Code Saturne: A Finite Volume Code for the computation of turbulent incompressible flows - Industrial Applications

Frédéric Archambeau, Namane Méchitoua, Marc Sakiz

Code_Saturne: a Finite Volume Code for the Computation of Turbulent Incompressible Flows - Industrial Applications

Frédéric Archambeau, Namane Méchitoua, Marc Sakiz
EDF R&D, Fluid Mechanics and Heat Transfer Department
6, Quai Watier, 78400, Chatou, France
frederic.archambeau@edf.fr, namane.mechitoua@edf.fr, marc.sakiz@edf.fr

Abstract

This paper describes the finite volume method implemented in *Code_Saturne*®, Électricité de France general-purpose computational fluid dynamic code for laminar and turbulent flows in complex two- and three-dimensional geometries. The code is used for industrial applications and research activities in several fields related to energy production (nuclear power thermal-hydraulics, gas and coal combustion, turbomachinery, heating, ventilation and air conditioning...).
The set of equations considered consists of the Navier-Stokes equations for incompressible flows completed with equations for turbulence modelling (eddy-viscosity model and second moment closure) and for additional scalars (temperature, enthalpy, concentration of species, ...). The time-marching scheme is based on a prediction of velocity followed by a pressure correction step. Equations for turbulence and scalars are resolved separately afterwards. The discretization in space is based on the fully conservative, unstructured finite volume framework, with a fully co-located arrangement for all variables. Specific effort has been put into the computation of gradients at cell centres.
Industrial applications illustrate important aspects of physical modelling such as turbulence (using Reynolds-Averaged Navier-Stokes equations or Large Eddy Simulation), combustion, conjugate heat transfer (coupled with the thermal code SYRTHES) and fluid-particle coupling with a lagrangian approach. These examples also demonstrate the capability of the code to tackle a large variety of meshes and cell geometries, including hybrid meshes with arbitrary interfaces.

Key words : Navier-Stokes, finite volume, unstructured mesh, co-located arrangement, gradient calculation, turbulent flows, incompressible flows, Reynolds-Averaged Navier-Stokes equations, Large Eddy Simulation, parallel computing, nuclear power, gas and coal combustion, *Code_Saturne*

1 Introduction

Over the past decade, Computational Fluid Dynamics (C.F.D.) has become an increasingly standard industrial simulation tool for design, performance improvement and analysis of operating conditions for environmental and safety reasons. Such an evolution has been made possible by factors such as the massive increase in computer capacities and the improvement of in-house and commercial C.F.D. software.

At Électricité de France (EDF), development of in-house codes has been a resolute strategic choice for more than fifteen years. In particular, for problems requiring local three-dimensional analyses with refined flow modelling, specific effort had been put into in-house "general purpose C.F.D. codes" such as N3S-EF [CHA 92] and ESTET-ASTRID [MAT 92]. In 1996, EDF initiated a program to unify the potentialities of these two products within the same software, *Code_Saturne*®. Indeed, N3S-EF, an unstructured finite element code, could tackle complex geometries, while ESTET-ASTRID, a structured finite volume code, provided refined physical modelling. In addition to making available to users the most advanced capabilities of these two complementary products, it was also a convenient way to benefit from the recent advances in design, software development, programming and meshing techniques, numerical schemes and physical modelling.

Code_Saturne is well suited for two- and three-dimensional calculations of steady or transient single-phase, incompressible, laminar or turbulent flows. It supports two Reynolds-Averaged Navier-Stokes (R.A.N.S.) models: a standard $k-\varepsilon$ and a second moment closure [LRR 75]. The flow-solver is based on a finite volume approach, with a fully co-located arrangement for all variables. The time discretization is based on a predictor-corrector scheme for the Navier-Stokes equations. An important asset of *Code_Saturne* relies in its ability to deal with any kind of mesh (hybrid, containing arbitrary interfaces[1] and any type of cell). Data management is ensured by an "Envelope module" that also allows communications with different mesh-generators, post-processors and other software when coupling is required (for example, conjugate heat transfer is routine calculation through coupling with the thermal code SYRTHES [PEN 97] [RUP 99]).

Code_Saturne is developed by EDF under quality assurance and is used at EDF in various industrial fields such as nuclear applications, process engineering (plasma, electric arcs, glass furnace), aeraulics (heating, ventilation and air conditioning, pollutant dispersion, contamination in medical premises...), and combustion (gas and coal furnaces). *Code_Saturne* 1.0 has been validated and released early in 2001. Efforts were then invested during one year into a qualification[2] process for nuclear

[1] "Arbitrary interfaces" refers to meshes in which two neighbouring control volumes do not necessarily share a whole face and may not have common edges or vertices.

[2] "Qualification" is employed here in the sense of validation on a narrow and specific range of real-life industrial applications with attention devoted to the method employed to conduct studies, to the quality of the results obtained and to their limitations with respect to the anticipated engineering use.

single-phase thermal-hydraulics. The next release of *Code_Saturne* is due by 2004.

This article presents the continuous equations, their discretization in time and space and the associated boundary conditions. An overview of several studies is proposed to illustrate some industrial applications that are being carried out with the code (ventilation study in hospitals, slagging in a coal furnace, pollutant prediction in a gas turbine, qualification for thermal schock in a pressure water reactor vessel, Large Eddy Simulation applied to thermal fatigue). Eventually, perspectives for future development are briefly put forward.

2 Nomenclature

2.1 Preliminary remarks

Let $\underline{u}$ and $\underline{v}$ be vectors and $\underline{\underline{\sigma}}$ a second order tensor. Their respective components are denoted u_i, v_i and σ_{ij}. In this paper, the following notations will be used:

$$\left\{ \begin{array}{lcl} \left[\underline{\underline{\mathrm{grad}}}\ \underline{u}\right]_{ij} & = & \dfrac{\partial u_i}{\partial x_j} \\ \left[\mathrm{div}(\underline{\underline{\sigma}})\right]_i & = & \dfrac{\partial \sigma_{ij}}{\partial x_j} \\ \left[\underline{u} \otimes \underline{v}\right]_{ij} & = & u_i\, v_j \end{array} \right. \tag{1}$$

With these notations, we have:

$$\left[\mathrm{div}(\underline{u} \otimes \underline{v})\right]_i = \frac{\partial (u_i\, v_j)}{\partial x_j}$$

The cross product of vectors $\underline{u}$ and $\underline{v}$ is noted $\underline{u} \times \underline{v}$.

2.2 Subscripts

i, j, k	refers either to vector/tensor components or to a cell number

2.3 Superscripts

t	transpose
(n)	n^{th} time step

2.4 List of symbols

C_p	specific heat
$\underline{\underline{Id}}$	identity matrix
$Neibrs(i)$	set of cells j sharing at least an interface (i,j) with cell i
P	pressure
$\underline{\underline{R}}$	Reynolds stress tensor
R_{ij}	ij^{th} component for $\underline{\underline{R}}$

S_{ij}	surface of interface (i,j) common to cells i and j (positive value)
S_ϕ	source term related to transported quantity ϕ
a	scalar quantity
a_i	discrete value of a associated with cell i
k	turbulent kinetic energy
$\underline{n}_{ij}$	unit vector normal to interface (i,j) common to cells i and j and oriented from cell i towards cell j
t	time
$\underline{u}$	velocity vector
u_i	i^{th} component for $\underline{u}$
Γ	mass source term
Ω_i	volume of cell i (of measure $\|\Omega_i\|$)
δ_{ij}	Kronecker delta
ε	dissipation rate of turbulent kinetic energy
λ	thermal conductivity
μ	dynamic molecular viscosity
ν	kinematic molecular viscosity
ρ	density
σ	molecular Prandtl-Schmidt number
$\underline{\underline{\sigma}}$	stress tensor
σ_t	turbulent Prandtl-Schmidt number
$\underline{\underline{\tau}}$	viscous stress tensor
$\overline{\phi}$	Reynolds average of quantity ϕ
$\tilde{\phi}$	Favre average of quantity ϕ
ϕ'	fluctuating part related to the Reynolds decomposition of ϕ
ϕ''	fluctuating part related to the Favre decomposition of ϕ

3 Continuous equations

3.1 Conservation laws

We consider the following system of continuous equations for mass and momentum [FER 99]:

$$\left\{ \begin{array}{l} \dfrac{\partial \rho}{\partial t} + \mathrm{div}(\rho\,\underline{u}) = \Gamma \\ \\ \dfrac{\partial}{\partial t}(\rho \underline{u}) + \mathrm{div}(\rho\,\underline{u} \otimes \underline{u}) = \mathrm{div}(\underline{\underline{\sigma}}) + \underline{S}_{\underline{u}} \end{array} \right. \tag{2}$$

In the above equations (2), ρ is the fluid density and $\underline{u}$ the velocity. Γ and $\underline{S}_{\underline{u}}$ represent source terms ($\underline{S}_{\underline{u}}$ also includes the momentum source issuing from the mass source term Γ). $\underline{\underline{\sigma}}$ stands for the stress tensor, containing the effect of pressure

P, and viscous stress $\underline{\underline{\tau}}$. Assuming $\underline{\underline{\sigma}}$ is a symmetric tensor, we have, for Newtonian fluids (to which we restrict the present considerations):

$$\begin{cases} \underline{\underline{\sigma}} = \underline{\underline{\tau}} - P\underline{\underline{Id}} \\ \quad \text{with} \quad \underline{\underline{\tau}} = 2\,\mu\ \underline{\underline{D}} - \lambda^v\, tr(\underline{\underline{D}})\, \underline{\underline{Id}} \\ \qquad\quad \text{and } \underline{\underline{D}} = \dfrac{1}{2}(\underline{\underline{\text{grad}}}\ \underline{u} + {}^t\underline{\underline{\text{grad}}}\ \underline{u}) \end{cases} \tag{3}$$

In equation (3), μ represents the dynamic molecular viscosity and $\lambda^v = \dfrac{2}{3}\mu$ is the bulk viscosity. $\underline{\underline{Id}}$ is the identity matrix.

In addition to mass and momentum equations, we consider the following equation for any intensive scalar property a:

$$\frac{\partial}{\partial t}(\rho a) + \text{div}\,((\rho \underline{u})\, a) = \text{div}(\underline{J_a}) + S_a \tag{4}$$

where $\underline{J_a}$ is the vector flux of scalar, which, using the Fick-Fourier law can usually be expressed as:

$$\underline{J_a} = K_a \underline{\text{grad}}\ a \tag{5}$$

with K_a standing for the molecular diffusivity pertaining to a.

3.2 Favre averaged equations

In case of turbulence, the equations have to be averaged (ensemble average). For a quantity ϕ, the Favre average [FAV 76] will be denoted $\tilde{\phi}$ and the Reynolds average $\overline{\phi}$. Fluctuating parts will be denoted ϕ'' and ϕ' respectively. We have the following relation:

$$\overline{\rho\phi} = \overline{\rho}\tilde{\phi} \tag{6}$$

The Favre decomposition of variables in mean and fluctuating parts is applied to velocity $\underline{u}$ and scalars a. With ϕ standing for any of these variables, we have:

$$\phi = \tilde{\phi} + \phi'' \tag{7}$$

the Reynolds decomposition is applied to density and pressure:

$$P = \overline{P} + P' \qquad \text{and} \qquad \rho = \overline{\rho} + \rho' \tag{8}$$

With these notations, the statistical approach to turbulence applied to (2) and (4) produces equations formally identical to their parents, except for the appearance of the turbulence correlations $\overline{\rho}\underline{\underline{R}} = \overline{\rho}\widetilde{\underline{u}'' \otimes \underline{u}''} = \overline{\rho \underline{u}'' \otimes \underline{u}''}$ and $\overline{\rho}\widetilde{a''\underline{u}''} = \overline{\rho a'' \underline{u}''}$. With constant molecular fluid properties (K, μ) and omitting the overline for brevity,

except for turbulent scalar flux, the following system is obtained:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \mathrm{div}(\rho\, \underline{u}) = \Gamma \\ \dfrac{\partial}{\partial t}(\rho \underline{u}) + \mathrm{div}(\rho\, \underline{u} \otimes \underline{u}) = -\underline{\mathrm{grad}}\ (P) + \mathrm{div}(\underline{\underline{\tau}} - \rho \underline{\underline{R}}) + \underline{S}_{\underline{u}} \\ \dfrac{\partial}{\partial t}(\rho a) + \mathrm{div}\left((\rho \underline{u})\, a\right) = \mathrm{div}(\underline{J}_a - \overline{\rho a'' \underline{u}''}) + S_a \end{cases} \tag{9}$$

The mass equation is used to exhibit the time derivative of the variables $\underline{u}$ and a. With ϕ standing for any of these variables, we have:

$$\begin{aligned} \frac{\partial}{\partial t}(\rho\phi) &= \rho \frac{\partial \phi}{\partial t} + \phi \frac{\partial \rho}{\partial t} \\ &= \rho \frac{\partial \phi}{\partial t} - \phi(\mathrm{div}(\rho \underline{u}) - \Gamma) \end{aligned} \tag{10}$$

The second term on the right-hand side can be included in S_ϕ as:

$$S'_\phi = S_\phi + \phi(\mathrm{div}(\rho \underline{u}) - \Gamma) \tag{11}$$

System (9) becomes:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \mathrm{div}(\rho\, \underline{u}) = \Gamma \\ \rho \dfrac{\partial \underline{u}}{\partial t} + \mathrm{div}(\rho\, \underline{u} \otimes \underline{u}) = -\underline{\mathrm{grad}}\ (P) + \mathrm{div}(\underline{\underline{\tau}} - \rho \underline{\underline{R}}) + \underline{S}'_{\underline{u}} \\ \rho \dfrac{\partial a}{\partial t} + \mathrm{div}\left((\rho \underline{u})\, a\right) = \mathrm{div}(\underline{J}_a - \overline{\rho a'' \underline{u}''}) + S'_a \end{cases} \tag{12}$$

3.3 Turbulence modelling

To close system (12), turbulent correlations need to be modelled. The first model that has been implemented in *Code_Saturne* is the "standard" high-Reynolds-number $k - \varepsilon$ model of Launder and Spalding [LAU 74]. Nevertheless, no eddy-viscosity model, however elaborate, can account for anisotropy of turbulence, which might prove of major importance in many applications including curvature, density stratification or swirl. For those types of flows, a second moment closure (Reynolds Stress Model) can yield decisive benefits: we have adopted the proposal by Launder, Reece and Rodi [LRR 75].
The continuous equations of the eddy-viscosity model and of the second moment closure implemented in *Code_Saturne* are simply recalled hereafter (equations (17) and (18)). Both models are operated with log-law-based wall functions. Details can be found in the appendix (section 11).

3.4 Large Eddy Simulation

Apart from the methods previously presented, to which one refers to as Reynolds Average Navier-Stokes methods (R.A.N.S. methods), *Code_Saturne* may also deal with turbulent flows using Large Eddy Simulation (L.E.S.) [LES 97], [POP 00]. The instantaneous Navier-Stokes equations are not averaged anymore, but are spatially filtered and therefore represent only the larger scales of the turbulent motion. As a result, the dissipation due to the smaller scales has to be modelled. This is achieved using a subgrid viscosity μ_t, in a similar way as for the turbulent viscosity of the $k-\varepsilon$ model, but with the essential difference that, in the latter approach, μ_t represents the turbulent dissipation over the whole spectrum of the fluctuating movement, whereas the L.E.S. subgrid viscosity only accounts for the dissipation due to the non-resolved scales. Figure 1 shows the graph of a common energy spectrum for Homogeneous Isotropic Turbulence, separating the simulated zone (explicitly calculated by solving the 3D instantaneous filtered equations) and the zone modelled through a subgrid viscosity.

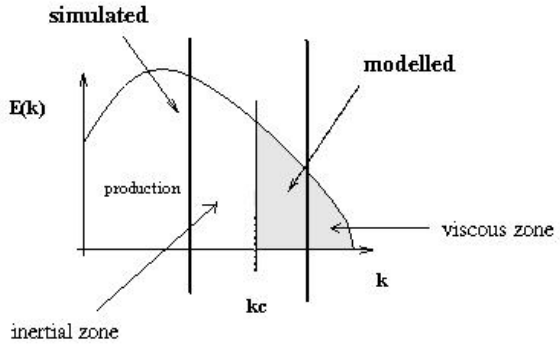


Figure 1: Graph of a common energy spectrum for Homogeneous Isotropic Turbulence: resolved and modelled zones in L.E.S.

One of the key features in L.E.S. is the determination of the subgrid viscosity μ_t. The Smagorinsky model provides a basic approach:

$$\mu_t = \rho(C_s\Delta)^2\sqrt{2D_{ij}D_{ij}} \tag{13}$$

where C_s is a constant (0.18 for Homogeneous Isotropic Turbulence), Δ is the length scale of the filter (usually $\Delta = 2(|\Omega|)^{1/3}$ where $|\Omega|$ is the volume of the cell). More advanced models for μ_t are also available in *Code_Saturne* (so-called "dynamic models", that allow an automatic computation of a local value for C_s).
Specific algorithms have also been implemented in *Code_Saturne* to reach the numerical accuracy needed for L.E.S. computations.

3.5 Complete set of continuous equations

Finally, equations for mass, momentum and scalar read:

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \mathrm{div}(\rho\,\underline{u}) = \Gamma & (14.a) \\ \rho\dfrac{\partial \underline{u}}{\partial t} + \mathrm{div}(\rho\,\underline{u}\otimes\underline{u}) = -\underline{\mathrm{grad}}\ (P) + \mathrm{div}(\underline{\underline{\tau}} - \rho\underline{\underline{R}}) + \underline{S'}_{\underline{u}} & (14.b) \\ \rho\dfrac{\partial a}{\partial t} + \mathrm{div}\left((\rho\underline{u})\,a\right) = \mathrm{div}\left((K_a + \dfrac{\mu_t}{\sigma_{a,t}})\underline{\mathrm{grad}}\ a\right) + S'_a & (14.c) \end{cases}$$

The present considerations are restricted to "incompressible flows", defined as flows for which it is possible to drop the time derivative in the mass equation (14.a) which is then replaced by:

$$\mathrm{div}(\rho\,\underline{u}) = \Gamma \tag{15}$$

Theoretically this restricts the use of the algorithm hereafter to configurations with a density constant in time and space. Still, by extension, it is also applied to situations when ρ is a function of scalars a (such as temperature or species mass fraction) but does not depend on the pressure. Hence, the general (user-defined) equation of state will be denoted:

$$\rho = \mathcal{F}(a) \tag{16}$$

The equations for the two additional variables of the $k-\varepsilon$ model are:

$$\begin{cases} \rho\dfrac{\partial k}{\partial t} + \mathrm{div}\left[\rho\underline{u}\,k - \left(\mu + \dfrac{\mu_t}{\sigma_k}\right)\underline{\mathrm{grad}}\ k\right] &=& \mathcal{P} + \mathcal{G} - \rho\varepsilon + S'_k \\ \rho\dfrac{\partial \varepsilon}{\partial t} + \mathrm{div}\left[\rho\underline{u}\,\varepsilon - \left(\mu + \dfrac{\mu_t}{\sigma_\varepsilon}\right)\underline{\mathrm{grad}}\ \varepsilon\right] &=& C_{\varepsilon 1}\dfrac{\varepsilon}{k}\left[\mathcal{P} + (1-C_{\varepsilon 3})\mathcal{G}\right] \\ & & \qquad -\rho C_{\varepsilon 2}\dfrac{\varepsilon^2}{k} + S'_\varepsilon \end{cases} \tag{17}$$

The equations for the seven additional variables of the second moment closure (R.S.M.) read:

$$\begin{cases} \rho\dfrac{\partial R_{ij}}{\partial t} + \mathrm{div}(\rho\underline{u}\,R_{ij} - \mu\,\underline{\mathrm{grad}}\ R_{ij}) &=& \mathcal{P}_{ij} + \mathcal{G}_{ij} + \Phi_{ij} + d_{ij} - \rho\varepsilon_{ij} + S'_{ij} \\ \rho\dfrac{\partial \varepsilon}{\partial t} + \mathrm{div}(\rho\underline{u}\,\varepsilon - \mu\,\underline{\mathrm{grad}}\ \varepsilon) &=& d_\varepsilon + C_{\varepsilon 1}\dfrac{\varepsilon}{k}\left[\mathcal{P} + \mathcal{G}_\varepsilon\right] \\ & & \qquad -\rho C_{\varepsilon 2}\dfrac{\varepsilon^2}{k} + S'_\varepsilon \end{cases} \tag{18}$$

The different terms on the right-hand side of equations (17) and (18) are explicited in the appendix, section 11.

4 Time discretization

Time discretization of the incompressible Navier-Stokes equations is achieved through a fractional step scheme ([BRE 91] [CHO 68] [TEM 79]), that can be associated with the SIMPLEC algorithm [VER 95]. Because of the dependence of the fluid density on temperature in many problems treated by *Code_Saturne*, the resolution of the scalar equations is treated as a third step in the algorithm. For the sake of clarity, turbulent terms are not taken into account in the algorithm detailed hereafter.

Let us consider time step n, extending from $t = t^{(n)}$ to $t = t^{(n+1)}$.

- $\phi^{(n)}$ denotes the value of a variable $\phi(t)$ at time $t = t^{(n)}$.
- $\Delta t = t^{(n+1)} - t^{(n)}$ is the (predetermined or user-defined) time step value.
- The momentum is $\underline{Q}^{(n)} = (\rho \underline{u})^{(n)}$.
- The pressure is $P^{(n)}$.
- Additional scalars are denoted $a^{(n)}$ (for example, temperature is $T^{(n)}$).
- The density is $\rho^{(n)} = \mathcal{F}(a^{(n)})$; it has been predetermined using the equation of state (16).
- Similarly, the other physical properties of the fluid (dynamic viscosity, conductivity and specific heat) have been calculated beforehand and are treated explicitly (hence, the (n) superscript referring to time step will be dropped for these properties).

