposant pas de condition à la limite de type mixte sur les variables, on peut donc conclure que $\underline{\underline{EM}}_n$ est la somme de matrices à diagonale simplement dominante et d'une matrice à diagonale strictement dominante (paragraphe 7.5.2). Elle est donc à **diagonale strictement dominante**, et donc **inversible** (de plus, la méthode itérative de Jacobi converge).

## 7.5.3 Sans prise en compte des termes issus de $\dfrac{\partial\rho}{\partial t}$ dans $\underline{\underline{EM}}_n$

### Introduction

Pour identifier les cas dans lesquels la matrice $\underline{\underline{EM}}_n$ est inversible, on va rechercher les conditions qui assurent la dominance de la diagonale. Par rapport à l'analyse présentée au paragraphe 7.5.2, seules diffèrent les considérations relatives aux contributions des termes différentiels d'ordre 1, puisqu'elles sont traitées au paragraphe 7.5.2 avec les termes issus de la prise en compte de $\dfrac{\partial\rho}{\partial t}$.

### Contributions des termes différentiels d'ordre 1

La contribution du terme de convection est la seule à prendre en compte. Elle s'écrit, d'après les équations (II.7.15) et la discrétisation explicitée pour le sous-programme `covofi` :

$$\frac{1}{2}\sum_{j\in Vois(i)}\left[(\ +m^n_{ij}+|\ m^n_{ij}|\ )\,\delta f_I^{n+1,k+1}+(\ m^n_{ij}-|\ m^n_{ij}|)\,\delta f_J^{n+1,k+1}\right] \tag{II.7.27}$$

[16] Ceci n'est en fait pas toujours vrai. En effet, pour chaque face $ij$, la transmittivité $\frac{K^n}{\overline{I'J'}}S_{ij}$ fait intervenir la mesure algébrique du segment $I'J'$, où $I'$ et $J'$ sont les projetés orthogonaux sur la normale à la face du centre des cellules voisines. Cette grandeur est une valeur algébrique et peut théoriquement devenir négative sur certains maillages pathologiques, contenant par exemple des mailles non convexes. On pourra se reporter au dernier point à traiter du sous-programme `matrix`.

$$\frac{1}{2}\sum_{k\in\gamma_b(i)}\left[(\,+m^n_{b_{ik}}+|\,m^n_{b_{ik}}|\,)\,\delta f_I^{\,n+1,k+1}+(\,m^n_{b_{ik}}-|m^n_{b_{ik}}|)\,\delta f_{b_{ik}}^{\,n+1,k+1}\right] \tag{II.7.28}$$

On constate que pour chaque ligne $I$, la contribution des faces internes (facteur de $\delta f_I^{\,n+1,k+1}$) à la diagonale est positive et qu'elle est négative aux extradiagonales (facteur de $\delta f_J^{\,n+1,k+1}$). **Cependant**, contrairement au cas présenté au paragraphe 7.5.2, la somme de ces contributions n'est pas nulle dans le cas général. Pour obtenir un résultat quant à la dominance de la diagonale, il faut faire intervenir la version discrète de la propriété (II.7.18) rappelée ci-dessous :

$$\int_{\Omega_i}\mathrm{div}(\rho\,\underline{u})\,d\Omega=0$$

Soit, sous forme discrète :

$$\sum_{j\in Vois(i)}m^n_{ij}+\sum_{k\in\gamma_b(i)}m^n_{b_{ik}}\;=0 \tag{II.7.29}$$

Il n'est donc pas possible d'analyser séparément les contributions des faces internes et celles des faces de bord (contrairement à la situation rencontrée au paragraphe 7.5.2). On se place ci-après dans le cas général d'une cellule qui a des faces internes *et* des faces de bord (si elle n'a que des faces internes, la démonstration est la même, mais plus simple. On peut l'écrire en considérant formellement que la cellule "a zéro faces de bord", c'est à dire que $\gamma_b(i)$ est l'ensemble vide).

Il faut alors considérer deux cas de figure, selon la valeur du flux de masse aux faces de bord (éventuelles) de la cellule :

- **Le flux de masse au bord est positif ou nul** ( $m^n_{b_{ik}}=(\rho\,\underline{u})^n_{b_{ik}}\cdot\underline{S}_{b_{ik}}\geqslant 0$). Cette situation correspond à des cellules qui ont des faces de bord de sortie standard (fluide sortant du domaine), de symétrie ou de paroi étanche (flux de masse nul). Les contributions s'écrivent alors :

$$\frac{1}{2}\sum_{j\in Vois(i)}\left[(\,+m^n_{ij}+|\,m^n_{ij}|\,)\,\delta f_I^{\,n+1,k+1}+(\,m^n_{ij}-|\,m^n_{ij}|)\,\delta f_J^{\,n+1,k+1}\right]+\sum_{k\in\gamma_b(i)}m^n_{b_{ik}}\,\delta f_I^{\,n+1,k+1} \tag{II.7.30}$$

  Dans ce cas, la somme des contributions à la diagonale est positive, les contributions aux extradiagonales sont négatives et, avec la relation (II.7.29), on vérifie que la somme des contributions diagonales et extradiagonales est nulle. On a donc **dominance simple** de la diagonale.

- **Le flux de masse au bord est strictement négatif**. Cette situation correspond à des cellules qui ont des faces de bord d'entrée standard (entrée de fluide dans le domaine). Les contributions considérées s'écrivent alors :

$$\frac{1}{2}\sum_{j\in Vois(i)}\left[(\,+m^n_{ij}+|\,m^n_{ij}|\,)\,\delta f_I^{\,n+1,k+1}+(\,m^n_{ij}-|\,m^n_{ij}|)\,\delta f_J^{\,n+1,k+1}\right]+\sum_{k\in\gamma_b(i)}m^n_{b_{ik}}\,\delta f_{b_{ik}}^{\,n+1,k+1} \tag{II.7.31}$$

  Il convient alors de distinguer plusieurs situations, selon le type de condition à la limite portant sur $f$ (on omet l'exposant $(n+1,k+1)$ pour alléger les notations) :
  - pour une condition de Dirichlet : $\delta f_{b_{ik}}=0$,
  - pour une condition de Neumann : $\delta f_{b_{ik}}=\delta f_I$,
  - pour une condition mixte ($f_{b_{ik}}=\alpha+\beta f_I$) : $\delta f_{b_{ik}}=\beta\,\delta f_I$.

  Selon le cas on se trouve dans une des situations suivantes :

  ⋆ si la condition à la limite de $f$ est de type **Dirichlet**, la contribution des faces de bord est nulle dans la matrice. La contribution des faces internes à la diagonale est positive, la contribution aux extradiagonales négative et la somme de ces contributions vaut $\sum_{j\in Vois(i)}m^n_{ij}$, soit avec la relation (II.7.29) :

$$\sum_{j\in Vois(i)}m^n_{ij}=-\sum_{k\in\gamma_b(i)}m^n_{b_{ik}}$$

Elle est strictement positive et la diagonale est donc **strictement dominante**.

⋆ si la condition à la limite de $f$ est de type **Neumann**, la contribution des faces de bord se réduit au terme : $\sum_{k\in\gamma_b(i)} m^n_{b_{ik}}\,\delta f_I^{n+1,k+1}$. La somme des contributions à la diagonale est alors $SC_i$:

$$SC_i = \frac{1}{2}\sum_{j\in Vois(i)} (\ +m^n_{ij} + |\ m^n_{ij}|\ ) + \sum_{k\in\gamma_b(i)} m^n_{b_{ik}}$$

En utilisant deux fois la relation (II.7.29), on obtient donc pour la diagonale :

$$SC_i = \frac{1}{2}\left[\sum_{j\in Vois(i)} |\ m^n_{ij}| + \sum_{k\in\gamma_b(i)} m^n_{b_{ik}}\right] = \frac{1}{2}\left[\sum_{j\in Vois(i)} (\ |\ m^n_{ij}| -\ m^n_{ij}\ )\right]$$

Cette grandeur est positive et égale à l'opposé de la somme des termes extradiagonaux qui sont tous négatifs. La diagonale est donc **simplement dominante**.

⋆ si la condition à la limite de $f$ est de type **mixte**, la somme des contributions dues aux faces de bord est :

$$\sum_{k\in\gamma_b(i)} \beta\ m^n_{b_{ik}}\ \delta f_I^{n+1,k+1} \tag{II.7.32}$$

On ne peut donc **pas conclure** quant au signe de cette contribution, le facteur $\beta$ étant choisi librement par l'utilisateur. Cette situation a été écartée dans le paragraphe 7.5.2.

**On peut donc conclure**, quand il n'y a pas de condition à la limite de type mixte, que la matrice associée aux contributions des termes différentiels d'ordre 1 (convectifs) est à diagonale **simplement dominante**, à condition que la relation (II.7.29) soit vérifiée exactement.

## Conclusion

En travaillant sur des maillages non pathologiques (à transmittivité positive, voir la note de bas de page numéro 16) et en n'imposant pas de condition à la limite de type mixte sur les variables, on peut donc conclure que $\underline{\underline{EM}}_n$ est à **diagonale strictement dominante**, donc **inversible** (et la méthode itérative de Jacobi converge) à condition que la relation (II.7.29) soit vérifiée exactement. Ce n'est généralement pas le cas (la précision du solveur de pression et la précision machine sont finies). Même si la contribution diagonale en $\dfrac{\rho_I^n\ |\Omega_i|}{\Delta t}$ peut suffire à assurer la dominance, on a cependant souhaité, dans *Code_Saturne*, s'affranchir du problème potentiel en prenant en compte les termes issus de $\dfrac{\partial\,\rho}{\partial\,t}$ dans la matrice.

# 7.6 Annexe 2 : Remarques à propos du respect du principe du maximum discret

## 7.6.1 Introduction

Les considérations exposées ici sont relatives au fait que, en continu, une variable qui n'est *que* convectée par un champ de débit à divergence nulle doit rester dans les bornes minimales et maximales définies par les conditions initiales et par les conditions aux limites en espace. Ainsi, les valeurs d'un scalaire passif initialement nul dont les conditions aux limites sont des conditions de Neumann homogène et des conditions de Dirichlet de valeur 1 devront nécessairement rester comprises dans l'intervalle $[0\,;\,1]$. C'est ce que l'on entend ici par "principe du maximum".

Soient $\underline{u}$ un champ de vitesse figé et connu et $t$ un réel positif. On considère le problème modèle $\mathcal{P}$ de convection des variables scalaires $\rho$ et $\rho\,f$, défini par :

$$\left\{\begin{array}{l} \dfrac{\partial \rho}{\partial t} + \mathrm{div}\,(\rho\,\underline{u}) = 0 \\ \dfrac{\partial (\rho f)}{\partial t} + \mathrm{div}\,((\rho\,\underline{u})\,f) = 0 \end{array}\right. \tag{II.7.33}$$

avec une condition initiale $f^0$ donnée ainsi que des conditions aux limites associées sur $f$ de type Dirichlet ou Neumann.
Dans *Code_Saturne*, la deuxième équation de $\mathcal{P}$ est réécrite en continu, en utilisant la première, sous la forme :

$$\rho\,\frac{\partial f}{\partial t} - f\,\mathrm{div}\,(\rho\,\underline{u}) + \mathrm{div}\,((\rho\,\underline{u})\,f) = 0 \tag{II.7.34}$$

et discrétisée temporellement comme suit :

$$\rho^n\,\frac{f^{n+1}-f^n}{\Delta t} - f^{n+1}\,\mathrm{div}\,(\rho\,\underline{u})^n + \mathrm{div}\,((\rho\,\underline{u})^n\,f^{n+1}) = 0 \tag{II.7.35}$$

Dans un premier temps, on va étudier la discrétisation spatiale associée à (II.7.35), qui correspond donc à la prise en compte de la contribution de $\dfrac{\partial \rho}{\partial t}$ dans l'équation en continu (et se traduit par la présence de $-\mathrm{div}((\rho\,\underline{u})^n)$ dans l'expression de ${f_s^{imp}}_I$), puis dans un deuxième temps, la discrétisation spatiale de l'expression ;

$$\rho^n\,\frac{f^{n+1}-f^n}{\Delta t} + \mathrm{div}\,((\rho\,\underline{u})^n\,f^{n+1}) = 0 \tag{II.7.36}$$

qui correspond à un problème de convection pure classique.

On étudiera ensuite un exemple simplifié (monodimensionnel à masse volumique constante).

Les considérations présentes mériteraient d'être approfondies.

### 7.6.2 Cas général

**Discrétisation spatiale de** $\rho^n \dfrac{f^{n+1}-f^n}{\Delta t} - f^{n+1}\,\mathbf{div}\,(\rho\,\underline{u})^n + \mathbf{div}\,((\rho\,\underline{u})^n\,f^{n+1}) = 0$

En intégrant sur une cellule $\Omega_i$ à l'aide de la formulation volumes finis habituelle, on obtient :

$$\begin{aligned}
&\int_{\Omega_i} [\rho^n \frac{f^{n+1}-f^n}{\Delta t} - f^{n+1}\,\mathrm{div}\,(\rho\,\underline{u})^n + \mathrm{div}\,((\rho\,\underline{u})^n\,f^{n+1})]\,d\Omega \\
&= \left[\rho_I^n \frac{|\Omega_i|}{\Delta t} - \Big(\sum_{j\in Vois(i)} (\rho\,\underline{u})_{ij}^n \cdot \underline{S}_{ij} + \sum_{k\in\gamma_b(i)} (\rho\,\underline{u})_{b_{ik}}^n \cdot \underline{S}_{b_{ik}}\Big)\right] f_I^{n+1} \\
&+ \sum_{j\in Vois(i)} (\rho\,\underline{u})_{ij}^n \cdot \underline{S}_{ij}\, f_{f\,ij}^{n+1} + \sum_{k\in\gamma_b(i)} (\rho\,\underline{u})_{b_{ik}}^n \cdot \underline{S}_{b_{ik}}\, f_{f\,b_{ik}}^{n+1} \\
&-\rho_I^n \frac{|\Omega_i|}{\Delta t} f_I^n
\end{aligned} \tag{II.7.37}$$

où $f_{f\,ij}^{n+1}$ et $f_{f\,b_{ik}}^{n+1}$ sont les valeurs de $f$ aux faces internes et de bord issues du choix du schéma convectif.

En reprenant les notations précédentes, en imposant un schéma décentré amont au premier membre (*i.e.* en exprimant $\delta f_{ij}^{n+1,k+1}$ et $\delta f_{b_{ik}}^{n+1,k+1}$) et en raisonnant en incréments (cf. sous-programme `navsto`), on aboutit à :

$$\begin{aligned}
&\rho_I^n \frac{|\Omega_i|}{\Delta t}\,\delta f_I^{n+1,k} \\
&+\frac{1}{2}\sum_{j\in Vois(i)} \Big[(-m_{ij}^n + |\,m_{ij}^n|\,)\,\delta f_I^{n+1,k+1} + (\,m_{ij}^n - |\,m_{ij}^n|)\,\delta f_J^{n+1,k+1}\Big] \\
&+\frac{1}{2}\sum_{k\in\gamma_b(i)} \Big[(-m_{b_{ik}}^n + |\,m_{b_{ik}}^n|\,)\,\delta f_I^{n+1,k+1} + (\,m_{b_{ik}}^n - |m_{b_{ik}}^n|)\,\delta f_{b_{ik}}^{n+1,k+1}\Big] \\
&= -\rho^n \frac{|\Omega_i|}{\Delta t}\,(f_I^{n+1,k} - f_I^n) \\
&- \left[\sum_{j\in Vois(i)} (\rho\,\underline{u})_{ij}^n \cdot \underline{S}_{ij}\, f_{ij}^{n+1,k} + \sum_{k\in\gamma_b(i)} (\rho\,\underline{u})_{b_{ik}}^n \cdot \underline{S}_{b_{ik}}\, f_{b_{ik}}^{n+1,k}\right] \\
&- \left(\sum_{j\in Vois(i)} (\rho\,\underline{u})_{ij}^n \cdot \underline{S}_{ij} + \sum_{k\in\gamma_b(i)} (\rho\,\underline{u})_{b_{ik}}^n \cdot \underline{S}_{b_{ik}}\right) f_I^{n+1,k}
\end{aligned} \tag{II.7.38}$$

avec :

$$\left\{\begin{array}{l} f_I^{n+1,0} = f_I^n \\ \delta f_I^{n+1,k+1} = f_I^{n+1,k+1} - f_I^{n+1,k}, k \in \mathbb{N} \end{array}\right. \tag{II.7.39}$$

et $(f^{n+1,k})_{k\in\mathbb{N}}$ suite convergeant vers $f^{n+1}$, $n$ entier donné, solution de (II.7.35) .

**Discrétisation spatiale de** $\rho^n \dfrac{f^{n+1}-f^n}{\Delta t} + \mathbf{div}\left((\rho\,\underline{u})^n\, f^{n+1}\right) = 0$

En procédant de façon analogue et en adoptant les mêmes hypothèses, on obtient :

$$\begin{aligned}
&\rho^n \frac{|\Omega_i|}{\Delta t}\,\delta f_I^{n+1,k+1} \\
&+\frac{1}{2}\sum_{j\in Vois(i)}\left[(+m_{ij}^n+|\,m_{ij}^n|\,)\,\delta f_I^{n+1,k+1}+(\,m_{ij}^n-|\,m_{ij}^n|)\,\delta f_J^{n+1,k+1}\right] \\
&+\frac{1}{2}\sum_{k\in\gamma_b(i)}\left[(+m_{b_{ik}}^n+|\,m_{b_{ik}}^n|\,)\,\delta f_I^{n+1,k+1}+(\,m_{b_{ik}}^n-|m_{b_{ik}}^n|)\,\delta f_{b_{ik}}^{n+1,k+1}\right] \\
&=-\rho^n\frac{|\Omega_i|}{\Delta t}\,(\,f_I^{n+1,k}-f_I^n\,) \\
&-\left[\sum_{j\in Vois(i)}(\rho\,\underline{u})_{ij}^n\cdot\underline{S}_{ij}\,f_{f\,ij}^{n+1,k}+\sum_{k\in\gamma_b(i)}(\rho\,\underline{u})_{b_{ik}}^n\cdot\underline{S}_{b_{ik}}\,f_{f\,b_{ik}}^{n+1,k}\right]
\end{aligned} \tag{II.7.40}$$

(où $f_{f\,ij}^{n+1}$ et $f_{f\,b_{ik}}^{n+1}$ sont les valeurs de $f$ aux faces internes et de bord issues du choix du schéma convectif)

avec :

$$\left\{\begin{array}{l} f_I^{n+1,0}=f_I^n \\ \delta f_I^{n+1,k+1}=f_I^{n+1,k+1}-f_I^{n+1,k},\,k\in\mathbb{N} \end{array}\right. \tag{II.7.41}$$

et $(f^{n+1,k})_{k\in\mathbb{N}}$ suite convergeant vers $f^{n+1}$, $n$ entier donné, solution de (II.7.36) .

## 7.6.3 Exemple pour le principe du maximum

On va maintenant se placer en monodimensionnel, sur un maillage régulier formé de trois cellules de pas $h$ constant (figure II.7.1) et étudier le comportement du premier membre pour les deux types d'expressions, entre le pas de temps $n\,\Delta t$ et le pas de temps $(n+1)\,\Delta t$, avec, comme condition initiale $f_1^0=f_2^0=f_3^0=0$ et comme conditions aux limites, une de type Dirichlet et l'autre de type Neumann homogène :

$$\left\{\begin{array}{l} f_{b_1}=1 \\ \left.\dfrac{\partial f}{\partial x}\right|_{b_2}=0 \end{array}\right. \tag{II.7.42}$$

On supposera de plus que :

- ⋆ le schéma convectif utilisé est le schéma upwind
- ⋆ la masse volumique est constante
- ⋆ $(\rho\,u)_{b_1}^n>0$, $(\rho\,u)_{12}^n>0$ , $(\rho\,u)_{23}^n>0$ , $(\rho\,u)_{b_2}^n>0$ et $S_{b_1}<0$.

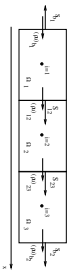


Figure II.7.1: Définition du domaine de calcul unidimensionnel considéré.

On s'intéresse à l'influence sur le respect du principe du maximum discret de la précision avec laquelle est vérifiée sous forme discrète la relation :

$$\forall i\in[1,N],\qquad \int_{\Omega_i}\mathrm{div}\,(\rho\,\underline{u})\;d\Omega=0$$

soit, ici :

$$-\ (\rho\ u)^n_{b_1}\,.\,S_{b_1}\ =\ (\rho\ u)^n_{12}\,.\,S_{12}\ =\ (\rho\ u)^n_{23}\,.\,S_{23}\ =\ (\rho\ u)^n_{b_2}\,.\,S_{b_2} \tag{II.7.43}$$

## Prise en compte de la contribution de $\dfrac{\partial \rho}{\partial t}$ dans la matrice

Le système à résoudre est alors, en omettant pour simplifier l'exposant $(n+1, k+1)$ :

$$\begin{array}{lllllll}
\rho_1^n\ \dfrac{|\Omega_1|}{\Delta t}\,\delta f_1 & & & - & (\rho\ u)^n_{b_1}\,.\,S_{b_1}\,\delta f_1 & = & -(\rho\ u)^n_{b_1}\,.\,S_{b_1}\,f_{b_1} \\
\rho_2^n\ \dfrac{|\Omega_2|}{\Delta t}\,\delta f_2 & + & (\rho\ u)^n_{12}\,.\,S_{12}\,\delta f_2 & - & (\rho\ u)^n_{12}\,.\,S_{12}\,\delta f_1 & = & 0 \\
\rho_3^n\ \dfrac{|\Omega_3|}{\Delta t}\,\delta f_3 & + & (\rho\ u)^n_{23}\,.\,S_{23}\,\delta f_3 & - & (\rho\ u)^n_{23}\,.\,S_{23}\,\delta f_2 & = & 0
\end{array} \tag{II.7.44}$$

ce qui donne :

$$\left\{\begin{array}{lll}
\delta f_1 = & f_{b_1}\ \dfrac{-(\rho\ u)^n_{b_1}\,.\,S_{b_1}}{\rho_1^n\ \dfrac{|\Omega_1|}{\Delta t} - (\rho\ u)^n_{b_1}\,.\,S_{b_1}} \\
\delta f_2 = & +\delta f_1\ \dfrac{(\rho\ u)^n_{12}\,.\,S_{12}}{\rho_2^n\ \dfrac{|\Omega_2|}{\Delta t} + (\rho\ u)^n_{12}\,.\,S_{12}} \\
\delta f_3 = & +\delta f_2\ \dfrac{(\rho\ u)^n_{23}\,.\,S_{23}}{\rho_3^n\ \dfrac{|\Omega_3|}{\Delta t} + (\rho\ u)^n_{23}\,.\,S_{23}}
\end{array}\right. \tag{II.7.45}$$

d'où :

$$\left\{\begin{array}{ll}
\delta f_1 & < 1 \\
\delta f_2 & < 1 \\
\delta f_3 & < 1
\end{array}\right. \tag{II.7.46}$$

On obtient donc bien une solution qui vérifie le principe du maximum discret, même pour des grands pas de temps $\Delta t$, et ce, quelle que soit la précision avec laquelle est vérifiée, à l'étape de correction, la forme discrète (II.7.43) de la conservation de la masse $\displaystyle\int_{\Omega_i} \mathrm{div}\,(\rho\,\underline{u})\ d\Omega = 0$ dont on ne s'est pas servi ici.

## Sans la contribution de $\dfrac{\partial \rho}{\partial t}$ dans la matrice

On obtient de même :

$$\begin{array}{lllllll}
\rho_1^n\ \dfrac{|\Omega_1|}{\Delta t}\,\delta f_1 & + & (\rho\ u)^n_{12}\,.\,S_{12}\,\delta f_1 & & & = & -(\rho\ u)^n_{b_1}\,.\,S_{b_1}\,f_{b_1} \\
\rho_2^n\ \dfrac{|\Omega_2|}{\Delta t}\,\delta f_2 & + & (\rho\ u)^n_{23}\,.\,S_{23}\,\delta f_2 & - & (\rho\ u)^n_{12}\,.\,S_{12}\,\delta f_1 & = & 0 \\
\rho_3^n\ \dfrac{|\Omega_3|}{\Delta t}\,\delta f_3 & - & (\rho\ u)^n_{23}\,.\,S_{23}\,\delta f_2 & + & (\rho\ u)^n_{b_2}\,.\,S_{b_2}\,\delta f_3 & = & 0
\end{array} \tag{II.7.47}$$

soit :

$$\left\{\begin{array}{lll}
\delta f_1 = & f_{b_1}\ \dfrac{-(\rho\ u)^n_{b_1}\,.\,S_{b_1}}{\rho_1^n\ \dfrac{|\Omega_1|}{\Delta t} + (\rho\ u)^n_{12}\,.\,S_{12}} \\
\delta f_2 = & \delta f_1\ \dfrac{(\rho\ u)^n_{12}\,.\,S_{12}}{\rho_2^n\ \dfrac{|\Omega_2|}{\Delta t} + (\rho\ u)^n_{23}\,.\,S_{23}} \\
\delta f_3 = & \delta f_2\ \dfrac{(\rho\ u)^n_{23}\,.\,S_{23}}{\rho_3^n\ \dfrac{|\Omega_3|}{\Delta t} + (\rho\ u)^n_{b_2}\,.\,S_{b_2}}
\end{array}\right. \tag{II.7.48}$$

Ici, on constate que le respect du principe du maximum discret :

$$\left\{\begin{array}{lcl} \delta f_1 & \leqslant & 1 \\ \delta f_2 & \leqslant & 1 \\ \delta f_3 & \leqslant & 1 \end{array}\right. \tag{II.7.49}$$

est équivalent à la condition :

$$\left\{\begin{array}{rcl} -\ (\rho\ u)^n_{b_1}\,.\,S_{b_1} & \leqslant & \rho_1^n\,\dfrac{|\Omega_1|}{\Delta t} + (\rho\ u)^n_{12}\,.\,S_{12} \\ (\rho\ u)^n_{12}\,.\,S_{12} & \leqslant & \rho_2^n\,\dfrac{|\Omega_2|}{\Delta t} + (\rho\ u)^n_{23}\,.\,S_{23} \\ (\rho\ u)^n_{23}\,.\,S_{23} & \leqslant & \rho_3^n\,\dfrac{|\Omega_3|}{\Delta t} + (\rho\ u)^n_{b_2}\,.\,S_{b_2} \end{array}\right. \tag{II.7.50}$$

Contrairement à la situation du paragraphe 7.6.3, on ne peut obtenir ici un résultat qu'en faisant intervenir l'égalité (II.7.43), forme discrète de la conservation de la masse. On obtient bien alors, à partir du système précédent :

$$\left\{\begin{array}{lcl} \delta f_1 & < & 1 \\ \delta f_2 & < & 1 \\ \delta f_3 & < & 1 \end{array}\right. \tag{II.7.51}$$

Si l'on s'intéresse à la cellule $\Omega_1$ et que l'on suppose $(\rho\ u)^n_{12}\,.\,S_{12} = -\,(\rho\ u)^n_{b_1}\,.\,S_{b_1} - \varepsilon(\rho\ u)^n_{12}\,.\,S_{12}$ (où $\varepsilon$ est la précision locale relative pour l'équation de conservation de la masse discrète), on constate que l'on obtient $\delta\, f_1 > f_{b_1} = 1$ (valeur non admissible) dès lors que :

$$\frac{1}{\varepsilon} < \frac{(\rho\ u)^n_{12}\,.\,S_{12}\Delta t}{\rho_1|\Omega_1|}$$

c'est-à-dire dès que le nombre de CFL local $\dfrac{(\rho\ u)^n_{12}\,.\,S_{12}\Delta t}{\rho_1|\Omega_1|}$ excède l'inverse de la précision relative locale $\varepsilon$.

### 7.6.4 Conclusion

Prendre en compte la contribution de $\dfrac{\partial\rho}{\partial t}$ dans la matrice permet un meilleur respect du principe du maximum discret, lorsque la précision de $\displaystyle\int_{\Omega_i} \mathrm{div}\,(\rho\,\underline{u})\ d\Omega = 0$ n'est pas exactement vérifiée.

# 8- Sous-programme gradmc

## 8.1 Fonction

Le but de ce sous-programme est de calculer, au centre des cellules, le gradient d'une fonction scalaire, également connue au centre des cellules. Pour obtenir la valeur de toutes les composantes du gradient, une méthode de minimisation par moindres carrés est mise en œuvre : elle utilise l'estimation d'une composante du gradient aux faces, obtenue à partir des valeurs de la fonction au centre des cellules voisines. Cette méthode est activée lorsque l'indicateur IMRGRA vaut 1 et on l'utilise alors pour le calcul des gradients de toutes les grandeurs. Elle est beaucoup plus rapide que la méthode utilisée par défaut (`IMRGRA`=0), mais présente l'inconvénient d'être moins robuste sur des maillages non orthogonaux, le gradient produit étant moins régulier.

## 8.2 Discrétisation

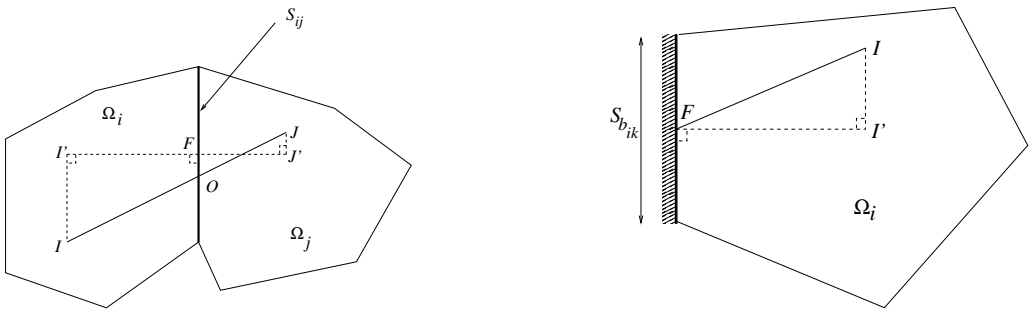



Figure II.8.1: Définition des différentes entités géométriques pour les faces internes (gauche) et de bord (droite).

On se reportera aux notations de la figure II.8.1. On cherche à calculer $\underline{G}_{c,i}$, gradient au centre de la cellule $i$ de la fonction scalaire $P$. Soit $\underline{G}_{f,ij} \cdot \underline{d}_{ij}$ une estimation à la face $ij$ (dont les voisins sont les cellules $i$ et $j$) du gradient projeté dans la direction du vecteur $\underline{d}_{ij}$ (à préciser). De même, on note $\underline{G}_{fb,ik} \cdot \underline{d}_{b,ik}$ une estimation à la face de bord $ik$ ($k^{\text{ième}}$ face de bord appuyée sur la cellule $i$) du gradient projeté dans la direction du vecteur $\underline{d}_{b,ik}$ (à préciser). L'idéal serait de pouvoir trouver un vecteur $\underline{G}_{c,i}$ tel que, pour toute face interne $ij$ ($j \in Vois(i)$) et toute face de bord $ik$ ($k \in \gamma_b(i)$), on ait :

$$\left\{ \begin{array}{l} \underline{G}_{c,i} \cdot \underline{d}_{ij} = \underline{G}_{f,ij} \cdot \underline{d}_{ij} \\ \underline{G}_{c,i} \cdot \underline{d}_{b,ik} = \underline{G}_{f,b,ik} \cdot \underline{d}_{b,ik} \end{array} \right. \tag{II.8.1}$$

Comme il est généralement impossible d'obtenir l'égalité, on cherche à minimiser la fonctionnelle $\mathcal{F}_i$ suivante :

$$\mathcal{F}_i(\underline{G}_{c,i}, \underline{G}_{c,i}) = \frac{1}{2} \sum_{j \in Vois(i)} \left[ \underline{G}_{c,i} \cdot \underline{d}_{ij} - \underline{G}_{f,ij} \cdot \underline{d}_{ij} \right]^2 + \frac{1}{2} \sum_{k \in \gamma_b(i)} \left[ \underline{G}_{c,i} \cdot \underline{d}_{b,ik} - \underline{G}_{f,b,ik} \cdot \underline{d}_{b,ik} \right]^2 \tag{II.8.2}$$

Pour ce faire, on annule la dérivée de $\mathcal{F}_i(\underline{G}_{c,i}, \underline{G}_{c,i})$ par rapport à chacune des trois composantes $(G_{c,i,x}, G_{c,i,y}, G_{c,i,z})$ du vecteur inconnu $\underline{G}_{c,i}$ et l'on résout le système qui en résulte.

Pour pouvoir inverser le système localement et donc à faible coût, on cherche à éviter les dépendances de $\underline{G}_{f,ij} \cdot \underline{d}_{ij}$ et de $\underline{G}_{f,b,ik} \cdot \underline{d}_{b,ik}$ au gradient $\underline{G}_{c,j}$ (gradient pris dans les cellules voisines). Un choix

particulier du vecteur $\underline{d}$ permet d'atteindre ce but :

$$\underline{d}_{ij} = \frac{\underline{IJ}}{||\underline{IJ}||} \quad \text{et} \quad \underline{d}_{b,ik} = \frac{(\underline{I'F})_l}{||\underline{I'F}||} = \underline{n}_{b,ik} \tag{II.8.3}$$

Ainsi, pour les faces internes, le vecteur $\underline{d}$ est le vecteur normé joignant le centre des cellules voisines. La quantité $\underline{G}_{f,ij} \cdot \underline{d}_{ij}$ est reliée directement aux valeurs de la variable $P$ prises au centre des cellules, sans faire intervenir de gradient :

$$\underline{G}_{f,ij} \cdot \underline{d}_{ij} = \frac{P_j - P_i}{||\underline{IJ}||} \tag{II.8.4}$$

Pour les faces de bord, il est possible d'opter pour un choix plus naturel sans pour autant faire intervenir le gradient des cellules voisines : on utilise pour $\underline{d}$ le vecteur normé orthogonal à la face, dirigé vers l'extérieur (le gradient le mieux connu, en particulier au bord, étant le gradient normal aux faces). On a alors :

$$\underline{G}_{f,b,ik} \cdot \underline{d}_{b,ik} = \frac{P_{b,ik} - P_{i'}}{||\underline{I'F}||} \tag{II.8.5}$$

On utilise alors les relations (II.8.6) au bord ($A_{ik}$ et $B_{ik}$ permettent de représenter les conditions aux limites imposées, $P_{b,ik}$ en est issue et représente la valeur à la face de bord) :

$$\left\{ \begin{array}{ll} P_{i'} & = P_i + \underline{II'}.\underline{G}_{c,i} \\ P_{b,ik} & = A_{ik} + B_{ik}\, P_{i'} = A_{ik} + B_{ik}\, (P_i + \underline{II'}.\underline{G}_{c,i}) \end{array} \right. \tag{II.8.6}$$

On obtient finalement :

$$\underline{G}_{f,b,ik} \cdot \underline{d}_{b,ik} = \frac{1}{||\underline{I'F}||} \left[ A_{ik} + (B_{ik} - 1)\, (P_i + \underline{II'}.\underline{G}_{c,i}) \right] \tag{II.8.7}$$

L'équation (II.8.7), qui fait intervenir $\underline{G}_{c,i}$, doit être utilisée pour modifier l'expression (II.8.2) de la fonctionnelle avant de prendre sa différentielle. Ainsi :

$$\begin{array}{rl} \mathcal{F}_i(\underline{G}_{c,i}, \underline{G}_{c,i}) = & \frac{1}{2} \sum_{j \in Vois(i)} \left[ \underline{G}_{c,i} \cdot \underline{d}_{ij} - \underline{G}_{f,ij} \cdot \underline{d}_{ij} \right]^2 + \\ & \frac{1}{2} \sum_{k \in \gamma_b(i)} \left[ \underline{G}_{c,i} \cdot (\underline{d}_{b,ik} - \frac{B_{ik} - 1}{||\underline{I'F}||}\, \underline{II'}) - \frac{1}{||\underline{I'F}||} \left( A_{ik} + (B_{ik} - 1)\, P_i \right) \right]^2 \end{array} \tag{II.8.8}$$

On annule alors la dérivée de $\mathcal{F}_i(\underline{G}_{c,i}, \underline{G}_{c,i})$ par rapport à chacune des trois composantes $(G_{c,i,x}, G_{c,i,y}, G_{c,i,z})$ du vecteur inconnu $\underline{G}_{c,i}$. On obtient, pour chaque cellule $i$, le système $3 \times 3$ local (II.8.9) :

$$\underbrace{\begin{bmatrix} C_{i,x\,x} & C_{i,x\,y} & C_{i,x\,z} \\ C_{i,y\,x} & C_{i,y\,y} & C_{i,y\,z} \\ C_{i,z\,x} & C_{i,z\,y} & C_{i,z\,z} \end{bmatrix}}_{\underline{C}_i} \underbrace{\begin{bmatrix} G_{c,i,x} \\ G_{c,i,y} \\ G_{c,i,z} \end{bmatrix}}_{\underline{G}_{c,i}} = \underbrace{\begin{bmatrix} T_{i,x} \\ T_{i,y} \\ T_{i,z} \end{bmatrix}}_{\underline{T}_i} \tag{II.8.9}$$

avec

$$\left\{ \begin{array}{ll} C_{i,l\,m} & = \sum_{j \in Vois(i)} (\underline{d}_{ij})_l (\underline{d}_{ij})_m + \sum_{k \in \gamma_b(i)} \left( \underline{d}_{b,ik} - \frac{B_{ik} - 1}{||\underline{I'F}||}\, \underline{II'} \right)_l \left( \underline{d}_{b,ik} - \frac{B_{ik} - 1}{||\underline{I'F}||}\, \underline{II'} \right)_m \\ T_{i,l} & = \sum_{j \in Vois(i)} (\underline{G}_{f,ij} \cdot \underline{d}_{ij})(\underline{d}_{ij})_l + \sum_{k \in \gamma_b(i)} \frac{1}{||\underline{I'F}||} \left( A_{ik} + (B_{ik} - 1)\, P_i \right) \left( \underline{d}_{b,ik} - \frac{B_{ik} - 1}{||\underline{I'F}||}\, \underline{II'} \right)_l \end{array} \right. \tag{II.8.10}$$

On obtient finalement :

$$\begin{aligned} C_{i,l\,m} &= \sum_{j\in Vois(i)} \frac{1}{||\underline{IJ}||^2}(\underline{IJ})_l(\underline{IJ})_m + \sum_{k\in\gamma_b(i)} \left(\underline{n}_{b,ik} + \frac{1-B_{ik}}{||\underline{I'F}||}\,\underline{II'}\right)_l \left(\underline{n}_{b,ik} + \frac{1-B_{ik}}{||\underline{I'F}||}\,\underline{II'}\right)_m \\ T_{i,l} &= \sum_{j\in Vois(i)} (P_j - P_i)\,\frac{(\underline{IJ})_l}{||\underline{IJ}||^2} + \sum_{k\in\gamma_b(i)} \frac{1}{||\underline{I'F}||}\,(A_{ik} + (B_{ik}-1)\,P_i)\left(\underline{n}_{b,ik} - \frac{B_{ik}-1}{||\underline{I'F}||}\,\underline{II'}\right)_l \end{aligned} \tag{II.8.11}$$

## 8.3 Mise en œuvre

La variable dont il faut calculer le gradient est contenue dans le tableau `PVAR`. Les conditions aux limites associées sont disponibles au travers des tableaux `COEFAP` et `COEFBP` qui représentent respectivement les grandeurs $A$ et $B$ utilisées ci-dessus. Les trois composantes du gradient seront contenues, en sortie du sous-programme, dans les tableaux `DPDX`, `DPDY` et `DPDZ`.

**• Calcul de la matrice**

Les `NCEL` matrices $\underline{\underline{C}}_i$ (matrices $3\times3$) sont stockées dans le tableau `COCG`, (de dimension `NCELET`$\times3\times3$). Ce dernier est initialisé à zéro, puis son remplissage est réalisé dans des boucles sur les faces internes et les faces de bord. Les matrices $\underline{\underline{C}}_i$ étant symétriques, ces boucles ne servent qu'à remplir la partie triangulaire supérieure, le reste étant complété à la fin par symétrie.

Pour éviter de réaliser plusieurs fois les mêmes calculs géométriques, on conserve, en sortie de sous-programme, dans le tableau `COCG`, l'inverse des `NCEL` matrices $\underline{\underline{C}}_i$. De plus, pour les `NCELBR` cellules qui ont au moins une face de bord, on conserve dans tableau `COCGB`, de dimension `NCELBR` $\times\,3\times3$, la contribution aux matrices $\underline{\underline{C}}_i$ des termes purement géométriques. On précise ces points ci-dessous. Notons donc dès à présent qu'il ne faut pas utiliser les tableaux `COCG` et `COCGB` par ailleurs comme tableaux de travail.

**Cellule ne possédant pas de face de bord**

Lorsque, pour une cellule, aucune des faces n'est une face de bord du domaine, l'expression de la matrice $\underline{\underline{C}}_i$ ne fait intervenir que des grandeurs géométriques et elle reste inchangée tant que le maillage n'est pas déformé. Son inverse n'est donc calculé qu'une seule fois, au premier appel de `GRADMC` avec `ICCOCG=1` (l'indicateur `INICOC`, local à `GRADMC`, est positionné à 0 dès lors que ces calculs géométriques ont été réalisés une fois). Le tableau `COCG` est ensuite réutilisé lors des appels ultérieurs au sous-programme `GRADMC`.

**Cellule possédant au moins une face de bord**

Lorsque l'ensemble des faces d'une cellule contient au moins une face de bord du domaine, un terme contributeur aux matrices $\underline{\underline{C}}_i$ est spécifique à la variable dont on cherche à calculer le gradient, au travers du coefficient $B_{ik}$ issu des conditions aux limites. Il s'agit de :

$$\sum_{k\in\gamma_b(i)} \left(\underline{n}_{b,ik} + \frac{1-B_{ik}}{||\underline{I'F}||}\,\underline{II'}\right)_l \left(\underline{n}_{b,ik} + \frac{1-B_{ik}}{||\underline{I'F}||}\,\underline{II'}\right)_m \tag{II.8.12}$$

Au premier appel réalisé avec `ICCOCG=1`, on calcule la contribution des faces internes et on les stocke dans le tableau `COCGB`, qui sera disponible lors des appels ultérieurs. En effet, la contribution des faces internes est de nature purement géométrique et reste donc inchangée tant que le maillage ne subit pas de déformation. Elle s'écrit :

$$\sum_{j\in Vois(i)} \frac{1}{||\underline{IJ}||^2}(\underline{IJ})_l(\underline{IJ})_m$$

À tous les appels réalisés avec `ICCOCG=1`, les termes qui dépendent des faces de bord (II.8.12) sont ensuite calculés et on additionne cette contribution et `COCGB` qui contient celle des faces internes : on obtient ainsi les matrices $\underline{\underline{C}}_i$ dans le tableau `COCG`. Leur inverse se calcule indépendamment pour chaque cellule et on le conserve dans `COCG` qui sera disponible lors des appels ultérieurs.

Lorsque GRADMC a été appelé une fois avec ICCOCG=1, des calculs peuvent être évités en positionnant l'indicateur ICCOCG à 0 (si ICCOCG est positionné à 1, tous les calculs relatifs aux cellules ayant au moins une face de bord sont refaits).

- Si GRADMC est utilisé pour calculer le gradient de la même variable (ou, plus généralement, d'une variable dont les conditions aux limites conduisent aux mêmes valeurs du coefficient $B_{ik}$), les matrices $\underline{\underline{C}}_i$ sont inchangées et leur inverse est disponible dans COCG (on positionne ICCOCG à 0 pour éviter de refaire les calculs).
- Dans le cas contraire, les termes relatifs aux faces de bord (II.8.12) sont recalculés et on additionne cette contribution et COCGB qui fournit celle des faces internes : on obtient ainsi les matrices $\underline{\underline{C}}_i$ dans COCG. Il reste alors à inverser ces matrices.

**Remarque :**
Pour sauvegarder les contributions géométriques dans COCGB, on a recours a une boucle portant sur les NCELBR cellules dont au moins une face est une face de bord du domaine. Le numéro de ces cellules est donné par IEL = ICELBR(II) (II variant de 1 à NCELBR). Les opérations réalisées dans cette boucle sont du type COCGB(II,1,1) = COCG(IEL,1,1). La structure (injective) de ICELBR permet de forcer la vectorisation de la boucle.

**• Inversion de la matrice**
On calcule les coefficients de la comatrice, puis l'inverse. Pour des questions de vectorisation, la boucle sur les NCEL éléments est remplacée par une série de boucles en vectorisation forcée sur des blocs de NBLOC=1024 éléments. Le reliquat (NCEL $-$ $E$(NCEL/1024) $\times$ 1024) est traité après les boucles. La matrice inverse est ensuite stockée dans COCG (toujours en utilisant sa propriété de symétrie).

**• Calcul du second membre et résolution**
Le second membre est stocké dans BX, BY et BZ. Le gradient obtenu par résolution des systèmes locaux est stocké dans DPDX, DPDY et DPDZ.

**• Remarque : gradient sans reconstruction**
(non consistant sur maillage non orthogonal)
Dans le cas où l'utilisateur souhaite ne pas reconstruire le gradient (*i.e.* ne pas inclure les termes de non orthogonalité au calcul du gradient), une méthode spécifique est mise en œuvre, qui n'a pas de rapport avec la méthode de moindres carrés présentée ci-dessus.

Le volume de la cellule $i$ est noté $\Omega_i$. $P_{ij}$ (resp. $P_{b,ik}$) représente la valeur estimée de la variable $P$ à la face interne $ij$ (resp. à la face de bord $ik$) de vecteur normal associé $\underline{S}_{ij}$ (resp. $\underline{S}_{b,ik}$). Le gradient est simplement calculé en utilisant la formule suivante :

$$\underline{G}_{c,i} = \quad = \frac{1}{\Omega_i}\left[\sum_{j\in Vois(i)} P_{ij}\underline{S}_{ij} + \sum_{k\in\gamma_b(i)} P_{b,ik}\underline{S}_{b,ik}\right] \tag{II.8.13}$$

Les valeurs aux faces sont obtenues simplement comme suit (avec $\alpha_{ij} = \dfrac{\overline{FJ'}}{\overline{I'J'}}$) :

$$\begin{cases} P_{ij} & = \alpha_{ij}P_i + (1-\alpha_{ij})P_j \\ P_{b,ik} & = A_{ik} + B_{ik}\,P_i \end{cases} \tag{II.8.14}$$

## 8.4 Points à traiter

**• Vectorisation forcée**
Il est peut-être possible de s'affranchir du découpage en boucles de 1024 si les compilateurs sont

capables d'effectuer la vectorisation sans cette aide. On note cependant que ce découpage en boucles de 1024 n'a pas de coût CPU supplémentaire, et que le coût mémoire associé est négligeable. Le seul inconvénient réside dans la relative complexité de l'écriture.

• **Choix du vecteur** $d$
Le choix $\underline{d}_{ij} = \frac{\overrightarrow{IJ}}{||\overrightarrow{IJ}||}$ permet de calculer simplement une composante du gradient à la face en ne faisant intervenir que les valeurs de la variable au centre des cellules voisines. Le choix $\underline{d}_{ij} = \frac{\overrightarrow{I'J'}}{||\overrightarrow{I'J'}||}$ serait également possible, et peut-être meilleur, mais conduirait naturellement à faire intervenir, pour le calcul de la composante du gradient normale aux faces, les valeurs de la variable aux points $I'$ et $J'$, et donc les valeurs du gradient dans les cellules voisines. Il en résulterait donc un système couplé, auquel un algorithme itératif (voir `GRADRC`) pourrait être appliqué. L'aspect temps calcul, atout majeur de la méthode actuelle, s'en ressentirait sans doute.

• **Amélioration de la méthode**
Cette méthode rencontre des difficultés dans le cas de maillages assez "non orthogonaux" (cas de la voiture maillé en tétraèdres par exemple). Une voie d'amélioration possible est d'utiliser un support étendu (le support est l'ensemble des cellules utilisées pour calculer le gradient en une cellule donnée). Un exemple est fourni sur la figure II.8.2 ci-dessous : si la cellule $I$ est la cellule courante, on choisit pour support les cellules de centre $J$ telles que la droite $(IJ)$ soit la plus orthogonale possible à une face de la cellule $I$.

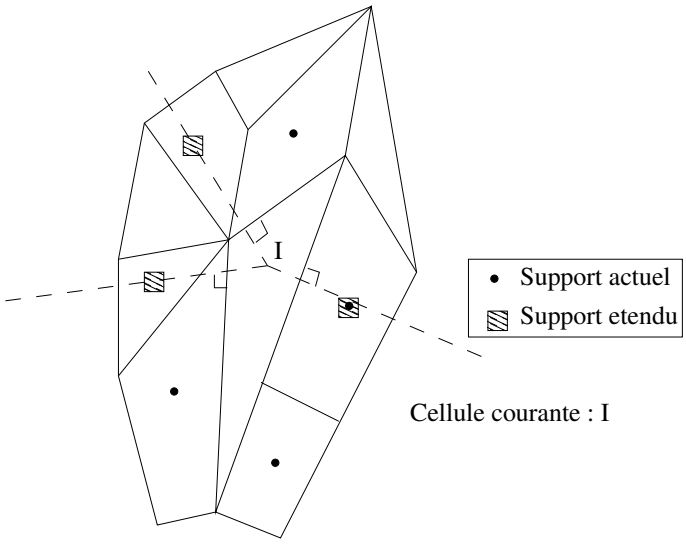



Figure II.8.2: Différents supports pour le calcul du gradient.

# 9- Sous-programme gradrc

## 9.1 Fonction

Le but de ce sous-programme est de calculer, au centre des cellules, le gradient d'une fonction scalaire, également connue au centre des cellules. Pour obtenir la valeur du gradient, une méthode itérative de reconstruction pour les maillages non orthogonaux est mise en œuvre : elle fait appel à un développement limité d'ordre 1 en espace sur la variable, obtenu à partir de la valeur de la fonction et de son gradient au centre de la cellule. Cette méthode, choisie comme option par défaut, correspond à `IMRGRA` = 0 et est utilisée pour le calcul des gradients de toutes les grandeurs. Cette technique est plus robuste mais beaucoup plus lente que la méthode par moindres carrés correspondant à `IMRGRA` = 1.

## 9.2 Discrétisation

### 9.2.1 Méthode générale

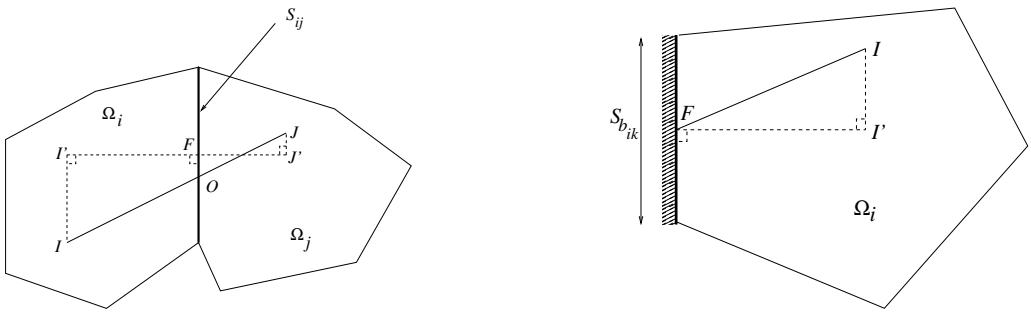



Figure II.9.1: Définition des différentes entités géométriques pour les faces internes (gauche) et de bord (droite).

On se reportera aux notations de la figure II.9.1, qui correspondent à celles employées dans le sous-programme `gradmc`. On cherche à calculer $\underline{G}_{c,i}$, gradient au centre de la cellule $i$ de la fonction scalaire $P$. Le volume de la cellule $i$ est noté $\Omega_i$. Soit $\underline{G}_{f,ij}$ la valeur du gradient à la face $ij$ dont les voisins sont les cellules $i$ et $j$. $P_{ij}$ (resp. $P_{b,ik}$) représente la valeur estimée de la variable $P$ à la face interne $ij$ (resp. à la face de bord $ik$, $k^{\text{ième}}$ face de bord appuyée sur la cellule $i$) de vecteur normal associé $\underline{S}_{ij}$ (resp. $\underline{S}_{b\,ik}$).

Par définition :

$$\begin{aligned} \underline{G}_{c,i} &= (\underline{\text{grad}}\ P)_I \\ \underline{G}_{f,ij} &= (\underline{\text{grad}}\ P)_{f,ij} \end{aligned}$$

On a :

$$\underline{G}_{f,ij} = (\underline{\text{grad}}\ P)_{O_{ij}} = (\underline{\text{grad}}\ P)_{F_{ij}} \text{ (à l'ordre 1 en } P)$$

Afin de prendre en compte les non orthogonalités éventuelles du maillage, on calcule le gradient $\underline{G}_{c,i}$

en effectuant un développement limité d'ordre 1 en espace. On obtient alors :

$$\begin{aligned}|\Omega_i|\,(\underline{\text{grad}}\ P)_I &\overset{déf}{=} \int_{\Omega_i} \underline{\text{grad}}\ P\, d\Omega = \sum_{j\in Vois(i)} P_{ij}\,\underline{S}_{ij} + \sum_{k\in\gamma_b(i)} P_{b_{ik}}\,\underline{S}_{b_{ik}} \\ &= \sum_{j\in Vois(i)} [P_{O_{ij}} + \underline{O_{ij}F_{ij}}\,.\,(\underline{\text{grad}}\ P)_{O_{ij}}]\,\underline{S}_{ij} + \sum_{k\in\gamma_b(i)} (\texttt{INC}\,A_{b,ik} + B_{b,ik}\,P_{I'})\,\underline{S}_{b_{ik}} \\ &= \sum_{j\in Vois(i)} [\,(\alpha_{ij}\,P_I + (1-\alpha_{ij})\,P_J)\,]\,\underline{S}_{ij} + \sum_{j\in Vois(i)} \left[\underline{O_{ij}F_{ij}}\,.\,(\underline{\text{grad}}\ P)_{f,ij}\right]\,\underline{S}_{ij} \\ &+ \sum_{k\in\gamma_b(i)} (\texttt{INC}\,A_{b,ik} + B_{b,ik}\,P_{I'})\,\underline{S}_{b_{ik}}\end{aligned} \tag{II.9.1}$$

La variable `INC` permet d'affecter correctement les conditions aux limites des quantités dont on veut prendre le gradient. En effet,

- `INC` $= 1$ correspond à un calcul de gradient de variable totale et donc à des conditions aux limites standards,
- `INC` $= 0$ correspond à un calcul de variable en incrément et donc à des conditions aux limites associées homogènes.

En faisant une approximation sur $P$ d'ordre 1 en espace à nouveau :

$$\left\{\begin{array}{ll}(\underline{\text{grad}}\ P)_{f,ij} & = \dfrac{1}{2}\,[(\underline{\text{grad}}\ P)_I + (\underline{\text{grad}}\ P)_J] \\ P_{I'} & = P_I + \underline{II'}\,.\,(\underline{\text{grad}}\ P)_I\end{array}\right.$$

d'où :

$$\begin{aligned}|\Omega_i|\,\underline{G}_{c,i} = \sum_{j\in Vois(i)} &\left[(\alpha_{ij}\,P_I + (1-\alpha_{ij})\,P_J) + \frac{1}{2}\,\underline{O_{ij}F_{ij}}\,.\,\left(\underline{G}_{c,i} + \underline{G}_{c,j}\right)\right]\,\underline{S}_{ij} \\ &+ \sum_{k\in\gamma_b(i)} \left[\texttt{INC}\,A_{b,ik} + B_{b,ik}\,P_I + B_{b,ik}\,\underline{II'}\,.\,\underline{G}_{c,i}\right]\,\underline{S}_{b_{ik}}\end{aligned}$$

en notant $\alpha_{ij} = \dfrac{\overline{FJ'}}{\overline{I'J'}}$.

On regroupe à gauche les termes en $\underline{G}_{c,i}$ et on obtient :

$$\begin{aligned}|\Omega_i|\,\underline{G}_{c,i} - \sum_{j\in Vois(i)} \frac{1}{2}\,(\underline{O_{ij}F_{ij}}\,.\,\underline{G}_{c,i})\underline{S}_{ij} - \sum_{k\in\gamma_b(i)} B_{b,ik}\,(\underline{II'}\,.\,\underline{G}_{c,i})\,\underline{S}_{b_{ik}} = \\ \sum_{j\in Vois(i)} [(\alpha_{ij}\,P_I + (1-\alpha_{ij})\,P_J)]\,\underline{S}_{ij} + \sum_{j\in Vois(i)} \frac{1}{2}\,(\underline{O_{ij}F_{ij}}\,.\,\underline{G}_{c,j})\underline{S}_{ij} \\ + \sum_{k\in\gamma_b(i)} (\texttt{INC}\,A_{b,ik} + B_{b,ik}\,P_I)\,\underline{S}_{b_{ik}}\end{aligned} \tag{II.9.2}$$

ce qui donne pour la direction $\eta$ ($\eta$, $\beta = x$, $y$ ou $z$) :

$$\begin{aligned}|\Omega_i|\,G_{c,i,\eta} - \sum_{j\in Vois(i)} \frac{1}{2}\left(\sum_\beta (\underline{O_{ij}F_{ij}})_{,\beta}\,G_{c,i,\beta}\right) S_{ij,\eta} - \sum_{k\in\gamma_b(i)} B_{b,ik}\left(\sum_\beta (\underline{II'})_{,\beta}\,G_{c,i,\beta}\right) S_{b_{ik},\eta} = \\ \sum_{j\in Vois(i)} [(\alpha_{ij}\,P_I + (1-\alpha_{ij})\,P_J)]\,S_{ij,\eta} + \sum_{j\in Vois(i)} \frac{1}{2}\left(\sum_\beta (\underline{O_{ij}F_{ij}})_{,\beta}\,G_{c,j,\beta}\right) S_{ij,\eta} \\ + \sum_{k\in\gamma_b(i)} (\texttt{INC}\,A_{b,ik} + B_{b,ik}\,P_I)\,S_{b_{ik},\eta}\end{aligned} \tag{II.9.3}$$

### 9.2.2 Cas sans reconstruction des non orthogonalités

Lorsque le maillage est orthogonal ou lorsqu'on ne veut pas reconstruire, seules les contributions d'ordre 0 au centre des cellules interviennent dans le calcul du gradient ($\underline{II'} = \underline{0}$, $\underline{OF} = \underline{0}$) :

$$\begin{aligned} |\Omega_i|\,\underline{G}_{c,i} &\overset{déf}{=} \int_{\Omega_i} \underline{\text{grad}}\; P\, d\Omega = \sum_{j\in Vois(i)} P_{ij}\,\underline{S}_{ij} + \sum_{k\in\gamma_b(i)} P_{b,ik}\,\underline{S}_{b_{ik}} \\ &= \sum_{j\in Vois(i)} \left[\,(\alpha_{ij}\,P_I + (1-\alpha_{ij})\,P_J)\,\right]\,\underline{S}_{ij} + \sum_{k\in\gamma_b(i)} (\texttt{INC}\,A_{b,ik} + B_{b,ik}\,P_I)\,\underline{S}_{b_{ik}} \end{aligned}$$

d'où :

$$\underline{G}_{c,i} \;=\; \frac{1}{|\Omega_i|}\cdot\left[\sum_{j\in Vois(i)} \left[\alpha_{ij}\,P_I + (1-\alpha_{ij})\,P_J)\,\right]\,\underline{S}_{ij} \;+\; \sum_{k\in\gamma_b(i)} (\texttt{INC}\,A_{b,ik} + B_{b,ik}\,P_I)\,\underline{S}_{b_{ik}}\right] \qquad \text{(II.9.4)}$$

Remarque
Le gradient non reconstruit $\underline{G}_{c,i}^{NRec}$ se calcule donc très facilement et directement *via* l'équation (II.9.4). Il n'est cependant pas consistant sur maillage non orthogonal.

### 9.2.3 Reconstruction

#### Méthode de résolution

Afin de pouvoir résoudre le système (II.9.2), on va impliciter les termes en $\underline{G}_{c,i}$, expliciter ceux en $\underline{G}_{c,j}$ et raisonner de facon itérative en introduisant une suite d'incréments $(\delta\,\underline{G}_i^k)_{k\in\mathbb{N}}$ définie par :

$$\left\{\begin{array}{l} \delta\,\underline{G}_i^0 = \underline{G}_{c,i}^{NRec} \\ \delta\,\underline{G}_i^{k+1} = \underline{G}_{c,i}^{k+1} - \underline{G}_{c,i}^{k} \end{array}\right. \qquad \text{(II.9.5)}$$

et de système associé, dans la direction $\eta$ :

$$\begin{aligned} &|\Omega_i|\,G_{c,i,\eta}^{k+1} - \sum_{j\in Vois(i)} \frac{1}{2}\left(\sum_{\beta} (\underline{O_{ij}F_{ij}})_{,\beta}\,G_{c,i,\beta}^{k+1}\right) S_{ij,\eta} - \sum_{k\in\gamma_b(i)} B_{b,ik}\left(\sum_{\beta} (\underline{II'})_{,\beta}\,G_{c,i,\beta}^{k+1}\right) S_{b_{ik},\eta} \\ &\quad = \sum_{j\in Vois(i)} \left[(\alpha_{ij}\,P_I + (1-\alpha_{ij})\,P_J)\right]\,S_{ij,\eta} + \sum_{j\in Vois(i)} \frac{1}{2}\left(\sum_{\beta} (\underline{O_{ij}F_{ij}})_{,\beta}\,G_{c,j,\beta}^{k}\right) S_{ij,\eta} \\ &\qquad\qquad + \sum_{k\in\gamma_b(i)} (\texttt{INC}\,A_{b,ik} + B_{b,ik}\,P_I)\,S_{b_{ik},\eta} \qquad \text{(II.9.6)} \end{aligned}$$

ou, comme :

$$\underline{G}_{c,i}^{k+1} = \underline{G}_{c,i}^{k} + \delta\,\underline{G}_i^{k+1}$$

$$
\begin{aligned}
|\Omega_i|\,\delta G_{i,\eta}^{k+1} - \sum_{j\in Vois(i)} \frac{1}{2}\left(\sum_{\beta}(\underline{O_{ij}F_{ij}})_{,\beta}\,\delta G_{c,i,\beta}^{k+1}\right) S_{ij,\eta} - \sum_{k\in\gamma_b(i)} B_{b,ik}\left(\sum_{\beta}(\underline{II'})_{,\beta}\,\delta G_{c,i,\beta}^{k+1}\right) S_{b_{ik},\eta} \\
= -|\Omega_i|\,G_{c,i,\eta}^{k} + \sum_{j\in Vois(i)} \left[(\alpha_{ij}\,P_I + (1-\alpha_{ij})\,P_J)\right] S_{ij,\eta} \\
+ \sum_{j\in Vois(i)} \left(\sum_{\beta}(\underline{O_{ij}F_{ij}})_{,\beta}\,\frac{1}{2}\,(G_{c,i,\beta}^{k} + G_{c,j,\beta}^{k})\right) S_{ij,\eta} \\
+ \sum_{k\in\gamma_b(i)} (\mathtt{INC}\,A_{b,ik} + B_{b,ik}\,(P_I + \left(\sum_{\beta}(\underline{II'})_{,\beta}\,G_{c,i,\beta}^{k}\right))\,S_{b_{ik},\eta} \qquad \text{(II.9.7)}
\end{aligned}
$$

L'équation (II.9.7) conduit à un système matriciel local par rapport à chacune des trois composantes $(\delta G_{i,x}^{k+1}, \delta G_{i,y}^{k+1}, \delta G_{i,z}^{k+1})$ du vecteur inconnu $\delta \underline{G}_i^{k+1}$. On obtient donc, pour chaque cellule $i$, le système $3\times 3$ suivant :

$$
\underbrace{\begin{bmatrix} C_{i,x\,x} & C_{i,x\,y} & C_{i,x\,z} \\ C_{i,y\,x} & C_{i,y\,y} & C_{i,y\,z} \\ C_{i,z\,x} & C_{i,z\,y} & C_{i,z\,z} \end{bmatrix}}_{\underline{\underline{C}}_i}
\underbrace{\begin{bmatrix} \delta G_{i,x}^{k+1} \\ \delta G_{i,y}^{k+1} \\ \delta G_{i,z}^{k+1} \end{bmatrix}}_{\delta \underline{G}_i^{k+1}}
= \underbrace{\begin{bmatrix} R_{i,x}^{k+1} \\ R_{i,y}^{k+1} \\ R_{i,z}^{k+1} \end{bmatrix}}_{\underline{R}_i^{k+1}} \qquad \text{(II.9.8)}
$$

avec, ($\eta$, $\beta =$ $x$, $y$ ou $z$) :

$$
\left\{\begin{array}{ll}
C_{i,\eta\,\eta} & = |\Omega_i| - \dfrac{1}{2}\displaystyle\sum_{j\in Vois(i)} (\underline{O_{ij}F_{ij}})_{,\eta}\,S_{ij,\eta} - \sum_{k\in\gamma_b(i)} B_{b,ik}\,(\underline{II'})_{,\eta}\,S_{b_{ik},\eta} \\
C_{i,\eta\,\beta} & = -\dfrac{1}{2}\displaystyle\sum_{j\in Vois(i)} (\underline{O_{ij}F_{ij}})_{,\beta}\,S_{ij,\eta} - \sum_{k\in\gamma_b(i)} B_{b,ik}\,(\underline{II'})_{,\beta}\,S_{b_{ik},\eta} \qquad \text{pour } \eta\neq\beta \\
R_{i,\eta}^{k+1} & = -|\Omega_i|\,G_{c,i,\eta}^{k} + \displaystyle\sum_{j\in Vois(i)} \left[(\alpha_{ij}\,P_I + (1-\alpha_{ij})\,P_J)\right] S_{ij,\eta} \\
& \quad + \displaystyle\sum_{j\in Vois(i)} \left(\sum_{\beta}(\underline{O_{ij}F_{ij}})_{,\beta}\,\frac{1}{2}\,(G_{c,i,\beta}^{k} + G_{c,j,\beta}^{k})\right) S_{ij,\eta} \\
& \quad + \displaystyle\sum_{k\in\gamma_b(i)} (\mathtt{INC}\,A_{b,ik} + B_{b,ik}\left(P_I + \left(\sum_{\beta}(\underline{II'})_{,\beta}\,G_{c,i,\beta}^{k}\right)\right)\,S_{b_{ik},\eta}
\end{array}\right. \qquad \text{(II.9.9)}
$$

L'inversion de la matrice $\underline{\underline{C}}_i$ permet de calculer $(\delta \underline{G}_i^{k+1})$ et donc $(\underline{G}_i^{k+1})$. Les itérations s'arrêtent lorsque la norme euclidienne du second membre $\underline{R}_i^{k+1}$ tend vers zéro (*i.e.* lorsque la norme euclidienne de $(\delta \underline{G}_i^{k})$ tend vers zéro) ou lorsque le nombre d'itérations en $k$ fixé maximal est atteint.

Remarque 3

Pour les conditions aux limites en pression, un traitement particulier est mis en œuvre, surtout utile dans les cas où un gradient de pression (hydrostatique ou autre) nécessite une attention spécifique aux bords, où une condition à la limite de type Neumann homogène est généralement inadaptée. Soit $P_{F_{b\,ik}}$ la valeur de la pression à la face associée, que l'on veut calculer.

On note que :

$$
\underline{I'F}_{b\,ik}\,.\,(\underline{\text{grad}}\ P)_I = \underline{I'F}_{b\,ik}\,.\,\underline{G}_{c,i} = \overline{I'F}_{b\,ik}\,.\,\left.\frac{\delta P}{\delta n}\right|_{F_{b\,ik}}
$$

avec les conventions précédentes.

**Sur maillage orthogonal** On se place dans le cas d'un maillage orthogonal , *i.e.* pour toute cellule $\Omega_I$, $I$ et son projeté $I'$ sont identiques. Soit $M_{b\,ik}$ le milieu du segment $IF_{b\,ik}$.
On peut écrire les égalités suivantes :

$$\begin{aligned} P_{F_{b\,ik}} &= P_{M_{b\,ik}} + \overline{M_{b\,ik}F_{b\,ik}} \cdot \left.\frac{\delta P}{\delta n}\right|_{M_{b\,ik}} + \overline{M_{b\,ik}F_{b\,ik}}^2 \cdot \frac{1}{2}\left.\frac{\delta^2 P}{\delta n^2}\right|_{M_{b\,ik}} + \mathcal{O}(h^3) \\ P_I &= P_{M_{b\,ik}} + \overline{M_{b\,ik}I} \cdot \left.\frac{\delta P}{\delta n}\right|_{M_{b\,ik}} + \overline{M_{b\,ik}I}^2 \cdot \frac{1}{2}\left.\frac{\delta^2 P}{\delta n^2}\right|_{M_{b\,ik}} + \mathcal{O}(h^3) \end{aligned}$$

avec $\overline{M_{b\,ik}I} = -\overline{M_{b\,ik}F_{b\,ik}}$.
On obtient donc :

$$P_{F_{b\,ik}} - P_I = \overline{IF}_{b\,ik} \cdot \left.\frac{\delta P}{\delta n}\right|_{M_{b\,ik}} + \mathcal{O}(h^3) \tag{II.9.10}$$

Grâce à la formule des accroissements finis :

$$\left.\frac{\delta P}{\delta n}\right|_{M_{b\,ik}} = \frac{1}{2}\left[\left.\frac{\delta P}{\delta n}\right|_I + \left.\frac{\delta P}{\delta n}\right|_{F_{b\,ik}}\right] + \mathcal{O}(h^2) \tag{II.9.11}$$

On s'intéresse aux cas suivants :

- condition à la limite de type Dirichlet

$P_{F_{b\,ik}} = P_{IMPOSE}$, aucun traitement particulier

- condition à la limite de type Neumann homogène standard

On veut imposer :

$$\left.\frac{\delta P}{\delta n}\right|_{F_{b\,ik}} = 0 + \mathcal{O}(h) \tag{II.9.12}$$

On a :

$$\left.\frac{\delta P}{\delta n}\right|_I = \left.\frac{\delta P}{\delta n}\right|_{F_{b\,ik}} + \mathcal{O}(h)$$

et comme :

$$P_{F_{b\,ik}} = P_I + \overline{IF}_{b\,ik} \cdot \left.\frac{\delta P}{\delta n}\right|_I + \mathcal{O}(h^2) \tag{II.9.13}$$

on en déduit :

$$P_{F_{b\,ik}} = P_I + \overline{IF}_{b\,ik} \cdot \left.\frac{\delta P}{\delta n}\right|_{F_{b\,ik}} + \mathcal{O}(h^2) \tag{II.9.14}$$

soit :

$$P_{F_{b\,ik}} = P_I + \mathcal{O}(h^2) \tag{II.9.15}$$

On obtient donc une approximation d'ordre 1.

- condition à la limite de type Neumann homogène améliorée

Des égalités (II.9.10, II.9.11), on tire :

$$P_{F_{b\,ik}} = P_I + \frac{1}{2}\overline{IF}_{b\,ik} \cdot \left.\frac{\delta P}{\delta n}\right|_I + \mathcal{O}(h^3)$$

On obtient donc une approximation d'ordre 2.

- condition à la limite de type extrapolation du gradient $\left.\frac{\delta P}{\delta n}\right|_{F_{b\,ik}} = \left.\frac{\delta P}{\delta n}\right|_I$

Des deux égalités (II.9.10, II.9.11), on déduit :

$$P_{F_{b\,ik}} = P_I + \overline{IF}_{b\,ik} \cdot \left.\frac{\delta P}{\delta n}\right|_I + \mathcal{O}(h^3) \tag{II.9.16}$$

On obtient donc également une approximation d'ordre 2.

**Sur maillage non orthogonal** Dans ce cas, on peut seulement écrire :

$$P_{F_{b\,ik}} = P_{I'} + \frac{1}{2}\,\underline{I'F}_{b\,ik} \cdot [\,(\underline{\text{grad}}\ P)_{I'} + (\underline{\text{grad}}\ P)_{F_{b\,ik}}\,] + \mathcal{O}(h^3) \tag{II.9.17}$$

- condition à la limite de type Dirichlet

$P_{F_{b\,ik}} = P_{IMPOSE}$, toujours aucun traitement particulier

- condition à la limite de type Neumann homogène standard

On veut :

$$\left.\frac{\delta P}{\delta n}\right|_{F_{b\,ik}} = 0 + \mathcal{O}(h) \tag{II.9.18}$$

ce qui entraîne :

$$\underline{I'F}_{b\,ik} \cdot (\underline{\text{grad}}\ P)_{F_{b\,ik}} = \mathcal{O}(h^2) \tag{II.9.19}$$

On peut écrire :

$$(\underline{\text{grad}}\ P)_{I'} = (\underline{\text{grad}}\ P)_{F_{b\,ik}} + \mathcal{O}(h)$$

d'où :

$$P_{F_{b\,ik}} = P_{I'} + \mathcal{O}(h^2) \tag{II.9.20}$$

On obtient donc une approximation d'ordre 1.

- condition à la limite de type Neumann homogène améliorée

Le gradient n'est connu qu'au centre des cellules $I$ et non aux points $I'$.

$$(\underline{\text{grad}}\ P)_{I'} = (\underline{\text{grad}}\ P)_{I} + \mathcal{O}(h)$$

d'où :

$$\begin{aligned} P_{F_{b\,ik}} &= P_{I'} + \frac{1}{2}\,\underline{I'F}_{b\,ik} \cdot [\,(\underline{\text{grad}}\ P)_{I'} + (\underline{\text{grad}}\ P)_{F_{b\,ik}}\,] + \mathcal{O}(h^3) \\ &= P_{I'} + \frac{1}{2}\,\underline{I'F}_{b\,ik} \cdot [\,(\underline{\text{grad}}\ P)_{I} + (\underline{\text{grad}}\ P)_{F_{b\,ik}}\,] + \mathcal{O}(h^2) \end{aligned} \tag{II.9.21}$$

Compte-tenu de la condition imposée et de l'équation (*II*.9.19), seule la contribution en $I$ reste.

$$P_{F_{b\,ik}} = P_{I'} + \frac{1}{2}\,\underline{I'F}_{b\,ik} \cdot (\underline{\text{grad}}\ P)_{I} + \mathcal{O}(h^2) \tag{II.9.22}$$

On obtient donc une approximation d'ordre 1.

- condition à la limite de type extrapolation du gradient $\left.\frac{\delta P}{\delta n}\right|_{F_{b\,ik}} = \left.\frac{\delta P}{\delta n}\right|_{I}$

En tenant compte de cette égalité, l'expression de $P_{F_{b\,ik}}$ devient :

$$P_{F_{b\,ik}} = P_{I'} + \underline{I'F}_{b\,ik} \cdot (\underline{\text{grad}}\ P)_{I} + \mathcal{O}(h^2) \tag{II.9.23}$$

On obtient également une approximation d'ordre 1.

- **Conclusion**

On peut récapituler toutes ces situations *via* la formule :

$$P_{F_{b\,ik}} = P_{I'} + \texttt{EXTRAP}\ (\underline{I'F}_{b\,ik} \cdot (\underline{\text{grad}}\ P)_{I})$$

avec `EXTRAP` valant 0, 0.5 ou 1.

Il ne faut en aucun cas utiliser `EXTRAP` avec des conditions de type Neumann non homogènes $\left.\frac{\delta P}{\delta n}\right|_F = g_{bord}$ , $g_{bord}$ donnée non nulle ou de type Robin (mixtes) plus généralement $aP_{F_{b\,ik}} + b \left.\frac{\delta P}{\delta n}\right|_{F_{b\,ik}} = g_{bord}$, le recours à `EXTRAP` n'ayant plus aucun sens.

## 9.3 Mise en œuvre

La variable dont il faut calculer le gradient est contenue dans le tableau `PVAR`. Les conditions aux limites associées sont disponibles au travers des tableaux `COEFAP` et `COEFBP` qui représentent respectivement les grandeurs $A$ et $B$ utilisées ci-dessus. Les trois composantes du gradient sont contenues, en sortie du sous-programme, dans les tableaux `DPDX`, `DPDY` et `DPDZ`.

• **Initialisations**
Le tableau (`BX`, `BY`, `BZ`) du second membre $\underline{R}_i$ est initialisé à zéro.
Le calcul du gradient cellule non reconstruit $\underline{G}_{c,i}^{NRec}$ est réalisé et stocké dans les tableaux `DPDX`, `DPDY` et `DPDZ`. Si aucune reconstruction n'est à faire, on a fini.

