ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

DOTTORATO DI RICERCA IN

Meccanica e Scienze Avanzate Dell'Ingegneria

CICLO XXXI

SETTORE CONCORSUALE DI AFFERENZA: 09/C2

SETTORE SCIENTIFICO-DISCIPLINARE: ING/IND-19 IMPIANTI NUCLEARI

Development of a computational platform for the simulation of low Prandtl number turbulent flows

Presentata da ROBERTO DA VIÀ

Coordinatore Dottorato	**Supervisore**
Chiar.mo Prof. Ing. MARCO CARRICATO	Chiar.mo Prof. Ing. SANDRO MANSERVISI

Esame finale anno 2019

Contents

Abstract

Mathematical modeling of physical phenomena is at the basis of many scientific field researches. Complex systems show multiscale and multiphysics aspects that cannot be always taken into account in detail. In the past many numerical codes have been developed and specialized to solve different aspects of turbulence and, in general, fluid motion for a very wide range of engineering applications. Nowadays, numerical code coupling and computational platforms are gaining a lot of interest for the simulation of very complex phenomena. This PhD study focuses on modeling physical systems with coupled simulations, in particular turbulent heat transfer for liquid metals. This type of fluids, known as low Prandtl number fluids, requires more sophisticated turbulent heat transfer models since those used to simulate fluids such as air or water lead to a sensible heat transfer overestimation. Seeking an increased numerical stability, a four logarithmic parameter turbulence model is proposed, starting from a model that has already been validated with simulations of Lead-Bismuth-Eutectic (LBE) fully developed turbulent flows. The turbulence model has been implemented in the finite element code FEMuS to perform an extensive validation by comparing obtained results with Direct Numerical Simulations and experimental data. Many simulations are performed, for fully developed turbulent flows in plane channels, cylindrical pipes and 19 pin nuclear reactor bundles and for turbulent forced and mixed convection over a backward facing step. When conservation equations of mass, momentum and energy need be coupled with dynamic two-equation or thermal turbulence four-equation models the use of numerical coupling becomes important. In order to dispose of a greater choice of dynamical turbulence models, a computational platform containing OpenFOAM and FEMuS codes has been developed.

Introduction

Mathematical modeling of physical phenomena is at the basis of many scientific field researches. Complex systems show multi-scale and multi-physics aspects that cannot be always taken into account in detail. Turbulence is a typical example where phenomena at different physical-space scales contribute to determine the fluid motion. Many numerical codes have been developed and specialized to solve different aspects of turbulence and fluid motion for a very wide range of engineering applications such as combustion, solid mechanics, turbulent flows, heat transfer and many others. Numerical code coupling represents a further step forward in modeling and simulations, in order to take into account several aspects of a physical system. With a relatively small effort, numerical codes can interact, within the frame of a *computational platform*, so that mutual influences between different physical phenomena can be analyzed. The development of huge and complex numerical codes can then be avoided and attention should be focused on efficient methods for data transfer and manipulation.

This PhD study focuses on two main objectives: the simulation of turbulent heat transfer for fluids with a very low Prandtl number value (Pr), and the development of a computational platform where turbulence models can be easily implemented and tested. Liquid metals, with $Pr \simeq 10^{-2}$, are becoming more and more appealing because of their physical properties and are currently considered as operative fluids in concentrating solar power systems and in IVth generation fast nuclear reactors. In latter application, the possibility to establish natural convection in the case of missing power supply represents a key feature to obtain an increased intrinsic safety level. Accurate means are needed to predict heat transfer and fluid flow under these circumstances, in order to properly design these systems. The turbulent heat transfer models provided by commercial codes are all based on Reynolds analogy, i.e. a similarity assumption between velocity and temperature fields. Due to the very Pr value, this assumption does not hold for liquid metals and more complex thermal turbulence model are needed. Many different models have been developed balancing model accuracy, complexity and numerical stability but their range of validity is still an open problem. In the present thesis, a new formulation of an existing four parameter turbulence model is proposed and implemented in FEMuS code, with the introduction of logarithmic variables, in order to overcome the stability issues of the original model. The model has been chosen since reliable results have already been obtained for simulations of Lead-Bismuth-Eutectic fully developed turbulent flows, with $Pr = 0.025$. For this model a more extensive validation is sought, in terms of different Pr values and of simulated geometries. Since the accuracy of turbulent heat transfer simulations depends on the choice of both dynamical and thermal turbulence models, a computational platform containing FEMuS and OpenFOAM codes has been developed. In this way the more complete set of dynamical turbulence models implemented in OpenFOAM can be tested in coupled simulations with the thermal turbulence model implemented in FEMuS.

The thesis is organized as follows. In Chapter 1 the Finite Element Method is briefly introduced as this is the numerical method used to solve partial differential equations in FEMuS. This numerical code is developed at Montecuccolino laboratory and is used for the implementation of the four logarithmic parameter turbulence model.

The problem of numerical code coupling is addressed in Chapter 2. The outline of the computational platform is described, with particular attention to SALOME platform and MED libraries since they are used to handle the data exchange between different codes. Many routines have been developed in the form of C++ classes and using MED data format, so that they can be generally used since they are not written for a specific code. They allow to obtain a more complete set of functions for data manipulation, together with the already existing routines provided by MED library. Finally the case of natural convection inside a square cavity is studied with a coupling between FEMuS code and OpenFOAM. Different procedures for data manipulation are studied.

The developed routines for numerical code coupling have been successfully used to define a new immersed boundary method that is described in Chapter 3, proving the great versatility of the computational platform. In the new algorithm a computational grid is used for the equation solution and another one is used to model the solid body. A key feature is the capability to deal with both static and moving arbitrarily shaped objects. Similarly to multi-phase problems, solid and fluid regions are distinguished using an indicator function field that is obtained as the result of a uniform field projection from the solid body mesh to the computational grid. A method for solid-fluid interface reconstruction on irregular bi-dimensional grids is presented and results are reported for a case of fluid flow around a static object and of impulsively started cylinder in a fluid at rest.

In Chapter 4 the Reynolds Averaged Navier-Stokes system of equations is introduced and the derivation of the four logarithmic parameter turbulence model is discussed. Reference behaviors for turbulent variables and velocity are introduced, together with a new law of the wall for temperature field.

Results of fully developed turbulent flows in the geometries of plane channel, cylindrical pipe and 19 pin hexagonal nuclear rector bundle are shown in Chapter 5. Whereas possible, a comparison with results from Direct Numerical Simulations is performed. Experimental correlations are used to evaluate heat transfer performances of the obtained results. Simulated fluids are Lead-Bismuth-Eutectic, with a Prandtl number equal to 0.025, and liquid sodium, with $Pr = 0.01$.

Finally, results of turbulent heat transfer over vertical backward facing step are discussed in Chapter 6. In recent years this problem has been extensively studied for forced and mixed convection cases, with $Pr = 0.0088$. This is an interesting problem since a flow recirculation occurs in the region behind the step, for the forced convection case. On the other hand, for the mixed convection case the recirculation area is much smaller and the model is tested in the presence of buoyancy forces. The numerical platform is used to perform coupled simulations between OpenFOAM and FEMuS, investigating the effect of a different dynamical turbulence model coupled with the thermal turbulence model implemented in FEMuS.

Chapter 1

Finite Element Method

The mathematical representation of a physical problem is usually achieved through the definition of proper equations describing its time-space behavior. Since these equations contain often differentials operators with linear or non linear dependence between state variables and space-time coordinates analytical solutions can be obtained only for very limited simplified cases. For the case of incompressible Newtonian fluid flow and Fourier heat transfer, the equations governing velocity, pressure and temperature field in steady conditions can be written as

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0\,, \tag{1.1}$$

$$\mathbf{u} \cdot \boldsymbol{\nabla}\mathbf{u} = -\frac{1}{\rho}\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot \left[\nu \left(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^T\right)\right]\,, \tag{1.2}$$

$$\mathbf{u} \cdot \boldsymbol{\nabla} T = \boldsymbol{\nabla} \cdot (\alpha \boldsymbol{\nabla} T)\;. \tag{1.3}$$

The above set of equations contains the incompressibility constrain (1.1), a transport equation for momentum (1.2) and a transport equation for thermal energy (1.3), where $\mathbf{u}$, T and p are fluid velocity, temperature and pressure and ρ, ν and α fluid density, kinematic viscosity and thermal diffusivity, respectively. The system is defined over a domain Ω enclosed by a boundary Γ. Numerical methods have been developed with the intent to provide a way to compute a solution for a system like (1.1-1.3). The most popular are the Finite Difference Method (FDM), the Finite Volume Method (FVM) and the Finite Element Method (FEM). All these methods are built to transform the differential problem into an algebraic one, from a continuous to a discrete representation of (1.1-1.3). For any numerical approximation a discrete computational domain Ω^h, made of $\mathcal{C}$ cells and $\mathcal{N}$ nodes is used. With the FDM method the equations are solved on the $\mathcal{N}$ nodes of discretized Ω^h domain and derivatives are approximated by using Taylor series expansions. Both FVM and FEM methods are based on the solution of equations in its integral form. In the following a brief introduction on Finite Element Method is given since this is the numerical method used for the solution of Partial Differential Equations in

FEMuS. Also this introduction on FEM will allows us to specify technical details on numerical algorithms when this is required.

1.1 Mathematical formulation

1.1.1 Function spaces

Finite element method is based on the knowledge of functional analysis and function spaces. Function spaces are spaces where functions are characterized by common properties. Among all function spaces, vector spaces are of particular interest for this work. Let us introduce the vector space as: V is a vector space if and only if for any given $f, g \in \mathcal{V}$, the following are verified

$$f + g \in \mathcal{V}\,, \tag{1.4}$$

$$\alpha f \in \mathcal{V}\,, \tag{1.5}$$

with α being an arbitrary scalar value. We consider a domain Ω which is a bounded subset of R^N. The space $L^p(\Omega)$ of integrable function up to p-th power, over domain Ω

$$L^p(\Omega) := \left\{ f : \int_\Omega |f(x)|^p dx < \infty \right\}\,, \tag{1.6}$$

is a vector space. The space of continuous functions with continuous derivatives up to k-th order, over domain Ω, is labeled as $C^k(\Omega)$. Among this class of space functions, $C^0(\Omega)$ and $C_0^\infty(\Omega)$ represent the space of continuous functions and infinitely differentiable functions, with vanishing value along boundary Γ. *Sobolev spaces* $H^k(\Omega)$ are defined as spaces of functions belonging to $L^2(\Omega)$ and with derivatives belonging to $L^2(\Omega)$ up to order k

$$H^k(\Omega) := \left\{ f : f \in L^2(\Omega) \wedge f^i \in L^2(\Omega)\ \forall i \in [1, ..., k] \right\}\,. \tag{1.7}$$

Sobolev space $H^0(\Omega)$ coincides with the space of integrable functions $L^2(\Omega)$. Function differentiability is here considered in a *weak* sense. The first order derivative of the function $f(\mathbf{x})$ is commonly defined, in a strong sense, as

$$f'(\mathbf{x}) = \lim_{|\mathbf{h}| \to 0} \frac{f(\mathbf{x} + \mathbf{h}) - f(\mathbf{x})}{|\mathbf{h}|}\,. \tag{1.8}$$

In order to introduce the weak differentiability, we define the set of locally integrable functions $L^1_{loc}(\Omega)$ as [1]

$$L^1_{loc}(\Omega) := \{ f : f \in L^1(K)\ \forall\ compact\ K \subset\ interior\ \Omega \}\,. \tag{1.9}$$

For a given function $f \in L^1_{loc}(\Omega)$ a weak derivative $D_w f$ exists if there exists a function $g \in L^1_{loc}(\Omega)$ such that

$$\int_\Omega g(\mathbf{x})\, \psi(\mathbf{x})\, d\Omega = - \int_\Omega f(\mathbf{x})\, \psi'(\mathbf{x})\, d\Omega \,, \qquad \forall \psi(\mathbf{x}) \in C_0^\infty(\Omega) \,. \qquad (1.10)$$

If such a g exists then $D_w f = g$. This definition is satisfied for standard differentiable functions but it opens the possibility to extend the concept of derivative also for wider class functions.

A *norm* can be defined for a generic vector space $\mathcal{V}(\Omega)$ as a functional that associates a real number to each function $f(\mathbf{x}) \in \mathcal{V}(\Omega)$. A norm must satisfy the following properties

$$\begin{aligned} \|f\| &\geq 0 \,, \ \forall\, f \in \mathcal{V}(\Omega) \text{ and } \|f\| = 0 \Leftrightarrow f = 0 \\ \|\alpha f\| &= \alpha \|f\| \,, \ \forall\, f \in \mathcal{V}(\Omega) \text{ and } \alpha \in \mathbb{R} \\ \|f + g\| &\leq \|f\| + \|g\| \,, \ \forall\, f, g \in \mathcal{V}(\Omega) \,. \end{aligned}$$

For $L^n(\Omega)$ the natural norm can be defined as

$$\|f(\mathbf{x})\|_{L^n} = \left(\int_\Omega f(\mathbf{x})^n d\Omega \right)^{\frac{1}{n}} \,, \qquad (1.11)$$

while for $H^1(\Omega)$ the natural norm is

$$\|f(\mathbf{x})\|_{H^1} = \left(\int_\Omega \left(f(\mathbf{x})^2 + f'(\mathbf{x})^2 \right) d\Omega \right)^{\frac{1}{2}} \,, \qquad (1.12)$$

and so on for the other Sobolev spaces. For further clarification the interested reader can refer to [2]. A vector space endowed with the metric induced by its norm is called a normed linear space. A complete space $\mathcal{V}$ is defined as a metric space in which every Cauchy sequence $\{s_j\}$ of elements that belong to $\mathcal{V}$ converges to an element $s \in \mathcal{V}$. A *Banach space* is a normed linear space $(\mathcal{V}, \|\cdot\|)$ which is complete with respect to the metric induced by its norm. It can be proven that the Sobolev spaces defined above are all Banach spaces with respect to their natural norms. A *scalar product* over a linear vector space $\mathcal{V}$ can be defined as a bi-linear symmetric form that couples two elements $a(\mathbf{x}), b(\mathbf{x}) \in \mathcal{V}(\Omega)$ with a real number

$$\langle a(\mathbf{x}), b(\mathbf{x}) \rangle : \mathcal{V}(\Omega) \times \mathcal{V}(\Omega) \to \mathbb{R} \qquad (1.13)$$

which satisfies the following properties

$$
\begin{aligned}
\langle a(\mathbf{x}), a(\mathbf{x})\rangle &> 0 \;,\; \forall\, a(\mathbf{x}) \neq 0 \\
\langle a(\mathbf{x}), a(\mathbf{x})\rangle &= 0 \;,\; \textit{iff}\ a(\mathbf{x}) = 0 \\
\langle a(\mathbf{x}), b(\mathbf{x})\rangle \;&=\; \langle b(\mathbf{x}), a(\mathbf{x})\rangle \;,\; \forall\, a(\mathbf{x}), b(\mathbf{x}) \in \mathcal{V}(\Omega) \\
\langle \alpha a(\mathbf{x}) + b(\mathbf{x}), c(\mathbf{x})\rangle &= \alpha\langle a(\mathbf{x}), c(\mathbf{x})\rangle + \langle b(\mathbf{x}), c(\mathbf{x})\rangle \;, \\
&\qquad\qquad \forall\, a(\mathbf{x}), b(\mathbf{x}), c(\mathbf{x}) \in \mathcal{V}(\Omega) \text{ and } \alpha \in \mathbb{R} \;.
\end{aligned}
$$