STEP 1

The first step consists in a prediction of the velocity, solving equation (14.b), with an explicit pressure gradient. All the source terms (including in particular the viscous terms depending on $\underline{\underline{\text{grad}}}^t\ \underline{u}$) have been collected into $\underline{S'_u}$. This composite source term is written $\underline{S'_u} = \underline{A} + \underline{\underline{B}}.\underline{u}$, so that it can partially be made implicit. To make it possible to solve the equations for the three components of velocity separately, $\underline{\underline{B}}$ is diagonal and the convective mass flux is treated explicitly.
The value of any variable ϕ obtained at the end of this first step is denoted ϕ^*. Hence, the system finally reads:

$$\left\{ \begin{array}{l} \dfrac{\underline{Q}^* - \underline{Q}^{(n)}}{\Delta t} + \text{div}\left[\underline{u}^* \otimes \underline{Q}^{(n)} - \mu \underline{\underline{\text{grad}}}\ \underline{u}^*\right] = -\underline{\text{grad}}\ P^{(n)} + \underline{A}^{(n)} + \underline{\underline{B}}^{(n)}.\underline{u}^* \\ \\ P^* = P^{(n)} \\ a^* = a^{(n)} \\ \\ \qquad\qquad \text{with } \underline{u}^* = \dfrac{\underline{Q}^*}{\rho^*} = \dfrac{\underline{Q}^*}{\rho^{(n)}} \end{array} \right. \tag{19}$$

STEP 2

The second step consists in a correction of the predicted velocity to take into account the pressure variation in (14.b). At this stage, the variation of the convection and

diffusion terms is neglected. To enforce mass conservation, equation (15) is taken into account. The system is therefore:

$$\begin{cases} \underline{Q}^{**} - \underline{Q}^{*} = -\Delta t \, \underline{\text{grad}} \, (P^{**} - P^{*}) & (20.a) \\ \text{div}(\underline{Q}^{**}) = \Gamma & (20.b) \\ a^{**} = a^{*} & (20.c) \end{cases}$$

In practice, to derive a system for pressure variation, one takes the divergence of equation (20.a) and uses equation (20.b) to eliminate $\underline{Q}^{**}$, which yields the following Poisson equation:

$$\text{div}\left[\Delta t \, \underline{\text{grad}} \, (P^{**} - P^{*})\right] = \text{div}(\underline{Q}^{*}) - \Gamma \tag{21}$$

STEP 3

The third step consists in the resolution of equation (14.c) for additional scalars (such as temperature for example). Following the same approach as for the velocity, the source term is written S'_a as $A_a + B_a a$ so that it can be partially implicited. Hence one gets:

$$\begin{cases} \underline{Q}^{(n+1)} = \underline{Q}^{**} \\ P^{(n+1)} = P^{**} \\ \rho^{(n)} \dfrac{a^{(n+1)} - a^{**}}{\Delta t} + \text{div} \left[a^{(n+1)} \underline{Q}^{**} - K_a \underline{\text{grad}} \, a^{(n+1)} \right] = A_a^{(n)} + B_a^{(n)} a^{(n+1)} \end{cases} \tag{22}$$

Once this step has been completed, the equation of state (16) can be used to update density. The other physical properties of the fluid may also be updated and the whole process can then start again.

When turbulence models are used, the resolution of the turbulent equations takes place between step 2 and step 3. The equations for the R.S.M. model are treated in a similar way as the equations for the scalars (equation of convection, diffusion and source terms). The dependence on the other turbulent variables is fully explicit, so that each equation is solved independently. For the $k - \varepsilon$ model however, equations for k and ε are solved simultaneously in order to partially take into account the equilibrium between these two variables. Details of the algorithm are provided in the appendix, section 12.

5 Discretization in space

5.1 Introduction

Discretization in space is achieved using a finite volume approach for a co-located arrangement of all variables. It is applied to the systems of equations resulting from the discretization in time presented in section 4. The methodology is detailed

hereafter for the first two steps (momentum and pressure correction). The extension to the last one (equations for scalars) is straightforward. Therefore, we only consider the following equations:

$$\begin{cases} \dfrac{\underline{\underline{Q}}^* - \underline{\underline{Q}}^{(n)}}{\Delta t} + \mathrm{div}\left[\underline{u}^* \otimes \underline{\underline{Q}}^{(n)} - \mu \underline{\underline{\mathrm{grad}}}\ \underline{u}^*\right] = -\underline{\mathrm{grad}}\ P^{(n)} + \underline{A}^{(n)} + \underline{\underline{B}}^{(n)}.\underline{u}^* \\ \mathrm{div}\left[\Delta t\, \underline{\mathrm{grad}}\ (P^{**} - P^*)\right] = \mathrm{div}(\underline{Q}^*) - \Gamma \\ \underline{Q}^{**} = \underline{Q}^* - \Delta t\, \underline{\mathrm{grad}}\ (P^{**} - P^*) \end{cases} \tag{23}$$

5.2 Definitions

- For the sake of clarity, the present considerations are restricted to cells for which no face is located on the domain boundary. Figure 2 represents entities related to face (i,j). F_{ij} is the centre of face (i,j). The cell "centres" I and J of neighbouring cells i and j are their respective mass centres. The point O_{ij} is defined as the intersection between the face (i,j) and the straight line (IJ). I' and J', defined on figure 2, are the points obtained by projecting I and J orthogonally on the line normal to face (i,j) and containing F_{ij}. We also define coefficient α_{ij} for linear interpolations as $\alpha_{ij} = \dfrac{(\underline{F_{ij}J'} \cdot \underline{n}_{ij})}{(\underline{I'J'} \cdot \underline{n}_{ij})}$

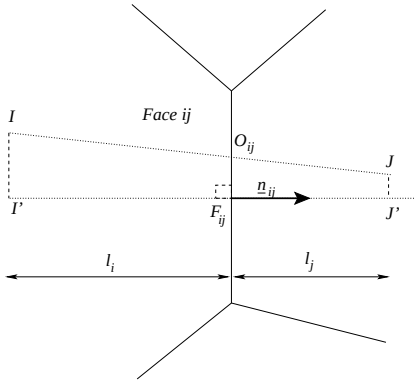


Figure 2: Notations for geometric entities related to face (i,j).

- For any continuous variable $b(\underline{x},t)$, $b_i^{(n)}$ represents the mean value computed over cell i at time $t = t^{(n)}$. Hence, assuming the definition of the cells is not time-dependent, we have:

$$b_i^{(n)} = \frac{1}{|\Omega_i|}\int_{\Omega_i} b(\underline{x}, t^{(n)})d\Omega \tag{24}$$

When Taylor expansions are required for higher order schemes, the value for $b_i^{(n)}$ is associated with the cell centre I.

For density, the equation of state (16) is used to determine $\rho_i^{(n)} = \mathcal{F}\left(a_i^{(n)}\right)$.

- $b_{i'}^{(n)}$ represents the approximate value of $b(\underline{x},t)$ at time $t=t^{(n)}$ at point I'. Assuming $(\underline{\text{grad}}\ (b))_i$, the discrete gradient of variable b, is known at cell i, $b_{i'}^{(n)}$ is determined as follows, using a first order approximation:

$$b_{i'}^{(n)} = b_i^{(n)} + \underline{II'}.(\underline{\text{grad}}\ b)_i \tag{25}$$

- Assuming $\underline{\text{grad}}\ b$ exists and is continuous, we define $\underline{G_i}$ as the operator providing the discrete gradient of b at cell i:

$$\underline{G_i}(b) = (\underline{\text{grad}}\ b)_i \tag{26}$$

  Two methods are available in *Code_Saturne* to calculate $\underline{G_i}(b)$. They are described hereafter, section 5.3.

- We define the operator $\mathcal{G}_{\underline{n},ij}$, providing the normal gradient at face (i,j):

$$\mathcal{G}_{\underline{n},ij}(b) = \frac{b_{j'} - b_{i'}}{(\underline{IJ} \cdot \underline{n}_{ij})} \tag{27}$$

- We finally define the interpolation operator $\mathcal{I}_{ij}$, providing values at face (i,j):

$$\mathcal{I}_{ij}(b) = \alpha_{ij} b_i + (1-\alpha_{ij}) b_j + \frac{1}{2}\underline{O_{ij}F_{ij}} \cdot \Big((\underline{\text{grad}}\ b)_i + (\underline{\text{grad}}\ b)_j \Big) \tag{28}$$

5.3 Gradient calculation

5.3.1 Introduction

We describe here the computation of the discrete gradient of b at cell i: $\underline{G_i}(b)$. Two methods are available in *Code_Saturne*: a "standard" one and a "least square" one[3]. Some numerical experiments related to the order of convergence in space of these methods are also presented in the appendix, section 13.

5.3.2 Standard method in *Code_Saturne*

The standard method follows [MUS 96]. Starting from equation:

$$(\underline{\text{grad}}\ b)_i \overset{def}{=} \frac{1}{|\Omega_i|} \int_{\Omega_i} \underline{\text{grad}}\ b\, d\Omega \tag{29}$$

Gauss theorem implies:

$$(\underline{\text{grad}}\ b)_i = \frac{1}{|\Omega_i|} \sum_{j \in Neibrs(i)} \int_{S_{ij}} b\, \underline{n}_{ij}\, dS \tag{30}$$

[3]Users may choose which method shall be used. Once they have made this decision, the selected method is applied to all discrete gradients that are required at cell centres.

Assuming $\underline{n}_{ij}$ is constant across interface (i,j) and denoting:

$$b_{ij} = \frac{1}{S_{ij}} \int_{S_{ij}} b \, dS \tag{31}$$

equation (30) reads:

$$(\underline{\text{grad}}\ b)_i = \frac{1}{|\Omega_i|} \sum_{j \in Neibrs(i)} b_{ij} S_{ij} \underline{n}_{ij} \tag{32}$$

A second order approximation is obtained for b_{ij} using the mid-point rule for integration:

$$b_{ij} \approx b_{F_{ij}} \tag{33}$$

Using notations defined on figure 2, a Taylor expansion is then applied to write the following second order approximation:

$$b_{F_{ij}} = b_{O_{ij}} + \underline{O_{ij}F_{ij}} . (\underline{\text{grad}}\ b)_{O_{ij}} + O(||\underline{O_{ij}F_{ij}}||^2) \tag{34}$$

With a linear interpolation for $b_{O_{ij}}$ and a mere average for $(\underline{\text{grad}}\ b)_{O_{ij}}$ in equation (34), system (30) finally reads:

$$|\Omega_i| \, (\underline{\text{grad}}\ b)_i = \sum_{j \in Neibrs(i)} \left\{ \alpha_{ij} b_i + (1-\alpha_{ij}) \, b_j + \tfrac{1}{2} \underline{O_{ij}F_{ij}} \cdot \Big((\underline{\text{grad}}\ b)_i + (\underline{\text{grad}}\ b)_j \Big) \right\} \cdot S_{ij} \underline{n}_{ij} \tag{35}$$

In (35), $(\underline{\text{grad}}\ b)_{O_{ij}}$ has been evaluated as $\frac{1}{2}(\underline{\text{grad}}\ b)_i + \frac{1}{2}(\underline{\text{grad}}\ b)_j$, a mere average. Indeed, this is sufficient to obtain a second order approximation for $b_{F_{ij}}$ (the linear interpolation $\alpha_{ij}(\underline{\text{grad}}\ b)_i + (1-\alpha_{ij})(\underline{\text{grad}}\ b)_j$ would not increase the order in space). Moreover, numerical experiments carried out on complex industrial cases indicated that evaluating $(\underline{\text{grad}}\ b)_{O_{ij}}$ as $\frac{1}{2}(\underline{\text{grad}}\ b)_i + \frac{1}{2}(\underline{\text{grad}}\ b)_j$ seemed to provide more robustness than using the linear interpolation.

For two-dimensional problems, on equilateral meshes (and also on triangular meshes for which the cell centres are the circumcentres), this method degenerates to the scheme studied in [BCH 00] and applied in [BCH 01] since, in these conditions, one has $\underline{O_{ij}F_{ij}} = \underline{0}$.

5.3.3 Least squares method

However, solving system (35) remains costly. Hence, a more direct method has been implemented. It is based on the least squares approach described in [MUS 96]. Assuming $||\underline{IJ}|| \neq 0$, the rationale is to determine the value for $(\underline{\text{grad}}\ b)_i$ that minimizes $\mathcal{J}_i^b(\underline{\text{grad}}\ b, \underline{\text{grad}}\ b)$, with:

$$\mathcal{J}_i^b(\underline{\text{grad}}\ b, \underline{\text{grad}}\ b) = \sum_{j \in Neibrs(i)} \frac{1}{||\underline{IJ}||^2} \left(b_j - b_i - (\underline{\text{grad}}\ b)_i \cdot \underline{IJ} \right)^2 \tag{36}$$

Contrary to system (35), the least squares method does not require solving an implicit set of equations and is therefore much faster.

It is pointed out that if, for a given cell i, there exist at least three faces (two in two-dimensional problems) for which the related vectors $\underline{IJ}$ are linearly independent, the minimization of (36) has a unique solution.
Moreover, let us consider system (36) on a (two-dimensional) triangular cell i. Neighbouring cells are denoted j, k and l. We also define the unit vectors $\underline{m}_{ip} = \dfrac{1}{||IP||}\underline{IP}$ and the scalar quantity $g_p = \dfrac{b_p - b_i}{||\underline{IP}||}$ (with p standing for j, k or l and P for the associated cell centres). With these notations, the minimum of $\mathcal{J}_i^b(\underline{\text{grad}}\ b, \underline{\text{grad}}\ b)$ reaches zero under the following condition:

$$g_j(\underline{m}_{ik} \times \underline{m}_{il}) + g_k(\underline{m}_{il} \times \underline{m}_{ij}) + g_l(\underline{m}_{ij} \times \underline{m}_{ik}) = \underline{0} \tag{37}$$

5.4 Discretization in space of the momentum equation

The momentum equation is integrated over cell i as follows, using the Gauss theorem for divergence terms and pressure gradient:

$$\begin{aligned} \frac{|\Omega_i|}{\Delta t}(\underline{Q}_i^* - \underline{Q}_i^{(n)}) + \sum_{j \in Neibrs(i)} (\underline{u}^*)_{ij}(\underline{Q}^{(n)} \cdot \underline{n})_{ij} S_{ij} - \sum_{j \in Neibrs(i)} (\mu \underline{\underline{\text{grad}}}\ \underline{u}^* \cdot \underline{n})_{ij} S_{ij} \\ = -|\Omega_i| G_i(P^{(n)}) + |\Omega_i| \underline{A}_i^{(n)} + |\Omega_i| \underline{\underline{B}}_i^{(n)} . \underline{u}_i^* \end{aligned} \tag{38}$$

with:

$$\underline{u}_i^* = \frac{\underline{Q}_i^*}{\rho_i^*} = \frac{\underline{Q}_i^*}{\rho_i^{(n)}} \tag{39}$$

Terms $\underline{A}_i^{(n)}$ and $\underline{\underline{B}}_i^{(n)}$ may require the gradient of any variable ϕ. It is evaluated at cell centres as $G_i(\phi)$. The method is the same for the pressure gradient $G_i(P^{(n)})$.

Face values for $(\underline{Q}^{(n)} \cdot \underline{n})_{ij}$ (mass flux through face (i,j)) are obtained directly from the previous pressure correction step. Their computation is described in the next section.
Several convection schemes are available for the determination of face values for the velocity $(\underline{u}^*)_{ij}$. With a first-order donor-cell scheme, we have:

$$\begin{aligned} (\underline{u}^*)_{ij} &= \underline{u}_i^* \qquad \text{if } (\underline{Q}^{(n)} \cdot \underline{n})_{ij} \geqslant 0 \\ &= \underline{u}_j^* \qquad \text{if } (\underline{Q}^{(n)} \cdot \underline{n})_{ij} < 0 \end{aligned} \tag{40}$$

A centred scheme and a second-order linear upwind scheme can also be used: $(\underline{u}^*)_{ij}$ is then determined from $(\underline{u}^*)_i$, $(\underline{u}^*)_j$ and gradients $G_i(u^*)$ and $G_j(u^*)$ computed at cells i and j.

For the diffusion term, $(\mu \underline{\underline{\text{grad}}}\ \underline{u}^* \cdot \underline{n})_{ij}$ is defined as follows:

$$(\mu\ \underline{\underline{\text{grad}}}\ \underline{u}^* \cdot \underline{n})_{ij} = \mu_{ij}\, \mathcal{G}_{\underline{n},ij}(\underline{u}^*) \tag{41}$$

for historical reasons, two options have been implemented for the face value of μ. The first one simply relies on a linear interpolation, while the second one ensures the continuity of the normal flux at interfaces:

$$\mu_{ij} = \alpha_{ij}\mu_i + (1-\alpha_{ij})\mu_j \qquad \text{or} \qquad \mu_{ij} = \frac{\mu_i \mu_j}{\alpha_{ij}\mu_i + (1-\alpha_{ij})\mu_j} \tag{42}$$

5.5 Pressure correction

The equation for pressure increment $\delta P = P^{**} - P^*$ is integrated as follows:

$$\Delta t \sum_{j \in Neibrs(i)} (\underline{\text{grad}}\ (\delta P) \cdot \underline{n})_{ij} S_{ij} = \sum_{j \in Neibrs(i)} (\underline{Q}^* \cdot \underline{n})_{ij} S_{ij} - |\Omega_i| \Gamma_i \tag{43}$$

The normal gradient on the left-hand side is evaluated as:

$$(\underline{\text{grad}}\ (\delta P) \cdot \underline{n})_{ij} = \mathcal{G}_{\underline{n},ij}(\delta P) \tag{44}$$

On the right-hand side, the "Rhie and Chow" interpolation [RHI 84] is used to compute face values for $(\underline{Q}^* \cdot \underline{n})_{ij}$. We first define $\underline{Q}^{*,P}$ at cell centres as the solution of the momentum equation without pressure gradient:

$$\underline{Q}_i^{*,P} = \underline{Q}_i^* + \Delta t G_i(P^{(n)}) \tag{45}$$

With this notation, we compute $(\underline{Q}^* \cdot \underline{n})_{ij}$ at interface (i,j) as:

$$(\underline{Q}^* \cdot \underline{n})_{ij} = \mathcal{I}_{ij}(\underline{Q}^{*,P}) - \Delta t \mathcal{G}_{\underline{n},ij}(P^{(n)}) \tag{46}$$

Once the equation for pressure increment has been solved, the mass flux through faces has to be updated to obtain $(\underline{Q}^{**} \cdot \underline{n})_{ij}$:

$$(\underline{Q}^{**} \cdot \underline{n})_{ij} = (\underline{Q}^* \cdot \underline{n})_{ij} - \Delta t \mathcal{G}_{\underline{n},ij}(\delta P) \tag{47}$$

It is pointed out that this mass flux naturally satisfies the discrete mass equation:

$$\sum_{j \in Neibrs(i)} (\underline{Q}^{**} \cdot \underline{n})_{ij} S_{ij} = |\Omega_i| \Gamma_i \tag{48}$$

It remains unchanged until the next time step (*i.e.* $(\underline{Q}^{(n+1)} \cdot \underline{n})_{ij} = (\underline{Q}^{**} \cdot \underline{n})_{ij}$) and is then used for the momentum equation, as previously stated in section 5.4.

The momentum at cell centres $\underline{Q}_i^*$ also needs to be corrected for the next time step. This is achieved as follows:

$$\underline{Q}_i^{**} = \underline{Q}_i^* - \Delta t G_i(\delta P) \tag{49}$$

The momentum then remains unchanged until the next time step ($\underline{Q}_i^{(n+1)} = \underline{Q}_i^{**}$). Finally, pressure also needs to be corrected for the next time step:

$$P_i^{(n+1)} = P_i^{**} = P_i^* + \delta P_i \tag{50}$$

6 Boundary conditions

6.1 Introduction

We focus here on the description of boundary conditions for the discrete momentum equation (38), for the Poisson equation (43) and for the associated momentum correction at cell centres (49). The approach described for the momentum equation can be easily extended to the discrete equation for scalars.

We refer to notations defined on figure 3, $\underline{n}$ standing for the outwards normal unit vector at boundary faces. On figure 3, I is the mass centre of cell i, F is the centre of the boundary face and I' is the point obtained by projecting I orthogonally on the line normal to the boundary face and containing F. Variable ϕ denotes any velocity component or scalar variable except pressure.

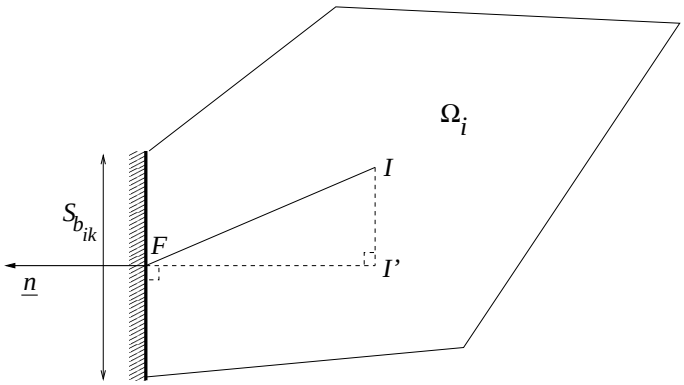


Figure 3: Geometrical entities related to a boundary face.

In the discrete momentum equation (38), conditions at boundaries essentially have to be specified for three types of terms:

- $\mathrm{div}(\underline{Q}^{(n)}\,\phi)$: for these "convection" terms, the finite volume integration requires a boundary value for $(\underline{Q}^{(n)} \cdot \underline{n})\phi$;
- $\mathrm{div}(K\underline{\mathrm{grad}}\,(\phi))$: for these "diffusion" terms, a boundary value is necessary for $K(\underline{\mathrm{grad}}\,(\phi) \cdot \underline{n})$ (where K stands for the fluid viscosity μ or for the scalar diffusivity K_a);
- S_ϕ: for the "source" terms depending on $\underline{\mathrm{grad}}\,(\phi)$, a boundary value for ϕ is required to compute the gradient at cell centres. The pressure gradient appearing on the right-hand side of the momentum equation also falls into this category of source terms.