### 9.3.1 Reconstruction

Sinon, on cherche à résoudre le système (II.9.7) en incréments de gradient $\delta \underline{G}_i^{k+1}$. Le gradient non reconstruit constitue alors une première estimation du gradient à calculer par incréments.
On effectue les opérations suivantes :

#### Phase préliminaire

**Calcul de la matrice, hors boucle en $k$**
Les `NCEL` matrices $\underline{\underline{C}}_i$ (matrices non symétriques $3\times3$) sont stockées dans le tableau `COCG`, (de dimension `NCELET` $\times 3 \times 3$). Ce dernier est initialisé à zéro, puis son remplissage est réalisé dans des boucles sur les faces internes et les faces de bord. Pour éviter de réaliser plusieurs fois les mêmes calculs géométriques, on conserve, en sortie de sous-programme, dans le tableau `COCG`, l'inverse des `NCEL` matrices $\underline{\underline{C}}_i$.

**Cellule ne possédant pas de face de bord**
Lorsque, pour une cellule, aucune des faces n'est une face de bord du domaine, l'expression de la matrice $\underline{\underline{C}}_i$ ne fait intervenir que des grandeurs géométriques. Son inverse peut être donc calculé une seule fois, stocké dans `COCG` et réutilisé si l'on rappelle `gradrc` séquentiellement et si on est sur un maillage fixe (indicateur `ICCOCG` positionné à 0).

**Cellule possédant au moins une face de bord**
Lorsque l'ensemble des faces d'une cellule contient au moins une face de bord du domaine, un terme contributeur aux matrices $\underline{\underline{C}}_i$ est spécifique à la variable dont on cherche à calculer le gradient, au travers du coefficient $B_{b,ik}$ issu des conditions aux limites. Il s'agit de :

$$- \sum_{k \in \gamma_b(i)} B_{b,ik} \left(\underline{II'}\right)_\beta S_{b_{ik},\eta}$$

Si, lors de l'appel précédent[1] à `gradrc`, les conditions aux limites relatives à la variable $P$ traitée conduisaient à des valeurs identiques de $B_{b,ik}$, les matrices $\underline{\underline{C}}_i$ sont donc inchangées et l'inverse est encore disponible dans `COCG`. Pour éviter de refaire les calculs associés, l'indicateur `ICCOCG`, passé en argument, est alors positionné à 0.

[1] donc, à partir du second appel au moins

Si, au contraire, les valeurs de $B_{b,ik}$ sont différentes de celles de l'appel précédent, il est alors indispensable de recalculer le terme et l'indicateur `ICCOCG` doit être positionné à 1.

Toutefois compte-tenu du coût total de l'inversion de ces matrices relativement au coût global du sous-programme, cette démarche de stockage et donc d'économie de temps C.P.U. est un peu superflue et risque d'engendrer des erreurs (indicateur `ICCOCG` positionné à 0 au lieu de 1) beaucoup plus pénalisantes que l'éventuel gain escompté.

**Inversion de la matrice**

On calcule les coefficients de la comatrice, puis l'inverse. Pour des questions de vectorisation, la boucle sur les `NCEL` éléments est remplacée par une série de boucles en vectorisation forcée sur des blocs de `NBLOC=1024` éléments. Le reliquat ($\texttt{NCEL} - E(\texttt{NCEL}/1024) \times 1024$) est traité après les boucles. La matrice inverse est ensuite stockée dans `COCG`.

## Phase itérative $k$, $k \in \mathbb{N}$

On suppose $\delta \underline{G}_i^k$ connu et donc $\underline{G}_{c,i}^k$ pour $k$ donné et sur toute cellule $\Omega_i$ et on veut calculer $\delta \underline{G}_i^{k+1}$ et $\underline{G}_{c,i}^{k+1}$ .

**Calcul du second membre $\underline{R}_i^{k+1}$ et résolution**

Le calcul proprement dit du second membre $\underline{R}_i^{k+1}$ correspondant au système (II.9.9) est effectué et stocké dans le tableau (`BX`, `BY`, `BZ`). Il est initialisé, à chaque pas $k$, par la valeur du gradient $\underline{G}_{c,i}^k$ multiplié par le volume de la cellule $|\Omega_i|$, avec $\underline{G}_{c,i}^0 = \underline{G}_{c,i}^{Nrec}$ . L'incrément ($\delta \underline{G}_i^{k+1}$) de gradient est obtenu par $\underline{\underline{C}}_i^{-1} \underline{R}_i^{k+1}$ et ajouté dans les tableaux `DPDX`, `DPDY` et `DPDZ` pour obtenir $\underline{G}_{c,i}^{k+1}$.

En ce qui concerne les conditions aux limites en pression, elles sont traitées comme suit dans *Code_Saturne*:

$$\left\{\begin{array}{ll} P_{I'} & = P_I + \underline{II'}.\underline{G}_{c,i} \\ P_{b,ik} & = \texttt{INC}\, A_{b,ik} + B_{b,ik}\, P_{I'} = \texttt{INC}\, A_{b,ik} + B_{b,ik}(P_I + \underline{II'}.\underline{G}_{c,i}) \\ P_{b_1,ik} & = P_I + \underline{IF}_{ij}.\underline{G}_{c,i} \\ P_{f,\,b\,ik} & = B_{b,ik}(\texttt{EXTRAP}\, P_{b_1,ik} + (1 - \texttt{EXTRAP})\, P_{b,ik}) + (1 - B_{b,ik}) P_{b,ik} \end{array}\right.$$

ce qui correspond à :

- lorsqu'on veut imposer des conditions de Dirichlet ($A_{b,ik} = P_{F_{b\,ik}}$, $B_{b,ik} = 0$),

$$P_{F_{b\,ik}} = P_{IMPOSE} \tag{II.9.24}$$

pour toute valeur de `EXTRAP`.

- lorsqu'on veut imposer des conditions de flux ($A_{b,ik} = 0$, $B_{b,ik} = 1$) (condition de type Neumann)

$$P_{F_{b\,ik}} = \texttt{EXTRAP}\ (P_I + (\underline{IF}_{b\,ik}.(\underline{\text{grad}}\ P)_I) + (1 - \texttt{EXTRAP}) P_{I'} \tag{II.9.25}$$

seules trois valeurs de `EXTRAP` sont licites.

- avec un maillage non orthogonal

L'ordre obtenu est égal à 1 dans tous les cas.

| | | | |
|---|---|---|---|
| `EXTRAP` $= 0$ | Neumann homogène | $P_{F_{b\,ik}} = P_{I'} + \mathcal{O}(h^2)$ |
| `EXTRAP` $= \frac{1}{2}$ | Neumann homogène amélioré | $P_{F_{b\,ik}} = P_{I'} + \frac{1}{2}\underline{I'F}_{b\,ik}.(\underline{\text{grad}}\ P)_I + \mathcal{O}(h^2)$ |
| `EXTRAP` $= 1$ | extrapolation du gradient | $P_{F_{b\,ik}} = P_{I'} + \underline{I'F}_{b\,ik}.(\underline{\text{grad}}\ P)_I + \mathcal{O}(h^2)$ |

- avec un maillage orthogonal

On peut atteindre l'ordre deux.

| | | | |
|---|---|---|---|
| `EXTRAP` | $= 0$ | Neumann homogène<br>on est à l'ordre 1 | $P_{F_{b\,ik}} = P_{I'} + \mathcal{O}(h^2)$ |
| `EXTRAP` | $= \frac{1}{2}$ | Neumann homogène amélioré<br>on est à l'ordre 2 | $P_{F_{b\,ik}} = P_I + \frac{1}{2}\,\underline{IF}_{b\,ik} \,.\, (\underline{\text{grad}}\ P)_I + \mathcal{O}(h^3)$ |
| `EXTRAP` | $= 1$ | extrapolation du gradient<br>on est à l'ordre 2 | $P_{F_{b\,ik}} = P_I + \underline{IF}_{b\,ik} \,.\, (\underline{\text{grad}}\ P)_I + \mathcal{O}(h^3)$ |

**Test de convergence de la méthode itérative de résolution**

On calcule la norme euclidienne `RESIDU` du second membre (`BX`, `BY`, `BZ`).
On arrête les itérations sur $k$ si le test de convergence pour cette norme ou le nombre de sweeps maximal `NSWRGP` est atteint. La valeur par défaut de `NSWRGP` est fixée à 100, ce qui permet un calcul suffisamment précis pour l'ordre d'espace considéré.
Sinon, on continue d'itérer sur $k$.

## 9.4 Points à traiter

**• Vectorisation forcée**

Il est peut-être possible de s'affranchir du découpage en boucles de 1024 si les compilateurs sont capables d'effectuer la vectorisation sans cette aide. On note cependant que ce découpage en boucles de 1024 n'a pas de coût CPU supplémentaire, et que le coût mémoire associé est négligeable. Le seul inconvénient réside dans la relative complexité de l'écriture.

**• Traitement des conditions aux limites de pression**

Actuellement, l'ordre deux décrit dans le cas `EXTRAP` $= \frac{1}{2}$ relativement aux conditions aux limites de pression n'existe pas dans *Code_Saturne* en non orthogonal. Mais en a-t-on vraiment besoin ?

**• Méthode itérative de résolution**

La méthode itérative de résolution adoptée dans ce sous-programme marche, *i.e.* converge, mais ne rentre dans aucun cadre théorique précis. Des réflexions sur le sujet pourraient éventuellement permettre d'exhiber certaines propriétés des matrices considérées, cerner les limites d'application ou expliquer certains comportements.

# 10- Sous-programme inimas

## 10.1 Fonction

Le but de ce sous-programme est principalement de calculer le flux de masse aux faces. Il prend une variable vectorielle associée au centre des cellules (généralement la vitesse), la projette aux faces en la multipliant par la masse volumique, et la multiplie scalairement par le vecteur surface. Plus généralement, `inimas` est aussi appelé comme première étape du calcul d'une divergence (terme en $\mathrm{div}(\rho\underline{\underline{R}})$ en $R_{ij} - \varepsilon$, filtre Rhie & Chow, ...).

## 10.2 Discrétisation

La figure II.10.1 rappelle les diverses définitions géométriques pour les faces internes et les faces de bord. On notera $\alpha = \dfrac{\overline{FJ'}}{\overline{I'J'}}$ (défini aux faces internes uniquement).

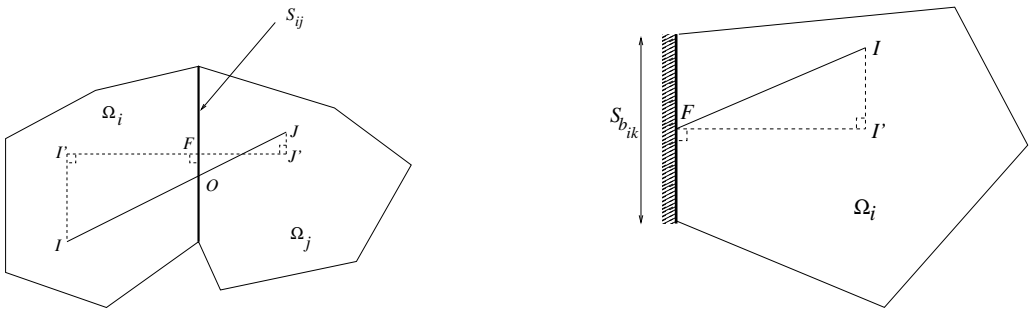



Figure II.10.1: Définition des différentes entités géométriques pour les faces internes (gauche) et de bord (droite).

### 10.2.1 Faces internes

On ne connaît pas la masse volumique à la face, cette dernière doit donc aussi être interpolée. On utilise la discrétisation suivante :

$$(\rho\underline{u})_F = \alpha(\rho_I\underline{u}_I) + (1-\alpha)(\rho_J\underline{u}_J) + \underline{\underline{\mathrm{grad}}}\,(\rho\underline{u})_O.\underline{OF} \qquad \text{(II.10.1)}$$

La partie en $\alpha(\rho_I\underline{u}_I) + (1-\alpha)(\rho_J\underline{u}_J)$ correspondant en fait à $(\rho\underline{u})_O$. Le gradient en $O$ est calculé par interpolation : $\underline{\underline{\mathrm{grad}}}\,(\rho\underline{u})_O = \dfrac{1}{2}\left[\underline{\underline{\mathrm{grad}}}\,(\rho\underline{u})_I + \underline{\underline{\mathrm{grad}}}\,(\rho\underline{u})_J\right]$. La valeur $\dfrac{1}{2}$ s'est imposée de manière heuristique au fil des tests comme apportant plus de stabilité à l'algorithme global qu'une interpolation faisant intervenir $\alpha$. L'erreur commise sur $\rho\underline{u}$ est en $O(h^2)$.

### 10.2.2 Faces de bord

Le traitement des faces de bord est nécessaire pour y calculer le flux de masse, bien sûr, mais aussi pour obtenir des conditions aux limites pour le calcul du $\underline{\underline{\mathrm{grad}}}\,(\rho\underline{u})$ utilisé pour les faces internes.

Pour les faces de bord, on connaît la valeur de $\rho_F$, qui est stockée dans la variable `ROMB`. De plus, les conditions aux limites pour $\underline{u}$ sont données par des coefficients $A$ et $B$ tels que :

$$u_{k,F} = A_k + B_k u_{k,I'} = A_k + B_k \left(u_{k,I} + \underline{\text{grad}}\,(u_k)_I . \underline{II'}\right) \tag{II.10.2}$$

($k \in \{1,2,3\}$ est la composante de la vitesse, l'erreur est en $O(B_k h)$)

On a donc à l'ordre 1 :

$$(\rho u_k)_F = \rho_F \left[A_k + B_k \left(u_{k,I} + \underline{\text{grad}}\,(u_k)_I . \underline{II'}\right)\right] \tag{II.10.3}$$

Mais pour utiliser cette formule, il faudrait calculer $\underline{\underline{\text{grad}}}\,(\underline{u})$ (trois appels à `GRDCEL`), alors qu'on a déjà calculé $\underline{\underline{\text{grad}}}\,(\rho\underline{u})$ pour les faces internes. Le surcoût en temps serait alors important. On réécrit donc :

$$\begin{aligned}
(\rho u_k)_F &= \rho_F A_k + \rho_F B_k u_{k,I'} && \text{(II.10.4)}\\
&= \rho_F A_k + B_k \frac{\rho_F}{\rho_{I'}} (\rho u_k)_{I'} && \text{(II.10.5)}\\
&= \rho_F A_k + B_k \frac{\rho_F}{\rho_{I'}} (\rho u_k)_I + B_k \frac{\rho_F}{\rho_{I'}} \underline{\text{grad}}\,(\rho u_k)_I . \underline{II'} && \text{(II.10.6)}
\end{aligned}$$

Pour calculer les gradients de $\rho\underline{u}$, il faudrait donc en théorie utiliser les coefficients de conditions aux limites équivalents :
$\tilde{A}_k = \rho_F A_k$
$\tilde{B}_k = B_k \dfrac{\rho_F}{\rho_{I'}}$

Ceci paraît délicat, à cause du terme en $\dfrac{\rho_F}{\rho_{I'}}$, et en particulier à l'erreur que l'on peut commettre sur $\rho_{I'}$ si la reconstruction des gradients est imparfaite (sur des maillages fortement non orthogonaux par exemple). On réécrit donc l'équation (II.10.6) sous la forme suivante :

$$(\rho u_k)_F = \rho_F A_k + B_k \frac{\rho_I \rho_F}{\rho_{I'}} u_{k,I} + B_k \frac{\rho_F}{\rho_{I'}} \underline{\text{grad}}\,(\rho u_k)_I . \underline{II'} \tag{II.10.7}$$

Pour le calcul du flux de masse au bord, on va faire deux approximations. Pour le deuxième terme, on va supposer $\rho_{I'} \approx \rho_I$ (ce qui conduit à une erreur en $O(B_k h)$ sur $\rho\underline{u}$ si $\rho_{I'} \neq \rho_I$). Pour le troisième terme, on va supposer $\rho_{I'} \approx \rho_F$. Cette dernière approximation est plus forte, mais elle n'intervient que dans la reconstruction des non-orthogonalités ; l'erreur finale reste donc faible (erreur en $O(B_k h^2)$ sur $\rho\underline{u}$ si $\rho_{I'} \neq \rho_F$). Et au final, le flux de masse au bord est calculé par :

$$\dot{m}_F = \sum_{k=1}^{3} \left[\rho_F A_k + B_k \rho_F u_{k,I} + B_k \underline{\text{grad}}\,(\rho u_k)_I . \underline{II'}\right] S_k \tag{II.10.8}$$

Pour le calcul des gradients, on repart de l'équation (II.10.5), sur laquelle on fait l'hypothèse que $\rho_{I'} \approx \rho_F$. Encore une fois, cette hypothèse peut être assez forte, mais les gradients obtenus ne sont utilisés que pour des reconstructions de non-orthogonalités ; l'erreur finale reste donc là encore assez faible. Au final, les gradients sont calculés à partir de la formule suivante :

$$(\rho u_k)_F = \rho_F A_k + B_k (\rho u_k)_{I'} \tag{II.10.9}$$

ce qui revient à utiliser les conditions aux limites suivantes pour $\rho\underline{u}$:
$\tilde{A}_k = \rho_F A_k$
$\tilde{B}_k = B_k$

Remarque

Dans la plupart des cas, les approximations effectuées n'engendrent aucune erreur. En effet :
- dans le cas d'une entrée on a généralement $B_k = 0$, avec un flux de masse imposé par la condition à

la limite.
- dans le cas d'une sortie, on a généralement flux nul sur les scalaires donc sur $\rho$, soit $\rho_F = \rho_{I'} = \rho_I$.
- dans le cas d'une paroi, on a généralement $B_k = 0$ et le flux de masse est imposé nul.
- dans le cas d'une symétrie, on a généralement $\rho_F = \rho_{I'} = \rho_I$ et le flux de masse est imposé nul.
Pour sentir un effet de ces approximations, il faudrait par exemple une paroi glissante ($B_k \neq 0$) avec un gradient de température ($\rho_F \neq \rho_I$).

## 10.3 Mise en œuvre

La vitesse est passée par les arguments `UX`, `UY` et `UZ`. Les conditions aux limites de la vitesse sont `COEFAX`, `COEFBX`, ... Le flux de masse résultat est stocké dans les variables `FLUMAS` (faces internes) et `FLUMAB` (faces de bord). `QDMX`, `QDMY` et `QDMZ` sont des variables de travail qui serviront à stocker $\rho\underline{u}$, et `COEFQA` servira à stocker les $\tilde{A}$.

**• Initialisation éventuelle du flux de masse**
Si `INIT` vaut 1, le flux de masse est remis à zéro. Sinon, le sous-programme rajoute aux variables `FLUMAS` et `FLUMAB` existantes le flux de masse calculé.

**• Remplissage des tableaux de travail**
$\rho\underline{u}$ est stocké dans `QDM`, et $\tilde{A}$ dans `COEFQA`.

**• Cas sans reconstruction**
On calcule alors directement

$$\texttt{FLUMAS} = \sum_{k=1}^{3} \left[\alpha(\rho_I u_{k,I}) + (1-\alpha)(\rho_J u_{k,J})\right] S_k$$

et

$$\texttt{FLUMAB} = \sum_{k=1}^{3} \left[\rho_F A_k + B_k \rho_F u_{k,I}\right] S_k$$

**• Cas avec reconstruction**
On répète trois fois de suite les opérations suivantes, pour $k = 1$, 2 et 3 :
- Appel de `GRDCEL` pour le calcul de $\underline{\text{grad}}\,(\rho u_k)$.
- Mise à jour du flux de masse

$$\texttt{FLUMAS} = \texttt{FLUMAS} + \left[\alpha(\rho_I u_{k,I}) + (1-\alpha)(\rho_J u_{k,J}) + \frac{1}{2}\left[\underline{\text{grad}}\,(\rho u_k)_I + \underline{\text{grad}}\,(\rho u_k)_J\right].\underline{OF}\right] S_k$$

et

$$\texttt{FLUMAB} = \texttt{FLUMAB} + \left[\rho_F A_k + B_k \rho_F u_{k,I} + B_k \underline{\text{grad}}\,(\rho u_k)_I.\underline{II'}\right] S_k$$

**• Annulation du flux de masse au bord**
Quand le sous-programme a été appelé avec la valeur `IFLMB0=1` (c'est-à-dire quand il est réellement appelé pour calculer un flux de masse, et pas pour calculer le terme en div($\rho\underline{\underline{R}}$) par exemple), le flux de masse au bord `FLUMAB` est forcé à 0, pour les faces de paroi et de symétrie (identifiées par `ISYMPA=0`).

# 11- Sous-programme itrmas/itrgrp

## 11.1 Fonction

Le but du sous-programme **itrmas** est de calculer un gradient de pression "facette". Il est utilisé dans la phase de correction de pression (deuxième phase de **navsto**) où le flux de masse est mis à jour à l'aide de termes en $-\Delta t_{ij}(\underline{\text{grad}}\ _f P)_{ij}.\underline{S}_{ij}$ et en $-\Delta t_{b_{ik}}(\underline{\text{grad}}\ _f P)_{b_{ik}}.\underline{S}_{b_{ik}}$.

Le sous-programme **itrgrp** calcule un gradient "facette" de pression et en prend la divergence, c'est-à-dire calcule le terme :

$$-\sum_{j\in Vois(i)} \Delta t_{ij}(\underline{\text{grad}}\ _f P)_{ij}.\underline{S}_{ij} - \sum_{k\in\gamma_b(i)} \Delta t_{b_{ik}}(\underline{\text{grad}}\ _f P)_{b_{ik}}.\underline{S}_{b_{ik}}$$

En pratique **itrgrp** correspond à la combinaison de **itrmas** et de **divmas**, mais permet par son traitement en un seul bloc d'éviter la définition de tableaux de travail de taille **NFAC** et **NFABOR**.

## 11.2 Discrétisation

La figure II.11.1 rappelle les diverses définitions géométriques pour les faces internes et les faces de bord.

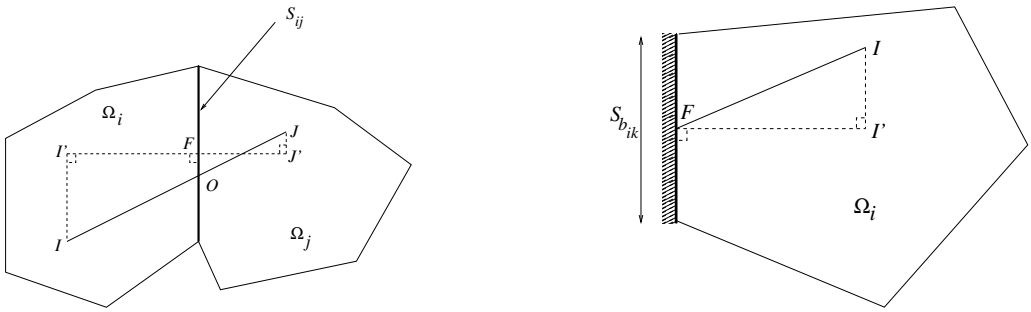



Figure II.11.1: Définition des différentes entités géométriques pour les faces internes (gauche) et de bord (droite).

### 11.2.1 Calcul sans reconstruction des non orthogonalités

Pour les faces internes, on écrit simplement :

$$-\Delta t_{ij}(\underline{\text{grad}}\ _f P)_{ij}.\underline{S}_{ij} = \frac{\Delta t_{ij} S_{ij}}{\overline{I'J'}}(P_I - P_J) \qquad \text{(II.11.1)}$$

Pour les faces de bord, on écrit :

$$-\Delta t_{b_{ik}}(\underline{\text{grad}}\ _f P)_{b_{ik}}.\underline{S}_{b_{ik}} = \frac{\Delta t_{b_{ik}} S_{b_{ik}}}{\overline{I'F}}\left((1-B_{b,ik})P_I - \texttt{INC}\times A_{b,ik}\right) \qquad \text{(II.11.2)}$$

Les pas de temps aux faces $\Delta t_{ij}$ et $\Delta t_{b_{ik}}$ sont calculés par interpolation par les sous-programmes **viscfa** (cas isotrope, **IPUCOU=0**) ou **visort** (cas anisotrope, **IPUCOU=1**).

### 11.2.2 Calcul avec reconstruction des non orthogonalités

Plusieurs discrétisations peuvent être proposées pour le traitement des non orthogonalités. Celle retenue dans le code est issue des premiers tests réalisés sur le prototype, et fait intervenir non seulement le pas de temps interpolé à la face, mais aussi les pas de temps dans chaque cellule. Il serait sans doute bon de revenir sur cette écriture et évaluer d'autres solutions. La forme utilisée pour les faces internes est :

$$-\Delta t_{ij}(\underline{\text{grad}}\ {}_f P)_{ij}\cdot\underline{S}_{ij} = \frac{\Delta t_{ij}S_{ij}}{\overline{I'J'}}(P_I - P_J) + (\underline{II'} - \underline{JJ'}).\frac{1}{2}\left[\Delta t_I(\underline{\text{grad}}\ P)_I + \Delta t_J(\underline{\text{grad}}\ P)_J\right]\frac{S_{ij}}{\overline{I'J'}} \quad \text{(II.11.3)}$$

Pour les faces de bord, on écrit :

$$-\Delta t_{b_{ik}}(\underline{\text{grad}}\ {}_f P)_{b_{ik}}\cdot\underline{S}_{b_{ik}} = \frac{\Delta t_{b_{ik}}S_{b_{ik}}}{\overline{I'F}}\left[(1-B_{b,ik})(P_I + \underline{II'}.(\underline{\text{grad}}\ P)_I) - \texttt{INC}\times A_{b,ik}\right] \quad \text{(II.11.4)}$$

## 11.3 Mise en œuvre

Les principaux arguments passés à `itrmas` et `itrgrp` sont la variable traitée `PVAR` (la pression), ses conditions aux limites, le pas de temps projeté aux faces[1] (`VISCF` et `VISCB`), le pas de temps au centre des cellules, éventuellement anisotrope (`VISELX`, `VISELY`, `VISELZ`). `itrmas` retourne les tableaux `FLUMAS` et `FLUMAB` (flux de masse aux faces) mis à jour. `itrgrp` retourne directement la divergence du flux de masse mis à jour, dans le tableau `DIVERG`.

**• Initialisation**
Si `INIT` vaut 1, les variables `FLUMAS` et `FLUMAB` ou `DIVERG` sont mises à zéro.

**• Cas sans reconstruction**
La prise en compte ou non des non orthogonalités est déterminée par l'indicateur `NSWRGR` de la variable traitée (nombre de sweeps de reconstruction des non orthogonalités dans le calcul des gradients), passé par la variable `NSWRGP`. Une valeur inférieure ou égale à 1 enclenche le traitement sans reconstruction. Des boucles sur les faces internes et les faces de bord utilisent directement les formules (II.11.1) et (II.11.2) pour remplir les tableaux `FLUMAS` et `FLUMAB` (`itrmas`) ou des variables de travail `FLUMAS` et `FLUMAB` qui servent à mettre à jour le tableau `DIVERG` (`itrgrp`).

À noter que les tableaux `VISCF` et `VISCB` contiennent respectivement $\frac{\Delta t_{ij}S_{ij}}{\overline{I'J'}}$ et $\frac{\Delta t_{b_{ik}}S_{b_{ik}}}{\overline{I'F}}$.

**• Cas avec reconstruction**
Après un appel à `GRDCEL` pour calculer le gradient cellule de pression, on remplit les tableaux `FLUMAS` et `FLUMAB` ou `DIVERG` là encore par une application directe des formules (II.11.3) et (II.11.4).

## 11.4 Points à traiter

Il est un peu redondant de passer en argument à la fois le pas de temps projeté aux faces et le pas de temps au centre des cellules. Il faudrait s'assurer de la réelle nécessité de cela, ou alors étudier des formulations plus simples de la partie reconstruction.

[1] Plus précisément, le pas de temps projeté aux faces, multiplié par la surface et divisé par $\overline{I'J'}$ ou $\overline{I'F}$, cf. `viscfa`

# 12- Sous-programme `matrix`

## 12.1 Fonction

Le but de ce sous-programme, appelé par `codits` et `covofi`, est de construire la matrice de convection/diffusion, incluant les contributions adéquates des termes sources, intervenant dans le membre de gauche d'équations discrétisées telles que celle de la quantité de mouvement, d'une équation de convection diffusion d'un scalaire ou de modèle de turbulence.
Le type d'équation considérée est, pour la variable scalaire $a$ :

$$\frac{\partial a}{\partial t} + \mathrm{div}(\,(\rho \underline{u})\, a) - \frac{\partial}{\partial x}\left(\beta \frac{\partial a}{\partial x}\right) = 0$$

La matrice ne s'applique qu'aux termes non reconstruits, les autres étant pris en compte au second membre et traités dans le sous-programme `bilsc2`. La partie convective, lorsqu'elle existe, est issue du schéma upwind pur, quelque soit le type de schéma convectif choisi par l'utilisateur. En effet, c'est, à l'heure actuelle, la seule façon d'obtenir un opérateur linéaire à diagonale dominante.

La matrice est donc associée à $\mathcal{EM}_{scal}$, opérateur agissant sur un scalaire $a$ (inspiré de celui vectoriel $\mathcal{EM}$ défini dans `navsto`) d'expression actuelle, pour tout cellule $\Omega_i$ de centre $I$ :

$$\begin{aligned}\mathcal{EM}_{scal}(a,I) &= f_s^{imp}\, a_I \\ &+ \sum_{j\in Vois(i)} F_{ij}^{amont}((\rho\underline{u})^n, a) + \sum_{k\in\gamma_b(i)} F_{b_{ik}}^{amont}((\rho\underline{u})^n, a) \\ &- \sum_{j\in Vois(i)} D_{ij}^{NRec}(\beta, a) - \sum_{k\in\gamma_b(i)} D_{b_{ik}}^{NRec}(\beta, a)\end{aligned}$$

avec :
• $f_s^{imp}$ le coefficient issu du terme instationnaire $\dfrac{\rho\,|\Omega_i|}{\Delta t}$, s'il y a lieu, et de l'implicitation de certains termes sources (contribution découlant de la prise en compte de la variation $\dfrac{\partial \rho}{\partial t}$ de la masse volumique $\rho$ au cours du temps, diagonale du tenseur de pertes de charges par exemple...): cette initialisation est en fait effectuée en amont de ce sous-programme,
• $F_{ij}^{amont}$ le flux numérique convectif scalaire décentré amont calculé à la face interne $ij$ de la cellule $\Omega_i$,
• $F_{b_{ik}}^{amont}$ le flux numérique convectif scalaire décentré amont associé calculé à la face de bord $ik$ de la cellule $\Omega_i$ jouxtant le bord du domaine $\Omega$,
• $D_{ij}^{NRec}$ le flux numérique diffusif scalaire non reconstruit associé calculé à la face interne $ij$ de la cellule $\Omega_i$,
• $D_{b_{ik}}^{NRec}$ le flux numérique diffusif scalaire non reconstruit associé calculé à la face de bord $ik$ de la cellule $\Omega_i$ jouxtant le bord du domaine $\Omega$,
• $Vois(i)$ représente toujours l'ensemble des cellules voisines de $\Omega_i$ et $\gamma_b(i)$ l'ensemble des faces de bord de $\Omega_i$.
Du fait de la résolution en incréments, $a$ est un incrément et ses conditions aux limites associées sont donc de type Dirichlet homogène ou de type Neumann homogène.

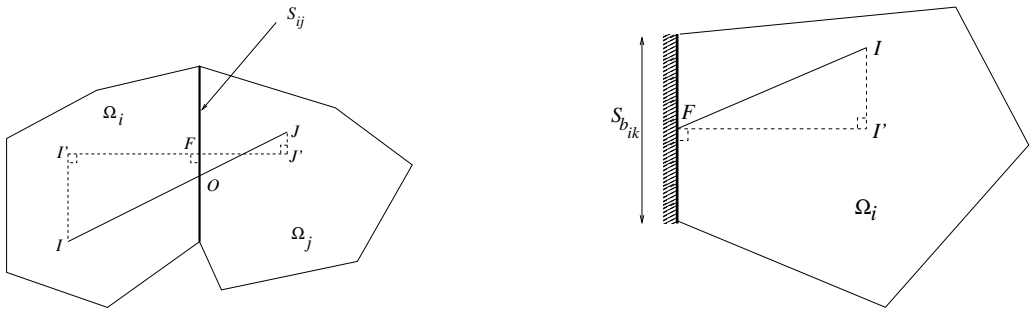



Figure II.12.1: Définition des différentes entités géométriques pour les faces internes (gauche) et de bord (droite).

## 12.2 Discrétisation

L'opérateur $\mathcal{EM}_{scal}$ s'écrit, pour tout $I$ centre de cellule :

$$\begin{aligned}\mathcal{EM}_{scal}(a,I) &= f_s^{imp}\, a_I \\ &+ \sum_{j\in Vois(i)} \left[(\rho\underline{u})^n_{ij}.\underline{S}_{ij}\right]\ a_{f,ij} + \sum_{k\in\gamma_b(i)} \left[(\rho\underline{u})^n_{b_{ik}}.\underline{S}_{b_{ik}}\right]\ a_{f\,b_{ik}} \\ &- \sum_{j\in Vois(i)} \beta_{ij}\frac{a_J - a_I}{\overline{I'J'}}S_{ij} - \sum_{k\in\gamma_b(i)} \beta_{b_{ik}}\frac{a_{b_{ik}} - a_I}{\overline{I'F}}S_{b_{ik}}\end{aligned} \qquad \text{(II.12.1)}$$

où $a_{f,ij} = a_I$ ou $a_J$ selon le signe de $(\rho\underline{u})^n_{ij}.\underline{S}_{ij}$ (schéma convectif upwind systématique), et avec $\overline{I'J'}$, mesure algébrique, orientée comme la normale sortante à la face, *i.e.* allant de $I$ vers $J$ pour la cellule $\Omega_i$ de centre $I$. On la notera $\overline{I'J'}^{\,I}$ lorsqu'on aura besoin d'expliciter clairement l'orientation.
$a_{f\,b_{ik}} = a_I$ ou $a_{b_{ik}}$ selon le signe de $(\rho\underline{u})^n_{b_{ik}}.\underline{S}_{b_{ik}}$ (schéma upwind systématique) et $a_{b_{ik}}$ valeur au bord est donnée directement par les conditions aux limites (valeur non reconstruite). $\overline{I'F}$, mesure algébrique, orientée relativement à la normale sortante à la face, *i.e.* allant de $I$ vers l'extérieur du domaine.
En général, sauf cas pathologiques, les mesures algébriques $\overline{I'J'}$ et $\overline{I'F}$ sont positives et correspondent aux distances $I'J'$ et $I'F$. On se reportera au paragraphe Points à traiter pour plus de détails.
Soit $\underline{\underline{EM}}_{scal}$ la matrice associée ; sa taille est *a priori* de `NCEL`$*$`NCEL`, mais compte-tenu de la nature de la structure de données, seuls deux tableaux `DA(NCEL)` contenant les valeurs diagonales et `XA(NFAC,*)` les contributions des termes extra-diagonaux sont nécessaires, avec `NCEL` nombre de cellules du maillage considéré et `NFAC` nombre de faces internes associé.
Du fait des simplifications effectuées sur la matrice (non reconstruction des termes), les composantes extradiagonales de la ligne $I$ ne sont différentes de zéro que pour les indices $J$ des cellules voisines de $I$. On peut donc stocker toutes les contributions non nulles de la matrice dans deux tableaux `DA(NCEL)` et `XA(NFAC,2)` :
• `DA(I)` est le coefficient de la colonne $I$ dans la ligne $I$,
• si `IFAC` est une face qui sépare les cellules $\Omega_i$ et $\Omega_j$, orientée de $I$ vers $J$, alors :
`XA(IFAC,1)` est le coefficient de la colonne $J$ dans la ligne $I$ et `XA(IFAC,2)` est le coefficient de la colonne $I$ dans la ligne $J$. Lorsque la matrice est symétrique, *i.e.* lorsqu'il n'y a pas de convection (`ICONVP` $= 0$) et que seule la diffusion est à prendre en compte, alors `XA(IFAC,1)` = `XA(IFAC,2)` et on réduit `XA` à `XA(NFAC,1)`.

Soit $m^n_{ij}$ ( $m^n_{b_{ik}}$ ) la valeur de $(\rho\underline{u})^n_{ij}.\underline{S}_{ij}$ (respectivement $(\rho\underline{u})^n_{b_{ik}}.\underline{S}_{b_{ik}}$).
Alors :

- contribution volumique : $f_s^{imp}\, a_I$

- contribution d'une face purement interne $ij$

L'expression

$$+ \sum_{j \in Vois(i)} F_{ij}^{amont}((\rho \underline{u})^n, a) - \sum_{j \in Vois(i)} D_{ij}^{NRec}(\beta, a)$$

s'écrit :

$$\begin{aligned} &\sum_{j \in Vois(i)} \left( \left[ (\rho \underline{u})^n_{ij} . \underline{S}_{ij} \right] \quad a_{f,ij} - \beta_{ij} \frac{a_J - a_I}{\overline{I'J'}} S_{ij} \right) \\ &= \sum_{j \in Vois(i)} \left[ \frac{1}{2} ( \ m^n_{ij} + | \ m^n_{ij} | \ ) a_I + \frac{1}{2} ( \ m^n_{ij} - | \ m^n_{ij} | ) a_J \right] - \sum_{j \in Vois(i)} \beta_{ij} \frac{a_J - a_I}{\overline{I'J'}} S_{ij} \end{aligned} \tag{II.12.2}$$

Ici, $\overline{I'J'} = \overline{I'J'}^{\,I}$.

■ contribution d'une face de bord *ik*

De même :

$$\begin{aligned} &\sum_{k \in \gamma_b(i)} F_{b_{ik}}^{amont}((\rho \underline{u})^n, a) - \sum_{k \in \gamma_b(i)} D_{b_{ik}}^{NRec}(\beta, a) \\ &= \sum_{k \in \gamma_b(i)} \left( \left[ (\rho \underline{u})^n_{b_{ik}} . \underline{S}_{b_{ik}} \right] \ a_{f\,b_{ik}} - \beta_{b_{ik}} \frac{a_{\,b_{ik}} - a_I}{\overline{I'F}} S_{b_{ik}} \right) \\ &= \sum_{k \in \gamma_b(i)} \left[ \frac{1}{2} ( \ m^n_{b_{ik}} + | \ m^n_{b_{ik}} | \ ) a_I + \frac{1}{2} ( \ m^n_{b_{ik}} - | m^n_{b_{ik}} | ) a_{\,b_{ik}} \right] - \sum_{k \in \gamma_b(i)} \beta_{b_{ik}} \frac{a_{\,b_{ik}} - a_I}{\overline{I'F}} S_{b_{ik}} \end{aligned} \tag{II.12.3}$$

avec :

$$a_{\,b_{ik}} = \texttt{INC}\, A_{\,b,ik} + B_{\,b,ik}\, a_I = B_{\,b,ik}\, a_I$$

$a$ n'étant associée qu'à des conditions aux limites de type Dirichlet homogène ou de type Neumann homogène.

## 12.3 Mise en œuvre

### 12.3.1 Initialisations

L'indicateur de symétrie `ISYM` de la matrice considérée est affecté comme suit :

• `ISYM` = 1 , si la matrice est symétrique ; on travaille en diffusion pure , `ICONVP` = 0 et `IDIFFP` = 1,

• `ISYM` = 2 , si la matrice est non symétrique ; on travaille soit en convection pure ( `ICONVP` = 1, `IDIFFP` = 0 ), soit en convection/diffusion ( `ICONVP` = 1, `IDIFFP` = 1 ).

Les termes diagonaux de la matrice sont stockés dans le tableau `DA(NCEL)`. Ceux extra-diagonaux le sont dans `XA(NFAC,1)` si la matrice est symétrique, dans `XA(NFAC,2)` sinon.

Le tableau `DA` est initialisé à zéro pour un calcul avec `ISTATP` = 0 (en fait, ceci ne concerne que les calculs relatifs à la pression). Sinon, on lui affecte la valeur `ROVSDT` comprenant la partie instationnaire, la contribution du terme continu en $-\ a\ \mathrm{div}(\rho \underline{u})^n$ et la partie diagonale des termes sources implicités. Le tableau `XA(NFAC,*)` est initialisé à zéro.

### 12.3.2 Calcul des termes extradiagonaux stockés dans `XA`

Ils ne se calculent que pour des faces purement internes `IFAC`, les faces de bord ne contribuant qu'à la diagonale.

#### matrice non symétrique ( présence de convection )

Pour chaque face interne `IFAC`, les contributions extradiagonales relatives au terme $a_I$ et à son voisin associé $a_J$ sont calculées dans `XA(IFAC,1)` et `XA(IFAC,2)` respectivement (pour une face orientée de I

vers J).
On a les relations suivantes :

$$\begin{array}{l} \texttt{XA(IFAC,1)} = \texttt{ICONVP} * \texttt{FLUI} - \texttt{IDIFFP} * \texttt{VISCF(IFAC)} \\ \texttt{XA(IFAC,2)} = \texttt{ICONVP} * \texttt{FLUJ} - \texttt{IDIFFP} * \texttt{VISCF(IFAC)} \end{array} \quad \text{(II.12.4)}$$

avec `FLUMAS(IFAC)` correspondant à $m_{ij}^n$, `FLUI` à $\frac{1}{2}( m_{ij}^n - \mid m_{ij}^n \mid )$, `VISCF(IFAC)` à $\beta_{ij}\frac{S_{ij}}{\overline{I'J'}}$.

`XA(IFAC,1)` représente le facteur de $a_J$ dans la dernière expression de (II.12.2).

`FLUJ` correspond à $-\frac{1}{2}( m_{ij}^n + \mid m_{ij}^n \mid )$. En effet, `XA(IFAC,2)` est le facteur de $a_I$ dans l'expression équivalente de la dernière ligne de (II.12.2), mais écrite en J.
Ce qui donne :

$$\sum_{i \in Vois(j)} \left[ \frac{1}{2}( m_{ji}^n + \mid m_{ji}^n \mid ) a_J + \frac{1}{2}( m_{ji}^n - \mid m_{ji}^n \mid ) a_I \right] - \sum_{i \in Vois(j)} \beta_{ji} \frac{a_I - a_J}{\overline{J'I'}} S_{ji} \quad \text{(II.12.5)}$$

Le terme recherché est donc : $\frac{1}{2}( m_{ji}^n - \mid m_{ji}^n \mid ) - \beta_{ji}\frac{S_{ji}}{\overline{J'I'}}$ .
Or :
$m_{ji}^n = -m_{ij}^n$ ($\underline{S}_{ji} = -\underline{S}_{ij}$ et $(\rho \underline{u})_{ji}^n = (\rho \underline{u})_{ij}^n$ ), avec $\overline{J'I'}$ mesure algébrique, orientée relativement à la normale sortante à la face, *i.e.* allant de $J$ vers $I$. On la note $\overline{J'I'}^{\,J}$.
On a la relation :

$$\overline{J'I'}^{\,J} = \overline{I'J'}^{\,I} = ( \overline{I'J'} ) \quad \text{(II.12.6)}$$

d'où la deuxième égalité dans (II.12.4).

### matrice symétrique ( diffusion pure )

Pour chaque face interne `IFAC`, la contribution extradiagonale relative au terme $a_I$ est calculée dans `XA(IFAC,1)` par la relation suivante :

$$\texttt{XA(IFAC,1)} = -\texttt{IDIFFP} * \texttt{VISCF(IFAC)} \quad \text{(II.12.7)}$$

avec `VISCF(IFAC)` à $\beta_{ij}\frac{S_{ij}}{\overline{I'J'}}$.

## 12.3.3 Calcul des termes diagonaux stockés dans `DA`

### matrice non symétrique ( présence de convection )

Pour chaque face interne $ij$ (`IFAC`) séparant les cellules $\Omega_i$ de centre $I$ et $\Omega_j$ de centre $J$, `DA(II)` est la quantité en facteur de $a_I$ dans la dernière expression de (II.12.2), soit :

$$\frac{1}{2}( m_{ij}^n + \mid m_{ij}^n \mid ) + \beta_{ij}\frac{S_{ij}}{\overline{I'J'}} \quad \text{(II.12.8)}$$

De même, pour `DA(JJ)`, on a :

$$\frac{1}{2}( -m_{ij}^n + \mid m_{ij}^n \mid ) + \beta_{ji}\frac{S_{ij}}{\overline{I'J'}} \quad \text{(II.12.9)}$$

d'après l'expression de (II.12.5) et l'égalité (II.12.6).
L'implantation dans *Code_Saturne* associée est la suivante :
pour toute face `IFAC` d'éléments voisins `II` = `IFACEL(1,IFAC)` et `JJ` = `IFACEL(2,IFAC)`,
on ajoute à `DA(II)` la contribution croisée $-$`XA(IFAC,2)` (*cf.* (II.12.8)) et à `DA(JJ)` la contribution $-$`XA(IFAC,1)` (*cf.* (II.12.9)).

### 12.3.4 Prise en compte des conditions aux limites

Elles interviennent juste dans le tableau `DA`, compte-tenu de leur écriture et définition. Elles se calculent *via* la dernière expression de (II.12.3). Pour chaque face `IFAC`, de l'élément de centre $I$, jouxtant le bord, on s'intéresse à :

$$\sum_{k\in\gamma_b(i)}\left[\frac{1}{2}(\ m^n_{b_{ik}}+|\ m^n_{b_{ik}}|\ )\,a_I+\frac{1}{2}(\ m^n_{b_{ik}}-|m^n_{b_{ik}}|)\,a_{\,b_{ik}}\right]-\sum_{k\in\gamma_b(i)}\beta_{b_{ik}}\frac{a_{\,b_{ik}}-a_I}{\overline{I'F}}S_{b_{ik}} \qquad \text{(II.12.10)}$$

avec :

$$a_{\,b_{ik}}=B_{b,ik}\,a_I$$

soit :

$$\left(\sum_{k\in\gamma_b(i)}\left[\frac{1}{2}(\ m^n_{b_{ik}}+|\ m^n_{b_{ik}}|\ )+\frac{1}{2}(\ m^n_{b_{ik}}-|m^n_{b_{ik}}|)B_{b,ik}\right]+\sum_{k\in\gamma_b(i)}\beta_{b_{ik}}\frac{1-\ B_{b,ik}}{\overline{I'F}}S_{b_{ik}}\right)a_I \qquad \text{(II.12.11)}$$

ce qui, pour le terme sur lequel porte la somme, se traduit par :
`ICONVP` $*$ ($-$`FLUJ` $+$ `FLUI` $*$ `COEFBP(IFAC)` $+$ `IDIFFP` $*$ `VISCB(IFAC)` $*$ ( $1-$ `COEFBP(IFAC)`)
avec, $m^n_{b_{ik}}$ représenté par `FLUMAB(IFAC)` , $\frac{1}{2}\ (\ m^n_{b_{ik}}+|\ m^n_{b_{ik}}|\ )$ par `- FLUJ` ,
$\frac{1}{2}\ (\ m^n_{b_{ik}}-|m^n_{b_{ik}}|)B_{b,ik}$ par `FLUI` , $B_{b,ik}$ par `COEFBP(IFAC)`, $\beta_{b_{ik}}\frac{S_{b_{ik}}}{\overline{I'F}}$ par `VISCB(IFAC)`.

### 12.3.5 Décalage du spectre

Lorsqu'il n'existe aucune condition à la limite de type Dirichlet et que `ISTATP` $=0$ (c'est-à-dire pour la pression uniquement), on déplace le spectre de la matrice $\underline{\underline{EM}}_{scal}$ de `EPSI` (*i.e.* on multiplie chaque terme diagonal par (1 + `EPSI`) ) afin de la rendre inversible. `EPSI` est fixé en dur dans `matrix` à $10^{-7}$.

## 12.4 Points à traiter

**• Initialisation**

Le tableau `XA` est initialisé à zéro lorsqu'on veut annuler la contribution du terme en $\frac{\rho\,|\Omega_i|}{\Delta t}$, *i.e.* `ISTATP` $=0$ . Ce qui ne permet donc pas la prise en compte effective des parties diagonales des termes sources à impliciter, décidée par l'utilisateur. Actuellement, ceci ne sert que pour la variable pression et reste donc *a priori* correct, mais cette démarche est à corriger dans l'absolu.

**• Nettoyage**

La contribution `ICONVP FLUI`, dans le calcul du terme `XA(IFAC,1)` lorsque la matrice est symétrique est inutile, car `ICONVP` $=0$.

**• Prise en compte du type de schéma de convection dans $\underline{\underline{EM}}_{scal}$**

Actuellement, les contributions des flux convectifs non reconstruits sont traitées par schéma décentré amont, quelque soit le schéma choisi par l'utilisateur. Ceci peut être handicapant. Par exemple, même sur maillage orthogonal, on est obligé de faire plusieurs sweeps pour obtenir une vitesse prédite correcte. Un schéma centré sans test de pente peut être implanté facilement, mais cette écriture pourrait, dans l'état actuel des connaissances, entraîner des instabilités numériques. Il serait souhaitable d'avoir d'autres schémas tout aussi robustes, mais plus adaptés à certaines configurations.

**• Maillage localement pathologique**

Il peut arriver, notamment au bord, que les mesures algébriques, $\overline{I'J'}$ ou $\overline{I'F}$ soient négatives (en cas de maillages non convexes par exemple). Ceci peut engendrer des problèmes plus ou moins graves : perte de l'existence et l'unicité de la solution (l'opérateur associé n'ayant plus les bonnes propriétés de régularité ou de coercivité), dégradation de la matrice (perte de la positivité) et donc résolution par solveur linéaire associé non approprié (gradient conjugué par exemple).
Une impression permettant de signaler et de localiser le problème serait souhaitable.

# 13- Sous-programme navsto

On s'intéresse à la résolution du système d'équations de Navier-Stokes tridimensionnelles monophasiques, à une pression, instationnaires, en incompressible ou faiblement dilatable, basées sur une discrétisation temporelle de type Euler implicite d'ordre 1 ou Crank-Nicolson d'ordre 2 et sur une discrétisation spatiale par volumes finis colocalisée.

## 13.1 Fonction

Dans ce sous-programme sont calculées, à un pas de temps donné, les variables vitesse et pression de ce problème en procédant en deux étapes issues d'une décomposition des opérateurs (méthode à pas fractionnaires).
Les variables sont donc supposées connues à l'instant $t^n$ et on cherche à les déterminer à l'instant[1] $t^{n+1}$. Soit $\Delta t^n = t^{n+1} - t^n$ le pas de temps associé. Dans un premier temps, on réalise l'étape de prédiction de la vitesse en résolvant l'équation de quantité de mouvement avec une pression explicite. Suit l'étape de correction de la pression (ou projection de la vitesse) qui permet d'obtenir un champ de vitesse à divergence nulle.

Les équations en continu sont donc :

$$\left\{ \begin{array}{l} \dfrac{\partial}{\partial t}(\rho \underline{u}) + \mathrm{div}(\rho\, \underline{u} \otimes \underline{u}) = \mathrm{div}(\underline{\underline{\sigma}}) + \underline{TS} - \underline{\underline{K}}\, \underline{u} \\ \mathrm{div}(\rho \underline{u}) = \Gamma \end{array} \right. \tag{II.13.1}$$

avec $\rho$ la masse volumique, $\underline{u}$ le champ de vitesse, $[\,\underline{TS} - \underline{\underline{K}}\, \underline{u}\,]$ les autres termes sources ($\underline{\underline{K}}$ est un tenseur diagonal positif par définition), $\underline{\underline{\sigma}}$ le tenseur de contraintes, $\underline{\underline{\tau}}$ le tenseur des contraintes visqueuses, $\mu$ la viscosité dynamique (moléculaire et éventuellement turbulente), $\kappa$ la viscosité de volume (usuellement nulle et négligée dans le code et dans la suite du document, sauf en compressible), $\underline{\underline{D}}$ le tenseur taux de déformation[2], $\Gamma$ le terme source de masse.

$$\left\{ \begin{array}{l} \underline{\underline{\sigma}} = \underline{\underline{\tau}} - P\underline{\underline{Id}} \\ \underline{\underline{\tau}} = 2\,\mu\, \underline{\underline{D}} + \ (\kappa \ - \frac{2}{3}\mu)\ tr(\underline{\underline{D}})\ \underline{\underline{Id}} \\ \underline{\underline{D}} = \frac{1}{2}(\underline{\underline{\mathrm{grad}}}\ \underline{u} + {}^t\underline{\underline{\mathrm{grad}}}\ \underline{u}) \end{array} \right. \tag{II.13.2}$$

On rappelle la définition des notations employées[3] :

$$\left\{ \begin{array}{lcl} \left[\underline{\underline{\mathrm{grad}}}\ \underline{a}\right]_{ij} & = & \partial_j a_i \\ \left[\mathrm{div}(\underline{\underline{\sigma}})\right]_i & = & \partial_j \sigma_{ij} \\ \left[\underline{a} \otimes \underline{b}\right]_{ij} & = & a_i\, b_j \end{array} \right.$$

et donc :

$$\left[\mathrm{div}(\underline{a} \otimes \underline{b})\right]_i \ = \ \partial_j(a_i\, b_j)$$

Remarque
Dans le cas de la prise en compte d'une masse volumique variable, l'équation de continuité s'écrit :

$$\frac{\partial \rho}{\partial t} + \mathrm{div}\,(\rho\, \underline{u}) = \Gamma$$

[1] La pression est supposée connue à l'instant $t^{n-1+\theta}$ et recherchée en $t^{n+\theta}$, avec $\theta = 1$ ou $1/2$ suivant le schéma en temps considéré.

[2] À ne pas confondre, malgré la même notation $D$, avec les flux diffusifs $\underline{D}_{ij}$ et $\underline{D}_{b_{ik}}$ décrits par la suite dans ce sous-programme.

[3] en utilisant la convention de sommation d'Einstein.

Cette équation n'est pas prise en compte dans l'étape de projection (on continue à résoudre seulement $\mathrm{div}(\rho\,\underline{u}) = \Gamma$) alors que le terme $\dfrac{\partial \rho}{\partial t}$ apparaît lors de l'étape de prédiction de la vitesse dans le sous-programme `preduv`. Si ce terme joue un rôle sensible, l'algorithme compressible de *Code_Saturne* (qui résout l'équation complète) est alors sans doute plus adapté.

# 13.2 Discrétisation

On pose :

$$\alpha_{ij} = \frac{\overline{FJ'}}{\overline{I'J'}} \text{ défini aux faces internes uniquement et}$$

$$\underline{u}_{K'} = \underline{u}_K + (\underline{\underline{\mathrm{grad}}}\ \underline{u})_K . \underline{KK'} \text{ à l'ordre 1 en espace, pour } K = I \text{ ou } J$$

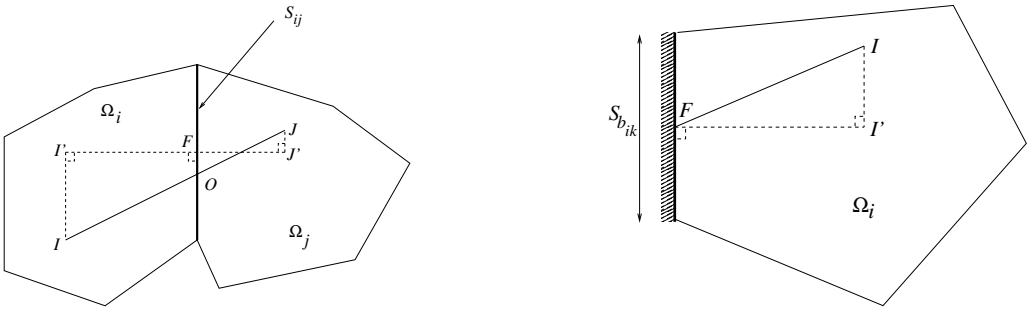



Figure II.13.1: Définition des différentes entités géométriques pour les faces internes (gauche) et de bord (droite).

## 13.2.1 Méthode à pas fractionnaires

### Introduction

Une des méthodes permettant de résoudre numériquement les équations de Navier-Stokes est de décomposer les opérateurs s'y rattachant en opérateurs moins complexes (qui peuvent être traités à l'aide d'algorithmes efficaces), moyennant des sous-pas intermédiaires dans un même pas de temps. Ici, deux sous-pas sont réalisés : le premier reprend les parties convective, diffusive et termes sources de l'équation de quantité de mouvement et constitue l'étape dite de prédiction de la vitesse, le second traite l'équation de continuité et est désigné comme l'étape de correction de pression ou de projection de la vitesse.

### Étape de prédiction des vitesses

La discrétisation en temps se fait en appliquant à la variable résolue un $\theta$-schéma au temps $n+\theta$ s'inspirant de la démarche utilisée pour l'équation de transport d'un scalaire[4].
La vitesse au temps $n+1$ n'étant disponible qu'après l'étape de projection, c'est ici une vitesse prédite au temps $n+1$ que l'on utilise pour interpoler :

$$\widetilde{\underline{u}}^{n+\theta} = \theta\,\widetilde{\underline{u}}^{n+1} + (1-\theta)\,\underline{u}^n \tag{II.13.3}$$

Avec[5] :

$$\begin{cases} \theta = 1 & \text{Pour un schéma de type Euler implicite d'ordre 1.} \\ \theta = 1/2 & \text{Pour un schéma de type Cranck-Nicolson d'ordre 2.} \end{cases} \tag{II.13.4}$$

[4] cf. `covofi`
[5] Dans le cas où $\theta = 1/2$, le pas de temps $\Delta t$ est supposé uniforme en temps et en espace.

Le champ de vitesse $\underline{\widetilde{u}}^{n+1}$ prédit est alors obtenu par :
• Linéarisation partielle de l'opérateur de convection engendrant un découplage des composantes de la vitesse.
• Explicitation de la pression.
• Explicitation ou extrapolation des grandeurs physiques (*i.e* $\rho$, $\mu$, $Cp$ ...) et du flux de masse.
• Explicitation ou extrapolation des termes sources explicites au temps $n+\theta_S$ tels que les contributions du gradient transposé, de la viscosité secondaire, de la partie extradiagonale des pertes de charges, de la masse injectée $\Gamma\,\underline{u_i}$, ...
• Les termes sources implicites linéaires par rapport à la vitesse (termes sources utilisateurs implicite, partie diagonale des pertes de charge $\underline{\underline{K_d}}\,\underline{u}$, sources de masse $\Gamma\,\underline{u}$, etc...) sont supposés pris au temps $n+\theta$ et sont rattachés au tenseur $\underline{\underline{K}}$[6].

Par souci de clarté, on suppose qu'en l'absence d'indication, les propriétes physiques $\Phi = \rho$, $\mu$ ... et le flux de masse $(\rho\underline{u})$ pris respectivement aux instants $n+\theta_\Phi$ et $n+\theta_F$, où $\theta_\Phi$ et $\theta_F$ dépendent des schémas en temps spécifiquement utilisés pour ces grandeurs[7].
On résout donc le système suivant, après réécriture des termes instationnaires à l'aide de la conservation de la masse :

$$\rho\left(\frac{\underline{\widetilde{u}}^{n+1}-\underline{u}^{n}}{\Delta t}\right)+\mathrm{div}\left(\underline{\widetilde{u}}^{n+\theta}\otimes(\rho\underline{u})\right)=\mathrm{div}(\underline{\underline{\sigma}}^{n+\theta})+\underline{TS}^{n+\theta_S}-\underline{\underline{K}}^{n}\,\underline{u}^{n+\theta}+\underline{\widetilde{u}}^{n+\theta}\,\mathrm{div}(\rho\underline{u}) \tag{II.13.5}$$

avec :

$$\underline{\underline{\sigma}}^{n+\theta}=\mu\;\underline{\underline{\mathrm{grad}}}\;\underline{\widetilde{u}}^{n+\theta}\underbrace{-P^{n-1+\theta}\,\underline{\underline{Id}}+(\,\mu\;{}^{t}\underline{\underline{\mathrm{grad}}}\;\underline{u})^{n+\theta_S}-\frac{2}{3}\,(\,\mu\,\mathrm{div}\,\underline{u})^{n+\theta_S}\,\underline{\underline{Id}}}_{\text{termes sources explicites}} \tag{II.13.6}$$

### Étape de correction de la pression (ou projection des vitesses)

Le champ de vitesse prédit est *a priori* à divergence non nulle. La seconde étape corrige la pression en imposant la nullité de la contrainte stationnaire pour la vitesse prise à l'instant $t^{n+1}$. On résout donc :

$$\begin{cases}\dfrac{(\rho\underline{u})^{n+1}-(\rho\underline{\widetilde{u}})^{n+1}}{\Delta t}=-\underline{\mathrm{grad}}\;\delta P^{n+\theta}\\ \mathrm{div}(\rho\underline{u})^{n+1}=\Gamma\end{cases} \tag{II.13.7}$$

où l'incrément de pression $\delta P^{n+\theta}$ vaut :

$$\delta P^{n+\theta}=P^{n+\theta}-P^{n-1+\theta} \tag{II.13.8}$$

Remarque

Les quantités $\rho$ et $\mu$ sont constantes lors de ces deux étapes. Leur variation éventuelle est effectuée au début du pas de temps suivant, après réactualisation des scalaires (température, fraction massique,...).

## 13.2.2 Discrétisation spatiale

On intègre classiquement sur les volumes de contrôle $\Omega_i$ (ou cellules) les équations discrétisées en temps.

[6] En réalité, les composantes de la vitesse étant découplées, seuls les termes linéaires par rapport à la composante résolue sont factorisés de la sorte. Les autres termes étant traités comme des termes explicites. On pourra se rapporter à `preduv` pour plus de détail.
[7] cf. `introd`

## Étape de prédiction des vitesses

**Second membre**

Si l'on ne tient pas compte des termes de convection et de diffusion issus du $\theta$-schéma, les termes sources volumiques explicites de l'équation (II.13.5) s'écrivent pour le système portant sur la quantité $(\widetilde{\underline{u}}^{n+1} - \underline{u}^n)$ :

$$-\underline{\text{grad}}\ P^{n-1+\theta} + \text{div}\left[(\mu\ {}^t\underline{\underline{\text{grad}}}\ \underline{u})^{n+\theta_S} - \frac{2}{3}(\mu\,\text{div}\,\underline{u})^{n+\theta_S}\underline{\underline{Id}}\right] + \underline{TS}^{n+\theta_S} - \underline{\underline{K}}^n\underline{u}^n + \underline{u}^n\,\text{div}(\rho\underline{u})$$

Pour intégrer ces termes sur une cellule $\Omega_i$, on multiplie leur valeur locale au centre par le volume de la cellule.

**Convection**

Après décomposition de $\widetilde{\underline{u}}$ à l'aide de la relation (II.13.3), l'intégration spatiale des parties convectives $\theta\,\text{div}\left(\widetilde{\underline{u}}^{n+1}\otimes(\rho\underline{u})\right)$ et $(1-\theta)\,\text{div}\,(\underline{u}^n\otimes(\rho\underline{u}))$ conduit à une somme de flux numériques $\underline{F}_{\,ij}$ calculés aux faces des cellules purement internes et de flux numériques $\underline{F}_{\,b_{ik}}$ calculés aux faces de bord du domaine $\Omega$. Soient $Vois(i)$ l'ensemble des centres des cellules voisines de $\Omega_i$ et $\gamma_b(i)$ l'ensemble des centres des faces de bord de $\Omega_i$, on a :

$$\int_{\Omega_i}\text{div}\,(\underline{u}\otimes(\rho\underline{u}))\ d\Omega = \sum_{j\in Vois(i)}\underline{F}_{\,ij}((\rho\underline{u}),\underline{u}) + \sum_{k\in\gamma_b(i)}\underline{F}_{\,b_{ik}}((\rho\underline{u}),\underline{u})$$

en posant :

$$\underline{F}_{\,ij}((\rho\underline{u}),\underline{u}) = \left[(\rho\underline{u})_{\,ij}.\underline{S}_{\,ij}\right]\ \underline{u}_{\,f,ij} \tag{II.13.9}$$

$$\underline{F}_{\,b_{ik}}((\rho\underline{u}),\underline{u}) = \left[(\rho\underline{u})_{\,b_{ik}}.\underline{S}_{\,b_{ik}}\right]\ \underline{u}_{\,f\,b_{ik}} \tag{II.13.10}$$

La valeur de $\underline{F}_{\,ij}$ dépend du type de schéma numérique choisi. Il en existe trois dans *Code_Saturne* :

- un schéma décentré amont d'ordre 1 (upwind) :

  $\underline{F}_{\,ij}((\rho\underline{u}),\underline{u}) = \underline{F}_{ij}^{amont}((\rho\underline{u}),\underline{u})$

  où : $\underline{u}_{\,f,ij} = \underline{u}_{\,I}$ si $(\rho\underline{u})_{\,ij}.\underline{S}_{\,ij} \geqslant 0$

  $\underline{u}_{\,f,ij} = \underline{u}_{\,J}$ si $(\rho\underline{u})_{\,ij}.\underline{S}_{\,ij} < 0$ ,
- un schéma centré d'ordre 2:

  $\underline{F}_{\,ij}((\rho\underline{u}),\underline{u}) = \underline{F}_{ij}^{centré}((\rho\underline{u}),\underline{u})$

  avec : $\underline{u}_{\,f,ij} = \alpha_{ij}\underline{u}_{\,I'} + (1-\alpha_{ij})\underline{u}_{\,J'}$ et $\underline{u}_{K'} = \underline{u}_K + (\underline{\underline{\text{grad}}}\ \underline{u})_K.\underline{KK'}$ pour $K = I$ ou $J$
- un schéma décentré amont d'ordre 2 SOLU (Second Order Linear Upwind) :

  $\underline{F}_{\,ij}((\rho\underline{u}),\underline{u}) = \underline{F}_{\,ij}^{\,SOLU}((\rho\underline{u}),\underline{u})$

  avec : $\underline{u}_{\,f,ij} = \begin{array}{ll}\underline{u}_{\,I} + \underline{IF}.(\underline{\underline{\text{grad}}}\ \underline{u})_{\,I} & \text{si } (\rho\underline{u})_{\,ij}.\underline{S}_{\,ij} \geqslant 0\\ \underline{u}_{\,J} + \underline{JF}.(\underline{\underline{\text{grad}}}\ \underline{u})_J & \text{si } (\rho\underline{u})_{\,ij}.\underline{S}_{\,ij} < 0.\end{array}$

La valeur de $\underline{F}_{\,b_{ik}}$ est calculée avec :

$$\begin{array}{ll}\underline{u}_{f\,b_{ik}} & = \underline{u}_{\,I} \text{ si } (\rho\underline{u})_{\,b_{ik}}.\underline{S}_{\,b_{ik}} \geqslant 0\\ & = \underline{u}_{\,b_{ik}} \text{ si } (\rho\underline{u})_{\,b_{ik}}.\underline{S}_{\,b_{ik}} < 0\end{array} \tag{II.13.11}$$

avec $\underline{u}_{\,b_{ik}}$ valeur au bord donnée directement par les conditions aux limites.

Remarque 2

En centré, on écrit en réalité (égalité conservant l'ordre un en espace) :

$$\underline{u}_{\,f,ij} = \alpha_{ij}\underline{u}_{\,I} + (1-\alpha_{ij})\underline{u}_{\,J} + \frac{1}{2}\left[(\underline{\underline{\text{grad}}}\ \underline{u})_I + (\underline{\underline{\text{grad}}}\ \underline{u})_J\right].\underline{OF}$$

pour des raisons de stabilité purement numérique.
REMARQUE 3

Un test de pente (qui peut introduire des non linéarités dans l'opérateur de convection) permet de basculer entre un schéma d'ordre deux et le schéma décentré amont d'ordre un. De plus, en mode standard, on utilise en tout point une valeur de $\widetilde{\underline{u}}_{f,ij}$ issue d'une moyenne barycentrique entre la valeur décentrée amont et la valeur d'ordre 2 (blending, spécifié par l'utilisateur).

**Diffusion**

De même, les parties diffusives $\theta$ div$(\mu \underline{\underline{\text{grad}}}\ \widetilde{\underline{u}}^{n+1})$ et $(1-\theta)$ div$(\mu \underline{\underline{\text{grad}}}\ \underline{u}^n)$ s'écrivent :

$$\int_{\Omega_i} \text{div}(\mu \underline{\underline{\text{grad}}}\ \underline{u}) d\Omega = \sum_{j \in Vois(i)} \underline{D}_{ij}(\mu, \underline{u}) + \sum_{k \in \gamma_b(i)} \underline{D}_{b_{ik}}(\mu, \underline{u})$$

avec :

$$\underline{D}_{ij}(\mu, \underline{u}) = \mu_{ij} \frac{\underline{u}_{J'} - \underline{u}_{I'}}{\overline{I'J'}} S_{ij} \tag{II.13.12}$$

et :

$$\underline{D}_{b_{ik}}(\mu, \underline{u}) = \mu_{b_{ik}} \frac{\underline{u}_{b_{ik}} - \underline{u}_{I'}}{\overline{I'F}} S_{b_{ik}} \tag{II.13.13}$$

en conservant les notations précédentes et avec $\underline{u}_{b_{ik}}$ la valeur au bord donnée directement par les conditions aux limites.
La viscosité $\mu_{ij}$ à la face est calculée à l'aide des valeurs aux centres selon une fonction $f$ donnée :

$$\mu_{ij} = f(\mu_I, \mu_J)$$

qui est, soit une moyenne arithmétique :

$$f(\mu_I, \mu_J) = \frac{1}{2}(\mu_I + \mu_J) \tag{II.13.14}$$

soit une moyenne géométrique :

$$f(\mu_I, \mu_J) = \frac{\mu_I \mu_J}{\alpha_{ij}\mu_I + (1-\alpha_{ij})\mu_J} \tag{II.13.15}$$

et la viscosité $\mu_{b_{ik}}$ est égale à :

$$\mu_{b_{ik}} = \mu_I \tag{II.13.16}$$

On introduit en outre, pour une utilisation ultérieure, les notations suivantes :

$$\underline{D}_{ij}^{NRec}(\mu, \underline{u}) = \mu_{ij} \frac{\underline{u}_J - \underline{u}_I}{\overline{I'J'}} S_{ij} \tag{II.13.17}$$

$$\underline{D}_{b_{ik}}^{NRec}(\mu, \underline{u}) = \mu_{b_{ik}} \frac{\underline{u}_{b_{ik}} - \underline{u}_I}{\overline{I'F}} S_{b_{ik}} \tag{II.13.18}$$

qui correspondent chacune à une valeur non reconstruite aux faces internes et de bord.

**Résolution**

Le système (II.13.5) pouvant comporter des non linéarités dues au recours au test de pente ou pouvant conduire *via* la reconstruction du gradient (cellule) à une matrice quasiment pleine en présence de non orthogonalités, on le résout de manière itérative avec la suite $(\widetilde{\underline{u}}^{n+1,k})_{k \in \mathbb{N}}$ définie par :

$$\left\{ \begin{array}{l} \widetilde{\underline{u}}^{n+1,0} = \underline{u}^n \\ \widetilde{\underline{u}}^{n+1,k+1} = \widetilde{\underline{u}}^{n+1,k} + \delta\widetilde{\underline{u}}^{k+1} \\ \mathcal{EM}(\delta\widetilde{\underline{u}}^{k+1}, I) = -\mathcal{E}(\widetilde{\underline{u}}^{n+1,k}, I) \end{array} \right. \tag{II.13.19}$$

ce qui définit également la suite $(\delta\widetilde{\underline{u}}^{k+1})_{k\in\mathbb{N}}$.
Les deux opérateurs $\mathcal{E}$ et $\mathcal{EM}$ ont pour expression respective :

$$\begin{aligned}\mathcal{E}(\widetilde{\underline{u}},I) &= \theta\,\mathcal{J}(\widetilde{\underline{u}},I) + (1-\theta)\,\mathcal{J}(\underline{u}^n,I) + |\Omega_I|\frac{\rho_I}{\Delta t}(\widetilde{\underline{u}}_I - \underline{u}_I^n)\\ &+ |\Omega_I|\left[(\underline{TS})_I^{n+\theta_S} - (\underline{\mathrm{grad}}\ P)_I^{n-1+\theta}\right]\\ &+ \sum_{j\in Vois(i)} \underbrace{\left((\mu\ {}^t\underline{\underline{\mathrm{grad}}}\ \underline{u})^{n+\theta_S} - \frac{2}{3}(\mu\,\mathrm{div}\,\underline{u})^{n+\theta_S}\underline{\underline{Id}}\right)_{ij}}_{\text{moyenne ou interpolation linéaire entre I' et J'}} \cdot \underline{S}_{ij}\\ &+ \sum_{k\in\gamma_b(i)} \underbrace{\left((\mu\ {}^t\underline{\underline{\mathrm{grad}}}\ \underline{u})^{n+\theta_S} - \frac{2}{3}(\mu\,\mathrm{div}\,\underline{u})^{n+\theta_S}\underline{\underline{Id}}\right)_{b_{ik}}}_{\text{issu des conditions aux limites}} \cdot \underline{S}_{b_{ik}}\end{aligned}$$

avec :

$$\begin{aligned}\mathcal{J}(\underline{v},I) &= |\Omega_I|\ [\underline{\underline{K}}^n - \mathrm{div}(\rho\underline{u})\,]_I\ \underline{v}_I\\ &+ \left(\sum_{j\in Vois(i)} \underline{F}_{ij}((\rho\underline{u}),\underline{v}) + \sum_{k\in\gamma_b(i)} \underline{F}_{b_{ik}}((\rho\underline{u}),\underline{v})\right)\\ &- \left(\sum_{j\in Vois(i)} \underline{D}_{ij}(\mu,\underline{v}) + \sum_{k\in\gamma_b(i)} \underline{D}_{b_{ik}}(\mu,\underline{v})\right)\end{aligned}$$

$$\begin{aligned}\mathcal{EM}(\delta\underline{u},I) &= |\Omega_I|\left(\frac{\rho_I}{\Delta t} + \theta\,[\underline{\underline{K}}^n - \mathrm{div}(\rho\underline{u})\,]_I\right)\ \delta\underline{u}_I\\ &+\theta\left(\sum_{j\in Vois(i)} \underline{F}_{ij}^{amont}((\rho\underline{u}),\delta\underline{u}) + \sum_{k\in\gamma_b(i)} \underline{F}_{b_{ik}}^{amont}((\rho\underline{u}),\delta\underline{u})\right)\\ &-\theta\left(\sum_{j\in Vois(i)} \underline{D}_{ij}^{NRec}(\mu,\delta\underline{u}) + \sum_{k\in\gamma_b(i)} \underline{D}_{b_{ik}}^{NRec}(\mu,\delta\underline{u})\right)\end{aligned}$$

De plus, on suppose que cette suite $(\widetilde{\underline{u}}^{n+1,k})_{k\in\mathbb{N}}$ converge vers $\widetilde{\underline{u}}^{n+1}$.

Remarque 4
Les conditions aux limites associées aux opérateurs $\mathcal{E}$ et $\mathcal{EM}$ du système (II.13.20) sont celles portant sur la vitesse $\underline{u}$. Elles sont de type Dirichlet homogène ou de type Neumann homogène sur $\delta\underline{u}$ si $\underline{u}$ a une condition de type Dirichlet ou de type Neumann respectivement. Elles sont mixtes dans le cas d'une condition de symétrie sur une face en biais par rapport aux axes.

Remarque 5
Les deux premières sommes de type $(\sum_{k\in\gamma_b(i)})$, *i.e.* comportant les termes en $\underline{F}_{b_{ik}}((\rho\underline{u}),\underline{u})$ et $\underline{D}_{b_{ik}}(\mu,\underline{u})$, utilisent les conditions aux limites de la vitesse.
Le terme volumique :

$$\sum_{j\in Vois(i)} \underbrace{\left((\mu\ {}^t\underline{\underline{\mathrm{grad}}}\ \underline{u})^{n+\theta_S} - \frac{2}{3}(\mu\,\mathrm{div}\,\underline{u})^{n+\theta_S}\underline{\underline{Id}}\right)_{ij}}_{\text{moyenne ou interpolation linéaire entre I' et J'}} \cdot \underline{S}_{ij}$$

de terme de bord associé :

$$\sum_{k\in\gamma_b(i)} \underbrace{\left((\mu\ {}^t\underline{\underline{\mathrm{grad}}}\ \underline{u})^{n+\theta_S} - \frac{2}{3}(\mu\,\mathrm{div}\,\underline{u})^{n+\theta_S}\underline{\underline{Id}}\right)_{b_{ik}}}_{\text{issu des conditions aux limites}} \cdot \underline{S}_{b_{ik}}$$

a un traitement particulier. En effet, pour une cellule $\Omega_i$ jouxtant le bord, la contribution du premier terme (en gradient transposé) est annulée, aucune condition à la limite correcte ne lui étant attribuée pour le moment.

REMARQUE 6
L'opérateur $\mathcal{EM}$ approche $\mathcal{E}$ (aucun terme n'est reconstruit et la partie convective est traitée systématiquement en schéma décentré amont). Ceci peut générer des imprécisions numériques non négligeables si la suite $(\widetilde{\underline{u}}^{n+1,k})_{k\in\mathbb{N}}$ n'a pas convergé.