Scalar products for Sobolev spaces $H^0(\Omega)$ and $H^1(\Omega)$ can be easily defined as

$$\langle f(\mathbf{x}), g(\mathbf{x})\rangle_{H^0} = \int_\Omega f(\mathbf{x})g(\mathbf{x})\; d\Omega \;, \tag{1.14}$$

$$\langle f(\mathbf{x}), g(\mathbf{x})\rangle_{H^1} = \int_\Omega \left(f(\mathbf{x})g(\mathbf{x}) + f'(\mathbf{x})\, g'(\mathbf{x})\right)\; d\Omega \;. \tag{1.15}$$

As a general rule, a norm can be defined from each space specific scalar product, i.e.

$$\|f(\mathbf{x})\|_{\mathcal{V}} = \langle f(\mathbf{x}), f(\mathbf{x})\rangle_{\mathcal{V}}^{\frac{1}{2}} \;. \tag{1.16}$$

A *Hilbert space* is finally defined as a function space with a scalar product and that is a Banach space with respect to the norm induced by its scalar product. Hilbert spaces $\mathcal{H}$ are characterized by some important properties. Since they are Banach spaces every Cauchy sequence of elements belonging to $\mathcal{H}$ converges to an element of $\mathcal{H}$. We consider a subset $\{e_i\}_{i=1}^{\infty}$ of functions e_i belonging to $\mathcal{H}$. The subset is called orthonormal if $||e_\alpha|| = 1$ for all α and $e_\alpha \perp e_\beta$ for any $\alpha \neq \beta$. If $\{e_i\}_{i=1}^{\infty}$ is an orthonormal subset then a general function $f \in \mathcal{H}$ can be expressed with a convergent series

$$f = \sum_{i=1}^{\infty} \langle f, e_i\rangle e_i \,. \tag{1.17}$$

The subset $\{e_i\}_{i=1}^{\infty}$ represents a basis for $\mathcal{H}$. The above mentioned Sobolev spaces $H^k(\Omega)$ are also Hilbert spaces.

1.1.2 Weak formulation

With the definitions given in the Section 1.1.1 it is possible to examine in which function spaces the variables contained in the system (1.1-1.3) should belong. From the incompressibility constrain (1.1) we have $\mathbf{u} \in C^1(\Omega)$ but from the momentum equation (1.2) $\mathbf{u} \in C^2(\Omega)$. For pressure and temperature fields we require

$p \in C^1(\Omega)$ and $T \in C^2(\Omega)$. If the physical properties ν and α depend on space coordinates they need to belong to $C^1(\Omega)$. These requirements are quite strong and they do not model a great number of real physical problem. Examples that does not have this solution regularity are many, for example two phase flows with two immiscible fluids with different viscosity μ, or cases with conjugate heat transfer where the thermal conductivity α is discontinuous. Similarly to the definition of differentiability in weak form, the weak formulation of a PDE is introduced to reduce function space requirements of the solved fields. The Weak formulation of a PDE problem is obtained by multiplying the equation with a smooth test function φ and then by integrating the product over the entire domain. We then have

$$\int_\Omega \boldsymbol{\nabla} \cdot \mathbf{u} \psi dx = 0 \,, \tag{1.18}$$

$$\begin{aligned}\int_\Omega \mathbf{u} \cdot \boldsymbol{\nabla} \mathbf{u} \cdot \boldsymbol{\varphi} dx = &- \int_\Omega \frac{1}{\rho} \boldsymbol{\nabla} p \cdot \boldsymbol{\varphi} dx + \\ &+ \int_\Omega \boldsymbol{\nabla} \cdot \left[\nu \left(\boldsymbol{\nabla} \mathbf{u} + \boldsymbol{\nabla} \mathbf{u}^T \right) \right] \cdot \boldsymbol{\varphi} dx \,,\end{aligned} \tag{1.19}$$

$$\int_\Omega \mathbf{u} \cdot \boldsymbol{\nabla} T \varphi dx = \int_\Omega \boldsymbol{\nabla} \cdot \left(\alpha \boldsymbol{\nabla} T \right) \varphi dx \,. \tag{1.20}$$

By using the divergence theorem, the diffusive contributions of (1.19) and (1.20) can be rewritten leading to the following form

$$\int_\Omega \boldsymbol{\nabla} \cdot \mathbf{u} \psi dx = 0 \,, \tag{1.21}$$

$$\begin{aligned}\int_\Omega \mathbf{u} \cdot \boldsymbol{\nabla} \mathbf{u} \cdot \boldsymbol{\varphi} dx = &- \int_\Omega \frac{1}{\rho} \boldsymbol{\nabla} p \cdot \boldsymbol{\varphi} dx + \int_\Gamma \left[\nu \left(\boldsymbol{\nabla} \mathbf{u} + \boldsymbol{\nabla} \mathbf{u}^T \right) \cdot \hat{\mathbf{n}} \right] \cdot \boldsymbol{\varphi} ds \\ &- \int_\Omega \left[\nu \left(\boldsymbol{\nabla} \mathbf{u} + \boldsymbol{\nabla} \mathbf{u}^T \right) \right] : \boldsymbol{\nabla} \boldsymbol{\varphi} dx \,,\end{aligned} \tag{1.22}$$

$$\int_\Omega \mathbf{u} \cdot \boldsymbol{\nabla} T \varphi dx = \int_\Gamma \alpha \boldsymbol{\nabla} T \cdot \hat{\mathbf{n}} \varphi ds - \int_\Omega \alpha \boldsymbol{\nabla} T \cdot \boldsymbol{\nabla} \varphi dx \,. \tag{1.23}$$

It is then obtained that $\mathbf{u} \in H^1(\Omega)$, $T \in H^1(\Omega)$, $\nu, \alpha \in L^2(\Omega)$. Pressure field should belong to $H^1(\Omega)$, but if $\boldsymbol{\nabla} p \cdot \boldsymbol{\varphi}$ is integrated by parts, then $p \in L^2(\Omega)$. A function space belonging is determined also for test functions ψ, $\boldsymbol{\varphi}$ and φ: $\psi \in L^2(\Omega)$, $\boldsymbol{\varphi} \in H^1(\Omega)$ and $\varphi \in H^1(\Omega)$. With integrals calculated over domain boundary Γ it is possible to impose boundary conditions. Boundary conditions are then classified as *natural* or *essential*. In the first case they directly appear within the equation, as for the case of imposed heat flux of viscous stress with terms

$$\int_\Gamma \alpha \boldsymbol{\nabla} T \cdot \hat{\mathbf{n}} \varphi ds \,, \quad \int_\Gamma \left[\nu \left(\boldsymbol{\nabla} \mathbf{u} + \boldsymbol{\nabla} \mathbf{u}^T \right) \cdot \hat{\mathbf{n}} \right] \cdot \boldsymbol{\varphi} ds \,.$$

The Dirichlet boundary conditions are classified *essential* since they modify the test function space in an essential way. A variable value fixed along a boundary Γ_d is an essential boundary condition which requires vanishing test function φ on Γ_d. If a Dirichlet boundary condition is set for temperature field on Γ_d then $\varphi \in H^1_{\Gamma_d}(\Omega)$. Essential boundary conditions therefore lead to a modification of function space where test functions belong.

1.1.3 The Finite Element

The definition of finite element based on Ciarlet's book can be expressed in the following way [3].
Definition *Let*

i) $\Omega_c \subseteq \mathbb{R}^n$ *be a bounded closed set with nonempty interior and piece-wise smooth boundary (the* **element domain***),*

ii) $\mathcal{P}$ *be a finite-dimensional space of functions on* Ω_c *(the* **shape functions***) and*

iii) $\mathcal{S} = \{S_1, S_2, ..., S_k\}$ *be a basis for* $\mathcal{P}'$ *(the set of* **nodal variables***)*

then (Ω_c, $\mathcal{P}$, $\mathcal{S}$*) is called a* **finite element**. The element Ω_c upon which basis functions are defined is labeled as canonical element. Different types of canonical elements can be defined, depending on the space dimension and on the Ω_c shape. The reference domain element Ω_c and the basis functions are defined *a priori*, in a sense that are independent from the geometry being studied. When the real domain Ω is discretized into Ω^h and has to be approximated with finite elements, then for each element Ω^h_j of Ω^h a transformation of coordinates is performed $\pi : \Omega^h_j \to \Omega_c$. This procedure allows to define an element where basis functions can be easily constructed and integrals can be easily calculated with the Gaussian quadrature. A very popular family of finite elements is the Lagrangian one. Starting from the mono-dimensional case, the canonical finite element is defined with coordinate ξ in the interval $[-1, 1]$. Construction of a piece-wise constant basis function $\varphi^0(\xi)$ is a trivial task as $\varphi^0(\xi)$ $\forall\xi \in [-1, 1]$. Basis functions have to be built with constraints, in particular they need to be linearly independent and the sum of all the basis functions, on each point of the canonical element, must be equal to 1. Linear and quadratic basis functions $\varphi^1(\xi)$ and $\varphi^2(\xi)$ may be constructed as follows

$$\varphi^1_0(\xi) = \frac{1}{2}(1 - \xi)\,, \tag{1.24}$$

$$\varphi^1_1(\xi) = \frac{1}{2}(1 + \xi)\,, \tag{1.25}$$

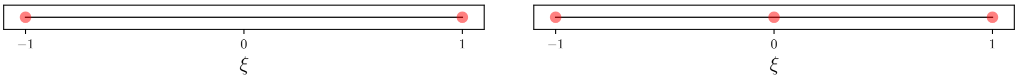


Figure 1.1: Mono dimensional canonical elements for linear and quadratic approximation.

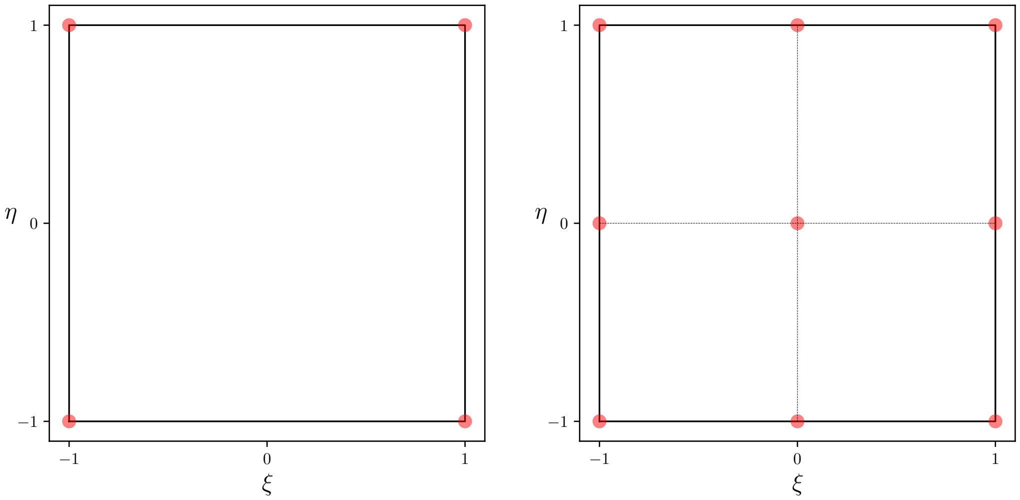


Figure 1.2: Two dimensional canonical elements for linear and quadratic approximation.

and

$$\varphi_0^2(\xi) = -\frac{1}{2}(1-\xi)\xi\,, \tag{1.26}$$

$$\varphi_1^2(\xi) = \frac{1}{2}(1+\xi)\xi\,, \tag{1.27}$$

$$\varphi_2^2(\xi) = (1+\xi)(1-\xi)\,. \tag{1.28}$$

For the linear case the nodes of the domain element are $N_0^1 := (\xi = -1)$ and $N_1^1 := (\xi = 1)$, while for the quadratic case the additional node $N_2^1 := (\xi = 0)$ is used, as sketched in Fig. 1.1 where linear and quadratic canonical domain elements, usually referred to EDGE2 and EDGE3, for one-dimensional case are reported. With superscript ^{1} we refer to the dimension 1 of the geometry. It can be easily seen that each basis function φ_i^o is equal to one on the node where the function is defined and zero on remaining nodes of the canonical domain element. This properties guarantees the linear independence of the basis functions. For each N_i^1 node of the canonical domain element, linear and quadratic basis function can

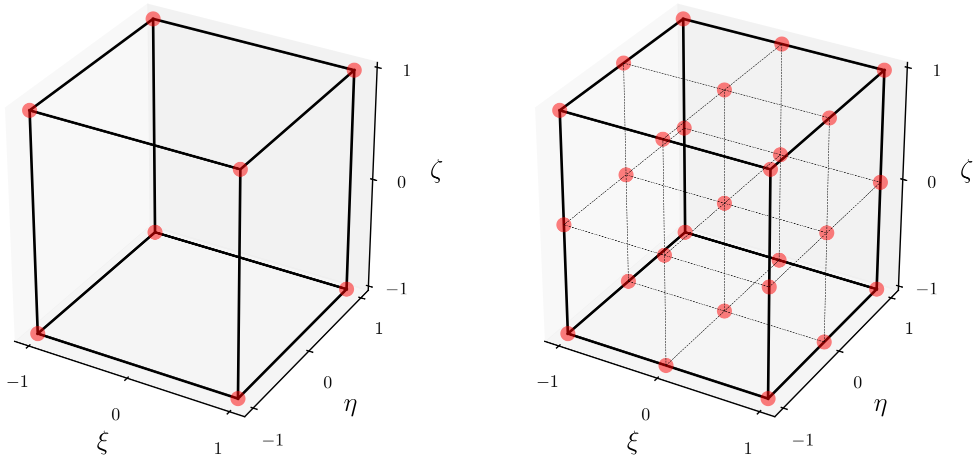


Figure 1.3: Three dimensional canonical elements for linear and quadratic approximation.

be calculated as

$$\varphi_i^1(\xi) = \frac{1}{2}(\xi_i\xi + 1)\,, \tag{1.29}$$

$$\varphi_i^2(\xi) = \left(1 - \frac{1}{2}|\xi_i|\right)\left[(2|\xi_i| - 1)\xi^2 + \xi_i\xi + (1 - |\xi_i|)\right]\,, \tag{1.30}$$

where ξ_i is the coordinate value of the node upon which basis function φ_i^o is defined, i.e. $\xi_0 = -1$ for node N_0^1. From these expressions, derivatives with respect to ξ can be easily computed

$$\frac{d\varphi_i^1}{d\xi} = \frac{\xi_i}{2}\,. \tag{1.31}$$

$$\frac{d^2\varphi_i^1}{d\xi^2} = 0\,, \tag{1.32}$$

$$\frac{d\varphi_i^2}{d\xi} = \left(1 - \frac{1}{2}|\xi_i|\right)\left[2(2|\xi_i| - 1)\xi + \xi_i\right]\,, \tag{1.33}$$

$$\frac{d^2\varphi_i^2}{d\xi^2} = 2\left(1 - \frac{1}{2}|\xi_i|\right)(2|\xi_i| - 1)\,, \tag{1.34}$$