To solve the Poisson equation (43) for pressure correction δP, boundary values for $\Delta t\,(\underline{\mathrm{grad}}\,(\delta P) \cdot \underline{n})$ (left-hand side) and $(\underline{Q^*} \cdot \underline{n})$ (right-hand side) are required. These

boundary conditions have to be prescribed in a coherent way since the mass flow rate at the end of the time step will result from the summation of these terms:

$$\underline{Q}^{n+1} \cdot \underline{n} = \underline{Q}^{*} \cdot \underline{n} - \Delta t \left(\underline{\mathrm{grad}}\ (\delta P) \cdot \underline{n}\right) \tag{51}$$

Eventually, to update the momentum at cell centres (equation (49)), a boundary treatment needs to be prescribed to compute the cell values of the gradient of the pressure variation.

The boundary treatment for the different terms identified above is detailed hereafer for inlets, outlets, walls and symmetry planes. At last, a subsection is devoted to a specific wall boundary condition applied to temperature in conjugate heat transfer simulations. Considerations related to turbulence modelling (equations (17) and (18)) have not been included here: details can be found in the appendix, section 14.

6.2 Inlet

Boundary conditions for all discrete terms are built under the following assumptions:

- inlet quantities $\phi_{\mathrm{inlet}}^{(n+1)}$ such as momentum $\underline{Q}_{\mathrm{inlet}}^{(n+1)}$ and scalars $a_{\mathrm{inlet}}^{(n+1)}$ are prescribed as Dirichlet conditions at the beginning of time step n;
- pressure gradient normal to inlet $(\underline{\mathrm{grad}}\ (P^{(n+1)}) \cdot \underline{n})_{\mathrm{inlet}}$ is given (usually zero, but an explicit extrapolation of the normal pressure gradient previously obtained in the boundary cell $\left((\underline{\mathrm{grad}}\ (P^{(n)}))_i \cdot \underline{n}\right)$ may also be used).

For convection terms, the boundary value for $Q_{\phi^*}^{(n)} = \left(\underline{Q}^{(n)} \cdot \underline{n}\right) \phi^*$ is:

$$\left(Q_{\phi^*}^{(n)}\right)_{\mathrm{inlet}} = \left(\underline{Q}_{\mathrm{inlet}}^{(n)} \cdot \underline{n}\right) \phi_{\mathrm{inlet}}^{(n+1)} \tag{52}$$

For diffusion terms, the boundary value for $D_{\phi}^{*} = K(\underline{\mathrm{grad}}\ (\phi^*) \cdot \underline{n})$ is:

$$(D_{\phi}^{*})_{\mathrm{inlet}} = K_i \, \frac{\phi_{\mathrm{inlet}}^{(n+1)} - \phi_{i'}^{*}}{(\underline{I'F} \cdot \underline{n})} \tag{53}$$

For source terms requiring the computation of the gradient of any variable ϕ^*, the value $\phi_{\mathrm{inlet}}^{(n+1)}$ is used as a boundary face value. For computation of pressure gradient, a boundary value for pressure $P_{\mathrm{inlet}}^{(n+1)}$ is obtained from the prescribed value of $\left(\underline{\mathrm{grad}}\ (P^{(n+1)}) \cdot \underline{n}\right)_{\mathrm{inlet}}$ and from the pressure value at the boundary cell i (using a first order approximation in space).

To solve the Poisson equation for pressure correction, the boundary value for the quantity $\Delta t\,(\underline{\mathrm{grad}}\ (\delta P) \cdot \underline{n})$ is set to zero, while $(\underline{Q}^{*} \cdot \underline{n})$ is computed at inlet as $(\rho_{\mathrm{inlet}}^{(n+1)}\ \underline{u}_{\mathrm{inlet}}^{(n+1)} \cdot \underline{n})$ so that the final inlet mass flow rate have the expected value.

Finally, the momentum correction at cell centres require the computation of the cell value for $\underline{\mathrm{grad}}\ (\delta P)$. The boundary value for δP is extrapolated (first order approximation in space) from the cell value $(\delta P)_i$ under the assumption that $\Delta t\,(\underline{\mathrm{grad}}\ (\delta P) \cdot \underline{n}) = 0$ (coherent with the fact that $\left(\underline{\mathrm{grad}}\ (P^{(n+1)}) \cdot \underline{n}\right)_{\mathrm{inlet}}$ has been used to predict $\underline{Q}^{*}$).

6.3 Outlet

Boundary conditions for all discrete terms are build under the following assumptions:

- homogeneous Neumann conditions apply to velocity and scalars;
- a Dirichlet condition $P_{\text{outlet}}^{(n+1)}$ applies to pressure (pressure is set to zero at an arbitrary outlet face and the pressure profile across outlet(s) is supposed to reproduce the pressure obtained upstream at the previous time step).

For convection terms, the mass flow rate $(\underline{Q}^{(n)} \cdot \underline{n})_{\text{outlet}}$ is the value previously computed at outlet as a result of the pressure correction step. Using the homogeneous Neumann condition for ϕ, the boundary value for $Q_{\phi^*}^{(n)} = \left(\underline{Q}^{(n)} \cdot \underline{n}\right) \phi^*$ is calculated at first order in space as:

$$(Q_{\phi^*}^{(n)})_{\text{outlet}} = \left(\underline{Q}^{(n)} \cdot \underline{n}\right)_{\text{outlet}} \phi_{i'}^* \tag{54}$$

For diffusion terms, the boundary value for $D_{\phi}^* = K(\underline{\text{grad}}\ (\phi^*) \cdot \underline{n})$ is set to zero.

For source terms requiring the computation of the gradient of any variable ϕ, the boundary value for ϕ is set to $\phi_{i'}$, (first order approximation in space of the homogeneous Neumann condition). For computation of the pressure gradient in the momentum equation, the Dirichlet condition for pressure immediately provides a boundary value.

To solve the Poisson equation for pressure correction, a homogeneous Dirichlet condition is applied to δP (since a Dirichlet condition $P_{\text{outlet}}^{(n+1)}$ has been applied to pressure) while $(\underline{Q}^* \cdot \underline{n})$ is computed as $(\rho_{\text{outlet}}^{(n)}\, \underline{u}_{i'}^* \cdot \underline{n})$[4].
Finally, the momentum correction at cell centres require the computation of the cell value for $\underline{\text{grad}}\ (\delta P)$. The boundary value for δP is set to zero.

6.4 Walls and symmetry planes

The present considerations are limited to laminar flows (considerations related to turbulence can be found in the appendix, section 14). Boundary conditions for all discrete terms are built under the following assumptions:

- zero mass flow rate normal to the boundary;
- for tangential velocity, the boundary conditions are homogeneous Dirichlet conditions at walls and homogeneous Neumann conditions at symmetry planes;
- for scalars, Dirichlet conditions may be used (for exemple, in case of a wall kept at a given temperature) but Neumann conditions can also be applied (for example at walls where a thermal flux is imposed or at symmetry planes where the normal gradient is zero);

[4] $\rho_{\text{outlet}}^{(n+1)}$ cannot be used for $(\underline{Q}^* \cdot \underline{n})$ because it is a function of $a_{\text{outlet}}^{(n+1)}$. Indeed, due to the Neumann boundary condition on a, $a_{\text{outlet}}^{(n+1)}$ is equal to $a_{i'}^{(n+1)}$, which will only be calculated later.

- pressure gradient normal to boundary $((\underline{\text{grad}}\ (P))_\text{b} \cdot \underline{n})$ is given (usually zero, but an explicit extrapolation the value obtained in the boundary cell $((\underline{\text{grad}}\ (P))_i \cdot \underline{n})$ may also be used).

For convection terms, the boundary value for $Q^{(n)}_{\phi^*} = (\underline{u}^{(Q)} \cdot \underline{n})\phi^*$ is simply set to zero.

For diffusion terms, if a Neumann condition applies to variable ϕ, it can be used in a straighforward way to evaluate the boundary value for $D^*_\phi = K(\underline{\text{grad}}\ (\phi^*) \cdot \underline{n})$. If a Dirichlet condition applies, the boundary value for D^*_ϕ is computed following the same approach as for inlets (equation 53).

For source terms requiring the computation of the gradient of any variable ϕ, a boundary face value ϕ_b is necessary:

- if a Dirichlet condition is applied to ϕ, it provides immediately a value for ϕ_b;
- for the velocity component normal to the boundary, ϕ_b is set to zero (coherent with the zero mass flux assumption);
- if a Neumann condition applies to ϕ, the boundary value is extrapolated from the boundary cell value using a first order approximation in space. The same treatment is used for the pressure gradient appearing in the momentum equation. Hence, if $(\underline{\text{grad}}\ (\phi) \cdot \underline{n})_\text{b}$ is provided at the boundary, we have:

$$\phi_b = \phi_{i'} + \overline{I'F}\ (\underline{\text{grad}}\ (\phi) \cdot \underline{n})_\text{b} \tag{55}$$

To solve the Poisson equation for pressure correction, the boundary values for $\Delta t\,(\underline{\text{grad}}\ (\delta P) \cdot \underline{n})$ and $(\underline{Q}^* \cdot \underline{n})$ are set to zero.

Finally, the momentum correction at cell centres require the computation of the cell value for $\underline{\text{grad}}\ (\delta P)$. The boundary value for δP is obtained from the cell value under the assumption that $\Delta t\,(\underline{\text{grad}}\ (\delta P) \cdot \underline{n}) = 0$ using a first order approximation in space.

6.5 Coupling with thermal code SYRTHES

This section describes the method used to apply a specific Dirichlet condition T_w to temperature at solid walls.

Indeed, in many applications, the effect of walls has to be taken into account precisely, especially with regards to thermal inertia. Therefore, *Code_Saturne* can be coupled to the EDF thermal code SYRTHES ([PEN 97], [RUP 99]) which solves the temperature in a specified solid domain. SYRTHES relies on a finite element technique to solve the general heat equation (56) where all properties can be time, space or temperature dependent.

$$\rho C_p \frac{\partial T}{\partial t} = \text{div}(k_s\,\underline{\text{grad}}\ T) + \Phi_v \tag{56}$$

T is the temperature, t the time, Φ_v a volumic source or sink, ρ and C_p, respectively the density and the specific heat. k_s (a tensor when the material is anisotropic) designates the conductive behaviour of the medium. Radiation phenomena from wall to wall can also be taken into account. For optimization reasons, only P1-isoP1 elements have been retained (6 nodes per triangle in 2D, 10 nodes per tetrahedron in 3D). More details on the possibilities of the finite element code SYRTHES can be found in [PEN 97], [PEN 98] and [RUP 99]. Like *Code_Saturne*, SYRTHES has been checked thoroughly against experimental and analytical test cases.
The thermal coupling between both codes is explicit and is performed every time step. Let T_w be the temperature of the solid at a solid node belonging to the interface between the fluid and the solid, and T_f the fluid temperature at the centre of the fluid cell adjacent to the wall. At time $t^{(n)}$, SYRTHES sends to *Code_Saturne* the value of $T_w^{(n)}$. *Code_Saturne* then calculates the local heat exchange coefficient $h^{(n)}$ (coherent with the temperature difference $T_f^{(n)} - T_w^{(n)}$) and sends the information $(h^{(n)}, T_f^{(n)})$ to SYRTHES. Using $T_w^{(n)}$ and $h^{(n)}$, *Code_Saturne* processes another time step and calculates $T_f^{(n+1)}$. In parallel, with $T_f^{(n)}$ and $h^{(n)}$, SYRTHES updates the solid temperature and calculates $T_w^{(n+1)}$. The procedure can then go on.

7 Practical use : mesh management

7.1 Introduction

Industrial problems encountered in power generation industry are quite often characterized by a complex geometry that needs to be accounted for in computations. This prerequisite has been kept in mind from the outset of *Code_Saturne* project. It has undoubtedly been one of the reasons that induced the choice for a fully co-located arrangement (previous experience had shown that staggered arrangements made the generation of complex grids quite tedious). But above all, it has been the very motivation for devoting specific attention to geometrical data management. The "Envelope" module of *Code_Saturne* has been developed and continuously enriched to serve this purpose. It has become one key asset of the code, allowing for example:

- the treatment of arbitrary interfaces that facilitates mesh generation without distorting elements or demanding too many cells;
- the domain partitioning for parallel computing.

A brief illustration is proposed in the following sections.

7.2 Arbitrary interfaces

Since *Code_Saturne* does not provide any build-in mesh generator, users usually produce their grids using various commercial software (as I-DEAS, ICEM-CFD or SIMAIL) that are employed depending on the user's expertise and on his specific assets. Hence, it is important to be able to import, connect and use grids produced by

any of these packages without being subjected to stringent constraints at interfaces. The use of "arbitrary interfaces" is a possible solution. "Arbitrary interfaces" refers to interfaces for which two neighbouring control volumes do not necessarily share a whole face and may not have common edges or vertices, as illustrated on figure 4.

Finite volume techniques based on face fluxes are quite naturally applicable to polyhedral cells with arbitrary interfaces. Indeed, on figure 4, the initial rectangular cells that have been connected are actually treated as if they had five faces, so that no modification of the discrete method is required and the basic coding remains practically unchanged, provided gradients are correctly evaluated.

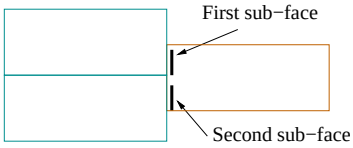


Figure 4: Sketch of an arbitrary interface.

A more important effort has been put into data management and underlying structures (specified using object-oriented analysis). The definition of local tolerance parameters has also been necessary to decide whether vertices needed to be collapsed, edges intersected and faces merged. This task demanded specific attention, especially because the arbitrary interfaces are not always plane and do not necessarily match perfectly for industrial three-dimensional grids.

The construction of the geometrical data structure is carried out in a pre-processing phase automatically chained to the flow solver itself. The key idea consists in constructing structures that are based on the face-to-cell connectivity. Since a face is always defined as a surfacic entity that has two neighbouring cells, this technique makes arbitrary interfaces quite straighforward to deal with. Detailed information are given in the code manuals [FO1 03] and [FO2 03].

As an illustration, figure 5 shows a computation (second moment closure, parallel) on a hybrid mesh. It has been generated by connecting "arbitrarily" to an initial mesh (consisting of tetrahedra) a simple additional part (prisms) so as to displace the boundary condition further downstream.
A second example is shown on figure 6. It represents the mesh used for a thermal-hydraulic nuclear study in the upper plenum of a pressure water reactor. Here, the use of arbitrary interfaces made cell size and mesh quality control much easier for the 10 separate parts of the computational domain. It has also been possible to test different grid refinements for a given part of the mesh without modifing the others. Eventually, additional flexibility has been provided by the fact that the 10 parts of the mesh could be generated independently and stored in separate files.

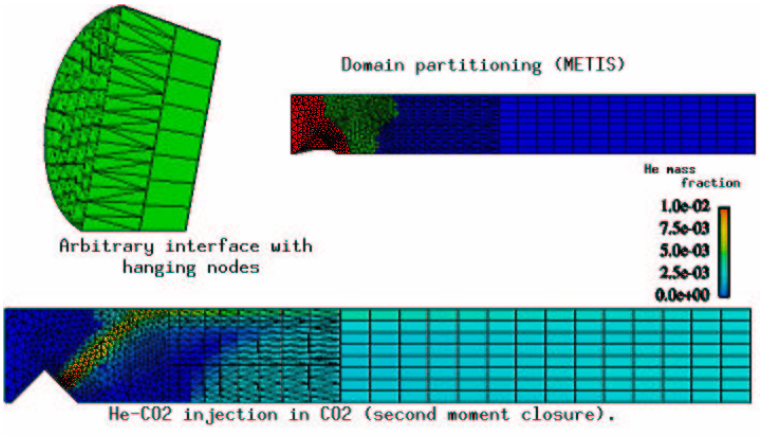


Figure 5: Helium-CO_2 injection in CO_2: second moment closure, hybrid mesh with arbitrary interfaces, parallel computation. Top-Left: detailed view of the arbitrary interface. Top-Right: colours identify the 3 sub-domains defined by MeTiS for parallel computing. Bottom: symmetry plane of the computational domain coloured by Helium mass fraction.

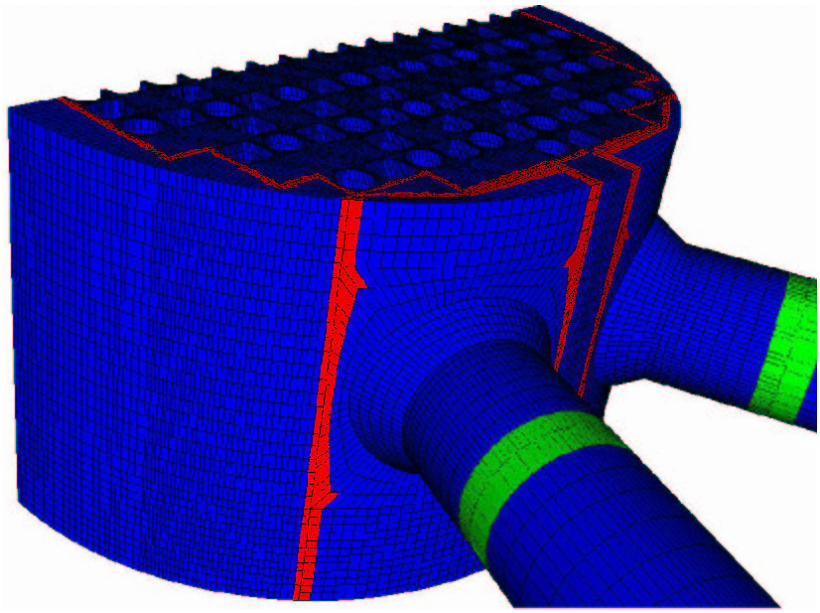


Figure 6: Upper plenum of a nuclear pressure water reactor (1.4 million cells). Faces connected to arbitrary interfaces are marked in red and green.

7.3 Parallel computing

Code_Saturne 1.1 is parallel and runs on distributed memory architectures. Indeed, industrial applications are more and more memory and CPU-time demanding. Hence, for a growing number of users, making the best out of PC clusters might be a way to retain reasonable performance without having access to relatively expensive supercomputers.

The techniques employed in *Code_Saturne* rely on a "Single Program - Multiple Data" approach. The first step consists in partitioning the mesh. To do so, the cells are first arranged into separate "sub-domains" (groups of cells) so that each cell belong to one and only one sub-domain. Each sub-domain is then associated with a unique process of the parallel virtual machine. For this purpose, the widespread MeTiS library [KAR 98], is applied to the (cell→cell) connectivity graph. We found that the sub-domains generated this way were quite well balanced and provided good performance.

The sequential algorithm requires minor changes to be parallelized. This task is achieved by defining "halo" cells to duplicate easily necessary data pertaining to "neighbouring" processes[5] as illustrated on figure 7. The communication required for this purpose is achieved with an MPI library. In particular, data needs to be exchanged when neighbouring cell values are required (flux calculation, matrix-product...) or when global quantities are necessary (dot product, minimum and maximum values...).

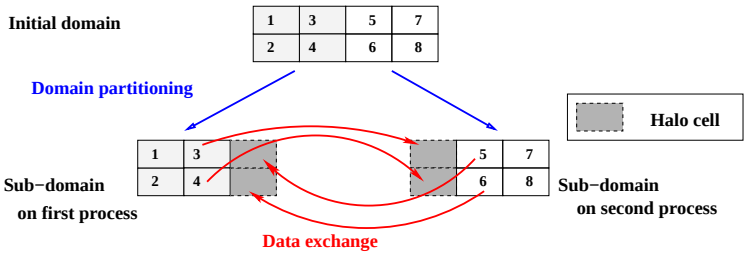


Figure 7: Sketch of halo structure for a fictitious domain mapped on two processes. Data fluxes between sub-domains are represented.

More efforts in defining data structures have been required for some specific capabilities such as periodic boundary conditions: figure 8 illustrates a 4-processor parallel computation of a centrifugal pump (the number of cells has been voluntarily kept small for this demonstration case).

[5] "Neighbouring" processes are defined from the mapping generated at domain partitioning as processes hosting neighbouring cells.

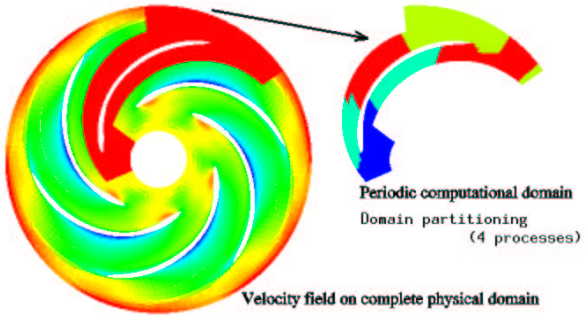


Figure 8: Centrifugal pump. Left: velocity field on the complete physical domain - Right: computational domain partitioning generated by MeTiS with periodic boundary conditions.

8 Application examples

8.1 Example of use of *Code_Saturne* as an open platform

Although the algorithm of *Code_Saturne* has been derived for incompressible flows, the code is also used as an open research and development platform for other types of applications involving numerical systems very different in nature.

For example, work has been carried out on a multidimensional non conservative method for compressible flows with a second moment closure for turbulence [PAH 00].

Indeed, the standard $k-\varepsilon$ model suffers limitations for a number of flows, especially those encountered in turbomachinery applications, which also tend to include compressibility effects. Second moment closures provide some improvement but robustness of the algorithm might be limited by realizability conditions and by the prominant hyperbolic nature of the system.