## Étape de correction de la pression

En prenant la divergence de la première équation du système (II.13.7), on obtient :

$$\mathrm{div}\left[(\rho\underline{u})^{n+1} - (\rho\widetilde{\underline{u}})^{n+1}\right] = \mathrm{div}(-\Delta t\,\underline{\mathrm{grad}}\ \delta P^{n+\theta}) \tag{II.13.20}$$

En utilisant la contrainte stationnaire $\mathrm{div}(\rho\underline{u})^{n+1} = \Gamma$, on a donc :

$$\mathrm{div}(\Delta t\,\underline{\mathrm{grad}}\ \ \delta P^{n+\theta}) = \mathrm{div}((\rho\widetilde{\underline{u}})^{n+1}) - \Gamma \tag{II.13.21}$$

soit :

$$\left\{\begin{array}{l} \mathrm{div}(\Delta t\,\underline{\mathrm{grad}}\ \ \delta P^{n+\theta}) = \mathrm{div}((\rho\widetilde{\underline{u}})^{n+1}) - \Gamma \\ (\rho\underline{u})^{n+1} = (\rho\widetilde{\underline{u}})^{n+1} - \Delta t\,\underline{\mathrm{grad}}\ \delta P^{n+\theta} \end{array}\right. \tag{II.13.22}$$

et :

$$\underline{u}^{n+1} = \widetilde{\underline{u}}^{n+1} - \frac{\Delta t}{\rho}\,\underline{\mathrm{grad}}\ \delta P^{n+\theta} \tag{II.13.23}$$

En intégrant sur une cellule :

$$\int_{\Omega_i} \mathrm{div}(\Delta t\underline{\mathrm{grad}}\ \ \delta P^{n+\theta})\,d\Omega = \sum_{j\in Vois(i)} \underline{D}_{ij}(\Delta t, \delta P^{n+\theta}) + \sum_{k\in\gamma_b(i)} \underline{D}_{b_{ik}}(\Delta t, \delta P^{n+\theta}) \tag{II.13.24}$$

et :

$$\int_{\Omega_i} \mathrm{div}(\rho\widetilde{\underline{u}})^{n+1} d\Omega = \sum_{j\in Vois(i)} \left[(\rho\widetilde{\underline{u}})_{ij}^{n+1}.\underline{S}_{ij}\right] + \sum_{k\in\gamma_b(i)} \left[(\rho\widetilde{\underline{u}})_{b_{ik}}^{n+1}.\underline{S}_{b_{ik}}\right] \tag{II.13.25}$$

On utilise le même formalisme que précédemment pour l'intégration du terme de diffusion de l'étape de prédiction. Les conditions aux limites sont de type Dirichlet homogène ou de type Neumann homogène sur $\delta P$ si $P$ a une condition de type Dirichlet ou de type Neumann respectivement.

### Calcul du second membre de l'équation portant sur l'incrément de pression.

La discrétisation de $\sum_{j\in Vois(i)} \left[(\rho\widetilde{\underline{u}})_{ij}^{n+1}.\underline{S}_{ij}\right] + \sum_{k\in\gamma_b(i)} \left[(\rho\widetilde{\underline{u}})_{b_{ik}}^{n+1}.\underline{S}_{b_{ik}}\right]$ est particulière. Le choix suivant noté $[\ ]^{Init}$, pour une cellule ne touchant pas le bord par exemple, est insatisfaisant avec la discrétisation et le schéma utilisés ici, en particulier avec l'équation (II.13.23) :

$$(\rho\widetilde{\underline{u}})_{ij}^{n+1}.\underline{S}_{ij} = \left[(\rho\widetilde{\underline{u}})_{ij}^{n+1}.\underline{S}_{ij}\right]^{Init} = \left[\alpha_{ij}(\rho\widetilde{\underline{u}})_{I'}^{n+1} + (1-\alpha_{ij})(\rho\widetilde{\underline{u}})_{J'}^{n+1}\right].\underline{S}_{ij} \tag{II.13.26}$$

Tout comme pour le calcul du flux numérique en centré, on utilise de façon licite l'approximation suivante :

$$\begin{array}{rl} (\rho\widetilde{\underline{u}})_{ij}^{n+1} & = \alpha_{ij}\rho_I\widetilde{\underline{u}}_I^{n+1} + (1-\alpha_{ij})\rho_J\widetilde{\underline{u}}_J^{n+1} \\ & + \dfrac{1}{2}\left[(\underline{\underline{\mathrm{grad}}}\ (\rho\widetilde{\underline{u}})^{n+1})_I + (\underline{\underline{\mathrm{grad}}}\ (\rho\widetilde{\underline{u}})^{n+1})_J\right].\underline{OF} \end{array} \tag{II.13.27}$$

mais ce n'est pas elle qui pose problème.

En fait, $\left[(\rho\widetilde{\underline{u}})_{ij}^{n+1}\right]^{Init}$ contient le terme $\underline{G}_{cel,ij}^{n}$, hérité de l'étape de prédiction, qui vaut :

$$\underline{G}_{cel,ij}^{n-1+\theta} = \alpha_{ij}\,\underline{\text{grad}}\ P_{I'}^{n-1+\theta} + (1-\alpha_{ij})\,\underline{\text{grad}}\ P_{J'}^{n-1+\theta}$$

Or, sur un maillage orthogonal régulier cartésien, à partir d'une vitesse $\widetilde{\underline{u}}^{n+1} = \underline{0}$ et d'une pression $P_I^{n-1+\theta} = (-1)^I$, on obtient $\underline{G}_{cel,ij}^{n-1+\theta} = 0$ d'où $\delta P^{n+\theta} = 0$ : l'irrégularité initiale de pression ne peut donc jamais être corrigée.

Pour remédier à cela, on modifie l'écriture $[\ ]^{Init}$ de $(\rho\widetilde{\underline{u}})_{ij}^{n+1}$ et de $(\rho\widetilde{\underline{u}})_{b_{ik}}^{n+1}$ en adoptant la valeur $[\ ]^{Corr}$ :

$$\begin{aligned}(\rho\widetilde{\underline{u}})_{ij}^{n+1}\cdot\underline{S}_{ij} &= \left[(\rho\widetilde{\underline{u}})_{ij}^{n+1}\right]^{Corr}\cdot\underline{S}_{ij}\\ &= \Big(\left[(\rho\underline{u})^{n+1} - \beta(-\Delta t\underline{\text{grad}}\ P^{n-1+\theta})\right]_{ij}^{Init} + \beta\,(-\underline{D}_{ij}(\Delta t, P^{n-1+\theta}))\Big)\cdot\underline{S}_{ij}\end{aligned} \tag{II.13.28}$$

et, pour les conditions aux limites d'entrée, de symétrie (quelconque) ou de paroi :

$$(\rho\widetilde{\underline{u}})_{b_{ik}}^{n+1}\cdot\underline{S}_{b_{ik}} = \left[(\rho\widetilde{\underline{u}})_{b_{ik}}^{n+1}\right]^{Corr}\cdot\underline{S}_{b_{ik}} = \rho_{b_{ik}}\underline{u}_{b_{ik}}^{n+1}\cdot\underline{S}_{b_{ik}}$$

pour les conditions aux limites de sortie :

$$\begin{aligned}(\rho\widetilde{\underline{u}})_{b_{ik}}^{n+1}\cdot\underline{S}_{b_{ik}} &= \left[(\rho\widetilde{\underline{u}})_{b_{ik}}^{n+1}\right]^{Corr}\cdot\underline{S}_{b_{ik}}\\ &= \left(\left[\rho_{b_{ik}}\underline{u}_{b_{ik}}^{n+1} - \beta(-\Delta t\underline{\text{grad}}\ P^{n-1+\theta})_{I'}\right] + \beta\,(-\underline{D}_{b_{ik}}(\Delta t, P^{n-1+\theta}))\right)\cdot\underline{S}_{b_{ik}}\end{aligned}$$

$\beta$ est appelé coefficient d'Arakawa. Lorsqu'il vaut 1 (valeur par défaut), il s'agit du filtre Rhie & Chow.

Remarque 7
On peut généraliser cette démarche à d'autres termes sources du même type, par exemple pour le modèle $R_{ij} - \varepsilon$.

**Résolution**

On construit une suite $(\delta P^{n+1,k})_{k\in\mathbb{N}}$ pour résoudre l'équation (II.13.24), qui peut conduire *via* la reconstruction du gradient cellule à une matrice quasiment pleine en présence de non orthogonalités, définie par :

$$\left\{\begin{array}{l}\delta P^{n+\theta,0} = 0\\ \delta P^{n+\theta,k+1} = \delta P^{n+\theta,k} + C_{relax}\delta(\delta P)^{n+\theta,k+1}\\ \mathcal{FM}(\delta(\delta P)^{n+\theta,k+1}, I) = \mathcal{F}(\delta P^{n+\theta,k}, I)\end{array}\right. \tag{II.13.29}$$

ce qui définit également la suite $(\delta(\delta P)^{n+\theta,k+1})_{k\in\mathbb{N}}$.
Les opérateurs $\mathcal{F}$ et $\mathcal{FM}$ ont pour expression :

$$\begin{aligned}\mathcal{F}(\delta P, I) &= \sum_{j\in Vois(i)}\left[\underline{D}_{ij}(\Delta t, \delta P) - \left[(\rho\widetilde{\underline{u}})_{ij}^{n+1}\right]^{Corr}\right]\\ &\quad + \sum_{k\in\gamma_b(i)}\left[\underline{D}_{b_{ik}}(\Delta t^n, \delta P) - \left[(\rho\widetilde{\underline{u}})_{b_{ik}}^{n+1}\right]^{Corr}\right] + \Gamma\end{aligned} \tag{II.13.30}$$

et :

$$\mathcal{FM}(\delta(\delta P), I) = \sum_{j\in Vois(i)}\left[-\underline{D}_{ij}^{NRec}(\Delta t, \delta(\delta P))\right] + \sum_{k\in\gamma_b(i)}\left[-\underline{D}_{b_{ik}}^{NRec}(\Delta t, \delta(\delta P))\right] \tag{II.13.31}$$

respectivement.
$C_{relax}$ est un coefficient de relaxation donné et fixé à 1 en standard. On suppose que la suite $(\delta P^{n+\theta,k})_{k\in\mathbb{N}}$ converge vers $\delta P^{n+\theta}$.

Au fur et à mesure des itérations, le flux de masse est mis à jour, en utilisant $\delta(\delta P)$. À convergence, le flux de masse actualisé obtenu est :

$$(\rho \underline{u})_{ij}^{n+1}.\underline{S}_{ij} = \left[(\rho \widetilde{\underline{u}})_{ij}^{n+1}\right]^{Corr}.\underline{S}_{ij} - \underline{D}_{ij}(\Delta t^n, \delta P^{n+\theta}) \tag{II.13.32}$$

et on calcule le nouveau champ de vitesse au centre des cellules grâce à l'égalité :

$$\underline{u}^{n+1} = \widetilde{\underline{u}}^{n+1} - \frac{\Delta t}{\rho}\underline{\text{grad}}\ \ \delta P^{n+\theta} \tag{II.13.33}$$

Remarque 8

Un traitement spécifique permet d'assurer que la conservation de la masse portant sur le bilan des flux de masse aux faces est toujours parfaitement vérifiée à l'issue de l'étape de correction, que la suite $(\delta P^{n+\theta,k+1})_{k\in\mathbb{N}}$ ait ou non atteint la convergence. En effet, $\delta P^{n+\theta,k_{fin}+1}$, dernier terme évalué de la suite, est donné par :

$$\mathcal{FM}(\delta(\delta P)^{n+\theta,k_{fin}+1}, I) = \mathcal{F}(\delta P^{n+\theta,k_{fin}}, I)$$

Au lieu de réactualiser classiquement le flux de masse, on le calcule comme suit :

$$(\rho \underline{u})_{ij}^{n+1}.\ \underline{S}_{ij} = \left[(\rho \widetilde{\underline{u}})_{ij}^{n+1}\right]^{Corr}.\underline{S}_{ij} - \underline{D}_{ij}(\Delta t, \delta P^{n+\theta,k_{fin}}) - \underline{D}_{ij}^{NRec}(\Delta t, \delta(\delta P)^{n+\theta,k_{fin}+1})$$

et :

$$(\rho \underline{u})_{b_{ik}}^{n+1}.\ \underline{S}_{b_{ik}} = \left[(\rho \widetilde{\underline{u}})_{b_{ik}}^{n+1}\right]^{Corr}.\underline{S}_{b_{ik}} - \underline{D}_{b_{ik}}(\Delta t, \delta P^{n+\theta,k_{fin}}) - \underline{D}_{b_{ik}}^{NRec}(\Delta t, \delta(\delta P)^{n+\theta,k_{fin}+1})$$

ce qui conduit bien à :

$$\begin{aligned}
&\sum_{j\in Vois(i)} (\rho \underline{u})_{ij}^{n+1}.\ \underline{S}_{ij} + \sum_{k\in\gamma_b(i)} (\rho \underline{u})_{b_{ik}}^{n+1}.\ \underline{S}_{b_{ik}} \\
&\quad= \sum_{j\in Vois(i)} \left[(\rho \widetilde{\underline{u}})_{ij}^{n+1}\right]^{Corr}.\underline{S}_{ij} + \sum_{k\in\gamma_b(i)} \left[(\rho \widetilde{\underline{u}})_{b_{ik}}^{n+1}\right]^{Corr}.\underline{S}_{b_{ik}} \\
&\quad- \sum_{j\in Vois(i)} \underline{D}_{ij}(\Delta t, \delta P^{n+\theta,k_{fin}}) - \sum_{k\in\gamma_b(i)} \underline{D}_{b_{ik}}(\Delta t, \delta P^{n+\theta,k_{fin}}) \\
&\quad- \sum_{j\in Vois(i)} \underline{D}_{ij}^{NRec}(\Delta t, \delta(\delta P)^{n+\theta,k_{fin}+1}) - \sum_{k\in\gamma_b(i)} \underline{D}_{b_{ik}}^{NRec}(\Delta t, \delta(\delta P)^{n+\theta,k_{fin}+1})
\end{aligned}$$

soit :

$$\sum_{j\in Vois(i)} (\rho \underline{u})_{ij}^{n+1}.\underline{S}_{ij} + \sum_{k\in\gamma_b(i)} (\rho \underline{u})_{b_{ik}}^{n+1}.\ \underline{S}_{b_{ik}} = -\mathcal{F}(\delta P^{n+\theta,k_{fin}}, I) + \mathcal{FM}(\delta(\delta P)^{n+\theta,k_{fin}+1}, I) + \Gamma$$

et donc :

$$\int_{\Omega_i} \text{div}(\rho \underline{u})^{n+1} d\Omega = \Gamma$$

## 13.3 Mise en œuvre

On se reportera aux sections relatives aux sous-programmes `Preduv` (prédiction des vitesses) et `Resolp` (correction de pression). Pour la reconstruction des vitesses par moindres carrés à partir des flux de masse aux faces, on pourra voir `Recvmc`.

# 14- Sous-programme preduv

## 14.1 Fonction

Dans ce sous-programme, on effectue l'étape de prédiction de la vitesse $\underline{u}$. Ceci consiste à résoudre l'équation de quantité de mouvement (Q.D.M.) en traitant la pression $p$ de manière explicite. La solution en vitesse-pression est obtenue après une étape de correction sur la pression effectuée dans le sous-programme `resolp`, en utilisant la loi de conservation de la masse :

$$\frac{\partial \rho}{\partial t} + \mathrm{div}(\rho \underline{u}) = \Gamma, \tag{II.14.1}$$

où $\Gamma$ est le terme source de masse[1].
L'équation de conservation de la quantité de mouvement moyenne obtenue par application du théorème fondamental de la dynamique est :

$$\frac{\partial (\rho \underline{u})}{\partial t} + \mathrm{div}(\rho \underline{u} \otimes \underline{u}) = \mathrm{div}(\underline{\underline{\sigma}}) + \underline{S} - \mathrm{div}(\rho \underline{\underline{R}}) \tag{II.14.2}$$

où :

$$\underline{\underline{\sigma}} = -p\underline{\underline{Id}} + \underline{\underline{\tau}} \tag{II.14.3}$$

avec pour les écoulements newtoniens, la relation linéaire suivante :

$$\begin{aligned} \underline{\underline{\tau}} &= 2\ \mu\ \underline{\underline{D}} + \lambda\ tr(\underline{\underline{D}})\ \underline{\underline{Id}} \\ \underline{\underline{D}} &= \frac{1}{2}\ (\underline{\underline{\mathrm{grad}}}\ \underline{u} + {}^t\underline{\underline{\mathrm{grad}}}\ \underline{u}) \end{aligned} \tag{II.14.4}$$

$\underline{\underline{\sigma}}$ représente le tenseur de contraintes, $\underline{\underline{\tau}}$ le tenseur des contraintes visqueuses, $\mu$ la viscosité dynamique (moléculaire et éventuellement turbulente), $\underline{\underline{D}}$ le tenseur taux de déformation[2], $\underline{\underline{R}}$ le tenseur de Reynolds qui apparaît lors de l'application de l'opérateur moyenne à l'équation instantanée, $\underline{S}$ les termes sources. $\lambda$ est le second coefficient de viscosité. Il est relié à la viscosité de volume $\kappa$ par la relation

$$\lambda = \kappa - \frac{2}{3}\mu \tag{II.14.5}$$

Quand l'hypothèse de Stokes est vérifiée, la viscosité de volume $\kappa$ est nulle, soit $3\lambda + 2\mu = 0$. Cette hypothèse est implicite dans le code et dans le reste du doument, sauf en compressible.

L'équation de conservation de la quantité de mouvement s'écrit finalement :

$$\begin{aligned} \rho \frac{\partial \underline{u}}{\partial t} = &- \underbrace{\mathrm{div}(\rho \underline{u} \otimes \underline{u})}_{\text{convection}} + \underbrace{\mathrm{div}(\mu\ \underline{\underline{\mathrm{grad}}}\ \underline{u})}_{\text{diffusion}} \\ &\underbrace{+ \mathrm{div}(\mu\ {}^t\underline{\underline{\mathrm{grad}}}\ \underline{u})}_{\text{terme en gradient transposé}} \quad \underbrace{- \frac{2}{3}\ \underline{\mathrm{grad}}\ (\mu\ \mathrm{div}\underline{u})}_{\text{viscosité secondaire}} \quad - \mathrm{div}(\rho \underline{\underline{R}}) - \ \underline{\mathrm{grad}}\ (p) + (\rho - \rho_0)\ \underline{g} + \underline{u}\, \mathrm{div}(\rho\, \underline{u}) \\ &+ \underbrace{\Gamma(\underline{u}_i - \underline{u})}_{\text{terme source de Q.D.M. dû à la source de masse}} - \underbrace{\rho\ \underline{\underline{K}}_{pdc} \underline{u}}_{\text{perte de charge}} + \underbrace{\underline{T}_s^{exp} + T_s^{imp}\ \underline{u}}_{\text{autres termes sources de Q.D.M.}} \end{aligned} \tag{II.14.6}$$

[1] en $kg.m^{-3}.s^{-1}$

[2] À ne pas confondre, malgré la même notation $D$, avec les flux diffusifs décrits dans le sous-programme `navsto`

avec $p$ définissant l'écart à la pression hydrostatique de référence (la pression hydrostatique réelle étant calculée avec la masse volumique $\rho$ et non $\rho_0$) :

$$p = p^* - \rho_0 \, \underline{g} \cdot \underline{X} \tag{II.14.7}$$

($\underline{X}$ étant le vecteur de composantes $x$, $y$ et $z$).
$\mu_t$, $\underline{\underline{K}}_{pdc}$, $\underline{u}_i$ représentent respectivement la viscosité dynamique turbulente, le tenseur des pertes de charge et la valeur de la variable associée à la source de masse.
La divergence du tenseur des contraintes de Reynolds s'écrit :

$$-\mathrm{div}(\rho \underline{\underline{R}}) = \begin{cases} \underline{0} & \text{en laminaire,} \\ -\frac{2}{3}\,\underline{\mathrm{grad}}\,(\mu_t\ \mathrm{div}\underline{u}) + \mathrm{div}(\mu_t\ (\underline{\underline{\mathrm{grad}}}\ \underline{u} + {}^t\underline{\underline{\mathrm{grad}}}\ \underline{u})) - \frac{2}{3}\,\underline{\mathrm{grad}}\,(\rho\, k) & \text{pour les modèles à viscosité turbulente,} \\ -\mathrm{div}(\rho\, \underline{\underline{R}}) & \text{pour les modèles au second ordre,} \\ -\frac{2}{3}\,\underline{\mathrm{grad}}\,(\mu_t\ \mathrm{div}\underline{u}) + \mathrm{div}(\mu_t\ (\underline{\underline{\mathrm{grad}}}\ \underline{u} + {}^t\underline{\underline{\mathrm{grad}}}\ \underline{u})) & \text{en LES} \end{cases} \tag{II.14.8}$$

Le terme source de masse fait intervenir la vitesse locale $\underline{u}$ et aussi une vitesse $\underline{u}_i$ associée à la masse injectée (ou retirée). Lorsque $\Gamma < 0$, on ôte de la masse au système et on a donc $\underline{u}_i = \underline{u}$. Le terme est nul (*i.e.* $\Gamma(\underline{u}_i - \underline{u}) = \underline{0}$). Quand $\Gamma > 0$, on a un terme non nul si $\underline{u}_i \neq \underline{u}$. Dans ce sous-programme, tous les termes intervenant dans l'équation de conservation de la quantité de mouvement, excepté les termes de convection et diffusion, sont calculés et transmis au sous-programme `codits` qui résout l'équation complète (convection-diffusion avec termes sources).

## 14.2 Discrétisation

Le terme convectif en $\mathrm{div}(\underline{u} \otimes \rho\, \underline{u})$ introduit une non linéarité et un couplage des composantes de la vitesse $\underline{u}$ dans l'équation (II.14.6). Une linéarisation et un découplage des trois composantes de la vitesse sont réalisés lors de la discrétisation de cette étape de prédiction.
En effet, soit :

$$\widetilde{\underline{u}} = \underline{u}^n + \delta\underline{u} \tag{II.14.9}$$

La contribution exacte du terme convectif à prendre en compte dans cette étape de prédiction serait :

$$\mathrm{div}(\widetilde{\underline{u}} \otimes \rho\, \widetilde{\underline{u}}) = \mathrm{div}(\underline{u}^n \otimes \rho\, \underline{u}^n) + \mathrm{div}(\delta\underline{u} \otimes \rho\, \underline{u}^n) + \underbrace{\mathrm{div}(\underline{u}^n \otimes \rho\, \delta\underline{u})}_{\text{terme couplant linéaire}} + \underbrace{\mathrm{div}(\delta\underline{u} \otimes \rho\, \delta\underline{u})}_{\text{terme couplant et non linéaire}} \tag{II.14.10}$$

Les deux derniers termes de l'expression (II.14.10) sont *a priori* négligés de manière à obtenir un système en vitesse qui soit découplé et donc, éviter l'inversion d'une matrice pouvant être de très grande taille. Ces deux termes peuvent néanmoins être pris en compte de manière plus ou moins approchée par extrapolation explicite du flux de masse en $n + \theta_F$ (pour le terme couplant linéaire provenant de la convection de $\underline{u}^n$ par $\delta\underline{u}$) et utilisation d'un point-fixe par sous itération sur le sous programme `navsto` (pour le terme non-linéaire, en spécifiant `NTERUP` $> 1$).

L'équation (II.14.6) est discrétisée au temps $n + \theta$ à l'aide d'un $\theta$-schéma, et le tenseur des pertes de charges décomposé en une partie diagonale $\underline{\underline{K}}_d$ et une extradiagonale $\underline{\underline{K}}_e$ (soit $\underline{\underline{K}}_{pdc} = \underline{\underline{K}}_d + \underline{\underline{K}}_e$).
• La pression est supposée connue en $n - 1 + \theta$ (décalage temporel pression-vitesse) et le gradient naturellement calculé à cet instant.
• Les termes sources de viscosité secondaire, de gradient transposé, ceux provenant du modèle de turbulence[3], $\rho\, \underline{\underline{K}}_e\ \underline{u}$, $(\rho - \rho_0)\underline{g}$ ainsi que $\underline{T}_s^{exp}$ et $\Gamma\, \underline{u}_i$ sont pris de manière explicite au temps $n$, ou extrapolés suivant le schéma en temps choisi pour les propriétés physique et les termes sources.
• Les termes sources $\underline{u}\ \mathrm{div}(\rho\, \underline{u})$, $\Gamma\ \underline{u}$, $T_s^{imp}\ \underline{u}$ et $-\rho\, \underline{\underline{K}}_d\ \underline{u}$ sont implicités est calculés à l'instant $n + \theta$.

[3] excepté $\mathrm{div}(\mu_t\ (\underline{\underline{\mathrm{grad}}}\ \underline{u}))$

• Le terme de diffusion $\mathrm{div}(\mu_{tot}\,\underline{\underline{\mathrm{grad}}}\,\underline{u})$ est implicité : la vitesse est prise à l'instant $n+\theta$ et la viscosité explicitée ou extrapolée.
• Enfin, le terme de convection en $\mathrm{div}(\,\underline{u}\otimes(\rho\underline{u})\,)$ est implicité : la composante résolue de la vitesse est prise en $n+\theta$, et le flux de masse, explicité, ou extrapolé en $n+\theta_F$.

Par souci de clarté, on suppose, en l'absence d'indication, que les propriétes physiques $\Phi$ ($\rho$, $\mu_{tot}$, ...) et le flux de masse ($\rho\underline{u}$) sont pris respectivement aux instants $n+\theta_\Phi$ et $n+\theta_F$, où $\theta_\Phi$ et $\theta_F$ dépendent des schémas en temps spécifiquement utilisés pour ces grandeurs[4].

La discrétisation temporelle de l'équation (II.14.6) s'écrit alors comme suit :

$$\begin{array}{c}\rho\,\dfrac{\widetilde{\underline{u}}^{n+1}-\underline{u}^{n}}{\Delta t}+\mathrm{div}(\,\widetilde{\underline{u}}^{n+\theta}\otimes(\rho\underline{u})\,)-\mathrm{div}(\mu_{tot}\,\underline{\underline{\mathrm{grad}}}\,\widetilde{\underline{u}}^{n+\theta})=\\ -\underline{\mathrm{grad}}\,p^{n-1+\theta}+\mathrm{div}(\rho\,\underline{u})\,\widetilde{\underline{u}}^{n+\theta}+(\Gamma\,\underline{u}_{i})^{n+\theta_S}-\Gamma^{n}\,\widetilde{\underline{u}}^{n+\theta}\\ -\rho\,\underline{\underline{K}}^{n}_{d}\,\widetilde{\underline{u}}^{n+\theta}-(\rho\,\underline{\underline{K}}_{e}\,\underline{u})^{n+\theta_S}+(\underline{T}^{exp}_{s})^{n+\theta_S}+T^{imp}_{s}\,\widetilde{\underline{u}}^{n+\theta}\\ +[\mathrm{div}(\mu_{tot}\,{}^{t}\underline{\underline{\mathrm{grad}}}\,\underline{u})]^{n+\theta_S}-\dfrac{2}{3}[\,\underline{\mathrm{grad}}\,(\mu_{tot}\,\mathrm{div}\underline{u})]^{n+\theta_S}+(\rho-\rho_0)\underline{g}-(\underline{turb})^{n+\theta_S}\end{array}\qquad\text{(II.14.11)}$$

où, par souci de simplification, on a posé :

$$\mu_{tot}=\begin{cases}\mu+\mu_t & \text{pour les modèles à viscosité turbulente ou en LES,}\\ \mu & \text{pour les modèles au second ordre ou en laminaire}\end{cases}\qquad\text{(II.14.12)}$$

et :

$$\underline{turb}^{n}=\begin{cases}\dfrac{2}{3}\underline{\mathrm{grad}}\,(\rho^{n}\,k^{n}) & \text{pour les modèles à viscosité turbulente,}\\ \mathrm{div}(\rho^{n}\,\underline{\underline{R}}^{n}) & \text{pour les modèles au second ordre,}\\ 0 & \text{en laminaire ou en LES}\end{cases}\qquad\text{(II.14.13)}$$

Par analogie avec l'écriture du $\theta$-schéma pour une variable scalaire, $\widetilde{\underline{u}}^{n+\theta}$ est interpolée à partir de la vitesse prédite $\widetilde{\underline{u}}^{n+1}$ de la manière suivante[5] :

$$\widetilde{\underline{u}}^{n+\theta}=\theta\,\widetilde{\underline{u}}^{n+1}+(1-\theta)\,\underline{u}^{n}\qquad\text{(II.14.14)}$$

Avec :

$$\begin{cases}\theta=1 & \text{Pour un schéma de type Euler implicite d'ordre 1.}\\ \theta=1/2 & \text{Pour un schéma de type Cranck-Nicolson d'ordre 2.}\end{cases}\qquad\text{(II.14.15)}$$

L'équation (II.14.11) est alors réécrite sous la forme :

$$\begin{array}{c}\underbrace{\left(\dfrac{\rho}{\Delta t}-\theta\,\mathrm{div}(\rho\,\underline{u})+\theta\;\Gamma^{n}+\theta\;\rho\,\underline{\underline{K}}^{n}_{d}-\theta\,T^{imp}_{s}\right)}_{f^{imp}_{s}}(\widetilde{\underline{u}}^{n+1}-\underline{u}^{n})\\ +\theta\,\mathrm{div}(\widetilde{\underline{u}}^{n+1}\otimes(\rho\underline{u}))-\theta\,\mathrm{div}(\mu_{tot}\,\underline{\underline{\mathrm{grad}}}\,\widetilde{\underline{u}}^{n+1})=\\ -(1-\theta)\,\mathrm{div}(\underline{u}^{n}\otimes(\rho\underline{u}))+(1-\theta)\,\mathrm{div}(\mu_{tot}\,\underline{\underline{\mathrm{grad}}}\,\underline{u}^{n})\end{array}\qquad\text{(II.14.16)}$$

$$f^{exp}_{s}\begin{cases}-\underline{\mathrm{grad}}\,p^{n-1+\theta}+\mathrm{div}(\rho\,\underline{u})\,\underline{u}^{n}+(\Gamma^{n}\,\underline{u}_{i})^{n+\theta_S}-\Gamma^{n}\,\underline{u}^{n}\\ -(\,\rho\,\underline{\underline{K}}_{e}\,\underline{u})^{n+\theta_S}-\rho\,\underline{\underline{K}}^{n}_{d}\,\underline{u}^{n}+(\underline{T}^{exp}_{s})^{n+\theta_S}+T^{imp}_{s}\,\underline{u}^{n}\\ +[\mathrm{div}(\mu_{tot}\,{}^{t}\underline{\underline{\mathrm{grad}}}\,\underline{u})]^{n+\theta_S}-\dfrac{2}{3}[\,\underline{\mathrm{grad}}\,(\mu_{tot}\,\mathrm{div}\underline{u})]^{n+\theta_S}+(\rho-\rho_0)\underline{g}-(\underline{turb})^{n+\theta_S}\end{cases}$$

d'où l'équation résolue par le sous-programme `codits` :

$$\begin{array}{c}f^{imp}_{s}(\widetilde{\underline{u}}^{n+1}-\underline{u}^{n})+\theta\,\mathrm{div}(\widetilde{\underline{u}}^{n+1}\otimes(\rho\underline{u}))-\theta\,\mathrm{div}(\,\mu_{tot}\,\underline{\underline{\mathrm{grad}}}\,\widetilde{\underline{u}}^{n+1})=\\ -(1-\theta)\,\mathrm{div}(\underline{u}^{n}\otimes(\rho\underline{u}))+(1-\theta)\,\mathrm{div}(\,\mu_{tot}\,\underline{\underline{\mathrm{grad}}}\,\underline{u}^{n})+\underline{f}^{\,exp}_{s}\end{array}\qquad\text{(II.14.17)}$$

[4] cf. `introd`
[5] si $\theta=1/2$, ou qu'une extrapolation est utilisée, l'ordre 2 n'est obtenu que si le pas de temps $\Delta t$ est uniforme en temps et en espace.

La méthode de discrétisation spatiale est développée dans le sous-programme `codits`.

REMARQUES :
■ Dans le cas standard sans extrapolation, le terme $-T_s^{imp}$ n'est ajouté à $f_s^{imp}$ que s'il est positif afin de ne pas affaiblir la dominance de la diagonale de la matrice à inverser.
■ Si une extrapolation est utilisée, par contre, $T_s^{imp}$ est ajouté à $f_s^{imp}$ quel que soit son signe. En effet, l'idée intuitive qui consiste à prendre :

$$\begin{cases} (\underline{T}_s^{exp} + T_s^{imp}\,\underline{u})^{\,n+\theta_S} & \text{si } T_s^{imp} > 0 \\ (\underline{T}_s^{exp})^{\,n+\theta_S} + T_s^{imp}\,\underline{u}^{n+\theta} & \text{sinon} \end{cases} \tag{II.14.18}$$

aboutit à une incohérence dans le traitement si $T_s^{imp}$ change de signe entre deux pas de temps.
■ la partie diagonale $\underline{\underline{K}}_d$ du terme de perte de charge est utilisée dans $f_s^{imp}$. En fait, pour être rigoureux, il faudrait ne retenir que les contributions positives (point signalé dans le sous-programme utilisateur associé `uskpdc`). Cette prise en compte sera à améliorer.
■ Le terme $\theta\,\Gamma^n - \theta\,\mathrm{div}(\rho\,\underline{u})$ ne pose pas de problème pour la dominance de la diagonale de la matrice car il est exactement compensé par le terme de convection (cf. `covofi`).

## 14.3 Mise en œuvre

L'équation de conservation de la quantité de mouvement est donc résolue de façon découplée. Ainsi, l'intégration des différents termes a été effectuée afin de traiter séparément l'équation obtenue pour chaque composante de la vitesse.
Dans le sous-programme `preduv`, on calcule pour chaque composante le second membre $f_s^{exp}$ du système (II.14.16), les termes implicites du système (à l'exception des termes de convection-diffusion), et le terme de viscosité totale aux faces internes[6] et de bord. Ces termes sont alors transmis au sous-programme `codits` qui construit et résout le système complet obtenu pour chaque composante de la vitesse avec les termes de convection-diffusion.

Le résidu de normalisation pour la résolution du système en pression (`resolp`) est calculé dans `preduv`. Il est défini par la norme de la grandeur

$$\mathrm{div}(\rho\,\widetilde{\underline{u}}^{n+1} + \Delta t \underline{\mathrm{grad}}\ P^{n-1+\theta}) - \Gamma$$

intégrée sur chaque cellule `IEL` du maillage ($\Omega_{iel}$) soit, symboliquement, par la racine carrée de la somme sur les cellules du maillage de la quantité

$$\texttt{XNORMP(IEL)} = \int_{\Omega_{iel}} [\mathrm{div}(\rho\,\widetilde{\underline{u}}^{n+1} + \Delta\,t\,\underline{\mathrm{grad}}\ P^{n-1+\theta}) - \Gamma\,]\,d\Omega.$$

Il représente le second membre du système qui porterait sur la pression si le gradient de pression n'était pas pris en compte lors de l'étape de prédiction des vitesses. On note que si l'on utilisait directement le second membre de l'équation portant sur l'incrément de pression, on obtiendrait, pour un calcul stationnaire mené à convergence, un résidu de normalisation tendant vers zéro, ce qui serait pénalisant et peu utile.

Au début de `preduv`, on ne dispose pas encore de $\widetilde{\underline{u}}^{n+1}$ et il n'est donc pas possible de calculer le résidu de normalisation en totalité. Cependant, le calcul du résidu complet à la fin de `preduv` n'est pas souhaitable non plus, car on devrait alors monopoliser un tableau de travail pour conserver le gradient de pression tout au long de `preduv`. Le calcul du résidu de normalisation est donc réalisé en deux fois.

[6] valeur nécessaire pour l'intégration du terme de diffusion dans `codits`, $(\mu_{tot})_{ij}\,\frac{\texttt{SURFN}}{\texttt{DIST}}$

La quantité $\int_{\Omega_{iel}} \mathrm{div}(\Delta t \, \underline{\mathrm{grad}} \; P^{n-1+\theta}) - \Gamma d\Omega$ est calculée au début de `preduv` et on y ajoute le complément $\int_{\Omega_{iel}} \mathrm{div}(\rho \, \underline{\widetilde{u}}^{n+1}) d\Omega$ à la fin de `preduv`.

On calcule donc tout d'abord le gradient de pression aux cellules à l'instant $n-1+\theta$ par un appel à `grdcel`. On utilise alors `inimas` pour évaluer $\Delta t \, S \, \underline{\mathrm{grad}} \; P^{n-1+\theta} \cdot \underline{n}$ aux faces (de surface $S$ et de normale $\underline{n}$). Pour cela, en entrée de `inimas` , le tableau de travail `TRAV` contient $\frac{\Delta t}{\rho} \, \underline{\mathrm{grad}} \; P^{n-1+\theta}$ ; en sortie, les tableaux `VISCF` et `VISCB` contiennent la valeur de $\Delta t \, S \, \underline{\mathrm{grad}} \; P^{n-1+\theta} \cdot \underline{n}$ aux faces internes et de bord respectivement.

On utilise ensuite `divmas` qui place alors dans `XNORMP` la valeur de $\int_{\Omega_{iel}} \mathrm{div}(\Delta t \, \underline{\mathrm{grad}} \; P^{n-1+\theta}) d\Omega$ aux cellules à partir des tableaux `VISCF` et `VISCB`.

On ajoute enfin à `XNORMP` la contribution $\int_{\Omega_{iel}} -\Gamma^n d\Omega$ du terme source de masse.

On applique pour $\rho \, \underline{\widetilde{u}} + \Delta t \underline{\mathrm{grad}} \; P$ les conditions aux limites de la vitesse. Les conditions aux limites utilisées pour le gradient de pression (ou plutôt pour $\frac{\Delta t}{\rho} \, \underline{\mathrm{grad}} \; P^{n-1+\theta}$) pour le calcul de $\int_{\Omega_{iel}} \mathrm{div}(\Delta t \, \underline{\mathrm{grad}} \; P^{n-1+\theta}) d\Omega$ sont donc les conditions aux limites de la vitesse homogénéisées : ainsi, on suppose que dans la direction normale aux entrées et aux parois, le gradient de pression (ou plutôt $\frac{\Delta t}{\rho} \, \underline{\mathrm{grad}} \; P^{n-1+\theta}$) est nul et que dans la direction normale aux symétries et aux sorties, il reste inchangé.

De plus, pour gagner du temps calcul lors du passage par `inimas`, on se contente, sur les maillages non orthogonaux, d'une évaluation des valeurs aux faces à l'ordre 1 en espace (pas de reconstruction : `NSWRP=1`). En effet, on cherche à évaluer un simple résidu de normalisation global : la précision locale n'a donc pas d'intérêt.

Le calcul du résidu sera complété à la fin de `preduv`.

**• Calcul en partie du résidu de normalisation pour l'étape de pression**

Dans cette première étape on calcule dans le tableau `XNORMP(NCELET)` la grandeur

$$\mathrm{div}(\Delta t \, \underline{\mathrm{grad}} \; P^{n-1+\theta}) - \Gamma$$

intégrée sur chaque cellule `IEL` du maillage ($\Omega_{iel}$) soit, symboliquement,

$$\texttt{XNORMP(IEL)} = \int_{\Omega_{iel}} \mathrm{div}(\Delta t \, \underline{\mathrm{grad}} \; P^{n-1+\theta}) - \Gamma d\Omega$$

On réalise cette opération en utilisant successivement `inimas` (calcul aux faces dans `VISCF` et `VISCB` de $\Delta t \, \underline{\mathrm{grad}} \; P^{n-1+\theta}$ à partir du tableau de travail `TRAV`$=\frac{\Delta t}{\rho} \underline{\mathrm{grad}} \; P^{n-1+\theta}$, assorti des conditions aux limites de vitesse homogènes et sans reconstruction) et `divmas` (calcul dans `XNORMP` de l'intégrale sur les cellules). Par une simple boucle, on ajoute ensuite la contribution du terme source de masse $\Gamma$. Ce calcul est complété à la fin de `preduv`.

**• Calcul en partie du terme $\underline{f}_s^{exp}$**

Pour représenter le second membre correspondant à chaque composante de la vitesse, on utilise les tableaux `TRAV(IEL,DIR)`, `TRAVA(IEL,DIR)` et `PROPCE`, où `IEL` est le numéro de la cellule et `DIR` la direction (x, y, z). Quatre cas sont à considérer suivant que les termes sources sont extrapolés en $n+\theta_S$, ou que l'on itère par un point fixe sur le système en vitesse-pression (`NTERUP`$>1$).
• Si on extrapole les termes sources et que l'on itère sur `navsto`

- `TRAV` reçoit les termes sources qui sont recalculés au cours de toutes les itérations sur `navsto` et qui ne sont pas extrapolés ($\underline{\text{grad}}\ P^{n-1+\theta}$ et $(\rho-\rho_0)\underline{g}$[7]).

- `TRAVA` reçoit les termes sources qui ne changent pas au cours des itérations sur `navsto` et qui ne sont pas extrapolés ($T_s^{imp}\,u^n$ , $-\rho\,\underline{\underline{K}}_d\,u^n$ , $-\Gamma^n\,u^n$, ...).

- `PROPCE` reçoit les termes sources devant être extrapolés.

• Sans itération sur `navsto`, `TRAVA` est inutile et son contenu est directement stocké dans `TRAV`.

• Sans extrapolation des termes sources, `PROPCE` est inutile et son contenu est directement stocké dans `TRAVA` (ou dans `TRAV` si `TRAVA` est inutile).
Ainsi, sans extrapolation des termes sources, et sans itération sur `navsto`, tout les termes sources vont directement dans `TRAV`.

- On dispose déjà du gradient de pression sur les cellules à l'instant $n-1+\theta$. Le terme de gravité est alors ajouté au vecteur `TRAV` qui contient déjà le gradient de pression. Ainsi, on a par exemple pour la direction x :

$$\texttt{TRAV}(\texttt{IEL},1) = |\Omega_{IEL}|(-(\frac{\partial p}{\partial x})_{\texttt{IEL}} + (\rho(\texttt{IEL})-\rho_0)g_x) \qquad \text{(II.14.19)}$$

- Si une extrapolation des termes sources est utilisée, le vecteur `TRAV` (ou `TRAVA`) reçoit à la première itération sur `navsto`, $-\theta_S$ fois la contribution au temps $n-1$ des termes sources devant être extrapolés[8] (stockée dans `PROPCE`). `PROPCE` est ensuite réinitialisé à zéro de façon à pouvoir recevoir plus tard la contribution au pas de temps courant des termes sources qui sont extrapolés.

- Le terme correspondant au modèle de turbulence n'est calculé que lors de la premiere itération sur `navsto` puis ajouté à `TRAVA`, `TRAV` ou `PROPCE` suivant que les termes sources sont extrapolés, ou que l'on itère sur `navsto`.

  ■ Modèles à viscosité turbulente :
  Si `IGRHOK` $=1$, alors on calcule $-\frac{2}{3}\,\rho\ \underline{\text{grad}}\ k$ (et non, comme on devrait, $-\frac{2}{3}\underline{\text{grad}}\ (\rho k)$) par simplification (cf. paragraphe 14.4). Le gradient de $k$ est calculé sur la cellule par le sous-programme `grdcel`.
  Si `IGRHOK` $=0$, ce terme est supposé être implicitement pris en compte dans la pression.
  ■ Modèles au second ordre :
  Le calcul du terme $-\text{div}(\rho\underline{\underline{R}})$ s'effectue en deux temps. Tout d'abord, on appelle le sous-programme `divrij` qui projette le vecteur $\underline{\underline{R}}.\underline{e}_{\texttt{DIR}}$ aux faces, pour la direction `DIR`. Puis, on appelle le sous-programme `divmas` qui en calcule la divergence.

- Les termes de viscosité secondaire $-\frac{2}{3}\underline{\text{grad}}\ (\mu_{tot}\text{div}\,\underline{u})$ et de gradient transposé $\text{div}(\mu_{tot}\,{}^t\underline{\underline{\text{grad}}}\ \underline{u})$ sont calculés (s'ils sont pris en compte *i.e.* `IVISSE(IPHAS)` $=1$, où `IPHAS` est le numéro de la phase traitée) par le sous-programme `vissec`. Il ne sont calculés qu'à la première itération sur

[7] en réalité $(\rho-\rho_0)\underline{g}$ ne change pas, mais il est rapide à calculer ce qui évite d'avoir un traitement supplémentaire pour ce terme.
[8] car $(\underline{T}_s^{exp})^{n+\theta_S} = (1+\theta_S)\,(\underline{T}_s^{exp})^n - \theta_S\,(\underline{T}_s^{exp})^{n-1}$

navsto. Au cours de cette étape, le tableau TRAV est utilisé comme tableau de travail lors de l'appel au sous-programme vissec. Il retrouve sa valeur à la fin de cet appel, son contenu étant temporairement stocké dans les vecteurs W7 à W9.

- Les termes correspondant aux pertes de charges ($\rho \underline{\underline{K}}_{pdc} u$), s'ils existent ( NCEPDP $> 0$ ), sont calculés par le sous-programme tsepdc à la première itération sur navsto. Ils sont décomposés en deux parties :
  ■ Une première, correspondant à la contribution des termes diagonaux ($-\rho \underline{\underline{K}}_{d} \underline{u}$) qui n'est pas extrapolée.
  ■ Une seconde, correspondant aux termes extradiagonaux ($-\rho \underline{\underline{K}}_{e} \underline{u}$) qui peut l'être ou non.
  Au cours de cette étape, le tableau TRAV est utilisé comme tableau de travail lors de l'appel au sous-programme tsepdc. Il retrouve sa valeur à la fin de cet appel, son contenu étant temporairement stocké dans les vecteurs W7 à W9.

**• Calcul du terme de viscosité aux faces** $(\mu_{tot})_{ij} \dfrac{\text{SURFN}}{\text{DIST}}$

Le calcul du terme de viscosité totale aux faces est effectué par le sous-programme viscfa et stocké dans les tableaux VISCF et VISCB pour les faces internes et faces de bord respectivement.
Lors de l'intégration des termes de convection-diffusion dans le sous-programme codits, on distingue les termes non reconstruits, intégrés dans la matrice $\underline{\underline{EM}}$ , de l'ensemble des termes (non reconstruits + gradients de reconstruction) associés à l'opérateur $\mathcal{E}$ (non linéaire)[9]. De la même manière, on distingue la viscosité totale aux faces utilisée dans $\mathcal{E}$, tableaux VISCF et VISCB, de la viscosité totale aux faces utilisée dans $\underline{\underline{EM}}$, tableaux VISCFI et VISCBI.
Pour les modèles à viscosité turbulente et en LES, ces deux tableaux sont identiques et contiennent $\mu_t + \mu$. Pour les modèles au second ordre, ils contiennent normalement $\mu$, mais pour des simples raisons de stabilité numérique, on peut choisir de mettre $\mu_t + \mu$ dans la matrice (*i.e.* dans $\underline{\underline{EM}}$) en conservant $\mu$ au second menbre (*i.e.* dans $\mathcal{E}$). De par la résolution par incréments, cette manipulation ne change pas le résultat. Cette option est activée par l'indicateur IRIJNU $= 1$
Si la vitesse n'est pas diffusée ( IDIFF(IUIPH) $< 1$), alors les termes VISCF et VISCB sont mis à zéro.

**• Calcul du second membre complet, de $f_s^{imp}$ et résolution de l'équation**

Les équations d'évolution des composantes de la quantité de mouvement sont résolues de façon découplée. On utilise, par conséquent, un seul tableau ROVSDT pour représenter la partie diagonale de la matrice obtenue pour chaque composante de la vitesse.
Pour chaque composante de la vitesse :

- Lors de la première itération sur le sous-programme navsto, les parties implicites et explicites des termes sources utilisateurs sont calculées par appel au sous-programme ustsns.
  ■ La partie implicite ($T_s^{imp}$) est conservée dans le vecteur XIMPA pour les itérations suivantes en cas d'utilisation du point-fixe sur le système en vitesse-pression, et la contribution issue des mêmes termes implicites ($T_s^{imp}\, \underline{u}^n$) ajoutée à TRAVA ou à TRAV.
  ■ La partie explicite ($T_s^{exp}$) est ajoutée à TRAVA, TRAV ou PROPCE suivant que les termes sources sont extrapolés, ou que l'on itère sur navsto.

- Le terme d'accumulation de masse ($\text{div}(\rho \underline{u})$) est calculé en appelant le sous-programme divmas avec en argument le flux de masse. Lors de la première itération faite sur le sous-programme navsto, le terme correspondant à la contribution explicite de l'accumulation de masse ($\underline{u}^n\ \text{div}(\rho \underline{u})$) est ajouté à TRAVA ou à TRAV. Le vecteur ROVSDT est initialisé par $\theta\, \text{div}(\rho \underline{u})$ (par cohérence avec ce qui est fait dans le sous-programme bilsc2) puis la contribution du terme instationnaire ($\dfrac{\rho}{\Delta t}$)

[9] par cohérence avec les opérateurs $\mathcal{EM}$ et $\mathcal{E}$ définis dans navsto

ajoutée à ce dernier.

- Le vecteur ROVSDT est ensuite complété avec la contribution des termes sources implicites utilisateur (stockée dans XIMPA) et avec celle des perte de charge ($\rho \underline{\underline{K}}_d$) si NCEPDP $> 0$.
  ■ Dans le cas ou les termes sources ne sont pas extrapolés, la partie implicite des termes sources utilisateur n'est ajoutée à ROVSDT que si elle est négative de façon à ne pas affaiblir la diagonale du sytème.
  ■ Dans le cas ou ils sont extrapolés par contre, elle est prise en compte quel que soit son signe.

- Les termes sources implicite et explicite de masse, s'ils existent ( NCESMP $> 0$ ), sont calculés à la première itération sur navsto par le sous-programme catsma. $\Gamma\,\underline{u}_i$ est ajouté à TRAV, TRAVA ou PROPCE pour être éventuellement extrapolé. $\Gamma\,\underline{u}^n$ est rajouté à TRAV ou TRAVA et $-\Gamma$ à ROVSDT.

- Le second membre est enfin assemblé en tenant compte de toutes les contributions stockées dans les tableaux PROPCE, TRAVA et TRAV.
  ■ Si les termes sources sont extrapolés alors :

$$\mathtt{SMBR} = (1-\theta_S)\,\mathtt{PROPCE} + \mathtt{TRAVA} + \mathtt{TRAV}$$

  ■ Sinon on a directement :

$$\mathtt{SMBR} = \mathtt{TRAVA} + \mathtt{TRAV}$$

- Prise en compte des physiques particulières (lagrangien, arc électrique, ...) ajoutés directement à SMBR.

- La resolution du système linéaire est faite par le sous-programme codits avec pour argument ROVSDT et SMBR.

- Si on utilise le couplage instationnaire renforcé vitesse-pression ( IPUCOU = 1 ) (uniquement disponible avec l'ordre 1, sans extrapolation des termes sources et sans itération sur navsto) on résout, en utilisant pour codits :

$$\underline{\underline{EM}}_{\mathtt{DIR}} \cdot (\underline{\underline{RHO}}^n)^{-1} \cdot \underline{T}_{\mathtt{DIR}} = \underline{\underline{\Omega}} \cdot \underline{1} \tag{II.14.20}$$

  avec $\underline{\underline{RHO}}^n$ le tenseur diagonal d'élément $\rho^n_{IEL}$, $\underline{\underline{\Omega}}$ le tenseur diagonal d'élément $|\Omega_{IEL}|$, $\underline{1}$ le vecteur de composantes toutes égales à 1.
  L'inversion du système par codits fournit $(\underline{\underline{RHO}}^n)^{-1} \cdot \underline{T}_{\mathtt{DIR}}$, qui est ensuite multiplié par $\underline{\underline{RHO}}^n$ pour obtenir $\underline{T}_{\mathtt{DIR}}$. Ceci est réalisé pour chaque composante DIR de la vitesse. $\underline{T}_{\mathtt{DIR}}$ est alors une approximation de type matrice diagonale de $\underline{\underline{RHO}}^n \cdot \underline{\underline{EM}}^{-1}_{\mathtt{DIR}}$, avec $\underline{\underline{EM}}_{\mathtt{DIR}}$ représentant toujours la partie implicite de l'équation de quantité de mouvement (*i.e.* ROVSDT + contribution des termes de convection-diffusion pris en compte dans le sous-programme matrix). $\underline{T}_{\mathtt{DIR}}$ intervient dans l'étape corrective (cf. sous-programme resolp).

Fin de la boucle sur les composantes de la vitesse.

**• Fin du calcul du résidu de normalisation pour l'étape de pression**

Comme indiqué précédemment, on peut maintenant compléter le calcul du résidu de normalisation pour l'étape de pression de resolp.

Le tableau `XNORMP` contient déjà $\int\limits_{\Omega_{iel}} \mathrm{div}(\Delta t \, \underline{\mathrm{grad}} \; P^{n-1+\theta}) - \Gamma d\Omega$ . On lui ajoute donc $\int\limits_{\Omega_{iel}} \mathrm{div}(\rho \, \widetilde{\underline{u^{n+1}}}) d\Omega$.

Pour cela, on procède comme précédemment pour le calcul de $\int\limits_{\Omega_{iel}} \mathrm{div}(\Delta t \, \underline{\mathrm{grad}} \; P^{n-1+\theta}) d\Omega$. Un appel à `inimas` permet d'obtenir $\rho \, S \, \widetilde{\underline{u}}^{n+1} \cdot \widetilde{\underline{n}}$ aux faces à partir de $\widetilde{\underline{u}}^{n+1}$ connu aux cellules (tableau `RTP`). Les conditions aux limites pour `inimas` sont naturellement celles de la vitesse. Comme précédemment, on se contente pour gagner du temps calcul lors du passage par `inimas`, d'une évaluation des valeurs aux faces à l'ordre 1 en espace sur les maillages non orthogonaux (pas de reconstruction : `NSWRP=1`). On utilise ensuite `divmas` pour calculer aux cellules la divergence $\int\limits_{\Omega_{iel}} \mathrm{div}(\rho \, \widetilde{\underline{u}}^{n+1}) d\Omega$ et l'ajouter directement à `XNORMP`.

Pour finir, le résidu de normalisation est déterminé et stocké dans `RNORMP(IIPHAS)` par un appel à `prodsc` (qui réalise le calcul de la somme sur les cellules du carré des valeurs de `XNORMP` et en prend la racine carrée).

On résume dans les tableaux (II.14.21), (II.14.22), (II.14.23) et (II.14.24) les différentes contributions (hors convection-diffusion) affectées à chacun des vecteurs TRAV, TRAVA, PROPCE et ROVSDT à l'itération $n$. On différencie pour chacun des schémas en temps appliqués aux les termes sources, deux cas suivant qu'un point fixe sur le système en vitesse-pression est utilisé ou non (itération sur navsto pour NTERUP$>1$). En l'absence d'indication les propriétés physiques $\Phi$ ($\rho$,$\mu$,etc...) sont supposées prises au temps $n+\theta_\Phi$, et le flux de masse ($\rho\underline{u}$) pris au temps $n+\theta_F$, où $\theta_\Phi$ et $\theta_F$ dépendent des schémas en temps spécifiquement utilisés pour ces grandeurs (cf. introd).

Les termes figurant dans ces tableaux sont écrits tels qu'ils ont été implantés dans le code, d'où l'origine de certaines différences par rapport à l'écriture adoptée dans l'équation (II.14.16).

Par souci de simplification, on pose :

$$\mu_{tot} = \begin{cases} \mu+\mu_t & \text{pour les modèles à viscosité turbulente ou en LES,} \\ \mu & \text{pour les modèles au second ordre ou en laminaire} \end{cases}$$

Avec extrapolation des termes sources

$$\underline{turb}^n = \begin{cases} \dfrac{2}{3}\rho^n \,\underline{\text{grad}}\,(k^n) & \text{pour les modèles à viscosité turbulente,} \\ \text{div}(\rho^n \underline{\underline{R}}^n) & \text{pour les modèles au second ordre,} \\ 0 & \text{en laminaire ou en LE.} \end{cases}$$

• NTERUP $=1$ : $\text{SMBR}^n = (1-\theta_S)\,\text{PROPCE}^n + \text{TRAV}^n$

| | |
|---|---|
| $\text{ROVSDT}^n$ | $\dfrac{\rho}{\Delta t} - \theta\,\text{div}(\rho\,\underline{u}) + \theta\;\Gamma^n + \theta\;\rho\,\underline{\underline{K}}_d^n - \theta\,T_s^{imp}$ |
| $\text{PROPCE}^n$ | $\underline{T}_s^{exp,\,n} - \rho^n\,\underline{\underline{K}}_e^n\,\underline{u}^n + \Gamma^n\,\underline{u}_i^n$<br>$-\underline{turb}^n + \text{div}(\mu_{tot}^n\,{}^t\underline{\underline{\text{grad}}}\,\underline{u}^n) + \dfrac{2}{3}\,\underline{\text{grad}}\,(\mu_{tot}^n\,\text{div}\dfrac{(\rho\underline{u})}{\rho^n})$ |
| $\text{TRAV}^n$ | $-\underline{\text{grad}}\,p^{n-1+\theta} + (\rho-\rho_0)\underline{g}$<br>$-\theta_S\,\text{PROPCE}^{n-1} - \rho\,\underline{\underline{K}}_d^n\,\underline{u}^n$<br>$+T_s^{imp}\;\underline{u}^n + \text{div}(\rho\,\underline{u})\,\underline{u}^n - \Gamma^n\;\underline{u}^n$ |

(II.14.21)

• NTERUP $>1$ (sous-itération $k$) : $\text{SMBR}^n = (1-\theta_S)\,\text{PROPCE}^n + \text{TRAVA}^n + \text{TRAV}^n$

| | |
|---|---|
| $\text{ROVSDT}^n$ | $\dfrac{\rho}{\Delta t} - \theta\,\text{div}(\rho\,\underline{u}) + \theta\;\Gamma^n + \theta\;\rho\,\underline{\underline{K}}_d^n - \theta\,T_s^{imp}$ |
| $\text{PROPCE}^n$ | $\underline{T}_s^{exp,\,n} - \rho^n\,\underline{\underline{K}}_e^n\,\underline{u}^n + \Gamma^n\,\underline{u}_i^n$<br>$-\underline{turb}^n + \text{div}(\mu_{tot}^n\,{}^t\underline{\underline{\text{grad}}}\,\underline{u}^n) + \dfrac{2}{3}\,\underline{\text{grad}}\,(\mu_{tot}^n\,\text{div}\dfrac{(\rho\underline{u})}{\rho^n})$ |
| $\text{TRAVA}^n$ | $-\theta_S\,\text{PROPCE}^{n-1} - \rho\,\underline{\underline{K}}_d^n\,\underline{u}^n + T_s^{imp}\;\underline{u}^n + \text{div}(\rho\,\underline{u})\,\underline{u}^n - \Gamma^n\;\underline{u}^n$ |
| $\text{TRAV}^n$ | $-\underline{\text{grad}}\,(p^{n+\theta})^{(k-1)} + (\rho-\rho_0)\underline{g}$ |

(II.14.22)

Sans extrapolation des termes sources

$$\underline{turb}^n = \begin{cases} \dfrac{2}{3}\rho \,\underline{\text{grad}}\,(k^n) & \text{pour les modèles à viscosité turbulente,} \\ \text{div}(\rho\,\underline{\underline{R}}^n) & \text{pour les modèles au second ordre,} \\ 0 & \text{en laminaire ou en LES.} \end{cases}$$

• NTERUP $= 1$ : $\texttt{SMBR}^n = \texttt{TRAV}^n$

| | |
|---|---|
| $\texttt{ROVSDT}^n$ | $\dfrac{\rho}{\Delta t} - \theta\,\mathrm{div}(\rho\,\underline{u}) + \Gamma^n + \rho\,\underline{\underline{K}}_d^n + Max(-T_s^{imp}, 0)$ |
| $\texttt{TRAV}^n$ | $-\underline{\mathrm{grad}}\ p^{n-1+\theta} + (\rho - \rho_0)\underline{g}$<br>$+\underline{T}_s^{exp} - \rho\,\underline{\underline{K}}_e^n\,\underline{u}^n + \Gamma^n\,\underline{u}_i^n$<br>$-\underline{turb}^n + \mathrm{div}(\mu_{tot}\,{}^t\underline{\underline{\mathrm{grad}}}\,\underline{u}^n) + \dfrac{2}{3}\,\underline{\mathrm{grad}}\,(\mu_{tot}\,\mathrm{div}\dfrac{(\rho\underline{u})}{\rho})$<br>$-\rho\,\underline{\underline{K}}_d^n\,\underline{u}^n + T_s^{imp}\,\underline{u}^n + \mathrm{div}(\rho\,\underline{u})\,\underline{u}^n - \Gamma^n\,\underline{u}^n$ |

(II.14.23)

• NTERUP $> 1$ (sous-itération $k$) : $\texttt{SMBR}^n = \texttt{TRAVA}^n + \texttt{TRAV}^n$

| | |
|---|---|
| $\texttt{ROVSDT}^n$ | $\dfrac{\rho}{\Delta t} - \theta\,\mathrm{div}(\rho\,\underline{u}) + \Gamma^n + \rho\,\underline{\underline{K}}_d^n + Max(-T_s^{imp}, 0)$ |
| $\texttt{TRAVA}^n$ | $\underline{T}_s^{exp} - \rho\,\underline{\underline{K}}_e^n\,\underline{u}^n + \Gamma^n\,\underline{u}_i^n$<br>$-\underline{turb}^n + \mathrm{div}(\mu_{tot}\,{}^t\underline{\underline{\mathrm{grad}}}\,\underline{u}^n) + \dfrac{2}{3}\,\underline{\mathrm{grad}}\,(\mu_{tot}\,\mathrm{div}\dfrac{(\rho\underline{u})}{\rho})$<br>$-\rho\,\underline{\underline{K}}_d^n\,\underline{u}^n + T_s^{imp}\,\underline{u}^n + \mathrm{div}(\rho\,\underline{u})\,\underline{u}^n - \Gamma^n\,\underline{u}^n$ |
| $\texttt{TRAV}^n$ | $-\underline{\mathrm{grad}}\ (p^{n+\theta})^{(k-1)} + (\rho - \rho_0)\underline{g}$ |

(II.14.24)

## 14.4 Points à traiter

**• Prise en compte du terme $\underline{\mathrm{grad}}\ (\rho k)$ pour les modèles à viscosité turbulente**
Pour les modèles à viscosité turbulente, on calcule $\rho\ \underline{\mathrm{grad}}\ k$ au lieu de $\underline{\mathrm{grad}}\ (\rho k)$. Cette approximation, historique, provient du fait que les conditions aux limites de $\rho k$ ne sont pas directement accessibles, contrairement à celles de $k$.

**• Prise en compte de la diagonale de $\underline{\underline{K}}_{pdc}$**
Actuellement, dans le sous-programme utilisateur `uskpdc`, une mise en garde explicite est écrite, mais en commentaire. La partie diagonale $\underline{\underline{K}}_d$ du tenseur de pertes de charge $\underline{\underline{K}}_{pdc}$ peut donc intervenir systématiquement dans le calcul du coefficient $f_s^{imp}$, que sa contribution $K_d$ soit positive ou non, si l'utilisateur n'y prend garde. Un test de positivité sur les éléments de $\underline{\underline{K}}_d$ assurant une prise en compte correcte (contribution renforçant réellement la diagonale de la matrice globale) devrait être implanté.

**• Écriture de $\underline{\underline{EM}}$**
Dans la résolution procédant par incréments, il n'est pas indispensable à convergence que la viscosité utilisée pour l'écriture de l'opérateur $\mathcal{E}$ soit la même que celle prise en compte dans $\underline{\underline{EM}}$, matrice du système en incréments. Ainsi, en $R_{ij} - \varepsilon$, la viscosité totale utilisée dans $\underline{\underline{EM}}$ contient la viscosité moléculaire mais aussi la viscosité turbulente si l'on choisit l'option `IRIJNU = 1` , alors que dans $\mathcal{E}$ intervient seule la viscosité moléculaire. Cet ajout de la viscosité turbulente qui n'a pas de raison d'apparaître en $R_{ij} - \varepsilon$, a été hérité des pratiques mises en œuvre dans ESTET et N3S-EF pour renforcer la stabilité (lissage éventuel de l'incrément). Mais, ce n'est peut être pas le seul effet produit. En outre, cette pratique n'a pas aujourd'hui montré son absolue nécessité dans *Code_Saturne*. Par conséquent, une étude approfondie serait intéressante.

**• Résidu de normalisation de l'étape de pression**
On pourra vérifier le calcul du résidu de normalisation et en particulier l'utilisation des conditions aux limites de vitesse.

**• Calcul des pertes de charges**

Avec extrapolation des termes sources on a :

$$(\underline{\underline{K}}_e \underline{u})^{n+\theta_S} + \underline{\underline{K}}_d^n \ \underline{u}^{n+\theta}$$

Il serait aussi envisageable d'utiliser :

$$(\underline{\underline{K}}_e \underline{u})^{n+\theta_S} + \underline{\underline{K}}_d^{n+\theta_S} \ \underline{u}^{n+\theta}$$

# 15- Sous-programme `recvmc`

## 15.1 Fonction

Le but de ce sous-programme est de calculer la vitesse au centre des cellules à partir du flux de masse aux faces, par moindres carrés. Utilisée après l'étape de correction de pression (*cf.* `navsto`) cette méthode est une alternative à la technique de reconstruction à partir du gradient de l'incrément de pression (technique standard). Elle est activée quand l'indicateur `IREVMC` vaut 1 ou 2.

On rappelle que, à la fin de l'étape de correction de pression, le flux de masse aux faces vaut :

$$(\rho \underline{u})_{ij}^{n+1}.\underline{S}_{ij} = (\rho \underline{\widetilde{u}})_{ij}^{n+1}.\underline{S}_{ij} - \underline{\underline{D}}_{ij}(\Delta t^n, \delta P^{n+\theta}) + \mathrm{RC}_{ij} \tag{II.15.1}$$

où $\underline{\widetilde{u}}$ est la vitesse issue de l'étape de prédiction, $D_{ij}$ un opérateur de gradient aux faces et $\mathrm{RC}_{ij}$ le terme d'Arakawa (cf. `navsto` pour une définition précise des notations). Une première méthode, activée par `IREVMC` = 2, consiste à partir directement de $(\rho \underline{u})_{ij}^{n+1}.\underline{S}_{ij}$ pour calculer $\underline{u}^{n+1}$ par moindres carrés. Son utilisation a montré qu'elle semblait plus diffusive que la méthode standard (par exemple, dans le cas de la cavité entraînée) et pouvait conduire à des résultats erronés sur des maillages ne comportant pas uniquement des tétraèdres (ou des prismes à base triangulaire en "2D") et des pavés (hexaèdres orthogonaux).
On note que, dans la méthode ci-dessus, on est parti d'une vitesse $\underline{\widetilde{u}}$ au centre des cellules, qu'on a projetée aux faces pour obtenir le flux de masse, et qu'on ramène au centre des cellules par moindres carrés. Fort de cette constatation, une méthode alternative est disponible, activée par `IREVMC` = 1. Elle consiste à n'appliquer la méthode des moindres carrés qu'à la partie $-\underline{\underline{D}}_{ij}(\Delta t^n, \delta P^{n+\theta}) + \mathrm{RC}_{ij}$ du flux de masse et à rajouter directement $\underline{\widetilde{u}}$ (connu au centre des cellules) au résultat obtenu[1]. Cette méthode donne des résultats sensiblement meilleurs.

## 15.2 Discrétisation

Soit une cellule $\Omega_i$, $\phi_{ij}$ le flux de masse (total ou uniquement la partie en gradient de pression) à travers la face la séparant d'une cellule voisine $\Omega_j$ et $\phi_{b_ik}$ le flux de masse (total ou uniquement la partie en gradient de pression)à travers la face de bord $b_{ik}$. L'idéal serait de pouvoir trouver un vecteur $\underline{v}_i$ telle que, pour toute cellule voisine $\Omega_j$ on ait :

$$\rho_i \underline{v}_i.\underline{S}_{ij} = \phi_{ij} \tag{II.15.2}$$

et l'équivalent aux faces de bords, *i.e.* :

$$\rho_i \underline{v}_i.\underline{S}_{b_{ik}} = \phi_{b_{ik}} \tag{II.15.3}$$

Comme c'est généralement impossible d'obtenir les deux égalités précédentes[2], on va simplement chercher à minimiser la fonction $F_i$ :

$$F_i = \sum_{j \in Vois(i)} \left[\rho_i \underline{v}_i.\underline{S}_{ij} - \phi_{ij}\right]^2 + \sum_{k \in \gamma_b(i)} \left[\rho_i \underline{v}_i.\underline{S}_{b_{ik}} - \phi_{b_{ik}}\right]^2 \tag{II.15.4}$$

Pour ce faire, on dérive $F_i$ par rapport aux trois composantes du vecteur $\underline{v}_i$, et on résout le système $3 \times 3$ local qui résulte :

[1] cette dernière étape est faite dans `navsto`.
[2] sauf en incompressible pour des triangles en 2D et des tétraèdres en 3D

$$\underline{\underline{\mathcal{S}}}^{\,i} \begin{bmatrix} v_{i,x} \\ v_{i,y} \\ v_{i,z} \end{bmatrix} = \begin{bmatrix} \frac{1}{\rho_i}\left(\sum_{j\in Vois(i)} \phi_{ij} S_{ij,x} + \sum_{k\in\gamma_b(i)} \phi_{b_{ik}} S_{b_{ik},x}\right) \\ \frac{1}{\rho_i}\left(\sum_{j\in Vois(i)} \phi_{ij} S_{ij,y} + \sum_{k\in\gamma_b(i)} \phi_{b_{ik}} S_{b_{ik},y}\right) \\ \frac{1}{\rho_i}\left(\sum_{j\in Vois(i)} \phi_{ij} S_{ij,z} + \sum_{k\in\gamma_b(i)} \phi_{b_{ik}} S_{b_{ik},z}\right) \end{bmatrix} \tag{II.15.5}$$

avec $\underline{\underline{\mathcal{S}}}^{\,i}$ matrice carrée $3\times 3$ d'élément $S^{\,i}_{ml}$ courant défini par :

$$S^{\,i}_{ml} = \sum_{j\in Vois(i)} S_{ij,\,l}\, S_{ij,\,m} + \sum_{k\in\gamma_b(i)} S_{b_{ik},\,l}\, S_{b_{ik},\,m} \tag{II.15.6}$$

## 15.3 Mise en œuvre

Le flux de masse est passé par les arguments `FLUMAS` et `FLUMAB`.

**• Calcul de la matrice**
Les `NCEL` matrices $3\times 3$ sont stockées dans le tableau de travail `COCG`, de dimension $NCELET\times 3\times 3$. Ce dernier est d'abord mis à zéro, puis son remplissage se fait dans des boucles sur les faces internes et les faces de bord. La matrice étant symétrique, ces boucles ne servent qu'à remplir la partie triangulaire supérieure, le reste étant rempli par symétrie à la fin.

**• Inversion de la matrice**
On calcule les coefficients de la comatrice, puis l'inverse. Pour des questions de vectorisation, la boucle sur les `NCEL` éléments est remplacée par une série de boucles en vectorisation forcée sur des blocs de `NBLOC=1024` éléments. Le reliquat ($\texttt{NCEL} - E(\texttt{NCEL}/1024)\times 1024$) est traité après les boucles. À la fin, la matrice inverse est stockée dans `COCG` (toujours en utilisant sa propriété de symétrie).

**• Calcul du second membre et résolution**
Le second membre est stocké dans `BX`, `BY` et `BZ`. La vitesse finale est stockée dans `UX`, `UY` et `UZ`.

## 15.4 Points à traiter

**• Vectorisation forcée**
Le découpage en boucles de 1024 est-il vraiment nécessaire ? Les machines vectorielles et les compilateurs sont-ils aujourd'hui capables d'effectuer la vectorisation sans cette aide ? On note cependant que ce découpage en boucles de 1024 n'a pas de coût CPU supplémentaire, et un coût mémoire négligeable. Le seul inconvénient réside dans la complexité de l'écriture.

**• Suppression de la méthode `IREVMC` = 2**
Sur un maillage "1D" d'hexaèdres tous orthogonaux sauf une face, on peut montrer que la méthode fait apparaître une composante de vitesse aberrante non nulle et directement déterminée par l'angle de non orthogonalité de la face (non consistance). On pourrait donc songer à supprimer purement cette méthode, dans la mesure où elle n'est *a priori* consistante que sur une classe réduite de maillages.

# 16- Sous-programme `resolp`

## 16.1 Fonction

Dans ce sous-programme appelé dans `navsto`, on effectue l'étape de projection de la vitesse (ou de correction de pression). L'équation de quantité de mouvement (prédiction) est résolue dans `preduv` avec une pression totalement explicite. Il en résulte un champ de vitesse qui ne satisfait pas l'équation de continuité. Deux algorithmes de correction sont proposés :

1. L'algorithme que l'on appellera "couplage faible vitesse-pression". C'est un algorithme largement implanté dans les codes industriels. Il ne couple la vitesse et la pression qu'à travers le terme de masse (c'est l'algorithme proposé par défaut). C'est un algorithme de type *SIMPLEC* proche du *SIMPLE*. Ce dernier prend en compte, en plus du terme de masse, les diagonales simplifiées de la convection, de la diffusion et des termes source implicites.
2. L'algorithme de couplage vitesse-pression renforcé (option `IPUCOU = 1`). C'est un algorithme qui couple la vitesse et la pression à travers tous les termes (convection, diffusion et termes source implicites) de l'équation de quantité de mouvement sans pour autant être exact. Il permet en pratique de prendre de grands pas de temps sans découpler totalement la vitesse et la pression.

Si $\delta p$ est l'incrément de pression (*i.e.* $p^{n+1} = p^n + \delta p$) et $\widetilde{u}$ la vitesse issue de l'étape de prédiction, l'étape de projection revient d'un point de vue continu à résoudre une équation de Poisson :

$$\mathrm{div}(\underline{\underline{T}}^n \ \underline{\mathrm{grad}}\ \delta p) = \mathrm{div}(\rho\, \widetilde{u}) \tag{II.16.1}$$

et à corriger la vitesse :

$$\underline{u}^{n+1} = \underline{u}^n - \frac{1}{\rho}\, \underline{\underline{T}}^n \ \underline{\mathrm{grad}}\ \delta p \tag{II.16.2}$$

$\underline{\underline{T}}^n$ est un tenseur d'ordre 2 dont les termes sont homogènes à un pas de temps.

## 16.2 Discrétisation

L'étape de prédiction de la vitesse est décrite dans `preduv`. On adopte ici une notation d'opérateurs pour simplifier la compréhension des algorithmes implantés. L'équation discrète de quantité de mouvement s'écrit, à un instant de résolution intermédiaire, dans chaque direction d'espace $\alpha$ ($\alpha \in \{1,2,3\}$) :

$$\underline{\underline{M}}^n_\alpha \ \underline{\underline{R}}^n \ (\widetilde{\underline{V_\alpha}} - \underline{V_\alpha}^n) + \underline{\underline{A_\alpha}}^n \ \widetilde{\underline{V_\alpha}} = -\ \underline{\underline{G_\alpha}}\ \underline{P}^n + \underline{S_\alpha}^n + \underline{\underline{I}}_{s,\alpha}\ \widetilde{\underline{V_\alpha}} \tag{II.16.3}$$

- $\underline{\underline{M}}^n_\alpha$, de dimension `NCEL` $\times$ `NCEL`, est une matrice diagonale contenant le rapport du volume d'une cellule et du pas de temps local ( $\underline{\underline{M}}^n_\alpha(i,i) = \dfrac{|\Omega_i|}{\Delta t^n_{\alpha,I}}$ ), $\Delta t^n_{\alpha,I}$ représente le pas de temps dans la direction d'espace $\alpha$ à la cellule $\Omega_i$ et au pas de temps $n$,
- $\underline{\underline{R}}^n$, de dimension `NCEL` $\times$ `NCEL`, est la matrice diagonale contenant la masse volumique (on la sépare de la matrice de masse pour l'étape de projection de $\rho\underline{u}$). Par définition, $\underline{\underline{R}}^n(i,i) = \rho^n_I$ et l'apparition du vide étant exclue, cette matrice est naturellement inversible,

- $\star$ $\underline{\widetilde{V_\alpha}}$, de dimension `NCEL`, est un tableau où est stockée la $\alpha^{\text{ème}}$ composante de la vitesse prédite $\underline{\widetilde{u}}$,
- $\star$ $\underline{V_\alpha}^n$, de dimension `NCEL`, est un tableau où est stockée la $\alpha^{\text{ème}}$ composante de la vitesse $\underline{u}^n$ à l'instant précédent $n$,
- $\star$ $\underline{\underline{A_\alpha}}^n$ désigne l'opérateur de convection/diffusion (cet opérateur n'est pas forcément linéaire à cause des tests de pente éventuels dans le terme de convection, il peut dépendre de $\underline{V_\alpha}$),
- $\star$ $\underline{\underline{G_\alpha}}$ est l'opérateur linéaire[1] gradient "cellules" dans la direction $\alpha$ (il est donc appliqué à des vecteurs de dimension `NCEL`),
- $\star$ $\underline{P}^n$, de dimension `NCEL`, est un tableau où est stockée la pression $p^n$ du pas de temps précédent à chaque cellule,
- $\star$ $\underline{S_\alpha}^n$ est le tableau de dimension `NCEL` qui contient tous les termes source explicites (voir `preduv` pour plus de détails),
- $\star$ $\underline{\underline{I}}_{s,\alpha}$ est le tenseur diagonal relatif aux termes source implicites des composantes de la vitesse.