Equations (1.29 – 1.34) are very useful for multi-dimensional canonical elements. Canonical elements for two and three-dimensional geometries are built as an extension of mono-dimensional elements EDGE2 and EDGE3 to additional coordinates η and ζ. Linear and quadratic Lagrangian canonical elements for two-dimensional cases are made of 4 and 9 nodes respectively and are usually referred

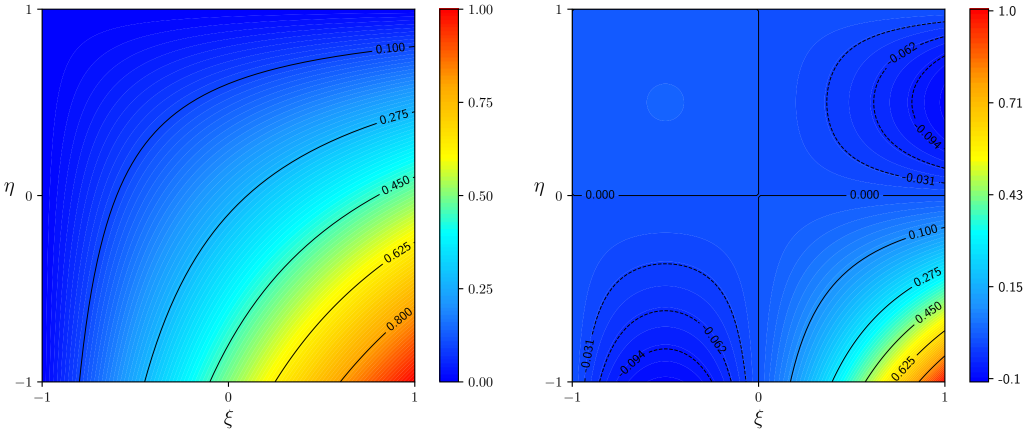


Figure 1.4: Comparison between linear (on left) and quadratic (on right) basis functions $\varphi_1^1(\xi,\eta)$ and $\varphi_1^2(\xi,\eta)$ defined on node N_1^2.

to QUAD4 and QUAD9. In the reference system $\xi - \eta$ the node $N_i^2(\xi,\eta)$ can be obtained as $N_i^2(\xi,\eta) = (N_k^1(\xi), N_l^1(\eta))$, resulting in

$$N_0^2 := (-1,-1)\,,\quad N_1^2 := (1,-1)\,,\quad N_2^2 := (1,1)\,,\quad N_3^2 := (-1,1)\,, \tag{1.35}$$

for canonical element QUAD4. Shape node functions for two-dimensional quadratic elements QUAD9 and for three-dimensional elements HEX8 and HEX27 can be easily derived, leading to elements represented in Fig. 1.2 and 1.3. Basis functions can be calculated with the following expressions

$$N_i^d \to \varphi_i^1(\boldsymbol{\xi}) = \prod_{j=1}^{d} \frac{1}{2}(\lambda_i\lambda + 1)\,, \tag{1.36}$$

$$N_i^d \to \varphi_i^2(\boldsymbol{\xi}) = \prod_{j=1}^{d} \left(1 - \frac{1}{2}|\lambda_i|\right) \left[(2|\lambda_i| - 1)\lambda^2 + \lambda_i\lambda + (1 - |\lambda_i|)\right] \tag{1.37}$$

where $\boldsymbol{\xi}$ is the vector of coordinates with dimension d and λ_j is the j-th coordinate of $\boldsymbol{\xi}$. This compact forms of linear and quadratic basis function, which is valid for 1D, 2D and 3D, can be easily implemented in a numerical code for the calculation of basis functions and their derivatives with a unique expression. A graphical comparison of linear and quadratic fields, for two-dimensional case, defined over node N_1^2 is given in Fig. 1.4. Therefore a function $f = f(\mathbf{x})$ can be approximated, on a discrete domain Ω_d, as

$$f(\mathbf{x}) = \sum_{i=1}^{n_d} f_i\varphi(\mathbf{x})\,, \tag{1.38}$$

where values f_i are calculated as

$$f_i = f(\mathbf{x_i}) \,\forall\, \mathbf{x_i} \in \Omega_d\,, \tag{1.39}$$

and n_d is the number of nodes used to discretize the real domain Ω.

1.2 Navier Stokes discretization

The variational formulation of the Navier-Stokes equation can be written as

$$\int_\Omega \boldsymbol{\nabla} \cdot \mathbf{u}\psi dx = 0 \qquad \forall \psi \in L^2(\Omega)\,, \tag{1.40}$$

$$\begin{aligned}\int_\Omega \mathbf{u} \cdot \boldsymbol{\nabla}\mathbf{u} \cdot \boldsymbol{\varphi} dx &= \int_\Omega \frac{1}{\rho} p \boldsymbol{\nabla} \cdot \boldsymbol{\varphi} dx - \int_\Omega \left[\nu \left(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^T\right)\right] : \boldsymbol{\nabla}\boldsymbol{\varphi} dx \\ &+ \int_\Gamma \left[\nu \left(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^T\right) \cdot \hat{\mathbf{n}} - \frac{1}{\rho} p \hat{\mathbf{n}}\right] \cdot \boldsymbol{\varphi} ds \qquad \forall \boldsymbol{\varphi} \in H^1(\Omega)\,,\end{aligned} \tag{1.41}$$

where the pressure contribution term has been integrated by parts. The discrete variational formulation of the Navier-Stokes equation is obtained by approximating the continuous domain Ω with the discretized one Ω_h and by introducing finite dimensional function spaces ${}^hL^2(\Omega_h) \subseteq L^2(\Omega)$ and ${}^hH^1(\Omega_h) \subseteq H^1(\Omega)$. In the discrete space the unknown functions $\mathbf{u}$ and p are restricted to a finite set of unknown values defined on N_h nodes of Ω_h. The discrete weak formulation is then simply obtained as

$$\int_{\Omega_h} \boldsymbol{\nabla} \cdot \mathbf{u}\psi dx = 0 \quad \forall \psi \in^h L^2(\Omega_h)\,, \tag{1.42}$$

$$\begin{aligned}\int_{\Omega_h} \mathbf{u} \cdot \boldsymbol{\nabla}\mathbf{u} \cdot \boldsymbol{\varphi} dx &= \int_{\Omega_h} \frac{1}{\rho} p \boldsymbol{\nabla} \cdot \boldsymbol{\varphi} dx - \int_{\Omega_h} \left[\nu \left(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^T\right)\right] : \boldsymbol{\nabla}\boldsymbol{\varphi} dx \\ &+ \int_{\Gamma_h} \left[\nu \left(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^T\right) \cdot \hat{\mathbf{n}} - \frac{1}{\rho} p \hat{\mathbf{n}}\right] \cdot \boldsymbol{\varphi} ds \quad \forall \boldsymbol{\varphi} \in^h H^1(\Omega_h)\,.\end{aligned} \tag{1.43}$$

Since (1.42 – 1.43) are valid for each test function ψ, $\boldsymbol{\varphi}$ belonging to ${}^hL^2(\Omega_h)$ and ${}^hH^1(\Omega_h)$, a system of $\mathcal{N}^h$ equations can be written by taking the discrete variational formulation (1.42 – 1.43) for each test function defined on the discrete domain. Once that $\mathbf{u}$ and p have been written with their finite element representation

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^{N_h} \mathbf{u}_i \varphi_i(\mathbf{x})\,, \quad p(\mathbf{x}) = \sum_{i=1}^{N_h} p_i \psi_i(\mathbf{x})\,, \tag{1.44}$$

then the system of equations can be written in matrix from as

$$\begin{pmatrix} A & B \\ B^T & C \end{pmatrix} \begin{pmatrix} \mathbf{u}_h \\ p_h \end{pmatrix} = \begin{pmatrix} f_u \\ f_p \end{pmatrix} , \tag{1.45}$$

Where $\mathbf{u}_h$ and p_h are ordered arrays representing the unknown velocity and pressure values, A is a squared matrix whose coefficients $A_{i,j}$ express a dependence between velocity component values $\mathbf{u}_{h,i}$ and $\mathbf{u}_{h,j}$, B is a matrix that relates velocity to pressure and C is a squared matrix containing non null coefficients only in the case that Dirichlet boundary conditions are imposed for pressure field. For the Navier-Stokes equation many boundary conditions can be formulated. The integral over boundary Γ of equation (1.43) can be written as

$$\begin{aligned} &\int_{\Gamma_h} \left[\nu \left(\boldsymbol{\nabla} \mathbf{u} + \boldsymbol{\nabla} \mathbf{u}^T \right) \cdot \hat{\mathbf{n}} - \frac{1}{\rho} p \hat{\mathbf{n}} \right] \cdot \boldsymbol{\varphi} ds = \int_{\Gamma_h} \left[(\boldsymbol{\sigma} \cdot \hat{\mathbf{n}}) \cdot \boldsymbol{\varphi} \right] ds = \\ &\qquad = \int_{\Gamma_h} \left[\left(\boldsymbol{\tau} - \frac{1}{\rho} \mathbf{I} p \right) \cdot \hat{\mathbf{n}} \right] \cdot \boldsymbol{\varphi} ds \,, \end{aligned} \tag{1.46}$$

where $\mathbf{I}$ is the identity matrix. Some of them are defined below, with $\hat{\mathbf{n}}$ and $\hat{\mathbf{t}}$ being unit vectors in normal and tangential direction of boundary side Γ

- *Fixed velocity*: all velocity components are set with Dirichlet boundary condition,
- *Pressure inlet/outlet*: Dirichlet boundary condition on pressure and $\mathbf{u} \cdot \hat{\mathbf{t}} = 0$
- *No stress*: $(\boldsymbol{\sigma} \cdot \hat{\mathbf{n}}) \cdot \hat{\mathbf{n}} = 0$ and $(\boldsymbol{\sigma} \cdot \hat{\mathbf{n}}) \cdot \hat{\mathbf{t}} = 0$,
- *Slip*: $\mathbf{u} \cdot \hat{\mathbf{n}} = 0$ and free stress along tangential direction, i.e. $(\boldsymbol{\sigma} \cdot \hat{\mathbf{n}}) \cdot \hat{\mathbf{t}} = 0$,
- *Stress*: $\mathbf{u} \cdot \hat{\mathbf{n}} = 0$ and prescribed stress along tangential direction, i.e. $(\boldsymbol{\tau} \cdot \hat{\mathbf{n}}) \cdot \hat{\mathbf{t}} = \lambda$,
- *Outflow*: Fixed pressure value and free stress along tangential direction, $(\boldsymbol{\tau} \cdot \hat{\mathbf{n}}) \cdot \hat{\mathbf{t}} = 0$.

Imposition of these boundary conditions can appear as a trivial task when boundary sides are oriented as the reference frame, but can be quite a complex task for the case of tilted boundaries. Since many boundary conditions set restrictions on values assumed by certain variables along normal or tangential direction of boundary sides, it is convenient to decompose velocity field $\mathbf{u}$ as sum of a normal vector $\mathbf{u}_{\perp}$ and a parallel vector $\mathbf{u}_{||}$, calculated as

$$\mathbf{u}_{\perp} = (\mathbf{u} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}} \,, \quad \mathbf{u}_{||} = (\mathbf{u} \cdot \hat{\mathbf{t}})\hat{\mathbf{t}} \,. \tag{1.47}$$

In the light of above mentioned effects of Dirichlet boundary conditions on test function belonging space, for the discretization of Navier-Stokes equation Hilbert

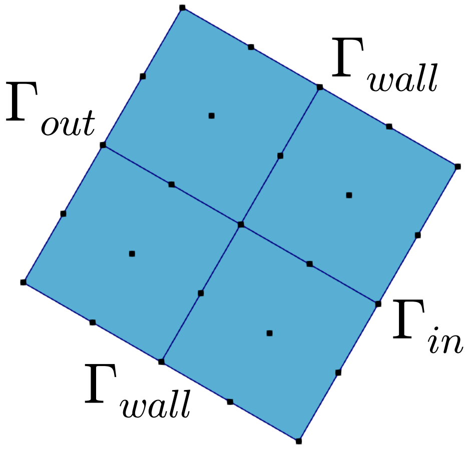


Figure 1.5: Channel mesh for the benchmark of Navier-Stokes discretization.

spaces $H^1_{\Gamma_\perp}(\Omega)$ and $H^1_{\Gamma_\parallel}(\Omega)$ are defined as spaces $H^1(\Omega)$ of vector functions $\boldsymbol{\varphi}$ whose normal and parallel components to boundary Γ are zero on Γ, so

$$\mathbf{u} \cdot \hat{\mathbf{n}} = 0 \text{ on } \Gamma \rightarrow \boldsymbol{\varphi} \in H^1_{\Gamma_\perp}(\Omega)\,, \tag{1.48}$$

$$\mathbf{u} \cdot \hat{\mathbf{t}} = 0 \text{ on } \Gamma \rightarrow \boldsymbol{\varphi} \in H^1_{\Gamma_\parallel}(\Omega)\,. \tag{1.49}$$

From (1.43) one can see that the variational formulation of Navier-Stokes equation becomes a scalar equation. For a two-dimensional case, it is common to use test functions $\boldsymbol{\varphi}$ calculated as

$$\boldsymbol{\varphi} = \varphi \begin{pmatrix} 1 \\ 0 \end{pmatrix}\,, \quad \boldsymbol{\varphi} = \varphi \begin{pmatrix} 0 \\ 1 \end{pmatrix}\,, \quad \varphi \in H^1(\Omega)\,. \tag{1.50}$$

Therefore two separate equations for velocity components u and v, $\mathbf{u} = (u, v)$, can be written. On those boundaries where either conditions (1.48) or (1.49) are set it is convenient to use the following expressions

$$\boldsymbol{\varphi} = \varphi \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}\,, \quad \boldsymbol{\varphi} = \varphi \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}\,, \quad \varphi \in H^1(\Omega)\,,$$

where n_1, n_2 and t_1, t_2 are the components of unit vectors $\hat{\mathbf{n}}$ and $\hat{\mathbf{t}}$ respectively. In this way a linear combination of equations for u and v is operated, obtaining an equation for tangential or normal velocity component, depending on the imposed boundary condition in order to satisfy the requirements on test function space. As a simple benchmark test it may be considered the case of tilted plane channel. Starting from an initial velocity profile aligned with channel axis, if the same boundary conditions are imposed on wall sides then a symmetric solution should