Within the scope of this work, a simplified second moment closure has been considered, dropping source terms that were expected to have a stabilizing effect:

$$\left\{\begin{array}{lcl} \dfrac{\partial \rho}{\partial t} + \mathrm{div}(\rho\,\underline{u}) & = & 0 \\ \dfrac{\partial}{\partial t}(\rho \underline{u}) + \mathrm{div}\left(\rho\,\underline{u}\otimes\underline{u} + P\underline{\underline{Id}} + \underline{\underline{\mathcal{R}}}\right) & = & 0 \\ \dfrac{\partial E}{\partial t} + \mathrm{div}\left(E\underline{u} + [P\underline{\underline{Id}} + \underline{\underline{\mathcal{R}}}]\underline{u}\right) & = & 0 \\ \dfrac{\partial \mathcal{R}_{ij}}{\partial t} + \mathrm{div}\left(\mathcal{R}_{ij}\underline{u}\right) + NCT_{ij} & = & 0 \end{array}\right. \tag{57}$$

with

$$\begin{cases} NCT_{ij} = \mathcal{R}_{ik}\dfrac{\partial u_j}{\partial x_k} + \mathcal{R}_{jk}\dfrac{\partial u_i}{\partial x_k} & \text{non conservative "production-destruction" term} \\ \underline{\underline{\mathcal{R}}} = \rho\underline{\underline{R}} \text{ and } E = \dfrac{P}{\gamma-1} + \dfrac{1}{2}\rho u_k u_k + \dfrac{1}{2}\mathcal{R}_{kk} \end{cases} \tag{58}$$

With W denoting the conservative m-variables vector ($W = (\rho, \rho\underline{u}, E, \underline{\underline{\mathcal{R}}})^t$), $\underline{C}$ the $m \times m$ matrix related to non conservative terms, system (57) can be written as:

$$\frac{\partial W}{\partial t} + \text{div}\,(\underline{F}(W)) + \underline{C}(W) \cdot \underline{\text{grad}}\,(W) = 0 \tag{59}$$

Non conservative terms are approximated following [BRU 00]:

$$\int_{\Omega_i} \underline{C}(W) \cdot \underline{\text{grad}}\,(W)\, d\Omega \approx \underline{C}(W_i) \cdot \int_{\Omega_i} \underline{\text{grad}}\,(W)\, d\Omega \tag{60}$$

Conservative terms are computed using the Rusanov scheme. It is very easy to implement in a three-dimensional environment [BER 02] and ensures the positivity of the density. However, it turns out to be quite diffusive, at least more diffusive than the approximate Godunov scheme denoted VFRoe-ncv [BUF 00] [GAL 02] [GAL 03].

The method used to solve the hyperbolic set of equations introduces a coupling between turbulence, mass, momentum and energy which provides an enhanced stability as compared to methods decoupling equations for turbulence, especially for flows at high turbulence levels in impingement regions or afterbodies, as those often encountered in turbomachinery. Application to the flow over ascending steps and around a turbine blade have been carried out to illustrate the method.

More details can be found in the appendix, section 15.

8.2 The Euler/Lagrange approach to turbulent polydispersed two-phase flows

8.2.1 Introduction

Two-phase flows are ubiquitous in many industrial processes or environmental concerns. A very important regime of two-phase flows is characterized by the presence of one phase, either solid, liquid or vapour, as separate inclusions embedded in the other phase. These flows are called dispersed turbulent two-phase flows and the two phases are referred to as the dispersed and the continuous phase respectively. The dispersed regime is always met when the dispersed phase is made up by solid particles (solid particles in a gas or a liquid turbulent flow). It is often found for liquid dispersed as separate droplets in a gas flow (sprays for example) or for two immiscible liquids.
In the Euler/Lagrange approach, the dispersed phase is simulated by tracking a large number of representative particules through the previously computed flow field [MIN 01] [MIP 01]. The objective is to follow the individual and instantaneous

history of each particle present in the carrying flow. However, the history of a given particle is a restrictive information, and generally one wishes to obtain data on the whole of the particle cloud and thus of the dispersed phase; these information are determined by statistical average on the entire or on a part of the particle cloud.
The applicability domain of the Euler/Lagrange approach is wide. However, there is a growing interest in the particle deposit thematic. Therefore, in this section, we focus on ash deposit in an industrial pulverized coal furnace, in the scope of power generation industry.

8.2.2 Pulverized coal furnace slagging

Ash deposit in Q600-type pulverized coal-fired boilers leads in particular to lower efficiency. It is due to deposit of coal particles on shell plates and exchangers. Two types of ash deposits are observed, depending on physical and chemical mechanisms: "slagging" occurs in the radiative zone of the boiler, (*i.e.* the burner region and the hopper), while "fouling" occurs in the convective zone, (*i.e.* essentially at exchanger level).

In the present study, only slagging is investigated [DAL 01]. Indeed, it was beyond reach to represent with sufficient refinement the exchangers, the burners and the burner region within the same three-dimensional calculation. Moreover, deposit due to inertial impact is the only factor taken into account (phenomena such as thermophoresis are neglected).

Calculation with *Code_Saturne* proceeds in two stages. The first one consists in carrying out a simulation of the burner region with an Eulerian approach applied to the equations of the mixture of gas and particles (homogeneous approach). Temperature and flowfield obtained by this way define the carrying phase into which pulverized coal particles are injected and tracked individually with a high accuracy Lagrangian approach (figure 9).

The Lagrangian approach takes into account wall-interactions. Particles following trajectories that lead to a wall become potential candidates to slagging. When a particle collides with the wall, its probability for remaining attached depends, among other things, on the particle speed and angle of incidence, on its viscosity, temperature, size and surface tension as well as on the surface condition of the existing deposit on the wall.

Figure 10 represents the distribution of the slagging mass flux density calculated on the furnace inner walls. The distribution is rather heterogeneous, with areas where slagging is particularly high (up to 50 $g.s^{-1}.m^{-2}$). These areas correspond to privileged regions of impact of coal particles. They are located:

- at the burners, where particles directly emitted from some burners are forming quasi-horizontal spots,
- at the top of the burners zone, under the exchanger system, where wider spots can be observed.

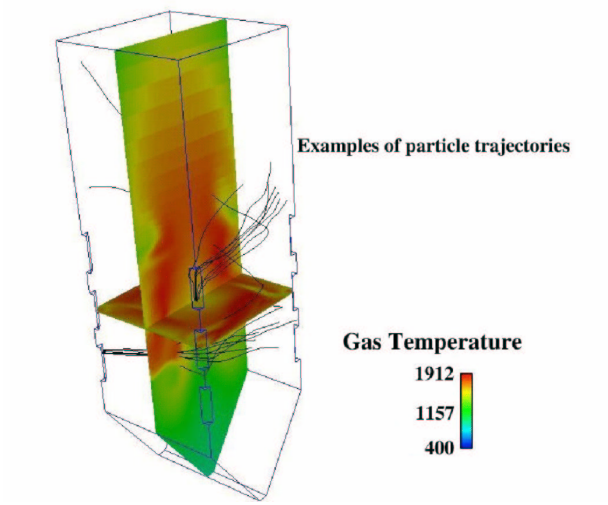


Figure 9: Gas Temperature (Kelvin) - Particles trajectories

This study illustrates the use of the Lagrangian approach as a predictive tool to investigate boiler design with respect to slagging.

8.3 Pollutant emission prediction for gas turbine

Facing more and more stringent environmental requirements and the will to optimize performance of thermal power stations, EDF decided to equip themselves with a simple prediction tool for gas turbine pollutant emissions. However, modelling pollutant formation requires data which are very difficult to obtain from gas turbine operators or constructors. That is why it has proved particularly useful to go through a three-dimensional computation stage in order to obtain relevant data in gas turbine pots and to formulate assumptions better adapted with the study of the formation of pollutant emissions.
In the present study, the capabilities of *Code_Saturne* have been used to model a gas turbine pot similar to the one installed in the thermal power plant of Rio Bravo in Mexico. The three-dimensional results have then been used to run a chemistry code in order to obtain NOx rate emitted by the turbine.

Meshing was the first step to go through. The commercial mesh generator SIMAIL has been used to create four separate meshes, without imposing particular constraints at interfaces. *Code_Saturne* has connected the four parts, dealing automatically with arbitrary interfaces and computations have been carried out using the composite mesh illustrated on figure 11.
Using the same technique, a second mesh representing one eighth of the geometry has been generated with periodic boundary conditions. Satisfactory results have been obtained at a much lower cost than with the complete domain.

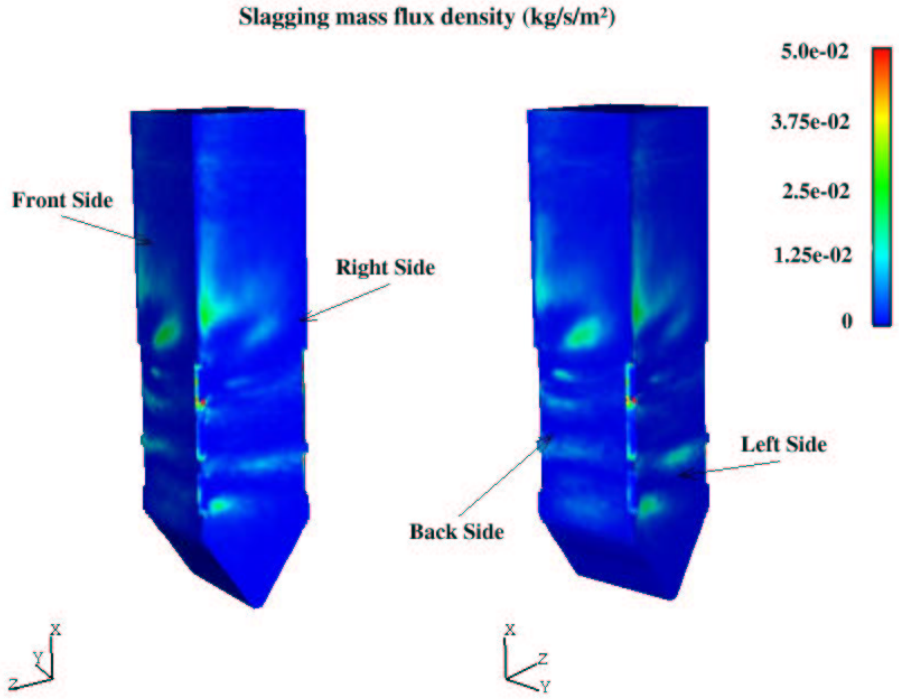


Figure 10: Distribution of the slagging mass flux density

Computations with *Code_Saturne* have been carried out with the basic Eddy Break Up combustion model (an adapted version dealing with variable gas composition at the inlets). Inlet swirl and film cooling have been taken into account in the boundary conditions. Figure 12 presents a snapshot with cutting planes coloured by mass fraction of combustion products.
In the last step of this study, the physical domain has been partitioned into a network of homogeneous reactors, using criteria based on temperature and fuel-air ratio. Some development directly implemented into *Code_Saturne* structure made this task relatively straightforward and allowed to extract from the three-dimensional computation the necessary data for a chemistry code to finally obtain the rate of NOx. This application is a typical illustration of the capabilities of *Code_Saturne* for turbulent reacting flows in complex geometry. It also points out the importance of connecting together different levels of modelling: a major asset of the approach has been the complementarity of the three-dimensional computations for reacting flows and of the refined chemistry analyses on simple networks of homogeneous reactors.

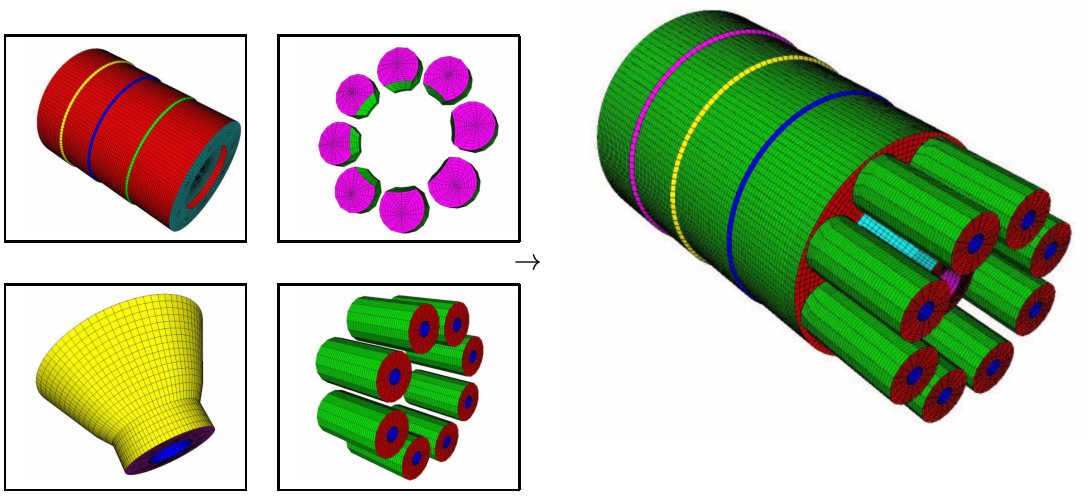


Figure 11: Composite mesh of the gas turbine pot (colours represent different boundary conditions)

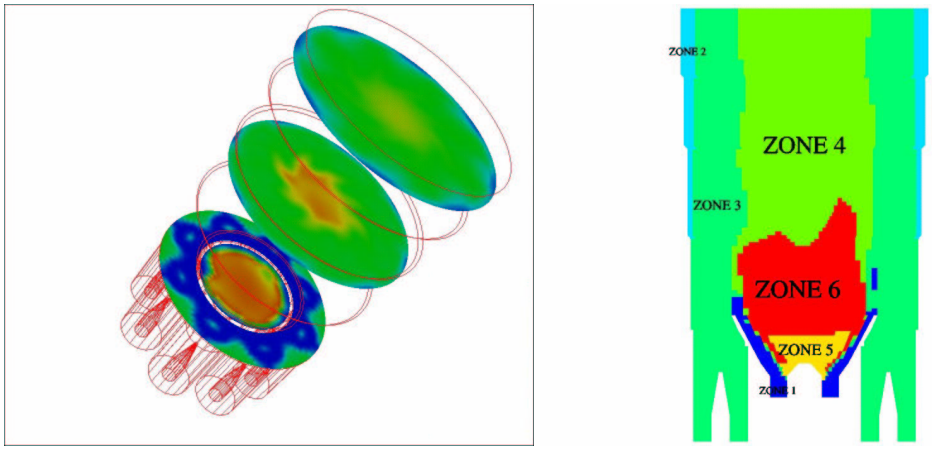


Figure 12: Mass fraction of combustion product (left) - Physical domain partitioning for chemistry analysis (right).

8.4 Ventilation and air conditioning in hospitals

In hospitals and more particularly in operating theaters, air quality is crucial since many infections might be transmitted via the airborne route. C.F.D. allows to analyse the air flow pattern (recirculations, particles trajectories...) and can prove an efficient tool to determine zones where contamination risks are potentially high.

The computational domain illustrated in figure 13 represents an operating theater with the medical staff and the patient. The advanced unstructured mesh capabilities of *Code_Saturne* allow to handle the complexity of the geometry (see the details of the operating lamp arm in figure 13). Heating, ventilation and air conditioning effects (HVAC) are accounted for in the simulations: heat is emitted by people and equipment, clean air is introduced through inlets located at the ceiling while contaminated air is extracted at ground level.

Simulations indicate the significant influence of operating lamps on the air flow pattern above the operating table. Indeed, several recirculations are bound to draw to the patient air that has already circulated in the room (figure 14).
The effects of opening the door to the operating theater have also been simulated. Calculations show that the contamination level in the vicinity of the patient remains limited. On the one hand, the fact that the pressure in the operating theater is kept slightly higher than in the corridor naturally limits the quantity of contaminant introduced in the room. More interesting is the fact that the computations, permitting to determine the local flow pattern, show that the "clean" air flow originating from the inlets located at the ceiling is sufficient to protect the operating zone. Hence, this methodology provides a way of evaluating the efficiency of the ceiling ventilation device in terms of safety for the patient.

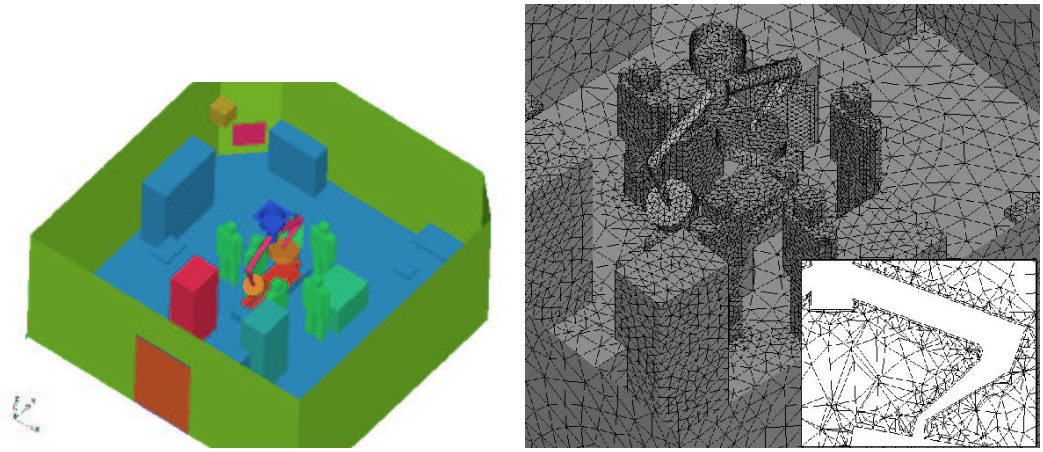


Figure 13: HVAC in operating theater. Computational domain, mesh and detail of the operating lamp arm.

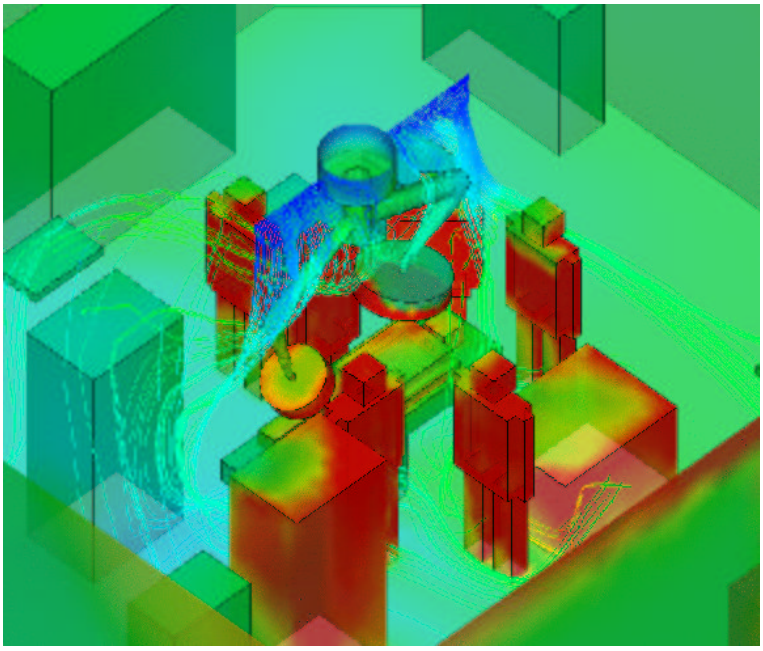


Figure 14: HVAC in operating theater : temperature field and particles trajectories originating from air inlets under the ceiling.

8.5 A thermal shock analysis for nuclear pressure water reactor vessel

8.5.1 Introduction

This section illustrates part of the qualification process of *Code_Saturne* for nuclear single phase thermal-hydraulics studies. [MAR 02] provides a more extensive analysis of the configuration.

To increase Reactor Pressure Vessel (RPV) life time, it is of major importance to assess the integrity of the vessel structure submitted to a particular type of thermal shock that would occur during a "small break" loss of coolant accident (SBLOCA) as a result from the necessary cold water safety injection. Up to now, the assessment of the French reactor vessel integrity has been performed with a specific approach derived from the codified fast fracture analysis (RCCM code) based on a selection of subclad defects and a set of loading transients. A three-dimensional approach for determining the temperature and heat transfer coefficient distributions is a route to a better knowledge and a more refined description of the transient cooling of the vessel. Before starting reactor computations, a preliminary qualification phase was necessary to demonstrate the capabilities of *Code_Saturne* to simulate the physical characteristics encountered in this kind of scenarii. To reach this objective, a well documented experimental set-up has been simulated, namely the CREARE 1/5 scale mixing facility: the geometry, including a cold leg and a plane downcomer, is sufficiently close to that of a RPV for the present application. The numerical simulation

takes into account the transport and mixing of cold water in the -relatively- hot environment through the admission leg (or "cold" leg) and the downcomer. Temperature transient is compared to experiments at several probe locations on the walls representative of the core barrel and of the vessel structure.

8.5.2 Geometry, initial and boundary conditions, fluid properties

The experimental CREARE 1/5 scale facility [ROT 82] consists of a horizontal leg (representative of the reactor "cold" leg) connected to a vertical plane section (representative of one third of a developed downcomer section). The safety injection pipe is connected to the cold leg at an angle of 60 degrees (figure 15).

Initially, the temperature of the fluid contained in the computational domain is 65.5 degrees Celsius. Mass flow rate is zero.

To model the experimental conditions, the far end of the cold leg is represented obturated and a transient fluid temperature is imposed. Constant temperature (17.8 degrees Celsius) and velocity (0.125 m/s) are prescribed at the safety injection inlet. The only outlet of the computational domain is located at the bottom of the downcomer. The other boundaries of the domain are treated as insulated walls.

Water properties are determined at a pressure of 1 bar. Dynamic viscosity, conductivity and specific heat are assumed to be constants while density is a function of temperature only.

8.5.3 Numerical model

The grid, shown in figure 15, has been produced as a result of previous grid sensitivity analyses. It contains 190 000 control volumes. To better adapt the size and orientation of cells, *Code_Saturne* capability to deal with arbitrary interfaces has been employed. The mesh has been generated in a multibloc-unstructured way, with specific refinement close to the injection and at the connection between the cold leg and the downcomer. The use of "O-grids" in the pipes allowed to control the mesh quality close to the walls.
Time marching computations have been carried out using the $k - \varepsilon$ model. The temperature transients obtained at different probe locations have been used to assess the time step influence: very little variation has been observed on computational results produced with a time step reduced by a factor of 2.