L'étape de correction consiste à imposer :

$$\text{div}(\,\rho\,\underline{u}\,) = \Gamma \tag{II.16.4}$$

où $\Gamma$ est un terme source de masse éventuel.

Soit $\underline{W}$ le tableau de dimension $3 \times$ `NCEL` qui contient toutes les composantes de la quantité de mouvement ($\underline{V}$ désigne $\underline{V}^n$, $\underline{V}^{n+1}$ ou $\underline{\widetilde{V}}$).

$$\underline{W} = \underline{\underline{R}}^n\ \underline{V} = \begin{pmatrix} \rho^n\ \underline{V_1} \\ \rho^n\ \underline{V_2} \\ \rho^n\ \underline{V_3} \end{pmatrix}$$

Soit $\underline{\underline{D}}$ l'opérateur de divergence. L'équation de continuité (II.16.4) s'écrit :

$$\underline{\underline{D}}\ \underline{W} = \underline{\Gamma}$$

$\underline{\Gamma}$ contient la valeur de $\Gamma$ au centre des cellules.

D'après l'équation discrète (II.16.3), on écrit pour tout $\alpha \in \{1,2,3\}$ :

$$(\underline{\underline{M}}_\alpha^n + \underline{\underline{A_\alpha}}^n\ (\underline{\underline{R}}^n)^{-1} - \underline{\underline{I}}_{s,\alpha}\ (\underline{\underline{R}}^n)^{-1})\ \underline{\underline{R}}^n\ \underline{\widetilde{V_\alpha}} = -\ \underline{\underline{G_\alpha}}\ \underline{P}^n + \underline{S_\alpha}^n + \underline{\underline{M_\alpha}}\ \underline{\underline{R}}^n\ \underline{V_\alpha}^n \tag{II.16.5}$$

On pose :

$$\underline{\underline{B_\alpha}} = \underline{\underline{M}}_\alpha^n + \underline{\underline{A_\alpha}}^n\ (\underline{\underline{R}}^n)^{-1} - \underline{\underline{I}}_{s,\alpha}\ (\underline{\underline{R}}^n)^{-1}$$

$$\underline{\underline{B}} = \begin{pmatrix} \underline{\underline{B_1}} & 0 & 0 \\ 0 & \underline{\underline{B_2}} & 0 \\ 0 & 0 & \underline{\underline{B_3}} \end{pmatrix}$$

$$\underline{\underline{G}} = \begin{pmatrix} \underline{\underline{G_1}} \\ \underline{\underline{G_2}} \\ \underline{\underline{G_3}} \end{pmatrix}$$

$$\underline{S}^n = \begin{pmatrix} \underline{S_1}^n + \underline{\underline{M}}_1\ \underline{\underline{R}}^n\ \underline{V_1}^n \\ \underline{S_2}^n + \underline{\underline{M}}_2\ \underline{\underline{R}}^n\ \underline{V_2}^n \\ \underline{S_3}^n + \underline{\underline{M}}_3\ \underline{\underline{R}}^n\ \underline{V_3}^n \end{pmatrix}$$

[1] en toute rigueur, à cause des limitations éventuelles de gradient qui peuvent être activée par l'utilisateur, cet opérateur peut ne plus être vraiment linéaire

On peut donc écrire :

$$\underline{\underline{B}}\ \underline{\widetilde{W}} = -\ \underline{\underline{G}}\ \underline{P}^{\,n} + \underline{S}^{\,n} \tag{II.16.6}$$

La méthode des pas fractionnaires se décompose en deux étapes :

1. résolution de l'équation (II.16.3) (cette équation nous donne l'équation(II.16.6)), soit :

$$\underline{\underline{M}}^{n}_{\alpha}\ \underline{\underline{R}}^{n}\ (\underline{\widetilde{V_\alpha}} - \underline{V_\alpha}^{\,n}) + \underline{\underline{A_\alpha}}^{\,n}\ \underline{\widetilde{V_\alpha}} - \underline{\underline{I}}_{s,\alpha}\ \underline{\widetilde{V_\alpha}} = -\ \underline{\underline{G_\alpha}}\ \underline{P}^{\,n} + \underline{S_\alpha}^{\,n} \tag{II.16.7}$$

2. soustraction[2] de l'équation de quantité de mouvement prise à l'instant $(n+1)$ de (II.16.7) :

$$\underline{\underline{M}}^{n}_{\alpha}\ \underline{\underline{R}}^{n}\ (\underline{V_\alpha}^{\,n+1} - \underline{\widetilde{V_\alpha}}) + \underline{\underline{A_\alpha}}^{\,n}\ (\underline{V_\alpha}^{\,n+1} - \underline{\widetilde{V_\alpha}}) - \underline{\underline{I}}_{s,\alpha}\ (\underline{V_\alpha}^{\,n+1} - \underline{\widetilde{V_\alpha}}) = -\ \underline{\underline{G_\alpha}}\ (\underline{P}^{\,n+1} - \underline{P}^{\,n}) \tag{II.16.8}$$

L'équation (II.16.8) nous donne :

$$\underline{\underline{B}}\ (\underline{W}^{n+1} - \underline{\widetilde{W}}) = -\ \underline{\underline{G}}\ (\underline{P}^{\,n+1} - \underline{P}^{\,n}) \tag{II.16.9}$$

avec :

$$\underline{W}^{\,n+1} = \underline{\underline{R}}^{n}\ \underline{V}^{\,n+1}$$

Or :

$$\underline{\underline{D}}\ \underline{W}^{\,n+1} = \underline{\Gamma}$$

D'où, en posant $\delta\underline{P} = \underline{P}^{\,n+1} - \underline{P}^{\,n}$,

$$\underline{\underline{D}}\ \underline{\underline{B}}^{-1}\ \underline{\underline{G}}\ \delta\underline{P} = \underline{\underline{D}}\ \underline{\widetilde{W}} - \underline{\Gamma} \tag{II.16.10}$$

Il nous reste donc à inverser le système (II.16.10) pour déterminer $\delta p$ (donc $p^{\,n+1}$) et corriger la vitesse pour obtenir $\underline{u}^{\,n+1}$. La correction de la vitesse se fait dans `navsto`, soit en incrémentant la vitesse par le gradient de l'incrément de pression $\delta p$ calculé, soit en reconstruisant la vitesse à partir du flux de masse actualisé par une méthode de moindres carrés (en appelant le sous-programme `recvmc`).

Le problème provient tout d'abord du calcul de $\underline{\underline{B}}^{-1}$. En effet, il a été jugé trop coûteux de calculer l'inverse de $\underline{\underline{B}}$. Les algorithmes "couplage faible vitesse-pression" et celui du "couplage vitesse-pression renforcé" correspondent à une approximation de cet opérateur.

Dans le cas de l'algorithme "couplage faible vitesse-pression", on suppose $\underline{\underline{B}}^{-1}_{\alpha} = \underline{\underline{M}}^{-1}_{\alpha}$ (on pourrait inclure aussi les termes diagonaux issus de la convection, de la diffusion et des termes source implicites).

Pour le "couplage vitesse-pression renforcé", on inverse le système[3] $\underline{\underline{B}}\ \underline{T} = \underline{\Omega}$ et on pose $\underline{\underline{B}}^{-1}_{\alpha} = \mathrm{diag}(T_\alpha)$. Cette étape a lieu dans le sous-programme `preduv`.

Les écritures à l'aide des opérateurs posent un inconvénient majeur avec la discrétisation colocalisée. En effet, l'opérateur[4] $\underline{\underline{D}}\ \underline{\underline{B}}^{-1}\ \underline{\underline{G}}$ peut découpler les nœuds pairs et impairs sur un maillage cartésien et régulier[5]. Pour éviter ce problème, on utilise l'opérateur $\underline{\underline{L}}$ (issu naturellement de la formulation volumes finis colocalisée de l'opérateur div( $\underline{\mathrm{grad}}$ )) défini en chaque cellule[6] $\Omega_i$ de centre $I$ par[7]:

$$(\underline{\underline{L}}\ \delta\underline{P})_I = \sum_{j \in Vois(i)} [\ \underline{\underline{T}}^{\,n}_{ij}\ (\underline{\mathrm{grad}}\ \delta p)_{f_{ij}}\ ] \,.\, \underline{S}_{ij} + \sum_{k \in \gamma_b(i)} [\ \underline{\underline{T}}^{\,n}_{b_{ik}}\ (\underline{\mathrm{grad}}\ \delta p)_{f_{b_{ik}}}\ ] \,.\, \underline{S}_{b_{ik}} \tag{II.16.11}$$

[2] On néglige la variation des termes sources explicites $\underline{S_\alpha}^{\,n}$ qui sont toujours estimés en $n$.

[3] $\underline{\Omega} = (|\Omega_1|, ..., |\Omega_{\texttt{NCEL}}|, |\Omega_1|, ..., |\Omega_{\texttt{NCEL}}|, |\Omega_1|, ..., |\Omega_{\texttt{NCEL}}|)$.

[4] On insiste sur le fait que l'opérateur $\underline{\underline{G}}$ est l'opérateur "gradients cellules" appliqué à la pression explicite lors de la prédiction de la vitesse.

[5] Si $u_i$ est la valeur d'une variable aux centres des cellules sur un maillage cartésien 1D, le Laplacien de $u$ calculé par cet opérateur en $i$ s'écrit : $\dfrac{u_{i-2} + 2u_i - u_{i+2}}{4h^2}$, où $h$ est le pas d'espace. D'où le découplage des cellules.

[6] On rappelle que $Vois(i)$ est l'ensemble des centres des cellules voisines de $\Omega_i$ et $\gamma_b(i)$ l'ensemble des centres éventuels des faces de bord de $\Omega_i$.

[7] Si $u_i$ est la valeur d'une variable aux centres des cellules sur un maillage cartésien 1D, le Laplacien de $u$ par ce dernier opérateur en $i$ s'écrit : $\dfrac{u_{i-1} + 2u_i - u_{i+1}}{4h^2}$, où $h$ est le pas d'espace.

$\underline{\underline{T}}^n$ un tenseur diagonal d'ordre 2 contenant les pas de temps dans les trois directions d'espace et $\underline{S}_{ij}$ (respectivement $\underline{S}_{b_{ik}}$) le vecteur surface de la face purement interne $ij$ (resp. de la face de bord $ik$). Le gradient $(\underline{\text{grad}}\ \delta p)_f$ présent dans l'équation (II.16.11) est un gradient facette[8].

D'un point de vue continu, on peut écrire[9]:

$$(\underline{\underline{L}}\ \underline{P}^{n+1})_I = \int_{\Omega_i} \text{div}(\ \underline{\underline{T}}^n\ \underline{\text{grad}}\ p^{n+1})\ d\Omega$$

L'opérateur $\underline{\underline{L}}$ ne pose pas de problème de découplage pair/impair sur les maillages cartésiens et réguliers.

Avec l'algorithme "couplage faible vitesse-pression" : $\underline{\underline{T}}_I^n = \begin{pmatrix} \Delta t_I^n & 0 & 0 \\ 0 & \Delta t_I^n & 0 \\ 0 & 0 & \Delta t_I^n \end{pmatrix}$

et avec l'algorithme "couplage vitesse-pression renforcé" : $\underline{\underline{T}}_I^n = \begin{pmatrix} T_{11,I}^n & 0 & 0 \\ 0 & T_{22,I}^n & 0 \\ 0 & 0 & T_{33,I}^n \end{pmatrix}$

La matrice du système de pression, dans le cas de l'utilisation du "couplage vitesse-pression renforcé", n'est pas facilement calculable à cause du terme :

$$[\ \underline{\underline{T}}_{ij}^n\ (\underline{\text{grad}}\ \delta p)_{f_{ij}}]\cdot\underline{S}_{ij}$$

pour une face interne $ij$ et du terme :

$$[\ \underline{\underline{T}}_{b_{ik}}^n\ (\underline{\text{grad}}\ \delta p)_{f_{b_{ik}}}]\cdot\underline{S}_{b_{ik}}$$

pour une face de bord.
En effet, à cause du changement de la viscosité suivant la direction de l'espace du fait de l'anisotropie de $\underline{\underline{T}}^n$, il faudrait calculer les gradients directionnels en fonction du gradient normal.

On pose pour tout scalaire $a$, sans préciser pour le moment la nature du gradient $\underline{\text{grad}}$ :

$$\underline{\widetilde{G}}\ a = \underline{\underline{T}}^n\ \underline{\text{grad}}\ a = \begin{pmatrix} T_{11}^n\ \dfrac{\partial a}{\partial x} \\ T_{22}^n\ \dfrac{\partial a}{\partial y} \\ T_{33}^n\ \dfrac{\partial a}{\partial z} \end{pmatrix}$$

On peut également définir :

$$\left[(\underline{\widetilde{G}}\ a)_{cell}\right]_{ij}\cdot\underline{S}_{ij} \overset{déf}{=} \left[\underline{\underline{T}}^n\ (\underline{\text{grad}}\ a)\right]_{ij}\cdot\underline{S}_{ij}$$

où $\underline{\text{grad}}\ a$ est le gradient cellule classique de $a$.

De même , on pose :

$$\left[(\underline{\widetilde{G}}\ a)_f\right]_{ij}\cdot\underline{S}_{ij} \overset{déf}{=} \left[\underline{\underline{T}}^n\ ((\underline{\text{grad}}\ a))_f\right]_{ij}\cdot\underline{S}_{ij}$$

où $(\underline{\text{grad}}\ a)_f$ est le gradient facette de $a$.

On doit calculer $\underline{\widetilde{G}}\ (\delta p)\cdot\underline{S}$ à la face.
S'il s'agit d'un gradient cellule, cela n'engendre aucune difficulté, les composantes du gradient cellule étant toutes parfaitement calculables.

---

[8] Sur maillage orthogonal, $(\underline{\text{grad}}\ p)_{f_{ij}}\cdot\underline{S}_{ij} = \dfrac{p_J - p_I}{\overline{IJ}} S_{ij}$. $I$ (resp. $J$) et $I'$ (resp. $J'$) sont en effet superposés.

[9] Dans le cas de l'algorithme "couplage faible vitesse-pression", $\widetilde{T}_I^n = \Delta t_I^n$.

Par contre, s'il s'agit d'un gradient facette, seule la décomposition sur la normale à la face est exploitable[10]. On a besoin, explicitement et séparément, des quantités $\frac{\partial(\delta p)}{\partial x}$, $\frac{\partial(\delta p)}{\partial y}$ et $\frac{\partial(\delta p)}{\partial z}$ ce qui rend difficile l'utilisation du gradient normal de l'incrément de la pression.
On fait donc une approximation sur le gradient de l'incrément de la pression en supposant qu'il est égal à sa composante normale[11], c'est à dire :

$$(\underline{\text{grad}}\ (\delta p))_{f} \approx ((\underline{\text{grad}}\ (\delta p))_{f} \cdot \underline{n})\ \underline{n} \tag{II.16.12}$$

où $\underline{n}$ est la normale extérieure unitaire. On a donc :

$$\widetilde{\underline{G}}\ (\delta p) \approx ((\underline{\text{grad}}\ (\delta p))_{f} \cdot \underline{n})\ (\underline{\underline{T}}^{n}\ \underline{n})$$

On peut donc se ramener au cas d'un pas de temps scalaire $\widetilde{T}^{n}_{ij}$ en posant :

$$\widetilde{T}^{n}_{ij} = (\underline{\underline{T}}^{n}_{ij}\ \underline{n}) \cdot \underline{n} \tag{II.16.13}$$

En effet :

$$\begin{aligned} [\ \underline{\underline{T}}^{n}_{ij}\ (\underline{\text{grad}}\ \delta p)_{f_{ij}}] \cdot \underline{S}_{ij} &\approx (\underline{\underline{T}}^{n}_{ij}\ \underline{n}) \cdot \underline{S}_{ij}\ (\underline{\text{grad}}\ (\delta p)_{f_{ij}} \cdot \underline{n}) \\ &= (\underline{\underline{T}}^{n}_{ij}\ \underline{n}) \cdot \underline{n}\ (\underline{\text{grad}}\ (\delta p)_{f_{ij}} \cdot \underline{n})\ S_{ij} \\ &= \widetilde{T}^{n}_{ij}\ (\underline{\text{grad}}\ (\delta p)_{f_{ij}} \cdot \underline{n})\ S_{ij} \end{aligned} \tag{II.16.14}$$

or :

$$\begin{aligned} \widetilde{T}^{n}_{ij}\ (\underline{\text{grad}}\ (\delta p)_{f_{ij}} \cdot \underline{n})\ S_{ij} &= \widetilde{T}^{n}_{ij}\ \underline{\text{grad}}\ (\delta p)_{f_{ij}} \cdot \underline{S}_{ij} \\ &= \widetilde{T}^{n}_{ij} \left[ P_J - P_I + (\ \underline{JJ'} - \underline{II'}\ ) \frac{1}{2}\ (\underline{\text{grad}}\ P_I + \underline{\text{grad}}\ P_J\ ) \right] \frac{S_{ij}}{\overline{I'J'}} \end{aligned} \tag{II.16.15}$$

Ceci intervient lors de la reconstruction des gradients au second membre du système final à résoudre, par appel aux sous-programmes `itrmas` et `itrgrp`.
En fait, l'approximation $(\widetilde{T}^{n}_{ij} = (\underline{\underline{T}}^{n}_{ij}\ \underline{n}) \cdot \underline{n})$ n'y est pas utilisée. En effet, puisqu'on calcule un gradient directionnel de l'incrément de pression avec le sous-programme `grdcel`, on se permet d'utiliser le tenseur $\underline{\underline{T}}$ pour corriger le terme $[\ \underline{\underline{T}}^{n}_{ij}\ (\underline{\text{grad}}\ \delta p)_{f_{ij}}] \cdot \underline{S}_{ij}$ .
En pratique, on le discrétise, pour une face interne $ij$, par[12] :

$$[\ \underline{\underline{T}}^{n}_{ij}\ (\underline{\text{grad}}\ \delta p)_{f_{ij}}] \cdot \underline{S}_{ij} = \left[ \widetilde{T}^{n}_{ij}\ \frac{P_J - P_I}{\overline{I'J'}} + \frac{\overline{JJ'} - \overline{II'}}{\overline{I'J'}}\ \frac{1}{2}\ (\underline{\underline{T}}^{n}_{I}\ \underline{\text{grad}}\ P_I + \underline{\underline{T}}^{n}_{J}\ \underline{\text{grad}}\ P_J\ ) \right] S_{ij} \tag{II.16.16}$$

au lieu de (II.16.15). On procède de même pour les termes de bord.
Par la suite, on notera quand même cette expression $\widetilde{T}^{n}_{ij}\ (\underline{\text{grad}}\ (\delta p)_{f_{ij}} \cdot \underline{S}_{ij})$.
Il reste le terme $\underline{\underline{D}}\ \widetilde{\underline{W}}$, qui *via* les gradients cellule de la pression présents dans l'équation de quantité de mouvement, découple les cellules paires et impaires sur une grille cartésienne. On utilise pour éviter ce problème un filtre de type Rhie & Chow qui permet de lisser la contribution de la pression de l'équation de quantité de mouvement. On écrit :

$$\begin{aligned} (\underline{\underline{D}}\ \widetilde{\underline{W}})_I = & \sum_{j \in Vois(i)} [\rho^{n} \widetilde{\underline{u}} + \alpha_{Arak}\ (\widetilde{\underline{G}}\ (p^{n}))_{cell}]_{f_{ij}} \cdot \underline{S}_{ij} - \alpha_{Arak} \sum_{j \in Vois(i)} \widetilde{T}^{n}_{ij}\ (\underline{\text{grad}}\ p^{n})_{f_{ij}} \cdot \underline{S}_{ij} \\ & + \sum_{k \in \gamma_b(i)} [\rho^{n} \widetilde{\underline{u}} + \alpha_{Arak}\ (\widetilde{\underline{G}}\ (p^{n}))_{cell}]_{f_{b_{ik}}} \cdot \underline{S}_{b_{ik}} - \alpha_{Arak} \sum_{k \in \gamma_b(i)} \widetilde{T}^{n}_{b_{ik}}\ (\underline{\text{grad}}\ p^{n})_{f_{b_{ik}}} \cdot \underline{S}_{b_{ik}} \\ & = \sum_{j \in Vois(i)} \widetilde{m}_{ij} + \sum_{k \in \gamma_b(i)} \widetilde{m}_{b_{ik}} \end{aligned} \tag{II.16.17}$$

[10] D'après la formule $\frac{p_{J'} - p_{I'}}{\overline{I'J'}} S_{ij}$.

[11] *i.e.* on ne prend pas en compte la composante tangentielle $((\underline{\text{grad}}\ (\delta p))_{f} \cdot \underline{\tau}).\underline{\tau}$, $\underline{\tau}$ étant un vecteur tangentiel unitaire.

[12] Le coefficient $\frac{1}{2}$ dans (II.16.16) est mis pour des raisons de stabilité. En effet, l'utilisation des coefficients de pondération aux faces serait plus exact mais probablement moins stable.

$\widetilde{m}_{ij}$ (resp. $\widetilde{m}_{b_{ik}}$) est le flux de masse à la face purement interne $ij$ (resp.à la face de bord $ik$) modifié par le filtre Rhie & Chow.
$(\underline{\text{grad}}\ p^n)_{f_{ij}}$ est un gradient facette alors que $[\rho^n \underline{\widetilde{u}} + \alpha_{Arak}\ (\underline{\widetilde{G}}\ (p^n))_{cell}]_{f_{ij}}$ est une valeur à la face interpolée à partir des valeurs estimées aux cellules (le gradient de pression dans ce terme est un gradient cellule). Cette précision s'applique également aux termes de bord.
Dans *Code_Saturne*, $\alpha_{Arak}$ est appelé pour des raisons historiques "le coefficient d'Arakawa". Il est noté `ARAK`.

Il faut remarquer, encore une fois, que pour le filtre Rhie & Chow, le tenseur $\underline{\underline{T}}^n$ est utilisé dans le terme volumique $\underline{\widetilde{G}}$ alors que l'approximation (II.16.13) est appliquée lors du calcul du gradient facette.

Finalement, d'après (II.16.10) et (II.16.11), on résout :

$$\sum_{j\in Vois(i)} \widetilde{T}^n_{ij}\ (\underline{\text{grad}}\ (\delta p))_{f_{ij}} \cdot \underline{S}_{ij} + \sum_{k\in\gamma_b(i)} \widetilde{T}^n_{b_{ik}}\ (\underline{\text{grad}}\ (\delta p))_{f_{b_{ik}}} \cdot \underline{S}_{b_{ik}} = \sum_{j\in Vois(i)} \widetilde{m}_{ij} + \sum_{k\in\gamma_b(i)} \widetilde{m}_{b_{ik}} - \Gamma_I \tag{II.16.18}$$

Pour la prise en compte des non orthogonalités, on résout ce système linéaire par incrément, en ajoutant au second membre les termes de reconstruction. Si $\delta(\delta p)$ est l'incrément de l'incrément (l'incrément de la variable $\delta p$ à calculer) et $k$ l'indice d'itération sur le point fixe, on résout exactement :

$$\begin{aligned}
\sum_{j\in Vois(i)} \widetilde{T}^n_{ij}\ \frac{(\delta(\delta p))_I^{k+1} - (\delta(\delta p))_J^{k+1}}{\overline{I'J'}}\ S_{ij} + \sum_{k\in\gamma_b(i)} \widetilde{T}^n_{b_{ik}}\ \frac{(\delta(\delta p))_I^{k+1} - (\delta(\delta p))_{b_{ik}}^{k+1}}{\overline{I'F}}\ S_{b_{ik}} \\
= \sum_{j\in Vois(i)} \widetilde{m}_{ij} + \sum_{k\in\gamma_b(i)} \widetilde{m}_{b_{ik}} \\
- \sum_{j\in Vois(i)} \widetilde{T}^n_{ij}\ (\underline{\text{grad}}\ (\delta p)^k)_{f_{ij}} \cdot \underline{S}_{ij} - \sum_{k\in\gamma_b(i)} \widetilde{T}^n_{b_{ik}}\ (\underline{\text{grad}}\ (\delta p)^k)_{f_{b_{ik}}} \cdot \underline{S}_{b_{ik}} - \Gamma_I \\
= \sum_{j\in Vois(i)} m^k_{ij} + \sum_{k\in\gamma_b(i)} m^k_{b_{ik}} - \Gamma_I
\end{aligned} \tag{II.16.19}$$

avec :

$$\left\{ \begin{array}{ll} (\delta(\delta p))^0 & = 0 \\ (\delta(\delta p))^{k+1} & = (\delta p)^{k+1} - (\delta p)^k \quad \forall k \in \mathbb{N} \end{array} \right. \tag{II.16.20}$$

Ce sont les gradients facette $(\underline{\text{grad}}\ (\delta p)^k)_{f_{ij}}$ et $(\underline{\text{grad}}\ (\delta p)^k)_{f_{b_{ik}}}$ qui seront reconstruits.

## 16.3 Mise en œuvre

On expose dans ce qui suit l'algorithme tel qu'il est écrit dans `resolp`.

$\underline{T}^n$ désignera un tableau de dimension 3 contenant les pas de temps locaux dans chaque direction (pour l'utilisation du "couplage vitesse-pression renforcé"). Dans le cas de l'algorithme "couplage faible vitesse-pression", on garde cette notation, les pas de temps étant égaux dans les trois directions de l'espace.

**• Calcul de la matrice du système à résoudre**

- calcul du coefficient de diffusion aux faces pour le Laplacien de pression (le coefficient de diffusion utilise le pas de temps de calcul ou celui du "couplage vitesse-pression renforcé"). Deux cas se présentent suivant l'algorithme de couplage vitesse/pression choisi par l'utilisateur :

  1. Appel de `viscfa` avec une viscosité totale égale au pas de temps $\Delta t^n_I$ pour l'algorithme "couplage faible vitesse-pression" (`IPUCOU` = 0),
  2. Appel de `visort` avec une viscosité totale diagonale pour l'algorithme du "couplage vitesse-pression renforcé" (`IPUCOU` = 1). C'est à ce niveau qu'est calculé $\widetilde{T}^n_{ij}$. Les pas de temps équivalents calculés auparavant dans la subroutine `preduv` sont stockés dans le tableau `TPUCOU`.

- Appel de `matrix` pour la construction de la matrice de diffusion de la pression (sans les termes de reconstruction qui ne peuvent pas être pris en compte si on veut garder une structure de matrice creuse) avec la viscosité calculée précédemment et le tableau `COEFB` des conditions aux limites de la pression $p^n$ (on impose une condition de Neumann homogène (resp. Dirichlet homogène) sur $\delta p$ pour une condition de Neumann (resp. Dirichlet) pour $p$).

**• Calcul du résidu de normalisation `RNORMP`**

Cette étape est réalisée directement dans `preduv` à partir de la version 1.1.0.s, et le résidu de normalisation est transmis par l'intermédiaire de la variable RNORMP(IPHAS).

Le tableau `TRAV` contient à ce niveau de *Code_Saturne* le second membre issu de `preduv` sans les termes source utilisateur. Le calcul de `RNORMP` se fait de la façon suivante :

1. $\texttt{TRAV}(I) = \widetilde{\underline{u}}_I - \dfrac{\Delta t_I^n}{\rho_I \ |\Omega_i|}\texttt{TRAV}(I) + \dfrac{(\rho_I - \rho_0)\Delta t_I^n}{\rho_I}\underline{g}$,
2. Appel de `inimas` pour calculer le flux de masse du vecteur `TRAV` (on calcule à chaque face $\rho_{ij}\ \texttt{TRAV}_{ij} \cdot \underline{S}_{ij}$, où $\underline{S}$ est le vecteur surface). On impose le nombre de *sweeps* (ou d'itérations) total à 1 ($\texttt{NSWRP} = 1$), ce qui signifie que l'on ne recontruit pas les gradients lors de ce passage dans `inimas` (pour des raisons de gain de temps de calcul). Les tableaux des conditions aux limites qui sont passés dans `inimas` sont ceux de la vitesse $\underline{u}^n$.
3. Appel de `divmas` pour calculer la divergence par cellule du flux de masse précédent. La divergence est stockée dans le tableau de travail `W1`.
4. Les termes source de masse stockés dans le tableau `SMACEL` sont ajoutés à `W1`.
$$\texttt{W1}(I) = \texttt{W1}(I) - \frac{|\Omega_i|}{\rho_I}\texttt{SMACEL}(I) \tag{II.16.21}$$
5. Appel de `prodsc` ($\texttt{RNORMP} = \sqrt{\texttt{W1.W1}}$). `RNORMP` servira dans le test d'arrêt du solveur de pression pour normaliser le résidu (voir la routine `gradco` pour l'inversion par gradient conjugué).

**• préparation de la résolution du système**

- Appel de `grdcel` pour le calcul du gradient de la pression $p^n$. Le résultat est stocké dans `TRAV`. `TRAV` contient donc à ce niveau $\dfrac{\partial p^n}{\partial x}$, $\dfrac{\partial p^n}{\partial y}$, $\dfrac{\partial p^n}{\partial z}$.
- Introduction du gradient cellule de la pression explicite $p^n$ pour l'utilisation du filtre Rhie & Chow.
$$\texttt{TRAV}(I) = \widetilde{\underline{u}}_I + \frac{\texttt{ARAK}}{\rho_I}\ \underline{\underline{T}}_I^n\ \underline{\text{grad}}\ p^n{}_I$$
`ARAK` représente le coefficient dit "d'Arakawa" que l'utilisateur peut modifier dans `usini1` et qui vaut 1 par défaut. Pour simplifier les notations, on pose $\texttt{ARAK} = \alpha_{Arak}$.
- Appel de `inimas` pour calculer le flux de masse de `TRAV`. Les conditions aux limites appliquées sont celles de la vitesse (voir sous-programme `navsto`). Ceci reste une approximation des conditions aux limites de `TRAV`. Le flux de masse est donc égal à (voir sous-programme `inimas` pour plus de détails sur le calcul aux faces de bord) :
$$m_{ij} = \left[\rho\widetilde{\underline{u}} + \alpha_{Arak}\ \underline{\underline{T}}^n\ \underline{\text{grad}}\ (p^n)\right]_{f_{ij}} \cdot \underline{S}_{ij}$$

- Appel de `itrmas` pour incrémenter le flux de masse aux faces[13] de

$$-\alpha_{Arak}\ \widetilde{T}^{n}_{ij}\ (\underline{\text{grad}}\ p^{n})_{f_{ij}} \cdot \underline{S}_{ij}.$$

- Appel de `clmlga` pour l'utilisation d'un algorithme de multigrille algébrique pour l'inversion de la matrice de pression.
- initialisation de $\delta p$, $\delta(\delta p)$ et `SMBR` à 0. `SMBR` servira à stocker le second membre. Dans le code, `RTP(*,IPRIPH)` et `DRTP` contiennent respectivement $\delta p$ et $\delta(\delta p)$.
- Appel de `divmas` pour le calcul de la divergence du flux de masse issu du dernier appel de `itrmas`. Cette divergence est stockée dans le tableau de travail `W7`.
- Ajout des contributions des termes source de masse[14] à `W7`.

$$\texttt{W7}(I) = \texttt{W7}(I) - |\Omega_i|\ \texttt{SMACEL}(I) \tag{II.16.22}$$

**• Boucle sur les non orthogonalités**

Une seule inversion permettrait de résoudre le problème en supposant que le maillage soit orthogonal. On décrit ci-dessous la boucle sur les non orthogonalités.

- début de la boucle sur $k$ (dans ce qui suit, on est en $k+1$)
  - ⋆ Actualisation de `SMBR` au début de la boucle[15].

$$\texttt{SMBR}(I) = -\texttt{W7}(I) - \texttt{SMBR}(I)$$

  - ⋆ Calcul de la norme de `SMBR` dans `prodsc`. On l'appelle `RESIDU`. Comme on résout le système par incrément, le second membre doit s'annuler à convergence.
  - ⋆ Si `RESIDU` $< 10 \times \varepsilon \times$ `RNORMP`, la convergence est atteinte[16].
    - ⇒ Appel de `itrmas` pour la réactualisation du flux de masse avec le gradient facette $(\underline{\text{grad}}\ (\delta p)^{k})_{f}$. On calcule donc à chaque face $\widetilde{T}^{n}_{ij}\ (\underline{\text{grad}}\ (\delta p)^{k})_{f_{ij}} \cdot \underline{S}_{ij}$ et $\widetilde{T}^{n}_{b\,ik}\ (\underline{\text{grad}}\ (\delta p)^{k})_{f_{b\,ik}} \cdot \underline{S}_{b\,ik}$.
    - ⇒ réactualisation[17] de la pression $p^{n+1} = p^{n} + \sum_{l=1}^{l=k} (\delta(\delta p))^{l}$.
  - ⋆ Sinon,
    - ⇒ $(\delta(\delta p))^{k+1} = 0$,
    - ⇒ Appel de `invers` pour l'inversion du système (II.16.19). Le test d'arrêt de l'algorithme d'inversion `RESIDU` est normalisé par `RNORMP` (voir `gradco` pour l'inversion de la pression).
  - ⋆ Si on atteint le nombre de *sweeps* maximal,
    - ⇒ Appel de `itrmas` pour l'incrémentation du flux de masse par le gradient de la pression $(\delta p)^{k}$.
    - ⇒ Second appel de `itrmas` pour l'incrémentation du flux de masse avec le gradient de $(\delta(\delta p))^{k+1}$ non reconstruit pour assurer une divergence finale nulle, ceci pour rester cohérent avec la matrice de pression qui ne prend pas en compte les non orthogonalités[18].

---

[13] $(\underline{\text{grad}}\ p^{n})_{f_{ij}} \cdot \underline{S}_{ij}$ est le gradient normal à la face égal, sur un maillage orthogonal, à $\dfrac{p^{n}_{J} - p^{n}_{I}}{\overline{IJ}} S_{ij}$.

[14] Le tableau `W7` contient le second membre sans le gradient de $\delta p$. Il reste donc inchangé à chaque *sweep*. En revanche, `SMBR` contient le second membre complet qui varie à chaque *sweep*.

[15] Le signe "-" résulte de la construction de la matrice.

[16] $\varepsilon$ est la tolérance associée à la pression qui peut être modifiée par l'utilisateur dans `usini1`, *via* le tableau `EPSILO`.

[17] $(\delta p)^{k} = \sum_{l=1}^{l=k} (\delta(\delta p))^{l}$ est stockée dans `RTP(*,IPRIPH)`.

[18] On pourra se référer au sous-programme `navsto` pour plus de détails.

$\Rightarrow$ mise à jour de l'incrément de pression $(\delta p)^{k+1} = (\delta p)^k + (\delta(\delta p))^{k+1}$.

$\star$ Sinon,

$\Rightarrow$ Incrémentation du flux de masse en prenant en compte un coefficient de relaxation. $(\delta p)^{k+1} = (\delta p)^k + \texttt{RELAX} \times (\delta(\delta)p)^{k+1}$. Le coefficient de relaxation est mis par défaut à 1 et peut être modifié dans `usini1`.

$\Rightarrow$ Appel de `itrgrp` pour le calcul de la partie en $\underline{\underline{T}}^n$ $\underline{\text{grad}}\ (\delta p)$ du second membre (auquel sera ajouté le tableau `W7` en début de boucle).

$$\texttt{SMBR}(I) = \sum_{j \in Vois(i)} \widetilde{T}_{ij}^{\,n}\ (\underline{\text{grad}}\ (\delta p)^k)_{f_{ij}} \cdot \underline{S}_{ij} + \sum_{k \in \gamma_b(i)} \widetilde{T}_{b_{ik}}^{\,n}\ (\underline{\text{grad}}\ (\delta p)^k)_{f_{b_{ik}}} \cdot \underline{S}_{b_{ik}}$$

- fin de la boucle

**• réactualisation de la pression**

On réactualise la pression par la somme des incréments de $\delta p$.

$$p^{n+1} = p^n + (\delta p)_{k_{conv}}$$

avec,

$$(\delta p)_{k_{conv}} = \sum_{k=1}^{k=k_{conv}} (\delta(\delta p))^k$$

En pratique, `RTP` contient $(\delta p)_{conv}^k$. On incrémente donc $p^n$ par `RTP` :

$$\texttt{RTP} = \texttt{RTPA} + \texttt{RTP}$$

## 16.4 Points à traiter

Il reste plusieurs problèmes à résoudre :

1. L'approximation du gradient de l'incrément de la pression par le gradient normal peut poser des problèmes de consistance, notamment avec la remarque ci-dessous.
2. L'approximation $\widetilde{T}^n \approx (\underline{\underline{T}}^n\ \underline{n})\,.\,\underline{n}$ n'est pas faite lors de la reconstruction des gradients au second membre de l'équation de pression. Elle n'est pas faite non plus pour le calcul du gradient cellule lors de l'application du filtre Rhie & Chow.
3. Utilisation de pondération $\dfrac{1}{2}$ lors de calculs de valeurs aux faces pour des raisons de stabilité numérique (ex. dans `itrmas` ou `itrgrp` lors de la reconstruction du gradient de l'incrément de pression à la face).
4. On pourra vérifier le calcul du résidu de normalisation (voir `preduv`).
5. Lors du calcul du flux de masse de $\widetilde{u} + \dfrac{\alpha}{\rho}\,\underline{\underline{T}}\ \underline{\text{grad}}\ p^n$ dans `inimas`, on utilise les conditions aux limites de la vitesse au temps $n$. Ceci reste un point peu clair particulièrement pour les conditions aux limites de sortie. Il faudra plus généralement analyser les conditions aux limites des variables dans `navsto`. Ce problème est à relier à celui signalé pour `vissec`.
6. Lors du test de convergence de la boucle sur les non orthogonalités, on multiplie la tolérance par 10. Est-ce réellement nécessaire ?
7. Il faudrait revoir le problème de relaxation lors de l'actualisation de la pression dans la boucle sur les non orthogonalités (un coefficient de relaxation dynamique serait peut-être intéressant).
8. Le filtre de Rhie & Chow peut s'avérer assez gênant dans certaines configurations. Il serait intéressant, avant toute travail de modification, de vérifier qu'il est vraiment utile, en identifiant clairement une configuration où il joue un rôle positif certain, si une telle configuration existe.

# 17- Sous-programme turbke

## 17.1 Fonction

Le but de ce sous-programme est de résoudre le système des équations de $k$ et $\varepsilon$ de manière semi-couplée.
Le système d'équations résolu est le suivant :

$$\left\{\begin{array}{l} \rho\dfrac{\partial k}{\partial t} + \mathrm{div}\left[\rho\underline{u}\,k - (\mu + \dfrac{\mu_t}{\sigma_k})\underline{\mathrm{grad}}\ k\right] = \mathcal{P} + \mathcal{G} - \rho\varepsilon + k\mathrm{div}(\rho\underline{u}) + \Gamma(k_i - k) \\ \qquad\qquad +\alpha_k k + \beta_k \\ \rho\dfrac{\partial \varepsilon}{\partial t} + \mathrm{div}\left[\rho\underline{u}\,\varepsilon - (\mu + \dfrac{\mu_t}{\sigma_\varepsilon})\underline{\mathrm{grad}}\ \varepsilon\right] = C_{\varepsilon_1}\dfrac{\varepsilon}{k}\left[\mathcal{P} + (1 - C_{\varepsilon_3})\mathcal{G}\right] - \rho C_{\varepsilon_2}\dfrac{\varepsilon^2}{k} + \varepsilon\mathrm{div}(\rho\underline{u}) \\ \qquad\qquad +\Gamma(\varepsilon_i - \varepsilon) + \alpha_\varepsilon\varepsilon + \beta_\varepsilon \end{array}\right. \quad \text{(II.17.1)}$$

$\mathcal{P}$ est le terme de production par cisaillement moyen :

$$\begin{aligned} \mathcal{P} &= -\rho R_{ij}\frac{\partial u_i}{\partial x_j} = -\left[-\mu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \frac{2}{3}\mu_t\frac{\partial u_k}{\partial x_k}\delta_{ij} + \frac{2}{3}\rho k\delta_{ij}\right]\frac{\partial u_i}{\partial x_j} \\ &= \mu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\frac{\partial u_i}{\partial x_j} - \frac{2}{3}\mu_t(\mathrm{div}\underline{u})^2 - \frac{2}{3}\rho k\mathrm{div}(\underline{u}) \\ &= \mu_t\left[2\left(\frac{\partial u}{\partial x}\right)^2 + 2\left(\frac{\partial v}{\partial y}\right)^2 + 2\left(\frac{\partial w}{\partial z}\right)^2 + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)^2\right] \\ &\qquad -\frac{2}{3}\mu_t(\mathrm{div}\underline{u})^2 - \frac{2}{3}\rho k\mathrm{div}(\underline{u}) \end{aligned}$$

$\mathcal{G}$ est le terme de production par gravité : $\mathcal{G} = -\dfrac{1}{\rho}\dfrac{\mu_t}{\sigma_t}\dfrac{\partial\rho}{\partial x_i}g_i$

La viscosité turbulente est $\mu_t = \rho C_\mu\dfrac{k^2}{\varepsilon}$.

Les constantes sont :
$C_\mu = 0,09$ ; $C_{\varepsilon_2} = 1,92$ ; $\sigma_k = 1$ ; $\sigma_\varepsilon = 1,3$
$C_{\varepsilon_3} = 0$ si $\mathcal{G} \geqslant 0$ (stratification instable) et $C_{\varepsilon_3} = 1$ si $\mathcal{G} \leqslant 0$ (stratification stable).

$\Gamma$ est un éventuel terme source de masse (tel que l'équation de conservation de masse devienne $\dfrac{\partial\rho}{\partial t} + \mathrm{div}(\rho\underline{u}) = \Gamma$). $\varphi_i$ ($\varphi = k$ ou $\varepsilon$) est la valeur de $\varphi$ associée à la masse injectée ou retirée. Dans le cas où on retire de la masse ($\Gamma < 0$), on a forcément $\varphi_i = \varphi$. De même, quand on injecte de la masse, on spécifie souvent aussi $\varphi_i = \varphi$. Dans ces deux cas, le terme disparaît de l'équation. Dans la suite du document, on qualifiera d'*injection forcée* les cas où on a $\Gamma > 0$ et $\varphi_i \neq \varphi$.

$\alpha_k$, $\beta_k$, $\alpha_\varepsilon$, $\beta_\varepsilon$ sont des termes sources utilisateur éventuels, conduisant à une implicitation partielle, imposés le cas échéant par le sous-programme `ustske`.

## 17.2 Discrétisation

La résolution se fait en trois étapes, afin de coupler partiellement les deux variables $k$ et $\varepsilon$. Pour simplifier, réécrivons le système de la façon suivante :

$$\left\{\begin{array}{l} \rho\dfrac{\partial k}{\partial t} = D(k) + S_k(k,\varepsilon) + k\mathrm{div}(\rho\underline{u}) + \Gamma(k_i - k) + \alpha_k k + \beta_k \\ \rho\dfrac{\partial \varepsilon}{\partial t} = D(\varepsilon) + S_\varepsilon(k,\varepsilon) + \varepsilon\mathrm{div}(\rho\underline{u}) + \Gamma(\varepsilon_i - \varepsilon) + \alpha_\varepsilon \varepsilon + \beta_\varepsilon \end{array}\right. \tag{II.17.2}$$

$D$ est l'opérateur de convection/diffusion. $S_k$ (resp. $S_\varepsilon$) est le terme source de $k$ (resp. $\varepsilon$).

Première phase : bilan explicite

On résout le bilan explicite :

$$\left\{\begin{array}{l} \rho^{(n)}\dfrac{k_e - k^{(n)}}{\Delta t} = D(k^{(n)}) + S_k(k^{(n)},\varepsilon^{(n)}) + k^{(n)}\mathrm{div}(\rho\underline{u}) + \Gamma(k_i - k^{(n)}) + \alpha_k k^{(n)} + \beta_k \\ \rho^{(n)}\dfrac{\varepsilon_e - \varepsilon^{(n)}}{\Delta t} = D(\varepsilon^{(n)}) + S_\varepsilon(k^{(n)},\varepsilon^{(n)}) + \varepsilon^{(n)}\mathrm{div}(\rho\underline{u}) + \Gamma(\varepsilon_i - \varepsilon^{(n)}) + \alpha_\varepsilon \varepsilon^{(n)} + \beta_\varepsilon \end{array}\right. \tag{II.17.3}$$

(le terme en $\Gamma$ n'est pris en compte que dans le cas de l'injection forcée)

Deuxième phase : couplage des termes sources

On implicite les termes sources de manière couplée :

$$\left\{\begin{array}{l} \rho^{(n)}\dfrac{k_{ts} - k^{(n)}}{\Delta t} = D(k^{(n)}) + S_k(k_{ts},\varepsilon_{ts}) + k^{(n)}\mathrm{div}(\rho\underline{u}) + \Gamma(k_i - k^{(n)}) + \alpha_k k^{(n)} + \beta_k \\ \rho^{(n)}\dfrac{\varepsilon_{ts} - \varepsilon^{(n)}}{\Delta t} = D(\varepsilon^{(n)}) + S_\varepsilon(k_{ts},\varepsilon_{ts}) + \varepsilon^{(n)}\mathrm{div}(\rho\underline{u}) + \Gamma(\varepsilon_i - \varepsilon^{(n)}) + \alpha_\varepsilon \varepsilon^{(n)} + \beta_\varepsilon \end{array}\right. \tag{II.17.4}$$

soit

$$\left\{\begin{array}{l} \rho^{(n)}\dfrac{k_{ts} - k^{(n)}}{\Delta t} = \rho^{(n)}\dfrac{k_e - k^{(n)}}{\Delta t} + S_k(k_{ts},\varepsilon_{ts}) - S_k(k^{(n)},\varepsilon^{(n)}) \\ \rho^{(n)}\dfrac{\varepsilon_{ts} - \varepsilon^{(n)}}{\Delta t} = \rho^{(n)}\dfrac{\varepsilon_e - \varepsilon^{(n)}}{\Delta t} + S_\varepsilon(k_{ts},\varepsilon_{ts}) - S_\varepsilon(k^{(n)},\varepsilon^{(n)}) \end{array}\right. \tag{II.17.5}$$

Le terme en $\mathrm{div}(\rho\underline{u})$ n'est pas implicité car il est lié au terme $D$ pour assurer que la matrice d'implicitation sera à diagonale dominante. Le terme en $\Gamma$ et les termes sources utilisateur ne sont pas implicités non plus, mais ils le seront dans la troisième phase.

Et on écrit (pour $\varphi = k$ ou $\varepsilon$)

$$S_\varphi(k_{ts},\varepsilon_{ts}) - S_\varphi(k^{(n)},\varepsilon^{(n)}) = (k_{ts} - k^{(n)})\left.\frac{\partial S_\varphi}{\partial k}\right|_{k^{(n)},\varepsilon^{(n)}} + (\varepsilon_{ts} - \varepsilon^{(n)})\left.\frac{\partial S_\varphi}{\partial \varepsilon}\right|_{k^{(n)},\varepsilon^{(n)}} \tag{II.17.6}$$

On résout donc finalement le système $2 \times 2$ :

$$\begin{pmatrix} \dfrac{\rho^{(n)}}{\Delta t} - \left.\dfrac{\partial S_k}{\partial k}\right|_{k^{(n)},\varepsilon^{(n)}} & -\left.\dfrac{\partial S_k}{\partial \varepsilon}\right|_{k^{(n)},\varepsilon^{(n)}} \\ -\left.\dfrac{\partial S_\varepsilon}{\partial k}\right|_{k^{(n)},\varepsilon^{(n)}} & \dfrac{\rho^{(n)}}{\Delta t} - \left.\dfrac{\partial S_\varepsilon}{\partial \varepsilon}\right|_{k^{(n)},\varepsilon^{(n)}} \end{pmatrix} \begin{pmatrix} (k_{ts} - k^{(n)}) \\ (\varepsilon_{ts} - \varepsilon^{(n)}) \end{pmatrix} = \begin{pmatrix} \rho^{(n)}\dfrac{k_e - k^{(n)}}{\Delta t} \\ \rho^{(n)}\dfrac{\varepsilon_e - \varepsilon^{(n)}}{\Delta t} \end{pmatrix} \tag{II.17.7}$$

Troisième phase : implicitation de la convection/diffusion

On résout le système :

$$\left\{\begin{array}{l} \rho^{(n)}\dfrac{k^{(n+1)}-k^{(n)}}{\Delta t} = D(k^{(n+1)}) + S_k(k_{ts},\varepsilon_{ts}) + k^{(n+1)}\mathrm{div}(\rho\underline{u}) + \Gamma(k_i - k^{(n+1)}) \\ \qquad\qquad +\alpha_k k^{(n+1)} + \beta_k \\ \rho^{(n)}\dfrac{\varepsilon^{(n+1)}-\varepsilon^{(n)}}{\Delta t} = D(\varepsilon^{(n+1)}) + S_\varepsilon(k_{ts},\varepsilon_{ts}) + \varepsilon^{(n+1)}\mathrm{div}(\rho\underline{u}) + \Gamma(\varepsilon_i - \varepsilon^{(n+1)}) \\ \qquad\qquad +\alpha_\varepsilon \varepsilon^{(n+1)} + \beta_\varepsilon \end{array}\right. \tag{II.17.8}$$

soit

$$\left\{\begin{array}{l} \rho^{(n)}\dfrac{k^{(n+1)}-k^{(n)}}{\Delta t} = D(k^{(n+1)}) - D(k^{(n)}) + \rho^{(n)}\dfrac{k_{ts}-k^{(n)}}{\Delta t} + (k^{(n+1)}-k^{(n)})\mathrm{div}(\rho\underline{u}) \\ \qquad\qquad -\Gamma(k^{(n+1)}-k^{(n)}) + \alpha_k(k^{(n+1)}-k^{(n)}) \\ \rho^{(n)}\dfrac{\varepsilon^{(n+1)}-\varepsilon^{(n)}}{\Delta t} = D(\varepsilon^{(n+1)}) - D(\varepsilon^{(n)}) + \rho^{(n)}\dfrac{\varepsilon_{ts}-\varepsilon^{(n)}}{\Delta t} + (\varepsilon^{(n+1)}-\varepsilon^{(n)})\mathrm{div}(\rho\underline{u}) \\ \qquad\qquad -\Gamma(\varepsilon^{(n+1)}-\varepsilon^{(n)}) + \alpha_\varepsilon(\varepsilon^{(n+1)}-\varepsilon^{(n)}) \end{array}\right. \tag{II.17.9}$$

Le terme en $\Gamma$ n'est là encore pris en compte que dans le cas de l'injection forcée. Le terme en $\alpha$ n'est pris en compte que si $\alpha$ est négatif, pour éviter d'affaiblir la diagonale de la matrice qu'on va inverser.

Précisions sur le couplage

Lors de la phase de couplage, afin de privilégier la stabilité et la réalisabilité du résultat, tous les termes ne sont pas pris en compte. Plus précisément, on peut écrire :

$$\left\{\begin{array}{l} S_k = \rho C_\mu \dfrac{k^2}{\varepsilon}\left(\tilde{\mathcal{P}} + \tilde{\mathcal{G}}\right) - \dfrac{2}{3}\rho k\mathrm{div}(\underline{u}) - \rho\varepsilon \\ S_\varepsilon = \rho C_{\varepsilon_1} C_\mu k\left(\tilde{\mathcal{P}} + (1-C_{\varepsilon_3})\tilde{\mathcal{G}}\right) - \dfrac{2}{3}C_{\varepsilon_1}\rho\varepsilon\mathrm{div}(\underline{u}) - \rho C_{\varepsilon_2}\dfrac{\varepsilon^2}{k} \end{array}\right. \tag{II.17.10}$$

en notant $\tilde{\mathcal{P}} = \left(\dfrac{\partial u_i}{\partial x_j} + \dfrac{\partial u_j}{\partial x_i}\right)\dfrac{\partial u_i}{\partial x_j} - \dfrac{2}{3}(\mathrm{div}\underline{u})^2$

et $\tilde{\mathcal{G}} = -\dfrac{1}{\rho\sigma_t}\dfrac{\partial\rho}{\partial x_i}g_i$

On a donc en théorie

$$\left\{\begin{array}{l} \dfrac{\partial S_k}{\partial k} = 2\rho C_\mu \dfrac{k}{\varepsilon}\left(\tilde{\mathcal{P}} + \tilde{\mathcal{G}}\right) - \dfrac{2}{3}\rho\mathrm{div}(\underline{u}) \\ \dfrac{\partial S_k}{\partial \varepsilon} = -\rho \\ \dfrac{\partial S_\varepsilon}{\partial k} = \rho C_{\varepsilon_1} C_\mu\left(\tilde{\mathcal{P}} + (1-C_{\varepsilon_3})\tilde{\mathcal{G}}\right) + \rho C_{\varepsilon_2}\dfrac{\varepsilon^2}{k^2} \\ \dfrac{\partial S_\varepsilon}{\partial \varepsilon} = -\dfrac{2}{3}C_{\varepsilon_1}\rho\mathrm{div}(\underline{u}) - 2\rho C_{\varepsilon_2}\dfrac{\varepsilon}{k} \end{array}\right. \tag{II.17.11}$$

En pratique, on va chercher à assurer $k_{ts} > 0$ et $\varepsilon_{ts} > 0$. En se basant sur un calcul simplifié, ainsi que sur le retour d'expérience d'ESTET, on montre qu'il est préférable de ne pas prendre en compte certains termes. Au final, on réalise le couplage suivant :

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}\begin{pmatrix} (k_{ts}-k^{(n)}) \\ (\varepsilon_{ts}-\varepsilon^{(n)}) \end{pmatrix} = \begin{pmatrix} \dfrac{k_e-k^{(n)}}{\Delta t} \\ \dfrac{\varepsilon_e-\varepsilon^{(n)}}{\Delta t} \end{pmatrix} \tag{II.17.12}$$

avec

$$\left\{\begin{array}{l} A_{11} = \dfrac{1}{\Delta t} - 2C_\mu \dfrac{k^{(n)}}{\varepsilon^{(n)}} \mathrm{Min}\left[\left(\tilde{\mathcal{P}} + \tilde{\mathcal{G}}\right), 0\right] + \dfrac{2}{3}\mathrm{Max}\left[\mathrm{div}(\underline{u}), 0\right] \\ A_{12} = 1 \\ A_{21} = -C_{\varepsilon_1} C_\mu \left(\tilde{\mathcal{P}} + (1 - C_{\varepsilon_3})\tilde{\mathcal{G}}\right) - C_{\varepsilon_2}\left(\dfrac{\varepsilon^{(n)}}{k^{(n)}}\right)^2 \\ A_{22} = \dfrac{1}{\Delta t} + \dfrac{2}{3} C_{\varepsilon_1} \mathrm{Max}\left[\mathrm{div}(\underline{u}), 0\right] + 2C_{\varepsilon_2} \dfrac{\varepsilon^{(n)}}{k^{(n)}} \end{array}\right. \tag{II.17.13}$$

(par définition de $C_{\varepsilon_3}$, $\tilde{\mathcal{P}} + (1 - C_{\varepsilon_3})\tilde{\mathcal{G}}$ est toujours positif)

## 17.3 Mise en œuvre

**• Calcul du terme de production**
On appelle trois fois `grdcel` pour calculer les gradients de $u$, $v$ et $w$. Au final, on a
$\mathtt{TINSTK} = 2\left(\dfrac{\partial u}{\partial x}\right)^2 + 2\left(\dfrac{\partial v}{\partial y}\right)^2 + 2\left(\dfrac{\partial w}{\partial z}\right)^2 + \left(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x}\right)^2 + \left(\dfrac{\partial u}{\partial z} + \dfrac{\partial w}{\partial x}\right)^2 + \left(\dfrac{\partial v}{\partial z} + \dfrac{\partial w}{\partial y}\right)^2$
et
$\mathtt{DIVU} = \dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y} + \dfrac{\partial w}{\partial z}$

(le terme $div(\underline{u})$ n'est pas calculé par `divmas`, pour correspondre exactement à la trace du tenseur des déformations qui est calculé pour la production)

**• Lecture des termes sources utilisateur**
Appel de `ustske` pour charger les termes sources utilisateurs. Ils sont stockés dans les tableaux suivants :
$\mathtt{W7} = \Omega\beta_k$
$\mathtt{W8} = \Omega\beta_\varepsilon$
$\mathtt{DAM} = \Omega\alpha_k$
$\mathtt{W9} = \Omega\alpha_\varepsilon$

Puis on ajoute le terme en $(div\underline{u})^2$ à `TINSTK`. On a donc
$\mathtt{TINSTK} = \tilde{\mathcal{P}}$

**• Calcul du terme de gravité**
Calcul uniquement si $\mathtt{IGRAKE} = 1$.
On appelle `grdcel` pour `ROM`, avec comme conditions aux limites $\mathtt{COEFA} = \mathtt{ROMB}$ et $\mathtt{COEFB} = \mathtt{VISCB} = 0$.
$\mathtt{PRDTUR} = \sigma_t$ est mis à 1 si on n'a pas de scalaire température.

$\tilde{\mathcal{G}}$ est calculé et les termes sources sont mis à jour :
$\mathtt{TINSTK} = \tilde{\mathcal{P}} + \tilde{\mathcal{G}}$
$\mathtt{TINSTE} = \tilde{\mathcal{P}} + \mathrm{Max}\left[\tilde{\mathcal{G}}, 0\right] = \tilde{\mathcal{P}} + (1 - C_{\varepsilon_3})\tilde{\mathcal{G}}$

Si $\mathtt{IGRAKE} = 0$, on a simplement
$\mathtt{TINSTK} = \mathtt{TINSTE} = \tilde{\mathcal{P}}$

**• Calcul du terme d'accumulation de masse**
On stocke $\mathtt{W1} = \Omega\mathrm{div}(\rho\underline{u})$ calculé par `divmas` (pour correspondre aux termes de convection de la matrice).

**• Calcul des termes sources explicites**
On affecte les termes sources explicites de $k$ et $\varepsilon$ pour la première étape.
$\mathtt{SMBRK} = \Omega\left(\mu_t(\tilde{\mathcal{P}} + \tilde{\mathcal{G}}) - \dfrac{2}{3}\rho^{(n)}k^{(n)}\mathrm{div}\underline{u} - \rho^{(n)}\varepsilon^{(n)}\right) + \Omega k^{(n)}\mathrm{div}(\rho\underline{u})$

$$\texttt{SMBRE} = \Omega\frac{\varepsilon^{(n)}}{k^{(n)}}\left(C_{\varepsilon_1}\left(\mu_t(\tilde{\mathcal{P}}+(1-C_{\varepsilon_3})\tilde{\mathcal{G}})-\frac{2}{3}\rho^{(n)}k^{(n)}\mathrm{div}\underline{u}\right)-C_{\varepsilon_2}\rho^{(n)}\varepsilon^{(n)}\right)+\Omega\varepsilon^{(n)}\mathrm{div}(\rho\underline{u})$$

soit $\texttt{SMBRK} = \Omega S_k^{(n)} + \Omega k^{(n)}\mathrm{div}(\rho\underline{u})$ et $\texttt{SMBRE} = \Omega S_\varepsilon^{(n)} + \Omega\varepsilon^{(n)}\mathrm{div}(\rho\underline{u})$.

**• Calcul des termes sources utilisateur**
On ajoute les termes sources utilisateur explicites à `SMBRK` et `SMBRE`, soit :
$\texttt{SMBRK} = \Omega S_k^{(n)} + \Omega k^{(n)}\mathrm{div}(\rho\underline{u}) + \Omega\alpha_k k^{(n)} + \Omega\beta_k$
$\texttt{SMBRE} = \Omega S_\varepsilon^{(n)} + \Omega\varepsilon^{(n)}\mathrm{div}(\rho\underline{u}) + \Omega\alpha_\varepsilon\varepsilon^{(n)} + \Omega\beta_\varepsilon$

Les tableaux `W7` et `W8` sont libérés, `DAM` et `W9` sont conservés pour être utilisés dans la troisième phase de résolution.

**• Calcul des termes de convection/diffusion explicites**
`bilsc2` est appelé deux fois, pour $k$ et pour $\varepsilon$, afin d'ajouter à `SMBRK` et `SMBRE` les termes de convection/diffusion explicites $D(k^{(n)})$ et $D(\varepsilon^{(n)})$. Ces termes sont d'abord stockés dans `W7` et `W8`, pour être conservés et réutilisés dans la troisième phase de résolution.

**• Termes source de masse**
Dans le cas d'une injection forcée de matière, on passe deux fois dans `catsma` pour ajouter les termes en $\Omega\Gamma(k_i - k^{(n)})$ et $\Omega\Gamma(\varepsilon_i - \varepsilon^{(n)})$ à `SMBRK` et `SMBRE`. La partie implicite ($\Omega\Gamma$) est stockée dans les variables `W2` et `W3`, qui seront utilisées lors de la troisième phase (les deux variables sont bien nécessaires, au cas où on aurait une injection forcée sur $k$ et pas sur $\varepsilon$, par exemple).

**• Fin de la première phase**
Ceci termine la première phase. On a
$$\texttt{SMBRK} = \Omega\rho^{(n)}\frac{k_e - k^{(n)}}{\Delta t}$$
$$\texttt{SMBRE} = \Omega\rho^{(n)}\frac{\varepsilon_e - \varepsilon^{(n)}}{\Delta t}$$

**• Phase de couplage**
(uniquement si $\texttt{IKECOU} = 1$)

On renormalise `SMBRK` et `SMBRE` qui deviennent les seconds membres du système de couplage.
$$\texttt{SMBRK} = \frac{1}{\Omega\rho^{(n)}}\texttt{SMBRK} = \frac{k_e - k^{(n)}}{\Delta t}$$
$$\texttt{SMBRE} = \frac{1}{\Omega\rho^{(n)}}\texttt{SMBRE} = \frac{\varepsilon_e - \varepsilon^{(n)}}{\Delta t}$$
et $\texttt{DIVP23} = \frac{2}{3}\mathrm{Max}\left[\mathrm{div}(\underline{u}), 0\right]$.

On remplit la matrice de couplage
$$\texttt{A11} = \frac{1}{\Delta t} - 2C_\mu\frac{k^{(n)}}{\varepsilon^{(n)}}\mathrm{Min}\left[\left(\tilde{\mathcal{P}}+\tilde{\mathcal{G}}\right), 0\right] + \frac{2}{3}\mathrm{Max}\left[\mathrm{div}(\underline{u}), 0\right]$$
$$\texttt{A12} = 1$$
$$\texttt{A21} = -C_{\varepsilon_1}C_\mu\left(\tilde{\mathcal{P}}+(1-C_{\varepsilon_3})\tilde{\mathcal{G}}\right) - C_{\varepsilon_2}\left(\frac{\varepsilon^{(n)}}{k^{(n)}}\right)^2$$
$$\texttt{A22} = \frac{1}{\Delta t} + \frac{2}{3}C_{\varepsilon_1}\mathrm{Max}\left[\mathrm{div}(\underline{u}), 0\right] + 2C_{\varepsilon_2}\frac{\varepsilon^{(n)}}{k^{(n)}}$$
On inverse le système $2\times 2$, et on récupère :
$\texttt{DELTK} = k_{ts} - k^{(n)}$
$\texttt{DELTE} = \varepsilon_{ts} - \varepsilon^{(n)}$

**• Fin de la deuxième phase**

On met à jour les variables SMBRK et SMBRE.

$\texttt{SMBRK} = \Omega\rho^{(n)}\dfrac{k_{ts}-k^{(n)}}{\Delta t}$

$\texttt{SMBRE} = \Omega\rho^{(n)}\dfrac{\varepsilon_{ts}-\varepsilon^{(n)}}{\Delta t}$

Dans le cas où on ne couple pas ($\texttt{IKECOU} = 0$), ces deux variables gardent la même valeur qu'à la fin de la première étape.

**• Calcul des termes implicites**

On retire à SMBRK et SMBRE la partie en convection diffusion au temps $n$, qui était stockée dans W7 et W8.

$\texttt{SMBRK} = \Omega\rho^{(n)}\dfrac{k_{ts}-k^{(n)}}{\Delta t} - \Omega D(k^{(n)})$

$\texttt{SMBRE} = \Omega\rho^{(n)}\dfrac{\varepsilon_{ts}-\varepsilon^{(n)}}{\Delta t} - \Omega D(\varepsilon^{(n)})$

On calcule les termes implicites, hors convection/diffusion, qui correspondent à la diagonale de la matrice.

$\texttt{TINSTK} = \dfrac{\Omega\rho^{(n)}}{\Delta t} - \Omega\mathrm{div}(\rho\underline{u}) + \Omega\Gamma + \Omega\mathrm{Max}[-\alpha_k, 0]$

$\texttt{TINSTE} = \dfrac{\Omega\rho^{(n)}}{\Delta t} - \Omega\mathrm{div}(\rho\underline{u}) + \Omega\Gamma + \Omega\mathrm{Max}[-\alpha_\varepsilon, 0]$

($\Gamma$ n'est pris en compte qu'en injection forcée, c'est-à-dire qu'il est forcément positif et ne risque pas d'affaiblir la diagonale de la matrice).

**• Résolution finale**

On passe alors deux fois dans codits, pour $k$ et $\varepsilon$, pour résoudre les équations du type :

$\texttt{TINST} \times (\varphi^{(n+1)} - \varphi^{(n)}) = D(\varphi^{(n+1)}) + \texttt{SMBR}$.

**• clipping final**

On passe enfin dans la routine clipke pour faire un clipping éventuel de $k^{(n+1)}$ et $\varepsilon^{(n+1)}$.

# 18- Sous-programme `turrij`

## 18.1 Fonction

Le but de ce sous-programme est de résoudre le système des équations des tensions de Reynolds et de la dissipation $\varepsilon$ de manière totalement découplée dans le cadre de l'utilisation du modèle $R_{ij}-\varepsilon$ LRR[1] (option `ITURB` = 30 dans `usini1`).
Le tenseur symétrique des tensions de Reynolds est noté $\underline{\underline{R}}$. Les composantes de ce tenseur représentent le moment d'ordre deux de la vitesse : $R_{ij}=\overline{u_i u_j}$.

Pour chaque composante $R_{ij}$, on résout :

$$\rho\frac{\partial R_{ij}}{\partial t}+\mathrm{div}(\rho\underline{u}\,R_{ij}-\mu\,\underline{\mathrm{grad}}\;R_{ij})= \begin{array}{l}\mathcal{P}_{ij}+\mathcal{G}_{ij}+\Phi_{ij}+d_{ij}-\varepsilon_{ij}+R_{ij}\,\mathrm{div}(\rho\underline{u})\\ +\Gamma(R_{ij}^{in}-R_{ij})+\alpha_{R_{ij}}R_{ij}+\beta_{R_{ij}}\end{array} \tag{II.18.1}$$

$\underline{\underline{\mathcal{P}}}$ est le tenseur de production par cisaillement moyen :

$$\mathcal{P}_{ij}=-\rho\left[R_{ik}\frac{\partial u_j}{\partial x_k}+R_{jk}\frac{\partial u_i}{\partial x_k}\right] \tag{II.18.2}$$

$\underline{\underline{\mathcal{G}}}$ est le tenseur de production par gravité :

$$\mathcal{G}_{ij}=\left[G_{ij}-C_3(G_{ij}-\frac{1}{3}\delta_{ij}G_{kk})\right] \tag{II.18.3}$$

avec

$$\left\{\begin{array}{c}G_{ij}=-\dfrac{3}{2}\dfrac{C_\mu}{\sigma_t}\dfrac{k}{\varepsilon}(r_i g_j+r_j g_i)\\ k=\dfrac{1}{2}R_{ll}\\ r_i=R_{ik}\dfrac{\partial\rho}{\partial x_k}\end{array}\right. \tag{II.18.4}$$

Dans ce qui précède, $k$ représente l'énergie turbulente[2], $g_i$ la composante de la gravité dans la direction $i$, $\sigma_t$ le nombre de Prandlt turbulent et $C_\mu$, $C_3$ des constantes définies dans Tab. 18.1.

$\underline{\underline{\Phi}}$ est le terme de corrélations pression-déformation. Il est modélisé avec le terme de dissipation $\underline{\underline{\varepsilon}}$ de la manière suivante :

$$\Phi_{ij}-[\varepsilon_{ij}-\frac{2}{3}\rho\;\delta_{ij}\varepsilon]=\phi_{ij,1}+\phi_{ij,2}+\phi_{ij,w} \tag{II.18.5}$$

Il en résulte :

$$\Phi_{ij}-\varepsilon_{ij}=\phi_{ij,1}+\phi_{ij,2}+\phi_{ij,w}-\frac{2}{3}\rho\;\delta_{ij}\varepsilon \tag{II.18.6}$$

[1] la description du SSG est prévue pour une version ultérieure de la documentation
[2] Les sommations se font sur l'indice $l$ et on applique plus généralement la convention de sommation d'Einstein.

Le terme $\phi_{ij,1}$ est un terme "lent" de retour à l'isotropie. Il est donné par :

$$\phi_{ij,1} = -\rho\, C_1 \frac{\varepsilon}{k}(R_{ij} - \frac{2}{3}k\delta_{ij}) \tag{II.18.7}$$

Le terme $\phi_{ij,2}$ est un terme "rapide" d'isotropisation de la production. Il est donné par :

$$\phi_{ij,2} = -\rho\, C_2(\mathcal{P}_{ij} - \frac{2}{3}\mathcal{P}\delta_{ij}) \tag{II.18.8}$$

avec,

$$\mathcal{P} = \frac{1}{2}\mathcal{P}_{kk}$$

Le terme $\phi_{ij,w}$ est appelé "terme d'echo de paroi". Il n'est pas utilisé par défaut dans *Code_Saturne*, mais peut être activé avec `IRIJEC` $= 1$. Si $y$ représente la distance à la paroi :

$$\begin{aligned}\phi_{ij,w} = \quad & \rho\, C_1' \frac{k}{\varepsilon}\left[R_{km}n_k n_m \delta_{ij} - \frac{3}{2}R_{ki}n_k n_j - \frac{3}{2}R_{kj}n_k n_i\right] f(\frac{l}{y}) \\ & +\rho\, C_2' \left[\phi_{km,2}n_k n_m \delta_{ij} - \frac{3}{2}\phi_{ki,2}n_k n_j - \frac{3}{2}\phi_{kj,2}n_k n_i\right] f(\frac{l}{y})\end{aligned} \tag{II.18.9}$$

$f$ est une fonction d'amortissement construite pour valoir 1 en paroi et tendre vers 0 en s'éloignant des parois.
La longueur $l$ représente $\dfrac{k^{\frac{3}{2}}}{\varepsilon}$, caractéristique de la turbulence. On prend :

$$f(\frac{l}{y}) = min(1,\ C_\mu^{0,75}\ \frac{k^{\frac{3}{2}}}{\varepsilon\ \kappa y}) \tag{II.18.10}$$

$d_{ij}$ est un terme de diffusion turbulente[3] qui vaut :

$$d_{ij} = C_S \frac{\partial}{\partial x_k}(\rho \frac{k}{\varepsilon} R_{km} \frac{\partial R_{ij}}{\partial x_m}) \tag{II.18.11}$$

On notera par la suite $\underline{\underline{A}} = C_S\, \rho\, \dfrac{k}{\varepsilon}\, \underline{\underline{R}}$. Ainsi, $d_{ij} = \mathrm{div}(\underline{\underline{A}}\, \underline{\mathrm{grad}}\ (R_{ij}))$ est une diffusion avec un coefficient tensoriel.

Le terme de dissipation turbulente $\underline{\varepsilon}$ est traité dans ce qui précède avec le terme $\underline{\underline{\Phi}}$.

$\Gamma$ est le terme source de masse[4], $R_{ij}^{in}$ est la valeur de $R_{ij}$ associée à la masse injectée ou retirée.

$(\alpha_{R_{ij}}\, R_{ij} + \beta_{R_{ij}})$ représente le terme source utilisateur (sous-programme `ustsri`) éventuel avec une décomposition permettant d'impliciter la partie $\alpha_{R_{ij}}\, R_{ij}$ si $\alpha_{R_{ij}} \geqslant 0$.

De même, on résout une équation de convection/diffusion/termes sources pour la dissipation $\varepsilon$. Cette équation est très semblable à celle du modèle $k-\varepsilon$ (voir `turbke`), seuls les termes de viscosité turbulente et de gravité changent. On résout :

$$\begin{aligned}\rho\frac{\partial \varepsilon}{\partial t} + \mathrm{div}\left[\rho \underline{u}\, \varepsilon - (\mu \underline{\mathrm{grad}}\ \varepsilon)\right] = \quad & d_\varepsilon + C_{\varepsilon 1}\frac{\varepsilon}{k}\left[\mathcal{P} + \mathcal{G}_\varepsilon\right] - \rho C_{\varepsilon 2}\frac{\varepsilon^2}{k} + \varepsilon \mathrm{div}(\rho \underline{u}) \\ & +\Gamma(\varepsilon^{in} - \varepsilon) + \alpha_\varepsilon \varepsilon + \beta_\varepsilon\end{aligned} \tag{II.18.12}$$

[3] Dans la littérature, ce terme contient en général la dissipation par viscosité moléculaire.

[4] Dans ce cas, l'équation de continuité s'écrit : $\dfrac{\partial \rho}{\partial t} + \mathrm{div}(\rho \underline{u}) = \Gamma$.

| $C_\mu$ | $C_\varepsilon$ | $C_{\varepsilon_1}$ | $C_{\varepsilon_2}$ | $C_1$ | $C_2$ | $C_3$ | $C_S$ | $C'_1$ | $C'_2$ |
|---|---|---|---|---|---|---|---|---|---|
| 0,09 | 0,18 | 1,44 | 1,92 | 1,8 | 0,6 | 0,55 | 0,22 | 0,5 | 0,3 |

Table 18.1: Définition des constantes utilisées.

$d_\varepsilon$ est le terme de diffusion turbulente :

$$d_\varepsilon = C_\varepsilon \frac{\partial}{\partial x_k}\left(\rho \frac{k}{\varepsilon} R_{km} \frac{\partial \varepsilon}{\partial x_m}\right) \tag{II.18.13}$$

On notera par la suite $\underline{\underline{A'}} = \rho\, C_\varepsilon \frac{k}{\varepsilon} \underline{\underline{R}}$. Le terme de diffusion turbulente est donc modélisé par :

$$d_\varepsilon = \mathrm{div}(\underline{\underline{A'}}\, \underline{\mathrm{grad}}\ (\varepsilon))$$

La viscosité turbulente usuelle ($\nu_t$) en modèle $k-\varepsilon$ est remplacée par le tenseur visqueux $\underline{\underline{A'}}$.

$\mathcal{P}$ est le terme de production par cisaillement moyen : $\mathcal{P} = \frac{1}{2}\mathcal{P}_{kk}$. Ce terme est modélisé avec la notion de viscosité turbulente dans le cadre du modèle $k-\varepsilon$. Dans le cas présent, il est calculé à l'aide des tensions de Reynolds (à partir de $\mathcal{P}_{ij}$).

$\mathcal{G}_\varepsilon$ est le terme de production des effets de gravité pour la variable $\varepsilon$.

$$\mathcal{G}_\varepsilon = max(0, \frac{1}{2} G_{kk}) \tag{II.18.14}$$

# 18.2 Discrétisation

La résolution se fait en découplant totalement les tensions de Reynolds entre elles et la dissipation $\varepsilon$. On résout ainsi une équation de convection/diffusion/termes sources pour chaque variable. Les variables sont résolues dans l'ordre suivant : $R_{11}$, $R_{22}$, $R_{33}$, $R_{12}$, $R_{13}$, $R_{23}$ et $\varepsilon$. L'ordre de la résolution n'est pas important puisque l'on a opté pour une résolution totalement découplée en n'implicitant que les termes pouvant être linéarisés par rapport à la variable courante[5].