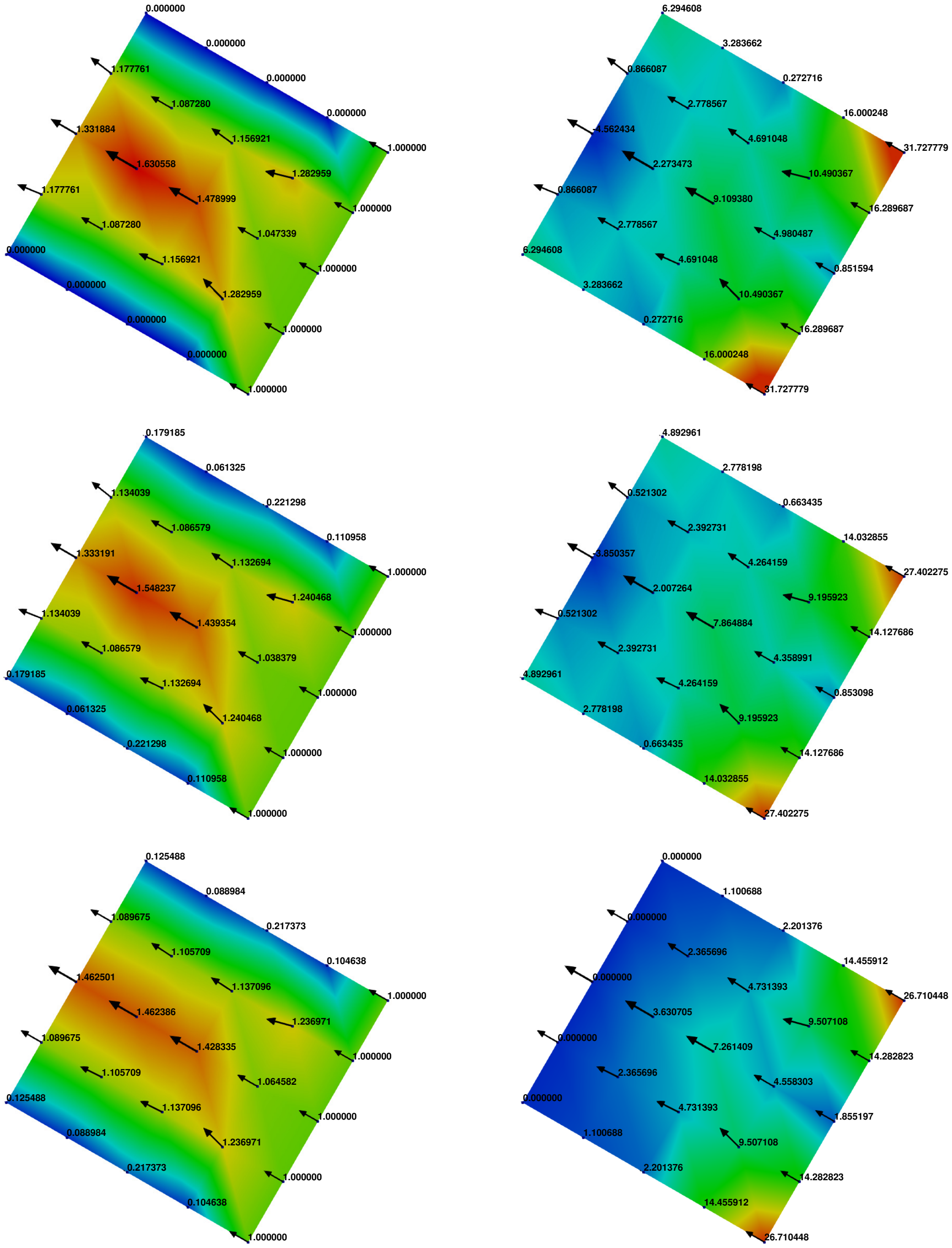


Figure 1.6: Velocity field (on left) and pressure field (on right), with respective node values, for various boundary conditions set on wall and outlet sides. From top to bottom: CASE A, CASE B and CASE C. Results obtained for Navier-Stokes discretization using (1.50) on boundaries.

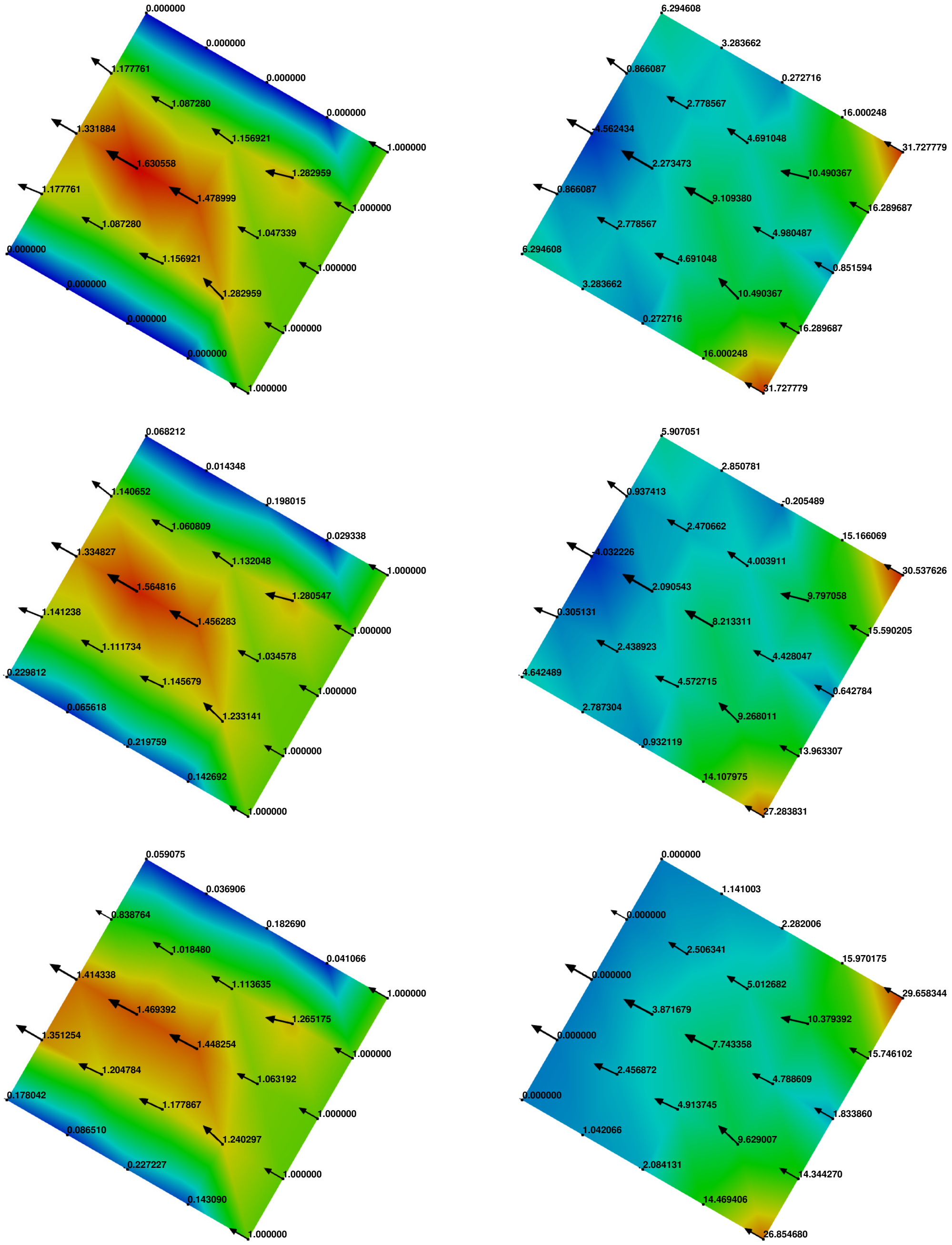


Figure 1.7: Velocity field (on left) and pressure field (on right), with respective node values, for various boundary conditions set on wall and outlet sides. From top to bottom: CASE A, CASE B and CASE C. Results obtained for Navier-Stokes discretization without using (1.50) on boundaries.

be achieved. A simple channel geometry consisting of a 2×2 QUAD9 elements is simulated and sketched in Fig. 1.5. Wall, inlet and outlet boundaries is labeled as Γ_w, Γ_i and Γ_o. For a given boundary Γ_a with normal and tangent unit vectors $\mathbf{n}$ and $\mathbf{t}$ respectively, we introduce the following test function spaces:

$$H^1_{\Gamma_a,\perp}(\Omega) \subset H^1(\Omega) : \boldsymbol{\varphi} \in H^1_{\Gamma_a,\perp}(\Omega) \text{ \textit{iff} } \boldsymbol{\varphi} \in H^1(\Omega) \wedge \boldsymbol{\varphi} \cdot \mathbf{n} = 0\,, \quad (1.51)$$

$$H^1_{\Gamma_a,\parallel}(\Omega) \subset H^1(\Omega) : \boldsymbol{\varphi} \in H^1_{\Gamma_a,\perp}(\Omega) \text{ \textit{iff} } \boldsymbol{\varphi} \in H^1(\Omega) \wedge \boldsymbol{\varphi} \cdot \mathbf{t} = 0\,. \quad (1.52)$$

Three different cases are considered:

- CASE A: the simplest one, with no slip boundary condition on wall surfaces and outflow on outlet section. This combination of b.c. does not set any particular restriction on test function components along boundary normal and tangential directions. The function space is then $\boldsymbol{\varphi} \in H^1_{\Gamma_w}(\Omega)$, i.e. $\boldsymbol{\varphi} = 0$ along boundary Γ_w.
- CASE B: a stress boundary condition is set on wall boundaries and outflow on outlet section. Stress boundary condition implies that velocity field, on Γ_w, has a zero component along wall normal direction, so that $\boldsymbol{\varphi} \in H^1_{\Gamma_{w,\perp}}(\Omega)$.
- CASE C: a stress and pressure outlet boundary conditions are set on wall and outlet sections respectively. In addition to the restriction set in case CASE B, a zero tangential component of velocity field is imposed on outlet section, so $\boldsymbol{\varphi} \in H^1_{\Gamma_{w,\perp},\Gamma_{o,\parallel}}(\Omega)$.

Results obtained with Navier-Stokes discretization discussed above, for the three different cases, are reported in Fig. 1.6. In particular, on the left velocity field with node values are reported on the left of the Figure, while on the right pressure values together with pressure field are shown. It can be seen that symmetry is preserved for each case, as expected.

Results for the same cases, obtained without using the test functions defined as in (1.50) on boundary sides, are reported in Fig. 1.7. It can be seen that for CASE A symmetry is obtained, as expected since no particular restriction is set on $\boldsymbol{\varphi}$. For case CASE B a slight non symmetry in the velocity field modulus is observed, and it is more marked for CASE C. Boundary conditions described above can be extended to three-dimensional cases. While for two-dimensional cases tangent direction is uniquely defined, in three-dimensional cases two families of tangent directions $\hat{\mathbf{t}}_1$ and $\hat{\mathbf{t}}_2$ can be defined imposing $\hat{\mathbf{t}}_1 \cdot \hat{\mathbf{t}}_2 = 0$. More elaborate boundary conditions can then be thought, in particular if $\hat{\mathbf{t}}_1$ is taken parallel to mean flow direction and $\hat{\mathbf{t}}_2$ as $\hat{\mathbf{t}}_2 = \hat{\mathbf{n}} \times \hat{\mathbf{t}}_1$. In particular stresses $\boldsymbol{\tau}$ can be applied on the direction parallel to mean flow, $\hat{\mathbf{t}}_1$, or along tangent direction $\hat{\mathbf{t}}_2$. Different cases of stress are then considered

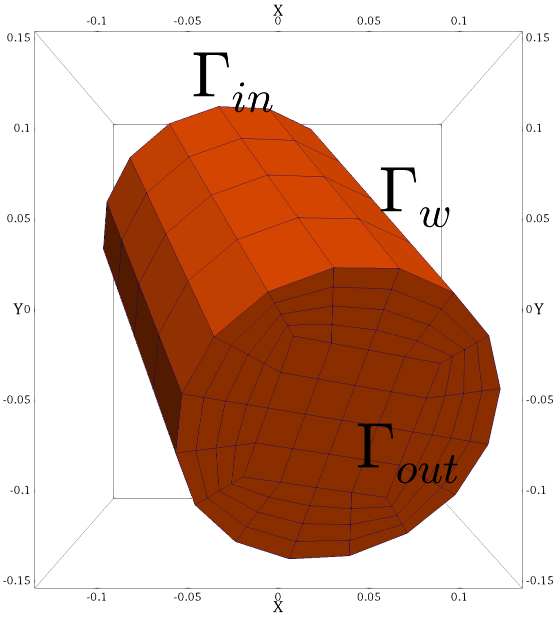


Figure 1.8: Tilted cylinder geometry used to test Navier-Stokes discretization.

- $(\boldsymbol{\tau} \cdot \hat{\mathbf{n}}) \cdot \hat{\mathbf{t}}_\mathbf{1} = \lambda |\mathbf{u} \cdot \hat{\mathbf{t}}_\mathbf{1}|$ wall stress proportional to velocity component along mean flow direction. This stress can be applied either to increase or to decrease velocity component $\mathbf{u} \cdot \hat{\mathbf{t}}_\mathbf{1}$.
- $(\boldsymbol{\tau} \cdot \hat{\mathbf{n}}) \cdot \hat{\mathbf{t}}_\mathbf{2} = \lambda |\mathbf{u} \cdot \hat{\mathbf{t}}_\mathbf{1}|$ stress imposed to accelerate or decelerate velocity component $\mathbf{u} \cdot \hat{\mathbf{t}}_\mathbf{2}$ along direction $\hat{\mathbf{t}}_\mathbf{1}$, inducing a rotation in the resulting velocity field.