8.5.4 Qualitative results analysis

As expected, the cold injection creates a stratification in the cold leg. The interface between cold and hot fluid extends towards the downcomer and towards the far end of the leg (figure 16). As it reaches the downcomer, a separation occurs: the cold water jet is driven against the core barrel and does not remain attached to the vessel side. Lateral oscillations can be observed in the downcomer, straight under the cold leg (figure 17). They originate in unsteady detachment, recirculations and buoyancy

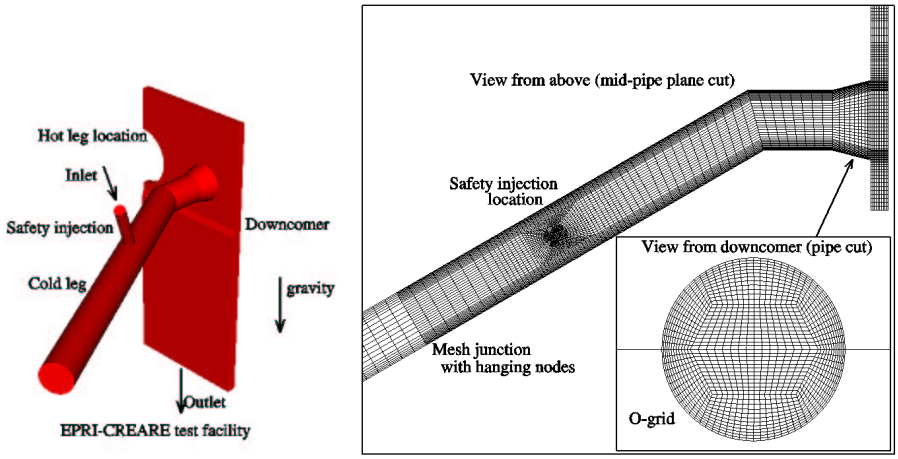


Figure 15: CREARE: views of the geometry and grid.

effects.

Anticipating over future reactor simulations for evaluating thermal loading, it must be noted that flow separation driving cold water jet away from the vessel wall and oscillations of large structures improving water mixing are two important phenomena that shall tend to reduce the severity of the thermal shock undergone by the vessel.

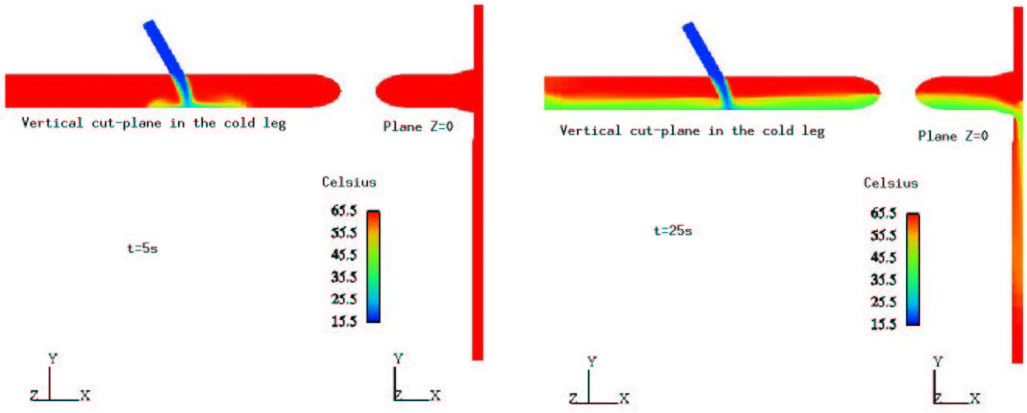


Figure 16: Stratification transient progress in cold leg and flow separation in downcomer (domain is coloured by temperature in Celsius, snapshots have been taken at 5 and 25 seconds after injection started).

8.5.5 Quantitative results analysis

So as to add a quantitative side to the qualification of *Code_Saturne*, temperature transients have been compared to experimental data obtained on the CREARE facility. We will concentrate here on a few probes only ([MAR 02] provides a more detailed analysis and comparisons to other C.F.D. codes). The subsequent consid-

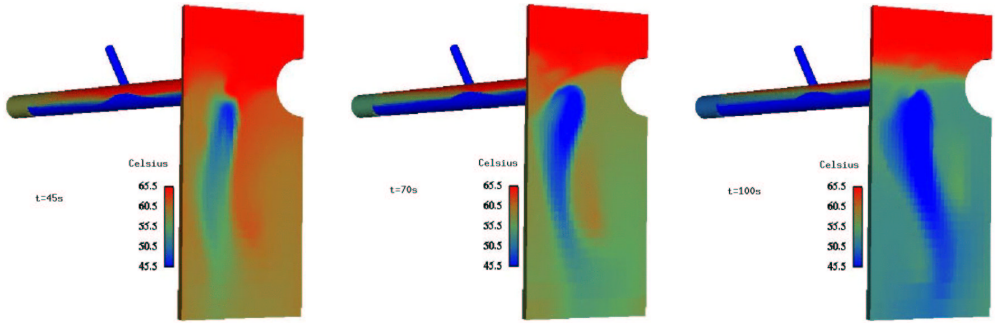


Figure 17: Oscillations in the downcomer (domain is coloured by temperature in Celsius, snapshots have been taken at 45, 70 and 100 seconds after injection started).

erations will be based on graphs presenting fluid temperature (Celsius) against time (seconds) obtained at the locations shown on figure 18.

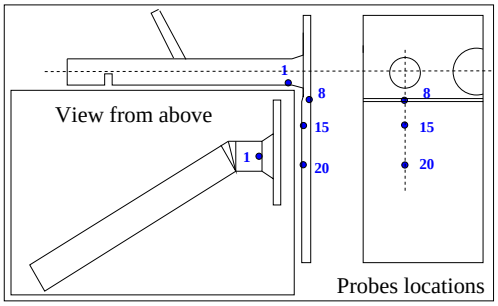


Figure 18: Location of some probes (thermocouples).

At probe 1, figure 19 indicates that the transient progress of the stratification in the cold leg is well reproduced. At probe 8, the transient is in good agreement with experiments: the cold fluid impact on the core barell, below the cold leg, is captured. At probes 15 and 20, (located at a lower position in the downcomer), the temperature decrease is also well predicted.

8.5.6 Conclusions

The calculations presented here illustrate the integration of a C.F.D. tool in an industrial approach with nuclear safety at stake. To complete the qualification of *Code_Saturne* for this configuration, additional sensitivity tests have been carried out and results have been compared to those produced by other commercial codes for many more probe locations. This work has finally demonstrated the capability of *Code_Saturne* to represent some important phenomena that need to be accounted for to evaluate accurately transient thermal loading on RPV in small break loss of coolant accident conditions.

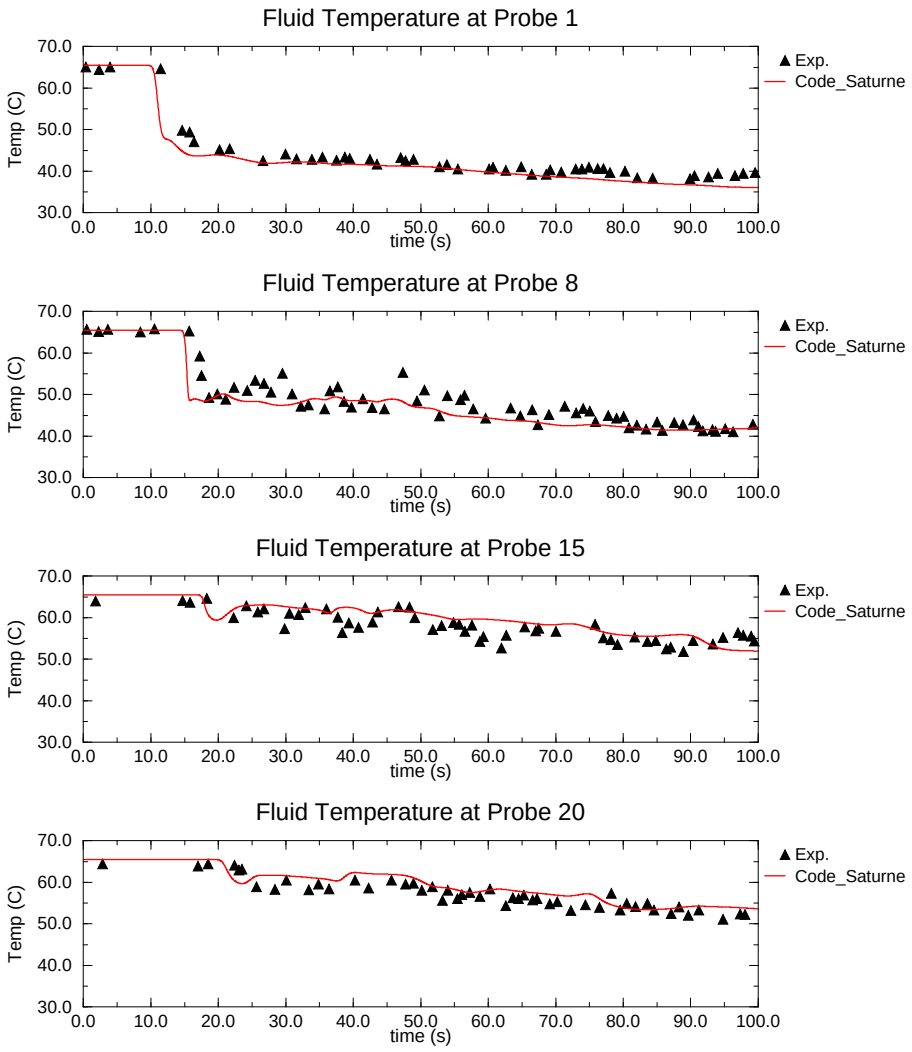


Figure 19: CREARE facility simulation. Fluid temperature transient at selected probes locations.

8.6 Large Eddy Simulation applied to thermal fatigue

8.6.1 Introduction

Thermal fatigue of the coolant circuits of pressure water reactor (P.W.R.) plants is a major issue for nuclear safety. The problem is especially acute in mixing zones, like T-junctions, where a large difference of water temperature between the two entry branches and the high level of turbulence can lead to large temperature fluctuations at the wall. Until recently, studies on the matter had been tackled at EDF using steady R.A.N.S. methods: the fluid flow is solved with a C.F.D. code using an averaged model, which leads to the knowledge of the mean temperature and temperature variance at each point of the wall. Based on these results, thermomechanic computations are performed to obtain the usage factor. However, these methods obliterate any fluid unsteady effects (like phase lag in the temperature oscillations between two points, which can generate important stresses, or the high frequency filtering effect of wall thermal inertia, which noticeably changes fluid-wall heat transfer). Hence, a new methodology of chained computations has been proposed, that allows to take into account these unsteady and three-dimensional effects. It has been applied to a Residual Heat Removal (RHR) system where cracks have been found and identified as originating from thermal fatigue. *Code_Saturne* is used to compute the fluid flow with a L.E.S. model, which gives access to the instantaneous temperature fluctuations. *Code_Saturne* is coupled to the thermal code SYRTHES, which propagates the temperature fluctuations into the wall thickness. Then, the instantaneous temperature field inside the wall is transfered to *Code_Aster*, EDF thermomechanic code, to perform mechanical computations and yield the instantaneous mechanical stresses undergone by the T-junction and the following elbow. The results of this first study are promising, although this method is still largely under development and validation. See paper [PEN 03] for full details.

8.6.2 Configuration

The global geometry of the RHR circuit is presented in figure 20. Due to the difficulty and the cost of a C.F.D. approach using Large Eddy Simulation, only the small portion surrounding the location where cracks have occured has been simulated (red rectangle in figure 20).
In this specific study, the horizontal hot branch and the vertical cold branch are set respectively at a temperature of 168 degrees Celsius and 41 degrees Celsius. The total flow rate is set to be 1000 m^3/h and the velocity ratio between the cold flow branch and the total flow rate is 20%. For the whole study, two meshes have to be designed: one for the fluid domain, and the other for the solid domain. The fluid mesh contains 401 472 hexaedric cells, while the solid mesh contains 688 320 tetrahedra, with a very fine discretization near the inner wall, in order to properly capture the instantaneous temperature variations.

8.6.3 Computational results

This calculation gives access to unsteady results on the entire solid and fluid domains. 10 seconds of physical time have been simulated, which already corresponds to a very

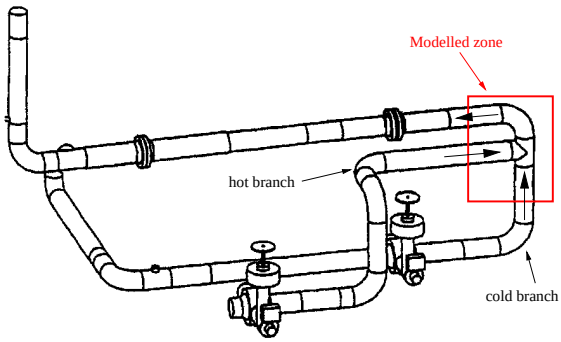


Figure 20: Sketch of the RHR at Civaux 1 (former configuration) and modelled zone

large computational effort (around 1 200 hours of CPU on a Fujitsu VPP5000 vector machine). All thermal results are presented in non-dimensional variables; let T_{cold} and T_{hot} be the cold and hot temperatures in the branches, the non-dimensional temperature is calculated as T_{adim} (varying between 0 and 1):

$$T_{adim} = \frac{T - T_{cold}}{T_{hot} - T_{cold}}$$

Figure 21 gives a snapshot of the fluid instantaneous temperature field in the symmetry plane at the physical time $t = 4\ s$. One may notice the very complex hot and cold structures, and especially those created at the the junction of the two jets, and convected further away. Figure 22 shows the instantaneous temperature in the solid, also at time $t = 4\ s$. It is clear that the field is much smoother than in the fluid, because of the strong attenuation due to the thermal inertia of the wall.

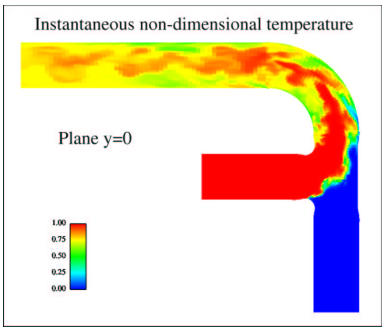


Figure 21: Instantaneous fluid temperature field in the symmetry plane ($t = 4\ s$)

In the study on thermal fatigue, the heat transfer from the fluid to the wall is a critical issue. Therefore, we shall now focus on points located at the fluid/solid interface. Figure 23 compares the fluid and solid temperature profiles at interface point C12 (cf. figure 22). It reveals the strong attenuation of the wall temperature signal compared to the very fluctuating temperature signal of the fluid in the near

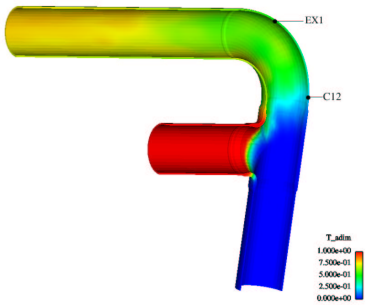


Figure 22: Instantaneous solid temperature field ($t = 4\ s$)

wall region. This attenuation is due to the thermal inertia of the wall and is strongly dependent on the frequency of the fluctuations. This dependence is naturally reproduced by our unsteady simulations, whereas it is totally unreachable through approaches based on Reynolds-averaged Navier-Stokes equations (R.A.N.S.).

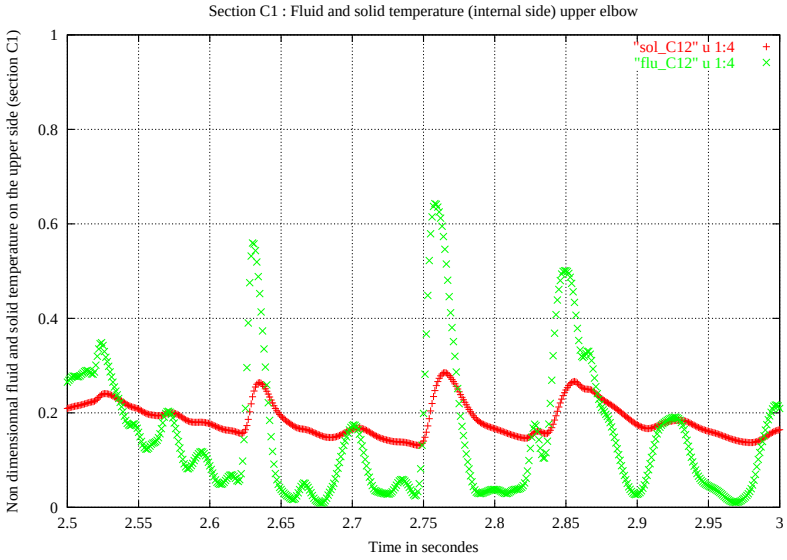


Figure 23: Fluid (green) and solid (red) temperature evolution at fluid/solide interface at point C12

The results of these simulations can also be compared to experimental data taken from on-site measurements. It should yet be underlined that on-site recording of the temperature inside the pipe is not possible. Hence, the measurements of temperature in the pipe at the fluid/solid interface are in fact obtained indirectly from deformation probes on the outside part of the wall. Also, the characteristics of the on-site configuration differ slightly from the simulation (22.2% on-site velocity ratio instead of 20% and 1025 m^3/h on-site flow instead of 1000 m^3/h). Moreover the locations of the probes differ slightly, since it is impossible on-site to install probes

on the symmetry plane. Figures 24 shows temperature profiles (over 3 seconds of physical time) taken from experimental data and from the simulation, at point C12. It appears that the turbulent fluctuating phenomena seem to be well captured, that fluctuation amplitude seems to be of the right order and that the experimental and simulated signals seem to have similar frequencies. Similar results are obtained on the other measurement points.

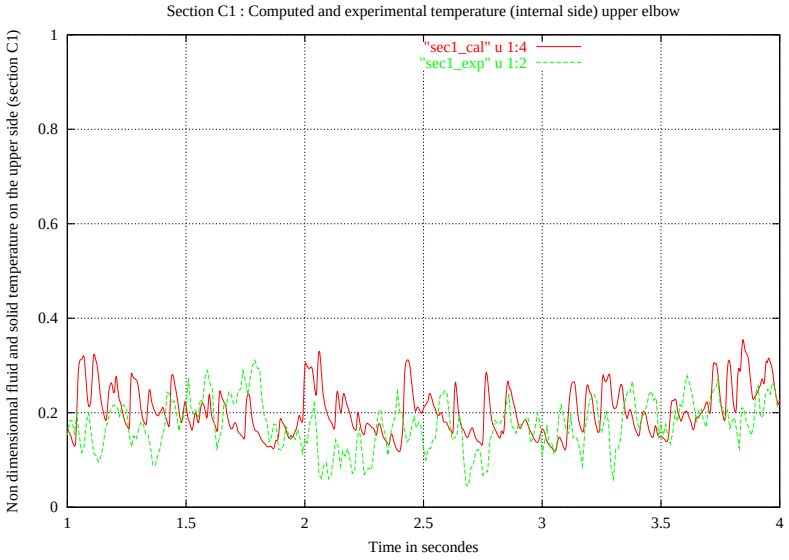


Figure 24: Experimental (green) and simulated (red) solid temperature profiles at fluid/solid interface, at point C12

Eventually, since the approach gives access to the instantaneous temperature evolution, one may also apply usual spectral analysis. Figure 25 presents the Spectral Density Power obtained from experimental signals (on-site and a mock-up) and from the L.E.S. approach for the probe located at the solid/fluid interface at point EX1 (cf. figure 22). The experimental spectrum seems slightly smoother. This is due to the fact that the experimental SDP has been done on a 600 s period while the L.E.S one is based only on a 3 s time interval, which forbids the use of any averaging spectral technique. Nevertheless, the different profiles seem in good agreement, and none of them shows any specific frequency peak.

8.6.4 Conclusion and perspectives

In the study presented here, *Code_Saturne* has been used to predict the three-dimensional unsteady temperature fluctuations in a T-junction using Large Eddy Simulation for the fluid domain and a full coupling with thermal code SYRTHES to propagate the temperature in the wall thickness. These solid thermal fields will then be exploited with a mechanical tool to obtain thermally induced stresses. The L.E.S. approach seems very promising when compared to experimental data, to get a better understanding and knowledge of the instantaneous thermal loads seen by

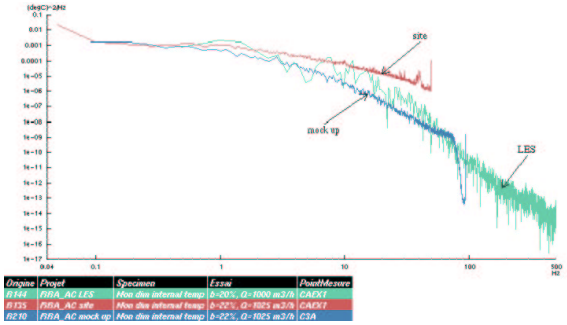


Figure 25: Experimental and L.E.S. solid temperature SDP at point EX1

pipeworks, although it is still very costly in terms of CPU and computing memory. Further studies are planned at EDF to improve CPU aspects and confirm on other industrial configurations the ability and the high interest of the thermally coupled approach presented here.

9 General conclusions and perspectives for *Code_Saturne*

We have described the finite volume method implemented in *Code_Saturne* for laminar and turbulent flows in complex two- and three-dimensional geometries. In time, a predictor-corrector method is used for the Navier-Stokes equations. The spatial discretization is based on the fully conservative, unstructured finite volume framework, with a fully co-located arrangement for all variables. To meet the requirements of the industrial target applications, algorithms for mesh management have been developed that allow to use hybrid meshes containing arbitrary interfaces and any kind of cells. Hence, specific effort has also been devoted to implement satisfactory schemes to compute gradients at cell centres on a wide range of unstructured meshes.

Since 2000, the code is being used for industrial applications and research activities in several fields related to energy production (nuclear power thermal-hydraulics, gas and coal combustion, turbomachinery, heating, ventilation and air conditioning). The requirement for improving robustness and precision of numerical schemes becomes all the more apparent as the complexity of physical modelling increases. Apart from fundamental work on robustness and precision (for example for gradient calculation on arbitrary meshes), future developments will focus on Large Eddy Simulation, especially for unsteady thermal effects on structures (thermal fatigue, crackling). Nevertheless, R.A.N.S. modelling will receive further attention (v2f [DUR 93] [LID 96], $k-\omega$ [MEN 94], SSG [SSG 91], ...). Model and code coupling are also domains of interest that will require attention (0D-3D coupling, dynamic and thermal fluid-structure interaction, zonal modelling coupled to refined simulation, L.E.S./R.A.N.S. coupling, Euler/Lagrange coupling).