Les équations sont résolues à l'instant $n+1$.

## 18.2.1 Variables tensions de Reynolds

Pour chaque composante $R_{ij}$, on écrit :

$$\begin{aligned} \rho^n \frac{R_{ij}^{n+1} - R_{ij}^n}{\Delta t^n} + \mathrm{div}\left[(\rho \underline{u})^n R_{ij}^{n+1} - \mu^n\ \underline{\mathrm{grad}}\ R_{ij}^{n+1}\right] = \quad & \mathcal{P}_{ij}^n + \mathcal{G}_{ij}^n \\ & + \phi_{ij,1}^{n,n+1} + \phi_{ij,2}^n + \phi_{ij,w}^n \\ & + d_{ij}^{n,n+1} - \frac{2}{3}\rho^n \varepsilon^n \delta_{ij} + R_{ij}^{n+1} \mathrm{div}(\rho \underline{u})^n \\ & + \Gamma(R_{ij}^{in} - R_{ij}^{n+1}) \\ & + \alpha_{R_{ij}}^n R_{ij}^{n+1} + \beta_{R_{ij}}^n \end{aligned} \tag{II.18.15}$$

$\mu^n$ est la viscosité moléculaire[6].
L'indice $(n, n+1)$ est relatif à une semi implicitation des termes (voir ci-dessous). Quand seul l'indice $(n)$ est utilisé, il suffit de reprendre l'expression des termes et de considérer que toutes les variables sont explicites.

[5] En effet, aucune variable n'est actualisée pour la résolution de la suivante.
[6] La viscosité peut dépendre éventuellement de la température ou d'autres variables.

Dans le terme $\phi_{ij,1}^{n,n+1}$ donné ci-dessous, la tension de Reynolds $R_{ij}$ est implicite (les tensions diagonales apparaissent aussi dans l'énergie turbulente $k$). Ainsi :

$$\phi_{ij,1}^{n,n+1} = -\rho^n\, C_1\, \frac{\varepsilon^n}{k^n} \left[ (1 - \frac{\delta_{ij}}{3}) R_{ij}^{n+1} - \delta_{ij} \frac{2}{3} (k^n - \frac{1}{2} R_{ii}^n) \right] \tag{II.18.16}$$

Le terme de diffusion turbulente $\underline{\underline{d}}$ s'écrit : $d_{ij} = \mathrm{div}\left[\underline{\underline{A}}\, \underline{\mathrm{grad}}\ R_{ij}\right]$. Le tenseur $\underline{\underline{A}}$ est toujours explicite. En intégrant sur un volume de contrôle (cellule) $\Omega_l$, le terme $\underline{\underline{d}}$ de diffusion turbulente de $R_{ij}$ s'écrit :

$$\int_{\Omega_l} d_{ij}^{n,n+1}\ d\Omega = \sum_{m \in Vois(l)} \left[\underline{\underline{A}}^n\, \underline{\mathrm{grad}}\ R_{ij}^{n+1}\right]_{lm} \cdot \underline{n}_{lm} S_{lm} \tag{II.18.17}$$

$\underline{n}_{lm}$ est la normale unitaire à la face[7] $\partial\Omega_{lm} = \Gamma_{lm}$ de la frontière $\partial\Omega_l = \underset{m}{\cup} \partial\Omega_{lm}$ de $\Omega_l$, face désignée par abus par $lm$ et $S_{lm}$ sa surface associée.

On décompose $\underline{\underline{A}}^n$ en partie diagonale $\underline{\underline{D}}^n$ et extra-diagonale $\underline{\underline{E}}^n$ :

$$\underline{\underline{A}}^n = \underline{\underline{D}}^n + \underline{\underline{E}}^n$$

Ainsi,

$$\begin{aligned} \int_{\Omega_l} d_{ij}\ d\Omega = & \sum_{m \in Vois(l)} \underbrace{\left[\underline{\underline{D}}^n\, \underline{\mathrm{grad}}\ R_{ij}\right]_{lm} \cdot \underline{n}_{lm} S_{lm}}_{\text{partie diagonale}} \\ + & \sum_{m \in Vois(l)} \underbrace{\left[\underline{\underline{E}}^n\, \underline{\mathrm{grad}}\ R_{ij}\right]_{lm} \cdot \underline{n}_{lm} S_{lm}}_{\text{partie extra-diagonale}} \end{aligned} \tag{II.18.18}$$

La partie extra-diagonale sera prise totalement explicite et interviendra donc dans l'expression regroupant les termes purement explicites $f_s^{exp}$ du second membre de `codits`.
Pour la partie diagonale, on introduit la composante normale du gradient de la variable principale $R_{ij}$. Cette contribution normale sera traitée en implicite pour la variable et interviendra à la fois dans l'expression de la matrice simplifiée du système résolu par `codits` et dans le second membre traité par `bilsc2`. La contribution tangentielle sera, elle, purement explicite et donc prise en compte dans $f_s^{exp}$ intervenant dans le second membre de `codits`.
On a :

$$\underline{\mathrm{grad}}\ R_{ij} = \underline{\mathrm{grad}}\ R_{ij} - (\underline{\mathrm{grad}}\ R_{ij} \cdot \underline{n}_{lm})\, \underline{n}_{lm} + (\underline{\mathrm{grad}}\ R_{ij} \cdot \underline{n}_{lm})\, \underline{n}_{lm} \tag{II.18.19}$$

Comme

$$\left[\underline{\underline{D}}^n\ [(\underline{\mathrm{grad}}\ R_{ij} \cdot \underline{n}_{lm})\, \underline{n}_{lm}]\right] \cdot \underline{n}_{lm} = \gamma_{lm}^n (\underline{\mathrm{grad}}\ R_{ij} \cdot \underline{n}_{lm})$$

avec :

$$\gamma_{lm}^n = (D_{11}^n)\, n_{1,lm}^2 + (D_{22}^n)\, n_{2,lm}^2 + (D_{33}^n)\, n_{3,lm}^2$$

on peut traiter ce terme $\gamma_{lm}^n$ comme une diffusion avec un coefficient de diffusion indépendant de la direction.

Finalement, on écrit :

$$\begin{aligned} & \int_{\Omega_l} d_{ij}^{n,n+1}\ d\Omega = \\ & + \sum_{m \in Vois(l)} \left[\ \underline{\underline{E}}^n\, \underline{\mathrm{grad}}\ R_{ij}^n\right]_{lm} \cdot \underline{n}_{lm} S_{lm} \\ & + \sum_{m \in Vois(l)} \left[\ \underline{\underline{D}}^n\, \underline{\mathrm{grad}}\ R_{ij}^n\right]_{lm} \cdot \underline{n}_{lm} S_{lm} \\ & - \sum_{m \in Vois(l)} \gamma_{lm}^n \left(\underline{\mathrm{grad}}\ R_{ij}^n \cdot \underline{n}_{lm}\right) S_{lm} + \sum_{m \in Vois(l)} \gamma_{lm}^n \left(\underline{\mathrm{grad}}\ R_{ij}^{n+1} \cdot \underline{n}_{lm}\right) S_{lm} \end{aligned} \tag{II.18.20}$$

[7] La notion de face purement interne ou de bord n'est pas explicitée ici, pour alléger l'exposé. Pour être rigoureux et homogène avec les notations adoptées, il faudrait distinguer $m \in Vois(l)$ et $m \in \gamma_b(l)$.

Les trois premiers termes sont totalement explicites et correspondent à la discrétisation de l'opérateur continu :

$$\mathrm{div}(\underline{\underline{E}}^n \, \underline{\mathrm{grad}} \; R_{ij}^n) + \mathrm{div}(\underline{\underline{D}}^n \, [\underline{\mathrm{grad}} \; R_{ij}^n - (\underline{\mathrm{grad}} \; R_{ij}^n \, . \, \underline{n}) \, \underline{n}] \,)$$

en omettant la notion de face.
Le dernier terme est implicite relativement à la variable $R_{ij}$ et correspond à l'opérateur continu :

$$\mathrm{div}(\underline{\underline{D}}^n \, (\underline{\mathrm{grad}} \; R_{ij}^{n+1} \, . \, \underline{n}) \, \underline{n})$$

### 18.2.2 Variable $\varepsilon$

On résout l'équation de $\varepsilon$ de façon analogue à celle de $R_{ij}$.

$$\begin{aligned} \rho^n \, \frac{\varepsilon^{n+1} - \varepsilon^n}{\Delta t^n} + \mathrm{div}((\rho \, \underline{u})^n \varepsilon^{n+1}) - \mathrm{div}(\mu^n \; \underline{\mathrm{grad}} \; \varepsilon^{n+1}) = \quad & d_\varepsilon^{n,n+1} \\ & + C_{\varepsilon_1} \frac{k^n}{\varepsilon^n} \, [\mathcal{P}^n + \mathcal{G}_\varepsilon^n] - \rho^n C_{\varepsilon_2} \frac{(\varepsilon^n)^2}{k^n} \\ & + \varepsilon^{n+1} \mathrm{div}(\rho \underline{u})^n \\ & + \Gamma(\varepsilon^{in} - \varepsilon^{n+1}) + \alpha_\varepsilon^n \varepsilon^{n+1} + \beta_\varepsilon^n \end{aligned} \tag{II.18.21}$$

Le terme de diffusion turbulente $d_\varepsilon^{n,n+1}$ est traité comme celui des variables $R_{ij}$ et s'écrit :

$$d_\varepsilon^{n,n+1} = \mathrm{div}\big[\underline{\underline{A'}}^n \; \underline{\mathrm{grad}} \; \varepsilon^{n+1}\big]$$

Le tenseur $\underline{\underline{A'}}$ est toujours explicite. On le décompose en une partie diagonale $\underline{\underline{D'}}^n$ et une partie extra-diagonale $\underline{\underline{E'}}^n$ :

$$\underline{\underline{A'}}^n = \underline{\underline{D'}}^n + \underline{\underline{E'}}^n$$

Ainsi :

$$\begin{aligned} \int_{\Omega_l} d_\varepsilon^{n,n+1} \; d\Omega = & \sum_{m \in Vois(l)} \big[\underline{\underline{E'}}^n \, \underline{\mathrm{grad}} \; \varepsilon^n\big]_{lm} \, . \, \underline{n}_{lm} S_{lm} \\ + \sum_{m \in Vois(l)} \big[\underline{\underline{D'}}^n \, \underline{\mathrm{grad}} \; \varepsilon^n\big]_{lm} \, . \, \underline{n}_{lm} S_{lm} & - \sum_{m \in Vois(l)} \eta_{lm}^n \, (\underline{\mathrm{grad}} \; \varepsilon^n \, . \, \underline{n}_{lm}) \, S_{lm} \\ & + \sum_{m \in Vois(l)} \eta_{lm}^n \, \big(\underline{\mathrm{grad}} \; \varepsilon^{n+1} \, . \, \underline{n}_{lm}\big) \, S_{lm} \end{aligned} \tag{II.18.22}$$

avec :

$$\eta_{lm}^n = (D'^n_{11}) \, n^2_{1,\,lm} + (D'^n_{22}) \, n^2_{2,\,lm} + (D'^n_{33}) \, n^2_{3,\,lm}.$$

On peut traiter ce terme $\eta_{lm}^n$ comme une diffusion avec un coefficient de diffusion indépendant de la direction.
Les trois premiers termes sont totalement explicites et correspondent à l'opérateur :

$$\mathrm{div}(\underline{\underline{E'}}^n \, \varepsilon^n) + \mathrm{div}(\underline{\underline{D'}}^n \, [\underline{\mathrm{grad}} \; \varepsilon^n - (\underline{\mathrm{grad}} \; \varepsilon^n . \, \underline{n}) \, \underline{n}] \,)$$

en omettant la notion de face.
Le dernier terme est implicite relativement à la variable $\varepsilon$ et correspond à l'opérateur :

$$\mathrm{div}(\underline{\underline{D'}}^n \, (\underline{\mathrm{grad}} \; \varepsilon^{n+1} . \, \underline{n}) \, \underline{n})$$

## 18.3 Mise en œuvre

La numéro de la phase traitée fait partie des arguments de `turrij`. On omettra volontairement de le préciser dans ce qui suit, on indiquera par ( ) la notion de tableau s'y rattachant.

- **Calcul des termes de production $\underline{\underline{\mathcal{P}}}$**

⋆ Initialisation à zéro du tableau PRODUC dimensionné à NCEL × 6.

⋆ On appelle trois fois grdcel pour calculer les gradients des composantes de la vitesse $u$, $v$ et $w$ prises au temps $n$.

Au final, on a :

$$\begin{aligned}
\texttt{PRODUC(1,IEL)} &= -2\left[R_{11}^n\frac{\partial u^n}{\partial x}+R_{12}^n\frac{\partial u^n}{\partial y}+R_{13}^n\frac{\partial u^n}{\partial z}\right] \text{ (production de } R_{11}^n)\\
\texttt{PRODUC(2,IEL)} &= -2\left[R_{12}^n\frac{\partial v^n}{\partial x}+R_{22}^n\frac{\partial v^n}{\partial y}+R_{23}^n\frac{\partial v^n}{\partial z}\right] \text{ (production de } R_{22}^n)\\
\texttt{PRODUC(3,IEL)} &= -2\left[R_{13}^n\frac{\partial w^n}{\partial x}+R_{23}^n\frac{\partial w^n}{\partial y}+R_{33}^n\frac{\partial w^n}{\partial z}\right] \text{ (production de } R_{33}^n)\\
\texttt{PRODUC(4,IEL)} &= -\left[R_{12}^n\frac{\partial u^n}{\partial x}+R_{22}^n\frac{\partial u^n}{\partial y}+R_{23}^n\frac{\partial u^n}{\partial z}\right]\\
&\quad -\left[R_{11}^n\frac{\partial v^n}{\partial x}+R_{12}^n\frac{\partial v^n}{\partial y}+R_{13}^n\frac{\partial v^n}{\partial z}\right] \text{ (production de } R_{12}^n)\\
\texttt{PRODUC(5,IEL)} &= -\left[R_{13}^n\frac{\partial u^n}{\partial x}+R_{23}^n\frac{\partial u^n}{\partial y}+R_{33}^n\frac{\partial u^n}{\partial z}\right]\\
&\quad -\left[R_{11}^n\frac{\partial w^n}{\partial x}+R_{12}^n\frac{\partial w^n}{\partial y}+R_{13}^n\frac{\partial w^n}{\partial z}\right] \text{ (production de } R_{13}^n)\\
\texttt{PRODUC(6,IEL)} &= -\left[R_{13}^n\frac{\partial v^n}{\partial x}+R_{23}^n\frac{\partial v^n}{\partial y}+R_{33}^n\frac{\partial v^n}{\partial z}\right]\\
&\quad -\left[R_{12}^n\frac{\partial w^n}{\partial x}+R_{22}^n\frac{\partial w^n}{\partial y}+R_{23}^n\frac{\partial w^n}{\partial z}\right] \text{ (production de } R_{23}^n)
\end{aligned}$$

**• Calcul du gradient de la masse volumique $\rho^n$ prise au début du pas de temps courant**[8] $(n+1)$

Ce calcul n'a lieu que si les termes de gravité doivent être pris en compte (IGRARI() = 1).

⋆ Appel de grdcel pour calculer le gradient de $\rho^n$ dans les trois directions de l'espace. Les conditions aux limites sur $\rho^n$ sont des conditions de Dirichlet puisque la valeur de $\rho^n$ aux faces de bord $ik$ (variable IFAC) est connue et vaut $\rho_{b\,ik}$. Pour écrire les conditions aux limites sous la forme habituelle,

$$\rho_{b\,ik} = \texttt{COEFA} + \texttt{COEFB}\,\rho_{I'}^n$$

on pose alors COEFA = PROPCE(IFAC,IPPROB(IROM(IPHAS))) et COEFB = VISCB = 0.
PROPCE(1,IPPROB(IROM(IPHAS))) (resp.VISCB) est utilisé en lieu et place de l'habituel COEFA (COEFB), lors de l'appel à grdcel.
On a donc :
GRAROX = $\dfrac{\partial\rho^n}{\partial x}$ , GRAROY = $\dfrac{\partial\rho^n}{\partial y}$ et GRAROZ = $\dfrac{\partial\rho^n}{\partial z}$ .

Le gradient de $\rho^n$ servira à calculer les termes de production par effets de gravité si ces derniers sont pris en compte.

**• Boucle ISOU de 1 à 6 sur les tensions de Reynolds**

Pour ISOU = $1, 2, 3, 4, 5, 6$, on résout respectivement et dans l'ordre les équations de $R_{11}$, $R_{22}$, $R_{33}$, $R_{12}$, $R_{13}$ et $R_{23}$ par l'appel au sous-programme resrij.
La résolution se fait par incrément $\delta R_{ij}^{n+1,k+1}$ , en utilisant la même méthode que celle décrite dans le sous-programme codits. On adopte ici les mêmes notations. SMBR est le second membre du système à inverser, système portant sur les incréments de la variable. ROVSDT représente la diagonale de la matrice, hors convection/diffusion.

[8] *i.e.* calculée à partir des variables du pas de temps précédent $n$ si nécessaire.

On va résoudre l'équation (II.18.15) sous forme incrémentale en utilisant `codits`, soit :

$$
\begin{aligned}
&\underbrace{\left(\frac{\rho_L^n}{\Delta t^n}+\rho_L^n\,C_1\,\frac{\varepsilon_L^n}{k_L^n}(1-\frac{\delta_{ij}}{3})-m_{lm}^n+\Gamma_L\,+max(-\alpha_{R_{ij}}^n,0)\right)|\Omega_l|\,(\delta R_{ij}^{n+1,p+1})_L}_{\texttt{ROVSDT}\text{ contribuant à la diagonale de la matrice simplifiée de }\texttt{matrix}}\\
&\underbrace{+\sum_{m\in Vois(l)}\left[m_{lm}^n\delta R_{ij,\,f_{lm}}^{n+1,p+1}-(\mu_{lm}^n+\gamma_{lm}^n)\,\frac{(\delta R_{ij}^{n+1,p+1})_M-(\delta R_{ij}^{n+1,p+1})_L)}{\overline{L'M'}}\,S_{lm}\right]}_{\text{convection upwind pur et diffusion non reconstruite relatives à la matrice simplifiée de }\texttt{matrix}^{9}}\\
&=-\frac{\rho_L^n}{\Delta t^n}\,\left((R_{ij}^{n+1,p})_L-(R_{ij}^n)_L\right)\\
&\underbrace{-\int_{\Omega_l}\left(\mathrm{div}\,[\,(\rho\,\underline{u})^n\,R_{ij}^{n+1,p}-(\mu^n\,+\gamma^n\,)\,\underline{\mathrm{grad}}\;R_{ij}^{n+1,p}\,]\right)\,d\Omega}_{\text{convection et diffusion traitées par }\texttt{bilsc2}}\\
&+\int_{\Omega_l}\left[\mathcal{P}_{ij}^{n+1,p}+\mathcal{G}_{ij}^{n+1,p}-\rho^n\,C_1\,\frac{\varepsilon^n}{k^n}\left[R_{ij}^{n+1,p}-\frac{2}{3}\,k^n\,\delta_{ij}\right]+\phi_{ij,2}^{n+1,p}+\phi_{ij,w}^{n+1,p}\right]\,d\Omega\\
&+\int_{\Omega_l}\left[-\frac{2}{3}\rho^n\varepsilon^n\delta_{ij}+\Gamma\,(\,R_{ij}^{in}-R_{ij}^{n+1,p}\,)+\alpha_{R_{ij}}^n\,R_{ij}^{n+1,p}+\beta_{R_{ij}}^n\right]\,d\Omega\\
&+\sum_{m\in Vois(l)}\left[\,\underline{\underline{E}}^n\,\underline{\mathrm{grad}}\;R_{ij}^{n+1,p}\right]_{lm}\,.\,\underline{n}_{lm}S_{lm}\\
&+\sum_{m\in Vois(l)}\left[\,\underline{\underline{D}}^n\,\underline{\mathrm{grad}}\;R_{ij}^{n+1,p}\right]_{lm}\,.\,\underline{n}_{lm}S_{lm}\\
&-\sum_{m\in Vois(l)}\gamma_{lm}^n\left(\underline{\mathrm{grad}}\;R_{ij}^{n+1,p}\,.\,\underline{n}_{lm}\right)S_{lm}\\
&+\sum_{m\in Vois(l)}m_{lm}^n\,(R_{ij}^{n+1,p})_L
\end{aligned}
\qquad \text{(II.18.23)}
$$

où on rappelle :
pour $n$ donné entier positif, on définit la suite $(R_{ij}^{n+1,p})_{p\in\mathbb{N}}$ par :

$$
\left\{\begin{array}{l} R_{ij}^{n+1,0}=R_{ij}^n \\ R_{ij}^{n+1,p+1}=R_{ij}^{n+1,p}+\delta R_{ij}^{n+1,p+1}\end{array}\right.
$$

$(\delta R_{ij}^{n+1,p+1})_L$ désigne la valeur sur l'élément $\Omega_l$ du $(p+1)$-ième incrément de $R_{ij}^{n+1}$, $m_{lm}^n$ le flux de masse à l'instant $n$ à travers la face $lm$, $\delta R_{ij,\,f_{lm}}^{n+1,p+1}$ vaut $(\delta R_{ij}^{n+1,p+1})_L$ si $m_{lm}^n \geqslant 0$, $(\delta R_{ij}^{n+1,p+1})_M$ sinon, $\mathcal{P}_{ij}^{n+1,p}$, $\phi_{ij,2}^{n+1,p}$, $\phi_{ij,w}^{n+1,p}$ les valeurs des quantités associées correspondant à l'incrément $(\delta R_{ij}^{n+1,p})$.

Tous ces termes sont calculés comme suit :

- Terme de gauche de l'équation (II.18.23)
  Dans `resrij` est calculée la variable `ROVSDT`. Les autres termes sont complétés par `codits`, lors de la construction de la matrice simplifiée , *via* un appel au sous-programme `matrix`. La quantité $(\mu_{lm}^n+\gamma_{lm}^n)$ à la face $lm$ est calculée lors de l'appel à `visort`.

- Second membre de l'équation (II.18.23)
  Le premier terme non détaillé est calculé par le sous-programme `bilsc2`, qui applique le schéma convectif choisi par l'utilisateur, qui reconstruit ou non selon le souhait de l'utilisateur les gradients intervenants dans la convection-diffusion.
  Les termes sans accolade sont, eux, complètement explicites et ajoutés au fur et à mesure dans `SMBR` pour former l'expression $f_s^{exp}$ de `codits`.

[9] Si `IRIJNU` = 1, on remplace $\mu_{lm}^n$ par $(\mu+\mu_t)_{lm}^n$ dans l'expression de la diffusion non reconstruite associée à la matrice simplifiée de `matrix` ($\mu_t$ désigne la viscosité turbulente calculée comme en $k-\varepsilon$).

On décrit ci-dessous les étapes de `resrij` :

- DELTIJ = 1, pour `ISOU` $\leqslant 3$ et DELTIJ = 0 Si `ISOU` $> 3$. Cette valeur représente le symbole de Kroeneker $\delta_{ij}$.
- Initialisation à zéro de `SMBR` (tableau contenant le second membre) et `ROVSDT` (tableau contenant la diagonale de la matrice sauf celle relative à la contribution de la diagonale des opérateurs de convection et de diffusion linéarisés [10]), tous deux de dimension `NCEL`.
- Lecture et prise en compte des termes sources utilisateur pour la variable $R_{ij}$

  Appel à `ustsri` pour charger les termes sources utilisateurs. Ils sont stockés comme suit. Pour la cellule $\Omega_l$ de centre $L$, représentée par `IEL`, on a :

$$\left\{ \begin{array}{ll} \texttt{ROVSDT(IEL)} & = |\Omega_l|\ \alpha_{R_{ij}} \\ \texttt{SMBR(IEL)} & = |\Omega_l|\ \beta_{R_{ij}} \end{array} \right.$$

  On affecte alors les valeurs adéquates au second membre `SMBR` et à la diagonale `ROVSDT` comme suit :

$$\left\{ \begin{array}{ll} \texttt{SMBR(IEL)} & = \texttt{SMBR(IEL)} + |\Omega_l|\ \alpha_{R_{ij}}\ (R_{ij}^n)_L \\ \texttt{ROVSDT(IEL)} & = \max\ (-\ |\Omega_l|\ \alpha_{R_{ij}}, 0) \end{array} \right.$$

  La valeur de `ROVSDT` est ainsi calculée pour des raisons de stabilité numérique. En effet, on ne rajoute sur la diagonale que les valeurs positives, ce qui correspond physiquement à impliciter les termes de rappel uniquement.
- Calcul du terme source de masse si $\Gamma_L > 0$

  Appel de `catsma` et incrémentation si nécessaire de `SMBR` et `ROVSDT` *via* :

$$\left\{ \begin{array}{l} \texttt{SMBR(IEL)} = \texttt{SMBR(IEL)} + |\Omega_l|\ \Gamma_L\ \left[(R_{ij}^{in})_L - (R_{ij}^n)_L\right] \\ \texttt{ROVSDT(IEL)} = \texttt{ROVSDT(IEL)} + |\Omega_l|\ \Gamma_L \end{array} \right.$$

- Calcul du terme d'accumulation de masse et du terme instationnaire

  On stocke `W1` $= \displaystyle\int_{\Omega_l} \mathrm{div}\,(\rho\,\underline{u})\,d\Omega$ calculé par `divmas` à l'aide des flux de masse aux faces internes $m_{lm}^n = \displaystyle\sum_{m \in Vois(l)} (\rho \underline{u})_{lm}^n . \underline{S}_{lm}$ (tableau `FLUMAS`) et des flux de masse aux bords $m_{b_{lk}}^n = \displaystyle\sum_{k \in \gamma_b(l)} (\rho \underline{u})_{b_{lk}}^n . \underline{S}_{b_{lk}}$ (tableau `FLUMAB`). On incrémente ensuite `SMBR` et `ROVSDT`.

$$\left\{ \begin{array}{ll} \texttt{SMBR(IEL)} & = \texttt{SMBR(IEL)} + \texttt{ICONV}\ (R_{ij}^n)_L\ (\displaystyle\int_{\Omega_l} \mathrm{div}\,(\rho\,\underline{u})\ d\Omega) \\ \texttt{ROVSDT(IEL)} & = \texttt{ROVSDT(IEL)} + \texttt{ISTAT}\ \dfrac{\rho_L^n\ |\Omega_l|}{\Delta t^n} - \texttt{ICONV}\ (\displaystyle\int_{\Omega_l} \mathrm{div}\,(\rho\,\underline{u})\ d\Omega) \end{array} \right.$$

- Calcul des termes sources de production, des termes $\phi_{ij,1} + \phi_{ij,2}$ et de la dissipation $-\dfrac{2}{3}\varepsilon\,\delta_{ij}$ :

  On effectue une boucle d'indice `IEL` sur les cellules $\Omega_l$ de centre $L$ :

$$\Rightarrow\ \texttt{TRPROD} = \frac{1}{2}(\mathcal{P}_{ii}^n)_L = \frac{1}{2}\left[\texttt{PRODUC(1,IEL)} + \texttt{PRODUC(2,IEL)} + \texttt{PRODUC(3,IEL)}\right]$$

$$\Rightarrow\ \texttt{TRRIJ}\ = \frac{1}{2}(R_{ii}^n)_L$$

$$\Rightarrow\ \texttt{SMBR(IEL)} = \texttt{SMBR(IEL)} + \rho_L^n|\Omega_l|\left[\frac{2}{3}\,\delta_{ij}\left(\frac{C_2}{2}\,(\mathcal{P}_{ii}^n)_L\ + (C_1 - 1)\ \varepsilon_L^n\right) + \ (1 - C_2)\ \texttt{PRODUC(ISOU,IEL)} - C_1\ \frac{2\,\varepsilon_L^n}{(R_{ii}^n)_L}\ (R_{ij}^n)_L\right]$$

[10] qui correspondent aux schémas convectif upwind pur et diffusif sans reconstruction.

$$\Rightarrow \texttt{ROVSDT(IEL)} = \texttt{ROVSDT(IEL)} + \rho_L^n \ |\Omega_l| \ (-\frac{1}{3} \ \delta_{ij} + 1) \ C_1 \ \frac{2 \ \varepsilon_L^n}{(R_{ii}^n)_L}$$

- Appel de `rijech` pour le calcul des termes d'écho de paroi $\phi_{ij,w}^n$ si `IRIJEC()` $= 1$ et ajout dans `SMBR`.
  `SMBR` $=$ `SMBR` $+ \phi_{ij,w}^n$
  Suivant son mode de calcul (`ICDPAR`), la distance à la paroi est directement accessible par `RA(IDIPAR+IEL-1)` (`|ICDPAR|` $= 1$) ou bien est calculée à partir de `IA(IIFAPA(IPHAS)+IEL - 1)`, qui donne pour l'élément `IEL` le numéro de la face de bord paroi la plus proche (`|ICDPAR|` $= 2$). Ces tableaux ont été renseigné une fois pour toutes au début de calcul.

- Appel de `rijthe` pour le calcul des termes de gravité $\mathcal{G}_{ij}^n$ et ajout dans `SMBR`.
  Ce calcul n'a lieu que si `IGRARI()` $= 1$. `SMBR` $=$ `SMBR` $+ \mathcal{G}_{ij}^n$

- Calcul de la partie explicite du terme de diffusion div $\left[\underline{\underline{A}} \ \underline{\text{grad}} \ R_{ij}^n\right]$, plus précisément des contributions du terme extradiagonal pris aux faces purement internes (remplissage du tableau `VISCF`), puis aux faces de bord (remplissage du tableau `VISCB`).

  ⋆ Appel de `grdcel` pour le calcul du gradient de $R_{ij}^n$ dans chaque direction. Ces gradients sont respectivement stockés dans les tableaux de travail `W1`, `W2` et `W3`.

  ⋆ boucle d'indice `IEL` sur les cellules $\Omega_l$ de centre $L$ pour le calcul de $\underline{\underline{E}}^n \ \underline{\text{grad}} \ R_{ij}^n$ aux cellules dans un premier temps :

  $$\Rightarrow \texttt{TRRIJ} = \frac{1}{2}(R_{ii}^n)_L$$

  $$\Rightarrow \texttt{CSTRIJ} = \rho_L^n \ C_S \ \frac{(R_{ii}^n)_L}{2 \, \varepsilon_L^n}$$

  $$\Rightarrow \texttt{W4(IEL)} = \rho_L^n \ C_S \ \frac{(R_{ii}^n)_L}{2 \, \varepsilon_L^n} \left[ (R_{12}^n)_L \ \texttt{W2(IEL)} + (R_{13}^n)_L \ \texttt{W3(IEL)} \right]$$

  $$\Rightarrow \texttt{W5(IEL)} = \rho_L^n \ C_S \ \frac{(R_{ii}^n)_L}{2 \, \varepsilon_L^n} \left[ (R_{12}^n)_L \ \texttt{W1(IEL)} + (R_{23}^n)_L \ \texttt{W3(IEL)} \right]$$

  $$\Rightarrow \texttt{W6(IEL)} = \rho_L^n \ C_S \ \frac{(R_{ii}^n)_L}{2 \, \varepsilon_L^n} \left[ (R_{13}^n)_L \ \texttt{W1(IEL)} + (R_{23}^n)_L \ \texttt{W2(IEL)} \right]$$

  ⋆ Appel de `vectds`[11] pour assembler $\left[\underline{\underline{E}}^n \ \underline{\text{grad}} \ R_{ij}^n\right] \cdot \underline{n}_{lm} S_{lm}$ aux faces $lm$.

  ⋆ Appel de `divmas` pour calculer la divergence du flux défini par `VISCF` et `VISCB`. Le résultat est stocké dans `W4`.
  Ajout au second membre `SMBR`.
  `SMBR` $=$ `SMBR` $+$ `W4`

  A l'issue de cette étape, seule la partie extradiagonale de la diffusion prise entièrement explicite :

  $$\sum_{m \in Vois(l)} \left[ \underline{\underline{E}}^n \ \underline{\text{grad}} \ R_{ij}^n \right]_{lm} \cdot \underline{n}_{lm} S_{lm}$$

  a été calculée.

- Calcul de la partie diagonale du terme de diffusion[12] :
  Comme on l'a déja vu, une partie de cette contribution sera traitée en implicite (celle relative à la composante normale du gradient) et donc ajoutée au second membre par `bilsc2` ; l'autre partie sera explicite et prise en compte dans $f_s^{exp}$.

  ⋆ On effectue une boucle d'indice `IEL` sur les cellules $\Omega_l$ de centre $L$ :

[11] Le résultat est stocké dans `VISCF` et `VISCB`. Dans `vectds`, les valeurs aux faces internes sont interpolées linéairement sans reconstruction et `VISCB` est mis à zéro.
[12] Seule la composante normale du gradient de $R_{ij}$ aux faces sera implicite.

$$\Rightarrow \texttt{TRRIJ} = \frac{1}{2}(R_{ii}^n)_L$$

$$\Rightarrow \texttt{CSTRIJ} = \rho_L^n \, C_S \, \frac{(R_{ii}^n)_L}{2\,\varepsilon_L^n}$$

$$\Rightarrow \texttt{W4(IEL)} = \rho_L^n \, C_S \, \frac{(R_{ii}^n)_L}{2\,\varepsilon_L^n} \, (R_{11}^n)_L$$

$$\Rightarrow \texttt{W5(IEL)} = \rho_L^n \, C_S \, \frac{(R_{ii}^n)_L}{2\,\varepsilon_L^n} \, (R_{22}^n)_L$$

$$\Rightarrow \texttt{W6(IEL)} = \rho_L^n \, C_S \, \frac{(R_{ii}^n)_L}{2\,\varepsilon_L^n} \, (R_{33}^n)_L$$

⋆ On effectue une boucle d'indice `IFAC` sur les faces purement internes $lm$ pour remplir le tableau `VISCF` :

⇒ Identification des cellules $\Omega_l$ et $\Omega_m$ de centre respectif $L$ (variable `II`) et $M$ (variable `JJ`), se trouvant de chaque côté de la face $lm$[13].

⇒ Calcul du carré de la surface de la face. La valeur est stockée dans le tableau `SURFN2`.

⇒ Interpolation du gradient de $R_{ij}^n$ à la face $lm$ (gradient facette $\left[\underline{\text{grad}}\ R_{ij}^n\right]_{lm}$) :

$$\begin{cases} \texttt{GRDPX} = \dfrac{1}{2}\left(\texttt{W1(II)} + \texttt{W1(JJ)}\right) \\ \texttt{GRDPY} = \dfrac{1}{2}\left(\texttt{W2(II)} + \texttt{W2(JJ)}\right) \\ \texttt{GRDPZ} = \dfrac{1}{2}\left(\texttt{W3(II)} + \texttt{W3(JJ)}\right) \end{cases}$$

⇒ Calcul du gradient de $R_{ij}^n$ normal à la face $lm$, $\left[\underline{\text{grad}}\ R_{ij}^n\right]_{lm} . \underline{n}_{lm} \, S_{lm}$.

`GRDSN` = `GRDPX SURFAC(1,IFAC)` + `GRDPY SURFAC(2,IFAC)` + `GRDPZ SURFAC(3,IFAC)`
`SURFAC` est le vecteur surface de la face `IFAC`.

⇒ calcul de $\left[\underline{\text{grad}}\ R_{ij}^n - (\underline{\text{grad}}\ R_{ij}^n \,.\, \underline{n}_{lm})\underline{n}_{lm}\right]$, les vecteurs étant calculés à la face $lm$ :

$$\begin{cases} \texttt{GRDPX} = \texttt{GRDPX} - \dfrac{\texttt{GRDSN}}{\texttt{SURFN2}} \texttt{SURFAC(1,IFAC)} \\ \texttt{GRDPY} = \texttt{GRDPY} - \dfrac{\texttt{GRDSN}}{\texttt{SURFN2}} \texttt{SURFAC(2,IFAC)} \\ \texttt{GRDPZ} = \texttt{GRDPZ} - \dfrac{\texttt{GRDSN}}{\texttt{SURFN2}} \texttt{SURFAC(3,IFAC)} \end{cases}$$

⇒ finalisation du calcul de l'expression totalement explicite

$$\left[\underline{\underline{D}}^n \left(\underline{\text{grad}}\ R_{ij}^n - (\underline{\text{grad}}\ R_{ij}^n \,.\, \underline{n}_{lm})\,\underline{n}_{lm}\right)\right] . \underline{n}_{lm}$$

$$\begin{aligned} \texttt{VISCF} = & \frac{1}{2}(\texttt{W4(II)} + \texttt{W4(JJ)})\,\texttt{GRDPX}\,\texttt{SURFAC(1,IFAC)}) + \\ & \frac{1}{2}(\texttt{W5(II)} + \texttt{W5(JJ)})\,\texttt{GRDPY}\,\texttt{SURFAC(2,IFAC)}) + \\ & \frac{1}{2}(\texttt{W6(II)} + \texttt{W6(JJ)})\,\texttt{GRDPZ}\,\texttt{SURFAC(3,IFAC)}) \end{aligned}$$

⋆ Mise à zéro du tableau `VISCB`.

[13] La normale à la face est orientée de L vers M.

⋆ Appel de `divmas` pour calculer la divergence de :

$$\underline{\underline{D}}^{n}\ \left(\underline{\text{grad}}\ R_{ij}^{n} - (\underline{\text{grad}}\ R_{ij}^{n}\,.\,\underline{n}_{lm})\underline{n}_{lm}\right)$$

défini aux faces dans `VISCF` et `VISCB`.

Le résultat est stocké dans le tableau `W1`.
Ajout au second membre `SMBR`.
`SMBR` = `SMBR` + `W1`

- Calcul de la viscosité orthotrope $\gamma_{lm}^{n}$ à la face $lm$ de la variable principale $R_{ij}^{n}$
  Ce calcul permet au sous-programme `codits` de compléter le second membre `SMBR` par :

$$\begin{aligned}&\sum_{m\in Vois(l)} \mu_{lm}^{n}\ \left(\underline{\text{grad}}\ R_{ij}^{n}\,.\,\underline{n}_{lm}\right) S_{lm} + \sum_{m\in Vois(l)} \left(\underline{\text{grad}}\ R_{ij}^{n}\,.\,\underline{n}_{lm}\right) \left[\underline{\underline{D}}^{n}\,\underline{n}_{lm}\right]_{lm}\,.\,\underline{n}_{lm}\ S_{lm}\\ &= \sum_{m\in Vois(l)} (\,\mu_{lm}^{n} + \gamma_{lm}^{n}\,)\ \left(\underline{\text{grad}}\ R_{ij}^{n}\,.\,\underline{n}_{lm}\right) S_{lm}\end{aligned} \tag{II.18.24}$$

  sans préciser la nature de la face $lm$, *via* l'appel à `bilsc2` et de disposer de la quantité $(\mu_{lm}^{n}+\gamma_{lm}^{n})$ pour construire sa matrice simplifiée.

  ⋆ On effectue une boucle d'indice `IEL` sur les cellules $\Omega_l$ :

  $\Rightarrow$ `TRRIJ` $= \dfrac{1}{2}(R_{ii}^{n})_L$

  $\Rightarrow$ `RCSTE` $= \rho_L^n\ C_S\ \dfrac{(R_{ii}^{n})_L}{2\,\varepsilon_L^n}$

  $\Rightarrow$ `W1(IEL)` $= \mu^n + \rho_L^n\ C_S\ \dfrac{(R_{ii}^{n})_L}{2\,\varepsilon_L^n}\ (R_{11}^{n})_L$

  $\Rightarrow$ `W2(IEL)` $= \mu^n + \rho_L^n\ C_S\ \dfrac{(R_{ii}^{n})_L}{2\,\varepsilon_L^n}\ (R_{22}^{n})_L$

  $\Rightarrow$ `W3(IEL)` $= \mu^n + \rho_L^n\ C_S\ \dfrac{(R_{ii}^{n})_L}{2\,\varepsilon_L^n}\ (R_{33}^{n})_L$

  ⋆ Appel de `visort` pour calculer la viscosité orthotrope [14] $\gamma_{lm}^{n} = (\underline{\underline{D}}^{n}\,\underline{n}_{lm}).\underline{n}_{lm}$ aux faces $lm$
  Le résultat est stocké dans les tableaux `VISCF` et `VISCB`.

- appel de `codits` pour la résolution de l'équation de convection/diffusion/termes sources de la variable $R_{ij}$. Le terme source est `SMBR`, la viscosité `VISCF` aux faces purement internes ( resp. `VISCB` aux faces de bord) et `FLUMAS` le flux de masse interne ( resp. `FLUMAB` flux de masse au bord). Le résultat est la variable $R_{ij}$ au temps $n+1$, donc $R_{ij}^{n+1}$. Elle est stockée dans le tableau `RTP` des variables mises à jour.

**• Appel de `reseps` pour la résolution de la variable $\varepsilon$**

On décrit ci-dessous le sous-programme `reseps`, les commentaires du sous-programme `resrij` ne seront pas répétés puisque les deux sous-programmes ne diffèrent que par leurs termes sources.

- Initialisation à zéro de `SMBR` et `ROVSDT`.
- Lecture et prise en compte des termes sources utilisateur pour la variable $\varepsilon$ :

[14] Comme dans le sous-programme `viscfa`, on multiplie la viscosité par $\dfrac{S_{lm}}{\overline{L'M'}}$, où $S_{lm}$ et $\overline{L'M'}$ représentent respectivement la surface de la face $lm$ et la mesure algébrique du segment reliant les projections des centres des cellules voisines sur la normale à la face. On garde dans ce sous-programme la possibilité d'interpoler la viscosité aux cellules linéairement ou d'utiliser une moyenne harmonique. La viscosité au bord est celle de la cellule de bord correspondante.

Appel de `ustsri` pour charger les termes sources utilisateurs. Ils sont stockés dans les tableaux suivants :
pour la cellule $\Omega_l$ représentée par `IEL` de centre $L$, on a :

$$\left\{ \begin{array}{ll} \texttt{ROVSDT(IEL)} & = |\Omega_l|\ \alpha_\varepsilon \\ \texttt{SMBR(IEL)} & = |\Omega_l|\ \beta_\varepsilon \end{array} \right.$$

On affecte alors les valeurs adéquates au second membre `SMBR` et à la diagonale `ROVSDT` comme suit :

$$\left\{ \begin{array}{ll} \texttt{SMBR(IEL)} & = \texttt{SMBR(IEL)} + \ |\Omega_l|\ \alpha_\varepsilon\ \varepsilon_L^n \\ \texttt{ROVSDT(IEL)} & = \max\ (-\ |\Omega_l|\ \alpha_\varepsilon, 0) \end{array} \right.$$

- Calcul du terme source de masse si $\Gamma_L > 0$ :

$$\left\{ \begin{array}{l} \texttt{SMBR(IEL)} = \texttt{SMBR(IEL)} + |\Omega_l|\ \Gamma_L\ (\varepsilon_L^{in} - \varepsilon_L^n) \\ \texttt{ROVSDT(IEL)} = \texttt{ROVSDT(IEL)} + |\Omega_l|\ \Gamma_L \end{array} \right.$$

- Calcul du terme d'accumulation de masse et du terme instationnaire
On stocke `W1` $= \displaystyle\int_{\Omega_l} \mathrm{div}\,(\rho\,\underline{u})\,d\Omega$ calculé par `divmas` à l'aide des flux de masse internes et aux bords.
On incrémente ensuite `SMBR` et `ROVSDT`.

$$\left\{ \begin{array}{ll} \texttt{SMBR(IEL)} & = \texttt{SMBR(IEL)} + \texttt{ICONV}\ \varepsilon_L^n\ (\displaystyle\int_{\Omega_l} \mathrm{div}\,(\rho\,\underline{u})\ d\Omega) \\ \texttt{ROVSDT(IEL)} & = \texttt{ROVSDT(IEL)} + \texttt{ISTAT}\ \dfrac{\rho_L^n\ |\Omega_l|}{\Delta t^n} - \texttt{ICONV}\ (\displaystyle\int_{\Omega_l} \mathrm{div}\,(\rho\,\underline{u})\ d\Omega) \end{array} \right.$$

- Traitement du terme de production $\rho\,C_{\varepsilon_1}\,\dfrac{\varepsilon}{k}\,\mathcal{P}$ et du terme de dissipation $-\,\rho\,C_{\varepsilon_2}\,\dfrac{\varepsilon}{k}\,\varepsilon$
pour cela, on effectue une boucle d'indice `IEL` sur les cellules $\Omega_l$ de centre $L$ :

$$\Rightarrow\ \texttt{TRPROD} = \frac{1}{2}(\mathcal{P}_{ii}^n)_L = \frac{1}{2}\left[\texttt{PRODUC(1,IEL)} + \texttt{PRODUC(2,IEL)} + \texttt{PRODUC(3,IEL)}\right]$$

$$\Rightarrow\ \texttt{TRRIJ}\ = \frac{1}{2}(R_{ii}^n)_L$$

$$\Rightarrow\ \texttt{SMBR(IEL)} = \texttt{SMBR(IEL)} + \rho_L^n|\Omega_l|\left[-C_{\varepsilon_2}\ \frac{2\,(\varepsilon_L^n)^2}{(R_{ii}^n)_L} + C_{\varepsilon_1}\ \frac{\varepsilon_L^n}{(R_{ii}^n)_L}\ (\mathcal{P}_{ii}^n)_L\right]$$

$$\Rightarrow\ \texttt{ROVSDT(IEL)} = \texttt{ROVSDT(IEL)} + C_{\varepsilon_2}\ \rho_L^n\ |\Omega_l|\ \frac{2\,\varepsilon_L^n}{(R_{ii}^n)_L}$$

- Appel de `rijthe` pour le calcul des termes de gravité $\mathcal{G}_\varepsilon^n$ et ajout dans `SMBR`.
`SMBR` $=$ `SMBR` $+\ \mathcal{G}_\varepsilon^n$
Ce calcul n'a lieu que si `IGRARI()` $= 1$.

- Calcul de la diffusion de $\varepsilon$
Le terme $\mathrm{div}\left[\mu\,\underline{\mathrm{grad}}\ (\varepsilon) + \underline{\underline{A'}}\,\underline{\mathrm{grad}}\ (\varepsilon)\right]$ est calculé exactement de la même manière que pour les tensions de Reynolds $R_{ij}$ en remplaçant $\underline{\underline{A}}$ par $\underline{\underline{A'}}$.

- Appel de `codits` pour la résolution de l'équation de convection/diffusion/termes sources de la variable principale $\varepsilon$. Le résultat $\varepsilon^{n+1}$ est stocké dans le tableau `RTP` des variables mises à jour.

**• clippings finaux**

On passe enfin dans le sous-programme `clprij` pour faire un clipping éventuel des variables $R_{ij}^{n+1}$ et $\varepsilon^{n+1}$. Le sous-programme `clprij` est appelé[15] avec `ICLIP` $= 2$ . Cette option [16] contient l'option `ICLIP` $= 1$ et permet de vérifier l'inégalité de Cauchy-Schwarz sur les grandeurs extra-diagonales du tenseur $\underline{\underline{R}}$ (pour $i \neq j$, $|R_{ij}|^2 \leq R_{ii}R_{jj}$).

[15] L'option `ICLIP` $= 1$ consiste en un clipping minimal des variables $R_{ii}$ et $\varepsilon$ en prenant la valeur absolue de ces variables puisqu'elles ne peuvent être que positives.

[16] Quand la valeur des grandeurs $R_{ii}$ ou $\varepsilon$ est négative, on la remplace par le minimum entre sa valeur absolue et (1,1) fois la valeur obtenue au pas de temps précédent.

## 18.4 Points à traiter

Sauf mention explicite, $\phi$ représentera une tension de Reynolds ou la dissipation turbulente ($\phi = R_{ij}$ ou $\varepsilon$).

- La vitesse utilisée pour le calcul de la production est explicite. Est-ce qu'une implicitation peut améliorer la précision temporelle de $\phi$ [17] ?
- Dans quelle mesure le terme d'écho de paroi est-il valide ? En effet, ce terme est remis en question par certains auteurs.
- On peut envisager la résolution d'un système hyperbolique pour les tensions de Reynolds afin d'introduire un couplage avec le champ de vitesse.
- Le flux au bord `VISCB` est annulé dans le sous-programme `vectds`. Peut-on envisager de mettre au bord la valeur de la variable concernée à la cellule de bord correspondant? De même, il faudrait se pencher sur les hypothèses sous-jacentes à l'annulation des contributions aux bords de `VISCB` lors du calcul de :

$$\left[\underline{\underline{D}}^n \left(\underline{\text{grad}}\ R_{ij}^n - (\underline{\text{grad}}\ R_{ij}^n \cdot \underline{n}_{lm})\, \underline{n}_{lm}\right)\right] \cdot \underline{n}_{lm}.$$

- Un problème de pondération apparaît plus généralement. Si on prend la partie explicite de $\underline{\underline{D}}\,\underline{\text{grad}}\ (\phi)$, la pondération aux faces internes utilise le coefficient $\frac{1}{2}$ avec pondération séparée de $\underline{\underline{D}}$ et $\underline{\text{grad}}\ (\phi)$, alors que pour $\underline{\underline{E}}\,\underline{\text{grad}}\ (\phi)$, on calcule d'abord ce terme aux cellules pour ensuite l'interpoler linéairement aux faces [18]. Ceci donne donc deux types d'interpolations pour des termes de même nature.
- On laisse la possibilité dans `visort` d'utiliser une moyenne harmonique aux faces. Est-ce que ceci est valable puisque les interpolations utilisées lors du calcul de la partie explicite de $\underline{\underline{A}}\,\underline{\text{grad}}\ \phi$ sont des moyennes arithmétiques ?
- Les techniques adoptées lors du clipping sont à revoir.
- On utilise dans le cadre du modèle $R_{ij} - \varepsilon$ une semi-implicitation de termes comme $\phi_{ij,1}$ ou $-\rho\, C_{\varepsilon_2}\, \frac{\varepsilon}{k}\, \varepsilon$. On peut envisager le même type d'implicitation dans `turbke` même en présence du couplage $k - \varepsilon$.
- L'adoption d'une résolution découplée fait perdre l'invariance par rotation.
- La formulation et l'implantation des conditions aux limites de paroi devront être vérifiées. En effet, il semblerait que, dans certains cas, des phénomènes "oscillatoires" apparaissent, sans qu'il soit aisé d'en déterminer la cause.
- L'implicitation partielle (du fait de la résolution découplée) des conditions aux limites conduit souvent à des calculs instables. Il conviendrait d'en connaître la raison. L'implicitation partielle avait été mise en œuvre afin de tenter d'utiliser un pas de temps plus grand dans le cas de jets axisymétriques en particulier.

[17] Cette remarque peut être généralisée. En effet, peut-on envisager d'actualiser les variables déjà résolues (sans réactualiser les variables turbulentes après leur résolution)? Ceci obligerait à modifier les sous-programmes tels que `condli` qui sont appelés au début de la boucle en temps.

[18] Cette interpolation se fait dans `vectds` avec des coefficients de pondération aux faces.

# 19- Sous-programme `viscfa`

## 19.1 Fonction

Dans ce sous-programme est calculé le coefficient de diffusion isotrope aux faces. Ce coefficient fait intervenir la valeur de la viscosité aux faces multipliée par le rapport surface de la face sur la distance algébrique $\overline{I'J'}$ ou $\overline{I'F}$(*cf.* figure II.19.1), rapport résultant de l'intégration du terme de diffusion. Par analogie du terme calculé, ce sous-programme est aussi appelé par le sous-programme `resolp` pour calculer le coefficient "diffusif" de la pression faisant intervenir le pas de temps.
La valeur de la viscosité aux faces est déterminée soit par une moyenne arithmétique, soit par une moyenne harmonique de la viscosité au centre des cellules, suivant le choix de l'utilisateur. Par défaut, cette valeur est calculée par une moyenne arithmétique.

## 19.2 Discrétisation

On rappelle dans la figure II.19.1, la définition des différents points géométriques utilisés par la suite.

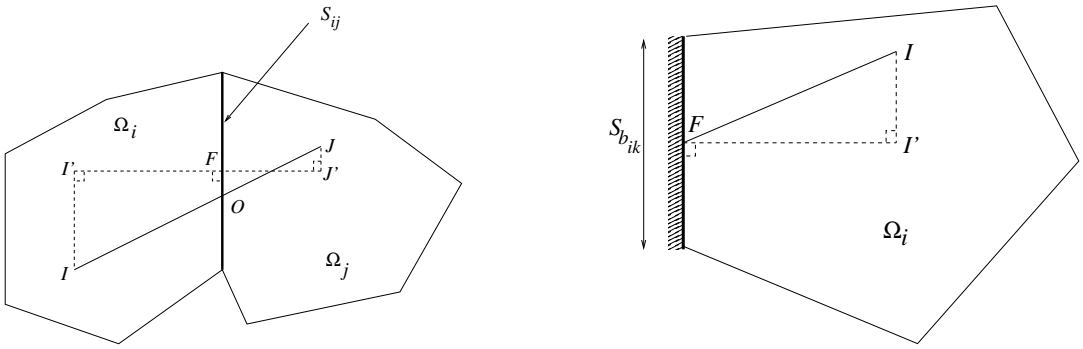



Figure II.19.1: Définition des différentes entités géométriques pour les faces internes (gauche) et de bord (droite).

L'intégration du terme de diffusion sur une cellule $\Omega_i$ est la suivante :

$$\int_{\Omega_i} \text{div}(\mu \underline{\text{grad}}\ (f))\, d\Omega = \sum_{j \in Vois(i)} \mu_{ij} \frac{f_{J'} - f_{I'}}{\overline{I'J'}} \,.\, S_{ij} + \sum_{k \in \gamma_b(i)} \mu_{b_{ik}} \frac{f_{b_{ik}} - f_{I'}}{\overline{I'F}} \,.\, S_{b_{ik}} \qquad \text{(II.19.1)}$$

Dans ce sous-programme, on calcule les termes de diffusion $\mu_{ij} \dfrac{S_{ij}}{\overline{I'J'}}$ et $\mu_{b_{ik}} \,.\, \dfrac{S_{b_{ik}}}{\overline{I'F}}$.

La valeur de la viscosité sur la face interne $ij$, $\mu_{ij}$, est calculée :

- ⋆ soit par moyenne arithmétique :

$$\mu_{ij} = \alpha_{ij}\mu_i + (1 - \alpha_{ij})\mu_j \qquad \text{(II.19.2)}$$

avec $\alpha_{ij} = 0.5$ car ce choix semble stabiliser, bien que cette interpolation soit d'ordre 1 en espace en convergence.

- ⋆ soit par moyenne harmonique :

$$\mu_{ij} = \frac{\mu_i \, \mu_j}{\alpha_{ij}\mu_i + (1 - \alpha_{ij})\mu_j}$$

avec $\alpha_{ij} = \dfrac{\overline{FJ'}}{\overline{I'J'}}$.

La valeur de la viscosité sur la face de bord $ik$, $\mu_{b_{ik}}$, est définie ainsi :

$$\mu_{b_{ik}} = \mu_I.$$

REMARQUE

Lors de l'appel de `viscfa` par le sous-programme `resolp`, le terme à considérer est :

$$\text{div}(\,\Delta t^n \ \underline{\text{grad}}\ (\delta p)\,)$$

soit :

$$\mu = \mu^n = \Delta t$$

## 19.3 Mise en œuvre

La valeur de la viscosité au centre des cellules est entrée en argument *via* la variable `VISTOT`. On calcule sa valeur moyenne aux faces et on la multiplie par le rapport surface `SURFN` sur la distance algébrique `DIST` pour une face interne (`SURFBN` et `DISTBR` respectivement pour une face de bord). La valeur du terme de diffusion résultant est mise dans le vecteur `VISCF` pour une face interne et `VISCB` pour une face de bord.
La variable `IMVISF` détermine quel type de moyenne est utilisé pour calculer la viscosité aux faces.
Si `IMVISF`= 0, la moyenne est arithmétique, sinon la moyenne est harmonique.

## 19.4 Points à traiter

L'obtention des interpolations utilisées dans le code *Code_Saturne* du paragraphe 19.2 est résumée dans le rapport de Davroux et al[1]. Les auteurs de ce rapport ont montré que, pour un maillage monodimensionnel irrégulier et avec une viscosité non constante, la convergence mesurée est d'ordre 2 en espace avec l'interpolation harmonique et d'ordre 1 en espace avec l'interpolation linéaire (pour des solution régulières).
Par conséquent, il serait préférable d'utiliser l'interpolation harmonique pour calculer la valeur de la viscosité aux faces. Des tests de stabilité seront nécessaires au préalable.
De même, on envisage d'extrapoler la viscosité sur les faces de bord plutôt que de prendre la valeur de la viscosité au centre de la cellule jouxtant cette face.
Dans le cas de la moyenne arithmétique, l'utilisation de la valeur 0.5 pour les coefficients $\alpha_{ij}$ serait à revoir.

[1] Davroux A., Archambeau F. et Hérard J.M., Tests numériques sur quelques méthodes de résolution d'une équation de diffusion en volumes finis, HI-83/00/027/A.

# 20- Sous-programme `visort`

## 20.1 Fonction

Dans ce sous-programme est calculé le coefficient de diffusion "orthotrope" aux faces. Ce type de coefficient se rencontre pour la diffusion de $R_{ij}$ et $\varepsilon$ en $R_{ij} - \varepsilon$ ( *cf.* `turrij`), ainsi que pour la correction de pression dans le cadre de l'algorithme avec couplage vitesse-pression renforcé (`resolp`). Ce coefficient fait intervenir la valeur de la viscosité aux faces multipliée par le rapport surface de la face sur la distance algébrique $\overline{I'J'}$, rapport résultant de l'intégration du terme de diffusion. La valeur de la viscosité aux faces est basée soit sur une moyenne arithmétique, soit sur une moyenne harmonique de la viscosité au centre des cellules.

## 20.2 Discrétisation

La figure II.20.1 rappelle les diverses définitions géométriques pour les faces internes et les faces de bord.

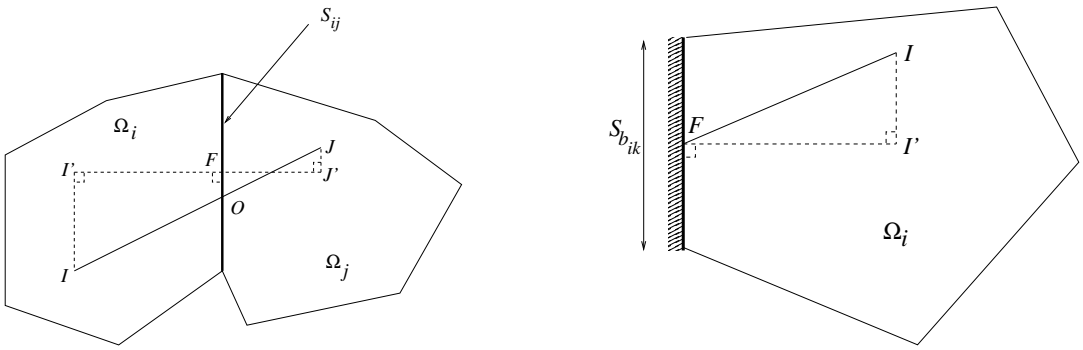



Figure II.20.1: Définition des différentes entités géométriques pour les faces internes (gauche) et de bord (droite).

L'intégration du terme de diffusion "orthotrope" sur une cellule est la suivante :

$$\int_{\Omega_i} \mathrm{div}\,(\underline{\underline{\mu}}\ \underline{\mathrm{grad}}\ f)\,d\Omega = \sum_{j \in Vois(i)} (\underline{\underline{\mu}}\ \underline{\mathrm{grad}}\ f)_{ij} \cdot \underline{S}_{ij} + \sum_{k \in \gamma_b(i)} (\underline{\underline{\mu}}\ \underline{\mathrm{grad}}\ f)_{b_{ik}} \cdot \underline{S}_{b_{ik}} \tag{II.20.1}$$

avec :

$$\underline{\underline{\mu}} = \begin{bmatrix} \mu_x & 0 & 0 \\ 0 & \mu_y & 0 \\ 0 & 0 & \mu_z \end{bmatrix} \tag{II.20.2}$$

et :

$$\begin{aligned} \underline{S}_{ij} &= S_{ij}\underline{n}_{ij} \\ \underline{S}_{b_{ik}} &= S_{b_{ik}}\underline{n}_{b_{ik}} \end{aligned} \tag{II.20.3}$$

Le terme $(\underline{\underline{\mu}}\ \underline{\mathrm{grad}}\ (f))_{ij}\underline{n}_{ij}$ est calculé à l'aide de la décomposition suivante :

$$(\underline{\underline{\mu}}\ \underline{\mathrm{grad}}\ f)_{ij} = (\underline{\mathrm{grad}}\ f \cdot \underline{n}_{ij})\ \underline{\underline{\mu}}\ \underline{n}_{ij} + (\underline{\mathrm{grad}}\ f . \underline{\tau}_{ij})\ \underline{\underline{\mu}}\ \underline{\tau}_{ij} \tag{II.20.4}$$

où $\underline{\tau}_{ij}$ représente un vecteur tangent (unitaire) à la face. Une décomposition similaire est utilisée aux faces de bord.

Dans la matrice, seul le terme $(\underline{\text{grad}}\ f \,.\, \underline{n}_{ij})\ \underline{\underline{\mu}}\ \underline{n}_{ij}$ est intégrable facilement en implicite. Par conséquent, la partie projetée sur $\underline{\tau}_{ij}$ est :

- négligée dans le cas du calcul des échelles de temps relatives au couplage vitesse-pression renforcé,
- traitée en explicite dans les termes de diffusion de $R_{ij} - \varepsilon$ (*cf.* `turrij`).

L'intégration implicite du terme de diffusion s'écrit :

$$\int_{\Omega_i} \text{div}\,(\underline{\underline{\mu}}\ \underline{\text{grad}}\ f)\,d\Omega = \sum_{j \in Vois(i)} (\underline{\underline{\mu}}\ \underline{n}_{ij}) \,.\, \underline{S}_{ij}\ \frac{f_{J'} - f_{I'}}{\overline{I'J'}} + \sum_{k \in \gamma_b(i)} (\underline{\underline{\mu}}\ \underline{n}_{b_{ik}}) \,.\, \underline{S}_{b_{ik}}\ \frac{f_{b_{ik}} - f_{I'}}{\overline{I'F}} \qquad \text{(II.20.5)}$$

Dans ce sous-programme, on calcule le terme $\dfrac{(\underline{\underline{\mu}}\ \underline{n}_{ij}) \,.\, \underline{S}_{ij}}{\overline{I'J'}}$ à l'aide la formule :

$$(\underline{\underline{\mu}}\ \underline{n}_{ij}) \,.\, \underline{n}_{ij} = \mu_{ij}^{moy} = \mu_{ij}^{x} (n_{ij}^{x})^2 + \mu_{ij}^{y} (n_{ij}^{y})^2 + \mu_{ij}^{z} (n_{ij}^{z})^2$$

soit encore :

$$\mu_{ij}^{moy} = \frac{\mu_{ij}^{x} (S_{ij}^{x})^2 + \mu_{ij}^{y} (S_{ij}^{y})^2 + \mu_{ij}^{z} (S_{ij}^{z})^2}{S_{ij}^2}$$

Au bord, on calcule de même :

$$\frac{(\underline{\underline{\mu}}\ \underline{n}_{b_{ik}}) \,.\, \underline{S}_{b_{ik}}}{\overline{I'F}}$$

avec :

$$(\underline{\underline{\mu}}\ \underline{n}_{b_{ik}}) \,.\, \underline{n}_{b_{ik}} = \mu_{b_{ik}}^{moy} = \frac{\mu_{I}^{x} (S_{b_{ik}}^{x})^2 + \mu_{I}^{y} (S_{b_{ik}}^{y})^2 + \mu_{I}^{z} (S_{b_{ik}}^{z})^2}{S_{b_{ik}}^2}$$

La valeur de la viscosité dans une direction $l$ sur la face, $\mu_{ij}^{l}$, est calculée :

- soit par interpolation linéaire :

$$\mu_{ij}^{l} = \alpha_{ij} \mu_{i}^{l} + (1 - \alpha_{ij}) \mu_{j}^{l} \qquad \text{(II.20.6)}$$

  avec $\alpha_{ij} = 0.5$ car ce choix semble stabiliser bien que cette interpolation soit d'ordre 1 en espace en convergence,

- soit par interpolation harmonique :

$$\mu_{ij}^{l} = \frac{\mu_{i}^{l}\ \mu_{j}^{l}}{\alpha_{ij} \mu_{i}^{l} + (1 - \alpha_{ij}) \mu_{j}^{l}}$$

  où :

$$\alpha_{ij} = \frac{\overline{FJ'}}{\overline{I'J'}}$$

## 20.3 Mise en œuvre

La viscosité orthotrope au centre des cellules est entrée en argument *via* les variables $\mathtt{W}_1$, $\mathtt{W}_2$ et $\mathtt{W}_3$. On calcule la valeur moyenne de chaque viscosité aux faces de façon arithmétique ou harmonique. Ensuite, on calcule la viscosité équivalente correspondant à $(\underline{\underline{\mu}}\ \underline{n}_{ij}) \,.\, \dfrac{\underline{S}_{ij}}{\overline{I'J'}}$ pour les faces internes et à $(\underline{\underline{\mu}}\ \underline{n}_{b_{ik}}) \,.\, \dfrac{\underline{S}_{b_{ik}}}{\overline{I'F}}$ pour les faces de bord.

Cette écriture fait intervenir les vecteurs surface stockés dans le tableau `SURFAC`, la norme de la surface `SURFN` et la distance algébrique `DIST` pour une face interne (`SURFBO`, `SURFBN` et `DISTBR` respectivement pour une face de bord). La valeur du terme de diffusion résultant est mise dans le vecteur `VISCF` (`VISCB` aux faces de bord).
La variable `IMVISF` détermine quel type de moyenne est utilisé pour calculer la viscosité dans une direction à la face. Si `IMVISF`= 0, alors la moyenne est arithmétique, sinon la moyenne est harmonique).

## 20.4 Points à traiter

L'obtention des interpolations utilisées dans le code *Code_Saturne* du paragraphe 20.2 est résumée dans le rapport de Davroux et al[1]. Les auteurs de ce rapport ont montré que, pour un maillage monodimensionnel irrégulier et avec une viscosité non constante, la convergence mesurée est d'ordre 2 en espace avec l'interpolation harmonique et d'ordre 1 en espace avec l'interpolation linéaire (pour des solutions régulières). Par conséquent, il serait préférable d'utiliser l'interpolation harmonique pour calculer la valeur de la viscosité aux faces. Des tests de stabilité seront nécessaires au préalable.
De même, on envisage d'extrapoler la viscosité sur les faces de bord plutôt que de prendre la valeur de la viscosité de la cellule jouxtant cette face.
Dans le cas de la moyenne arithmétique, l'utilisation de la valeur 0.5 pour les coefficients $\alpha_{ij}$ serait à revoir.

[1] Davroux A., Archambeau F. et Hérard J.M., Tests numériques sur quelques méthodes de résolution d'une équation de diffusion en volumes finis, HI-83/00/027/A.

# 21- Sous-programme `vissec`

## 21.1 Fonction

Dans ce sous-programme sont calculés les termes de gradient transposé et de viscosité secondaire (fluide newtonien). Ils seront traités en explicite pur. En effet, si on traitait ces termes en implicite, cela reviendrait à coupler les équations des différentes composantes de la vitesse, ce qui n'est pas compatible avec le schéma de résolution actuel (*cf.* sous-programme `navsto`).

## 21.2 Discrétisation

L'intégration des termes de gradient transposé $\mathrm{div}\,(\mu_{tot}\,{}^t\underline{\underline{\mathrm{grad}}}\,(\underline{v}))$ et de viscosité secondaire $-\frac{2}{3}\,\underline{\mathrm{grad}}\,(\mu_{tot}\,\mathrm{div}(\underline{v}))$ est la suivante[1] :

$$\begin{aligned}
&\int_{\Omega_i} \mathrm{div}(\mu_{tot}\,{}^t\underline{\underline{\mathrm{grad}}}\,(\underline{v}))\,d\Omega \\
&= \sum_{l=x,y,z}\left[\sum_{j\in Vois(i)} \mu_{tot,ij}((\frac{\partial v_x}{\partial l})_{moy,ij}\,n^x_{ij} + (\frac{\partial v_y}{\partial l})_{moy,ij}\,n^y_{ij} + (\frac{\partial v_z}{\partial l})_{moy,ij}\,n^z_{ij})S_{ij}\right. \\
&\left. + \sum_{k\in\gamma_b(i)} \mu_{tot,b_{ik}}((\frac{\partial v_x}{\partial l})_{moy,b_{ik}}\,n^x_{b_{ik}} + (\frac{\partial v_y}{\partial l})_{moy,b_{ik}}\,n^y_{b_{ik}} + (\frac{\partial v_z}{\partial l})_{moy,b_{ik}}\,n^z_{b_{ik}})S_{b_{ik}}\right]\underline{e}_l \\
&-\frac{2}{3}\int_{\Omega_i} \underline{\mathrm{grad}}\,(\mu_{tot}\mathrm{div}\,(\underline{v}))\,d\Omega \\
&= -\frac{2}{3}\sum_{l=x,y,z}\left[\sum_{j\in Vois(i)} (\mu_{tot}\,\mathrm{div}(\underline{v}))_{ij}S^l_{ij} + \sum_{k\in\gamma_b(i)} (\mu_{tot}\,\mathrm{div}(\underline{v}))_{b_{ik}}S^l_{b_{ik}}\right]\underline{e}_l
\end{aligned}$$

Le terme de viscosité $\mu_{tot}$ est calculé en fonction du modèle de turbulence utilisé :

$$\mu_{tot} = \begin{cases} \mu + \mu_t & \text{pour les modèles à viscosité turbulente ou en LES,} \\ \mu & \text{pour les modèles au second ordre ou en laminaire.} \end{cases}$$

où $\mu$ et $\mu_t$ représentent respectivement la viscosité dynamique moléculaire et la viscosité dynamique turbulente.

## 21.3 Mise en œuvre

### 21.3.1 Terme de gradient transposé

Avant de calculer l'intégration des deux termes, on calcule dans un premier temps la viscosité totale en fonction du modèle de turbulence considéré.
Les termes calculés dans ce sous-programme, appelé par `preduv`, interviennent dans le second membre de l'équation de quantité de mouvement, et sont donc directement stockés dans le tableau correspondant `TRAV`.
Le terme $\mathrm{div}\,(\mu_{tot}\,{}^t\underline{\underline{\mathrm{grad}}}\,(\underline{v}))$ est calculé en opérant comme suit.

[1] la viscosité de volume $\kappa$ est supposée nulle, cf. `navsto`

On effectue une boucle sur les composantes $v_\alpha$ où $\alpha = x, y, z$ de la vitesse ($\alpha$ correspond à `ISOU` dans le code) :

- on calcule le gradient cellule de $v_\alpha$ par un appel au sous-programme `grdcel`.
- on initialise un tableau nommé $\mathtt{W}_6$ à la valeur 1 pour les cellules internes, et à la valeur 0 pour les cellules de bord. Ce tableau sert par la suite à ne pas considérer la contribution du terme de gradient transposé sur les cellules de bord. En effet, on ne sait pas écrire de conditions aux limites correctes pour le gradient transposé. On préfère donc tout simplement annuler son effet sur les cellules de bord (*cf.* paragraphe 21.4).
- pour chaque direction $l$ ($l = x, y, z$), $l$ correspondant à `IDIM` dans le code, on calcule pour chaque cellule $\Omega_i$ dont le centre correspond à la variable `II` ou `JJ` (pour les centres voisins) dans le code :

$$\begin{aligned}
&\mathtt{TRAV}(i,l) = \mathtt{TRAV}(i,l) \\
&+\mathtt{W}_6(i) \left[ \sum_{j \in Vois(i)} \mu_{tot,ij} \left(\frac{\partial v_\alpha}{\partial l}\right)_{moy,ij} S^\alpha_{ij} + \sum_{k \in \gamma_b(i)} \mu_{tot,b_{ik}} \left(\frac{\partial v_\alpha}{\partial l}\right)_{moy,b_{ik}} S^\alpha_{b_{ik}} \right] \\
&\text{avec } \left(\frac{\partial v_\alpha}{\partial l}\right)_{moy,ij} = \frac{1}{2}\left[ \left(\frac{\partial v_\alpha}{\partial l}\right)_i + \left(\frac{\partial v_\alpha}{\partial l}\right)_j \right]
\end{aligned}$$

Fin de la boucle sur les composantes de la vitesse.

### 21.3.2 Terme de viscosité secondaire

Le terme de seconde viscosité $-\frac{2}{3}\underline{\text{grad}}\,(\mu_{tot}\text{div}\,(\underline{v}))$ est calculé de la façon suivante :

- on calcule la valeur de la vitesse sur la face $ij$ en divisant le flux masse connu à la face par la densité moyenne $\rho_{moy,ij}$ de la face ($\rho_{moy,ij} = \frac{\rho_i + \rho_j}{2}$).
- on calcule ensuite l'intégrale volumique de la divergence de la vitesse sur chaque cellule en appelant le sous-programme `divmas`.
- on calcule alors pour chaque cellule $\Omega_i$ le terme $-\frac{2}{3}(\mu_{tot}\,\text{div}(\underline{v}))_i$ que l'on met dans le tableau de travail $\mathtt{W}_4$. La valeur de ce terme sur la face interne $ij$ est obtenue en prenant la moyenne arithmétique des valeurs des deux cellules avoisinantes (tableau `VISCF`) et celle sur la face de bord est prise égale la valeur de la cellule avoisinante (tableau `VISCB`).
- on calcule alors pour chaque direction $l$ le terme final, *i.e.* :

$$\mathtt{TRAV}(i,l) = \mathtt{TRAV}(i,l) - \frac{2}{3}\left[ \sum_{j \in Vois(i)} (\mu_{tot}\,\text{div}(\underline{v}))_{moy,ij}\, S^l_{ij} + \sum_{k \in \gamma_b(i)} (\mu_{tot}\text{div}\,(\underline{v}))_{moy,b_{ik}}\, S^l_{b_{ik}} \right]$$

Le traitement est similaire pour le terme de viscosité de volume dans le module compressible.

## 21.4 Points à traiter

L'intégration du terme de gradient transposé pose un problème de compatibilité. En effet, le gradient transposé provient de l'écriture de la divergence du tenseur des contraintes (*cf.* `preduv`), soit :

$$\text{div}\,(\underline{\sigma}) = \text{div}\,(-p\underline{\underline{Id}} + \underline{\tau})$$

où :

$$\underline{\underline{\tau}} = 2\mu \, [\underbrace{\frac{1}{2}\,(\underline{\underline{\mathrm{grad}}}\ \underline{v} \ + \ {}^{t}\underline{\underline{\mathrm{grad}}}\ \underline{v})}_{\text{partie 1}} - \underbrace{\frac{2}{3}\,tr(\,\frac{1}{2}\,(\underline{\underline{\mathrm{grad}}}\ \underline{v} \ + \ {}^{t}\underline{\underline{\mathrm{grad}}}\ \underline{v}))\ \underline{\underline{Id}}}_{\text{partie 2}}\,]$$

Or, lorsque l'on intègre la première partie de la divergence de $\underline{\underline{\tau}}$, on implicite le terme $\mathrm{div}\,(\mu\,\underline{\underline{\mathrm{grad}}}\ \underline{v})$ et on explicite le gradient transposé $\mathrm{div}\,(\mu\,{}^{t}\underline{\underline{\mathrm{grad}}}\ \underline{v})$. Ce traitement fait intervenir la vitesse au centre des cellules. Elle ne vérifie pas exactement la condition $\mathrm{div}\,(\rho\underline{v}) = 0$. En effet, au cours de l'étape de correction, on utilise un filtre Rhie et Chow (*cf.* `resolp`) et la vitesse n'est mise à jour qu'à la fin de l'étape. Par contre, lorsque l'on intègre la deuxième partie de la divergence de $\underline{\underline{\tau}}$ de façon explicite, on utilise la vitesse issue du flux masse aux faces qui vérifie la condition $\mathrm{div}\,(\rho\underline{v}) = 0$ (du moins à $\rho$ constant, l'interpolation de $\rho$ à la face étant également un point à considérer). Ainsi, la discrétisation de ces deux parties n'est pas totalement cohérente. Il serait utile de baser la discrétisation de ces termes sur une vitesse vérifiant la contrainte $\mathrm{div}\,(\rho\underline{v}) = 0$.

Pour la même raison, il est difficile de connaître les conditions aux limites du terme en gradient transposé. Sur les cellules de bord, on sait uniquement que la contrainte totale normale doit équilibrer le frottement et toutes les autres forces. Or, le tenseur des contraintes est scindé en une partie explicite et une partie implicite, donc c'est un peu difficile d'utiliser cette condition physique.