The above mentioned boundary conditions are tested in a three-dimensional simulation of a tilted cylinder, as shown in Fig. 1.8, with a mesh made of 40 HEX27 cells. Two different cases are analyzed where symmetric solutions are expected. In Fig. 1.9 result are reported for the case of pressure outlet boundary condition, on outlet section, and decelerating stress $(\boldsymbol{\tau} \cdot \hat{\mathbf{n}}) \cdot \hat{\mathbf{t}}_\mathbf{1} = \lambda |\mathbf{u} \cdot \hat{\mathbf{t}}_\mathbf{1}|$ boundary condition on wall surfaces. For convenience the velocity field is decomposed as sum of $\mathbf{u}_{\parallel}$ and $\mathbf{u}_{\perp}$, where the first is the velocity field aligned with cylinder axis and the latter is the velocity field that lays on cylinder cross sections. Being $\hat{\mathbf{c}}$ the unit vector aligned with cylinder axis, the two velocity fields are obtained as

$$\mathbf{u}_{\parallel} = (\mathbf{u} \cdot \hat{\mathbf{c}})\hat{\mathbf{c}} \,, \quad \mathbf{u}_{\perp} = \mathbf{u} - \mathbf{u}_{\parallel} \,. \tag{1.53}$$

On top of Fig. 1.9 velocity field vectors are reported, while on bottom fields of $\mathbf{u}_{\parallel}$ and $\mathbf{u}_{\perp}$ are shown with representative numerical node values to see that a

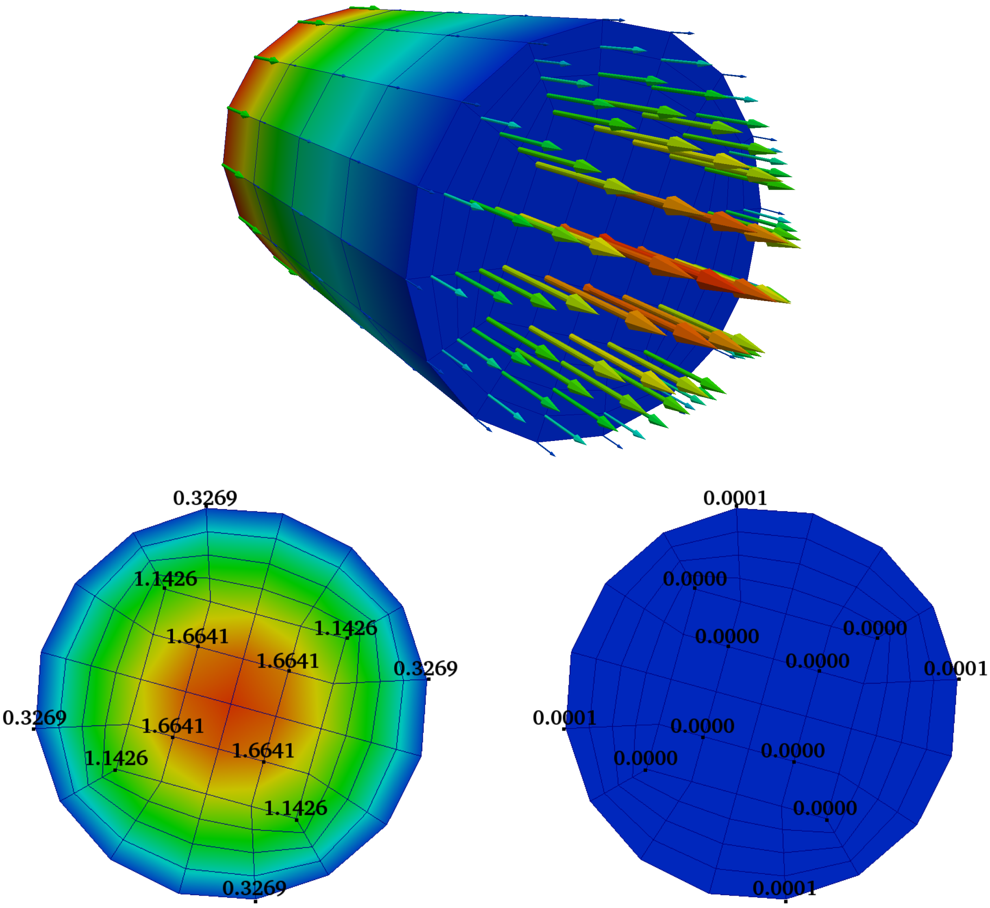


Figure 1.9: Results for tilted cylinder simulation with induced rotating secondary flow boundary condition. On top velocity field decomposition into $\mathbf{u}_{||}$ and $\mathbf{u}_{\perp}$. On bottom outlet view of $\mathbf{u}_{||}$ (on left) and $\mathbf{u}_{\perp}$ (on right) values.

symmetric solution has been obtained. Pressure outlet boundary condition require $\mathbf{u}_{\perp}$ to be zero. On wall surface the stress boundary condition leads to decreasing $\mathbf{u}_{||}$ values along the stream-wise direction. For velocity field $\mathbf{u}_{\perp}$, a stress condition in the form $(\boldsymbol{\tau} \cdot \hat{\mathbf{n}}) \cdot \hat{\mathbf{t}}_{\mathbf{2}} = \lambda |\mathbf{u} \cdot \hat{\mathbf{t}}_{\mathbf{2}}|$ has been seen. Depending on the sign of λ, the condition would lead to a decrease or an increase of $(\mathbf{u} \cdot \hat{\mathbf{t}}_{\mathbf{2}})\hat{\mathbf{t}}_{\mathbf{2}}$ velocity field. For the case of cylinder geometry a null velocity field $(\mathbf{u} \cdot \hat{\mathbf{t}}_{\mathbf{2}})\hat{\mathbf{t}}_{\mathbf{2}}$ is obtained for a sufficiently high value λ. In particular, if a very high value of lambda is set, then the stress boundary condition can be used to set a zero value of $(\mathbf{u} \cdot \hat{\mathbf{t}}_{\mathbf{2}})\hat{\mathbf{t}}_{\mathbf{2}}$ in a weak sense.

Results for the case of stress boundary condition $(\boldsymbol{\tau} \cdot \hat{\mathbf{n}}) \cdot \hat{\mathbf{t}}_{\mathbf{2}} = \lambda |\mathbf{u} \cdot \hat{\mathbf{t}}_{\mathbf{1}}|$ applied on wall surface and outflow b.c. on outlet section are shown in Fig. 1.10. For

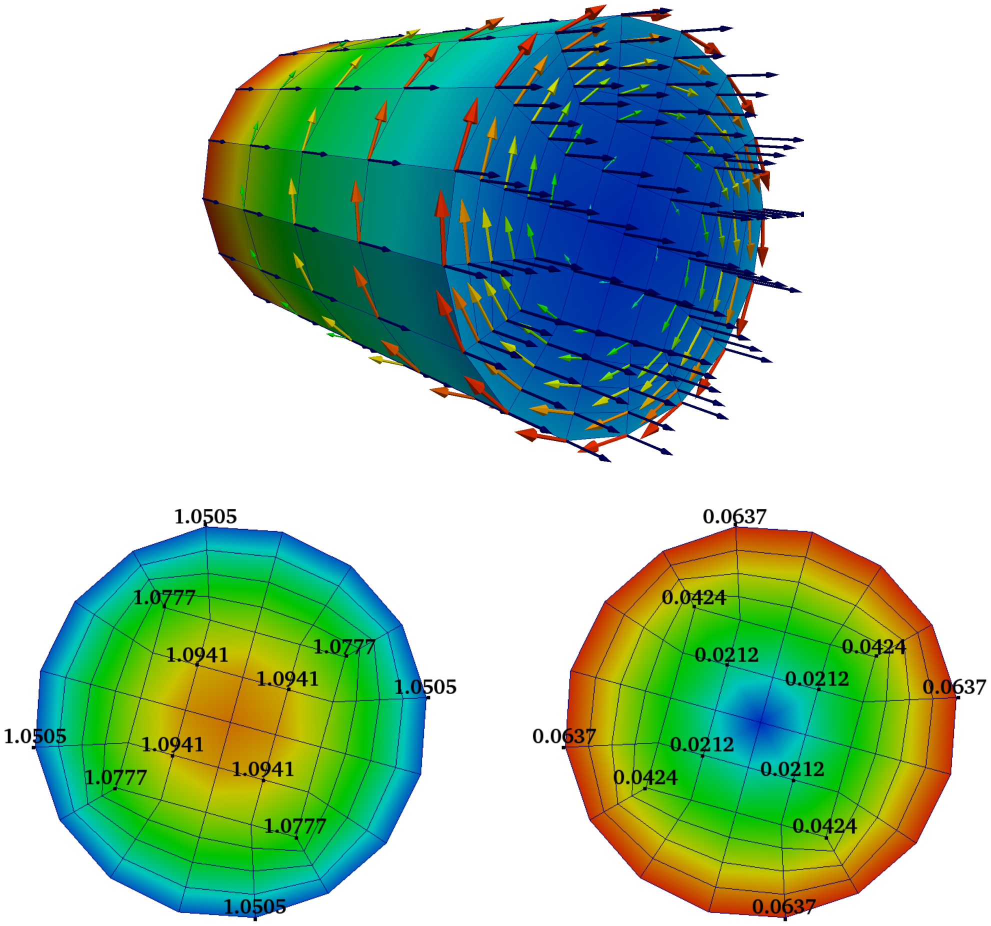


Figure 1.10: Results for tilted cylinder simulation with induced rotating secondary flow boundary condition. On top velocity field decomposition into $\mathbf{u}_{||}$ and $\mathbf{u}_{\perp}$. On bottom outlet view of $\mathbf{u}_{||}$ (on left) and $\mathbf{u}_{\perp}$ (on right) values.

velocity component $(\mathbf{u} \cdot \hat{\mathbf{t}}_1)\hat{\mathbf{t}}_1$ a no stress boundary condition has been set on the wall. On the top of Fig. 1.10 the decomposition of the velocity field as sum of $\mathbf{u}_{||}$ and $\mathbf{u}_{\perp}$ is represented with vectors for each velocity field. As already mentioned before, this stress boundary condition induces a rotating velocity field on cylinder cross section and the magnitude of $\mathbf{u}_{\perp}$ increases along stream-wise direction. As one can see from the outlet section on the bottom of Fig. 1.10 the $\mathbf{u}_{||}$ and $\mathbf{u}_{\perp}$ values reported are symmetric.

Chapter 2

Computational platform

Nowadays the rapid increase of available computational power allows scientists and engineers to perform numerical simulations of complex systems that can involve many scales and several different physical phenomena. During the years, many numerical codes have been developed to solve different problems ranging from physics and mathematics to engineering and biology. In Fig. 2.1 some open-source codes are reported with relative physical modeling. Among them OpenFOAM, TrioCFD and Saturne are specialized for fluid dynamic simulations, with modules for compressible and incompressible Navier-Stokes equations, multiphase flows and turbulence models [4, 5, 6]. Code-Aster [7] contains thermomechanical solvers to study elasticity problems, fracture propagation, effect of

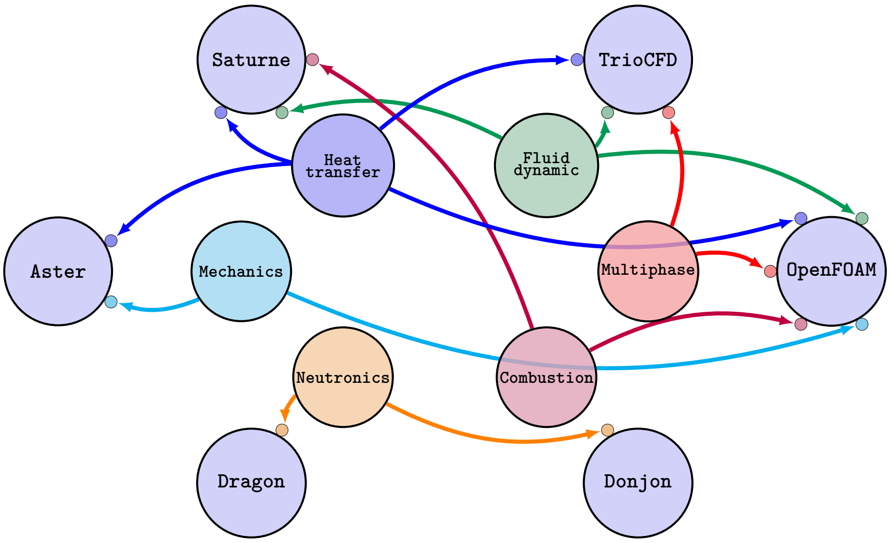


Figure 2.1: Available open source codes with relative implemented physical problems.

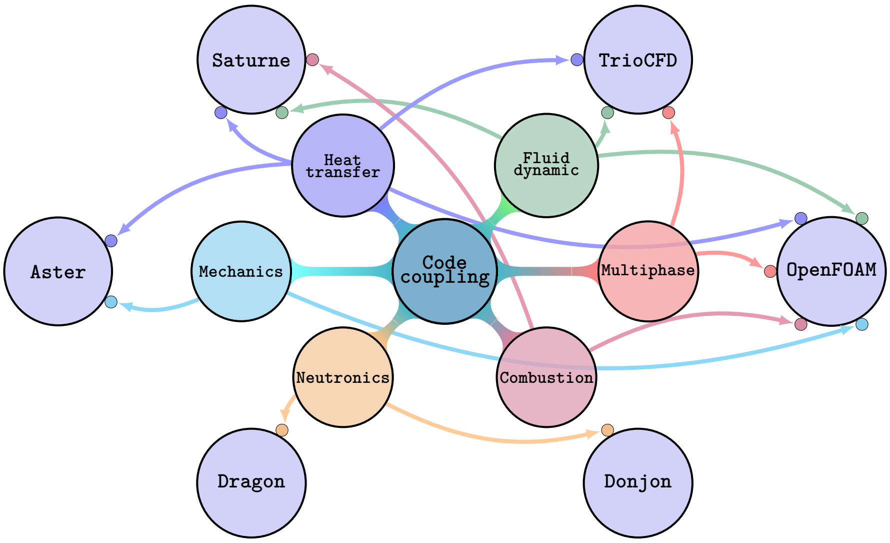


Figure 2.2: Data exchange diagram of numerical code coupling.

nuclear irradiation and many others. Dragon and Donjon codes are used to solve neutronic problems [8]. In particular the first code solves the neutron transport equation on fuel assembly geometries and allows to calculate nuclear properties such as cross sections. Donjon code is used to solve neutron multi-group diffusion equations on wider geometries. Many codes are available, with overlapping physical model “areas”, but none of them allows to simulate a highly complex problem involving structural mechanics, fluid flow, heat transfer and neutron diffusion. Researchers and scientists can follow different approaches to deal with a highly complex simulation. One main approach is to develop a new numerical code where all the physical phenomena of interest are modeled. Another possibility is to couple already existing and validated codes. With the first option a huge amount of time and effort are needed to create such a code while with the latter it is possible to take advantage of validated codes and expertise developed in years. From the computational platform point of view, development of numerical codes can be performed where needed, avoiding overlapping areas between different codes. The data exchange diagram of numerical code coupling is shown in Fig. 2.2 where the possible interaction among different physical problems is shown. It is clear that with the coupling modeling approach, efficient methods for data exchange are needed so that code communication does not act as a bottleneck and does not reduce simulation performances. The design of a “computational environment” where complex physical phenomena can be modeled is a major scope of this PhD work, alongside with turbulence modeling. Within the platform the numerical codes are added as “modules”, their execution is controlled and the data exchange is organized. Since the final goal is to simulate turbulent flows of low

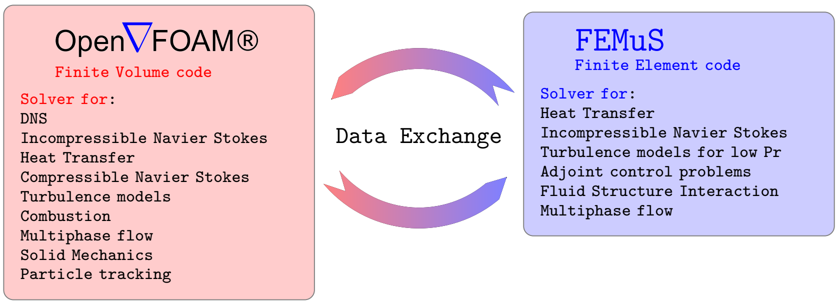


Figure 2.3: Coupling between FEMuS and OpenFOAM codes.

Pr fluids, a coupling between FEMuS [9] and OpenFOAM [4] codes has been realized. In this way the thermal turbulence modeling implemented in FEMuS [10, 11, 12] can be coupled with the wide choice of dynamical turbulence models implemented in OpenFOAM, as has been considered in Chapter 6. A schematic representation of the coupling between the two different codes and of the several modules implemented in OpenFOAM and FEMuS is given in Fig. 2.3. The developed computational platform can be found at [13].

In this Chapter the problem of numerical code coupling is addressed in order to realize the computational platform. After a brief introduction on the SALOME platform, the coupling methods used in this thesis are explained, together with routines that have been developed to facilitate the data exchange between different codes. Particular attention is paid to FEMuS and OpenFOAM codes, since they have been used to perform the simulations discussed in this thesis. A case of natural convection in a square cavity is used to test the coupling procedure between FEMuS and OpenFOAM, providing guidelines that have been used also for the simulation of turbulent flows over vertical backward facing step, as discussed in Chapter 6.