Eventually, the use of *Code_Saturne* as a general-purpose C.F.D. code for production

and as a development platform will be further extended to a community of industrial partners and universities determined to participate in R&D work on numerical schemes and physical modelling.

To supplement the panorama, it is pointed out that for historical reasons, the domain of application addressed by *Code_Saturne* is essentially limited to single-phase flows, with extensions to multiphase problems where homogeneous approaches are valid (pulverized coal for example) or for which a Lagrangian method is indicated (particle deposition for example). Nevertheless, *Code_Saturne* is also the basis for *Mercure_Saturne*, EDF software for atmospheric simulations. Moreover, the structure of *Code_Saturne* and the key features of the algorithm have been used as a starting point for an Eulerian multiphase version, *Saturne_Polyphasique*. The latter has since then benefited from a sustained development effort. At the present time, *Saturne_Polyphasique* constitutes the basis for the refined three-dimensional thermal-hydraulic steam-water flow simulations within EDF-CEA joint development programme "NEPTUNE" [DEL03] [CHA03] [MEC03]. The code also contributes to the research work carried out at the Institut de Mécanique des Fluides de Toulouse on various multiphase flows, with gas or solid inclusions [SSV 04] [FPS 04].

10 Ackowledgements

The authors would like to underline that this paper is based on the work of many other contributors, from developer to end user, to whom they are most grateful, and especially: Sofiane Benhamadouche, Ludovic Bénito, Alexandre Davroux, Alexandre Douce, Alain Escaich, Yvan Fournier, Brigitte Gest, Alain Martin, Guillaume Périgaud, Jean-Luc Rousset and Ugo Schück.
The authors also wish to express their sincere gratitude to Jean-Marc Hérard, Dominique Laurence, Jean-Pierre Minier, Christophe Péniguel, Pierre Plion and Sandro Dal Secco for their pertinent advice.
Eventually, the authors would like to point out that this work has been possible thanks to EDF strategic choices and R&D funding for developing C.F.D. software and expertise on numerical methods.

11 Appendix A: continuous equations for turbulence modelling

11.1 Introduction

To close system (12), turbulent correlations need to be modelled. A fairly simple but widely used approach for industrial applications relies on eddy-viscosity models. We describe hereafter the "standard" high-Reynolds-number $k-\varepsilon$ model of Launder and Spalding [LAU 74]. Nevertheless, no eddy-viscosity model, however elaborate, can account for anisotropy of turbulence, which might prove of major importance in many applications including curvature, density stratification or swirl. For those types of flows, a second moment closure can yield decisive benefits and the second

model presented hereafter is based on the proposal by Launder, Reece and Rodi [LRR 75]. Both models are operated with log-law-based wall functions.

11.2 Two-equation model

With the $k-\varepsilon$ model proposed by Launder and Spalding [LAU 74] (see also [MOP 94]), k denoting the turbulent kinetic energy and ε the dissipation rate of turbulent kinetic energy, the velocity correlations are modelled in 3D as follows:

$$-\rho\underline{\underline{R}} = 2\,\mu_t\,\underline{\underline{D}} - \frac{2}{3}\mu_t\,tr(\underline{\underline{D}})\,\underline{\underline{Id}} - \frac{2}{3}\rho k\underline{\underline{Id}} \tag{61}$$

Thus, the turbulent kinetic energy k is defined as:

$$k = \frac{1}{2}tr(\underline{\underline{R}}) = \frac{1}{2}R_{ii} \tag{62}$$

The turbulent viscosity μ_t appearing in (61) is modelled as:

$$\mu_t = \rho\,C_\mu\frac{k^2}{\varepsilon} \tag{63}$$

The two equations of the model read:

$$\left\{\begin{array}{rcl} \dfrac{\partial}{\partial t}(\rho k) + \mathrm{div}\left[\rho\underline{u}\,k - \left(\mu + \dfrac{\mu_t}{\sigma_k}\right)\underline{\mathrm{grad}}\;k\right] & = & \mathcal{P} + \mathcal{G} - \rho\varepsilon + S_k \\ \dfrac{\partial}{\partial t}(\rho\varepsilon) + \mathrm{div}\left[\rho\underline{u}\,\varepsilon - \left(\mu + \dfrac{\mu_t}{\sigma_\varepsilon}\right)\underline{\mathrm{grad}}\;\varepsilon\right] & = & C_{\varepsilon_1}\dfrac{\varepsilon}{k}\left[\mathcal{P} + (1-C_{\varepsilon_3})\mathcal{G}\right] \\ & & -\rho\,C_{\varepsilon_2}\dfrac{\varepsilon^2}{k} + S_\varepsilon \end{array}\right. \tag{64}$$

In system (64), $\mathcal{P}$ accounts for production of kinetic energy through mean shear stress. In a three-dimensional context, we have:

$$\begin{aligned} \mathcal{P} &= -\rho R_{ij}\frac{\partial u_i}{\partial x_j} \\ &= \left[\mu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\mu_t\frac{\partial u_k}{\partial x_k}\delta_{ij} - \frac{2}{3}\rho k\delta_{ij}\right]\frac{\partial u_i}{\partial x_j} \\ &= \mu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\frac{\partial u_i}{\partial x_j} - \frac{2}{3}\mu_t(\mathrm{div}\underline{u})^2 - \frac{2}{3}\rho k\mathrm{div}(\underline{u}) \\ &= \mu_t\left[2\left(\frac{\partial u_1}{\partial x_1}\right)^2 + 2\left(\frac{\partial u_2}{\partial x_2}\right)^2 + 2\left(\frac{\partial u_3}{\partial x_3}\right)^2\right. \\ &\quad \left. + \left(\frac{\partial u_1}{\partial x_2} + \frac{\partial u_2}{\partial x_1}\right)^2 + \left(\frac{\partial u_1}{\partial x_3} + \frac{\partial u_3}{\partial x_1}\right)^2 + \left(\frac{\partial u_2}{\partial x_3} + \frac{\partial u_3}{\partial x_2}\right)^2\right] \\ &\quad - \frac{2}{3}\mu_t(\mathrm{div}\underline{u})^2 - \frac{2}{3}\rho k\mathrm{div}(\underline{u}) \end{aligned}$$

C_μ	C_{ε_1}	C_{ε_2}	C_{ε_3}	σ_k	σ_ε
0.09	1.44	1.92	1 or 0	1	1.3

Table 1: Constants used for the $k-\varepsilon$ model.

$\mathcal{G}$ is the turbulent kinetic energy production-destruction term related to density effects [ROD 84]. With $\underline{g}$ denoting the gravity, the exact expression for $\mathcal{G}$ is then $\mathcal{G} = \overline{\rho'' \underline{u}''} \cdot \underline{g}$. Under the assumption that ρ is a function of the intensive scalar a only (that is: $\rho = \mathcal{F}(a)$), the following approximation might be used:

$$\mathcal{G} = \overline{a'' \frac{d\mathcal{F}}{d\,a} \underline{u}''} \cdot \underline{g} \tag{65}$$

Assuming $\dfrac{d\mathcal{F}}{d\,a}$ constant, denoting $\sigma_{a,t}$ the turbulent Prandtl-Schmidt number and using a gradient hypothesis for scalar flux modelling (equation (82)), we have:

$$\begin{aligned} \mathcal{G} &= -\frac{d\mathcal{F}}{d\,a}\frac{1}{\rho}\frac{\mu_t}{\sigma_{a,t}} \underline{\mathrm{grad}}\ (a) \cdot \underline{g} \\ &= -\frac{1}{\rho}\frac{\mu_t}{\sigma_{a,t}} \underline{\mathrm{grad}}\ (\rho) \cdot \underline{g} \end{aligned} \tag{66}$$

S_k and S_ε are additional source terms for k and ε respectively (they also include sources originating from the mass source term Γ).

The constants pertaining to the model are given in table 1. The value for C_{ε_3} depends on the nature of the stratified configuration: $C_{\varepsilon_3} = 0$ if $\mathcal{G} \geqslant 0$ (unstable stratification) and $C_{\varepsilon_3} = 1$ if $\mathcal{G} \leqslant 0$ (stable stratification). The rationale for considering gravity effects in the equation for ε only for positive values of $\mathcal{G}$ (unstable stratification) is inherited from proposals by Gibson and Launder [GIB 76], also applied later by Viollet [VIO 80].

11.3 Second moment closure

The second moment closure implemented in *Code_Saturne* is based on the following equations for each component of the symmetric Reynolds stress tensor and for the dissipation rate of turbulent kinetic energy ε (see [DPR 01]):

$$\left\{ \begin{array}{lcl} \dfrac{\partial}{\partial t}(\rho R_{ij}) + \mathrm{div}(\rho \underline{u}\, R_{ij} - \mu\, \underline{\mathrm{grad}}\ R_{ij}) & = & \mathcal{P}_{ij} + \mathcal{G}_{ij} + \Phi_{ij} + d_{ij} - \rho\varepsilon_{ij} + S_{ij} \\ \dfrac{\partial}{\partial t}(\rho\varepsilon) + \mathrm{div}(\rho \underline{u}\, \varepsilon - \mu\, \underline{\mathrm{grad}}\ \varepsilon) & = & d_\varepsilon + C_{\varepsilon_1}\dfrac{\varepsilon}{k}\left[\mathcal{P} + \mathcal{G}_\varepsilon\right] - \rho C_{\varepsilon_2}\dfrac{\varepsilon^2}{k} + S_\varepsilon \end{array} \right. \tag{67}$$

$\underline{\underline{\mathcal{P}}}$ is the kinetic energy production-destruction tensor related to mean shear stress. It does not requires any modelling:

$$\mathcal{P}_{ij} = -\rho\left[R_{ik}\frac{\partial u_j}{\partial x_k} + R_{jk}\frac{\partial u_i}{\partial x_k}\right] \tag{68}$$

$\underline{\underline{\mathcal{G}}}$ is the production-destruction tensor related to gravity effects [ROD 84]:

$$\mathcal{G}_{ij} = \left[G_{ij} - C_3 (G_{ij} - \frac{1}{3}\delta_{ij} G_{ll}) \right] \tag{69}$$

With the same assumptions as those made to derive equation (66), denoting g_i the gravity components, we have:

$$\left\{ \begin{array}{l} G_{ij} = -\dfrac{3}{2}\dfrac{C_\mu}{\sigma_{a,t}}\dfrac{k}{\varepsilon}(r_i g_j + r_j g_i) \\ \text{with} \quad r_i = R_{ik}\dfrac{\partial \rho}{\partial x_k} \end{array} \right. \tag{70}$$

The pressure-strain term $\underline{\underline{\Phi}}$ and the dissipation tensor $\rho\underline{\underline{\varepsilon}}$ are modelled as follows:

$$\Phi_{ij} - \rho\varepsilon_{ij} = \phi_{ij,1} + \phi_{ij,2} + \phi_{ij,w} - \frac{2}{3}\rho\ \delta_{ij}\varepsilon \tag{71}$$

The first term $\phi_{ij,1}$ is often referred to as the "return-to-isotropy" term, since its major role seems to promote a return to isotropy of the Reynolds stress tensor. The following modelling has been adopted [ROT 51]:

$$\phi_{ij,1} = -\rho\, C_1 \frac{\varepsilon}{k}(R_{ij} - \frac{2}{3}k\delta_{ij}) \tag{72}$$

Although the "rapid term" $\phi_{ij,2}$ has received considerable attention over the years, the rather simple modelling proposed by Naot *et al.* [NAO 70] ("isotropisation of production") has generally been found to provide good results over a wide range of flows. This is the modelling adopted here:

$$\phi_{ij,2} = -\rho\, C_2 (\mathcal{P}_{ij} - \frac{2}{3}\mathcal{P}\delta_{ij}) \tag{73}$$

with:

$$\mathcal{P} = \frac{1}{2}\mathcal{P}_{ll} \tag{74}$$

The last term $\phi_{ij,w}$ is often referred to as a wall-echo term. In a channel flow, it tends to diminish the autocorrelation of the fluctuations of the velocity component normal to the wall. By default, this term is not activated in *Code_Saturne*. With $l = \dfrac{k^{\frac{3}{2}}}{\varepsilon}$ a turbulent lenght scale and y the distance to the nearest wall, $\phi_{ij,w}$ reads:

$$\begin{array}{rl} \phi_{ij,w} = & \rho\, C_1' \dfrac{k}{\varepsilon}\left[R_{km} n_k n_m \delta_{ij} - \dfrac{3}{2} R_{ki} n_k n_j - \dfrac{3}{2} R_{kj} n_k n_i \right] f(\dfrac{l}{y}) \\ & + \rho\, C_2' \left[\phi_{km,2} n_k n_m \delta_{ij} - \dfrac{3}{2}\phi_{ki,2} n_k n_j - \dfrac{3}{2}\phi_{kj,2} n_k n_i \right] f(\dfrac{l}{y}) \end{array} \tag{75}$$

The damping function f is unity at the wall and vanishes away from it:

$$f(\frac{l}{y}) = \min(1,\ C_\mu^{0,75}\ \frac{k^{\frac{3}{2}}}{\varepsilon\ \kappa y}) \tag{76}$$

C_μ	C_ε	C_{ε_1}	C_{ε_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 2: Constants used for the second moment closure.

The turbulent diffusion term d_{ij} is modelled according to Daly and Harlow [DAL 70]:

$$d_{ij} = C_S \frac{\partial}{\partial x_k}(\rho \frac{k}{\varepsilon} R_{km} \frac{\partial R_{ij}}{\partial x_m}) \tag{77}$$

The equation for the dissipation rate ε is similar to that derived for the $k-\varepsilon$ model. The only modifications are related to turbulent diffusion and gravity effects:

$$d_\varepsilon = C_\varepsilon \frac{\partial}{\partial x_k}\left(\rho \frac{k}{\varepsilon} R_{km} \frac{\partial \varepsilon}{\partial x_m}\right) \tag{78}$$

$$\mathcal{G}_\varepsilon = \max(0, \frac{1}{2} G_{ll}) \tag{79}$$

For the same reasons as those invoked for the $k-\varepsilon$ model, $\mathcal{G}_\varepsilon$ vanishes for negative values of G_{ll} (stable stratification).

S_{ij} and S_ε are additional source terms for R_{ij} and ε respectively (they also include sources originating from the mass source term Γ).

The constants pertaining to the model are given in table 2.

11.4 Sets of equations for turbulence: summary

For both models, the mass equation is used to exhibit the time derivative of the variables k, ε and R_{ij} (equation (10)). The source term is modified as indicated in equation (11). The final equations for the $k-\varepsilon$ model and the second moment closure are the following:

$$\left\{ \begin{array}{rcl} \rho \dfrac{\partial k}{\partial t} + \mathrm{div}\left[\rho \underline{u}\, k - \left(\mu + \dfrac{\mu_t}{\sigma_k}\right) \underline{\mathrm{grad}}\ k\right] & = & \mathcal{P} + \mathcal{G} - \rho\varepsilon + S_k' \\ \rho \dfrac{\partial \varepsilon}{\partial t} + \mathrm{div}\left[\rho \underline{u}\, \varepsilon - \left(\mu + \dfrac{\mu_t}{\sigma_\varepsilon}\right) \underline{\mathrm{grad}}\ \varepsilon\right] & = & C_{\varepsilon_1} \dfrac{\varepsilon}{k}\left[\mathcal{P} + (1-C_{\varepsilon_3})\mathcal{G}\right] - \rho C_{\varepsilon_2} \dfrac{\varepsilon^2}{k} + S_\varepsilon' \end{array} \right. \tag{80}$$

$$\left\{ \begin{array}{rcl} \rho \dfrac{\partial R_{ij}}{\partial t} + \mathrm{div}(\rho \underline{u}\, R_{ij} - \mu\, \underline{\mathrm{grad}}\ R_{ij}) & = & \mathcal{P}_{ij} + \mathcal{G}_{ij} + \Phi_{ij} + d_{ij} - \rho\varepsilon_{ij} + S_{ij}' \\ \rho \dfrac{\partial \varepsilon}{\partial t} + \mathrm{div}(\rho \underline{u}\, \varepsilon - \mu\, \underline{\mathrm{grad}}\ \varepsilon) & = & d_\varepsilon + C_{\varepsilon_1} \dfrac{\varepsilon}{k}\left[\mathcal{P} + \mathcal{G}_\varepsilon\right] - \rho C_{\varepsilon_2} \dfrac{\varepsilon^2}{k} + S_\varepsilon' \end{array} \right. \tag{81}$$

11.5 Scalar flux modelling

Turbulent scalar flux is modelled in *Code_Saturne* by a simple gradient hypothesis:

$$-\overline{\rho a'' \underline{u}''} = \frac{\mu_t}{\sigma_{a,t}} \underline{\text{grad}}\ a \tag{82}$$

where $\sigma_{a,t}$ is the turbulent Prandtl-Schmidt number.

12 Appendix B: simultaneous resolution of turbulence equations for the $k-\varepsilon$ model

For the $k-\varepsilon$ model, equations for k and ε are solved simultaneously in order to partially take into account the equilibrium between these two variables. The algorithm is divided in three steps.
Let us rewrite the system (17) as follows:

$$\left\{ \begin{array}{l} \rho \dfrac{\partial k}{\partial t} = D(k) + \mathcal{S}_k(k,\varepsilon) \\ \\ \rho \dfrac{\partial \varepsilon}{\partial t} = D(\varepsilon) + \mathcal{S}_\varepsilon(k,\varepsilon) \end{array} \right. \tag{83}$$

D is the convection/diffusion operator. $\mathcal{S}_k$ (resp. $\mathcal{S}_\varepsilon$) are the source terms for k (resp. ε), including production and dissipation terms.
STEP 1: explicit resolution
The system is solved explicitly, yielding k_e and ε_e:

$$\left\{ \begin{array}{l} \rho^{(n)} \dfrac{k_e - k^{(n)}}{\Delta t} = D(k^{(n)}) + \mathcal{S}_k(k^{(n)}, \varepsilon^{(n)}) \\ \\ \rho^{(n)} \dfrac{\varepsilon_e - \varepsilon^{(n)}}{\Delta t} = D(\varepsilon^{(n)}) + \mathcal{S}_\varepsilon(k^{(n)}, \varepsilon^{(n)}) \end{array} \right. \tag{84}$$

STEP 2: coupling of the equations through the source terms
The source terms in each equation are written as follows:

$$\left\{ \begin{array}{l} \rho^{(n)} \dfrac{k_{st} - k^{(n)}}{\Delta t} = D(k^{(n)}) + \mathcal{S}_k(k_{st}, \varepsilon_{st}) \\ \\ \rho^{(n)} \dfrac{\varepsilon_{st} - \varepsilon^{(n)}}{\Delta t} = D(\varepsilon^{(n)}) + \mathcal{S}_\varepsilon(k_{st}, \varepsilon_{st}) \end{array} \right. \tag{85}$$

which yields, subtracting (84) from (85):

$$\left\{ \begin{array}{l} \rho^{(n)} \dfrac{k_{st} - k^{(n)}}{\Delta t} = \rho^{(n)} \dfrac{k_e - k^{(n)}}{\Delta t} + \mathcal{S}_k(k_{st}, \varepsilon_{st}) - \mathcal{S}_k(k^{(n)}, \varepsilon^{(n)}) \\ \\ \rho^{(n)} \dfrac{\varepsilon_{st} - \varepsilon^{(n)}}{\Delta t} = \rho^{(n)} \dfrac{\varepsilon_e - \varepsilon^{(n)}}{\Delta t} + \mathcal{S}_\varepsilon(k_{st}, \varepsilon_{st}) - \mathcal{S}_\varepsilon(k^{(n)}, \varepsilon^{(n)}) \end{array} \right. \tag{86}$$

Then, one can write (for $\varphi = k$ or ε):

$$\mathcal{S}_\varphi(k_{st},\varepsilon_{st}) - \mathcal{S}_\varphi(k^{(n)},\varepsilon^{(n)}) \approx (k_{st}-k^{(n)}) \left(\frac{\partial \mathcal{S}_\varphi}{\partial k}\right)^{(n)} + (\varepsilon_{st}-\varepsilon^{(n)}) \left(\frac{\partial \mathcal{S}_\varphi}{\partial \varepsilon}\right)^{(n)} \quad (87)$$

where

$$\left(\frac{\partial \mathcal{S}_\varphi}{\partial k}\right)^{(n)} = \frac{\partial \mathcal{S}_\varphi}{\partial k}(k^{(n)},\varepsilon^{(n)}) \qquad \text{and} \qquad \left(\frac{\partial \mathcal{S}_\varphi}{\partial \varepsilon}\right)^{(n)} = \frac{\partial \mathcal{S}_\varphi}{\partial \varepsilon}(k^{(n)},\varepsilon^{(n)})$$

Which leads to a 2×2 linear system to be solved in each cell:

$$\begin{pmatrix} \dfrac{\rho^{(n)}}{\Delta t} - \left(\dfrac{\partial \mathcal{S}_k}{\partial k}\right)^{(n)} & -\left(\dfrac{\partial \mathcal{S}_k}{\partial \varepsilon}\right)^{(n)} \\ -\left(\dfrac{\partial \mathcal{S}_\varepsilon}{\partial k}\right)^{(n)} & \dfrac{\rho^{(n)}}{\Delta t} - \left(\dfrac{\partial \mathcal{S}_\varepsilon}{\partial \varepsilon}\right)^{(n)} \end{pmatrix} \begin{pmatrix} (k_{st}-k^{(n)}) \\ (\varepsilon_{st}-\varepsilon^{(n)}) \end{pmatrix} = \begin{pmatrix} \rho^{(n)} \dfrac{k_e - k^{(n)}}{\Delta t} \\ \rho^{(n)} \dfrac{\varepsilon_e - \varepsilon^{(n)}}{\Delta t} \end{pmatrix} \quad (88)$$

STEP 3: implicit treatment of convection and diffusion
The following system corresponds to equations (17), where the convection/diffusion operator is fully implicited, and the source terms partially implicited:

$$\begin{cases} \rho^{(n)} \dfrac{k^{(n+1)} - k^{(n)}}{\Delta t} = D(k^{(n+1)}) + \mathcal{S}_k(k_{st},\varepsilon_{st}) \\ \rho^{(n)} \dfrac{\varepsilon^{(n+1)} - \varepsilon^{(n)}}{\Delta t} = D(\varepsilon^{(n+1)}) + \mathcal{S}_\varepsilon(k_{st},\varepsilon_{st}) \end{cases} \quad (89)$$

which gives:

$$\begin{cases} \rho^{(n)} \dfrac{k^{(n+1)} - k^{(n)}}{\Delta t} = D(k^{(n+1)}) - D(k^{(n)}) + \rho^{(n)} \dfrac{k_{st} - k^{(n)}}{\Delta t} \\ \rho^{(n)} \dfrac{\varepsilon^{(n+1)} - \varepsilon^{(n)}}{\Delta t} = D(\varepsilon^{(n+1)}) - D(\varepsilon^{(n)}) + \rho^{(n)} \dfrac{\varepsilon_{st} - \varepsilon^{(n)}}{\Delta t} \end{cases} \quad (90)$$

These last equations are solved separately and using the same algorithms as for the equation for scalars.