Actuellement, la contribution aux cellules de bord du terme de gradient transposé est annulée, ce qui élimine l'influence des conditions aux limites mais n'est naturellement pas satisfaisant. Quelques essais d'intégration des conditions aux limites pour ce terme n'ont pas été concluants jusqu'à présent. Cependant, des essais supplémentaires sont envisageables.

# 22- Sous-programme vortex

## 22.1 Fonction

Ce sous-programme est dédié à la génération des conditions d'entrée turbulente utilisées en LES.

La méthode des vortex est basée sur une approche de tourbillons ponctuels. L'idée de la méthode consiste à injecter des tourbillons 2D dans le plan d'entrée du calcul, puis à calculer le champ de vitesse induit par ces tourbillons au centre des faces d'entrée.

## 22.2 Discrétisation

Pour utiliser la méthode, on se place tout d'abord dans un repère local défini de manière à ce que le plan $(0yz)$, où sont injectés les vortex, soit confondu avec le plan d'entrée du calcul (voir figure II.22.1).

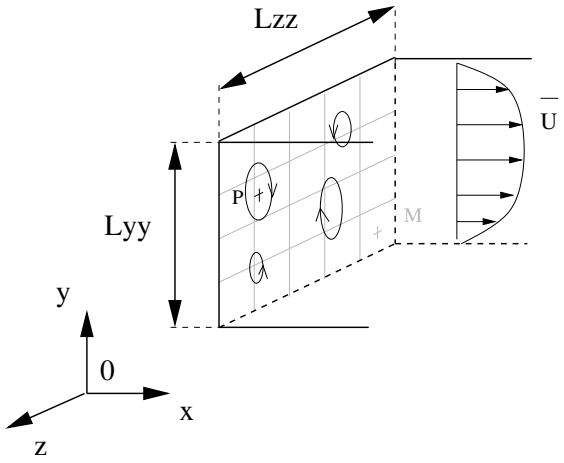



Figure II.22.1: Définiton des différentes grandeurs dans le repère local correspondant à l'entrée d'une conduite de section carrée.

$u$, $v$ et $w$ sont les composantes de la vitesse fluctuante (principale et transverse) dans ce plan, et $\omega(y,z) = \dfrac{\partial w}{\partial y} - \dfrac{\partial v}{\partial z}$ la vorticité dans la direction normale au plan d'entrée. $\overline{U}(y,z)$ représente ici la vitesse principale moyenne imposée par l'utilisateur dans le plan d'entrée.

Chaque vortex $p$ va être caractérisé par sa fonction de forme $\xi$ (identique pour tous les vortex), sa circulation $\Gamma_p$, son rayon $\sigma_p$ et les coordonnées $(y_p, z_p)$ du point $P$ où est situé le vortex dans le plan $(0yz)$.

Pour cela, on suppose que la vorticité générée par un vortex $p$ au point $M$ de coordonnée $(y,z)$ s'écrit :

$$\omega_p(y,z) = \Gamma_p \, \xi_{\sigma_p}(r)$$

où $r = \sqrt{(y-y_p)^2 + (z-z_p)^2}$ est la distance séparant le point $M$ du point $P$.

Dans la méthode implantée, la fonction de forme est de type gaussienne modifiée :

$$\xi_\sigma(r) = \frac{1}{2\pi\sigma^2}\left(2e^{-\frac{r^2}{2\sigma^2}} - 1\right) e^{-\frac{r^2}{2\sigma^2}}$$

Le champ de vitesse induit par cette distribution de vorticité s'obtient par inversion des deux équations de poisson suivantes qui sont déduites de la condition d'incompressibilité dans la plan[1] :

$$\frac{\partial \omega}{\partial y} = \Delta w \quad \text{et} \quad \frac{\partial \omega}{\partial y} = -\Delta v$$

Dans le cas général, ce système peut être intégré à l'aide de la formule de Biot et Savart.

Dans le cas d'une distribution de vorticité de type gaussienne modifiée, les composantes de vitesse vérifient :

$$\begin{cases} v_p(y,x) = -\dfrac{1}{2\pi}\dfrac{(z-z_p)}{r^2}\left(1-e^{-\frac{r^2}{2\sigma^2}}\right) e^{-\frac{r^2}{2\sigma^2}} \\ w_p(y,z) = \dfrac{1}{2\pi}\dfrac{(y-y_p)}{r^2}\left(1-e^{-\frac{r^2}{2\sigma^2}}\right) e^{-\frac{r^2}{2\sigma^2}} \end{cases}$$

Ces relations s'étendent de façon immédiate au cas de $N$ vortex distincts. Le champ de vitesse induit par la distribution de vorticité

$$\omega(y,z) = \sum_{p=1}^{N} \Gamma_p \, \xi_{\sigma_p}(r) \tag{II.22.1}$$

vaut au point $M$ :

$$v(x,y) = \textstyle\sum_{p=1}^{N} \Gamma_p \, v_p(y,z) \quad \text{et} \quad w(y,z) = \sum_{p=1}^{N} \Gamma_p \, w_p(y,z)$$

## 22.2.1 Paramètres physiques

### Marche en temps

La position initiale de chaque vortex est tirée de manière aléatoire. On calcul les déplacements successifs de chacun des vortex dans le plan d'entrée par intégration explicite du champ de vitesse lagrangien :

$$\frac{dy_p}{dt} = V(y,z) \quad \text{et} \quad \frac{dz_p}{dt} = W(y,z)$$

Les vortex sont alors assimilés à des particules ponctuelles qui sont convectées par le champ $(V,W)$. Ce champ peut être imposé par des tirages aléatoires ou bien déduit de la vitesse induite par les vortex dans le plan d'entrée. Dans ce cas $V(x,y) = \overline{V}(y,z) + v(y,z)$ et $W(y,z) = \overline{W}(y,z) + w(y,z)$ où $\overline{V}$ et $\overline{W}$ sont les composantes de la vitesse transverse moyenne qu'impose l'utilisateur à l'aide des fichiers de données.

### Intensité et durée de vie des vortex

Il serait possible, à partir de l'équation de transport de la vorticité, d'obtenir un modèle d'évolution pour l'intensité du vecteur tourbillon $\omega_p$ associé à chacun des vortex. En pratique, on préfère utiliser un modèle simplifié dans lequel la circulation des tourbillons ne dépend que de la postion de ces derniers à l'instant considéré. La circulation initiale de chaque vortex est alors obtenue à partir de la relation :

$$|\Gamma_p| = 4\sqrt{\frac{\pi \, S \, k}{3N\,[2ln(3) - 3ln(2)]}}$$

où $S$ est la surface du plan d'entrée, $N$ le nombre de vortex, et $k$ l'énergie cinétique turbulente au point où se trouve le vortex à l'instant considéré. Le signe de $\Gamma_p$ correspond au sens de rotation du vortex et est tiré aléatoirement.

[1] *i.e* $\dfrac{\partial v}{\partial y} + \dfrac{\partial w}{\partial w} = 0$

Ce paramètre est celui qui contrôle l'intensité des fluctuations. Sa dépendance en $k$ exprime que, plus l'écoulement est turbulent, plus les vortex sont intenses. La valeur de $k$ est spécifiée par l'utilisateur. Elle peut être constante ou imposée à partir de profils d'énergie cinétique turbulente en entrée.

Pour éviter que des structures trop allongées ne se développent au niveau de l'entrée, l'utilisateur doit également spécifier un temps limites $\tau_p$ au bout duquel le vortex $p$ va être détruit. Ce temps $\tau_p$ peut être pris constant ou estimé au moyen de la relation :

$$\tau_p = \frac{5 C_\mu k^{\frac{3}{2}}}{\varepsilon \overline{U}}$$

$\overline{U}$ et $\varepsilon$ représentent respectivement la vitesse moyenne principale et la dissipation turbulente au point où est initialement généré le vortex ($C_\mu = 0,09$).
Lorsque le temps écoulé depuis la création du vortex $p$ est supérieur à $\tau_p$, le vortex est détruit et un nouveau vortex généré (sa position et le signe de $\Gamma_p$ sont tirés de façon aléatoire).

### Taille des vortex

La taille des vortex peut être prise constante, ou calculée à partir des relations :

$$\sigma = \frac{C_\mu^{\frac{3}{4}} k^{\frac{3}{2}}}{\varepsilon} \quad \text{ou} \quad \sigma = max[L_t, L_k]$$

avec:

$$L_t = \sqrt{\left(\frac{5\nu k}{\varepsilon}\right)} \quad \text{et} \quad L_k = 200 \left(\frac{\nu^3}{\varepsilon}\right)^{\frac{1}{4}}$$

où $\nu$, $k$ et $\varepsilon$ sont la viscosité dynamique, l'énergie cinétique turbulente et la dissipation turbulente au point où se trouve le vortex.

Dans tous les cas, la taille du vortex doit être supérieure à la taille des mailles en entrée afin que le vortex soit effectivement simulé.

## 22.2.2 Conditions aux limites

Le champ de vitesse généré à l'aide de la relation II.22.2 ne tient pas compte *a priori* des conditions aux limites appliquées sur les bords du plan d'entrée. Pour obtenir des valeurs de la vitesse qui soient cohérentes sur les frontières du domaine d'entrée, des "vortex images", pseudo-vortex situés en dehors du domaine d'entrée, sont générés à des positions particulières et leur champ de vitesse associé est superposé au champ précédemment calculé.
Seuls les cas d'une conduite rectangulaire et d'une conduite circulaire permettent la génération de ces pseudo-vortex. On distingue pour cela trois types de conditions aux limites.

### Condition de paroi

On crée, pour chaque vortex $P$ contenu dans le plan d'entrée, un vortex image $P'$ identique à $P$ (*i.e* de même caractéristiques) et symétrique de $P$ par rapport au point $J$ ($J$ étant la projection orthogonalement à la paroi du point $M$ correspondant au centre de la face où l'on cherche à calculer la vitesse). La figure II.22.2 illustre la technique dans le cas d'une conduite carrée. Dans ce cas les coordonnées du vortex situé en $P'$ vérifient $(y_{p'} + y_p)/2 = y_J$ et $(z_{p'} + z_p)/2 = z_J$. Le champ de vitesse perçu depuis le point $M$ au niveau du point $J$ est nul, ce qui est bien l'effet recherché.

### Condition de symétrie

La technique est identique à celle utilisée pour les conditions de paroi, mais seule la composante pour la vitesse normale au bord est modifiée dans ce cas.

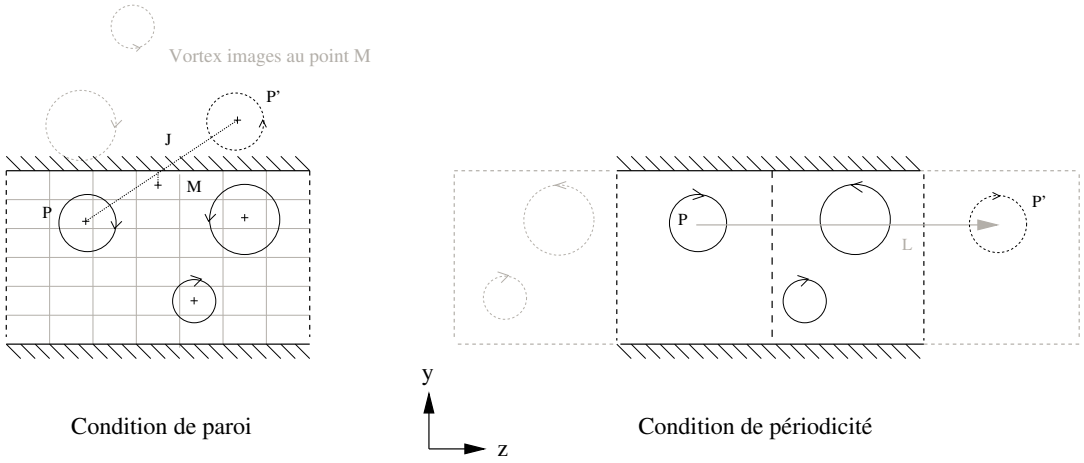



Figure II.22.2: Principe de génération des "vortex images" suivant le type de conditions aux limites dans une conduite carrée.

### Condition de périodicité

On crée pour chaque vortex, un vortex images $P'$ identique à $P$ mais translaté d'une quantité $L$ correspondant à la longueur qui sépare les deux plans de la section d'entrée où sont appliquées les conditions de périodicité. Dans le cas où il y a deux directions de périodicité, on crée deux vortex image.

## 22.2.3 Composante de vitesse principale

La méthode des vortex ne générant pas de fluctuation $u$ de la vitesse dans la direction principale, la dernière composante est calculée à partir d'une équation de Langevin. Les coefficients de cette équation sont déterminés par identification des expressions obtenues pour les contraintes de Reynolds en $R_{ij} - \varepsilon$. Dans le cas d'un écoulement en canal plan, cette technique conduit à l'équation :

$$\frac{du}{dt} = -\frac{C_1}{2T}u + \left(\frac{2}{3}C_2 - 1\right)\frac{\partial U}{\partial y}v + \sqrt{C_0\varepsilon}dW_i$$

avec $T = \dfrac{k}{\varepsilon}$, $C_1 = 1,8$, $C_2 = 0,6$, $C_0 = \frac{14}{15}$, et $dW_i$ une variable alétoire Gaussienne de variance $\sqrt{dt}$.

En pratique, l'équation de Langevin n'améliore pas vraiment les résultats. Elle n'est utilisée que dans le cas de conduites circulaires.

# 22.3 Mise en œuvre

- ⋆ Aprés une étape de préparation de la mémoire (`memvor`), on repère dans `usvort` les faces d'entrée pour lesquelles la méthode va être utilisée.

- ⋆ Vérification des dimensions rentrées (`vervor`).

- ⋆ Le sous-programme `vorpre` se charge ensuite de préparer le calcul (transmission de la géométrie des entrées à tous les processeurs en cas de parallélisme, et construction d'un tableau de connectivité). Le sous-programme procède ainsi :

- On compte, pour chaque entrée `IENT`, le nombre de faces où est appliquée la méthode. Celui-ci est stocké dans le tableau `ICVOR(IENT)`. Un passage dans la sous-routine `memvor` (avec `IAPPEL = 2`) permet d'allouer la mémoire nécessaire à cette phase de préparation.
- Pour chaque processeur, on stocke les coordonnées des faces d'entrée repérées précédemment dans les tableaux de travail `RA(IW1X),RA(IW1Y),RA(IW1Z),...`
- On regarde ensuite pour chaque processeur (boucle `IPROC=1, NRANGP-1`), si le processeur `IPROC` a des données à envoyer aux autres processeurs (afin que tous disposent des coordonnées).
  - Si c'est le cas : `ICVOR(IENT)>0`, et on place les données à envoyer dans les tableaux de travail `RA(IW2X),RA(IW2Y),RA(IW2Z),...`. La valeur `NCOMV = ICVOR(IENT)` correspond alors à la longueur des tableaux à envoyer.
  - Sinon, on ne fait rien et `NCOM=0`.
- Le processeur numéro `IPROC` distribue à tous les autres processeurs la valeur `NCOM`. Si `NCOM > 0`, il envoie également les données contenues dans les tableaux de travails `RA(IW2X),...`. Ces données sont ensuite stockées par tous les processeurs dans les tableaux `RA(IXYZV+III),...` afin de libérer les tableaux de travail pour la communication suivante, et l'indice `III = III + NCOM` est incrémenté de manière à ranger les valeurs de façon chronologique.

  → Au final de la boucle sur `IPROC`, chaque processeur dispose des coordonnées des faces d'entrée pour lesquelles la méthode va être utilisée, et il est donc simple de construire la connectivité.

- Construction de la connectivité. Au final, la vitesse au centre de la `II` éme face d'entrée utilisant la méthode est contenue à la `IA(IIFAGL+II)` ème ligne du tableau `RA(IUVORT)`.
- La routine se termine par un appel au sous-programme `memvor` ( avec `IAPPEL = 3`) afin de réserver la mémoire utile à la méthode des vortex.

Cette phase d'initialisation est réalisée une seule fois au début du calcul. C'est après cette phase seulement que commence la méthode des vortex proprement dite.

⋆ Initialisation des variables avant intervention utilisateur (`inivor`).

⋆ Appel au sous-programme utilisateur `usvort` (`IAPPEL = 2`).

⋆ Vérification des paramètres rentrés (`vervor`).

⋆ Calcul de la vitesse par la méthode des vortex (`vortex`)

- Initialisation du calcul génération du champ initial par appel au sous-programme `vorini` :
  - Construction du repére local (et calcul de l'équation du plan d'entrée suivant les cas), localisation du centre de l'entrée, et transformation des coordonnées de l'entrée dans le repère local. Les tableaux `YZCEL(II,1)` et `YZCEL(II,2)` contiennent les coordonnées des faces du plan d'entrée une fois ramenées dans le repère $(0yz)$ (`II` est compris entre 1 et `NCEVOR` où `NCEVOR`=`ICVOR` représente le nombre de faces pour lesquelles la méthode va être utilisée a cette entrée).
  - Lecture du fichier de données, et initialisation des tableaux `XDAT`, `YDAT`, `UDAT`, `VDAT`, `WDAT`, `DUYDAT`, `KDAT`, `EPSDAT`, ...
  - Si on ne fait pas de suite (`ISUIVO=0`) ou que l'on réinitialise le calcul (`INITVO=1`), tirage aléatoire de la position des vortex et de leur sens de rotation, ainsi que calcul de leur durée de vie. Les positions sont stockées dans les tableaux `YZVOR(IVOR,1)` et `YZVOR(IVOR,2)` (`IVOR` désignant le numéro du vortex).

  - Stockage de la vitesse principale moyenne au centre de la cellule dans le tableau `XU`, et recherche pour chaque vortex, de la face d'entrée qui lui est la plus proche.
- Déplacement des vortex par appel au sous-programme `vordep` :
  - Convection des vortex.
  - Traitement des conditions aux limites. Les vortex qui sortent du domaine de calcul sont replacés à leur position d'origine.
  - Régénération des vortex "morts". Si le temps de vie cumulé `TEMPS(II)` du vortex `II` est supérieur à sont temps de vie limite `TPSLIM(II)`, alors le vortex est détruit, et un nouveau vortex est généré.
  - Recherche pour chaque vortex de la face d'entrée qui lui est la plus proche après déplacement (mise à jour du tableau `IVORCE`).
- Calcul du champ de vitesse induit par appel au sous-programme `vorvit` :
  - Calcul de l'intensité du vortex.
  - Calcul de la taille du vortex.
  - Calcul du champ de vitesse induit par l'ensemble des vortex au centre des faces d'entrée.
  - Traitement suivant les cas, des conditions de périodicité de symétrie et des conditions de paroi par génération de vortex images.
  - Ajout de la vitesse moyenne dans les directions transverse aux tableaux `XV` et `XW`.
- Génération des fluctuations de vitesse dans la direction principale par appel au sous-programme `vorlgv`.

⋆ appel au sous-programme `vor2cl` :

- Communication en cas de parallélisme de la vitesse calculée en entrée par le processeur 0 aux autres processeurs.
- Application des conditions aux limites après utilisation d'un changement de repère éventuel.

## 22.4 Points à traiter

Il serait possible de gagner de la mémoire en liberant l'espace aloué aux tableaux `IW1X`,...,`IW2V` après le passage dans `vorpre`.

# Part III

# Module compressible

# 1- Sous-programme `cfbl**`

## 1.1 Fonction

On s'intéresse à la résolution des équations de Navier-Stokes en compressible, en particulier pour des configurations sans choc. Le schéma global correspond à une extension des algorithmes volumes finis mis en œuvre pour simuler les équations de Navier-Stokes en incompressible.

Dans les grandes lignes, le schéma est constitué d'une étape "acoustique" fournissant la masse volumique (ainsi qu'une prédiction de pression et un débit acoustique), suivie de la résolution de l'équation de la quantité de mouvement ; on résout ensuite l'équation de l'énergie et, pour terminer, la pression est mise à jour. Moyennant une contrainte sur la valeur du pas de temps, le schéma permet d'assurer la positivité de la masse volumique.

La thermodynamique prise en compte à ce jour est celle des gaz parfaits, mais l'organisation du code à été prévue pour permettre à l'utilisateur de fournir ses propres lois.

Pour compléter la présentation, on pourra se reporter à la référence suivante :
[**Mathon**] P. Mathon, F. Archambeau, J.-M. Hérard : "Implantation d'un algorithme compressible dans *Code_Saturne* ", HI-83/03/016/A

Le cas de validation "tube à choc" de la version 1.2 de *Code_Saturne* permettra également d'apporter quelques compléments (tube à choc de Sod, discontinuité de contact instationnaire, double détente symétrique, double choc symétrique).

### 1.1.1 Notations

| Symbole | Unité | Signification |
|---|---|---|
| $C_p$, $C_{p_i}$ | $J/(kg.K)$ | capacité calorifique à pression constante $C_p = \frac{\partial h}{\partial T}\big)_P$ |
| $C_v$, $C_{vi}$ | $J/(kg.K)$ | capacité calorifique à volume constant $C_v = \frac{\partial \varepsilon}{\partial T}\big)_\rho$ |
| $\mathcal{D}_{f/b}$ | $m^2/s$ | diffusivité moléculaire du composant $f$ dans le bain |
| $E$ | $J/m^3$ | énergie totale volumique $E = \rho e$ |
| $F$ | | centre de gravité d'une face |
| $H$ | $J/kg$ | enthalpie totale massique $H = \frac{E+P}{\rho}$ |
| $I$ | | point de co-location de la cellule $i$ |
| $I'$ | | pour une face $ij$ partagée entre les cellules $i$ et $j$, $I'$ est le projeté de $I$ sur la normale à la $ij$ passant par $F$, centre de $ij$ |
| $K$ | $kg/(m.s)$ | diffusivité thermique |
| $M$, $M_i$ | $kg/mol$ | masse molaire ($M_i$ pour le constituant $i$) |
| $P$ | $Pa$ | pression |
| $\underline{Q}$ | $kg/(m^2.s)$ | vecteur quantité de mouvement $\underline{Q} = \rho \underline{u}$ |
| $\underline{Q}_{ac}$ | $kg/(m^2.s)$ | vecteur quantité de mouvement issu de l'étape acoustique |
| $Q$ | $kg/(m^2.s)$ | norme de $\underline{Q}$ |
| $R$ | $J/(mol.K)$ | constante universelle des gaz parfaits |
| $S$ | $J/(K.m^3)$ | entropie volumique |
| $\mathcal{S}$ | $[f].kg/(m^3.s)$ | Terme de production/dissipation volumique pour le scalaire $f$ |
| $T$ | $K$ | température ($> 0$) |
| $Y_i$ | | fraction massique du composé $i$ ($0 \leqslant Y_i \leqslant 1$) |

| Symbole | Unité | Signification |
|---|---|---|
| $c^2$ | $(m/s)^2$ | carré de la vitesse du son $c^2 = \frac{\partial P}{\partial \rho}\Big)_s$ |
| $e$ | $J/kg$ | énergie totale massique $e = \varepsilon + \frac{1}{2}u^2$ |
| $\underline{f}_v$ | $N/kg$ | $\rho \underline{f}_v$ représente le terme source volumique pour la quantité de mouvement : gravité, pertes de charges, tenseurs des contraintes turbulentes, forces de Laplace... |
| $\underline{g}$ | $m/s^2$ | accélération de la pesanteur |
| $h$ | $J/kg$ | enthalpie massique $h = \varepsilon + \frac{P}{\rho}$ |
| $i$ | | indice faisant référence à la cellule $i$ ; $f_i$ est la valeur de la variable $f$ associée au point de co-location $I$ |
| $I'$ | | indice faisant référence à la cellule $i$ ; $f'_I$ est la valeur de la variable $f$ associée au point $I'$ |
| $\underline{j} \wedge \underline{B}$ | $N/m^3$ | forces de Laplace |
| $r$, $r_i$ | $J/(kg.K)$ | constante massique des gaz parfaits $r = \frac{R}{M}$ (pour le constituant $i$, on a $r_i = \frac{R}{M_i}$) |
| $s$ | $J/(K.kg)$ | entropie massique |
| $t$ | $s$ | temps |
| $\underline{u}$ | $m/s$ | vecteur vitesse |
| $u$ | $m/s$ | norme de $\underline{u}$ |

| Symbole | Unité | Signification |
|---|---|---|
| $\beta$ | $kg/(m^3.K)$ | $\beta = \frac{\partial P}{\partial s}\big)_\rho$ |
| $\gamma$ | $kg/(m^3.K)$ | constante caractéristique d'un gaz parfait $\gamma = \frac{C_p}{C_v}$ |
| $\varepsilon$ | $J/kg$ | énergie interne massique |
| $\kappa$ | $kg/(m.s)$ | viscosité dynamique en volume |
| $\lambda$ | $W/(m.K)$ | conductivité thermique |
| $\mu$ | $kg/(m.s)$ | viscosité dynamique ordinaire |
| $\rho$ | $kg/m^3$ | densité |
| $\underline{\varphi}_f$ | $[f].kg/(m^2.s)$ | vecteur flux diffusif du composé $f$ |
| $\varphi_f$ | $[f].kg/(m^2.s)$ | norme de $\underline{\varphi}_f$ |
| $\underline{\underline{\Sigma}}^v$ | $kg/(m^2.s^2)$ | tenseur des contraintes visqueuses |
| $\underline{\Phi}_s$ | $W/m^2$ | vecteur flux conductif de chaleur |
| $\Phi_s$ | $W/m^2$ | norme de $\underline{\Phi}_s$ |
| $\Phi_v$ | $W/kg$ | $\rho\Phi_v$ représente le terme source volumique d'énergie, comprenant par exemple l'effet Joule $\underline{j} \cdot \underline{E}$, le rayonnement... |

### 1.1.2 Système d'équations laminaires de référence

L'algorithme développé propose de résoudre l'équation de continuité, les équations de Navier-Stokes ainsi que l'équation d'énergie totale de manière conservative, pour des écoulements compressibles.

$$\left\{ \begin{array}{l} \dfrac{\partial \rho}{\partial t} + \mathrm{div}(\underline{Q}) = 0 \\ \dfrac{\partial \underline{Q}}{\partial t} + \underline{\mathrm{div}}(\underline{u} \otimes \underline{Q}) + \underline{\mathrm{grad}}\ P = \rho \underline{f}_v + \underline{\mathrm{div}}(\underline{\underline{\Sigma}}^v) \\ \dfrac{\partial E}{\partial t} + \mathrm{div}(\underline{u}(E+P)) = \rho \underline{f}_v \cdot \underline{u} + \mathrm{div}(\underline{\underline{\Sigma}}^v \underline{u}) - \mathrm{div}\,\underline{\Phi}_s + \rho \Phi_v \end{array} \right. \tag{III.1.1}$$

Nous avons présenté ici le système d'équations laminaires, mais il faut préciser que la turbulence ne pose pas de problème particulier dans la mesure où les équations supplémentaires sont découplées du système (III.1.1).

### 1.1.3 Expression des termes intervenant dans les équations

- Énergie totale volumique :

$$E = \rho e = \rho \varepsilon + \frac{1}{2}\rho u^2 \tag{III.1.2}$$

  avec l'énergie interne $\varepsilon(P, \rho)$ donnée par l'équation d'état

- Forces volumiques : $\rho \underline{f}_v$ (dans la plupart des cas $\rho \underline{f}_v = \rho \underline{g}$)

- Tenseur des contraintes visqueuses pour un fluide Newtonien :

$$\underline{\underline{\Sigma}}^v = \mu(\underline{\underline{\mathrm{grad}}}\ \underline{u} + {}^t\underline{\underline{\mathrm{grad}}}\ \underline{u}) + (\kappa - \frac{2}{3}\mu)\mathrm{div}\,\underline{u}\ \underline{\underline{Id}} \tag{III.1.3}$$

  avec $\mu(T, \ldots)$ et $\kappa(T, \ldots)$ mais souvent $\kappa = 0$

- Flux de conduction de la chaleur : loi de Fourier

$$\underline{\Phi}_s = -\lambda \underline{\mathrm{grad}}\ T \tag{III.1.4}$$

  avec $\lambda(T, \ldots)$

- Source de chaleur volumique : $\rho \Phi_v$

### 1.1.4 Équations d'état et expressions de l'énergie interne

#### Gaz parfait

Équation d'état : $P = \rho r T$

Énergie interne massique : $\varepsilon = \dfrac{P}{(\gamma - 1)\rho}$

Soit :

$$P = (\gamma - 1)\rho(e - \frac{1}{2}u^2) \tag{III.1.5}$$

### Mélange de gaz parfaits

On considère un mélange de $N$ constituants de fractions massiques $(Y_i)_{i=1...N}$

Équation d'état : $P = \rho\, r_{mélange}\, T$

Énergie interne massique : $\varepsilon = \dfrac{P}{(\gamma_{mélange} - 1)\rho}$

Soit :

$$P = (\gamma_{mélange} - 1)\rho(e - \frac{1}{2}u^2) \tag{III.1.6}$$

avec $\gamma_{mélange} = \dfrac{\sum_{i=1}^{N} Y_i C_{pi}}{\sum_{i=1}^{N} Y_i C_{vi}}$ et $r_{mélange} = \sum_{i=1}^{N} Y_i r_i$

### Equation d'état de Van der Waals

Cette équation est une correction de l'équation d'état des gaz parfaits pour tenir compte des forces intermoléculaires et du volume des molécules constitutives du gaz. On introduit deux coefficients correctifs : $a\ [Pa.m^6/kg^2]$ est lié aux forces intermoléculaires et $b\ [m^3/kg]$ est le covolume (volume occupé par les molécules).

Équation d'état : $(P + a\rho^2)(1 - b\rho) = \rho r T$

Énergie interne massique : $\varepsilon = \dfrac{(P + a\rho^2)(1 - b\rho)}{(\hat{\gamma} - 1)\rho} - a\rho$

Soit :

$$P = (\hat{\gamma} - 1)\frac{\rho}{(1 - b\rho)}(e - \frac{1}{2}u^2 + a\rho) - a\rho^2 \tag{III.1.7}$$

avec $\hat{\gamma} = 1 + \dfrac{r}{C_v} = \dfrac{C_p}{C_v}\left(\dfrac{P - a\rho^2(1 - 2b\rho)}{P + a\rho^2}\right) + \dfrac{2a\rho^2(1 - b\rho)}{P + a\rho^2}$

## 1.1.5 Calcul des grandeurs thermodymamiques

### Pour un gaz parfait à $\gamma$ constant

**Equation d'état :** $P = \rho r T$

On suppose connues la chaleur massique à pression constante $C_p$ et la masse molaire $M$ du gaz, ainsi que les variables d'état.

**Chaleur massique à volume constant :** $C_v = C_p - \dfrac{R}{M} = C_p - r$

**Constante caractéristique du gaz :** $\gamma = \dfrac{C_p}{C_v} = \dfrac{C_p}{C_p - r}$

**Vitesse du son :** $c^2 = \gamma \dfrac{P}{\rho}$

**Entropie :** $s = \dfrac{P}{\rho^\gamma}$ et $\beta = \left.\dfrac{\partial P}{\partial s}\right)_\rho = \rho^\gamma$

*Remarque :* L'entropie choisie ici n'est pas l'entropie physique, mais une entropie mathématique qui vérifie $c^2 \left.\dfrac{\partial s}{\partial P}\right)_\rho + \left.\dfrac{\partial s}{\partial \rho}\right)_P = 0$

**Pression :** $P = (\gamma - 1)\rho\varepsilon$

**Energie interne :** $\varepsilon = C_v T = \dfrac{1}{\gamma - 1}\dfrac{P}{\rho}$ avec $\varepsilon_{sup} = 0$

**Enthalpie :** $h = C_p T = \dfrac{\gamma}{\gamma - 1}\dfrac{P}{\rho}$

## Pour un mélange de gaz parfaits

Une intervention de l'utilisateur dans le sous-programme utilisateur `uscfth` est nécessaire pour pouvoir utiliser ces lois.

**Equation d'état :** $P = \rho\, r_{mél}\, T$ avec $r_{mél} = \sum_{i=1}^{N} Y_i r_i = \sum_{i=1}^{N} Y_i \dfrac{R}{M_i}$

On suppose connues la chaleur massique à pression constante des différents constituants $C_{p_i}$, la masse molaire $M_i$ des constituants du gaz, ainsi que les variables d'état (dont les fractions massiques $Y_i$).

**Masse molaire du mélange :** $M_{mél} = \left(\sum_{i=1}^{N} \dfrac{Y_i}{M_i}\right)^{-1}$

**Chaleur massique à pression constante du mélange :**

$C_{p_{mél}} = \sum_{i=1}^{N} Y_i C_{p_i}$

**Chaleur massique à volume constant du mélange :**

$C_{v mél} = \sum_{i=1}^{N} Y_i C_{vi} = C_{p_{mél}} - \dfrac{R}{M_{mél}} = C_{p_{mél}} - r_{mél}$

**Constante caractéristique du gaz :** $\gamma_{mél} = \dfrac{C_{p_{mél}}}{C_{v mél}} = \dfrac{C_{p_{mél}}}{C_{p_{mél}} - r_{mél}}$

**Vitesse du son :** $c^2 = \gamma_{mél}\dfrac{P}{\rho}$

**Entropie :** $s = \dfrac{P}{\rho^{\gamma_{mél}}}$ et $\beta = \dfrac{\partial P}{\partial s}\Big)_{\rho} = \rho^{\gamma_{mél}}$

**Pression :** $P = (\gamma_{mél} - 1)\rho\varepsilon$

**Energie interne :** $\varepsilon = C_{v\,mél}\, T$ avec $\varepsilon_{sup} = 0$

**Enthalpie :** $h = C_{p_{mél}}\, T = \dfrac{\gamma_{mél}}{\gamma_{mél} - 1}\dfrac{P}{\rho}$

### Pour un gaz de Van der Waals

Ces lois n'ont pas été programmées, mais l'utilisateur peut intervenir dans le sous-programme utilisateur `uscfth` s'il souhaite le faire.

**Equation d'état :** $(P + a\rho^2)(1 - b\rho) = \rho r T$

avec $a$ $[Pa.\,m^6/kg^2]$ lié aux forces intermoléculaires et $b$ $[m^3/kg]$ le covolume (volume occupé par les molécules).

On suppose connus les coefficients $a$ et $b$, la chaleur massique à pression constante $C_p$, la masse molaire $M$ du gaz et les variables d'état.

**Chaleur massique à volume constant :** $C_v = C_p - r\dfrac{P + a\rho^2}{P - a\rho^2(1 - 2b\rho)}$

**Constante "équivalente" du gaz :** $\hat{\gamma} = 1 + \dfrac{r}{C_v} = \dfrac{C_p}{C_v}\left(\dfrac{P - a\rho^2(1 - 2b\rho)}{P + a\rho^2}\right) + \dfrac{2a\rho^2(1 - b\rho)}{P + a\rho^2}$

**Vitesse du son :** $c^2 = \hat{\gamma}\dfrac{P + a\rho^2}{\rho(1 - b\rho)} - 2a\rho$

**Entropie :** $s = (P + a\rho^2)\left(\dfrac{1 - b\rho}{\rho}\right)^{\hat{\gamma}}$ et $\beta = \dfrac{\partial P}{\partial s}\Big)_{\rho} = \left(\dfrac{\rho}{1 - b\rho}\right)^{\hat{\gamma}}$

**Pression :** $P = (\hat{\gamma} - 1)\dfrac{\rho}{(1 - b\rho)}(\varepsilon + a\rho) - a\rho^2$

**Energie interne :** $\varepsilon = C_v T - a\rho$ avec $\varepsilon_{sup} = -a\rho$

**Enthalpie :** $h = \dfrac{\hat{\gamma} - b\rho}{\hat{\gamma} - 1}\dfrac{P + a\rho^2}{\rho} - 2a\rho$

## 1.1.6 Algorithme de base

On suppose connues toutes les variables au temps $t^n$ et on cherche à les déterminer à l'instant $t^{n+1}$. On résout en deux blocs principaux : d'une part le système masse-quantité de mouvement, de l'autre

l'équation portant sur l'énergie et les scalaires transportés. Dans le premier bloc, on distingue le traitement du système (couplé) acoustique et le traitement de l'équation de la quantité de mouvement complète.

Au début du pas de temps, on commence par mettre à jour les propriétés physiques variables (par exemple $\mu(T)$, $\kappa(T)$, $C_p(Y_1, \ldots, Y_N)$ ou $\lambda(T)$), puis on résout les étapes suivantes :

1. **Acoustique : sous-programme `cfmsvl`**
   Résolution d'une équation de convection-diffusion portant sur $\rho^{n+1}$.
   On obtient à la fin de l'étape $\rho^{n+1}$, $Q_{ac}^{n+1}$ et éventuellement une prédiction de la pression $P^{pred}(\rho^{n+1}, e^n)$.

2. **Quantité de mouvement : sous-programme `cfqdmv`**
   Résolution d'une équation de convection-diffusion portant sur $u^{n+1}$ qui fait intervenir $Q_{ac}^{n+1}$ et $P^{pred}$.
   On obtient à la fin de l'étape $u^{n+1}$.

3. **Énergie totale : sous-programme `cfener`**
   Résolution d'une équation de convection-diffusion portant sur $e^{n+1}$ qui fait intervenir $Q_{ac}^{n+1}$, $P^{pred}$ et $u^{n+1}$.
   On obtient à la fin de l'étape $e^{n+1}$ et une valeur actualisée de la pression $P(\rho^{n+1}, e^{n+1})$.

4. **Scalaires passifs**
   Résolution d'une équation de convection-diffusion standard par scalaire, avec $Q_{ac}^{n+1}$ pour flux convectif.

## 1.2 Discrétisation

On se reportera aux sections relatives aux sous-programmes `cfmsvl` (masse volumique), `cfqdmv` (quantité de mouvement) et `cfener` (énergie). La documentation du sous-programme `cfxtcl` fournit des éléments relatifs aux conditions aux limites.

## 1.3 Mise en œuvre

Le module compressible est une "physique particulière" activée lorsque le mot-clé `IPPMOD(ICOMPF)` est positif ou nul.

Dans ce qui suit, on précise les inconnues et les propriétés principales utilisées dans le module. On fournit également un arbre d'appel simplifié des sous-programmes du module : initialisation avec `initi1` puis (`iniva0` et) `inivar` et enfin, boucle en temps avec `tridim`.

### 1.3.1 Inconnues et propriétés

Les `NSCAPP` inconnues scalaires associées à la physique particulière sont définies dans `cfvarp` dans l'ordre suivant :

- la masse volumique `RTP(*,ISCA(IRHO(IPHAS)))`,
- l'énergie totale `RTP(*,ISCA(IENERG(IPHAS)))`,
- la température `RTP(*,ISCA(ITEMPK(IPHAS)))`

On souligne que la température est définie en tant que variable "RTP" et non pas en tant que propriété physique "PROPCE". Ce choix a été motivé par la volonté de simplifier la gestion des conditions aux limites, au prix cependant d'un encombrement mémoire légèrement supérieur (une grandeur RTP consomme plus qu'une grandeur PROPCE).

La pression et la vitesse sont classiquement associées aux tableaux suivants :

- pression : RTP(*,IPR(IPHAS))
- vitesse : RTP(*,IU(IPHAS)), RTP(*,IV(IPHAS)), RTP(*,IW(IPHAS)).

Outre les propriétés associées en standard aux variables identifiées ci-dessus, le tableau PROPCE contient également :

- la chaleur massique à volume constant $C_v$, stockée dans PROPCE(*,IPPROC(ICV(IPHAS))), si l'utilisateur a indiqué dans uscfth qu'elle était variable.
- la viscosité en volume PROPCE(*,IPPROC(IVISCV(IPHAS))) si l'utilisateur a indiqué dans uscfx2 qu'elle était variable.

Pour la gestion des conditions aux limites et en particulier pour le calcul du flux convectif par le schéma de Rusanov aux entrées et sorties (hormis en sortie supersonique), on dispose des tableaux suivants dans PROPFB :

- flux convectif de quantité de mouvement au bord pour les trois composantes dans les tableaux PROPFB(*,IPPROB(IFBRHU(IPHAS))) (composante $x$), PROPFB(*,IPPROB(IFBRHV(IPHAS))) (composante $y$) et PROPFB(*,IPPROB(IFBRHW(IPHAS))) (composante $z$)
- flux convectif d'énergie au bord PROPFB(*,IPPROB(IFBENE(IPHAS)))

et on dispose également dans IA :

- d'un tableau d'entiers dont la première "case" est IA(IIFBRU), dimensionné au nombre de faces de bord et permettant de repérer les faces de bord pour lesquelles on calcule le flux convectif par le schéma de Rusanov,
- d'un tableau d'entiers dont la première "case" est IA(IIFBET), dimensionné au nombre de faces de bord et permettant de repérer les faces de paroi à température ou à flux thermique imposé.

## 1.3.2 Arbre d'appel simplifié

| | | | | |
|---|---|---|---|---|
| `usini1` | | | | Initialisation des mots-clés utilisateur généraux et positionnement des variables |
| | `usppmo` | | | Définition du module "physique particulière" employé |
| | `varpos` | | | Positionnement des variables |
| | | `pplecd` | | Branchement des physiques particulières pour la lecture du fichier de données éventuel |
| | | `ppvarp` | | Branchement des physiques particulières pour le positionnement des inconnues |
| | | | `cfvarp` | Positionnement des inconnues spécifiques au module compressible |
| | | | | `uscfth` Appelé avec ICCFTH=-1, pour indiquer que $C_p$ et $C_v$ sont constants ou variables |
| | | | | `uscfx2` Conductivité thermique moléculaire constante ou variable et viscosité en volume constante ou variable (ainsi que leur valeur, si elles sont constantes) |
| | | `ppprop` | | Branchement des physiques particulières pour le positionnement des propriétés |
| | | | `cfprop` | Positionnement des propriétés spécifiques au module compressible |
| `ppini1` | | | | Branchement des physiques particulières pour l'initialisation des mots-clés spécifiques |
| | `cfini1` | | | Initialisation des mots-clés spécifiques au module compressible |
| | `uscfi1` | | | Initialisation des mots-clés utilisateur spécifiques au module compressible |

Table 23.1: Sous-programme `initi1` : initialisation des mots-clés et positionnement des variables

| | | | |
|---|---|---|---|
| `ppiniv` | | | Branchement des physiques particulières pour l'initialisation des variables |
| | `cfiniv` | | Initialisation des variables spécifiques au module compressible |
| | | `memcfv` | Réservation de tableaux de travail locaux |
| | | `uscfth` | Initialisation des variables par défaut (en calcul suite : seulement $C_v$ ; si le calcul n'est pas une suite : $C_v$, la masse volumique et l'énergie) |
| | | `uscfxi` | Initialisation des variables par l'utilisateur (seulement si le calcul n'est pas une suite) |

Table 23.2: Sous-programme `inivar` : initialisation des variables

| | | | |
|---|---|---|---|
| `phyvar` | | | Calcul des propriétés physiques variables |
| | `ppphyv` | | Branchement des physiques particulières pour le calcul des propriétés physiques variables |
| | | `cfphyv` | Calcul des propriétés physiques variables pour le module compressible |
| | | `uscfpv` | Calcul par l'utilisateur des propriétés physiques variables pour le module compressible ($C_v$ est calculé dans `uscfth` qui est appelé par `uscfpv`) |

Table 23.3: Sous-programme `tridim` : partie 1 (propriétés physiques)

| | | | |
|---|---|---|---|
| `dttvar` | | | Calcul du pas de temps variable |
| | `cfdttv` | | Calcul de la contrainte liée au CFL en compressible |
| | | `memcft` | Gestion de la mémoire pour le calcul de la contrainte en CFL |
| | | `cfmsfl` | Calcul du flux associé à la contrainte en CFL |
| `precli` | | | Initialisation des tableaux avant calcul des conditions aux limites (`IITYPF`, `ICODCL`, `RCODCL`) |
| | `ppprcl` | | Initialisations spécifiques aux différentes physiques particulières avant calcul des conditions aux limites (pour le module compressible : `IZFPPP`, `IA(IIFBRU)`, `IA(IIFBET)`, `RCODCL`, flux convectifs pour la quantité de mouvement et l'énergie) |
| `ppclim` | | | Branchement des physiques particulières pour les conditions aux limites (en lieu et place de `usclim`) |
| | `uscfcl` | | Intervention de l'utilisateur pour les conditions aux limites (en lieu et place de `usclim`, même pour les variables qui ne sont pas spécifiques au module compressible) |
| `condli` | | | Traitement des conditions aux limites |
| | `pptycl` | | Branchement des physiques particulières pour le traitement des conditions aux limites |
| | | `cfxtcl` | Traitement des conditions aux limites pour le compressible |
| | | `uscfth` | Calculs de thermodynamique pour le calcul des conditions aux limites |
| | | `cfrusb` | Flux de Rusanov (entrées ou sorties sauf sortie supersonique) |

Table 23.4: Sous-programme `tridim` : partie 2 (pas de temps variable et conditions aux limites)

| | | | |
|---|---|---|---|
| `memcfm` | | | Gestion de la mémoire pour la résolution de l'étape "acoustique" |
| `cfmsvl` | | | Résolution de l'étape "acoustique" |
| | `cfmsfl` | | Calcul du "flux de masse" aux faces (noté $\rho\,\underline{w}\cdot\underline{n}\,S$ dans la documentation du sous-programme `cfmsvl`) |
| | | `cfdivs` | Calcul du terme en divergence du tenseur des contraintes visqueuses (trois appels), éventuellement |
| | | | Après `cfmsfl`, on impose le flux de masse aux faces de bord à partir des conditions aux limites |
| | `cfmsvs` | | Calcul de la "viscosité" aux faces (notée $\Delta\,t\,c^2\frac{S}{d}$ dans la documentation du sous-programme `cfmsvl`) |
| | | | Après `cfmsvs`, on annule la viscosité aux faces de bord pour que le flux de masse soit bien celui souhaité |
| | `codits` | | Résolution du système portant sur la masse volumique |
| | `clpsca` | | Impression des bornes et clipping éventuel (pas de clipping en standard) |
| | `uscfth` | | Gestion éventuelle des bornes par l'utilisateur |
| | `cfbsc3` | | Calcul du flux de masse acoustique aux faces (noté $\underline{\underline{Q}}_{ac}\cdot\underline{n}$ dans la documentation du sous-programme `cfmsvl`) |
| | `uscfth` | | Actualisation de la pression, éventuellement |
| `cfqdmv` | | | Résolution de la quantité de mouvement |
| | `cfcdts` | | Résolution du système |
| | | `cfbsc2` | Calcul des termes de convection et de diffusion au second membre |

Table 23.5: Sous-programme `tridim` : partie 3 (Navier-Stokes)

| | | | |
|---|---|---|---|
| `scalai` | | | Résolution des équations sur les scalaires |
| | `cfener` | | Résolution de l'équation sur l'énergie totale |
| | | `memcfe` | Gestion de la mémoire locale |
| | | `cfdivs` | Calcul du terme en divergence du produit "tenseur des contraintes par vitesse" |
| | | `uscfth` | Calcul de l'écart "énergie interne - $C_v\,T$" ($\varepsilon_{sup}$) |
| | | `cfcdts` | Résolution du système |
| | | `cfbsc2` | Calcul des termes de convection et de diffusion au second membre |
| | | `clpsca` | Impression des bornes et clipping éventuel (pas de clipping en standard) |
| | | `uscfth` | Gestion éventuelle des bornes par l'utilisateur |
| | | `uscfth` | Mise à jour de la pression |

Table 23.6: Sous-programme `tridim` : partie 4 (scalaires)

Le sous-programme `cfbsc3` est similaire à `bilsc2`, mais il produit des flux aux faces et n'est écrit que pour un schéma upwind, à l'ordre 1 en temps (ce qui est cohérent avec les choix faits dans l'algorithme compressible).

Le sous-programme `cfbsc2` est similaire à `bilsc2`, mais n'est écrit que pour un schéma d'ordre 1 en temps. Le sous-programme `cfbsc2` permet d'effectuer un traitement spécifique aux faces de bord pour lesquelles on a appliqué un schéma de Rusanov pour calculer le flux convectif total. Ce sous-programme est appelé pour la résolution de l'équation de la quantité de mouvement et de l'équation de l'énergie. On pourra se reporter à la documentation du sous-programme `cfxtcl`.

Le sous-programme `cfcdts` est similaire à `codits` mais fait appel à `cfbsc2` et non pas à `bilsc2`. Il diffère de `codits` par quelques autres détails qui ne sont pas gênants dans l'immédiat : initialisation de PVARA et de SMBINI, ordre en temps (ordre 2 non pris en compte).

## 1.4 Points à traiter

Des actions complémentaires sont identifiées ci-après, dans l'ordre d'urgence décroissante (on se reportera également à la section "Points à traiter" de la documentation des autres sous-programmes du module compressible).

- Assurer la cohérence des sous-programmes suivants (ou, éventuellement, les fusionner pour éviter qu'ils ne divergent) :
  - `cfcdts` et `codits`,
  - `cfbsc2` et `bilsc2`,
  - `cfbsc3` et `bilsc2`.
- Permettre les suites de calcul incompressible/compressible et compressible/incompressible.
- Apporter un complément de validation (exemple : IPHYDR).
- Assurer la compatibilité avec certaines physiques particulières, selon les besoins. Par exemple : arc électrique, rayonnement, combustion.
- Identifier les causes des difficultés rencontrées sur certains cas académiques, en particulier :
  - canal subsonique (comment s'affranchir des effets indésirables associés aux conditions d'entrée et de sortie, comment réaliser un calcul périodique, en particulier pour la température dont le gradient dans la direction de l'écoulement n'est pas nul, si les parois sont adiabatiques),
  - cavité fermée sans vitesse ni effets de gravité, avec température ou flux thermique imposé en paroi (il pourrait être utile d'extrapoler le gradient de pression au bord : la pression dépend de la température et une simple condition de Neumann homogène est susceptible de créer un terme source de quantité de mouvement parasite),
  - maillage non conforme (non conformité dans la direction transverse d'un canal),
  - "tube à choc" avec terme source d'énergie.
- Compléter certains points de documentation, en particulier les conditions aux limites thermiques pour le couplage avec SYRTHES.
- Améliorer la rapidité à faible nombre de Mach (est-il possible de lever la limite actuelle sur la valeur du pas de temps ?).
- Enrichir, au besoin :
  - les thermodynamiques prises en compte (multiconstituant, gamma variable, Van der Waals...),
  - la gamme des conditions aux limites d'entrée disponibles (condition à débit massique et débit enthalpique imposés par exemple).
- Tester des variantes de l'algorithme :
  - prise en compte des termes sources de l'équation de la quantité de mouvement autres que la gravité dans l'équation de la masse résolue lors de l'étape "acoustique" (les tests réalisés avec cette variante de l'algorithme devront être repris dans la mesure où, dans `cfmsfl`, IIROM et IIROMB n'étaient pas initialisés),
  - implicitation du terme de convection dans l'équation de la masse (éliminer cette possibilité si elle n'apporte rien),
  - étape de prédiction de la pression,

  - non reconstruction de la masse volumique pour le terme convectif (actuellement, les termes convectifs sont traités avec décentrement amont, d'ordre 1 en espace ; pour l'équation de la quantité de mouvement et l'équation de l'énergie, on utilise les valeurs prises au centre des cellules sans reconstruction : c'est l'approche standard de *Code_Saturne*, traduite dans `cfbsc2` ; par contre, dans `cfmsvl`, on reconstruit les valeurs de la masse volumique utilisées pour le terme convectif ; il n'y a pas de raison d'adopter des stratégies différentes, d'autant plus que la reconstruction de la masse volumique ne permet pas de monter en ordre et augmente le risque de dépassement des bornes physiques),
  - montée en ordre en espace (en vérifier l'utilité et la robustesse, en particulier relativement au principe du maximum pour la masse volumique),
  - montée en ordre en temps (en vérifier l'utilité et la robustesse).

- Optimiser l'encombrement mémoire.

# 2- Sous-programme `cfener`

## 2.1 Fonction

Pour les notations et l'algorithme dans son ensemble, on se reportera à `cfbase`.

Après masse (acoustique) et quantité de mouvement, on considère un dernier pas fractionnaire (de $t^{**}$ à $t^{***}$) au cours duquel seule varie l'énergie totale $E = \rho e$.

$$\left\{ \begin{array}{l} \rho^{***} = \rho^{**} = \rho^{n+1} \\ \\ \underline{Q}^{***} = \underline{Q}^{**} = \underline{Q}^{n+1} \\ \\ \dfrac{\partial \rho e}{\partial t} + \mathrm{div}\left( \underline{Q}_{ac} \left( e + \dfrac{P}{\rho} \right) \right) = \rho \underline{f}_v \cdot \underline{u} + \mathrm{div}(\underline{\underline{\Sigma}}^v \underline{u}) - \mathrm{div}\, \underline{\Phi}_s + \rho \Phi_v \end{array} \right. \tag{III.2.1}$$

Pour conserver la positivité de l'énergie, il est indispensable ici, comme pour les scalaires, d'utiliser le flux de masse convectif acoustique $\underline{Q}_{ac}^{n+1}$ compatible avec l'équation de la masse. De plus, pour obtenir des propriétés de positivité sur les scalaires, un schéma upwind pour le terme convectif doit être utilisé (mais les termes sources introduisent des contraintes supplémentaires qui peuvent être prépondérantes et gênantes).

À la fin de cette étape, on actualise éventuellement (mais par défaut non) une deuxième et dernière fois la pression en utilisant la loi d'état pour obtenir la pression finale :

$$P^{n+1} = P(\rho^{n+1}, \varepsilon^{n+1}) \tag{III.2.2}$$

## 2.2 Discrétisation

### 2.2.1 Discrétisation en temps

La modélisation des flux de chaleur choisie jusqu'à présent est de la forme $-\mathrm{div}(\underline{\Phi}_s) = \mathrm{div}(\lambda \underline{\mathrm{grad}}\ T)$.

Pour faire apparaître un terme diffusif stabilisant dans la matrice de résolution, on cherche à exprimer le flux diffusif de chaleur ($-\mathrm{div}(\underline{\Phi}_s)$) en fonction de la variable résolue (l'énergie totale).

Avec $\varepsilon_{sup}(P, \rho)$ dépendant de la loi d'état, on exprime l'énergie totale de la façon suivante :

$$e = \varepsilon + \frac{1}{2}u^2 = (C_v T + \varepsilon_{sup}) + \frac{1}{2}u^2 \tag{III.2.3}$$

En supposant $C_v$ constant[1], on a alors :

$$-\mathrm{div}(\underline{\Phi}_s) = \mathrm{div}(K \underline{\mathrm{grad}}\ (e - \frac{1}{2}u^2 - \varepsilon_{sup})) \qquad \text{avec } K = \lambda / C_v \tag{III.2.4}$$

Lorsqu'un modèle de turbulence est activé, on conserve la même forme de modélisation pour les flux thermiques et $K$ intègre alors la diffusivité turbulente. On pourra se reporter à la documentation de `cfxtcl` à ce sujet.

[1] Pour $C_v$ non constant, les développements restent à faire : on pourra se reporter à P. Mathon, F. Archambeau, J.-M. Hérard : "Implantation d'un algorithme compressible dans *Code_Saturne* ", HI-83/03/016/A

Avec la formulation (III.2.4), on peut donc impliciter le terme en $\underline{\text{grad}}\ e$.

De plus, puisque la vitesse a déjà été résolue, on implicite également le terme en $\underline{\text{grad}}\ \frac{1}{2}u^2$. L'exposant $n+\frac{1}{2}$ de $\varepsilon_{sup}$ indique que l'implicitation de ce terme est partielle (elle dépend de la forme de la loi d'état).

Par ailleurs, on implicite le terme de convection, le terme de puissance des forces volumiques, éventuellement le terme de puissance des forces de pression (suivant la valeur de `IGRDPP`, on utilise la prédiction de pression obtenue après résolution de l'équation portant sur la masse volumique ou bien la pression du pas de temps précédent) et le terme de puissance des forces visqueuses. On implicite le terme de puissance volumique en utilisant $\rho^{n+1}$.

On obtient alors l'équation discrète portant sur $e$ :

$$\begin{aligned}\frac{(\rho e)^{n+1}-(\rho e)^n}{\Delta t^n} + \text{div}(\underline{Q}_{ac}^{n+1}e^{n+1}) - \text{div}(K^n \underline{\text{grad}}\ e^{n+1}) = \rho^{n+1}\underline{f}_v\cdot\underline{u}^{n+1} - \text{div}(\underline{Q}_{ac}^{n+1}\frac{\widetilde{P}}{\rho^{n+1}}) \\ + \text{div}((\underline{\underline{\Sigma}}^v)^{n+1}\underline{u}^{n+1}) - \text{div}(K^n \underline{\text{grad}}\ (\tfrac{1}{2}(u^2)^{n+1}+\varepsilon_{sup}^{n+\frac{1}{2}})) + \rho^{n+1}\Phi_v\end{aligned} \tag{III.2.5}$$

avec $\widetilde{P}=P^{Pred}$ ou $P^n$ suivant la valeur de `IGRDPP` ($P^n$ par défaut).

En pratique, dans *Code_Saturne*, on résout cette équation en faisant apparaître à gauche l'écart $e^{n+1}-e^n$. Pour cela, on écrit la dérivée en temps discrète sous la forme suivante :

$$\begin{aligned}\frac{(\rho e)^{n+1}-(\rho e)^n}{\Delta t^n} &= \frac{\rho^{n+1}e^{n+1}-\rho^n e^n}{\Delta t^n} \\ &= \frac{\rho^n e^{n+1}-\rho^n e^n}{\Delta t^n} + \frac{\rho^{n+1}e^{n+1}-\rho^n e^{n+1}}{\Delta t^n} \\ &= \frac{\rho^n}{\Delta t^n}\left(e^{n+1}-e^n\right) + e^{n+1}\frac{\rho^{n+1}-\rho^n}{\Delta t^n}\end{aligned} \tag{III.2.6}$$

et l'on utilise l'équation de la masse discrète pour écrire :

$$\frac{(\rho e)^{n+1}-(\rho e)^n}{\Delta t^n} = \frac{\rho^n}{\Delta t^n}\left(e^{n+1}-e^n\right) - e^{n+1}\text{div}\,\underline{Q}_{ac}^{n+1} \tag{III.2.7}$$

## 2.2.2 Discrétisation en espace

### Introduction

On intègre l'équation (III.2.5) sur la cellule $i$ de volume $\Omega_i$ et l'on procède comme pour l'équation de la masse et de la quantité de mouvement.

On obtient alors l'équation discrète suivante :

$$\begin{aligned}&\frac{\Omega_i}{\Delta t^n}(\rho_i^{n+1}e_i^{n+1}-\rho_i^n e_i^n) + \sum_{j\in V(i)}\left(e^{n+1}\underline{Q}_{ac}^{n+1}\right)_{ij}\cdot\underline{S}_{ij} - \sum_{j\in V(i)}\left(K^n\underline{\text{grad}}\ (e^{n+1})\right)_{ij}\cdot\underline{S}_{ij} \\ &= \Omega_i\rho_i^{n+1}\underline{f}_{vi}\cdot\underline{u}_i^{n+1} - \sum_{j\in V(i)}\left(\frac{P^{Pred}}{\rho^{n+1}}\,\underline{Q}_{ac}^{n+1}\right)_{ij}\cdot\underline{S}_{ij} + \sum_{j\in V(i)}\left((\underline{\underline{\Sigma}}^v)^{n+1}\underline{u}^{n+1}\right)_{ij}\cdot\underline{S}_{ij} \\ &- \sum_{j\in V(i)}\left(K^n\underline{\text{grad}}\ \left(\frac{1}{2}(u^2)^{n+1}+\varepsilon_{sup}^{n+\frac{1}{2}}\right)\right)_{ij}\cdot\underline{S}_{ij} + \Omega_i\rho_i^{n+1}\Phi_{vi}\end{aligned} \tag{III.2.8}$$

## Discrétisation de la partie "convective"

La valeur à la face s'écrit :

$$\left(e^{n+1}\underline{Q}_{ac}^{n+1}\right)_{ij}\cdot\underline{S}_{ij}=e_{ij}^{n+1}(\underline{Q}_{ac}^{n+1})_{ij}\cdot\underline{S}_{ij} \tag{III.2.9}$$

avec un décentrement sur la valeur de $e^{n+1}$ aux faces :

$$\begin{array}{rcccl} e_{ij}^{n+1} & = & e_i^{n+1} & \text{si} & (\underline{Q}_{ac}^{n+1})_{ij}\cdot\underline{S}_{ij}\geqslant 0 \\ & = & e_j^{n+1} & \text{si} & (\underline{Q}_{ac}^{n+1})_{ij}\cdot\underline{S}_{ij}<0 \end{array} \tag{III.2.10}$$

que l'on peut noter :

$$e_{ij}^{n+1}=\beta_{ij}e_i^{n+1}+(1-\beta_{ij})e_j^{n+1} \tag{III.2.11}$$

avec

$$\left\{\begin{array}{lll} \beta_{ij}=1 & \text{si} & (\underline{Q}_{ac}^{n+1})_{ij}\cdot\underline{S}_{ij}\geqslant 0 \\ \beta_{ij}=0 & \text{si} & (\underline{Q}_{ac}^{n+1})_{ij}\cdot\underline{S}_{ij}<0 \end{array}\right. \tag{III.2.12}$$

## Discrétisation de la partie "diffusive"

La valeur à la face s'écrit :

$$\begin{array}{c} \left(K^n\underline{\text{grad}}\ (e^{n+1})\right)_{ij}\cdot\underline{S}_{ij}=K_{ij}^n\left(\dfrac{\partial e}{\partial n}\right)_{ij}^{n+1}S_{ij} \\ \text{et} \\ \left(K^n\underline{\text{grad}}\ \left(\frac{1}{2}(u^2)^{n+1}+\varepsilon_{sup}^{n+\frac{1}{2}}\right)\right)_{ij}\cdot\underline{S}_{ij}=K_{ij}^n\left(\dfrac{\partial\left(\frac{1}{2}u^2+\varepsilon_{sup}\right)}{\partial n}\right)_{ij}^{n+\frac{1}{2}}S_{ij} \end{array} \tag{III.2.13}$$

avec une interpolation linéaire pour $K^n$ aux faces (et en pratique, $\alpha_{ij}=\frac{1}{2}$) :

$$K_{ij}^n=\alpha_{ij}K_i^n+(1-\alpha_{ij})K_j^n \tag{III.2.14}$$

et un schéma centré avec reconstruction pour le gradient normal aux faces :

$$\left(\frac{\partial e}{\partial n}\right)_{ij}^{n+1}=\frac{e_{J'}^{n+1}-e_{I'}^{n+1}}{\overline{I'J'}}\quad\text{et}\quad\left(\frac{\partial\left(\frac{1}{2}u^2+\varepsilon_{sup}\right)}{\partial n}\right)_{ij}^{n+\frac{1}{2}}=\frac{\left(\frac{1}{2}u^2+\varepsilon_{sup}\right)_{J'}^{n+\frac{1}{2}}-\left(\frac{1}{2}u^2+\varepsilon_{sup}\right)_{I'}^{n+\frac{1}{2}}}{\overline{I'J'}} \tag{III.2.15}$$

## Discrétisation de la puissance des forces de pression

Ce terme est issu du terme convectif, on le discrétise donc de la même façon.

$$\left(\frac{\widetilde{P}}{\rho^{n+1}}\ \underline{Q}_{ac}^{n+1}\right)_{ij}\cdot\underline{S}_{ij}=\left(\frac{\widetilde{P}}{\rho^{n+1}}\right)_{ij}(\underline{Q}_{ac}^{n+1})_{ij}\cdot\underline{S}_{ij} \tag{III.2.16}$$

avec un décentrement sur la valeur de $\dfrac{P}{\rho}$ aux faces :

$$\left(\frac{\widetilde{P}}{\rho^{n+1}}\right)_{ij}=\beta_{ij}\frac{\widetilde{P}_i}{\rho_i^{n+1}}+(1-\beta_{ij})\frac{\widetilde{P}_j}{\rho_j^{n+1}}\quad\text{avec}\quad\left\{\begin{array}{lll} \beta_{ij}=1 & \text{si} & (\underline{Q}_{ac}^{n+1})_{ij}\cdot\underline{S}_{ij}\geqslant 0 \\ \beta_{ij}=0 & \text{si} & (\underline{Q}_{ac}^{n+1})_{ij}\cdot\underline{S}_{ij}<0 \end{array}\right. \tag{III.2.17}$$

### Discrétisation de la puissance des forces visqueuses

On calcule les termes dans les cellules puis on utilise une interpolation linéaire (on utilise $\alpha_{ij} = \frac{1}{2}$ dans la relation ci-dessous) :

$$\left(\left(\underline{\underline{\Sigma}}^{v}\right)^{n+1}\underline{u}^{n+1}\right)_{ij} \cdot \underline{S}_{ij} = \left\{\alpha_{ij}\left(\left(\underline{\underline{\Sigma}}^{v}\right)^{n+1}\underline{u}^{n+1}\right)_{i} + (1-\alpha_{ij})\left(\left(\underline{\underline{\Sigma}}^{v}\right)^{n+1}\underline{u}^{n+1}\right)_{j}\right\} \cdot \underline{S}_{ij} \qquad \text{(III.2.18)}$$

### Remarques

Les termes "convectifs" associés à $\text{div}\left(\left(e^{n+1} + \frac{\widetilde{P}}{\rho^{n+1}}\right)\underline{Q}_{ac}^{n+1}\right)$ sont calculés avec un décentrement amont (consistant, d'ordre 1 en espace). Les valeurs utilisées sont bien prises au centre de la cellule amont ($e_i$, $P_i$, $\rho_i$) et non pas au projeté $I'$ du centre de la cellule sur la normale à la face passant par son centre de gravité (sur un cas test en triangles, l'utilisation de $P'_I$ et de $\rho'_I$ pour le terme de transport de pression a conduit à un résultat insatisfaisant, mais des corrections ont été apportées aux sources depuis et il serait utile de vérifier que cette conclusion n'est pas remise en question).

Les termes diffusifs associés à $\text{div}\left(K\,\underline{\text{grad}}\,\left(e + \frac{1}{2}u^2 + \varepsilon_{sup}\right)\right)$ sont calculés en utilisant des valeurs aux faces reconstruites pour s'assurer de la consistance du schéma.

## 2.3 Mise en œuvre

Après une étape de gestion de la mémoire (`memcfe`), on calcule les différents termes sources (au centre des cellules) :

- source volumique de chaleur (`ustssc`),
- source associée aux sources de masse (`catsma`),
- source associée à l'accumulation de masse $\text{div}\,\underline{Q}_{ac}$ (directement dans `cfener`),
- dissipation visqueuse (`cfdivs`),
- transport de pression (directement dans `cfener`),
- puissance de la pesanteur (directement dans `cfener`),
- termes diffusifs en $\text{div}\left(K\,\underline{\text{grad}}\,\left(\frac{1}{2}u^2 + \varepsilon_{sup}\right)\right)$ (calcul de $\varepsilon_{sup}$ par `uscfth`, puis calcul du terme diffusif directement dans `cfener`).

Le système (III.2.8) est résolu par une méthode d'incrément et résidu en utilisant une méthode de Jacobi (`cfcdts`).

L'impression des bornes et la limitation éventuelle de l'énergie sont ensuite effectuées par `clpsca` suivi de `uscfth` (intervention utilisateur optionnelle).

On actualise enfin la pression et on calcule la température (`uscfth`).

Pour terminer, en parallèle ou en périodique, on échange les variables pression, énergie et température.

## 2.4 Points à traiter

• **Choix de $\widetilde{P}$**
En standard, on utilise $\widetilde{P} = P^n$, mais ce n'est pas le seul choix possible. On pourrait étudier le comportement de l'algorithme avec $P^{Pred}$ et $P^{n+1}$ (avec $P^{n+1}$, en particulier, $\dfrac{\widetilde{P}}{\rho^{n+1}}$ est évalué avec la masse volumique et l'énergie prises au même instant).

• **Terme source dans l'équation de l'énergie**
La présence d'un terme source externe dans l'équation de l'énergie génère des oscillations de vitesse qu'il est important d'analyser et de comprendre.

# 3- Sous-programme cfmsvl

## 3.1 Fonction

Pour les notations et l'algorithme dans son ensemble, on se reportera à `cfbase`.

On considère un premier pas fractionnaire au cours duquel l'énergie totale est fixe. Seules varient la masse volumique et le flux de masse acoustique normal aux faces (défini et calculé aux faces).

On a donc le système suivant, entre $t^n$ et $t^*$ :

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \mathrm{div}\underline{Q}_{ac} = 0 \\ \dfrac{\partial \underline{Q}_{ac}}{\partial t} + \underline{\mathrm{grad}}\ P = \rho \underline{f} \\ \underline{Q}^* = \underline{Q}^n \\ e^* = e^n \end{cases} \tag{III.3.1}$$

Une partie des termes sources de l'équation de la quantité de mouvement peut être prise en compte dans cette étape (les termes les plus importants, en prêtant attention aux sous-équilibres).

Il faut noter que si $\underline{f}$ est effectivement nul, on aura bien un système "acoustique", mais que si l'on place des termes supplémentaires dans $\underline{f}$, la dénomination est abusive (on la conservera cependant).

On obtient $\rho^* = \rho^{n+1}$ en résolvant (III.3.1), et l'on actualise alors le flux de masse acoustique $\underline{Q}_{ac}^{n+1}$, qui servira pour la convection (en particulier pour la convection de l'enthalpie totale et de tous les scalaires transportés).

Suivant la valeur de `IGRDPP`, on actualise éventuellement la pression, en utilisant la loi d'état :

$$P^{Pred} = P(\rho^{n+1}, \varepsilon^n)$$

## 3.2 Discrétisation

### 3.2.1 Discrétisation en temps

Le système (III.3.1) discrétisé en temps donne :

$$\begin{cases} \dfrac{\rho^{n+1} - \rho^n}{\Delta t^n} + \mathrm{div}\underline{Q}_{ac}^{n+1} = 0 \\ \dfrac{\underline{Q}_{ac}^{n+1} - \underline{Q}^n}{\Delta t^n} + \underline{\mathrm{grad}}\ P^* = \rho^n \underline{f}^n \\ Q^* = Q^n \\ e^* = e^n \end{cases} \tag{III.3.2}$$

$$\begin{array}{lrcl} \text{avec} & \underline{f}^n & = & \underline{0} \\ \text{ou} & \underline{f}^n & = & \underline{g} \\ \text{ou même} & \underline{f}^n & = & \underline{f}_v + \dfrac{1}{\rho^n}\left(-\mathrm{div}(\underline{u}\otimes\underline{Q}) + \underline{\mathrm{div}}(\underline{\underline{\Sigma}}^v) + \underline{j}\wedge\underline{B}\right)^n \end{array} \tag{III.3.3}$$

Dans la pratique nous avons décidé de prendre $\underline{f}^n = \underline{g}$ :

- le terme $\underline{j}\wedge\underline{B}$ n'a pas été testé,
- le terme $\underline{\mathrm{div}}(\underline{\underline{\Sigma}}^v)$ était négligeable sur les tests réalisés,
- le terme $\mathrm{div}(\underline{u}\otimes\underline{Q})$ a paru déstabiliser les calculs (mais au moins une partie des tests a été réalisée avec une erreur de programmation et il faudrait donc les reprendre).

Le terme $\underline{Q}^n$ dans la 2ème équation de (III.3.2) est le vecteur "quantité de mouvement" qui provient de l'étape de résolution de la quantité de mouvement du pas de temps précédent, $\underline{Q}^n = \rho^n\underline{u}^n$. On pourrait théoriquement utiliser un vecteur quantité de mouvement issu de l'étape acoustique du pas de temps précédent, mais il ne constitue qu'un "prédicteur" plus ou moins satisfaisant (il n'a pas "vu" les termes sources qui ne sont pas dans $\underline{f}^n$) et cette solution n'a pas été testée.

On écrit alors la pression sous la forme :

$$\underline{\mathrm{grad}}\ P = c^2\,\underline{\mathrm{grad}}\ \rho + \beta\,\underline{\mathrm{grad}}\ s \tag{III.3.4}$$

avec $c^2 = \left.\dfrac{\partial P}{\partial\rho}\right|_s$ et $\beta = \left.\dfrac{\partial P}{\partial s}\right|_\rho$ tabulés ou analytiques à partir de la loi d'état.

On discrétise l'expression précédente en :

$$\underline{\mathrm{grad}}\ P^* = (c^2)^n\underline{\mathrm{grad}}\ (\rho^{n+1}) + \beta^n\underline{\mathrm{grad}}\ (s^n) \tag{III.3.5}$$

On obtient alors une équation portant sur $\rho^{n+1}$ en substituant l'expression de $\underline{Q}_{ac}^{n+1}$ issue de la 2ème équation de (III.3.2) dans la 1ère équation de (III.3.2) :

$$\frac{\rho^{n+1}-\rho^n}{\Delta t^n} + \mathrm{div}(\underline{w}^n\rho^n) - \mathrm{div}\left(\Delta t^n(c^2)^n\underline{\mathrm{grad}}\ (\rho^{n+1})\right) = 0 \tag{III.3.6}$$

où :

$$\underline{w}^n = \underline{u}^n + \Delta t^n\left(\underline{f}^n - \frac{\beta^n}{\rho^n}\underline{\mathrm{grad}}\ (s^n)\right) \tag{III.3.7}$$

Formulation alternative (programmée mais non testée) avec le terme de convection implicite :

$$\frac{\rho^{n+1}-\rho^n}{\Delta t^n} + \mathrm{div}(\underline{w}^n\rho^{n+1}) - \mathrm{div}\left(\Delta t^n(c^2)^n\underline{\mathrm{grad}}\ (\rho^{n+1})\right) = 0 \tag{III.3.8}$$

## 3.2.2 Discrétisation en espace

### Introduction

On intègre l'équation précédente ( (III.3.6) ou (III.3.8) ) sur la cellule $i$ de volume $\Omega_i$. On transforme les intégrales de volume en intégrales surfaciques et l'on discrétise ces intégrales. Pour simplifier l'exposé, on se place sur une cellule $i$ dont aucune face n'est sur le bord du domaine.

On obtient alors l'équation discrète suivante[1] :

$$\Omega_i \frac{\rho_i^{n+1}-\rho_i^n}{\Delta t^n} + \sum_{j\in Vois(i)} (\rho^{n+\frac{1}{2}}\underline{w}^n)_{ij}\cdot\underline{S}_{ij} - \sum_{j\in Vois(i)} \left(\Delta t^n (c^2)^n \underline{\text{grad}}\ (\rho^{n+1})\right)_{ij}\cdot\underline{S}_{ij} = 0 \qquad \text{(III.3.9)}$$

## Discrétisation de la partie "convective"

La valeur à la face s'écrit :

$$(\rho^{n+\frac{1}{2}}\underline{w}^n)_{ij}\cdot\underline{S}_{ij} = \rho_{ij}^{n+\frac{1}{2}}\underline{w}_{ij}^n\cdot\underline{S}_{ij} \qquad \text{(III.3.10)}$$

avec, pour $\underline{w}_{ij}^n$, une simple interpolation linéaire :

$$\underline{w}_{ij}^n = \alpha_{ij}\underline{w}_i^n + (1-\alpha_{ij})\underline{w}_j^n \qquad \text{(III.3.11)}$$

et un décentrement sur la valeur de $\rho^{n+\frac{1}{2}}$ aux faces :

$$\begin{array}{rcll} \rho_{ij}^{n+\frac{1}{2}} & = & \rho_{I'}^{n+\frac{1}{2}} & \text{si} \quad \underline{w}_{ij}^n\cdot\underline{S}_{ij} \geqslant 0 \\ & = & \rho_{J'}^{n+\frac{1}{2}} & \text{si} \quad \underline{w}_{ij}^n\cdot\underline{S}_{ij} < 0 \end{array} \qquad \text{(III.3.12)}$$

que l'on peut noter :

$$\rho_{ij}^{n+\frac{1}{2}} = \beta_{ij}\rho_{I'}^{n+\frac{1}{2}} + (1-\beta_{ij})\rho_{J'}^{n+\frac{1}{2}} \qquad \text{(III.3.13)}$$

avec

$$\left\{ \begin{array}{ll} \beta_{ij} = 1 & \text{si} \quad \underline{w}_{ij}^n\cdot\underline{S}_{ij} \geqslant 0 \\ \beta_{ij} = 0 & \text{si} \quad \underline{w}_{ij}^n\cdot\underline{S}_{ij} < 0 \end{array} \right. \qquad \text{(III.3.14)}$$

## Discrétisation de la partie "diffusive"

La valeur à la face s'écrit :

$$\left(\Delta t^n (c^2)^n \underline{\text{grad}}\ (\rho^{n+1})\right)_{ij}\cdot\underline{S}_{ij} = \Delta t^n (c^2)_{ij}^n \left(\frac{\partial\rho}{\partial n}\right)_{ij}^{n+1} S_{ij} \qquad \text{(III.3.15)}$$

avec, pour assurer la continuité du flux normal à l'interface, une interpolation harmonique de $(c^2)^n$ :

$$(c^2)_{ij}^n = \frac{(c^2)_i^n (c^2)_j^n}{\alpha_{ij}(c^2)_i^n + (1-\alpha_{ij})(c^2)_j^n} \qquad \text{(III.3.16)}$$

et un schéma centré pour le gradient normal aux faces :

$$\left(\frac{\partial\rho}{\partial n}\right)_{ij}^{n+1} = \frac{\rho_{J'}^{n+1}-\rho_{I'}^{n+1}}{\overline{I'J'}} \qquad \text{(III.3.17)}$$

## Système final

On obtient maintenant le système final, portant sur $(\rho_i^{n+1})_{i=1\ldots N}$ :

$$\frac{\Omega_i}{\Delta t^n}(\rho_i^{n+1}-\rho_i^n) + \sum_{j\in Vois(i)} \rho_{ij}^{n+\frac{1}{2}}\underline{w}_{ij}^n\cdot\underline{S}_{ij} - \sum_{j\in Vois(i)} \Delta t^n (c^2)_{ij}^n \frac{\rho_{J'}^{n+1}-\rho_{I'}^{n+1}}{\overline{I'J'}}\ S_{ij} = 0 \qquad \text{(III.3.18)}$$

[1] L'exposant $^{n+\frac{1}{2}}$ signifie que le terme peut être implicite ou explicite. En pratique on a choisi $\rho^{n+\frac{1}{2}} = \rho^n$.