2.1 SALOME Platform

Numerical platform SALOME has been developed by Commissariat à l'énergie atomique (CEA) and Électricité de France (EDF) with the intent to provide an advanced open-source platform for Computer Aided Engineering (CAE) purposes [14]. Moreover SALOME platform aims at offering means that can be used as a *standard* for numerical code integration and coupling. Among the many modules integrated within SALOME platform, the most important ones are GEOM, for the creation of CAD models, MESH, for the realization of computational grids and PARAVIS, a software for data visualization and post processing. Additional modules are YACS, for numerical code execution control and data exchange orga-

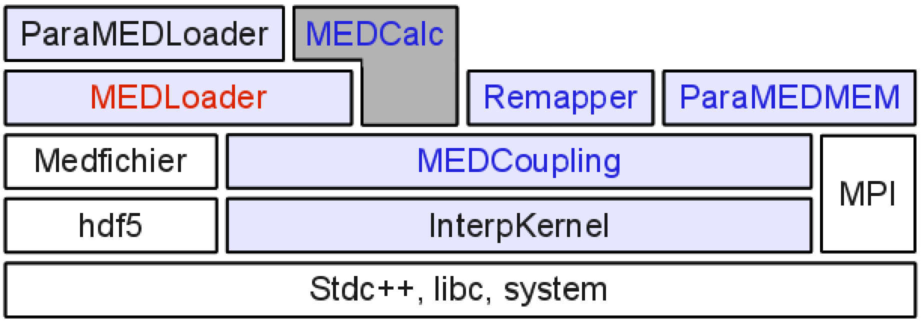


Figure 2.4: Layer architecture of MED library. Image taken from `http://docs.salome-platform.org/latest/dev/MEDCoupling/library.html`

nization into a workflow, and MED, a very interesting library for data storage and manipulation that will be described in the following. From this brief description of SALOME platform it is clear that the platform can be used to deal with each computational simulation step, i.e. geometry and mesh generation, code(s) execution and solution visualization.

MED module. The MED module is referred to as a standard mean to handle numerical fields and meshes. Three main libraries are provided. The “MED file” is a C and Fortran API that implements functions for reading and writing fields and meshes on file. The produced files, with *.med* extension, are based on HDF format, which is an efficient format for storing large amount of data. The “MED memory” is a C++ and python API that allows to store meshes and fields in computer memory. A numerical field can then be read from file, using MED file library, and stored in memory by using MED memory library. A huge set of functions is provided with MED memory library. They are used to access, modify, create meshes and perform operations on fields. With the MED memory library data exchange between numerical codes can be performed at memory level, avoiding read/write operations and data access on hard disk, which would act as a bottleneck to simulation performances. Finally the MED CORBA is an application programming interface (API) written to facilitate the distributed computation inside SALOME platform, providing functions for data accessing. MED module is also present at a Graphical User Interface level and contains a set of functions that allows to perform data manipulation on data stored in med file format. When MED library is observed with deeper detail, than its layer structure can be sketched, as in Fig. 2.4. In this Figure each block depends on the block it covers along the vertical line. Therefore ParaMEDLoader functions depend from the MEDLoader library that depends from MEDFicher and so on. It is easy to see that the whole library is written in C++ and depends from its standard library. The most important

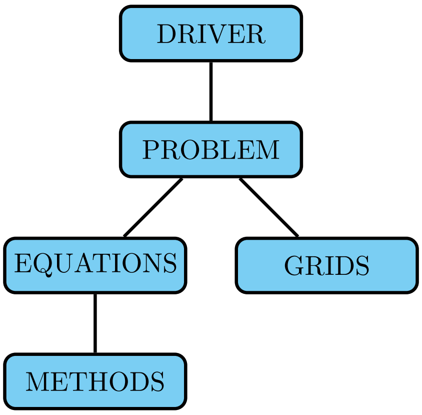


Figure 2.5: Sketch of code optimal structure for integration in the computational platform.

modules are MEDLoader, MEDCalc, Remapper, MEDCoupling and InterpKernel. MEDLoader provides functions for input/output of fields and meshes in med file format. The module is based on Medfichier, the MED file format, that writes and reads on hfd5 format. MEDCalc module is the GUI mentioned above, while MEDCoupling contains functions for arithmetic operations and interpolation. In particular interpolation can be made using InterpKernel functionalities for point location algorithms and determination of mesh overlapping, i.e. cell intersections. Remapper class actually performs the field transfer from source to receiver support.

2.2 Code optimal structure and interfaces

The optimal structure of a numerical code, for its integration on the computational platform, is here discussed [15, 16]. Five representative classes are introduced to depict the basic functions needed to implement the code inside the computational platform and control its execution. These five are labeled as *driver*, *problem*, *equations*, *grids*, *methods*. The code structure is sketched in Fig. 2.5. The structure follows a top-bottom hierarchical order, with *driver* class being the top level one. This class directly communicate with the *supervisor*, which can be SALOME platform, if YACS module is used to organize the data exchange workflow, or simply the `main` function of a C++ program. The *problem* class is specialized for different physics and should contain three basic functions: *setType*, *setMesh* and *solve*. The first command sets the problem type (Navies-Stokes, energy, etc.), the second one sets and prepares the mesh that should be available in both MED and code specific formats for data exchange and, finally, the *solve* command controls the solution of the discrete system. The *equations* class inherits the system particular

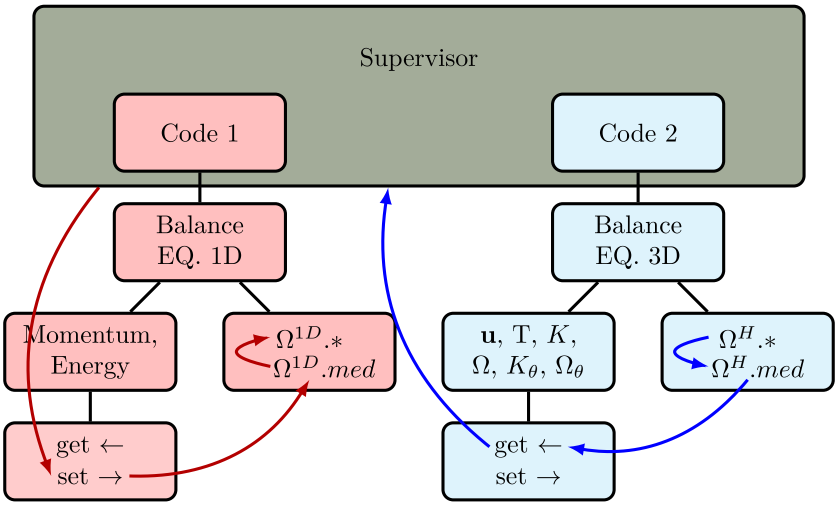


Figure 2.6: Example of workflow between two different codes controlled by a supervisor.

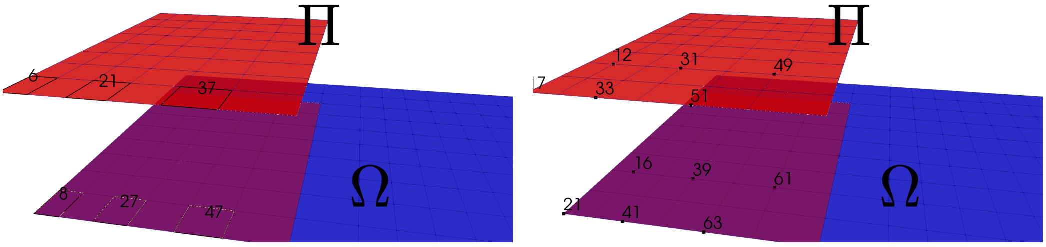


Figure 2.7: Different cells and nodes numbering between code grid Ω and interface MED duplicate Π.

class which contains the assembly and solver of the generic code. The solution is here user accessible and a field can be extracted or set by using *methods* class with specific functions *setField* and *getField*. Finally the *grids* class is an extension of the mesh class, contained inside *problem*, and allows the exchange of data between different mesh formats, i.e. the code specific one and MED. The representation of the workflow followed during a coupling operation is reported in Fig. 2.6. Code 2 and Code 1 are communicating in a process where a numerical field is extracted from Code 2 and given to Code 1. At Supervisor level, through the driver class, the solution is stored into a MED duplicate of Code 2 mesh, created by *grids* class, by *getField* function of *methods* class. The numerical field is now available in computer memory as a MED memory object and, with an inverse process, can be stored as solution of Code 1 after a proper field manipulation has been done at Supervisor level. It is then convenient to introduce code interfaces: these objects mix the functionalities of both *grids* and *methods* class. An interface is created

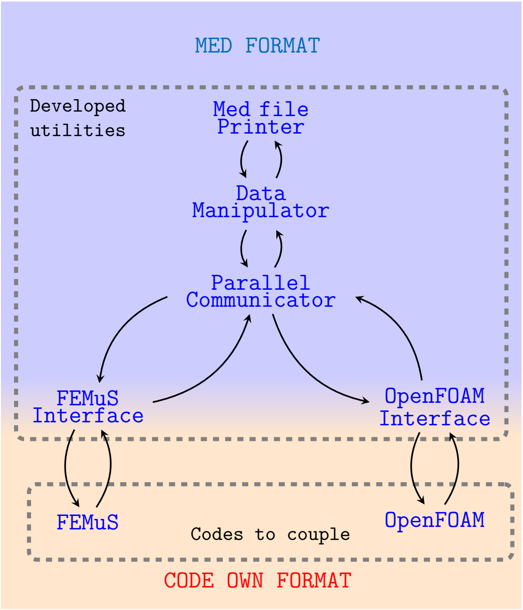


Figure 2.8: Workflow for numerical code coupling between FEMuS and OpenFOAM and developed utilities.

on a part of the computational grid Ω, that can be either a volume or a boundary side. A MED grid duplicate Π of the interested mesh part is created together with maps π^i that allow to associate nodes and cells numbering of Ω to nodes and cells numbering of Π, as described in Fig. 2.7. Through the interface object functions *getField* and *setField* can be called so that a MED field can be obtained or set a solution after a proper solution reordering with maps π^i.

2.3 Developed routines

A closer view to the workflow occurring during code coupling is given in Fig. 2.8. Here Code 1 and Code 2 structures are replaced by FEMuS and OpenFOAM with relative interface structures. Operations that take place at Supervisor level are those standing on light blue background and are all defined on MED memory structures, while code own data format operation area is represented by light red background. Interfaces are built to work between code own format and MED data structure. The routines reported in this Figure have been developed during this PhD study, namely `Interface`, `Parallel Communicator` and `Data Manipulator`. Class `Med file printer` has also been developed as a wrapper of MED functions for printing meshes and numerical fields on file with *.med* extension. From the simplified workflow depicted in Fig. 2.8 the important thing

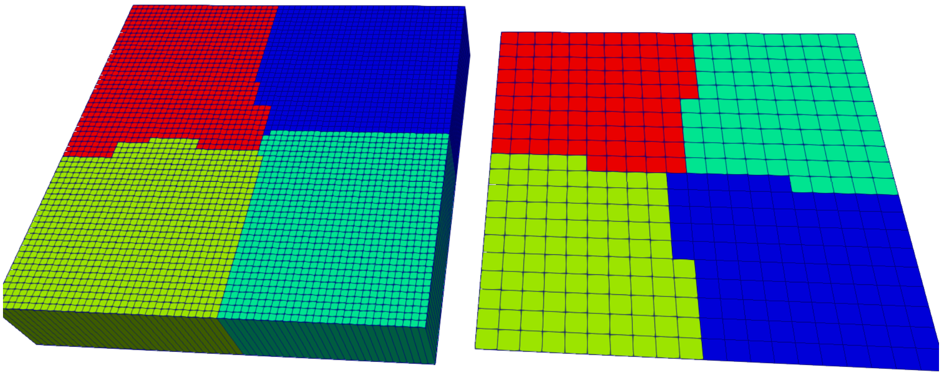
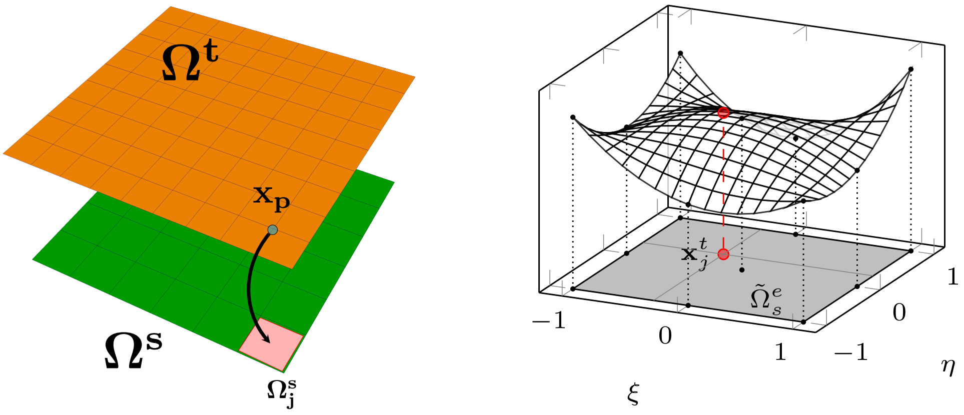


Figure 2.9: Example of different meshes that can be used when coupling two different numerical codes. Colors show domain partitioning for parallel computation.

to notice is that the coupling procedure is not very intrusive, within the single code, as the main modification that has to be made is the addition of the interface class. The advantage of using MED data structure as a standard one is that the above mentioned developed classes are not tailored for a specific code but can be generally used where MED library is used. When performing a coupling between different codes, as a general rule different meshes, i.e. with different space dimensions, domain discretization and partitioning for parallel computation, are used. An example is given in Fig. 2.9 where two different meshes are reported. By referring to FEMuS - OpenFOAM coupling described in Fig. 2.8, three-dimensional mesh on the left of Fig. 2.9 is the one used in OpenFOAM as three-dimensional computational grids are needed also for two-dimensional simulations, while the grid on the right is the one used in FEMuS. The different colors show the domain partitioning used for these two meshes. It is then necessary to provide means for solution gathering and scattering, but also functions for data manipulation in order to meet each code numerical scheme requirements. The first task is accomplished by `Parallel Communicator` class that makes use of MPI library functions for gathering and scattering MED fields and meshes on/to selected processors. The latter task is handled by `Data Manipulator`. MED library offers several functions for data manipulation and interpolation. In particular interpolation functions are provided for $P0$ and $P1$ fields, i.e. piece-wise and node-wise with linear approximation, that can be either conservative or non conservative depending on the expressed field nature, intensive or extensive and with maximum value preserving or conservative integral value constrain. The offered algorithms can then be used for $P0 \leftrightarrow P0$, $P0 \leftrightarrow P1$ and $P1 \leftrightarrow P1$. In `Data Manipulator` class a new point-wise $P2 \leftrightarrow P2$ interpolation algorithm has been developed, together with



a)

b)

Figure 2.10: Point-wise $P2 \leftrightarrow P2$ interpolation: target mesh node location on source grid a) and interpolation, within canonical element, after position reconstruction b).

a function for Gauss integration of $P2$ fields and a function for a conservative transformation of $P2$ fields into $P0$ fields. These routines have been developed since $P2$ approximation is widely used in FEMuS code and they will be used to couple FEMuS and OpenFOAM codes, as will be shown in the following. A more complete set of interpolation functions is then made available. It is remarked that the developed routines can be generally used in presence of MED library.