13 Appendix C: numerical experiments for determination of order in space

13.1 Introduction

The order of convergence in space of the gradient calculation methods described in section 5.3 is evaluated for a Poisson equation on elementary computational domains. A more detailed description of this work may be found in [DAH 00].

13.2 Numerical test cases definition

The equation considered here is the following Poisson problem, in a two-dimensional context:

$$\begin{cases} \text{div}\,(\underline{\text{grad}}\; b) = f \\ \text{Dirichlet conditions for } b \text{ at boundaries} \end{cases} \tag{91}$$

Using notations defined in section 5.2, the discrete form of (91) is the following (for the sake of clarity, boundary faces are not considered):

$$\sum_{j \in Neibrs(i)} \mathcal{G}_{\underline{n},ij}(b) S_{ij} = |\Omega_i|\, f_i \tag{92}$$

As indicated section 5.2, the expression for $\mathcal{G}_{\underline{n},ij}(b)$ involves $G_i(b)$ (gradient at cell centres). The numerical tests presented here have been carried out with the two methods available in *Code_Saturne* for computing $G_i(b)$ (these methods are described in section 5.3 and referred to as "standard" and "least squares" methods.)
However, for triangular meshes[6], if the cell centres are the circumcentres, the expression for the gradient at cell faces adopted here degenerates to:

$$\mathcal{G}_{\underline{n},ij}(b) = \frac{b_j - b_i}{(\underline{IJ}, \underline{n})} \tag{93}$$

This is a particularly interesting property since for strictly Delaunay meshes[7], it has been demonstrated theoretically that the convergence of the solution obtained with this scheme is first order and, for equilateral triangles, second order because the approximation of the diffusion flux is then second order (see [EYM 00], [EYM 01], [HER 95]). Moreover, for $\mathcal{C}^2$ solutions, numerical experiments in [BCH 00] also exhibited second order convergence on non-Delaunay meshes.

The computational domains selected for the Poisson problem are an equilateral and a scalene triangle, shown on figure 26. Each grid consists of triangular cells obtained by applying a scaling factor to the triangle representing the computational domain. The numerical tests have been carried out on grids containing 4, 16, 64, 256, 1024 an 4096 triangles (the associated scaling factors are 2, 4, 8, 16, 32 and 64). The coarser grids are shown on figure 26 for the two computational domains. The three first grids are shown on figure 27 for the equilateral domain.
The cell centres used in *Code_Saturne* being the mass centres they are also the circumcentres for the equilateral mesh. Moreover, it is pointed out that the equilateral mesh is a particular case of stricly Delaunay mesh. Therefore, on the equilateral mesh, second order convergence *must* be reached.

Four different functions have been used for the right-hand side f of equation (91). They are listed in table 3 as well as the corresponding analytical solutions $\tilde{b}$.

[6] For general quadrangular meshes, see [COU 96], [FA1 92] and [FA2 92] for example.

[7] "Strictly Delaunay" refers to meshes consisting of triangles for which all angles are strictly lower than 90°.

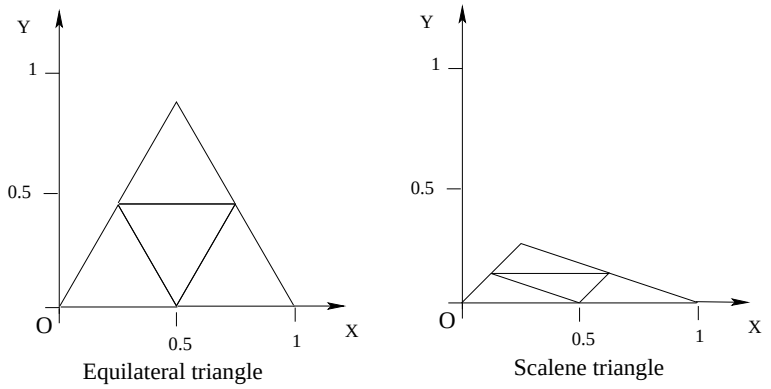
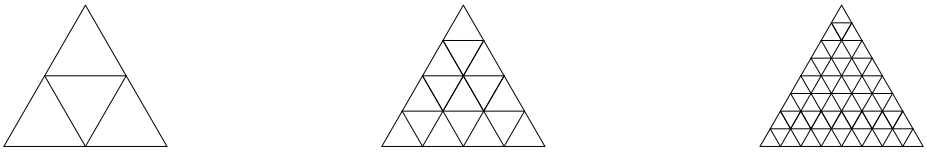


Figure 26: Computational domains and coarser meshes.



4 cells 16 cells 64 cells

Figure 27: Three first grids used for the equilateral domain.

case	f	$\tilde{b}$
1	0	$xy - y$
2	4	$x^2 + y^2$
3	0	$3x^2y - y^3$
4	$-2sin(x+y)$	$sin(x+y)$

Table 3: Expression of the four right-hand sides f and of the associated analytical solutions $\tilde{b}$ to the Poisson equation.

Dirichlet conditions[8] have been applied as boundary conditions for b. Their values have been evaluated at the centres of the boundary faces using the analytical expressions for the solution $\tilde{b}$.

To summarize, the numerical tests presented here for the Poisson problem have been carried out:

- on equilateral and scalene meshes;
- with six refinement levels;
- with four right-hand sides regular functions;
- with the standard and the least squares methods for computing gradients at cell centres.

13.3 Convergence results

Errors have been evaluated for cell-centre values of the solution, for the gradient computed at cell centres and for the gradient normal to faces. With Φ_l representing the values of these quantities computed on their associated geometrical entities E_l (cells or faces) and with $\tilde{\Phi}(L)$ standing for the corresponding analytical expression, error $e_{\Phi,E}$ is defined as:

$$e_{\Phi,E} = \sqrt{\frac{\sum\limits_{E_l} \left(\Phi_l - \tilde{\Phi}(L)\right)^2}{\sum\limits_{E_l} \tilde{\Phi}(L)^2}} \tag{94}$$

A global summary to the different computations is, as expected, that the convergence order is approximately 2 for the solution b, 1 for the gradient $\underline{\text{grad}}\ b$ and 1 for the gradient normal to faces. With the method used, these convergence orders are the best that can be expected (second order spatial approximations have been employed).
More precisely, with the "standard" gradient calculation, the order of convergence (computed on the results obtained on the four finest grids) is between 1.84 and 2.00 for the solution b and between 0.96 and 0.99 for the gradient at the cell centres[9]. The order of convergence obtained for the normal gradient is always strictly superior to 1 (between 1.88 and 2.24 on the equilateral domain) and between 1.58 and 1.70 on the scalene domain. Similar observations have been reported in [BCH 00]. First order is theoretically proven in [EYM 00], second order on equilateral meshes is due to the second order consistency error of the fluxes.

[8] The same conclusions for convergence orders have been obtained with boundary conditions consisting of exact Dirichlet conditions at $y = 0, x \in [0; 0.5]$ and exact Neumann conditions elsewhere.

[9] There are some exceptions, which might be due to the regularity of the mesh and of the solution. For example, the order of convergence is 2 for the gradient obtained on the equilateral domain for the second right-hand side function (Table 3).

Solution				
	Standard method		Least squares method	
Case	Equilateral	Scalene	Equilateral	Scalene
1	1.97	1.84	1.98	1.86
2	2.00	1.96	2.00	1.95
3	1.99	1.84	1.99	1.91
4	1.99	1.88	1.99	1.84
Gradient				
	Standard method		Least squares method	
Case	Equilateral	Scalene	Equilateral	Scalene
1	0.99	0.99	0.99	0.99
2	2.00	0.99	2.00	1.87
3	0.99	0.96	0.99	0.96
4	0.99	0.96	0.99	0.98
Normal gradient				
	Standard method		Least squares method	
Case	Equilateral	Scalene	Equilateral	Scalene
1	1.89	1.70	1.89	1.68
2	2.24	1.60	2.25	1.88
3	1.88	1.65	1.88	1.75
4	1.91	1.58	1.91	1.71

Table 4: Orders of convergence obtained for the solution, the gradient and the normal gradient, with the standard and the least squares methods, four different right-hand sides (cases 1 to 4 in table 3) and two different domains (equilateral and scalene).

Finally, it is pointed out that the least squares method exhibits the same orders of convergence as the "standard" gradient calculation. But naturally, this does not imply that the *accuracy* is the same on a given mesh (especially on a quite "coarse" one, as industrial meshes often tend to be). The order of convergence is between 1.84 and 2.00 for the solution b and between 0.96 and 0.99 for the gradient[10]. The order of convergence for the normal gradient is between 1.88 and 2.25 on the equilateral domain and between 1.68 and 1.88 on the scalene domain.

14 Appendix D: boundary conditions for turbulence

14.1 Introduction

The high-Reynolds approach presented here is based on the use of wall functions to bridge the viscous sublayer and determine shear stress [LAU 74]. The default

[10]There are some exceptions, as for the "standard" method: the order of convergence of the gradient obtained with the second right-hand side function is 2.00 on the equilateral domain and 1.87 on the scalene domain

approach is based on a so-called "two-scale modelling", with a logarithmic law[11] for the tangential components of the velocity. Similar wall functions are used for the scalars (the approach proposed by Arpaci and Larsen [ARP 84] has been adopted for temperature).

Hence, to account for the physical phenomena related to the wall layer, it is necessary to modify the wall boundary condition to compute:

- $\mathrm{div}(\underline{\underline{\tau}}_t) = \mathrm{div}(\underline{\underline{\tau}} - \rho\underline{\underline{R}})$ for the components of the velocity tangent to the wall (momentum equation (14.b))
- $\mathrm{div}(\Phi) = \mathrm{div}\left((K_a + \dfrac{\mu_t}{\sigma_{a,t}})\underline{\mathrm{grad}}\ a\right)$ for scalars (equation (14.c))

The approach is presented for $\mathrm{div}(\underline{\underline{\tau}}_t)$ only. Its extension to $\mathrm{div}(\Phi)$ is natural since the formulation for Φ is similar to that for $\underline{\underline{\tau}}_t$ with the $k-\varepsilon$ model.

Moreover, boundary treatment also need to be specified at inlets and at walls for the terms appearing in equations for variables k, ε and R_{ij}. At outlet and symmetry planes, the standard treatment described for scalars (section 6) applies and will not be recalled here.

Eventually, in this appendix as in the main section, $\underline{n}$ denotes the unit vector normal to the boundary and oriented outwards. When it is required (at walls) the tangential velocity is defined as:

$$\underline{u}_{tg} = \underline{u} - (\underline{u} \cdot \underline{n})\underline{n}$$

and the unit vector $\underline{t}$ tangent to the boundary is:

$$\underline{t} = \frac{\underline{u}_{tg}}{||\underline{u}_{tg}||}$$

14.2 Wall boundary condition for the momentum equation

For the computation of $\mathrm{div}(\underline{\underline{\tau}}_t)$, a boundary value is required for the total wall shear stress: $\tau_w = ||(\underline{\underline{\tau}}_t\,\underline{n}) \cdot \underline{t}||$. With the previous definitions, the following modelling is adopted:

$$\tau_w \quad = \quad \rho u^* u_k \tag{95}$$

where u_k and u^* are friction velocities determined from variables obtained close to the wall, namely the turbulent kinetic energy k and the tangential velocity $||\underline{u}_{tg}||$. More precisely, let us assume that the geometrical point I' (figure 3) is located at a distance y from the wall where the logarithmic law is valid, that is:

$$\left\{ \begin{array}{l} \dfrac{||\underline{u}_{tg}||}{u^*} = \dfrac{1}{\kappa} ln(y^+) + C \\ \quad \text{with } y^+ = \rho u_k y/\mu \qquad \kappa = 0.42 \qquad \text{and } C = 5.2 \end{array} \right. \tag{96}$$

[11] The logarithmic law of the wall degenerates to a linear law in the viscous sublayer.

Since ρ and μ are known, it is possible to evaluate $\tau_w = \rho u^* u_k$ using:

$$\left\{ \begin{array}{rcl} u_k & = & C_\mu^{\frac{1}{4}} k_{i'}^{\frac{1}{2}} \\ u^* & = & ||\underline{u}_{tg,i'}||/(\frac{1}{\kappa} ln(y^+) + C) \end{array} \right. \tag{97}$$

14.3 Inlet boundary conditions for turbulence

At inlet, the approach described for scalars in section 6.2 applies to variables k, ε and R_{ij}. However, the inlet values for the Dirichlet conditions need to be specified. Indeed, various inlet quantities might be available from experimental data or analytical considerations (for example velocity or scalars) but for turbulent variables, further assumptions are often required.

With the $k-\varepsilon$ model, the following conditions are set by default:

$$k = \frac{u^{*2}}{\sqrt{C_\mu}} \quad \text{and} \quad \varepsilon = \frac{u^{*3/2}}{\kappa D_h/10} \tag{98}$$

where D_h is the user-prescribed hydraulic diameter of the inlet, and u^* is a friction velocity, determined from experimental correlations. For example, for a developed pipe flow with zero roughness, we have:

$$u^* = \sqrt{U_{ref} \lambda^*/8} \quad \begin{array}{lrcll} \text{with} & \lambda^* & = & 0.3164 Re^{-\frac{1}{4}} & \text{for } Re \leqslant 30000 \\ \text{and} & \lambda^* & = & 0.1840 Re^{-\frac{1}{5}} & \text{for } Re > 30000 \end{array} \tag{99}$$

where U_{ref} is a user-defined bulk velocity and $Re = \dfrac{\rho U_{ref} D_h}{\mu}$ is the bulk Reynolds number.

For second moment closure, the inlet conditions might be derived from (98):

$$R_{ij} = \frac{2}{3} \frac{u^{*2}}{\sqrt{C_\mu}} \delta_{ij} \quad \text{and} \quad \varepsilon = \frac{u^{*3/2}}{\kappa D_h/10} \tag{100}$$

These conditions suffer some limitations and users might be driven to prescribe more appropriate values depending on the cases studied (indeed, these conditions do not account for the shear stress across vertical inlets and the dissipation rate remains a crude approximation based on a mixing length applicable only to energy-equilibrium flows).

14.4 Wall boundary conditions for turbulence

In this section, ϕ denotes any variable pertaining to the turbulence models (k, ε or R_{ij}). Boundary conditions for all discrete terms are build under the following assumptions:

- zero mass flow rate normal to the boundary;

- for the $k-\varepsilon$ model, the following profiles are assumed to be valid close to the wall:

$$\left\{\begin{array}{lcll} k & = & {u_k}^2/C_\mu^{\frac{1}{2}} & \text{that is } \dfrac{\partial k}{\partial y} = 0 \\ \varepsilon & = & {u_k}^3/\kappa y & \text{that is } \dfrac{\partial \varepsilon}{\partial y} = -{u_k}^3/\kappa y^2 \end{array}\right. \tag{101}$$

- for the second moment closure, the local orthonormal basis $(\underline{n}, \underline{t}, \underline{t_2})$ is defined ($\underline{n}$ and $\underline{t}$ are defined in section 14.1). Denoting $\hat{R}_{ij}$ the components of $\underline{\underline{R}}$ in this local frame, the formulation of the boundary conditions relies on the assumption that the following equations are valid close to the wall:

$$\left\{\begin{array}{lcll} \left(\underline{\text{grad}}\ (\hat{R}_{ii}) \cdot \underline{n}\right) & = & 0 & \text{(no summation implied)} \\ \hat{R}_{12} & = & u^* u_k & \\ \hat{R}_{13} = \hat{R}_{23} & = & 0 & \\ \varepsilon & = & {u_k}^3/\kappa y & \text{that is } \dfrac{\partial \varepsilon}{\partial y} = -{u_k}^3/\kappa y^2 \end{array}\right. \tag{102}$$

For convection terms, the boundary value for $Q_\phi = (\underline{Q}^{(n)} \cdot \underline{n})\phi^*$ is merely set to zero, as it was for scalars in section 6.4.
For diffusion terms and source terms requiring the computation of the gradient of k, ε or R_{ij}, the boundary treatment is the same as that detailed for the scalars (section 6.4), using the Neumann and Dirichlet conditions defined by systems (101) and (102).

15 Appendix E: non conservative methods applied to compressible flows with second moment closure

15.1 Introduction

To illustrate the use of *Code_Saturne* as an open research and development platform, this section proposes a brief overview of the work presented in [PAH 00].

The standard $k-\varepsilon$ model suffers limitations for a number of flows, especially those encountered in turbomachinery applications, which also tend to include compressibility effects. Second moment closures provide some improvement but robustness of the algorithm might be limited by realizability conditions and by the prominant hyperbolic nature of the system. Some considerations related to this second aspect are presented here.

15.2 Sub-set of equations considered

We consider the second moment closure described in detail in [BRU 00]. Only the following terms are retained (in particular, source terms expected to have a

stabilizing effect are not considered here):

$$\left\{\begin{array}{lcl} \dfrac{\partial \rho}{\partial t} + \mathrm{div}(\rho\,\underline{u}) & = & 0 \\ \dfrac{\partial}{\partial t}(\rho\underline{u}) + \mathrm{div}\left(\rho\,\underline{u}\otimes\underline{u} + P\underline{\underline{Id}} + \underline{\underline{\mathcal{R}}}\right) & = & 0 \\ \dfrac{\partial E}{\partial t} + \mathrm{div}\left(E\underline{u} + [P\underline{\underline{Id}} + \underline{\underline{\mathcal{R}}}]\underline{u}\right) & = & 0 \\ \dfrac{\partial \mathcal{R}_{ij}}{\partial t} + \mathrm{div}\left(\mathcal{R}_{ij}\underline{u}\right) + NCT_{ij} & = & 0 \end{array}\right. \tag{103}$$

with

$$\left\{\begin{array}{l} NCT_{ij} = \mathcal{R}_{ik}\dfrac{\partial u_j}{\partial x_k} + \mathcal{R}_{jk}\dfrac{\partial u_i}{\partial x_k} \quad \text{non conservative "production-destruction" term} \\ \underline{\underline{\mathcal{R}}} = \rho\underline{\underline{R}} \ \text{ and } \ E = \dfrac{P}{\gamma-1} + \dfrac{1}{2}\rho u_k u_k + \dfrac{1}{2}\mathcal{R}_{kk} \end{array}\right. \tag{104}$$

15.3 Numerical scheme

With W denoting the conservative m-variables vector ($W = (\rho, \rho\underline{u}, E, \underline{\underline{\mathcal{R}}})^t$), $\underline{C}$ the $m \times m$ matrix related to non conservative terms, system (103) can be written as:

$$\frac{\partial W}{\partial t} + \mathrm{div}\left(\underline{F}(W)\right) + \underline{C}(W)\cdot \underline{\mathrm{grad}}\ (W) = 0 \tag{105}$$

With $\delta W = W^{(n+1)} - W^{(n)}$, discretization in time leads to the following formulation:

$$\frac{\delta W}{\Delta t} + \mathrm{div}\left(\underline{F}(W^{(n)})\right) + \underline{C}(W^{(n)})\cdot \underline{\mathrm{grad}}\ (W^{(n)}) = 0 \tag{106}$$

The superscript is dropped hereafter for variables evaluated at time step n.

Let W_i stand for the value of W at cell i. For the sake of clarity, we assume here that no face pertaining to cell i is on the boundary of the computational domain. With the finite volume formulation, we have:

$$\frac{|\Omega_i|}{\Delta t}\delta W_i + \sum_{j\in Neibrs(i)} \int_{S_{ij}} \underline{F}(W)\cdot\underline{n}_{ij}\,dS + \int_{\Omega_i} \underline{C}(W)\cdot\underline{\mathrm{grad}}\ (W)\,d\Omega = 0 \tag{107}$$

Non conservative terms are approximated as follows [BRU 00]:

$$\int_{\Omega_i} \underline{C}(W)\cdot\underline{\mathrm{grad}}\ (W)\,d\Omega \approx \underline{C}(W_i)\cdot\int_{\Omega_i}\underline{\mathrm{grad}}\ (W)\,d\Omega \tag{108}$$

and, using the Gauss theorem:

$$\int_{\Omega_i} \underline{C}(W)\cdot\underline{\mathrm{grad}}\ (W)\,d\Omega \approx \underline{C}(W_i)\cdot\sum_{j\in Neibrs(i)}\int_{S_{ij}} W\underline{n}_{ij}\,dS \tag{109}$$

Interface values for W are computed as $\frac{1}{2}(W_i + W_j)$ and we finally have:

$$\int_{\Omega_i} \underline{C}(W) \cdot \underline{\text{grad}}\,(W)\, d\Omega \approx \sum_{j \in Neibrs(i)} \frac{1}{2}(W_i + W_j)(\underline{C}(W_i) \cdot \underline{n}_{ij}) S_{ij} \tag{110}$$

Conservative terms are computed using the Rusanov scheme. The implementation of this scheme in a three-dimensional environment is particularly simple [BER 02]: it requires nothing more that the computation of a spectral radius $\rho_{rus}(i)$ in each cell i to evaluate the conservative flux as follows:

$$\int_{S_{ij}} \underline{F}(W) \cdot \underline{n}_{ij} dS \approx \left(\frac{1}{2} \left(\underline{F}(W_i) + \underline{F}(W_j) \right) \cdot \underline{n}_{ij} - \frac{1}{2} \max_{k \in \{i,j\}} \left(\rho_{rus}(k) \right) (W_j - W_i) \right) S_{ij} \tag{111}$$

The spectral radius accounts for conservative and non conservative terms. Indeed, it is determined as follows.