### Remarque : interpolation aux faces pour le terme de diffusion

Le choix de la forme de la moyenne pour le cofacteur du flux normal n'est pas sans conséquence sur la vitesse de convergence, surtout lorsque l'on est en présence de fortes inhomogénéités.

On utilise une interpolation harmonique pour $c^2$ afin de conserver la continuité du flux diffusif normal $\Delta t(c^2)\dfrac{\partial \rho}{\partial n}$ à l'interface $ij$. En effet, on suppose que le flux est dérivable à l'interface. Il doit donc y être continu.
Écrivons la continuité du flux normal à l'interface, avec la discrétisation suivante[2] :

$$\left(\Delta t(c^2)\frac{\partial \rho}{\partial n}\right)_{ij} = \Delta t(c^2)_i \frac{\rho_{ij}-\rho_{I'}}{\overline{I'F}} = \Delta t(c^2)_j \frac{\rho_{J'}-\rho_{ij}}{\overline{FJ'}} \tag{III.3.19}$$

En égalant les flux à gauche et à droite de l'interface, on obtient

$$\rho_{ij} = \frac{\overline{I'F}\,(c^2)_j \rho_{J'} + \overline{FJ'}\,(c^2)_i \rho_{I'}}{\overline{I'F}\,(c^2)_j + \overline{FJ'}\,(c^2)_i} \tag{III.3.20}$$

On introduit cette formulation dans la définition du flux (par exemple, du flux à gauche) :

$$\left(\Delta t(c^2)\frac{\partial \rho}{\partial n}\right)_{ij} = \Delta t(c^2)_i \frac{\rho_{ij}-\rho_{I'}}{\overline{I'F}} \tag{III.3.21}$$

et on utilise la définition de $(c^2)_{ij}$ en fonction de ce même flux

$$\left(\Delta t(c^2)\frac{\partial \rho}{\partial n}\right)_{ij} \overset{\text{déf}}{=} \Delta t(c^2)_{ij} \frac{\rho_{J'}-\rho_{I'}}{\overline{I'J'}} \tag{III.3.22}$$

pour obtenir la valeur de $(c^2)_{ij}$ correspondant à l'équation (III.3.16) :

$$(c^2)_{ij} = \frac{\overline{I'J'}\,(c^2)_i (c^2)_j}{\overline{FJ'}\,(c^2)_i + \overline{I'F}\,(c^2)_j} \tag{III.3.23}$$

## 3.3 Mise en œuvre

Le système (III.3.18) est résolu par une méthode d'incrément et résidu en utilisant une méthode de Jacobi pour inverser le système si le terme convectif est implicite et en utilisant une méthode de gradient conjugué si le terme convectif est explicite (qui est le cas par défaut).

Attention, les valeurs du flux de masse $\rho\,\underline{w}\cdot\underline{S}$ et de la viscosité $\Delta t\,c^2\frac{S}{d}$ aux faces de bord, qui sont calculées dans `cfmsfl` et `cfmsvs` respectivement, sont modifiées immédiatement après l'appel à ces sous-programmes. En effet, il est indispensable que la contribution de bord de $\left(\rho\,\underline{w} - \Delta t\,(c^2)\,\underline{\text{grad}}\ \rho\right)\cdot\underline{S}$ représente exactement $\underline{Q}_{ac}\cdot\underline{S}$. Pour cela,

- immédiatement après l'appel à `cfmsfl`, on remplace la contribution de bord de $\rho\,\underline{w}\cdot\underline{S}$ par le flux de masse exact, $\underline{Q}_{ac}\cdot\underline{S}$, déterminé à partir des conditions aux limites,
- puis, immédiatement après l'appel à `cfmsvs`, on annule la viscosité au bord $\Delta t\,(c^2)$ pour éliminer la contribution de $-\Delta t\,(c^2)\,(\underline{\text{grad}}\ \rho)\cdot\underline{S}$ (l'annulation de la viscosité n'est pas problématique pour la matrice, puisqu'elle porte sur des incréments).

Une fois qu'on a obtenu $\rho^{n+1}$, on peut actualiser le flux de masse acoustique aux faces $(\underline{Q}_{ac}^{n+1})_{ij}\cdot\underline{S}_{ij}$, qui servira pour la convection des autres variables :

$$(\underline{Q}_{ac}^{n+1})_{ij}\cdot\underline{S}_{ij} = -\left(\Delta t^n (c^2)^n \underline{\text{grad}}\ (\rho^{n+1})\right)_{ij}\cdot\underline{S}_{ij} + \left(\rho^{n+\frac{1}{2}}\underline{w}^n\right)_{ij}\cdot\underline{S}_{ij} \tag{III.3.24}$$

[2] On ne reconstruit pas les valeurs de $\Delta t\,c^2$ aux points $I'$ et $J'$.

Ce calcul de flux est réalisé par `cfbsc3`. Si l'on a choisi l'algorithme standard, équation (III.3.6), on complète le flux dans `cfmsvl` immédiatement après l'appel à `cfbsc3`. En effet, dans ce cas, la convection est explicite ($\rho^{n+\frac{1}{2}} = \rho^n$, obtenu en imposant `ICONV(ISCA(IRHO(IPHAS)))=0`) et le sous-programme `cfbsc3`, qui calcule le flux de masse aux faces, ne prend pas en compte la contribution du terme $\rho^{n+\frac{1}{2}}\,\underline{w}^n \cdot \underline{S}$. On ajoute donc cette contribution dans `cfmsvl`, après l'appel à `cfbsc3`. Au bord, en particulier, c'est bien le flux de masse calculé à partir des conditions aux limites que l'on obtient.

On actualise la pression à la fin de l'étape, en utilisant la loi d'état :

$$P_i^{pred} = P(\rho_i^{n+1}, \varepsilon_i^n) \tag{III.3.25}$$

## 3.4 Points à traiter

Le calcul du flux de masse au bord n'est pas entièrement satisfaisant si la convection est traitée de manière implicite (algorithme non standard, non testé, associé à l'équation (III.3.8), correspondant au choix $\rho^{n+\frac{1}{2}} = \rho^{n+1}$ et obtenu en imposant `ICONV(ISCA(IRHO(IPHAS)))=1`). En effet, après `cfmsfl`, il faut déterminer la vitesse de convection $\underline{w}^n$ pour qu'apparaisse $\rho^{n+1}\underline{w}^n \cdot \underline{n}$ au cours de la résolution par `codits`. De ce fait, on doit déduire une valeur de $\underline{w}^n$ à partir de la valeur du flux de masse. Au bord, en particulier, il faut donc diviser le flux de masse issu des conditions aux limites par la valeur de bord de $\rho^{n+1}$. Or, lorsque des conditions de Neumann sont appliquées à la masse volumique, la valeur de $\rho^{n+1}$ au bord n'est pas connue avant la résolution du système. On utilise donc, au lieu de la valeur de bord inconnue de $\rho^{n+1}$ la valeur de bord prise au pas de temps précédent $\rho^n$. Cette approximation est susceptible d'affecter la valeur du flux de masse au bord.

# 4- Sous-programme `cfqdmv`

## 4.1 Fonction

Pour les notations et l'algorithme dans son ensemble, on se reportera à `cfbase`.

Dans le premier pas fractionnaire (`cfmsvl`), on a résolu une équation sur la masse volumique, obtenu une prédiction de la pression et un flux convectif "acoustique". On considère ici un second pas fractionnaire au cours duquel seul varie le vecteur flux de masse $\underline{Q} = \rho\underline{u}$ (seule varie la vitesse au centre des cellules). On résout l'équation de Navier-Stokes indépendamment pour chaque direction d'espace, et l'on utilise le flux de masse acoustique calculé précédemment comme flux convecteur (on pourrait aussi utiliser le vecteur quantité de mouvement du pas de temps précédent). De plus, on résout en variable $\underline{u}$ et non $\underline{Q}$.

Le système à résoudre entre $t^*$ et $t^{**}$ est (on exclut la turbulence, dont le traitement n'a rien de particulier dans le module compressible) :

$$\left\{\begin{array}{l} \rho^{**} = \rho^* = \rho^{n+1} \\ \\ \dfrac{\partial \rho\underline{u}}{\partial t} + \underline{\mathrm{div}}(\underline{u} \otimes \underline{Q}_{ac}) + \underline{\mathrm{grad}}\ P = \rho\underline{f}_v + \underline{\mathrm{div}}(\underline{\underline{\Sigma}}^v) \\ \\ e^{**} = e^* = e^n \end{array}\right. \tag{III.4.1}$$

La résolution de cette étape est similaire à l'étape de prédiction des vitesses du schéma de base de *Code_Saturne*.

## 4.2 Discrétisation

### 4.2.1 Discrétisation en temps

On implicite le terme de convection, éventuellement le gradient de pression (suivant la valeur de `IGRDPP`, en utilisant la pression prédite lors de l'étape acoustique) et le terme en gradient du tenseur des contraintes visqueuses. On explicite les autres termes du tenseur des contraintes visqueuses. On implicite les forces volumiques en utilisant $\rho^{n+1}$.

On obtient alors l'équation discrète suivante :

$$\begin{array}{l} \dfrac{(\rho\underline{u})^{n+1} - (\rho\underline{u})^n}{\Delta t^n} + \underline{\mathrm{div}}(\underline{u}^{n+1} \otimes \underline{Q}_{ac}^{n+1}) - \underline{\mathrm{div}}(\mu^n \underline{\underline{\mathrm{grad}}}\ \underline{u}^{n+1}) \\ \\ \quad = \rho^{n+1}\underline{f}_v - \underline{\mathrm{grad}}\ \widetilde{P} + \underline{\mathrm{div}}\left(\mu^n\ {}^t\underline{\underline{\mathrm{grad}}}\ \underline{u}^n + (\kappa^n - \tfrac{2}{3}\mu^n)\mathrm{div}\underline{u}^n\ \underline{\underline{Id}}\right) \end{array} \tag{III.4.2}$$

avec $\widetilde{P} = P^n$ ou $P^{Pred}$ suivant la valeur de `IGRDPP` ($P^n$ par défaut).

En pratique, dans *Code_Saturne*, on résout cette équation en faisant apparaître à gauche l'écart $\underline{u}^{n+1} - \underline{u}^n$. Pour cela, on écrit la dérivée en temps discrète sous la forme suivante :

$$\begin{aligned}\frac{(\rho\underline{u})^{n+1}-(\rho\underline{u})^{n}}{\Delta t^n} &= \frac{\rho^{n+1}\,\underline{u}^{n+1}-\rho^{n}\,\underline{u}^{n}}{\Delta t^n}\\ &= \frac{\rho^{n}\,\underline{u}^{n+1}-\rho^{n}\,\underline{u}^{n}}{\Delta t^n}+\frac{\rho^{n+1}\,\underline{u}^{n+1}-\rho^{n}\,\underline{u}^{n+1}}{\Delta t^n}\\ &= \frac{\rho^{n}}{\Delta t^n}\left(\underline{u}^{n+1}-\underline{u}^{n}\right)+\underline{u}^{n+1}\frac{\rho^{n+1}-\rho^{n}}{\Delta t^n}\end{aligned} \tag{III.4.3}$$

et l'on utilise alors l'équation de la masse discrète pour écrire :

$$\frac{(\rho\underline{u})^{n+1}-(\rho\underline{u})^{n}}{\Delta t^n} = \frac{\rho^{n}}{\Delta t^n}\left(\underline{u}^{n+1}-\underline{u}^{n}\right)-\underline{u}^{n+1}\mathrm{div}\,\underline{Q}_{ac}^{n+1} \tag{III.4.4}$$

## 4.2.2 Discrétisation en espace

### Introduction

On intègre l'équation (III.4.2) sur la cellule $i$ de volume $\Omega_i$ et on obtient l'équation discrétisée en espace :

$$\begin{aligned}&\frac{\Omega_i}{\Delta t^n}(\rho_i^{n+1}\underline{u}_i^{n+1}-\rho_i^{n}\underline{u}_i^{n})+\sum_{j\in V(i)}(\underline{u}^{n+1}\otimes\underline{Q}_{ac}^{n+1})_{ij}\cdot\underline{S}_{ij}-\sum_{j\in V(i)}\left(\mu^n\underline{\underline{\mathrm{grad}}}\;\underline{u}^{n+1}\right)_{ij}\cdot\underline{S}_{ij}\\ &=\Omega_i\rho_i^{n+1}\underline{f}_{vi}-\Omega_i(\underline{\mathrm{grad}}\;\widetilde{P})_i+\sum_{j\in V(i)}\left(\mu^n\;{}^t\underline{\underline{\mathrm{grad}}}\;\underline{u}^n+(\kappa^n-\frac{2}{3}\mu^n)\mathrm{div}\underline{u}^n\;\underline{\underline{Id}}\right)_{ij}\underline{S}_{ij}\end{aligned} \tag{III.4.5}$$

### Discrétisation de la partie "convective"

La valeur à la face s'écrit :

$$(\underline{u}^{n+1}\otimes\underline{Q}_{ac}^{n+1})_{ij}\cdot\underline{S}_{ij}=\underline{u}_{ij}^{n+1}(\underline{Q}_{ac}^{n+1})_{ij}\cdot\underline{S}_{ij} \tag{III.4.6}$$

avec un décentrement sur la valeur de $\underline{u}^{n+1}$ aux faces :

$$\begin{array}{rcll}\underline{u}_{ij}^{n+1} &=& \underline{u}_{i}^{n+1} & \text{si}\quad (\underline{Q}_{ac}^{n+1})_{ij}\cdot\underline{S}_{ij}\geqslant 0\\ &=& \underline{u}_{j}^{n+1} & \text{si}\quad (\underline{Q}_{ac}^{n+1})_{ij}\cdot\underline{S}_{ij}<0\end{array} \tag{III.4.7}$$

que l'on peut noter :

$$\underline{u}_{ij}^{n+1}=\beta_{ij}\underline{u}_i^{n+1}+(1-\beta_{ij})\underline{u}_j^{n+1} \tag{III.4.8}$$

avec

$$\left\{\begin{array}{ll}\beta_{ij}=1 & \text{si}\quad (\underline{Q}_{ac}^{n+1})_{ij}\cdot\underline{S}_{ij}\geqslant 0\\ \beta_{ij}=0 & \text{si}\quad (\underline{Q}_{ac}^{n+1})_{ij}\cdot\underline{S}_{ij}<0\end{array}\right. \tag{III.4.9}$$

### Discrétisation de la partie "diffusive"

La valeur à la face s'écrit :

$$\left(\mu^n\underline{\underline{\mathrm{grad}}}\;\underline{u}^{n+1}\right)_{ij}\underline{S}_{ij}=\mu_{ij}^n\left(\frac{\partial\underline{u}}{\partial n}\right)_{ij}^{n+1}S_{ij} \tag{III.4.10}$$

avec une interpolation linéaire pour $\mu^n$ aux faces (en pratique avec $\alpha_{ij}=\frac{1}{2}$) :

$$\mu_{ij}^n=\alpha_{ij}\mu_i^n+(1-\alpha_{ij})\mu_j^n \tag{III.4.11}$$

et un schéma centré pour le gradient normal aux faces :

$$\left(\frac{\partial \underline{u}}{\partial n}\right)_{ij}^{n+1} = \frac{\underline{u}_{J'}^{n+1} - \underline{u}_{I'}^{n+1}}{\overline{I'J'}} \tag{III.4.12}$$

### Discrétisation du gradient de pression

On utilise `grdcel` standard. Suivant la valeur de `IMRGRA`, cela correspond à une reconstruction itérative ou par moindres carrés.

### Discrétisation du "reste" du tenseur des contraintes visqueuses

On calcule des gradients aux cellules et on utilise une interpolation linéaire aux faces (avec, en pratique, $\alpha_{ij} = \frac{1}{2}$) :

$$\begin{aligned}\left(\mu^n\ {}^t\underline{\underline{\mathrm{grad}}}\ \underline{u}^n + (\kappa^n - \tfrac{2}{3}\mu^n)\mathrm{div}\underline{u}^n\ \underline{\underline{Id}}\right)_{ij} \cdot \underline{S}_{ij} = \Big\{ &\alpha_{ij}\left(\mu^n\ {}^t\underline{\underline{\mathrm{grad}}}\ \underline{u}^n + (\kappa^n - \tfrac{2}{3}\mu^n)\mathrm{div}\underline{u}^n\ \underline{\underline{Id}}\right)_i \\ &+(1-\alpha_{ij})\left(\mu^n\ {}^t\underline{\underline{\mathrm{grad}}}\ \underline{u}^n + (\kappa^n - \tfrac{2}{3}\mu^n)\mathrm{div}\underline{u}^n\ \underline{\underline{Id}}\right)_j \Big\} \cdot \underline{S}_{ij}\end{aligned} \tag{III.4.13}$$

## 4.3 Mise en œuvre

On résout les trois directions d'espace du système (III.4.5) successivement et indépendamment :

$$\left\{\begin{array}{l} \dfrac{\Omega_i}{\Delta t^n}(\rho_i^{n+1} u_{i(\alpha)}^{n+1} - \rho_i^n u_{i(\alpha)}^n) + \displaystyle\sum_{j\in V(i)} u_{ij(\alpha)}^{n+1} (\underline{\underline{Q}}_{ac}^{n+1})_{ij} \cdot \underline{S}_{ij} - \sum_{j\in V(i)} \mu_{ij}^n \frac{u_{j(\alpha)}^{n+1} - u_{i(\alpha)}^{n+1}}{\overline{I'J'}} S_{ij} \\ \qquad = \Omega_i \rho_i^{n+1} f_{vi(\alpha)} - \Omega_i (\underline{\mathrm{grad}}\ \widetilde{P})_{i(\alpha)} \\ \qquad + \displaystyle\sum_{j\in V(i)} \left((\mu^n\ {}^t\underline{\underline{\mathrm{grad}}}\ \underline{u}^n)_{ij} \cdot \underline{S}_{ij}\right)_{(\alpha)} + \sum_{j\in V(i)} \left((\kappa^n - \frac{2}{3}\mu^n)\mathrm{div}\underline{u}^n\right)_{ij} S_{ij(\alpha)} \\ i = 1 \ldots N \qquad \text{et} \qquad (\alpha) = x,\ y,\ z \end{array}\right. \tag{III.4.14}$$

Chaque système associé à une direction est résolu par une méthode d'incrément et résidu en utilisant une méthode de Jacobi.

# 5- Sous-programme `cfxtcl`

## 5.1 Fonction

Pour le traitement des conditions aux limites, on considère le système (III.5.1)

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \mathrm{div}(\underline{Q}) = 0 \\ \dfrac{\partial \underline{Q}}{\partial t} + \underline{\mathrm{div}}(\underline{u} \otimes \underline{Q}) + \underline{\mathrm{grad}}\ P = \rho \underline{f}_v + \underline{\mathrm{div}}(\underline{\underline{\Sigma}}^v) \\ \dfrac{\partial E}{\partial t} + \mathrm{div}(\underline{u}(E+P)) = \rho \underline{f}_v \cdot \underline{u} + \mathrm{div}(\underline{\underline{\Sigma}}^v \underline{u}) - \mathrm{div}\,\underline{\Phi}_s + \rho \Phi_v \end{cases} \tag{III.5.1}$$

en tant que système hyperbolique portant sur la variable vectorielle $\underline{W} = {}^t(\rho, \underline{Q}, E)$.

Le système s'écrit alors :

$$\frac{\partial \underline{W}}{\partial t} + \sum_{i=1}^{3} \frac{\partial}{\partial x_i} \underline{F}_i(\underline{W}) = \sum_{i=1}^{3} \frac{\partial}{\partial x_i} \underline{F}_i^D(\underline{W}, \nabla \underline{W}) + \underline{\mathcal{S}} \tag{III.5.2}$$

où les $\underline{F}_i(\underline{W})$ sont les vecteurs flux convectifs et les $\underline{F}_i^D(\underline{W})$ sont les vecteurs flux diffusifs dans les trois directions d'espace, et $\underline{\mathcal{S}}$ est un terme source.

La démarche classique de *Code_Saturne* est adoptée : on impose les conditions aux limites en déterminant, pour chaque variable, des valeurs numériques de bord. Ces valeurs sont calculées de telle façon que, lorsqu'on les utilise dans les formules standard donnant les flux discrets, on obtienne les contributions souhaitées au bord.

Pour rendre compte des flux convectifs (aux entrées et aux sorties en particulier), on fait abstraction des flux diffusifs et des termes sources pour résoudre un problème de Riemann qui fournit un vecteur d'état au bord. Celui-ci permet de calculer un flux, soit directement (par les formules discrètes standard), soit en appliquant un schéma de Rusanov (schéma de flux décentré).

En paroi, on résout également, dans certains cas, un problème de Riemann pour déterminer une pression au bord.

## 5.2 Discrétisation

### 5.2.1 Introduction

#### Objectif

On résume ici les différentes conditions aux limites utilisées pour l'algorithme compressible afin de fournir une vue d'ensemble. Pour atteindre cet objectif, il est nécessaire de faire référence à des éléments relatifs à la discrétisation et au mode d'implantation des conditions aux limites.

Lors de l'implantation, on a cherché à préserver la cohérence avec l'approche utilisée dans le cadre standard de l'algorithme incompressible de *Code_Saturne*. Il est donc conseillé d'avoir pris connaissance du mode de traitement des conditions aux limites incompressibles avant d'aborder les détails de l'algorithme compressible.

Comme pour l'algorithme incompressible, les conditions aux limites sont imposées par le biais d'une valeur de bord associée à chaque variable. De plus, pour certaines frontières (parois à température imposée ou à flux thermique imposé), on dispose de deux valeurs de bord pour la même variable, l'une d'elles étant dédiée au calcul du flux diffusif. Enfin, sur certains types d'entrée et de sortie, on définit également une valeur du flux convectif au bord.

Comme pour l'algorithme incompressible, l'utilisateur peut définir, pour chaque face de bord, des conditions aux limites pour chaque variable, mais on conseille cependant d'utiliser uniquement les types prédéfinis décrits ci-après (entrée, sortie, paroi, symétrie) qui ont l'avantage d'assurer la cohérence entre les différentes variables et les différentes étapes de calcul.

## Parois

**Pression** : on doit disposer d'une condition pour le calcul du gradient qui intervient dans l'étape de quantité de mouvement. On dispose de deux types de condition, au choix de l'utilisateur :

- par défaut, la pression imposée au bord est proportionnelle à la valeur interne (la pression au bord est obtenue comme solution d'un problème de Riemann sur les équations d'Euler avec un état miroir ; on distingue les cas de choc et de détente et, dans le cas d'une détente trop forte, une condition de Dirichlet homogène est utilisée pour éviter de voir apparaître une pression négative),
- si l'utilisateur le souhaite (`ICFGRP(IPHAS)=1`), le gradient de pression est imposé à partir du profil de pression hydrostatique.

**Vitesse et turbulence** : traitement standard (voir la documentation des sous-programmes `condli` et `clptur`).

**Scalaires passifs** : traitement standard (flux nul par défaut imposé dans `typecl`).

**Masse volumique** : traitement standard des scalaires (flux nul par défaut imposé dans `typecl`).

**Énergie et température**[1] : traitement standard des scalaires (flux nul par défaut imposé dans `typecl`), hormis pour le calcul du flux diffusif dans le cas de parois à température imposée ou à flux thermique imposé.

**Flux diffusif pour l'énergie en paroi** : l'utilisateur peut choisir (dans `uscfcl`) entre une température de paroi imposée et un flux thermique diffusif (ou "conductif") imposé. S'il ne précise rien, on considère que la paroi est adiabatique (flux thermique diffusif imposé et de valeur nulle). Dans tous les cas, il faut donc disposer d'un moyen d'imposer le flux diffusif souhaité. Pour cela, on détermine une valeur de bord pour l'énergie qui, introduite dans la formule donnant le flux discret, permettra d'obtenir la contribution attendue (voir le paragraphe 5.2.5). Conformément à l'approche classique de *Code_Saturne*, cette valeur est stockée sous la forme d'un couple de coefficients (de type `COEFAF`, `COEFBF`). Il est important de souligner que cette valeur de bord ne doit être utilisée que pour le calcul du flux diffusif : dans les autres situations pour lesquelles une valeur de bord de l'énergie ou de la température est requise (calcul de gradient par exemple), on utilise une condition de flux nul (traitement standard des scalaires). Pour cela, on dispose d'une seconde valeur de bord qui est stockée au moyen d'un couple de coefficients (`COEFA`, `COEFB`) distinct du précédent.

**Flux convectifs** : le flux de masse dans la direction normale à la paroi est pris nul. De ce fait, les flux convectifs seront nuls quelle que soit les valeurs de bord imposées pour les différentes variables transportées.

## Symétrie

Les conditions appliquées sont les conditions classiques de l'algorithme incompressible (vitesse normale nulle, flux nul pour les autres variables).

[1] Le gradient de température est *a priori* inutile, mais peut être requis par l'utilisateur.

Elles sont imposées dans le sous-programme `typecl` essentiellement. Pour la pression, la condition de flux nul est imposée dans `cfxtcl` (au début des développements, on appliquait le même traitement qu'en paroi, mais une condition de flux nul a été préférée afin de s'affranchir des problèmes potentiels dans les configurations 2D).

### Entrées et sorties

On obtient, par résolution d'un problème de Riemann au bord, complété par des relations de thermodynamique (`uscfth`), des valeurs de bord pour toutes les variables (on suppose qu'en entrée, toutes les composantes de la vitesse sont fournies ; elles sont supposées nulles par défaut, hormis pour les entrées à $(\rho, \underline{u})$ imposés, `IERUCF`, pour lesquelles il faut fournir la vitesse explicitement).

Ces valeurs de bord sont utilisées de deux façons :

- elles sont utilisées pour calculer les flux convectifs, en faisant appel au schéma de Rusanov (sauf en sortie supersonique) ; ces flux sont directement intégrés au second membre des équations à résoudre.
- elles servent de valeur de Dirichlet dans toutes les autres configurations pour lesquelles une valeur de bord est requise (calcul de flux diffusif, calcul de gradient...)

Deux cas particuliers :

- aux entrées ou sorties pour lesquelles toutes les variables sont imposées (`IESICF`), on utilise une condition de Neumann homogène pour la pression (hormis pour le calcul du gradient intervenant dans l'équation de la quantité de mouvement, qui est pris en compte par le flux convectif déterminé par le schéma de Rusanov). Ce choix est arbitraire (on n'a pas testé le comportement de l'algorithme si l'on conserve une condition de Dirichlet sur la pression), mais a été fait en supposant qu'une condition de Neumann homogène serait *a priori* moins déstabilisante, dans la mesure où, pour ce type de frontière, l'utilisateur peut imposer une valeur de pression très différente de celle régnant à l'intérieur du domaine (la valeur imposée est utilisée pour le flux convectif).
- pour les grandeurs turbulentes et les scalaires utilisateur, si le flux de masse est entrant et que l'on a fourni une valeur de Dirichlet (`RCODCL(*,*,1)` dans `uscfcl`), on l'utilise, pour le calcul du flux convectif et du flux diffusif ; sinon, on utilise une condition de Neumann homogène (le concept de sortie de type 9 ou 10 est couvert par cette approche).

## 5.2.2 Problème de Riemann au bord

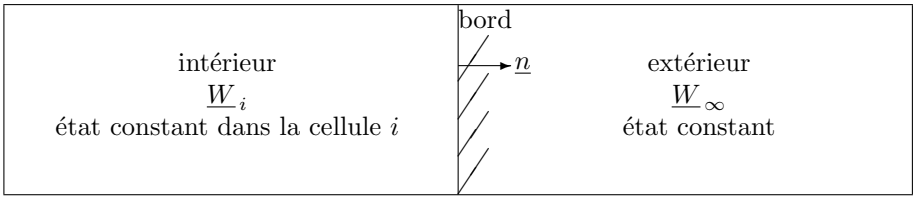
### Introduction

On cherche à obtenir un état au bord, pour les entrées, les sorties et les parois.

Pour cela, on fait abstraction des flux diffusifs et des sources. Le système résultant est alors appelé système d'équations d'Euler. On se place de plus dans un repère orienté suivant la normale au bord considéré $(\underline{\tau}_1, \underline{\tau}_2, \underline{n})$ et l'on ne considère que les variations suivant cette normale. Le système devient donc :

$$\frac{\partial \underline{W}}{\partial t} + \frac{\partial}{\partial n}\underline{F}_n(\underline{W}) = 0 \quad \text{avec} \quad \underline{F}_n(\underline{W}) = \sum_{i=1}^{3} n_i \underline{F}_i(\underline{W}) \quad \text{et} \quad \frac{\partial}{\partial n} = \sum_{i=1}^{3} n_i \frac{\partial}{\partial x_i} \tag{III.5.3}$$

Pour déterminer les valeurs des variables au bord, on recherche l'évolution du problème instationnaire suivant, appelé problème de Riemann :

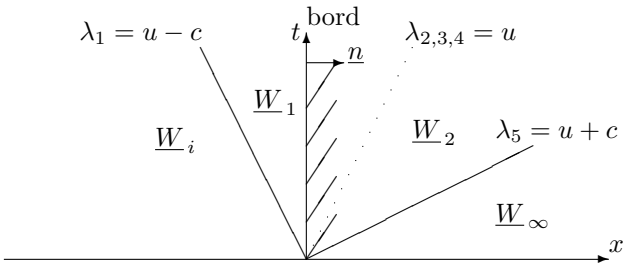



avec $\underline{W}_\infty$ dépendant du type de bord et différent de $\underline{W}_i$ *a priori*.

Pour résoudre ce problème de Riemann, on utilisera les variables non-conservatives $\widetilde{\underline{W}} = {}^t(\rho, \underline{u}, P)$ et l'on retrouvera l'énergie grâce à l'équation d'état.

Pour alléger l'écriture, dans le présent paragraphe 5.2.2, on notera aussi $\underline{W}$ le vecteur ${}^t(\rho, \underline{u}, P)$ et $\underline{u} = \underline{u}_\tau + u\,\underline{n}$ (en posant $u = \underline{u} \cdot \underline{n}$ et $\underline{u}_\tau = \underline{u} - (\underline{u} \cdot \underline{n})\underline{n}$).

La solution est une suite d'états constants, dont les valeurs dépendent de $\underline{W}_i$ et $\underline{W}_\infty$, séparés par des ondes se déplaçant à des vitesses données par les valeurs propres du système $(\lambda_i)_{i=1\ldots5}$. On représente les caractéristiques du système sur le schéma suivant :




Comme valeurs des variables au bord, on prendra les valeurs correspondant à l'état constant qui contient le bord ($\underline{W}_1$ dans l'exemple précédent).

Il faut remarquer que la solution du problème de Riemann dépend de la thermodynamique et devra donc être calculée et codée par l'utilisateur si la thermodynamique n'a pas été prévue (en version 1.2, la seule thermodynamique prévue est celle des gaz parfaits).

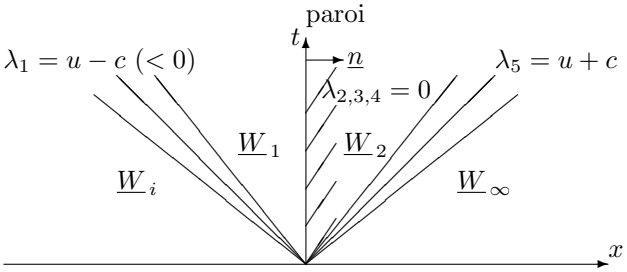
## En paroi, pour la condition de pression (sans effet de gravité)

Pour les faces de paroi, on définit à l'extérieur du domaine un état miroir $\underline{W}_\infty$ par :

$$\underline{W}_i = \begin{pmatrix} \rho_i \\ \underline{u}_{\tau i} \\ u_i \\ P_i \end{pmatrix} \qquad \underline{W}_\infty = \begin{pmatrix} \rho_\infty & = & \rho_i \\ \underline{u}_{\tau\infty} & = & \underline{u}_{\tau i} \\ u_\infty & = & -u_i \\ P_\infty & = & P_i \end{pmatrix} \tag{III.5.4}$$

Les solutions dépendent de l'orientation de la vitesse dans la cellule de bord :

1. Si $u_i \leqslant 0$, la solution est une double détente symétrique.

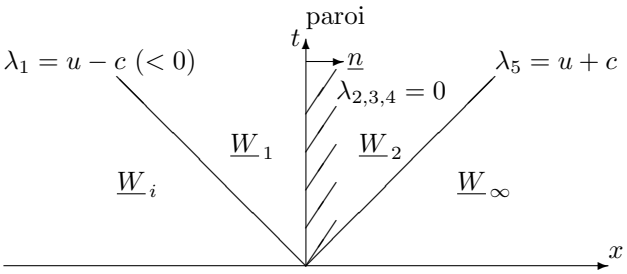



$$\underline{W}_{paroi} = \underline{W}_1 = \underline{W}_2 \quad \begin{cases} \rho_p = \rho_1 = \rho_2 \\ u_p = u_1 = u_2 \\ P_p = P_1 = P_2 \end{cases}$$

La conservation de la vitesse tangentielle à travers la 1-onde donne $\underline{u}_{\tau p} = \underline{u}_{\tau i}$. Par des considérations de symétrie on trouve $u_p = 0$. Puis on obtient $\rho_p$ et $P_p$ en écrivant la conservation des invariants de Riemann à travers la 1-détente :

$$\begin{cases} u_1 + \displaystyle\int_0^{\rho_1} \frac{c}{\rho} d\rho = u_i + \int_0^{\rho_i} \frac{c}{\rho} d\rho \\ \\ s_1 = s_i \end{cases} \Rightarrow \begin{cases} \displaystyle\int_{\rho_i}^{\rho_1} \frac{c}{\rho} d\rho = u_i & \Rightarrow \rho_p = \rho_1 \\ \\ s(P_1, \rho_1) = s(P_i, \rho_i) & \Rightarrow P_p = P_1 \end{cases} \tag{III.5.5}$$

2. Si $u_i > 0$, la solution est un double choc symétrique.




$$\underline{W}_{paroi} = \underline{W}_1 = \underline{W}_2 \quad \begin{cases} \rho_p = \rho_1 = \rho_2 \\ u_p = u_1 = u_2 \\ P_p = P_1 = P_2 \end{cases}$$

De même que précédemment, on trouve $\underline{u}_{\tau p} = \underline{u}_{\tau i}$ et $u_p = 0$, puis $\rho_p$ et $P_p$ en écrivant les relations de saut à travers le 1-choc :

$$\begin{cases} \rho_1 \rho_i (u_1 - u_i)^2 = (P_1 - P_i)(\rho_1 - \rho_i) \\ \\ 2\rho_1 \rho_i (\varepsilon_1 - \varepsilon_i) = (P_1 + P_i)(\rho_1 - \rho_i) \end{cases} \text{ avec } \varepsilon = \varepsilon(P, \rho) \Rightarrow \begin{cases} \rho_p = \rho_1 \\ \\ P_p = P_1 \end{cases} \tag{III.5.6}$$

Pour les gaz parfaits, avec $M_i = \dfrac{\underline{u}_i \cdot \underline{n}}{c_i}$ (Nombre de Mach de paroi), on a :

- Cas détente ($M_i \leqslant 0$) :

$$\begin{cases} P_p = 0 & \text{si} \quad 1 + \dfrac{\gamma - 1}{2} M_i < 0 \\ \\ P_p = P_i \left( 1 + \dfrac{\gamma - 1}{2} M_i \right)^{\frac{2\gamma}{\gamma - 1}} & \text{sinon} \end{cases}$$

$$\rho_p = \rho_i \left( \frac{P_p}{P_i} \right)^{\frac{1}{\gamma}}$$

- Cas choc ($M_i > 0$) :

$$P_p = P_i \left( 1 + \frac{\gamma(\gamma+1)}{4} M_i^2 + \gamma M_i \sqrt{1 + \frac{(\gamma+1)^2}{16} M_i^2} \right)$$

$$\rho_p = \rho_i \left( \frac{P_p - P_i}{P_p - P_i - \rho_i (\underline{u}_i \cdot \underline{n})^2} \right)$$

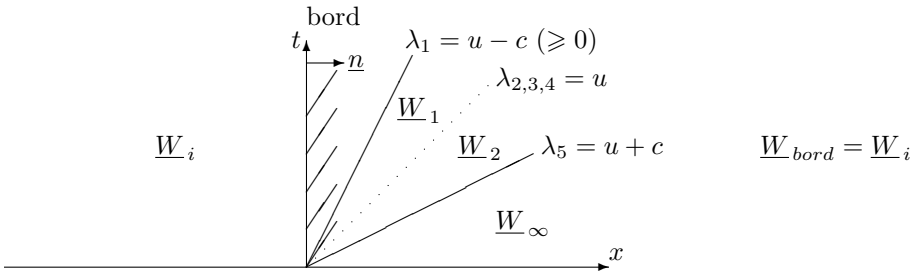
En pratique, le flux convectif normal à la paroi est nul et seule la condition de pression déterminée ci-dessus est effectivement utilisée (pour le calcul du gradient sans effet de gravité).

## En sortie

Il existe deux cas de traitement des conditions en sortie, selon le nombre de Mach normal à la face de bord ($c_i$ est la vitesse du son dans la cellule de bord) :

$$M_i = \frac{u_i}{c_i} = \frac{\underline{u}_i \cdot \underline{n}}{c_i}$$

**Sortie supersonique (condition ISSPCF de `uscfcl`) :** $M_i \geqslant 1 \Rightarrow u_i - c_i \geqslant 0$

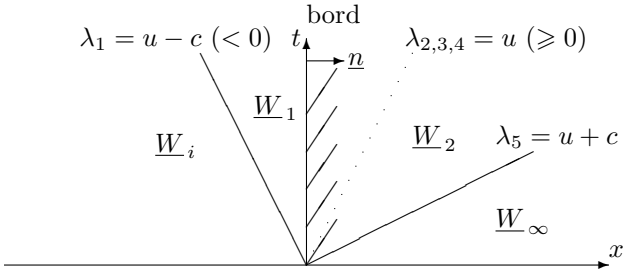



$$\underline{W}_{bord} = \underline{W}_i$$

Toutes les caractéristiques sont sortantes, on connaît donc toutes les conditions au bord :

$$\begin{cases} \rho_b = \rho_i \\ \underline{u}_{\tau b} = \underline{u}_{\tau i} \\ u_b = u_i \\ P_b = P_i \end{cases} \tag{III.5.7}$$

**Sortie subsonique (condition ISOPCF de `uscfcl`) :** $0 \leqslant M_i < 1 \Rightarrow (u_i \geqslant 0 \text{ et } u_i - c_i < 0)$




$$\underline{W}_{bord} = \underline{W}_1 \qquad \begin{cases} \rho_b = \rho_1 \\ u_b = u_1 \\ P_b = P_1 \end{cases}$$

On a une caractéristique entrante, on doit donc imposer une seule condition au bord (en général la pression de sortie $P_{ext}$).

On connaît alors $P_b = P_{ext}$ et $\underline{u}_{\tau b} = \underline{u}_{\tau i}$ (par conservation de la vitesse tangentielle à travers la 1-onde). Pour trouver les inconnues manquantes ($\rho_b$ et $u_b$) on doit résoudre le passage de la 1-onde :

1. Si $P_{ext} \leqslant P_i$, on a une 1-détente.

   On écrit la conservation des invariants de Riemann à travers la 1-détente :

$$\left\{ \begin{array}{l} s_1 = s_i \\ \\ u_1 + \displaystyle\int_0^{\rho_1} \frac{c}{\rho} d\rho = u_i + \int_0^{\rho_i} \frac{c}{\rho} d\rho \end{array} \right. \Rightarrow \left\{ \begin{array}{ll} s(P_{ext}, \rho_1) = s(P_i, \rho_i) & \Rightarrow \rho_b = \rho_1 \\ \\ u_1 = u_i - \displaystyle\int_{\rho_i}^{\rho_1} \frac{c}{\rho} d\rho & \Rightarrow u_b = u_1 \end{array} \right. \tag{III.5.8}$$

2. Si $P_{ext} > P_i$, on a un 1-choc.

   On écrit les relations de saut à travers le 1-choc :

$$\left\{ \begin{array}{l} \rho_1 \rho_i (u_1 - u_i)^2 = (P_{ext} - P_i)(\rho_1 - \rho_i) \\ \\ 2\rho_1 \rho_i (\varepsilon(P_{ext}, \rho_1) - \varepsilon(P_i, \rho_i)) = (P_{ext} + P_i)(\rho_1 - \rho_i) \end{array} \right. \Rightarrow \left\{ \begin{array}{l} \rho_b = \rho_1 \\ \\ u_b = u_1 \end{array} \right. \tag{III.5.9}$$

   Pour les gaz parfaits, on a :

   - Cas détente ($P_{ext} \leqslant P_i$) :

$$P_b = P_{ext}$$

$$\rho_b = \rho_i \left( \frac{P_{ext}}{P_i} \right)^{\frac{1}{\gamma}}$$

   - Cas choc ($P_{ext} > P_i$) :

$$P_b = P_{ext}$$

$$\rho_b = \rho_i \left( \frac{P_{ext} - P_i}{P_{ext} - P_i - \rho_i (\underline{u}_i \cdot \underline{n} - \underline{u}_b \cdot \underline{n})^2} \right) = \rho_i \left( \frac{(\gamma+1) P_{ext} + (\gamma-1) P_i}{(\gamma-1) P_{ext} + (\gamma+1) P_i} \right)$$

La valeur de la masse volumique au bord intervient en particulier dans le flux de masse.

## En entrée

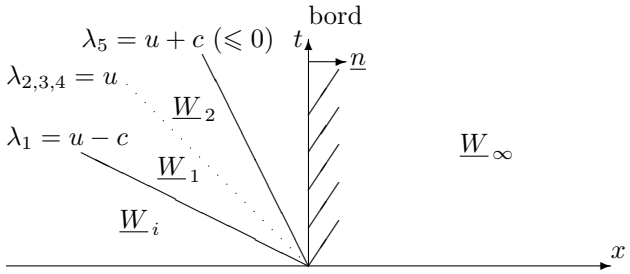
L'utilisateur impose les valeurs qu'il souhaite pour les variables en entrée :

$$\underline{W}_{ext} = \left( \begin{array}{c} \rho_{ext} \\ \underline{u}_{\tau ext} \\ u_{ext} \\ P_{ext} \end{array} \right)$$

De même que précédemment, il existe deux cas de traitement des conditions en entrée, pilotés par le nombre de Mach entrant, normalement à la face de bord (avec $c_{ext}$ la vitesse du son en entrée) :

$$M_{ext} = \frac{u_{ext}}{c_{ext}} = \frac{\underline{u}_{ext} \cdot \underline{n}}{c_{ext}}$$

**Entrée supersonique (condition IESICF de `uscfcl`) :** $M_{ext} \leqslant -1 \Rightarrow u_{ext} + c_{ext} \leqslant 0$



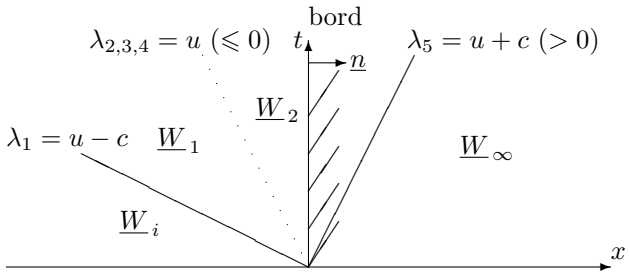

$\underline{W}_\infty$ $\qquad$ $\underline{W}_{bord} = \underline{W}_\infty = \underline{W}_{ext}$

Toutes les caractéristiques sont entrantes, toutes les conditions au bord sont donc imposées par l'utilisateur.

$$\begin{cases} \rho_b = \rho_{ext} \\ \underline{u}_{\tau b} = \underline{u}_{\tau ext} \\ u_b = u_{ext} \\ P_b = P_{ext} \end{cases} \tag{III.5.10}$$

**Entrée subsonique (condition IERUCF de `uscfcl`) :**

$$-1 < M_{ext} \leqslant 0 \Rightarrow (u_{ext} \leqslant 0 \text{ et } u_{ext} + c_{ext} > 0)$$




$\underline{W}_\infty$ $\qquad$ $\underline{W}_{bord} = \underline{W}_2$ $\quad \begin{cases} \rho_b = \rho_2 \\ u_b = u_2 \\ P_b = P_2 \end{cases}$

On a une caractéristique sortante. L'utilisateur doit donc laisser un degré de liberté.

En général, on impose le flux de masse en entrée, donc $\rho_{ext}$ et $u_{ext}$, et l'on calcule la pression au bord en résolvant le passage des 1∼4-ondes. On connaît aussi $\underline{u}_{\tau b} = \underline{u}_{\tau ext}$, par conservation de la vitesse tangentielle à travers la 5-onde.

1. Si $u_{ext} \geqslant u_i$, on a une 1-détente.

   On écrit la conservation des invariants de Riemann à travers la 1-détente et la conservation de la vitesse et de la pression à travers le contact :

$$\begin{cases} u_1 + \displaystyle\int_0^{\rho_1} \frac{c}{\rho} d\rho = u_i + \int_0^{\rho_i} \frac{c}{\rho} d\rho \\ s_1 = s_i \end{cases} \Rightarrow \begin{cases} \displaystyle\int_{\rho_i}^{\rho_1} \frac{c}{\rho} d\rho = u_i - u_{ext} & \Rightarrow \rho_1 \\ s(P_2, \rho_1) = s(P_i, \rho_i) & \Rightarrow P_b = P_2 \end{cases} \tag{III.5.11}$$

$$\begin{cases} u_1 = u_2 = u_{ext} \\ P_1 = P_2 \end{cases}$$

2. Si $u_{ext} < u_i$, on a un 1-choc.

   On écrit les relations de saut à travers le 1-choc et la conservation de la vitesse et de la pression à travers le contact :

$$\left\{\begin{array}{l} \rho_1\rho_i(u_1-u_i)^2=(P_1-P_i)(\rho_1-\rho_i) \\ 2\rho_1\rho_i(\varepsilon_1-\varepsilon_i)=(P_1+P_i)(\rho_1-\rho_i) \\ \varepsilon=\varepsilon(P,\rho) \end{array}\right. \Rightarrow \left\{\begin{array}{l} \rho_1 \\ P_b=P_2 \end{array}\right. \qquad \text{(III.5.12)}$$

$$\left\{\begin{array}{l} u_1=u_2=u_{ext} \\ P_1=P_2 \end{array}\right.$$

   Pour les gaz parfaits, on a :

   - Cas détente ($\delta M \leqslant 0$) :

$$\left\{\begin{array}{ll} P_b=0 & \text{si} \quad 1+\dfrac{\gamma-1}{2}\delta M<0 \\ P_b=P_i\left(1+\dfrac{\gamma-1}{2}\delta M\right)^{\frac{2\gamma}{\gamma-1}} & \text{sinon} \end{array}\right.$$

$$\rho_b=\rho_{ext}$$

   - Cas choc ($\delta M > 0$) :

$$P_b=P_i\left(1+\frac{\gamma(\gamma+1)}{4}\delta M^2+\gamma\delta M\sqrt{1+\frac{(\gamma+1)^2}{16}\delta M^2}\right)$$

$$\rho_b=\rho_{ext}$$

### 5.2.3 Condition de pression en paroi avec effets de gravité

Le problème de Riemann considéré précédemment ne prend pas en compte les effets de la gravité. Or, dans certains cas, si l'on ne prend pas en compte le gradient de pression "hydrostatique", on peut obtenir une solution erronée (en particulier, par exemple, on peut créer une source de quantité de mouvement non physique dans un milieu initialement au repos).

Écrivons l'équilibre local dans la maille de bord :

$$\underline{\text{grad}}\ P=\rho\underline{g} \qquad \text{(III.5.13)}$$

Pour simplifier la résolution, on peut utiliser la formulation de (III.5.13) en incompressible (c'est cette approche qui a été adoptée dans *Code_Saturne*) :

$$(\underline{\text{grad}}\ P)_i=\rho_i\underline{g} \quad \text{ce qui donne} \quad P_{paroi}=P_i+\rho_i\underline{g}\cdot(\underline{x}_{paroi}-\underline{x}_i) \qquad \text{(III.5.14)}$$

Une autre approche (dépendante de l'équation d'état) consiste à résoudre l'équilibre local avec la formulation compressible (III.5.13), en supposant de plus que la maille est isentropique :

$$\left\{\begin{array}{l} \underline{\text{grad}}\ P=\rho\underline{g} \\ P=P(\rho,s_i) \end{array}\right. \qquad \text{(III.5.15)}$$

Ce qui donne, pour un gaz parfait :

$$P_{paroi}=P_i\left(1+\frac{\gamma-1}{\gamma}\frac{\rho_i}{P_i}\underline{g}\cdot(\underline{x}_{paroi}-\underline{x}_i)\right)^{\frac{\gamma}{\gamma-1}} \qquad \text{(III.5.16)}$$

**Remarque :** la formule issue de l'incompressible (III.5.14) est une linéarisation de la formule (III.5.16). Dans les cas courants elle s'éloigne très peu de la formule exacte. Dans des conditions extrêmes, si l'on considère par exemple de l'air à $1000K$ et $10bar$, avec une accélération de la pesanteur $g = 1000m/s^2$ et une différence de hauteur entre le centre de la cellule et le centre de la face de bord de $10m$, l'expression (III.5.16) donne $P_{paroi} = 1034640,4Pa$ et l'expression (III.5.14) donne $P_{paroi} = 1034644,7Pa$, soit une différence relative de moins de $0,001\%$. On voit aussi que la différence entre la pression calculée au centre de la cellule et celle calculée au bord est de l'ordre de $3\%$.

### 5.2.4 Schéma de Rusanov pour le calcul de flux convectifs au bord

#### Introduction

Le schéma de Rusanov est utilisé pour certains types de conditions aux limites afin de passer du vecteur d'état calculé au bord comme indiqué précédemment (solution du problème de Riemann) à un flux convectif de bord (pour la masse, la quantité de mouvement et l'énergie). L'utilisation de ce schéma (décentré amont) permet de gagner en stabilité.

Le schéma de Rusanov est appliqué aux frontières auxquelles on considère qu'il est le plus probable de rencontrer des conditions en accord imparfait avec l'état régnant dans le domaine, conditions qui sont donc susceptibles de déstabiliser le calcul : il s'agit des entrées et des sorties (frontières de type IESICF, ISOPCF, IERUCF, IEQHCF). En sortie supersonique (ISSPCF) cependant, le schéma de Rusanov est inutile et n'est donc pas appliqué : en effet, pour ce type de frontière, l'état imposé au bord est exactement l'état amont et le décentrement du schéma de Rusanov n'apporterait donc rien.

#### Principe

Pour le calcul du flux décentré de Rusanov, on considère le système hyperbolique constitué des seuls termes convectifs issus des équations de masse, quantité de mouvement et énergie. Ce système est écrit, par changement de variable, en non conservatif (on utilise la relation $P = \dfrac{\rho\varepsilon}{\gamma - 1}$ et on note $u_\xi$ les composantes de $\underline{u}$) :

$$\left\{\begin{array}{lcll} \dfrac{\partial \rho}{\partial t} & + & \rho \underline{\mathrm{div}}\, \underline{u} + \underline{u}\, \underline{\mathrm{grad}}\ \rho & = 0 \\ \dfrac{\partial u_\xi}{\partial t} & + & \underline{u}\, \underline{\mathrm{grad}}\ u_\xi + \dfrac{1}{\rho}\dfrac{\partial P}{\partial \xi} & = 0 \\ \dfrac{\partial P}{\partial t} & + & \gamma\, P\, \mathrm{div} \underline{u} + \underline{u}\, \underline{\mathrm{grad}}\ P & = 0 \end{array}\right. \tag{III.5.17}$$

En notant le vecteur d'état $\underline{W} = (\rho, \underline{u}, P)^t$, ce système est noté :

$$\frac{\partial \underline{W}}{\partial t} + \mathrm{div}\, \underline{F}(\underline{W}) = 0 \tag{III.5.18}$$

Avec $\delta \underline{W}$ l'incrément temporel du vecteur d'état, $\underline{n}$ la normale à une face, $ij$ la face interne partagée par les cellules $i$ et $j$ et $ik$ la face de bord $k$ associée à la cellule $i$, la discrétisation spatiale conduit à :

$$\frac{|\Omega_i|}{\Delta t}\delta \underline{W}_i + \sum_{j \in Vois(i)} \int_{S_{ij}} \underline{F}(\underline{W})\, \underline{n}\, dS + \sum_{k \in \gamma_b(i)} \int_{S_{b_{ik}}} \underline{F}(\underline{W})\, \underline{n}\, dS = 0 \tag{III.5.19}$$

Sur une face de bord donnée, on applique le schéma de Rusanov pour calculer le flux comme suit :

$$\frac{1}{|S_{b_{ik}}|} \int_{S_{b_{ik}}} \underline{F}(\underline{W})\, \underline{n}\, dS = \frac{1}{2}\left(\underline{F}(\underline{W}_i) + \underline{F}(\underline{W}_{b_{ik}})\right) \cdot \underline{n}_{b_{ik}} - \frac{1}{2}\rho_{rus\, b_{ik}} \left(\underline{W}_{b_{ik}} - \underline{W}_i\right) = \underline{F}_{rus\, b_{ik}}(\underline{W}) \tag{III.5.20}$$

Dans cette relation, $\underline{W}_{b_{ik}}$ est le vecteur d'état $\underline{W}_\infty$, connu au bord (tel qu'il résulte de la résolution du problème de Riemann au bord présentée plus haut pour chaque type de frontière considéré).

## Paramètre de décentrement $\rho_{rus\,b_{ik}}$

Pour chaque face de bord, le scalaire $\rho_{rus\,b_{ik}}$ est la plus grande valeur du rayon spectral de la matrice jacobienne $\dfrac{\partial\,\underline{F}_n(\underline{W})}{\partial\underline{W}}$ obtenu pour les vecteurs d'état $\underline{W}_i$ et $\underline{W}_{b_{ik}}$.

$\underline{F}_n$ est la composante du flux $\underline{F}$ dans la direction de la normale à la face de bord, $\underline{n}_{b_{ik}}$. Utiliser $\underline{F}_n$ pour la détermination du paramètre de décentrement $\rho_{rus\,b_{ik}}$ relève d'une approche classique qui consiste à remplacer le système tridimensionnel initial par le système unidimensionnel projeté dans la direction normale à la face, en négligeant les variations du vecteur d'état $\underline{W}$ dans la direction tangeante à la face :

$$\frac{\partial\underline{W}}{\partial t}+\frac{\partial\,\underline{F}_n(\underline{W})}{\partial\underline{W}}\,\frac{\partial\underline{W}}{\partial n}=0 \tag{III.5.21}$$

De manière plus explicite, si l'on se place dans un repère de calcul ayant $\underline{n}_{b_{ik}}$ comme vecteur de base, et si l'on note $u$ la composante de vitesse associée, le système est le suivant (les équations portant sur les composantes transverses de la vitesse sont découplées, associées à la valeur propre $u$, comme le serait un scalaire simplement convecté et ne sont pas écrites ci-après) :

$$\left\{\begin{array}{lcl}
\dfrac{\partial\rho}{\partial t}+\rho\dfrac{\partial\,u}{\partial\,n}+u\,\dfrac{\partial\,\rho}{\partial\,n} & = & 0\\
\dfrac{\partial u}{\partial t}+u\,\dfrac{\partial\,u}{\partial\,n}+\dfrac{1}{\rho}\,\dfrac{\partial P}{\partial n} & = & 0\\
\dfrac{\partial P}{\partial t}+\gamma\,P\,\dfrac{\partial\,u}{\partial\,n}+u\,\dfrac{\partial\,P}{\partial\,n} & = & 0
\end{array}\right. \tag{III.5.22}$$

La matrice jacobienne associée est donc :

$$\begin{pmatrix} u & \rho & 0\\ 0 & u & \dfrac{1}{\rho}\\ 0 & \gamma\,P & 0\end{pmatrix} \tag{III.5.23}$$

Les valeurs propres sont $u$ et $u\pm c$ (avec $c=\sqrt{\frac{\gamma\,P}{\rho}}$). Le rayon spectral est donc $|u|+c$ et le paramètre de décentrement s'en déduit :

$$\rho_{rus\,b_{ik}}=max\left(|u_i|+c_i,|u_{b_{ik}}|+c_{b_{ik}}\right) \tag{III.5.24}$$

## Expression des flux convectifs

Les flux convectifs calculés par le schéma de Rusanov pour les variables masse, quantité de mouvement et énergie représentent donc la discrétisation des termes suivants :

$$\left\{\begin{array}{l}
\mathrm{div}(\underline{Q})\\
\underline{\mathrm{div}}(\underline{u}\otimes\underline{Q})+\underline{\mathrm{grad}}\;P\\
\mathrm{div}\left(\underline{Q}\,(e+\dfrac{P}{\rho})\right)
\end{array}\right. \tag{III.5.25}$$

Pour une face de bord $ik$ adjacente à la cellule $i$ et avec la valeur précédente de $\rho_{rus\,b_{ik}}$, on a :

$$\left\{\begin{array}{lcl}
\displaystyle\int_{S_{b_{ik}}} \underline{Q}\cdot\underline{n}\,dS & = & \displaystyle\frac{1}{2}\left((\underline{Q}_i+\underline{Q}_{b_{ik}})\cdot\underline{n}_{b_{ik}}\right)S_{b_{ik}} \\
 & & \displaystyle-\frac{1}{2}\rho_{rus\,b_{ik}}\,(\rho_{b_{ik}}-\rho_i)\,S_{b_{ik}} \\
\displaystyle\int_{S_{b_{ik}}} (\underline{u}\otimes\underline{Q}+\underline{\mathrm{grad}}\ P)\cdot\underline{n}\,dS & = & \displaystyle\frac{1}{2}\left(\underline{u}_i(\underline{Q}_i\cdot\underline{n}_{b_{ik}})+P_i\,\underline{n}_{b_{ik}}+\underline{u}_{b_{ik}}(\underline{Q}_{b_{ik}}\cdot\underline{n}_{b_{ik}})+P_{b_{ik}}\,\underline{n}_{b_{ik}}\right)S_{b_{ik}} \\
 & & \displaystyle-\frac{1}{2}\rho_{rus\,b_{ik}}\left(\underline{Q}_{b_{ik}}-\underline{Q}_i\right)S_{b_{ik}} \\
\displaystyle\int_{S_{b_{ik}}} (e+\frac{P}{\rho})\,\underline{Q}\cdot\underline{n}\,dS & = & \displaystyle\frac{1}{2}\left((e_i+\frac{P_i}{\rho_i})\,(\underline{Q}_i\cdot\underline{n}_{b_{ik}})+(e_{b_{ik}}+\frac{P_{b_{ik}}}{\rho_{b_{ik}}})(\underline{Q}_{b_{ik}}\cdot\underline{n}_{b_{ik}})\right)S_{b_{ik}} \\
 & & \displaystyle-\frac{1}{2}\rho_{rus\,b_{ik}}\,(\rho_{b_{ik}}\,e_{b_{ik}}-\rho_i\,e_i)\,S_{b_{ik}}
\end{array}\right. \tag{III.5.26}$$

## 5.2.5 Conditions aux limites pour le flux diffusif d'énergie

### Rappel

Pour le flux de diffusion d'énergie, les conditions aux limites sont imposées de manière similaire à ce qui est décrit dans la documentation de `clptur` et de `condli`. La figure (III.5.1) rappelle quelques notations usuelles et l'équation (III.5.27) traduit la conservation du flux normal au bord pour la variable $f$.

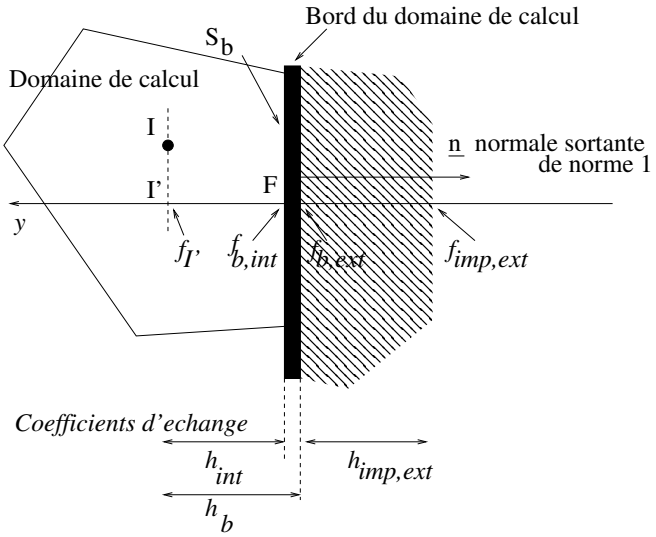



Figure III.5.1: Cellule de bord.

$$\underbrace{h_{int}(f_{b,int}-f_{I'})}_{\phi_{int}}=\underbrace{h_b(f_{b,ext}-f_{I'})}_{\phi_b}=\left\{\begin{array}{ll}
\underbrace{h_{imp,ext}(f_{imp,ext}-f_{b,ext})}_{\phi_{réel\ imposé}} & \text{(condition de Dirichlet)} \\
\underbrace{\phi_{imp,ext}}_{\phi_{réel\ imposé}} & \text{(condition de Neumann)}
\end{array}\right. \tag{III.5.27}$$

L'équation (III.5.28) rappelle la formulation des conditions aux limites pour une variable $f$.

$$f_{b,int} = \begin{cases} \dfrac{h_{imp,ext}}{h_{int}+h_r h_{imp,ext}} & f_{imp,ext} + \dfrac{h_{int}+h_{imp,ext}(h_r-1)}{h_{int}+h_r h_{imp,ext}} f_{I'} & \text{(condition de Dirichlet)} \\ \dfrac{1}{h_{int}} & \phi_{imp,ext} + f_{I'} & \text{(condition de Neumann)} \end{cases} \tag{III.5.28}$$

Les coefficients d'échange sont définis comme suit[2] :

$$\begin{cases} h_{int} = \dfrac{\alpha}{\overline{\overline{I'F}}} \\ h_r = \dfrac{h_{int}}{h_b} \\ h_b = \dfrac{\phi_b}{f_{b,ext}-f_{I'}} = \dfrac{\rho\, C\, u_k}{f^+_{I'}} \end{cases} \tag{III.5.29}$$

Dans *Code_Saturne*, on note les conditions aux limites de manière générale sous la forme suivante :

$$f_{b,int} = A_b + B_b\, f_{I'} \tag{III.5.30}$$

avec $A_b$ et $B_b$ définis selon le type des conditions :

$$\text{Dirichlet} \begin{cases} A_b = \dfrac{h_{imp,ext}}{h_{int}+h_r h_{imp,ext}} f_{imp,ext} \\ B_b = \dfrac{h_{int}+h_{imp,ext}(h_r-1)}{h_{int}+h_r h_{imp,ext}} \end{cases} \qquad \text{Neumann} \begin{cases} A_b = \dfrac{1}{h_{int}} \phi_{imp,ext} \\ B_b = 1 \end{cases} \tag{III.5.31}$$

## Flux diffusif d'énergie

Dans le module compressible, on résout une équation sur l'énergie, qui s'écrit, si l'on excepte tous les termes hormis le flux de diffusion et le terme instationnaire, pour faciliter la présentation :

$$\begin{aligned} \frac{\partial \rho e}{\partial t} &= -\mathrm{div}\, \underline{\Phi}_s \\ &= \mathrm{div}(K\, \underline{\mathrm{grad}}\ T) \quad \text{avec} \quad K = \lambda + C_p \frac{\mu_t}{\sigma_t} \\ &= \mathrm{div}\left( K\, \underline{\mathrm{grad}}\ \frac{e - \frac{1}{2}u^2 - \varepsilon_{sup}}{C_v} \right) \\ &= \mathrm{div}\left( \frac{K}{C_v}\, \underline{\mathrm{grad}}\ (e - \frac{1}{2}u^2 - \varepsilon_{sup}) \right) \quad \text{si} \quad C_v \text{ est constant} \\ &= \mathrm{div}\left( \frac{K}{C_v}\, \underline{\mathrm{grad}}\ e \right) - \mathrm{div}\left( \frac{K}{C_v}\, \underline{\mathrm{grad}}\ (\frac{1}{2}u^2 + \varepsilon_{sup}) \right) \end{aligned} \tag{III.5.32}$$

La décomposition en $e$ et $\frac{1}{2}u^2 + \varepsilon_{sup}$ est purement mathématique (elle résulte du fait que l'on résout en énergie alors que le flux thermique s'exprime en fonction de la température). Aussi, pour imposer un flux de bord ou une température de bord (ce qui revient au même puisque l'on impose toujours finalement la conservation du flux normal), on *choisit* de reporter la totalité de la condition à la limite sur le terme $\frac{K}{C_v}\, \underline{\mathrm{grad}}\ e$ et donc d'annuler le flux associé au terme $\frac{K}{C_v}\, \underline{\mathrm{grad}}\ (\frac{1}{2}u^2 + \varepsilon_{sup})$ (en pratique, pour l'annuler, on se contente de ne pas l'ajouter au second membre de l'équation). Conformément à l'approche retenue dans *Code_Saturne* et rappelée précédemment, on déterminera donc une valeur

[2] On rappelle que, comme dans `condli`, $\alpha$ désigne $\lambda + C_p \frac{\mu_t}{\sigma_t}$ si $f$ est la température, $\frac{\lambda}{C_p} + \frac{\mu_t}{\sigma_t}$ si $f$ représente l'enthalpie. Le coefficient $C$ représente $C_p$ pour la température et vaut 1 pour l'enthalpie. La grandeur adimensionnelle $f^+$ est obtenue par application d'un principe de similitude en paroi : pour la température, elle dépend du nombre de Prandlt moléculaire, du nombre de Prandtl turbulent et de la distance adimensionnelle à la paroi $y^+$ dans la cellule de bord.

de bord *fictive* de l'énergie qui permette de reconstruire le flux diffusif total attendu à partir de la discrétisation du seul terme $\frac{K}{C_v}\underline{\text{grad}}\ e$.

Remarque : dans la version 1.2.0, on utilise $\frac{K}{C_v} = \left(\frac{\lambda}{C_v} + \frac{\mu_t}{\sigma_t}\right)$, à partir de 1.2.1, on utilise la valeur $\frac{K}{C_v} = \left(\frac{\lambda}{C_v} + \frac{C_p}{C_v}\frac{\mu_t}{\sigma_t}\right)$. On notera que le nombre de Prandtl turbulent $\sigma_t$ est associé à la variable résolue et peut être fixé par l'utilisateur.

## Condition de Neumann

La conservation du flux s'écrit :

$$\underbrace{h_{int}(e_{b,int} - e_{I'})}_{\phi_{int}} = \underbrace{\phi_{imp,ext}}_{\phi_{réel\ imposé}} \tag{III.5.33}$$

On a donc dans ce cas :

$$\left\{ \begin{array}{rcl} A_b & = & \dfrac{1}{h_{int}}\phi_{imp,ext} \\ B_b & = & 1 \end{array} \right. \tag{III.5.34}$$

## Condition de Dirichlet

On suppose que la condition de Dirichlet porte sur la température $T_{b,ext}$.

La conservation du flux s'écrit :

$$\underbrace{h_{int}(e_{b,int} - e_{I'})}_{\phi_{int}\ \text{(forme numérique du flux)}} = \underbrace{h_b(T_{b,ext} - T_{I'})}_{\phi_b\ \text{qui intègre l'effet de couche limite}} = \underbrace{h'_{imp,ext}(T_{imp,ext} - T_{b,ext})}_{\phi_{réel\ imposé}} \tag{III.5.35}$$

Avec pour les coefficients d'échange :

$$\left\{ \begin{array}{rcl} h_{int} & = & \dfrac{K}{C_v\,\overline{I'F}} \\ h_b & = & \dfrac{\phi_b}{T_{b,ext} - T_{I'}} = \dfrac{\rho\, C_p\, u_k}{T^+_{I'}} \end{array} \right. \tag{III.5.36}$$

On tire $T_{b,ext}$ de la seconde partie de l'égalité (III.5.35) traduisant la conservation du flux :

$$T_{b,ext} = \frac{h'_{imp,ext}\,T_{imp,ext} + h_b\,T_{I'}}{h_b + h'_{imp,ext}} \tag{III.5.37}$$

En utilisant cette valeur et la première partie de l'équation de conservation du flux (III.5.35), on obtient :

$$e_{b,int} = \frac{h_b\,h'_{imp,ext}}{h_{int}\,(h_b + h'_{imp,ext})}\,(T_{imp,ext} - T_{I'}) + e_{I'} \tag{III.5.38}$$

On utilise alors $T_{I'} = \frac{1}{C_v}\left(e_{I'} - \frac{1}{2}u_i^2 - \varepsilon_{sup,i}\right)$ pour écrire (sans reconstruction pour la vitesse et $\varepsilon_{sup}$) :

$$e_{b,int} = \frac{-\frac{h_b\,h'_{imp,ext}}{C_v} + h_{int}\,(h_b + h'_{imp,ext})}{h_{int}\,(h_b + h'_{imp,ext})}\,e_{I'} + \frac{h_b\,h'_{imp,ext}}{h_{int}\,(h_b + h'_{imp,ext})}\left(T_{imp,ext} + \frac{\frac{1}{2}u_i^2 + \varepsilon_{sup,i}}{C_v}\right) \tag{III.5.39}$$

Et on a donc, avec $h'_r = \frac{h_{int}}{\frac{h_b}{C_v}}$ :

$$e_{b,int} = \underbrace{\frac{h'_{imp,ext}}{C_v\, h_{int} + h'_r\, h'_{imp,ext}} \left( C_v\, T_{imp,ext} + \frac{1}{2} u_i^2 + \varepsilon_{sup,i} \right)}_{A_b} + \underbrace{\frac{C_v\, h_{int} + h'_{imp,ext}(h'_r - 1)}{C_v\, h_{int} + h'_r\, h'_{imp,ext}}}_{B_b} e_{I'} \tag{III.5.40}$$

Avec ces notations, $h_b$ est le coefficient habituellement calculé pour la température.

Le coefficient $h'_{imp,ext}$ est le coefficient d'échange externe qui est imposé pour la température[3]. Pour obtenir l'équivalent dimensionnel de $h'_{imp,ext}$ pour l'énergie, il faut diviser sa valeur par $C_v$ (ce qui ne fait pas de différence dans la majorité des cas, car il est habituellement pris infini).

## 5.3 Mise en œuvre

### 5.3.1 Introduction

Les conditions aux limites sont imposées par une suite de sous-programmes, dans la mesure où l'on a cherché à rester cohérent avec la structure standard de *Code_Saturne*.

Dans `ppprcl` (appelé par `precli`), on initialise les tableaux avant le calcul des conditions aux limites :

- `IZFPPP` (numéro de zone, inutilisé, fixé à zéro),
- `IA(IIFBRU)` (repérage des faces de bord pour lesquelles on applique un schéma de Rusanov : initialisé à zéro, on imposera la valeur 1 dans `cfrusb` pour les faces auxquelles on applique le schéma de Rusanov)
- `IA(IIFBET)` (repérage des faces de paroi à température ou à flux thermique imposé : initialisé à 0, on imposera la valeur 1 dans `cfxtcl` lorsque la température ou le flux est imposé),
- `RCODCL(*,*,1)` (initialisé à `-RINFIN` en prévision du traitement des sorties réentrantes pour lesquelles l'utilisateur aurait fourni une valeur à imposer en Dirichlet),
- flux convectifs de bord pour la quantité de mouvement et l'énergie (initialisés à zéro).

Les types de frontière (`ITYPFB`) et les valeurs nécessaires (`ICODCL`, `RCODCL`) sont imposés par l'utilisateur dans `uscfcl`.

On convertit ensuite ces données dans `condli` pour qu'elles soient directement utilisables lors du calcul des matrices et des seconds membres.

Pour cela, `cfxtcl` permet de réaliser le calcul des valeurs de bord et, pour certaines frontières, des flux convectifs. On fait appel, en particulier, à `uscfth` (utilisation de la thermodynamique) et à `cfrusb` (flux convectifs par le schéma de Rusanov). Lors de ces calculs, on utilise `COEFA` et `COEFB` comme tableaux de travail (transmission de valeurs à `uscfth` en particulier) afin de renseigner `ICODCL` et `RCODCL`. Après `cfxtcl`, le sous-programme `typecl` complète quelques valeurs par défaut de `ICODCL` et de `RCODCL`, en particulier pour les scalaires passifs.

Après `cfxtcl` et `typecl`, les tableaux `ICODCL` et `RCODCL` sont complets. Ils sont utilisés dans la suite de `condli` et en particulier dans `clptur` pour construire les tableaux `COEFA` et `COEFB` (pour l'énergie, on dispose de deux couples (`COEFA`, `COEFB`) afin de traiter les parois).

[3] Le coefficient $h'_{imp,ext}$ est utile pour les cas où l'on souhaite relaxer la condition à la limite : pour la température, cela correspond à imposer une valeur sur la face externe d'une paroi unidimensionnelle idéale, sans inertie, caractérisée par un simple coefficient d'échange.

On présente ci-après les points dont l'implantation diffère de l'approche standard. Il s'agit de l'utilisation d'un schéma de Rusanov pour le calcul des flux convectifs en entrée et sortie (hormis sortie supersonique) et du mode de calcul des flux diffusifs d'énergie en paroi. On insiste en particulier sur l'impact des conditions aux limites sur la construction des seconds membres de l'équation de la quantité de mouvement et de l'équation de l'énergie (`cfqdmv` et `cfener`).

### 5.3.2 Flux de Rusanov pour le calcul des flux convectifs en entrée et sortie

Le schéma de Rusanov est utilisé pour calculer des flux convectifs de bord (masse, quantité de mouvement et énergie) aux entrées et des sorties de type IESICF, ISOPCF, IERUCF, IEQHCF.

La gestion des conditions aux limites est différente de celle adoptée classiquement dans *Code_Saturne*, bien que l'on se soit efforcé de s'y conformer le mieux possible.

En volumes finis, il faut disposer de conditions aux limites pour trois utilisations principales au moins :

- imposer les flux de convection,
- imposer les flux de diffusion,
- calculer les gradients pour les reconstructions.

Dans l'approche standard de *Code_Saturne*, les conditions aux limites sont définies par variable et non pas par terme discret[4]. On dispose donc, *pour chaque variable*, d'une valeur de bord dont devront être déduits les flux de convection, les flux de diffusion et les gradients[5]. Ici, avec l'utilisation d'un schéma de Rusanov, dans lequel le flux convectif est traité dans son ensemble, il est impératif de disposer d'un moyen d'imposer directement sa valeur au bord[6].

Le flux convectif calculé par le schéma de Rusanov sera ajouté directement au second membre des équations de masse, de quantité de mouvement et d'énergie. Comme ce flux contient, outre la contribution des termes convectifs "usuels" ($\mathrm{div}(\underline{Q})$, $\mathrm{div}(\underline{u} \otimes \underline{Q})$ et $\mathrm{div}(\underline{Q}\, e)$), celle des termes en $\underline{\mathrm{grad}}\ P$ (quantité de mouvement) et $\mathrm{div}(\underline{Q}\, \frac{P}{\rho})$ (énergie), il faut veiller à ne pas ajouter une seconde fois les termes de bord issus de $\underline{\mathrm{grad}}\ P$ et de $\mathrm{div}(\underline{Q}\, \frac{P}{\rho})$ au second membre des équations de quantité de mouvement et d'énergie.