2.3.1 Point-wise $P2 \leftrightarrow P2$ field projection

A projection function is needed when a source field Φ_s has to be interpolated from the source mesh Ω_s to a target grid Ω_t in order to obtain target field Φ_t. The basic idea is to calculate each Φ_t node value using finite element representation of Φ_s. The $P2P2$ projection relies on a search point algorithm, where for each target mesh node $\mathbf{x}^t_j$ the source mesh element Ω^e_s containing $\mathbf{x}^t_j$ is determined, as shown in Fig. 2.10 a). Then a reconstruction step is needed, in order to locate the $\mathbf{x}^t_j$ node within $\tilde{\Omega}^e_s$ which is the transformed Ω^e_s element into canonical finite element Ω^e. The finite element representation of Φ_s can be used and interpolation weights can be calculated. Finally the value of Φ_t on node $\mathbf{x}^t_j$ can be determined, as shown in Fig. 2.10 b). Field projection can be expressed in matrix form as

$$\mathbf{\Phi}_t = \boldsymbol{P}\mathbf{\Phi}_s \,, \tag{2.1}$$

where $\mathbf{\Phi}_s$ and $\mathbf{\Phi}_t$ are the array containing the values of fields Φ_s and Φ_t calculated on the nodes of Ω_s and Ω_t, respectively, and the matrix $\boldsymbol{P}$ is the projection operator. The implementation of the projection algorithm can be divided in two parts:

the construction of $\boldsymbol{P}$ and the projection of $\boldsymbol{\Phi}_s$.

Inverse mapping

Let us consider two different discrete computational domains Ω_s and Ω_t divided into $\mathcal{N}_s^e$ and $\mathcal{N}_t^e$ finite elements, respectively. Let $\boldsymbol{x}_s$ and $\boldsymbol{x}_t$ be the $\mathcal{N}_s$ and $\mathcal{N}_t$ coordinates of the nodes in Ω_s and Ω_t, respectively. Let us also introduce a space dependent scalar field

$$\Phi = \Phi(\boldsymbol{x}) \, , \tag{2.2}$$

where $\boldsymbol{x}$ is the spatial coordinate. A finite approximation of Φ is a vector $\boldsymbol{\Phi}$ in which (2.2) is evaluated over a discrete vector of coordinates. We can now introduce $\boldsymbol{\Phi}_s$ based on the nodes belonging to Ω_s as

$$\boldsymbol{\Phi}_s = \Phi(\boldsymbol{x}_s) \, . \tag{2.3}$$

Now, we can evaluate s in any position through the approximation $\boldsymbol{s}_s$ using

$$\Phi_i(\boldsymbol{x}) = \sum_{e=1}^{\mathcal{N}_s^e} \sum_{j=1}^{n_e} \Phi_{s,j}^e(\boldsymbol{x}_j^{s,e}) \varphi_j^e(\boldsymbol{x}) \, , \tag{2.4}$$

where $\boldsymbol{x}_j^{s,e}$ and $s_{s,j}^e$ are the values of $\boldsymbol{x}_s$ and $\boldsymbol{s}_s$ of the node j in the element e, respectively. $\varphi_j^e(\boldsymbol{x})$ are the approximation functions and n_e is the number of nodes that compose the element e. In our case those functions belong to the Lagrange family and have different properties that allows them to approximate the solution. Further information about different interpolating functions can be found in [17, 18]. It is convenient to introduce a local coordinate system in the element e so that the interpolation functions have always the same expression. We now consider the transformation π that transforms the local canonical element into the original element, defined as

$$\pi = \sum_{j=1}^{n_e} \boldsymbol{x}_j^e \varphi_j^e(\boldsymbol{\xi}_j^e) \, , \tag{2.5}$$

where $\boldsymbol{\xi}_j^e$ are the coordinates of the points in the local system of reference. If an isoparametric element formulation is adopted, then the transformation π is a quadratic one and curved elements can be considered. If the source mesh Ω_s is made of straight elements, then a subparametric element formulation can be used, with a linear transformation π [3]. Finally we can now approximate the field $\Phi(\mathbf{x})$ and rewrite (2.4) as

$$\begin{cases} \Phi_i(\boldsymbol{x}) = \sum_{e=1}^{\mathcal{N}_s^e} \sum_{j=1}^{n_e} \boldsymbol{\Phi}_{s,j}^e(\boldsymbol{x}_j^{s,e}) \varphi_j^e(\boldsymbol{\xi}) \, , \\ \boldsymbol{x}_j^{s,e} = \pi(\boldsymbol{\xi}_j^e) \, , \end{cases} \tag{2.6}$$

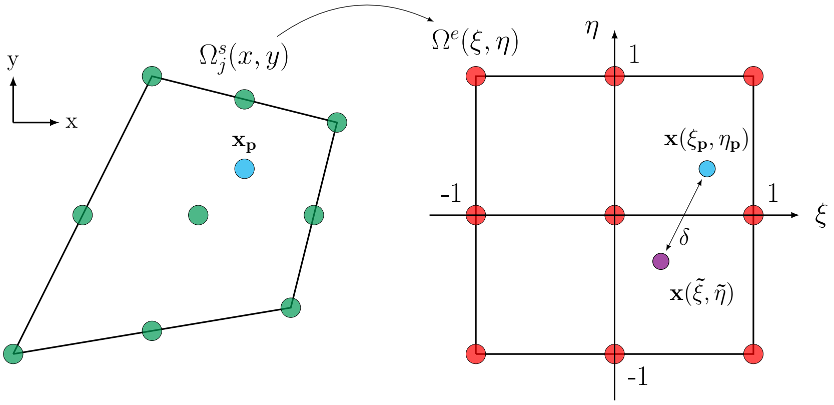

$\Omega_j^s(x,y)$
$\Omega^e(\xi,\eta)$
y
x
$\mathbf{x_p}$
η
1
$\mathbf{x}(\xi_\mathbf{p},\eta_\mathbf{p})$
-1
1
ξ
δ
$\mathbf{x}(\tilde{\xi},\tilde{\eta})$
-1

Figure 2.11: Reconstruction of node **x** position inside canonical element $\Omega^e(\xi,\eta)$.

or

$$\Phi_i(\boldsymbol{x}) = \sum_{e=1}^{\mathcal{N}_s^e} \sum_{j=1}^{n_e} \Phi_{s,j}^e(\boldsymbol{x}_j^{s,e}) \varphi_j^e(\pi^{-1}(\boldsymbol{x}_j^{s,e}))\,. \tag{2.7}$$

We can now define a projection operator that project the field $\boldsymbol{\Phi}_s$ from Ω_s to Ω_t as

$$\boldsymbol{\Phi}_t = \boldsymbol{P}\boldsymbol{\Phi}_s = \sum_{e=1}^{\mathcal{N}_s^e} \sum_{j=1}^{n_e} \boldsymbol{\Phi}_{s,j}^e \varphi_j^e(\pi^{-1}(\boldsymbol{x}_j^{t,e}))\,. \tag{2.8}$$

We remark that the interpolation functions are different from zero only inside the element that contains the point on which the field is interpolated. The interested reader can find more details on the interpolation of a computational field in [17, 18]. From equation (2.8) the mapping π is required to be invertible [3, 1].

Construction of operator $\boldsymbol{P}$

This projection operator is a $n \times m$ matrix where n is the number of nodes in the target mesh while m is the number of nodes of the source one. The $i-th$ row of $\boldsymbol{P}$ contains the coefficient that interpolates the field from some values of the source mesh into the $i-th$ node of the target one. In order to build the matrix $\boldsymbol{P}$ we proceed as follows. For every node of the target mesh Ω_t we find the cell (Ω_s^*) in the source mesh domain Ω_s containing the selected node. This functionality is provided by MEDCoupling library, as will be shown in the paragraph regarding the numerical implementation of the P2P2 algorithm, and is based on intersection calculation between cells and balls. Coordinates ($\boldsymbol{\xi}$) of the selected node in the canonical element transformation of Ω_s^* must then be calculated. In some cases the calculation of $(\boldsymbol{\xi})$ can be performed analytically but we use an iterative procedure that can be generally applied [19]. With the symbol $\boldsymbol{x}$ we refer to the coordinates of a node of Ω_t and with $\tilde{\boldsymbol{x}}$ we indicate the coordinates evaluated with the canonical coordinates $\boldsymbol{\xi}$, so $\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}}(\boldsymbol{\xi})$. A function f that represents the squared distance between points $\boldsymbol{x}$ and $\tilde{\boldsymbol{x}}$ can be usefully defined. In this way when $f(\boldsymbol{\xi}^e) = 0$ then $(\boldsymbol{\xi}^e)$ is the set of coordinates that exactly reproduces the node $\boldsymbol{x}$ in the canonical element, as graphically shown in Fig. 2.11 where δ is the squared root value of f. The function f is always positive and the tuple $\boldsymbol{\xi}^e$ represents an absolute minimum of f, so $f(\boldsymbol{\xi}^e)_{,i} = 0$ for each $i = 1, \dots, n_t$ where n_t is the size of $\boldsymbol{\xi}$ and $f(\boldsymbol{\xi}^e)_{,i}$ indicates the partial derivative of f with respect to the i-th coordinate of the canonical element. The search of the coordinates $\boldsymbol{\xi}$ is then transformed into a problem for the minimization of f. If we indicate with $\hat{\boldsymbol{x}}_i$ the set of nodes that defines the element in the donor mesh Ω_s containing $\boldsymbol{x}$ then f can be written as

$$f(\boldsymbol{\xi}) = \sum_{j=1}^{d} \left[x^j - \sum_{i=0}^{n} \hat{x}_i^j \varphi_i(\boldsymbol{\xi}) \right]^2 , \tag{2.9}$$

where x^j is the coordinate in the *j*-th direction and φ_i is the interpolating functions defined on the *i*-th cell node. In (2.9) d denotes the space dimension and n the dimension of the node set $\hat{\boldsymbol{x}}_i$. The minimum of the function f is calculated with a line search strategy where the descent direction is computed with a Newton's method. We start from an initial guess point $\boldsymbol{\xi}_0$ and then at the generic iteration *k* the new point $\boldsymbol{\xi}_k$ is calculated as

$$\boldsymbol{\xi}_k = \boldsymbol{\xi}_{k-1} + \Delta\boldsymbol{\xi} \;\rightarrow\; \boldsymbol{\xi}_k = \boldsymbol{\xi}_{k-1} - \gamma \mathbf{H}^{-1} \cdot \boldsymbol{\nabla} f , \tag{2.10}$$

where γ is equal to 0.5 and $\mathbf{H}$ is the Hessian matrix of f. The expression of the first and second derivatives of f are respectively

$$f(\boldsymbol{\xi})_{,l} = -2 \sum_{j=1}^{d} \left[\left(x^j - \sum_{i=1}^{n} \hat{x}_i^j \varphi_i(\boldsymbol{\xi}) \right) \sum_{i=1}^{n} \hat{x}_i^j \varphi_{i,l}(\boldsymbol{\xi}) \right] \tag{2.11}$$

and

$$\begin{aligned} f(\boldsymbol{\xi})_{,lk} = -2 \sum_{j=1}^{d} \Bigg[& \left(x^j - \sum_{i=1}^{n} \hat{x}_i^j \varphi_i(\boldsymbol{\xi}) \right) \sum_{i=1}^{n} \hat{x}_i^j \varphi_{i,lk}(\boldsymbol{\xi}) + \\ & - \sum_{i=1}^{n} \hat{x}_i^j \varphi_{i,l}(\boldsymbol{\xi}) \sum_{i=1}^{n} \hat{x}_i^j \varphi_{i,k}(\boldsymbol{\xi}) \Bigg] . \end{aligned} \tag{2.12}$$

The algorithm runs till iteration n is reached, namely when $f(\boldsymbol{\xi}_n)$ is smaller than a threshold value. When the local coordinate of i-th target mesh node is found the values of the interpolating functions on that node can be calculated and stored into $\boldsymbol{P}$.

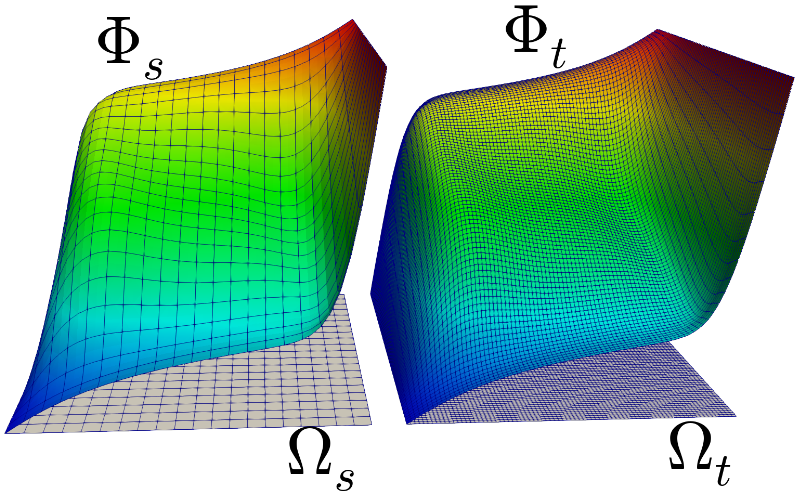


Figure 2.12: Point-wise interpolation example of a quadratic field Φ_s from a source grid Ω_s to a target Ω_t.

Field Projection

Once that operator $\boldsymbol{P}$ has been built the field projection step is performed as $\boldsymbol{\Phi}_t = \boldsymbol{P}\boldsymbol{\Phi}_s$. If the projection algorithm is applied to a case where the mesh changes with time then the operator $\boldsymbol{P}$ must be computed every time the projection step is performed, otherwise it can be just computed once. An example of point-wise field interpolation is given in Fig. 2.12 where a source field Φ_s has been interpolated from computational grid Ω_s to a refined one Ω_t. A successful implementation of this projection algorithm has been provided in [20], where a Fluid Structure Interaction (FSI) problem has been coupled with a Multiphase problem. A moving mesh algorithm is in fact used for the FSI problem in order to better represent the solid domain deformation, while a fixed Cartesian grid is used to compute the advection of the phase field with Volume Of Fluid method (VOF). The point-wise projection algorithm allows the projection of the velocity field from the time changing mesh of the FSI problem to the structured Cartesian grid of the VOF algorithm, for a more accurate prediction of multi-phase advection. The test application in [20] involved projection over the entire simulated domain but the algorithm can be used also for boundary field projection, as described in [21], where a numerical code coupling between multidimensional domains has been realized with decomposed domain method. In this case communication between different codes occur only on boundary interfaces. The algorithm has been developed as a tool able to facilitate numerical code coupling but can be used also within a code, allowing to change the mesh during the simulation, so that the

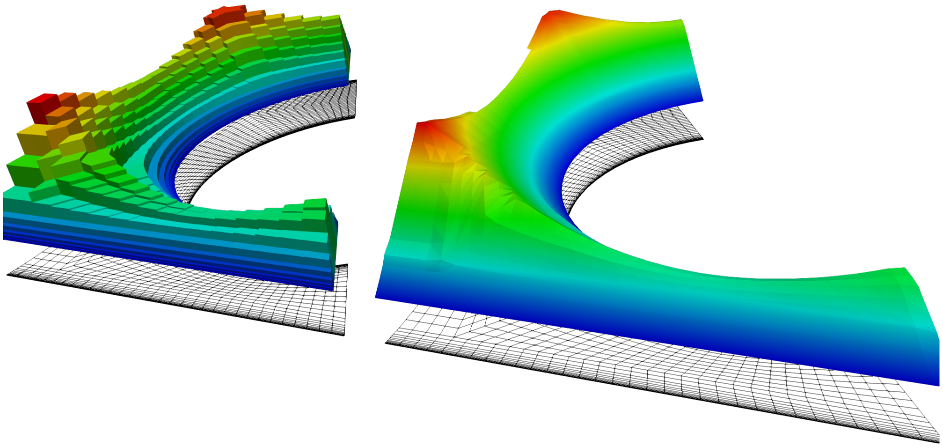


Figure 2.13: Example of Galerkin interpolation of piece-wise into node-wise field representing wall distance.

simulation can be started on a coarser grid and then continued on a finer one.