In the global computational frame, $(\underline{e}_x, \underline{e}_y, \underline{e}_z)$, the three components of $\underline{F}(W)$ and $\underline{C}(W)$ are respectively denoted $F_x(W)$, $F_y(W)$, $F_z(W)$ and $C_x(W)$, $C_y(W)$, $C_z(W)$. We also introduce the $m \times m$ matrices $\underline{J}$ and $\underline{B}$ and their components. With k standing for x, y or z, we have:

$$J_k(W) = \frac{\partial}{\partial W} F_k(W) \text{ and } \quad B_k(W) = J_k(W) + C_k(W) \tag{112}$$

Hence, system (106) reads:

$$\frac{\delta W}{\Delta t} + B_x(W) \frac{\partial W}{\partial x} + B_y(W) \frac{\partial W}{\partial y} + B_z(W) \frac{\partial W}{\partial z} = 0 \tag{113}$$

Similarly, for each interface (i, j) of the mesh, let $(\underline{n}_{ij}, \underline{\tau}_{1,ij}, \underline{\tau}_{2,ij})$ be a local basis and let us denote $(\xi_{ij}, \zeta_{ij}, \eta_{ij})$ the related coordinates. Neglecting variations of W on the interface equation (113) reads:

$$\frac{\delta W}{\Delta t} + B_{\xi_{ij}}(W) \frac{\partial W}{\partial \xi_{ij}} = 0 \tag{114}$$

With $\rho_{rus}(i)$ denoting the spectral radius of $B_{\xi_{ij}}(W_i)$, the Rusanov scheme can be applied to approximate the flux at interface (i, j) using expression (111). It is pointed out that matrix $B_{\xi_{ij}}(W)$ accounts for conservative and non conservative terms and is the Jacobian matrix if non conservative terms vanish.

15.4 Interesting properties

Eigenvalues of $B_{\xi_{ij}}(W)$ take the following values:

$$u_{n,ij}, \quad u_{n,ij} \pm c_{1,ij} \text{ and } u_{n,ij} \pm c_{2,ij} \tag{115}$$

$$\begin{aligned} &\text{with: } u_{n,ij} = \underline{u} \cdot \underline{n}_{ij}, \quad c_{1,ij} = \sqrt{3R_{nn,ij} + \frac{\gamma P}{\rho}}, \quad c_{2,ij} = \sqrt{R_{nn,ij}} \\ &\text{and: } R_{nn,ij} = \underline{n}_{ij}^t \underline{\underline{\mathcal{R}}}\, \underline{n}_{ij} \end{aligned} \tag{116}$$

Since $c_{1,ij}$ and $c_{2,ij}$ account for pressure and turbulence effects, the scheme is expected to be more stable than methods decoupling equations for second moments R_{ij}, especially for flows at high turbulence levels in impingement regions or afterbodies, as those often encountered in turbomachinery.

In addition to its simple formulation, the Rusanov scheme benefits from another important asset: it garantees the positivity of the density. Nevertheless, it turns out to be quite diffusive, and at least more diffusive than the approximate Godunov scheme denoted VFRoe-ncv [BUF 00] [GAL 02] [GAL 03]: figure 28 shows a comparison of pressure obtained with VFRoe-ncv and Rusanov on coarse and fine meshes for a one-dimensional shock tube.

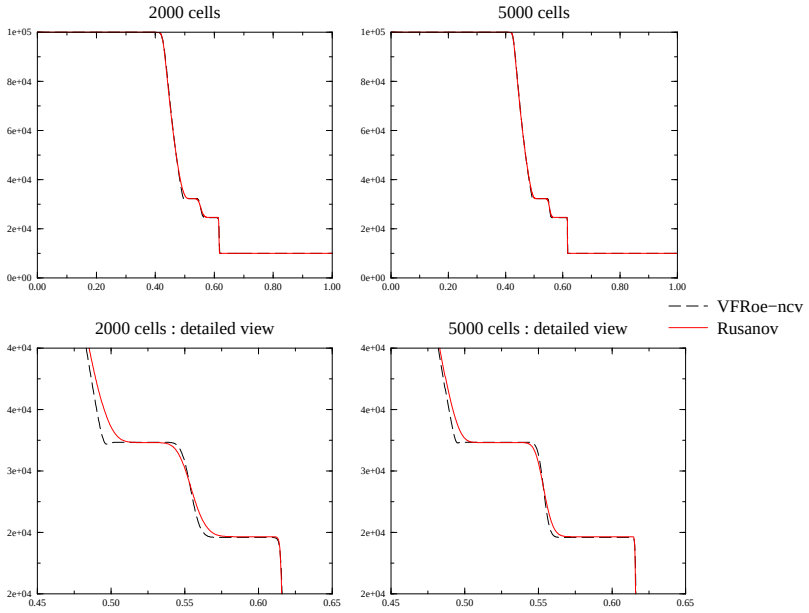


Figure 28: Pressure profile obtained with VFRoe-ncv and Rusanov on coarse and fine meshes for a one-dimensional shock tube.

This research work would require an extension to take into account a complete physical modelling. Nevertheless, to illustrate the capability to carry out two- and three-dimensional computations with these developments included in *Code_Saturne*, figures 29, 30, 31 and 32 represent iso-Mach contours obtained above ascending steps and around a turbine blade with grids constituted of hexaedra and prisms (*Code_Saturne* data structure supports any mesh).

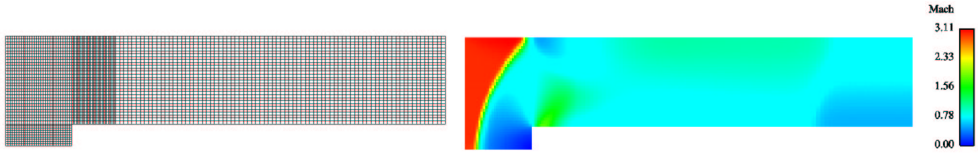


Figure 29: Iso-Mach contours above an ascending step obtained with the Rusanov scheme on a mesh consisting of hexaedra.

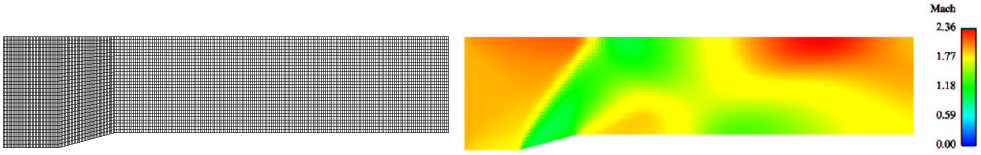


Figure 30: Iso-Mach contours above an inclined ascending step obtained with the Rusanov scheme on a mesh consisting of hexaedra.

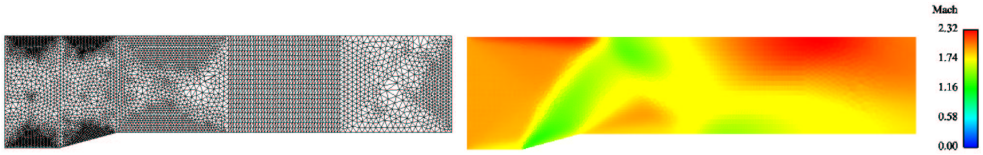


Figure 31: Iso-Mach contours above an inclined ascending step obtained with the Rusanov scheme on a mesh consisting of prisms.

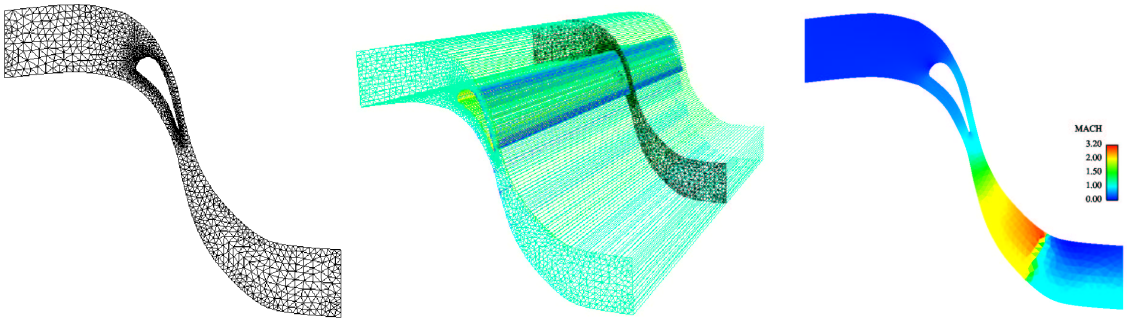


Figure 32: Iso-Mach contours around a turbine blade obtained with the Rusanov scheme on a mesh consisting of prisms.

16 Bibliography

[ARP 84] Arpaci V.S., Larsen P.S., (1984), "Convection Heat Transfer", Englewood Cliffs, N.J. Prentice-Hall

[BCH 00] Boivin S., Cayré F., Hérard J.-M.(2000), "A Finite Volume Method to Solve the Navier-Stokes Equations for Incompressible Flows on Unstructured Meshes", Int. J. of Thermal Sciences, Vol. 39-8, pp. 806-825

[BCH 01] Boivin S., Cayré F., Hérard J.-M., (2001), "Un schéma volumes finis pour la simulation d'écoulements diphasiques gaz-solide à deux phases incompressibles sur maillage triangulaire", Revue Européenne des Éléments Finis, Vol. 10-5, pp. 539-574

[BER 02] Berthon C., Coquel F., Hérard J.M., Uhlmann M., (2002), "An Approximate Solution of the Riemann Problem for a Realizable Second Moment Turbulent Closure", Shock Waves, Vol. 11-4, pp. 245-269

[BRE 91] Brezzi F., Fortin M., (1991), "Mixed and Hybrid Finite Element Methods", Springer-Verlag

[BRU 00] Brun G., Hérard J.-M., Jeandel D., Uhlmann M., (2000), "An Approximate Roe-type Riemann Solver for a Class of Realizable Second Order Closures", IJCFD, Vol. 13-3, pp. 223-249

[BUF 00] Buffard T., Gallouët T., Hérard J.-M., (2000), "A sequel to a rough Godunov scheme. Application to real gas flows", Computers and Fluids, Vol. 29-7, pp. 813-847

[CHA 92] Chabard J.P., Métivet B., Pot G., Thomas B., (1992), "An efficient finite element method for the computation of 3D turbulent incompressible flows", Finite Elements in Fluids, Vol. 8, Hemisphere

[CHA 03] Chabard J.-P., (2003), "The Joint CEA and EDF Research Programme for a New Generation of Nuclear Reactor Computational Tools.", in Minutes of the Post FISA-2003 Workshop: "Advanced Multiphysic Computations for Nuclear Reactor Safety".

[CHO 68] Chorin A., (1968), "Numerical Solution of the Navier-Stokes Equations", Math. Comput., Vol. 22, pp. 745-762

[COU 96] Coudière Y., Vila J.-P., Villedieu P., (1996), "Convergence Rate of a Finite Volume Scheme for a Two Dimensional Convection-Diffusion Problem", Mathematical Modelling and Numerical Analysis, Vol.33-3, pp. 493-516

[DAL 70] Daly B.J., Harlow F.H., (1970), "Transport Equation in Turbulence", Phys. Fluids, Vol. 13, p. 2634

[DAL 01] Dal Secco S., (2001), "Modélisation par une méthode Lagrangienne de l'encrassement dans le foyer d'une chaudière Q600 dû à l'impact inertiel des particules de charbon aux parois de la chaudière", EDF-R&D internal report, HI-81/01/033/A, in French

[DAH 00] Davroux A., Archambeau F., Hérard J.-M., (2000), "Tests numériques sur quelques méthodes de résolution d'une équation de diffusion en volumes finis", EDF-R&D internal report, HI-83/00/027/A, in French

[DEL 03] Delbecq J.-M., Banner D., (2003), "EDF's Experience with Supercomputing and Challenges Ahead. Towards the Multiphysic and Multiscale Approach.", Conference on Supercomputing in Nuclear Applications, Paris, France, September 22-24, 2003

[DPR 01] Durbin P.A., Petterson Reif B.A., (2001), "Statistical Theory and Modeling for Turbulent Flows", John Wiley & Sons
[DUR 93] Durbin P.A., (1993), "A Reynolds stress model for near-wall turbulence", Journal of Fluid Mechanics, pp. 465-498.
[EYM 00] Eymard R., Gallouët T., Herbin R., (2000), "The Finite Volume method", in "Handbook of Numerical Analysis", P.G. Ciarlet and J.L. Lions Editors, North Holland, pp. 715-1022.
[EYM 01] Eymard R., Gallouët T., Herbin R., (2001), "Finite Volume approximation of elliptic problems and convergence of an approximate gradient", Applied Numerical Mathematics, Vol. 37, pp. 31-53.
[FA1 92] Faille I., (1992), "A Control Volume Method to Solve an Elliptic Equation on a Two-Dimensional Irregular Mesh", Computational Methods for Applied Mechanical Engineering, Vol. 100, pp. 275-290
[FA2 92] Faille I., (1992), "Modélisation bidimensionnelle de la genèse et de la migration des hydrocarbures dans un bassin sédimentaire", PhD Thesis, Université Joseph Fourier - Grenoble 1, in French.
[FAV 76] Favre A., Kovasznay L.S.G., Dumas R., Gaviglio J., Coantic M., (1976), "La turbulence en mécanique des fluides", Gauthier-Villars, in French
[FER 99] Ferziger J.H., Perić M., (1999), "Computational Methods for Fluid Dynamics", Springer, 389 p.
[FO1 03] Fournier Y., (2003), "*Code_Saturne* Version 1.1 : guide pratique et théorique du module Enveloppe", EDF-R&D internal report, HI-83/03/007/A, in French
[FO2 03] Fournier Y., (2003), "*Code_Saturne* Version 1.1 : manuel informatique du module Enveloppe", EDF-R&D internal report, HI-83/03/008/A, in French
[FPS 04] Fede P., Patino G., Simonin O., (2004), "Multifluid Modeling of the Effect of Inter-Particle Collisions in Polydispersed Gas-Solid Turbulent Flows", submitted to Third International Symposium on Two-Phase Flow Modelling and Experimentation, Pisa, Italy, September 22-24, 2004
[GAL 02] Gallouët T., Hérard J.-M., Seguin N., (2002), "Some recent finite volume schemes to compute Euler equations using real gas EOS", Int. J. for Num. Meth. in Fluids, Vol. 39-12, pp. 1073-1138.
[GAL 03] Gallouët T., Hérard J.-M., Seguin N., (2003), "On the use of some symmetrizing variables to deal with vacuum", CALCOLO Vol. 40-3, to appear.
[GES 02] Gest B., Archambeau F., Béchaud C., Benhamadouche S., Sakiz M. (2002), "Documentation théorique et informatique du noyau de *Code_Saturne* 1.1", EDF-R&D internal report, HI-83/02/008/A, in French
[GIB 76] Gibson M.M., Launder B.E., (1976), "On the calculation of horizontal turbulent free shear flow under gravitational influence", ASME J. Heat Transfer, Vol. 98c, pp. 81-87
[HER 95] Herbin R., (1995), "An Error Estimate for a Finite-Volume Scheme for a Diffusion-Convection Problem on a Triangular Mesh", Numerical Methods Part. Diffusion Equation, Vol. 11, pp. 165-173
[KAR 98] Karypis G., Kumar V., (1998), "MeTiS – A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes and Computing Fill-Reducing Orderings of Sparse Matrices – Version 4.0", University of Minnesota, September 1998

[LAU 74] Launder B.E., Spalding D.B., (1974), "The Numerical Computation of Turbulent Flow", Comp. Meth. in Appl. Mech. And Engng., Vol. 3, pp. 269-289
[LES 97] Lesieur M., (1997), "Turbulence in Fluids", Kluwer Academic Publishers, Dordrecht
[LID 96] Lien F.S., Durbin P.A., (1996), "Non-linear $k - \varepsilon - v^2$ modeling with application to high-lift", Center for Turbulence Reasearch,Proceedings of the summer school program, pp. 5-25
[LRR 75] Launder B.E., Reece G.J., Rodi W., (1975), "Progress in the Development of a Reynolds-Stress Turbulent Closure", J. Fluid Mech., Vol. 68, part 3, pp. 537-566
[MAR 02] Martin A., Bellet S., (2002), "CFD-Tool for Thermal-Hydraulics Pressurized Thermal Shock Analysis. Qualification of *Code_Saturne*.", Proceedings of Pressure Vessel and Piping Conference, 4-8 August 2002, Vancouver
[MAT 92] Mattéi J.D., Simonin O., (1992), "Logiciel ESTET. Manuel théorique de la version 3.1. Tome I. Modélisations physiques", EDF-R&D internal report, HE-44/92.38/A, in French
[MEC 03] Méchitoua N., Boucker M., Laviéville J., Hérard J.-M., Pigny S., Serre G., (2003), "An Unstructured Finite Volume Solver for Two-Phase Water/Vapour Flows Modelling Based on an Elliptic Oriented Fractional Step Method", NURETH 10, 10^{th} International Meeting on Nuclear Reactor Thermal-Hydraulics, Seoul, South Korea, October 5-9, 2003
[MEN 94] Menter F.R., (1994), "Two-equation eddy-viscosity turbulence models for engineering applications", AIAA Journal, pp. 1598-1605
[MIN 01] Minier J.P., (2001), "Probabilistic approach to turbulent two-phase flows modelling and simulation: theoretical and numerical issues", Monte-Carlo Methods and Applications, Vol. 7(3-4), pp. 295-310, 2001
[MIP 01] Minier J.P., Peirano E. (2001), "The PDF approach to turbulent dispersed two-phase flows", Physics Reports, Vol. 352, n° 1-3, October 2001
[MOP 94] Mohammadi B., Pironneau O., (1994), "Analysis of the K-Epsilon Turbulence Model", John Wiley & Sons
[MUS 96] Musaferija S., Gosman D., (1996), "Finite-Volume CFD Procedure and Adaptive Error Control Strategy for Grids of Arbitrary Topology", J. Comp. Phys., Vol. 138, pp. 766-787, 1997, CP97853
[NAO 70] Naot D., Shavit A., Wolfshtein M., (1970), "Interactions between components of the turbulent velocity tensor", Israel J. Tech., Vol. 8, p. 259
[PAH 00] Périgaud G., Archambeau F., Hérard J.-M., (2000), "Méthodes non conservatives multidimensionnelles pour modèle de turbulence au second ordre.", EDF-R&D internal report, HI-83/00/028/A, in French
[PEN 97] Péniguel C., Rupp I., (1997), "Coupling Heat Conduction and Radiation in Complex 2D and 3D Geometries", Proceeding Numerical Methods in Thermal Problems, Swansea.
[PEN 98] Péniguel C. (1998), "Heat transfer simulation for industrial applications: needs, limitations, expectations", International Journal of Heat and Fluid Flow, Vol. 19, pp. 102-104.
[PEN 03] Péniguel C., Sakiz M., Benhamadouche S., Stéphan J.M., Vindeirinho C., (2003), "Presentation of a Numerical 3D Approach to Tackle Thermal Striping

in a PWR Nuclear T-Junction.", Proceedings of ASME PVP, July 20-24, 2003, Cleveland, USA

[POP 00] Pope S.B., (2000), "Turbulent Flows", Cambridge University Press

[RHI 84] Rhie C.M., Chow W.L., (1983) "Numerical Study of a Turbulent Flow past an Airfoil with Trailing Edge Separation", AIAA Journal, Vol. 21, No. 11, pp. 1525-1532, November 1983

[ROD 84] Rodi W., (1984), "Turbulence Models and their Application in Hydraulics", state-of-the-art paper presented by the IAHR-Section on Fundamentals of Division II: Experimental and Mathematical Fluid Dynamics

[ROT 51] Rotta J., (1951), "Statistische Theorie nichthomogener Turbulenz I", Z. Phys., Vol. 129, p. 547

[ROT 82] Rothe P.H., Ackerson M.F., (1982), "Fluid and Thermal Mixing in a Model Cold Leg and Downcomer With Loop Flow", NP-2312, Research Project 347-1, Interim Report, April 1982, prepared by CREARE Inc. for EPRI.

[RUP 99] Rupp I., Péniguel C., (1999), "Coupling Heat Conduction, Radiation and Convection in Complex Geometries", Int. Journal of Numerical Methods for Heat and Fluid Flow, Vol. 9-3, pp. 240-256

[SSG 91] Speziale C.G., Sarkar S., Gatski T.B., (1991), "Modelling the pressure-strain correlation of turbulence: an invariant dynamical system approach", Journal of Fluid Mechanics, pp. 245-272

[SSV 04] Saulnier C., Simonin O., Védrine D., (2004), "Eulerian Multiphase Modeling of Feed Injection and Vaporisation in FCC Riser Reactor", submitted to Third International Symposium on Two-Phase Flow Modelling and Experimentation, Pisa, Italy, September 22-24, 2004

[TEM 79] Temam R., (1979), "Navier-Stokes Equations, Theory and Numerical Analysis", Ed. North-Holland, Amsterdam.

[VER 95] Versteeg H.K., Malalasekera W., (1995), "An introduction to Computational Fluid Dynamics: The Finite Volume Method", Longman Scientific and Technical.

[VIO 88] Viollet P.L., (1988), "On the numerical modelling of stratified flows", Physical processes in estuaries, (J. Dronkers and W. van Leussen Eds), pp. 257-277, Springer-Verlag