Pour la masse, le flux convectif calculé par le schéma de Rusanov définit simplement le flux de masse au bord (`PROPFB(IFAC,IPPROB(IFLUMA(ISCA(IENERG(IPHAS))))))`).

Pour la quantité de mouvement, le flux convectif calculé par le schéma de Rusanov est stocké dans les tableaux `PROPFB(IFAC,IPPROB(IFBRHU(IPHAS)))`, `PROPFB(IFAC,IPPROB(IFBRHV(IPHAS)))` et `PROPFB(IFAC,IPPRO`
Il est ensuite ajouté au second membre de l'équation directement dans `cfqdmv` (boucle sur les faces de bord). Comme ce flux contient la contribution du terme convectif usuel $\underline{\mathrm{div}}(\underline{u} \otimes \underline{Q})$, il ne faut pas l'ajouter dans le sous-programme `cfbsc2`. De plus, le flux convectif calculé par le schéma de Rusanov contient la contribution du gradient de pression. Or, le gradient de pression est calculé dans `cfqdmv` au moyen de `grdcel` et ajouté au second membre sous forme de contribution volumique (par cellule) : il faut donc retirer la contribution des faces de bord auxquelles est appliqué le schéma de Rusanov, pour ne pas la compter deux fois (cette opération est réalisée dans `cfqdmv`).

Pour l'énergie, le flux convectif calculé par le schéma de Rusanov est stocké dans le tableau `PROPFB(IFAC,IPPROB(IFBE`
Pour les faces auxquelles n'est pas appliqué le schéma de Rusanov, on ajoute la contribution du terme

[4] Par exemple, pour un scalaire convecté et diffusé, on définit une valeur de bord unique *pour le scalaire* et non pas une valeur de bord pour le *flux convectif* et une valeur de bord pour le *flux diffusif*.

[5] Néanmoins, pour certaines variables comme la vitesse par exemple, *Code_Saturne* dispose de deux valeurs de bord (et non pas d'une seule) afin de pouvoir imposer de manière indépendante le gradient normal et le flux de diffusion.

[6] Il serait possible de calculer une valeur de bord fictive des variables d'état qui permette de retrouver le flux convectif calculé par le schéma de Rusanov, mais cette valeur ne permettrait pas d'obtenir un flux de diffusion et un gradient satisfaisants.

de transport de pression $\mathrm{div}(\underline{Q}\,\frac{P}{\rho})$ au second membre de l'équation dans `cfener` et on complète le second membre dans `cfbsc2` avec la contribution du terme convectif usuel $\mathrm{div}(\underline{Q}\,e)$. Pour les faces auxquelles est appliqué le schéma de Rusanov, on ajoute directement le flux de Rusanov au second membre de l'équation dans `cfener`, en lieu et place de la contribution du terme de transport de pression et l'on prend garde de ne pas comptabiliser une seconde fois le flux convectif usuel $\underline{\mathrm{div}}(\underline{Q}\,e)$ dans le sous-programme `cfbsc2`.

C'est l'indicateur `IA(IIFBRU)` (renseigné dans `cfrusb`) qui permet, dans `cfbsc2`, `cfqdmv` et `cfener`, de repérer les faces de bord pour lesquelles on a calculé un flux convectif avec le schéma de Rusanov.

### 5.3.3 Flux diffusif d'énergie

#### Introduction

Une condition doit être fournie sur toutes les frontières pour le calcul du flux diffusif d'énergie.

Il n'y a pas lieu de s'étendre particulièrement sur le traitement de certaines frontières. Ainsi, aux entrées et sorties, on dispose d'une valeur de bord (issue de la résolution du problème de Riemann) que l'on utilise dans la formule discrète classique donnant le flux[7]. La situation est simple aux symétries également, où un flux nul est imposé.

Par contre, en paroi, les conditions de température ou de flux thermique imposé doivent être traitées avec plus d'attention, en particulier lorsqu'une couche limite turbulente est présente.

#### Coexistence de deux conditions de bord

Comme indiqué dans la partie "discrétisation", les conditions de température ou de flux conductif imposé en paroi se traduisent, pour le flux d'énergie, au travers du terme $\mathrm{div}\left(\frac{K}{C_v}\,\underline{\mathrm{grad}}\;e\right)$, en imposant une condition de flux nul sur le terme $-\mathrm{div}\left(\frac{K}{C_v}\,\underline{\mathrm{grad}}\;(\frac{1}{2}\,u^2+\varepsilon_{sup})\right)$. Les faces IFAC concernées sont repérées dans `cfxtcl` par l'indicateur `IA(IIFBET+IFAC-1+(IPHAS-1)*NFABOR) = 1` (qui vaut 0 sinon, initialisé dans `ppprcl`).

Sur ces faces, on calcule une valeur de bord de l'énergie, qui, introduite dans la formule générale de flux utilisée au bord dans *Code_Saturne*, permettra de retouver le flux souhaité. La valeur de bord est une simple valeur numérique sans signification physique et ne doit être utilisée que pour calculer le flux diffusif.

En plus de cette valeur de bord destinée à retrouver le flux diffusif, il est nécessaire de disposer d'une seconde valeur de bord de l'énergie afin de pouvoir en calculer le gradient.

Ainsi, comme pour la vitesse en $k-\varepsilon$, il est nécessaire de disposer pour l'énergie de deux couples de coefficients (`COEFA`,`COEFB`), correspondant à deux valeurs de bord distinctes, dont l'une est utilisée pour le calcul du flux diffusif spécifiquement.

#### Calcul des `COEFA` et `COEFB` pour les faces de paroi à température imposée

Les faces de paroi `IFAC` à température imposée sont identifées par l'utilisateur dans `uscfcl` au moyen de l'indicateur `ICODCL(IFAC,ISCA(ITEMPK(IPHAS)))=5` (noter que ce tableau est associé à la température).

Dans `cfxtcl`, on impose alors `ICODCL(IFAC,ISCA(IENERG(IPHAS)))=5` et on calcule la quantité $C_v\,T_{imp,ext}+\frac{1}{2}u_I^2+\varepsilon_{sup,I}$, que l'on stocke dans `RCODCL(IFAC,ISCA(IENERG(IPHAS)),1)` (on ne reconstruit pas les valeurs de $u^2$ et $\varepsilon_{sup}$ au bord, cf. §5.4).

[7] Les valeurs de $u^2$ et de $\varepsilon_{sup}$ ne sont pas reconstruites pour le calcul du gradient au bord dans $\mathrm{div}\left(\frac{K}{C_v}\,\underline{\mathrm{grad}}\;(\frac{1}{2}\,u^2+\varepsilon_{sup})\right)$

À partir de ces valeurs de ICODCL et RCODCL, on renseigne ensuite dans clptur les tableaux de conditions aux limites permettant le calcul du flux : COEFA(*,ICLRTP(ISCA(IENERG(IPHAS)),ICOEFF)) et COEFB(*,ICLRTP(ISCA(IENERG(IPHAS)),ICOEFF)) (noter l'indicateur ICOEFF qui renvoie aux coefficients dédiés au flux diffusif).

## Calcul des COEFA et COEFB pour les faces de paroi à flux thermique imposé

Les faces de paroi IFAC à flux thermique imposé sont identifées par l'utilisateur dans uscfcl au moyen de l'indicateur ICODCL(IFAC,ISCA(ITEMPK(IPHAS)))=3 (noter que le tableau est associé à la température).

Dans cfxtcl, on impose alors ICODCL(IFAC,ISCA(IENERG(IPHAS)))=3 et on transfère la valeur du flux de RCODCL(IFAC,ISCA(ITEMPK(IPHAS)),3) à RCODCL(IFAC,ISCA(IENERG(IPHAS)),3).

À partir de ces valeurs de ICODCL et RCODCL, on renseigne ensuite dans condli les tableaux de conditions aux limites permettant le calcul du flux, COEFA(*,ICLRTP(ISCA(IENERG(IPHAS)),ICOEFF)) et COEFB(*,ICLRTP(ISCA(IENERG(IPHAS)),ICOEFF)) (noter l'indicateur ICOEFF qui renvoie aux coefficients dédiés au flux diffusif).

## Gradient de l'énergie en paroi à température ou à flux thermique imposé

Dans les deux cas (paroi à température ou à flux thermique imposé), on utilise les tableaux COEFA(*,ICLRTP(ISCA(II) COEFB(*,ICLRTP(ISCA(II),ICOEF)) (noter le ICOEF) pour disposer d'une condition de flux nul pour l'énergie (avec II=IENERG(IPHAS)) et pour la température (avec II=ITEMPK(IPHAS)) si un calcul de gradient est requis.

Un gradient est en particulier utile pour les reconstructions de l'énergie sur maillage non orthogonal. Pour la température, il s'agit d'une précaution, au cas où l'utilisateur aurait besoin d'en calculer le gradient.

## Autres frontières que les parois à température ou à flux thermique imposé

Pour les frontières qui ne sont pas des parois à température ou à flux thermique imposé, les conditions aux limites de l'énergie et de la température sont complétées classiquement dans condli selon les choix faits dans cfxtcl pour ICODCL et RCODCL.

En particulier, dans le cas de conditions de Dirichlet sur l'énergie (entrées, sorties), les deux jeux de conditions aux limites sont identiques (tableaux COEFA, COEFB avec ICOEFF et ICOEF).

Si un flux est imposé pour l'énergie totale (condition assez rare, l'utilisateur ne raisonnant pas, d'ordinaire, en énergie totale), on le stocke au moyen de COEFA(*,ICLRTP(ISCA(IENERG(IPHAS)),ICOEFF)) et COEFB(*,ICLRTP(ISCA(IENERG(IPHAS)),ICOEFF)) (tableaux associés au flux diffusif). Pour le gradient, une condition de flux nul est stockée dans COEFA(*,ICLRTP(ISCA(IENERG(IPHAS)),ICOEF)) et COEFB(*,ICLRTP(ISCA(IENERG(IPHAS)),ICOEF)). On peut remarquer que les deux jeux de conditions aux limites sont identiques pour les faces de symétrie.

## Impact dans cfener

Lors de la construction des seconds membres, dans cfener, on utilise les conditions aux limites stockées dans les tableaux associés au flux diffusif COEFA(*,ICLRTP(ISCA(IENERG(IPHAS)),ICOEFF)) et COEFB(*,ICLRTP(ISCA(IENERG(IPHAS)),ICOEFF)) pour le terme de flux diffusif $\mathrm{div}\left(\frac{K}{C_v}\,\underline{\mathrm{grad}}\ e\right)$ en prenant soin d'annuler la contribution de bord du terme $-\mathrm{div}\left(\frac{K}{C_v}\,\underline{\mathrm{grad}}\ (\frac{1}{2}\,u^2+\varepsilon_{sup})\right)$ sur les faces pour lesquelles cette condition prend les deux termes en compte, c'est-à-dire sur les faces pour lesquelles IA(IIFBET+IFAC-1+(IPHAS-1)*NFABOR) = 1.

Pour tous les autres termes qui requièrent une valeur de bord, on utilise les conditions aux limites que l'on a stockées au moyen des deux tableaux `COEFA(*,ICLRTP(ISCA(IENERG(IPHAS)),ICOEF))` et `COEFB(*,ICLRTP(ISCA(IENERG(IPHAS)),ICOEF))`. Ces conditions sont donc en particulier utilisées pour le calcul du gradient de l'énergie, lors des reconstructions sur maillage non orthogonal.

## 5.4 Points à traiter

Apporter un complément de test sur une cavité fermée sans vitesse et sans gravité, avec flux de bord ou température de bord imposée. Il semble que le transfert d'énergie *via* les termes de pression génère de fortes vitesses non physiques dans la première maille de paroi et que la conduction thermique ne parvienne pas à établir le profil de température recherché. Il est également possible que la condition de bord sur la pression génère une perturbation (une extrapolation pourrait se révéler indispensable).

Il pourrait être utile de généraliser à l'incompressible l'approche utilisée en compressible pour unifier simplement le traitement des sorties de type 9 et 10.

Il pourrait être utile d'étudier plus en détail l'influence de la non orthogonalité des mailles en sortie supersonique (pas de reconstruction, ce qui n'est pas consistant pour les flux de diffusion).

De même, il serait utile d'étudier l'influence de l'absence de reconstruction pour la vitesse et $\varepsilon_{sup}$ dans la relation $T_{I'} = \dfrac{1}{C_v}\left(e_{I'} - \dfrac{1}{2}u_i^2 - \varepsilon_{sup,i}\right)$ utilisée pour les parois à température imposée.

Apporter un complément de documentation pour le couplage avec SYRTHES (conversion énergie température). Ce n'est pas une priorité.

Pour les thermodynamiques à $\gamma$ variable, il sera nécessaire de modifier non seulement `uscfth` mais également `cfrusb` qui doit disposer de $\gamma$ en argument.

Pour les thermodynamiques à $C_v$ variable, il sera nécessaire de prendre en compte un terme en $\underline{\text{grad}}\ C_v$, issu des flux diffusifs, au second membre de l'équation de l'énergie (on pourra cependant remarquer qu'actuellement, en incompressible, on néglige le terme en $\underline{\text{grad}}\ C_p$ dans l'équation de l'enthalpie).

# Part IV

# Module électrique

# 1- Sous-programme `elec**`

On s'intéresse à la résolution des équations de la magnétohydrodynamique, constituées de la réunion des équations de l'aérothermodynamique et des équations de Maxwell.

On se place dans deux cadres d'utilisation bien spécifiques et distincts, qui permettront chacun de réaliser des simplifications : les études dites "d'arc électrique" (dans lesquelles sont prises en compte les forces de Laplace et l'effet Joule) et les études dites "Joule" (dans lesquelles seul l'effet Joule est pris en compte).

Les études d'arc électrique sont associées en grande partie, pour EDF, aux problématiques relatives aux transformateurs. Les études Joule sont plus sécifiquement liées aux phénomènes rencontrés dans les fours verriers.

Outre la prise en compte ou non des forces de Laplace, ces deux types d'études se différencient également par le mode de détermination de l'effet Joule (utilisation d'un potentiel complexe pour les études Joule faisant intervenir un courant alternatif non monophasé).

On décrit tout d'abord les équations résolues pour les études d'arc électrique. Les spécificités des études Joule seront abordées ensuite.

Pour l'arc électrique, les références [douce] et [delalondre] pourront compléter la présentation :

**[delalondre]** Delalondre, Clarisse : "Modélisation aérothermodynamique d'arcs électriques à forte intensité avec prise en compte du déséquilibre thermodynamique local et du transfert thermique à la cathode", Thèse de l'Université de Rouen, 1990

**[douce]** Douce, Alexandre : "Modélisation 3-D du chauffage d'un bain métallique par plasma d'arc transféré. Application à un réacteur axisymétrique", HE-26/99/027A, HE-44/99/043A, Thèse de l'Ecole Centrale Paris et EDF, 1999

# 1.1 Fonction

## 1.1.1 Notations

**Variables utilisées**

| | | |
|---|---|---|
| $\underline{A}$ | potentiel vecteur réel | $kg\,m\,s^{-2}\,A^{-1}$ |
| $\underline{B}$ | champ magnétique | $T$ (ou $kg\,s^{-2}\,A^{-1}$) |
| $\underline{D}$ | déplacement électrique | $A\,s\,m^{-2}$ |
| $\underline{E}$ | champ électrique | $V\,m^{-1}$ |
| $E$ | énergie totale massique | $J\,kg^{-1}$ (ou $m^2\,s^{-2}$) |
| $e$ | énergie interne massique | $J\,kg^{-1}$ (ou $m^2\,s^{-2}$) |
| $e_c$ | énergie cinétique massique | $J\,kg^{-1}$ (ou $m^2\,s^{-2}$) |
| $\underline{H}$ | excitation magnétique | $A\,m^{-1}$ |
| $h$ | enthalpie massique | $J\,kg^{-1}$ (ou $m^2\,s^{-2}$) |
| $\underline{j}$ | densité de courant | $A\,m^{-2}$ |
| $P$ | pression | $kg\,m^{-1}\,s^{-2}$ |
| $P_R$, $P_I$ | potentiel scalaire réel, imaginaire | $V$ (ou $kg\,m^2\,s^{-3}\,A^{-1}$) |
| $\underline{u}$ | vitesse | $m\,s^{-1}$ |
| | | |
| $\varepsilon$ | permittivité électrique | $F\,m^{-1}$ (ou $m^{-3}\,kg^{-1}\,s^4\,A^2$) |
| $\varepsilon_0$ | permittivité électrique du vide | $8,854\,10^{-12}\ F\,m^{-1}$ (ou $m^{-3}\,kg^{-1}\,s^4\,A^2$) |
| $\mu$ | perméabilité électrique | $H\,m^{-1}$ (ou $m\,kg\,s^{-2}\,A^{-2}$) |
| $\mu_0$ | perméabilité électrique du vide | $4\,\pi\,10^{-7}\ H\,m^{-1}$ (ou $m\,kg\,s^{-2}\,A^{-2}$) |
| $\sigma$ | conductivité électrique | $S\,m^{-1}$ (ou $m^{-3}\,kg^{-1}\,s^3\,A^2$) |

**Notations d'analyse vectorielle**

On rappelle également la définition des notations employées[1] :

$$\left\{\begin{array}{lcl} \left[\underline{\underline{\mathrm{grad}}}\ \underline{a}\right]_{ij} & = & \partial_j a_i \\ \left[\mathrm{div}(\underline{\underline{\sigma}})\right]_i & = & \partial_j \sigma_{ij} \\ \left[\underline{a}\otimes\underline{b}\right]_{ij} & = & a_i\,b_j \end{array}\right.$$

et donc :

$$\left[\mathrm{div}(\underline{a}\otimes\underline{b})\right]_i = \partial_j(a_i\,b_j)$$

## 1.1.2 Arcs électriques

### Introduction

Pour les études d'arc électrique, on calcule, à un pas de temps donné :

- la vitesse $\underline{u}$, la pression $P$, la variable énergétique enthalpie $h$ (et les grandeurs turbulentes),
- un potentiel scalaire réel $P_R$ (dont le gradient permet d'obtenir le champ électrique $\underline{E}$ et la densité de courant $\underline{j}$),
- un potentiel vecteur réel $\underline{A}$ (dont le rotationnel permet d'obtenir le champ magnétique $\underline{B}$).

Le champ électrique, la densité de courant et le champ magnétique sont utilisés pour calculer les termes sources d'effet Joule et les forces de Laplace qui interviennent respectivement dans l'équation de l'enthalpie et dans celle de la quantité de mouvement.

[1] en utilisant la convention de sommation d'Einstein.

## Équations continues

### Système d'équations

Les équations continues qui sont résolues sont les suivantes :

$$\left\{\begin{array}{l} \mathrm{div}(\rho\underline{u}) = 0 \\ \dfrac{\partial}{\partial t}(\rho\underline{u}) + \mathrm{div}(\rho\,\underline{u}\otimes\underline{u}) = \mathrm{div}(\underline{\underline{\sigma}}) + \underline{TS} + \underline{j}\times\underline{B} \\ \dfrac{\partial}{\partial t}(\rho h) + \mathrm{div}(\rho\,\underline{u}h) = \Phi_v - \mathrm{div}\left(\left(\dfrac{\lambda}{C_p} + \dfrac{\mu_t}{\sigma_t}\right)\underline{\mathrm{grad}}\ h\right) + P_J \\ \mathrm{div}(\sigma\,\underline{\mathrm{grad}}\ P_R) = 0 \\ \mathrm{div}(\underline{\underline{\mathrm{grad}}}\ \underline{A}) = -\mu_0\underline{j} \end{array}\right. \tag{IV.1.1}$$

avec les relations suivantes :

$$\left\{\begin{array}{l} P_J = \underline{j}\cdot\underline{E} \\ \underline{E} = -\underline{\mathrm{grad}}\ P_R \\ \underline{j} = \sigma\underline{E} \end{array}\right. \tag{IV.1.2}$$

### Équation de la masse

C'est l'équation résolue en standard par *Code_Saturne* (contrainte stationnaire). Elle n'a pas de traitement particulier dans le cadre du module présent. Un terme source de masse peut être pris en compte au second membre si l'utilisateur le souhaite. Pour simplifier l'exposé le terme source sera supposé nul ici, dans la mesure où il n'est pas spécifique au module électrique.

### Équation de la quantité de mouvement

Elle présente, par rapport à l'équation standard résolue par *Code_Saturne*, un seul terme additionnel ($\underline{j}\times\underline{B}$) qui rend compte des forces de Laplace. Pour l'obtenir, on fait l'hypothèse que le milieu est électriquement neutre.

En effet, une charge $q_i$ (Coulomb) animée d'une vitesse $\underline{v}_i$ subit, sous l'effet du champ électrique $\underline{E}$ ($V\,m^{-1}$) et du champ magnétique $\underline{B}$ (Tesla), une force $\underline{f}_i$ ($kg\,m\,s^{-2}$) :

$$\underline{f}_i = q_i\,(\underline{E} + \underline{v}_i\times\underline{B}) \tag{IV.1.3}$$

Avec $n_i$ charges de type $q_i$ par unité de volume et en sommant sur tous les types de charge $i$ (électrons, ions, molécules ionisées...), on obtient la force de Laplace totale $\underline{F}_L$ ($kg\,m^{-2}\,s^{-2}$) subie par unité de volume :

$$\underline{F}_L = \sum_i [n_i\,q_i\,(\underline{E} + \underline{v}_i\times\underline{B})] \tag{IV.1.4}$$

On introduit alors la densité de courant $\underline{j}$ ($A\,m^{-2}$) :

$$\underline{j} = \sum_i n_i\,q_i\,\underline{v}_i \tag{IV.1.5}$$

Avec l'hypothèse que le milieu est électriquement neutre (à un niveau macroscopique) :

$$\sum_i n_i\,q_i = 0 \tag{IV.1.6}$$

la force totale $\underline{F}_L$ s'écrit alors :

$$\underline{F}_L = \underline{j}\times\underline{B} \tag{IV.1.7}$$

et on peut donc écrire l'équation de la quantité de mouvement :

$$\frac{\partial}{\partial t}(\rho\underline{u}) + \mathrm{div}(\rho\,\underline{u}\otimes\underline{u}) = \mathrm{div}(\underline{\underline{\sigma}}) + \underline{TS} + \underline{j}\times\underline{B} \tag{IV.1.8}$$

### Équation de l'enthalpie

Elle est obtenue à partir de l'équation de l'énergie après plusieurs approximations utilisées en standard dans *Code_Saturne* et en prenant en compte le terme d'effet Joule lié à l'énergie électromagnétique.

Énergie électromagnétique

Avec les mêmes notations que précédemment mais sans qu'il soit besoin de supposer que le milieu est électriquement neutre, la puissance reçue par une charge $q_i$ (particule douée de masse) de vitesse $\underline{v}_i$ (vitesse du porteur de charge, contenant éventuellement l'effet de la vitesse du fluide) sous l'effet du champ électrique $\underline{E}$ ($V\,m^{-1}$) et du champ magnétique $\underline{B}$ ($T$) est (sans sommation sur $i$) :

$$P_i = \underline{f}_i \cdot \underline{v}_i = q_i(\underline{E} + \underline{v}_i \times \underline{B}) \cdot \underline{v}_i = q_i \underline{v}_i \cdot \underline{E} \tag{IV.1.9}$$

Avec $n_i$ charges par unité de volume et en sommant sur tous les types de charges $i$, on obtient la puissance totale par unité de volume :

$$P_J = \sum_i n_i\, q_i\, \underline{v}_i \cdot \underline{E} \tag{IV.1.10}$$

On introduit alors la densité de courant $\underline{j} = \sum_i n_i\, q_i\, \underline{v}_i$ (en $A\,m^{-2}$) et on obtient l'expression usuelle de la puissance électromagnétique dissipée par effet Joule (en $W\,m^{-3}$) :

$$P_J = \underline{j} \cdot \underline{E} \tag{IV.1.11}$$

Pour reformuler la puissance dissipée par effet Joule et obtenir une équation d'évolution de l'énergie électromagnétique, on utilise alors les équations de Maxwell. Les équations s'écrivent (lois d'Ampère et de Faraday) :

$$\left\{ \begin{array}{l} \dfrac{\partial \underline{D}}{\partial t} - \underline{\mathrm{rot}}\ \underline{H} = -\underline{j} \\ \dfrac{\partial \underline{B}}{\partial t} + \underline{\mathrm{rot}}\ \underline{E} = 0 \end{array} \right. \tag{IV.1.12}$$

On a donc :

$$P_J = \underline{j} \cdot \underline{E} = \left( -\frac{\partial \underline{D}}{\partial t} + \underline{\mathrm{rot}}\ \underline{H} \right) \cdot \underline{E} \tag{IV.1.13}$$

On utilise alors la relation suivante :

$$\underline{\mathrm{rot}}\ \underline{H} \cdot \underline{E} = \underline{H} \cdot \underline{\mathrm{rot}}\ \underline{E} - \mathrm{div}(\underline{E} \times \underline{H}) \tag{IV.1.14}$$

En effet, elle permet de faire apparaître un terme en divergence, caractéristique d'une redistribution spatiale :

$$\underline{j} \cdot \underline{E} \quad = \quad -\frac{\partial \underline{D}}{\partial t} \cdot \underline{E} + \underline{H} \cdot \underline{\mathrm{rot}}\ \underline{E} - \mathrm{div}(\underline{E} \times \underline{H}) \tag{IV.1.15}$$

Et en utilisant la loi de Faraday pour faire apparaître la dérivée en temps du champ magnétique :

$$\underline{j} \cdot \underline{E} \quad = \quad -\frac{\partial \underline{D}}{\partial t} \cdot \underline{E} - \underline{H} \cdot \frac{\partial \underline{B}}{\partial t} - \mathrm{div}(\underline{E} \times \underline{H}) \tag{IV.1.16}$$

Dans le cadre de *Code_Saturne*, on fait les hypothèses suivantes :

- la perméabilité $\varepsilon$ et la permittivité $\mu$ sont constantes et uniformes (pour les gaz, en pratique, on utilise les propriétés du vide $\varepsilon_0$ et $\mu_0$).
- on utilise $\underline{B} = \mu \underline{H}$ et $\underline{D} = \varepsilon \underline{E}$

On a alors :

$$\underline{j}\cdot\underline{E} \quad = \quad -\frac{\varepsilon_0}{2}\frac{\partial E^2}{\partial t} - \frac{1}{2\,\mu_0}\frac{\partial B^2}{\partial t} - \frac{1}{\mu_0}\mathrm{div}(\underline{E}\times\underline{B}) \tag{IV.1.17}$$

Énergie totale

On établit l'équation de l'énergie totale en prenant en compte la puissance des forces de Laplace et le terme d'effet Joule.

Sans prendre en compte l'énergie électromagnétique, le premier principe de la thermodynamique s'écrit d'ordinaire sous la forme suivante (pour un volume matériel suivi sur une unité de temps) :

$$d\int_V \rho E dV = \delta Q + \delta W \tag{IV.1.18}$$

Dans cette relation, $E$ est l'énergie totale par unité de masse[2], soit $E = e + e_c$, $e$ étant l'énergie interne massique et $e_c = \frac{1}{2}\,\underline{u}\cdot\underline{u}$ l'énergie cinétique massique. Le terme $\delta Q$ représente la chaleur reçue au travers des frontières du domaine considéré tandis que le terme $\delta W$ représente le travail des forces extérieures reçu par le système (y compris les forces dérivant d'une énergie potentielle).

Pour prendre en compte l'énergie électromagnétique, il suffit d'intégrer à la relation (IV.1.18) la puissance des forces de Laplace $(\underline{j}\times\underline{B})\cdot\underline{u}$ et le terme d'effet Joule $\underline{j}\cdot\underline{E}$ (transformation volumique d'énergie électromagnétique en énergie totale[3]). Dans cette relation, la vitesse $\underline{u}$ est la vitesse du fluide et non pas celle des porteurs de charge : elle n'est donc pas nécessairement coliéaire au vecteur $\underline{j}$ (par exemple, si le courant est dû à des électrons, la vitesse du fluide pourra être considérée comme décorrélée de la vitesse des porteurs de charges ; par contre, si le courant est dû à des ions, la vitesse du fluide pourra être plus directement influencée par le déplacement des porteurs de charge). Ainsi, le premier principe de la thermodynamique s'écrit :

$$d\int_V \rho E dV = \delta Q + \delta W + \underline{j}\cdot\underline{E}\,V\,dt + (\underline{j}\times\underline{B})\cdot\underline{u}\,V\,dt \tag{IV.1.19}$$

et l'équation locale pour l'énergie totale est alors :

$$\frac{\partial}{\partial t}(\rho E) + \mathrm{div}(\rho\,\underline{u}E) = \mathrm{div}(\underline{\underline{\sigma}}\,\underline{u}) + \underline{TS}\cdot\underline{u} + (\underline{j}\times\underline{B})\cdot\underline{u} + \Phi_v - \mathrm{div}\underline{\Phi}_s + \underline{j}\cdot\underline{E} \tag{IV.1.20}$$

Le terme $\Phi_v$ représente les termes sources volumiques d'énergie autres que l'effet Joule (par exemple, il inclut le terme source de rayonnement, pour un milieu optiquement non transparent). Le terme $\underline{\Phi}_s$ est le flux d'énergie surfacique[4].

Enthalpie

Pour obtenir une équation sur l'enthalpie, qui est la variable énergétique choisie dans *Code_Saturne* dans le module électrique, on soustrait tout d'abord à l'équation de l'énergie totale celle de l'énergie cinétique pour obtenir une équation sur l'énergie interne.

L'équation de l'énergie cinétique (obtenue à partir de l'équation de la quantité de mouvement écrite sous forme non conservative) est :

$$\frac{\partial}{\partial t}(\rho e_c) + \mathrm{div}(\rho\,\underline{u}e_c) = \mathrm{div}(\underline{\underline{\sigma}}\,\underline{u}) - \underline{\underline{\sigma}} : \left(\underline{\underline{\mathrm{grad}}}\ (\underline{u})\right)^t + \underline{TS}\cdot\underline{u} + (\underline{j}\times\underline{B})\cdot\underline{u} \tag{IV.1.21}$$

de sorte que, pour l'énergie interne, on a :

$$\frac{\partial}{\partial t}(\rho e) + \mathrm{div}(\rho\,\underline{u}e) = \underline{\underline{\sigma}} : \left(\underline{\underline{\mathrm{grad}}}\ (\underline{u})\right)^t + \Phi_v - \mathrm{div}\Phi_s + \underline{j}\cdot\underline{E} \tag{IV.1.22}$$

[2] Ne pas confondre le scalaire $E$, énergie totale, avec le vecteur $\underline{E}$, champ électrique.

[3] Le terme en divergence $-\frac{1}{\mu_0}\mathrm{div}(\underline{E}\times\underline{B})$ traduit une redistribution spatiale d'énergie électromagnétique : ce n'est donc pas un terme source pour l'énergie totale.

[4] Dans *Code_Saturne*, il est modélisé par une hypothèse de gradient et inclut également la "diffusion" turbulente.

et enfin, pour l'enthalpie $h = e + \frac{P}{\rho}$ :

$$\frac{\partial}{\partial t}(\rho h) + \mathrm{div}(\rho \underline{u} h) = \underline{\underline{\sigma}} : \left(\underline{\underline{\mathrm{grad}}}\ (\underline{u})\right)^t + \Phi_v - \mathrm{div}\Phi_s + \underline{j} \cdot \underline{E} + \rho \frac{d}{dt}\left(\frac{P}{\rho}\right) \tag{IV.1.23}$$

En faisant apparaître la pression dans le tenseur des contraintes $\underline{\underline{\sigma}} = -P\underline{\underline{Id}} + \underline{\underline{\tau}}$, on peut écrire :

$$\frac{\partial}{\partial t}(\rho h) + \mathrm{div}(\rho \underline{u} h) = \underline{\underline{\tau}} : \left(\underline{\underline{\mathrm{grad}}}\ (\underline{u})\right)^t + \Phi_v - \mathrm{div}\Phi_s + \underline{j} \cdot \underline{E} + \frac{dP}{dt} \tag{IV.1.24}$$

Les approximations habituelles de *Code_Saturne* consistent alors à négliger le terme "d'échauffement" issu du tenseur des contraintes $\underline{\underline{\tau}} : \left(\underline{\underline{\mathrm{grad}}}\ (\underline{u})\right)^t$ et le terme en dérivée totale de la pression $\frac{dP}{dt}$, supposés faibles en comparaison des autres termes dans les applications traitées (exemple : terme d'effet Joule important, effets de compressibilité faibles...). De plus, le terme de flux est modélisé en suivant une hypothèse de gradient appliqué à l'enthalpie (et non pas à la température), soit donc :

$$\frac{\partial}{\partial t}(\rho h) + \mathrm{div}(\rho \underline{u} h) = \Phi_v - \mathrm{div}\left(\left(\frac{\lambda}{C_p} + \frac{\mu_t}{\sigma_t}\right) \underline{\mathrm{grad}}\ h\right) + \underline{j} \cdot \underline{E} \tag{IV.1.25}$$

### Équations électromagnétiques

Elles sont obtenues à partir des équations de Maxwell sous les hypothèses détaillées dans [douce], paragraphe 3.3.

Densité de courant

La relation liant la densité de courant et le champ électrique est issue de la loi d'Ohm que l'on suppose pouvoir utiliser sous la forme simplifiée suivante :

$$\underline{j} = \sigma\, \underline{E} \tag{IV.1.26}$$

Champ électrique

Le champ électrique s'obtient à partir d'un potentiel vecteur.

En effet, la loi de Faraday s'écrit :

$$\frac{\partial \underline{B}}{\partial t} + \underline{\mathrm{rot}}\ \underline{E} = 0 \tag{IV.1.27}$$

Avec une hypothèse quasi-stationnaire, il reste :

$$\underline{\mathrm{rot}}\ \underline{E} = 0 \tag{IV.1.28}$$

Il est donc possible de postuler l'existence d'un potentiel scalaire $P_R$ tel que :

$$\underline{E} = -\underline{\mathrm{grad}}\ P_R \tag{IV.1.29}$$

Potentiel scalaire

Le potentiel scalaire est solution d'une équation de Poisson.

En effet, la conservation de la charge $q$ s'écrit :

$$\frac{\partial q}{\partial t} + \mathrm{div}(\underline{j}) = 0 \tag{IV.1.30}$$

Pour un milieu électriquement neutre (à l'échelle macroscopique), on a $\frac{\partial q}{\partial t} = 0$ soit donc :

$$\mathrm{div}(\underline{j}) = 0 \tag{IV.1.31}$$

C'est-à-dire, avec la loi d'Ohm (IV.1.26),

$$\mathrm{div}(\sigma\,\underline{E}) = 0 \tag{IV.1.32}$$

Avec (IV.1.29), on obtient donc une équation permettant de calculer le potentiel scalaire :

$$\mathrm{div}(\sigma\,\underline{\mathrm{grad}}\ P_R) = 0 \tag{IV.1.33}$$

Champ magnétique

Le champ magnétique s'obtient à partir d'un potentiel vecteur.

En effet, la loi d'Ampère s'écrit :

$$\frac{\partial \underline{D}}{\partial t} - \underline{\mathrm{rot}}\ \underline{H} = -\underline{j} \tag{IV.1.34}$$

Sous les hypothèses indiquées précédemment, on écrit :

$$\varepsilon_0\,\mu_0\,\frac{\partial \underline{E}}{\partial t} - \underline{\mathrm{rot}}\ \underline{B} = -\mu_0\underline{j} \tag{IV.1.35}$$

Avec une hypothèse quasi-stationnaire, il reste :

$$\underline{\mathrm{rot}}\ \underline{B} = \mu_0\underline{j} \tag{IV.1.36}$$

De plus, la conservation du flux magnétique s'écrit[5] :

$$\mathrm{div}\,\underline{B} = 0 \tag{IV.1.37}$$

et on peut donc postuler l'existence d'un potentiel vecteur $\underline{A}$ tel que :

$$\underline{B} = \underline{\mathrm{rot}}\ \underline{A} \tag{IV.1.38}$$

Potentiel vecteur

Le potentiel vecteur est solution d'une équation de Poisson.

En prenant le rotationnel de (IV.1.38) et avec (IV.1.36), on obtient :

$$-\underline{\mathrm{rot}}\ (\underline{\mathrm{rot}}\ \underline{A}) = -\mu_0\underline{j} \tag{IV.1.39}$$

Avec la relation donnant le Laplacien[6] d'un vecteur $\mathrm{div}(\underline{\underline{\mathrm{grad}}}\ \underline{a}) = \underline{\mathrm{grad}}\ (\mathrm{div}\underline{a}) - \underline{\mathrm{rot}}\ (\underline{\mathrm{rot}}\ \underline{a})$ et sous la contrainte[7] que $\mathrm{div}\underline{A} = 0$, on obtient finalement une équation permettant de calculer le potentiel vecteur :

$$\mathrm{div}\,(\underline{\underline{\mathrm{grad}}}\ \underline{A}) = -\mu_0\underline{j} \tag{IV.1.40}$$

## 1.1.3 Effet Joule

### Introduction

Pour les études Joule, on calcule, à un pas de temps donné :

- la vitesse $\underline{u}$, la pression $P$, la variable énergétique enthalpie $h$ (et les grandeurs turbulentes éventuelles),
- un potentiel scalaire réel $P_R$,

[5] Prendre la divergence de la loi de Faraday, avec $\mathrm{div}(\underline{\mathrm{rot}}\ \underline{E}) = 0$ (par analyse vectorielle) donne $\mathrm{div}\underline{B} = \mathrm{cst}$.

[6] En coordonnées cartésiennes, le Laplacien du vecteur $\underline{a}$ est le vecteur dont les composantes sont égales au Laplacien de chacune des composantes de $\underline{a}$.

[7] La condition $\mathrm{div}\underline{A} = 0$, dite "jauge de Coulomb", est nécessaire pour assurer l'unicité du potentiel vecteur.

- et, si le courant n'est ni continu, ni alternatif monophasé, un potentiel scalaire imaginaire $P_I$.

Le gradient du potentiel permet d'obtenir le champ électrique $\underline{E}$ et la densité de courant $\underline{j}$ (partie réelle et, éventuellement, partie imaginaire). Le champ électrique et la densité de courant sont utilisés pour calculer le terme source d'effet Joule qui intervient dans l'équation de l'enthalpie.

**La puissance instantanée dissipée par effet Joule** est égale au produit instantané $\underline{j} \cdot \underline{E}$. Dans le cas général, $\underline{j}$ et $\underline{E}$ sont des signaux alternatifs ($\underline{j} = \underline{|j|} cos(\omega\, t + \phi_j)$ et $\underline{E} = \underline{|E|} cos(\omega\, t + \phi_E)$) que l'on peut représenter par des complexes ($\underline{j} = \underline{|j|}\, e^{i\,(\omega\, t + \phi_j)}$ et $\underline{E} = \underline{|E|}\, e^{i\,(\omega\, t + \phi_E)}$). La puissance instantanée s'écrit alors $(\underline{|j|} \cdot \underline{|E|}) cos(\omega\, t + \phi_j) cos(\omega\, t + \phi_E)$.

- **En courant continu** ($\omega = \phi_j = \phi_E = 0$), la puissance se calcule donc simplement comme le produit scalaire $P_J = \underline{|j|} \cdot \underline{|E|}$. Le calcul de la puissance dissipée par effet Joule ne pose donc pas de problème particulier car les variables densité de courant et champ électrique résolues par *Code_Saturne* sont précisément $\underline{|j|}$ et $\underline{|E|}$ (les variables sont réelles).
- **En courant alternatif**, la periode du courant est beaucoup plus petite que les échelles de temps des phénomènes thermohydrauliques pris en compte. Il n'est donc pas utile de disposer de la puissance instantanée dissipée par effet Joule : la moyenne sur une période est suffisante et elle s'écrit[8] : $P_J = \frac{1}{2}(\underline{|j|} \cdot \underline{|E|}) cos(\phi_j - \phi_E)$. Cette formule peut également s'écrire de manière équivalente sous forme complexe : $P_J = \frac{1}{2}\underline{j} \cdot \underline{E}^*$, où $\underline{E}^*$ est le complexe conjugué de $\underline{E}$.
  - En courant alternatif monophasé ($\phi_j = \phi_E$), en particulier, la formule donnant la puissance se simplifie sous la forme $P_J = \frac{1}{2}(\underline{|j|} \cdot \underline{|E|})$, ou encore : $P_J = \frac{1}{\sqrt{2}}\underline{|j|} \cdot \frac{1}{\sqrt{2}}\underline{|E|}$. Il s'agit donc du produit des valeurs efficaces. Or, les variables résolues par *Code_Saturne* en courant alternatif monophasé sont précisément les valeurs efficaces (valeurs que l'on dénomme abusivement "valeurs réelles" dans le code source).
  - En courant alternatif non monophasé (triphasé, en particulier), la formule donnant la puissance est utilisée directement sous la forme $P_J = \frac{1}{2}\underline{j} \cdot \underline{E}^*$. On utilise pour la calculer les variables résolues qui sont la partie réelle et la partie imaginaire de $\underline{j}$ et $\underline{E}$.
- **En conclusion**,
  - en continu, les variables résolues $\underline{j}_{Res}$ et $\underline{E}_{Res}$ sont les variables réelles continues et la puissance se calcule par la formule suivante : $P_J = \underline{j}_{Res} \cdot \underline{E}_{Res}$
  - en alternatif monophasé, les variables résolues $\underline{j}_{Res}$ et $\underline{E}_{Res}$ sont les valeurs efficaces et la puissance se calcule par la formule suivante : $P_J = \underline{j}_{Res} \cdot \underline{E}_{Res}$
  - en alternatif non monophasé, les variables résolues $\underline{j}_{Res,R}$, $\underline{j}_{Res,I}$ et $\underline{E}_{Res,R}$, $\underline{E}_{Res,I}$ sont la partie réelle et la partie imaginaire de $\underline{j}$ et $\underline{E}$, et la puissance se calcule par la formule suivante : $P_J = \frac{1}{2}(\underline{j}_{Res,R} \cdot \underline{E}_{Res,R} - \underline{j}_{Res,I}\underline{E}_{Res,I})$

**Le potentiel imaginaire n'est donc utilisé dans le code que lorsque le courant est alternatif et non monophasé.** En particulier, le potentiel imaginaire n'est pas utilisé lorsque le courant est continu ou alternatif monophasé. En effet, la partie imaginaire n'est introduite en complément de la partie réelle que dans le cas où il est nécessaire de disposer de deux grandeurs pour définir le potentiel, c'est-à-dire lorsqu'il importe de connaître son amplitude et sa phase. En courant continu, on n'a naturellement besoin que d'une seule information. En alternatif monophasé, la valeur de la phase importe peu (on ne travaille pas sur des grandeurs électriques instantanées) : il suffit de connaître l'amplitude du potentiel et il est donc inutile d'introduire une variable imaginaire.

**La variable dénommée "potentiel réel", $P_R$, représente une valeur efficace si le courant est monophasé et une partie réelle sinon.** De manière plus explicite, pour un potentiel physique

[8] L'intégrale de $cos^2 x$ sur un intervalle de longueur $2\,\pi$ est $\pi$.

alternatif sinusoïdal $Pp$, de valeur maximale notée $Pp_{\max}$, de phase notée $\phi$, la variable $P_R$ représente $\frac{1}{\sqrt{2}}\,Pp_{\max}$ en monophasé et $Pp_{\max}\,cos\phi$ sinon. En courant continu, $P_R$ représente naturellement le potentiel (réel, continu). **Il est donc indispensable de prêter une attention particulière aux valeurs de potentiel imposées aux limites** (facteur $\frac{1}{\sqrt{2}}$ ou $cos\phi$).

## Équations continues

### Système d'équations

Les équations continues qui sont résolues sont les suivantes :

$$\left\{\begin{array}{l} \mathrm{div}(\rho\underline{u}) = 0 \\ \dfrac{\partial}{\partial t}(\rho\underline{u}) + \mathrm{div}(\rho\,\underline{u}\otimes\underline{u}) = \mathrm{div}(\underline{\underline{\sigma}}) + \underline{TS} \\ \dfrac{\partial}{\partial t}(\rho h) + \mathrm{div}(\rho\,\underline{u}h) = \Phi_v - \mathrm{div}\left(\left(\dfrac{\lambda}{C_p} + \dfrac{\mu_t}{\sigma_t}\right)\underline{\mathrm{grad}}\ h\right) + P_J \\ \mathrm{div}(\sigma\,\underline{\mathrm{grad}}\ P_R) = 0 \\ \mathrm{div}(\sigma\,\underline{\mathrm{grad}}\ P_I) = 0 \quad \text{en alternatif non monophasé uniquement} \end{array}\right. \tag{IV.1.41}$$

avec, en continu ou alternatif monophasé :

$$\left\{\begin{array}{l} P_J = \underline{j}\cdot\underline{E} \\ \underline{E} = -\underline{\mathrm{grad}}\ P_R \\ \underline{j} = \sigma\underline{E} \end{array}\right. \tag{IV.1.42}$$

et, en alternatif non monophasé (avec $i^2 = -1$) :

$$\left\{\begin{array}{l} P_J = \dfrac{1}{2}\,\underline{j}\cdot\underline{E}^* \\ \underline{E} = -\underline{\mathrm{grad}}\ (P_R + i\,P_I) \\ \underline{j} = \sigma\underline{E} \end{array}\right. \tag{IV.1.43}$$

### Équation de la masse

C'est l'équation résolue en standard par *Code_Saturne* (contrainte stationnaire d'incompressibilité). Elle n'a pas de traitement particulier dans le cadre du module présent. Un terme source de masse peut être pris en compte au second membre si l'utilisateur le souhaite. Pour simplifier l'exposé, le terme source sera supposé nul ici, dans la mesure où il n'est pas spécifique au module électrique.

### Équation de la quantité de mouvement

C'est l'équation résolue en standard par *Code_Saturne* (les forces de Laplace ($\underline{j}\times\underline{B}$) sont supposées négligeables).

### Équation de l'enthalpie

On l'établit comme dans le cas des arcs électriques[9] à partir de l'équation de l'énergie après plusieurs approximations utilisées en standard dans *Code_Saturne* et en prenant en compte le terme d'effet Joule lié à l'énergie électromagnétique.

Par rapport à l'équation utilisée pour les études d'arc électrique, seule l'expression de l'effet Joule

[9] À ceci près que la puissance des forces de Laplace n'apparaît pas du tout, au lieu de disparaître lorsque l'on soustrait l'équation de l'énergie cinétique à celle de l'énergie totale.

diffère lorsque le courant est alternatif non monophasé.

#### Équations électromagnétiques

Elles sont obtenues comme indiqué dans la partie relative aux arcs électriques, mais on ne conserve que les relations associées à la densité de courant, au champ électrique et au potentiel dont il dérive.

## 1.2 Discrétisation

La discrétisation des équations ne pose pas de problème particulier (ajout de termes sources explicites pour l'effet Joule et les forces de Laplace, équations de Poisson pour la détermination des potentiels).

Un point sur les conditions aux limites doit cependant être fait ici, en particulier pour préciser la méthode de recalage automatique des potentiels.

### 1.2.1 Arcs électriques

#### Conditions aux limites

Seules les conditions aux limites pour les potentiels sont à préciser.

**Les conditions aux limites sur le potentiel scalaire** sont des conditions de Neumann homogènes sur toutes les frontières hormis à la cathode et à l'anode. À la cathode, on impose une condition de Dirichlet homogène (potentiel nul par convention). À l'anode, on impose une condition de Dirichlet permettant de fixer la différence de potentiel souhaitée entre l'anode et la cathode. L'utilisateur peut fixer le potentiel de l'anode directement ou demander qu'un recalage automatique du potentiel soit effectué pour atteindre une intensité de courant prédéterminée.

Lorsque le recalage automatique est demandé (`IELCOR`=1), l'utilisateur doit fixer la valeur cible de l'intensité, `COUIMP`, ($A$) et une valeur élevée de départ de la différence de potentiel entre l'anode et la cathode[10], `DPOT`, ($V$). Le recalage est effectué en fin de pas temps et permet de disposer, pour le pas de temps suivant, de valeurs recalées des forces de Laplace et de l'effet Joule.

- Pour effectuer le recalage, *Code_Saturne* détermine l'intégrale de l'effet Joule estimé sur le domaine (en $W$) et en compare la valeur au produit de l'intensité `COUIMP` par la différence de potentiel[11] `DPOT`. Un coefficient multiplicatif de recalage `COEPOT` en est déduit (pour éviter des variations trop brusques, on s'assure qu'il reste borné).
- On multiplie alors par `COEPOT` la différence de potentiel entre l'anode et la cathode, `DPOT`, et le vecteur $\underline{j}$. L'effet Joule, produit de $\underline{j}$ par $\underline{E}$, est multiplié par le carré de `COEPOT`. Pour assurer la cohérence du post-traitement des variables, le potentiel vecteur et le potentiel scalaire sont également multipliés par `COEPOT`.
- Le champ électrique n'étant pas explicitement stocké, on ne le recale pas. Le potentiel vecteur et les forces de Laplace seront déduits de la densité de courant et intégreront donc naturellement le recalage.

**Les conditions aux limites sur le potentiel vecteur** sont des conditions de Neumann homogène sur toutes les frontières hormis sur une zone de bord arbitrairement choisie (paroi par exemple) pour laquelle une condition de Dirichlet est utilisée afin que le système soit inversible (la valeur imposée est la valeur du potentiel vecteur calculée au pas de temps précédent).

[10] Plus précisément, l'utilisateur doit imposer un potentiel nul en cathode et le potentiel `DPOT` à l'anode, en utilisant explicitement, dans le sous-programme utilisateur `uselcl`, la variable `DPOT` qui sera automatiquement recalée au cours du calcul.

[11] `DPOT` est la différence de potentiel imposée entre l'anode et la cathode au pas de temps qui s'achève. `DPOT` a conditionné le champ électrique et la densité de courant utilisés pour le calcul de l'effet Joule.

### 1.2.2 Effet Joule

#### Conditions aux limites

Seules les conditions aux limites pour les potentiels sont à préciser.

**Les conditions aux limites sur le potentiel scalaire** sont à préciser au cas par cas selon la configuration des électrodes. Ainsi, on dispose classiquement de conditions de Neumann homogènes ou de Dirichlet (potentiel imposé). On peut également avoir besoin d'imposer des conditions d'antisymétrie (en utilisant des conditions de Dirichlet homogènes par exemple). L'utilisateur peut également souhaiter qu'un recalage automatique du potentiel soit effectué pour atteindre une valeur prédéterminée de la puissance dissipée par effet Joule.

Lorsque le recalage automatique est demandé (`IELCOR`=1), l'utilisateur doit fixer la valeur cible de la puissance dissipée dans le domaine, `PUISIM`, ($V.A$). Il doit en outre, sur les frontières où il souhaite que le potentiel (réel ou complexe) s'adapte automatiquement, fournir en condition à la limite une valeur initiale du potentiel et la multiplier par la variable `COEJOU` qui sera automatiquement recalée au cours du calcul (`COEJOU` vaut 1 au premier pas de temps). Le recalage est effectué en fin de pas temps et permet de disposer, pour le pas de temps suivant, d'une valeur recalée de l'effet Joule.

- Pour effectuer le recalage, *Code_Saturne* détermine l'intégrale de l'effet Joule estimé sur le domaine (en $W$) et en compare la valeur à la puissance cible. Un coefficient multiplicatif de recalage `COEPOT` en est déduit (pour éviter des variations trop brusques, on s'assure qu'il reste borné entre 0,75 et 1,5).
- On multiplie alors par `COEPOT` le facteur multiplicatif `COEJOU` utilisé pour les conditions aux limites. La puissance dissipée par effet Joule est multipliée par le carré de `COEPOT`. Pour assurer la cohérence du post-traitement des variables, le potentiel est également multiplié par `COEPOT`.
- Le champ électrique n'étant pas explicitement stocké, on ne le recale pas.

On notera que la variable `DPOT` est également recalée et qu'elle peut donc être utilisée si besoin pour imposer les conditions aux limites.

## 1.3 Mise en œuvre

### 1.3.1 Introduction

Le module électrique est une "physique particulière" activée lorsque les mots-clés `IPPMOD(IELARC)` (arc électrique) ou `IPPMOD(IELJOU)` (Joule) sont strictement positifs. Les développements concernant la conduction ionique (mot-clé `IPPMOD(IELION)`) ont été prévus dans le code mais restent à réaliser. Pour l'arc électrique, dans la version actuelle de *Code_Saturne*, seule est opérationnelle l'option `IPPMOD(IELARC)`=2 : la version 2D axisymétrique qui permettrait de s'affranchir du potentiel vecteur (option `IPPMOD(IELARC)`=1) n'est pas activable. Pour l'effet Joule, lorsqu'il n'est pas utile d'introduire un potentiel scalaire complexe (en courant continu ou alternatif monophasé), on utilise `IPPMOD(IELJOU)`=1. Lorsqu'un potentiel scalaire complexe est indispensable (courant alternatif triphasé, par exemple), on utilise `IPPMOD(IELJOU)`=2.

Dans ce qui suit, on précise les inconnues et les propriétés principales utilisées dans le module. On fournit également un arbre d'appel simplifié des sous-programmes du module (initialisation avec `initi1` puis `inivar` et boucle en temps avec `tridim`). Les sous-programmes marqués d'un astérisque sont détaillés ensuite.

### 1.3.2 Inconnues et propriétés

Les développements ont été réalisés pour une unique phase (NPHAS=1).

Les NSCAPP inconnues scalaires associées à la physique particulière sont définies dans elvarp dans l'ordre suivant (en particulier afin de limiter le stockage en mémoire lors de la résolution séquentielle des scalaires par scalai) :

- l'enthalpie RTP(*,ISCA(IHM)),
- un potentiel scalaire réel RTP(*,ISCA(IPOTR)),
- un potentiel scalaire imaginaire RTP(*,ISCA(IPOTI)) *ssi* IPPMOD(IELJOU)=2 (études Joule en courant alternatif non monophasé),
- les trois composantes d'un potentiel vecteur réel RTP(*,ISCA(IPOTVA(i))) (avec i variant de 1 à 3) *ssi* IPPMOD(IELARC)=2 (arc électrique),
- NGAZG-1 fractions massiques RTP(*,ISCA(IYCOEL(j))) (avec j variant de 1 à NGAZG-1) pour un fluide à NGAZG constituants (avec NGAZG strictement supérieur à 1). En arc électrique, la composition est fournie dans le fichier de données dp_ELE. La fraction massique du dernier constituant n'est pas stockée en mémoire. Elle est déterminée chaque fois que nécessaire en calculant le complément à l'unité des autres fractions massiques (et, en particulier, lorsque elthht est utilisé pour le calcul des propriétés physiques).

Outre les propriétés associées en standard aux variables scalaires identifiées ci-dessus, le tableau PROPCE contient également :

- la température, PROPCE(*,IPPROC(ITEMP)). En théorie, on pourrait éviter de stocker cette variable, mais l'utilisateur est presque toujours intéressé par sa valeur en post-traitement et les propriétés physiques sont souvent données par des lois qui en dépendent explicitement. Son unité (Kelvin ou Celsius) dépend des tables enthalpie-température fournies par l'utilisateur.
- la puissance électromagnétique dissipée par effet Joule, PROPCE(*,IPPROC(IEFJOU)) (terme source positif pour l'enthalpie),
- les trois composantes des forces de Laplace, PROPCE(*,IPPROC(ILAPLA(i))) (avec i variant de 1 à 3) en arc électrique (IPPMOD(IELARC)=2).

La conductivité électrique est *a priori* variable et conservée dans le tableau de propriétés aux cellules PROPCE(*,IPPROC(IVISLS(IPOTR))). Elle intervient dans l'équation de Poisson portant sur le potentiel scalaire. Lorsque le potentiel scalaire a une partie imaginaire, la conductivité n'est pas dupliquée : les entiers IPPROC(IVISLS(IPOTI)) et IPPROC(IVISLS(IPOTR)) pointent sur la même case du tableau PROPCE. La conductivité associée au potentiel vecteur est uniforme et de valeur unité (VISLS0(IPOTVA(i))=1.D0 avec i variant de 1 à 3).

Le champ électrique, la densité de courant et le champ magnétique ne sont stockés que de manière temporaire (voir elflux).

### 1.3.3 Arbre d'appel simplifié

| | | | |
|---|---|---|---|
| `usini1` | | | Initialisation des mots-clés utilisateur généraux et positionnement des variables |
| | `usppmo` | | Définition du module "physique particulière" employé |
| | `varpos` | | Positionnement des variables |
| | | `pplecd` | Branchement des physiques particulières pour la lecture de fichier de données |
| | | `ellecd`* | Lecture du fichier de données pour les arcs électriques `dp_ELE` |
| | | `ppvarp` | Branchement des physiques particulières pour le positionnement des inconnues |
| | | `elvarp`* | Positionnement des inconnues (enthalpie, potentiels, fractions massiques) |
| | | `ppprop` | Branchement des physiques particulières pour le positionnement des propriétés |
| | | `elprop`* | Positionnement des propriétés (température, effet Joule, forces de Laplace) |
| `ppini1` | | | Branchement des physiques particulières pour l'initialisation des mots-clés spécifiques |
| | `elini1` | | Initialisation des mots-clés pour le module électrique |
| | `useli1` | | Initialisation des mots-clés utilisateur pour le module électrique |
| | `elveri` | | Vérification des mots-clés pour le module électrique |

Table 28.1: Sous-programme `initi1` : initialisation des mots-clés et positionnement des variables

| | | | |
|---|---|---|---|
| `ppiniv` | | | Branchement des physiques particulières pour l'initialisation des variables |
| | `eliniv`* | | Initialisation des variables spécifiques au module électrique |
| | | `elthht`* | Transformation température-enthalpie et enthalpie-température par interpolation sur la base du fichier de données `dp_ELE` (arc électrique uniquement) |
| | | `useliv` | Initialisation des variables par l'utilisateur |
| | | `elthht`* | Transformation température-enthalpie et enthalpie-température par interpolation sur la base du fichier de données `dp_ELE` (arc électrique uniquement) |

Table 28.2: Sous-programme `inivar` : initialisation des variables

| | | | |
|---|---|---|---|
| `phyvar` | | | Calcul des propriétés physiques variables |
| | `ppphyv` | | Branchement des physiques particulières pour le calcul des propriétés physiques variables |
| | `elphyv` | | Calcul des propriétés physiques variables pour le module électrique. En arc électrique, les propriétés sont calculées par interpolation à partir des tables fournies dans le fichier de données `dp_ELE` |
| | | `elthht`* | Transformation température-enthalpie et enthalpie-température par interpolation sur la base du fichier de données `dp_ELE` (arc électrique uniquement) |
| | | `uselph` | Calcul par l'utilisateur des propriétés physiques variables pour le module électrique. Pour les études Joule, en particulier, les propriétés doivent être fournies ici sous forme de loi (des exemples sont disponibles) |
| | | `usthht` | Transformation température-enthalpie et enthalpie-température fournie par l'utilisateur (plus spécifiquement pour les études Joule, pour lesquelles on ne dispose pas d'un fichier de données à partir duquel réaliser des interpolations avec `elthht`) |

Table 28.3: Sous-programme `tridim` : partie 1 (propriétés physiques)

## 1.3.4 Précisions

• `ellecd`

Ce sous-programme réalise la lecture du fichier de données spécifique aux arcs électriques. On donne ci-dessous, à titre d'exemple, l'entête explicative et deux lignes de données d'un fichier type. Ces valeurs sont interpolées chaque fois que nécessaire par `elthht` pour déterminer les propriétés physiques du fluide à une température (une enthalpie) donnée.

```
# Nb d'especes NGAZG et Nb de points NPO (le fichier contient NGAZG blocs de NPO lignes chacun)
# NGAZG NPO
   1   238
#
#  Proprietes
#      T           H          ROEL       CPEL          SIGEL        VISEL        XLABEL        XKABEL
#  Temperature  Enthalpie   Masse vol. Chaleur       Conductivite  Viscosite   Conductivite   Coefficient
#                           volumique  massique       electrique   dynamique     thermique   d'absorption
#       K         J/kg         kg/m3     J/(kg K)       Ohm/m        kg/(m s)     W/(m K)         -
#
   300.00       14000.       1.6225       520.33      0.13214E-03  0.34224E-04  0.26712E-01   0.0000
   400.00       65800.       1.2169       520.33      0.13214E-03  0.34224E-04  0.26712E-01   0.0000
```

• `elvarp`

Ce sous-programme permet de positionner les inconnues de calcul listées précédemment. On y précise également que la chaleur massique à pression constante est variable, ainsi que la conductivité de tous les scalaires associés au module électrique, hormis la conductivité de l'éventuel potentiel vecteur (celle-ci est uniforme et de valeur unité).

• `elprop`

C'est dans ce sous-programme que sont positionnées les propriétés stockées dans le tableau `PROPCE`, et en particulier la température, l'effet Joule et les forces de Laplace.

| | | |
|---|---|---|
| `ppclim` | | Branchement des physiques particulières pour les conditions aux limites |
| | `uselcl` | Intervention de l'utilisateur pour les conditions aux limites (en lieu et place de `usclim`, même pour les variables qui ne sont pas spécifiques au module électrique). Si un recalage automatique des potentiels est demandé (`IELCOR=1`), il doit être pris en compte ici par le biais des variables `DPOT` ou `COEJOU` (voir la description des conditions aux limites). |
| `navsto` | | Résolution des équations de Navier-Stokes |
| | `preduv` | Prédiction de la vitesse : prise en compte des forces de Laplace calculées dans `elflux` au pas de temps précédent |
| `''turb''` | | Turbulence : résolution des équations pour les modèles nécessitant des équations de convection-diffusion |
| `scalai`* | | Résolution des équations portant sur les scalaires associés aux physiques particulières et des scalaires "utilisateur" |
| | `covofi` | Résolution successive de l'enthalpie, du potentiel scalaire réel et, si `IPPMOD(IELJOU)=2`, de la partie imaginaire du potentiel scalaire (appels successifs à `covofi` qui appelle `eltssc` pour le calcul du terme d'effet Joule au second membre de l'équation de l'enthalpie) |
| | `elflux`* | Calcul du champ électrique, de la densité de courant et de l'effet Joule (premier de deux appels au cours du pas de temps courant) |
| | `uselrc`* | Recalage automatique éventuel de la densité de courant, de l'effet Joule, des potentiels et des coefficients `DPOT` et `COEJOU`. Ce recalage, s'il a été demandé par l'utilisateur (`IELCOR`=1), est effectué à partir du deuxième pas de temps. |
| | `covofi` | Résolution successive, si `IPPMOD(IELARC)=2`, des trois composantes du potentiel vecteur. On procède par appels successifs à `covofi` qui appelle `eltssc` pour le calcul du second membre de l'équation de Poisson portant sur chaque composante du potentiel. |
| | `covofi` | Résolution successive des `NGAZG-1` fractions massiques caractérisant la composition du fluide, s'il est multiconstituant. On procède par appels successifs à `covofi`. |
| | `elflux`* | En arc électrique, calcul du champ magnétique et des trois composantes des forces de Laplace (deuxième et dernier appel au cours du pas de temps courant) |
| | `covofi` | Résolution des scalaires "utilisateur" |

Table 28.4: Sous-programme `tridim` : partie 2 (conditions aux limites, Navier-Stokes, turbulence et scalaires)

| | | |
|---|---|---|
| `postlc` | | Post-traitement |
| | `ecrevo` | Écriture des variables à post-traiter |
| | `uselen` | Ajout au post-traitement de variables calculées par l'utilisateur. En exemple activé standard sont post-traités, s'ils existent, l'opposé du champ électrique (*i.e.* le gradient du potentiel scalaire, réel ou complexe), le vecteur densité de courant imaginaire (en effet Joule), le champ magnétique (en arc électrique) et enfin le module et l'argument du potentiel (en effet Joule, avec `IPPMOD(IELJOU)=4`) |

Table 28.5: Sous-programme `tridim` : partie 3 (post-traitement)

• `eliniv`

Ce sous-programme permet de réaliser les initialisations par défaut spécifiques au module.

En particulier, en $k-\varepsilon$, les deux variables turbulentes sont initialisées à $10^{-10}$ (choix historique arbitraire, mais réputé, lors de tests non référencés, permettre le démarrage de certains calculs qui échouaient avec une initialisation classique).

Les potentiels sont initialisés à zéro, de même que l'effet Joule. En arc électrique, les forces de Laplace sont initialisées à zéro.

Le fluide est supposé monoconstituant (seule est présente la première espèce).

En arc électrique, l'enthalpie est initialisée à la valeur de l'enthalpie du mélange supposé monoconstituant à la température `T0(IPHAS)` donnée dans `usini1`. En effet Joule, l'enthalpie est initialisée à zéro (mais l'utilisateur peut fournir une valeur différente dans `useliv`).

• `elthht`

Ce sous-programme permet de réaliser (en arc électrique) les interpolations nécessaires à la détermination des propriétés physiques du fluide, à partir des tables fournies dans le fichier de données `dp_ELE`.

On notera en particulier que ce sous-programme prend en argument le tableau `YESP(NESP)` qui représente la fraction massique des `NGAZG` constituants du fluide. Dans le code, on ne résout que la fraction massique des `NGAZG`-1 premiers constituants. Avant chaque appel à `elthht`, la fraction massique du dernier constituant doit être calculée comme le complément à l'unité des autres fractions massiques.

• `scalai`, `elflux`, `uselrc`

Le sous-programme `scalai` permet de calculer, dans l'ordre souhaité, les `NSCAPP` scalaires "physique particulière" associés au module électrique, puis de calculer les grandeurs intermédiaires nécessaires et enfin de réaliser les opérations qui permettent d'assurer le recalage automatique des potentiels, lorsqu'il est requis par l'utilisateur (*i.e.* si `IELCOR=1`).

Les `NSCAPP` scalaires "physique particulière" sont calculés successivement par un appel à `covofi` placé dans une boucle portant sur les `NSCAPP` scalaires. L'algorithme tire profit de l'ordre spécifique dans lequel ils sont définis et donc résolus (dans l'ordre : enthalpie, potentiel scalaire, potentiel vecteur, fractions massiques).

Pour éviter des variations trop brutales en début de calcul, le terme source d'effet Joule n'est pris en compte dans l'équation de l'enthalpie qu'à partir du troisième pas de temps.

Après la résolution de l'enthalpie et du potentiel scalaire (réel ou complexe), le sous-programme `elflux` permet de calculer les trois composantes du champ électrique (que l'on stocke dans des tableaux de travail), puis la densité de courant et enfin l'effet Joule, que l'on conserve dans le tableau `PROPCE(*,IPPROC(IEFJOU))` pour le pas temps suivant (après recalage éventuel dans `uselrc` comme indiqué ci-après). Lorsque `IPPMOD(IELJOU)=2`, l'apport de la partie imaginaire est pris en compte pour le calcul de l'effet Joule. Lorsque `IPPMOD(IELARC)=2` (arc électrique), le vecteur densité de courant est conservé dans `PROPCE`, en lieu et place des forces de Laplace `PROPCE(*,IPPROC(ILAPLA(i)))` : il est utilisé pour le calcul du potentiel vecteur dans le second appel à `elflux`, après recalage éventuel par `uselrc` (en effet, il n'est plus nécessaire de conserver les forces de Laplace à ce stade puisque la seule équation dans laquelle elles interviennent est l'équation de la quantité de mouvement et qu'elle a déjà été résolue).

À la suite de `elflux`, le sous-programme `uselrc` effectue le recalage permettant d'adapter automa-

tiquement les conditions aux limites portant sur les potentiels, si l'utilisateur l'a demandé (*i.e.* si `IELCOR=1`). On se reportera au paragraphe relatif aux conditions aux limites. On précise ici que le coefficient de recalage `COEPOT` permet d'adapter l'effet Joule `PROPCE(*,IPPROC(IEFJOU))` et la différence de potentiel `DPOT` (utile pour les conditions aux limites portant sur le potentiel scalaire au pas de temps suivant[12]). Pour les cas d'arc électrique, `COEPOT` permet également de recaler le vecteur densité de courant que l'on vient de stocker temporairement dans `PROPCE(*,IPPROC(ILAPLA(i)))` et qui va servir immédiatement à calculer le potentiel vecteur. Pour les cas Joule, on recale en outre le coefficient `COEJOU` (utile pour les conditions aux limites portant sur le potentiel scalaire au pas de temps suivant).

Pour les cas d'arc électrique (`IPPMOD(IELARC)=2`), après `elflux` et `uselrc`, la résolution séquentielle des inconnues scalaires se poursuit dans `scalai` avec le calcul des trois composantes du potentiel vecteur. Le second membre de l'équation de Poisson considérée dépend de la densité de courant qui, dans `elflux`, a été temporairement stockée dans le tableau `PROPCE(*,IPPROC(ILAPLA(i)))` et qui, dans `uselrc`, vient d'être recalée si `IELCOR=1`. Les valeurs du potentiel vecteur obtenues intègrent donc naturellement l'éventuel recalage.

Pour les cas d'arc électrique (`IPPMOD(IELARC)=2`), un second appel à `elflux` permet alors de calculer le champ magnétique que l'on stocke dans des tableaux de travail et les forces de Laplace que l'on stocke dans `PROPCE(*,IPPROC(ILAPLA(i)))` pour le pas de temps suivant (la densité de courant, que l'on avait temporairement conservée dans ce tableau, ne servait qu'à calculer le second membre de l'équation de Poisson portant sur le potentiel vecteur : il n'est donc plus nécessaire de la conserver).

La résolution séquentielle des inconnues scalaires spécifiques au module se poursuit dans `scalai`, avec le calcul des `NGAZG`-1 fractions massiques permettant de définir la composition du fluide.

Pour terminer, `scalai` permet la résolution des scalaires "utilisateurs" (appel à `covofi` dans une boucle portant sur les `NSCAUS` scalaires utilisateur).

On peut remarquer pour finir que les termes sources des équations de la quantité de mouvement (forces de Laplace) et de l'enthalpie (effet Joule) sont disponibles à la fin du pas de temps $n$ pour une utilisation au pas de temps $n+1$ (de ce fait, pour permettre les reprises de calcul, ces termes sources sont stockés dans le fichier suite auxiliaire, ainsi que `DPOT` et `COEJOU`).

[12] *A priori*, `DPOT` n'est pas nécessaire pour les cas Joule.

## 1.4 Points à traiter

• **Mobilité ionique**
Le module est à développer.

• **Conditions aux limite en Joule**
La prise en compte de conditions aux limites couplées entre électrodes reste à faire.

• **Compressible en arc électrique**
Les développements du module compressible de *Code_Saturne* doivent être rendus compatibles avec le module arc électrique.

# Part V

# Module combustion

# 1-
# Sous-programme co**, cp**, fu** and so

## 1.1 Fonction

From a CFD point of vue combustion is a (somestimes very) complicated way to determine $\rho$. Models needs few extra fields of scalar with regular transport equation, somes of them with a reactive or interfacial source term.

Modeling of combustion is able to deal with gaz phase combustion (diffusion, premix, partial premix), and with solid or liquid fuels.

Combustion of condensed fuels involves one-way interfacial flux due to phenomenon in the condensed phase (evaporation or pyrolisis) and reciprocal ones (heterogeneous combustion). Many of the species injected in the gaz phase are afterwards involved in gaz phase combsution.

That is the reason why a lot of modules are similar for gaz, coal and fuel combusiton modelling. Obviously, the thermodynamical description of gaz species is similar in every version as close as possible of the JANAF rules.

Every models are developped in both an adiabatic version and an undiabatic (permeatic) one, so in addition with standard, the rule to call models is : IPPMOd(index model) = -1 unused IPPMOD(index model) = 0 simplest adiabatic version IPPMOD(index model) = 1 simplest permeatic version Eventually IPPMOD(index model) = 2.p p° adiabatic version IPPMOD(index model) = 2.p+1 P° permeatic version

Every permeatic version involves the transport of enthalpy (one more variable).

### 1.1.1 Gaz combustion modelling

Combustion of gaz is limited by disponibility (in the same fluid particle) of both fuel and oxidant and by kinetic effects (a lot of chemical reactions involved can be described using an Arrhenius law with an high activation energy). The mixing of mass (atoms) incoming with fuel and oxydant is described by a mixture fraction (mass fraction of mass incoming with fuel), this variable is not affected by combustion. A progress variabl is used to describe the transformation of the mixture from fuel and oxydant to product (carbon dioxyde and so on).

Combsution of gaz is, traditionnaly, splitted in premix and diffusion. In premix combustion process a first stage of mixing have been realised (without blast ...) and the mixture is introduced in the boiler (or combustion can). In industrial common conditions the combustion is mainly limited by the mixing of fresh gaz (inert) and burnt ones resulting in the inflammation of the first and their conversion to burnt ones ; so an assumption of chemistry much faster than mixing induces an intermittent regime. The gaz flow is constituted of totally fresh and totally burnt gaz (the flamme containing the gaz during their transformation) is "extremuly" thin. With previous assumptions, Spalding have established the "Eddy Break Up" model, wich allows a complete descrition with only one progress variable (mixture fraction is homogeneous). In diffusion flame the fuel and the oxydant are introduced by two (at least) inlets, in common industiral conditions, their mixing is the main limitation and the mixture fraction is enough to qualify a fluid particle, but in turbulent flow a probability density function of the mixture fraction is needed to qualify the thermodynamical state of the bulk. So both the mean and the variance of the mixture fraction are needed (two variables).

In the real world the chemistry is not so fast and, often, the mixing is not as homogeneous as wished. Then the industrial combustion occurs in partial premix combsution. Partial premix occurs if the mixing is not finished (at molecular level) when the mixing is introduced, or if air, or fuel, are staggered,

or if a diffusion flame is blowed off. For these situations, and specifically for lean premix gaz turbines Libby & Williams have developped a model allowing a descirption of both mixing and chemical limitations. A collaboration between the LCD Poitiers and EDF R&D allows a simpler version of their algorithm. Not only the mean and the variance of both mixture fraction and progress variable are needed but so the covariance (five variables).

### 1.1.2 Coal combustion modelling

Combustion of coal is the main way to produce electricity in the world. The coal is a natural product with a very complex composition, during the industrial process of milling the raw coal is broken in tiny particles of different sizes. After its introduction in the boiler, the caol particles undergoes drying, devolatilisation (heating of coal turn it in a mixture of char and gases), heterogenous combustion (of char leaving to carbone monoxide) leaving an ash particle. Each of therse phenomena are taken in account for some class of particles : a class is caracterised by a coal (it is useful to burn mixture of coals with differents ranks or mixture of coal with biomasse ...) and an initla diameter. For each class, *Code_Saturne* computes the number and the mass of particles by unit mass of mixture. The main assumption is to solve only one speed (and pressure) field : it means the discrepancy of speed between coal particles and gases is supposed negligible. Due to the radiation and heterogeneous combustion, temperature can be different for gas and different size particles : so the specific enthalpy of each particle class is solved. The descripton of coal pyrolisis proposed by Kobayashi and bhayakar is used, leaving at two source terms for light and heavy volatile matters (the moderate temperature reaction produces gases with low molecular mass, the high temperature reaction produces heavier gases and less char) represented by passive scalar : mixture fraction. The description of heterogeneous reaction (who produce carbon monoxide) leads to a source term for the carbon : a mixture fraction who can't be greater than the results of stoechiometric oxydation of char by air. The retained model for the gaz phase combusiton is diffusion flammelets surrounding each particle, so the previous gaseous fuels are mixed in a local mean fuel and the mixing with air is represented by a pdf between air and mean local fuel constructed with the variance of a passive scalar linked with air (interfacial mass flux produce a source term for this scalar).

### 1.1.3 Heavy Fuel Oil combustion modelling

Combustion of heavy fuel oil have been hugely used to produced electrical energy. Environemental regulation turns it more difficult and less acceptable, a focus is needed on pollutant emission mainly soulphur oxide and particles (condensation of sulphuric acid can aggregate soot). The description of fuel evaporation is done with respect of its heaviness : after a minimum temperature is reached, the gain of enthalpy is distributed between heating and evaporation. This way the evaporation takes place on a range of temperature (which can be large). The "total" evaporation is common for light oil but impossible for heavy ones, so a particle similar to char is leaved ; the heterogeneous oxydation of this char particle is very similar to coal char ones. Injection of fuel is described (2006 version) with only one class of particle, the number and mass of particles is calculated eveywhere. And so the enthalpy. So three variables are used to describe the condensed pahse. Like for coal, only one speed field is solved. The model for gas combustion is very similar to coal ones but a special is paid to sulphur (assumed to leave the particle as H2S during evaporation and to be converted to SO2 during gas combustion).