2.3.2 Galerkin point-wise projection

The algorithm described in the previous section can be used to project numerical fields, with both P1 or P2 approximation, between different computational grids. A different interpolation technique is the Galerkin projection scheme. This particular method can be retrieved from a reasoning in terms of L_2 norm [22]. Again, a generic field Φ_s, defined on a source computational domain Ω_s with $\mathcal{N}_s$ basis functions φ_s, needs to be interpolated on a target computational grid Ω_t that is made of $\mathcal{N}_t$ nodes and basis functions φ_t. Interpolated field Φ_t can be seen as the optimal interpolation in the L_2 norm, namely

$$(||\Phi_s - \Phi_t||)^2 = \min_{\Phi \in \mathcal{V}_t} (||\Phi_s - \Phi||_{L^2})^2 \ , \tag{2.13}$$

where $\mathcal{V}_t$ is the set of all possible fields that can be obtained as linear combination of φ_t, i.e. $\mathcal{V}_t = \text{span}\{\varphi_t\}$. Minimization of the L_2 norm requires that the $\mathcal{N}_t$ derivatives of $||\Phi_s - \Phi||_{L^2}$ with respect to the coefficients of Φ are zero

$$\int_\Omega \frac{\partial}{\partial \Phi_i} (\Phi_s - \Phi)^2 dV = 0 \, . \tag{2.14}$$

Field Φ can be written as linear combination of basis functions φ_t, leading to

$$\int_\Omega \frac{\partial}{\partial \Phi_i}\left(\Phi_s - \sum_1^{\mathcal{N}_t} \Phi_i \varphi_t^i\right)^2 dV = \\ = \int_\Omega 2\varphi_t^i \left(\Phi_s - \sum_1^{\mathcal{N}_t} \Phi_i \varphi_t^i\right) dV = 0\,, \\ \forall i \in \{1, ..., \mathcal{N}_t\}\,, \quad \varphi_t^i \in L^2(\Omega)\,. \tag{2.15}$$

Equation 2.15 can be rewritten in terms of Φ_t, obtaining

$$\int_\Omega \varphi_t^i \Phi_t dV = \int_\Omega \varphi_t^i \Phi_s dV\,, \quad \forall i \in \{1, ..., \mathcal{N}_t\}\,, \quad \varphi_t^i \in L^2(\Omega)\,. \tag{2.16}$$

Equation (2.16) is known as Galerkin projection and is obtained through the minimization of the L_2 norm of field difference between source Φ_s and interpolated Φ_t fields. Differently from the algorithm previously described, Galerkin projection requires the solution of a system of equations, the discretized system obtained from (2.16). The implementation of this interpolation scheme is not a straightforward task. Equation (2.16) can be rewritten by expanding both Φ_s and Φ_t with relative basis functions φ_s and φ_t

$$\sum_{j=1}^{\mathcal{N}_t} \int_\Omega \varphi_t^i \varphi_t^j \Phi_t^j dV = \sum_{k=1}^{\mathcal{N}_s} \int_\Omega \varphi_t^i \varphi_s^k \Phi_s^k dV\,, \tag{2.17}$$

so that the final discretized system of equations is expressed as

$$A_{tt}\mathbf{\Phi}_t = A_{ts}\mathbf{\Phi}_s\,. \tag{2.18}$$

Particularly complex to compute is matrix A_{ts} which contains products $\varphi_t^i \varphi_s^k$, i.e. products between basis functions defined over different computational grids. Much easier are the cases where computational grids Ω_s and Ω_t coincide or at least one of them is a structured grid. In these cases Galerkin scheme can be used to obtain point-wise fields as a result of piece-wise fields projection. An example is given in Fig. 2.13 where a wall distance field, obtained as piece-wise field, has been projected with (2.16) on the same computational grid. Following this approach, a norm for $H^1(\Omega)$ space can be written as

$$||f(\mathbf{x})||_{H^1,\gamma} = \left(\int_\Omega \left[(f)^2 + \gamma (f')^2\right] d\mathbf{x}\right)^{1/2}\,, \tag{2.19}$$

with γ being a positive constant value. A Galerkin interpolation scheme can be constructed from the minimization of the square of $H^1(\Omega)$ norm $||\Phi_s - \Phi_t||_{H^1,\gamma}$, i.e.

$$
\begin{aligned}
(||\Phi_s - \Phi_t||)^2 &= \min_{\Phi \in \mathcal{V}_t} (||\Phi_s - \Phi||_{H^1,\gamma})^2 = \\
&= \min_{\Phi \in \mathcal{V}_t} \int_\Omega \left[(\Phi_s - \Phi)^2 + \gamma (\boldsymbol{\nabla} \Phi_s - \boldsymbol{\nabla} \Phi)^2 \right] d\mathbf{x} \qquad (2.20)
\end{aligned}
$$

In order for Φ_t to minimize the squared of norm $||\Phi_s - \Phi_t||_{H^1,\gamma}$, the derivatives of equation (2.20) with respect to all Φ_t^i coefficient must be equal to zero,

$$
\frac{\partial}{\partial \Phi_t^i} \int_\Omega \left[(\Phi_s - \Phi)^2 + \gamma (\boldsymbol{\nabla} \Phi_s - \boldsymbol{\nabla} \Phi)^2 \right] d\mathbf{x} = 0 \ \ \forall i \in \{1, ..., \mathcal{N}_t\}\,, \qquad (2.21)
$$

leading to the following system of equations

$$
\begin{aligned}
\sum_j \int_\Omega \gamma \boldsymbol{\nabla} \varphi_t^i \cdot \boldsymbol{\nabla} \varphi_t^j \Phi_t^j dV + \sum_j \int_\Omega \varphi_t^i \varphi_t^j \Phi_t^j dV &= \\
= \sum_k \int_\Omega \gamma \boldsymbol{\nabla} \varphi_t^i \cdot \boldsymbol{\nabla} \varphi_s^k \Phi_s^k dV &+ \sum_k \int_\Omega \varphi_t^i \varphi_s^k \Phi_s^k dV\,, \\
\forall i \in \{1, ..., \mathcal{N}_t\}\,, \quad \varphi_t^i &\in H^1(\Omega)\,. \qquad (2.22)
\end{aligned}
$$

While Galerkin interpolation (2.16) guarantees a conservation of source field integral value, the scheme (2.20) balances source field integral value conservation and differentiation properties through weight coefficient γ, more commonly seen as a diffusion coefficient in (2.22). For values of weight coefficient $\gamma \to 0$ interpolation scheme (2.16) is retrieved.

A reasoning in terms of L_2 norm can be applied also to point-wise $P2P2$ interpolation. After recalling the finite element representation of source and target fields Φ_s and Φ_t

$$
\Phi_s(\mathbf{x}) := \sum_{i=1}^{\mathcal{N}_s} \Phi_i^s \varphi_i(\mathbf{x})\,, \qquad (2.23)
$$

$$
\Phi_t(\mathbf{x}) := \sum_{j=1}^{\mathcal{N}_t} \Phi_j^t \psi_j(\mathbf{x})\,, \qquad (2.24)
$$

the following expression is obtained if interpolation scheme (2.8) is introduced in (2.24)

$$
\Phi_t(\mathbf{x}) = \sum_{j=1}^{\mathcal{N}_t} \psi_j(\mathbf{x}) \sum_{k=1}^{n_e^s} \Phi_k^{s,e} \varphi_k^e(\mathbf{x})\,. \qquad (2.25)
$$

In the above equation the source mesh element containing j-th target mesh node has been labeled with apex ^{e}. Source mesh element e contains n_e^s nodes upon which basis functions φ_k^e are defined. We consider again the L_2 norm of source-target fields difference

$$||\Phi_s - \Phi_t||_2 = \int_\Omega \left(\sum_{i=1}^{\mathcal{N}_s} \varphi_i(\mathbf{x})\Phi_i^s - \sum_{j=1}^{\mathcal{N}_t} \psi_j(\mathbf{x})\Phi_j^t \right)^2 dV\,, \tag{2.26}$$

In order for the L_2 norm to be minimum, the $\mathcal{N}_t$ derivatives of (2.26) with respect to coefficients Φ_j^t must be zero. As target coefficients Φ_j^t have been expressed as a linear combination source ones Φ_i^s, derivatives with respect to Φ_j^t reduces to derivatives with respect to Φ_i^s. L_2 norm (2.26) is rewritten as

$$||\Phi_s - \Phi_t||_2 = \int_\Omega \left(\sum_{i=1}^{\mathcal{N}_s} \varphi_i(\mathbf{x})\Phi_i^s - \sum_{j=1}^{\mathcal{N}_t} \psi_j(\mathbf{x}) \sum_{k=1}^{n_e^s} \Phi_k^{s,e}\varphi_k^e(\mathbf{x}) \right)^2 dV\,, \tag{2.27}$$

so, for the minimization it is obtained

$$\begin{aligned}
&\frac{\partial}{\partial \Phi_l^s} \int_\Omega \left(\sum_{i=1}^{\mathcal{N}_s} \varphi_i(\mathbf{x})\Phi_i^s - \sum_{j=1}^{\mathcal{N}_t} \psi_j(\mathbf{x}) \sum_{k=1}^{n_e^s} \Phi_k^{s,e}\varphi_k^e(\mathbf{x}) \right)^2 dV = \\
&=2 \int_\Omega \Bigg\{ \left(\sum_{i=1}^{\mathcal{N}_s} \varphi_i(\mathbf{x})\Phi_i^s - \sum_{j=1}^{\mathcal{N}_t} \psi_j(\mathbf{x}) \sum_{k=1}^{n_e^s} \Phi_k^{s,e}\varphi_k^e(\mathbf{x}) \right) \cdot \\
&\qquad \left[\varphi_l(\mathbf{x}) - \frac{\partial}{\partial \Phi_l^s} \left(\sum_{j=1}^{\mathcal{N}_t} \psi_j(\mathbf{x}) \sum_{k=1}^{n_e^s} \Phi_k^{s,e}\varphi_k^e(\mathbf{x}) \right) \right] \Bigg\} dV = 0\,.
\end{aligned} \tag{2.28}$$

The term within squared brackets becomes

$$\varphi_l(\mathbf{x}) - \frac{\partial}{\partial \Phi_l^s} \left(\sum_{j=1}^{\mathcal{N}_t} \psi_j(\mathbf{x}) \sum_{k=1}^{n_e^s} \Phi_k^{s,e}\varphi_k^e(\mathbf{x}) \right) = \varphi_l(\mathbf{x}) - \sum_{j=1}^{\mathcal{N}_t} \psi_j(\mathbf{x})\varphi_r^e(\mathbf{x})\,, \tag{2.29}$$

where in local node numbering, within source mesh element Ω_e^s, r-th node is associated to l-th node in global node numbering. Again (2.31) is simplified as

$$\varphi_l(\mathbf{x}) - \sum_{j=1}^{\mathcal{N}_t} \psi_j(\mathbf{x})\varphi_r^e(\mathbf{x}) = \varphi_l(\mathbf{x}) \left(1 - \sum_{j=1}^{\mathcal{N}_t} \psi_j(\mathbf{x}) \right)\,, \tag{2.30}$$

obtaining at last this final form of (2.28)

$$\begin{aligned}
&\int_\Omega \Bigg\{ \left(\sum_{i=1}^{\mathcal{N}_s} \varphi_i(\mathbf{x})\Phi_i^s - \sum_{j=1}^{\mathcal{N}_t} \psi_j(\mathbf{x}) \sum_{k=1}^{n_e^s} \Phi_k^{s,e}\varphi_k^e(\mathbf{x}) \right) \cdot \\
&\qquad \varphi_i(\mathbf{x}) \left(1 - \sum_{j=1}^{\mathcal{N}_t} \psi_j(\mathbf{x}) \right) \Bigg\} dV = 0 \quad \forall i \in \{1, ..., \mathcal{N}_s\}\,.
\end{aligned} \tag{2.31}$$

Minimization of L_2 norm is then automatically verified as, from basis function construction, on each point of the domain the sum

$$\sum_{j=1}^{\mathcal{N}_t} \psi_j(\mathbf{x}) \tag{2.32}$$

is equal to 1. Point-wise algorithm described in Section 2.3.1 allows to minimize the L_2 norm of field difference between source and target fields. With Galerkin projection method a system of equation is solved to compute the set of coefficients that allows to minimize the L_2 norm. The key aspects of the two different methods are seen below

$$\text{Galerkin:} \qquad \underbrace{\int_\Omega \psi_t^i \left(\Phi_t - \Phi_s\right) dV = 0\,,}_{conservation} \tag{2.33}$$

$$\text{Point-wise:} \qquad \underbrace{\left(1 - \sum_{j=1}^{\mathcal{N}_t} \psi_j\right)}_{basis\ function\ contruction} \cdot \int_\Omega \varphi_s^i \left(\Phi_t - \Phi_s\right) dV = 0\,. \tag{2.34}$$

As reported above, with Galerkin projection the minimization of L_2 norm of field difference is obtained with a field conservation constraint. The terms within round brackets in (2.34) has been taken out from the integral as the sum over basis functions ψ_j is independent from spatial coordinates $(\mathbf{x})$. This highlights the fact that minimization of L_2 norm is achieved as a consequence of basis function construction, independently form integral value, so field conservation is not guaranteed. Point-wise algorithm should be used when projection is performed from a coarser to a finer grid, while Galerkin method is an useful tool for projection in the opposite direction. Moreover, point-wise projection $P2P2$ algorithm can be implemented as a function based on MED data format, while Galerkin method require an implementation within the code in order to solve the system of equations.

Numerical implementation of P2P2 algorithm

The implemented algorithm can be used for point-wise projection on boundary surfaces or volume domains, and for 1D, 2D and 3D geometries. In particular both triangle and rectangle based elements are considered. As mentioned above, the fundamental steps of the algorithm are

i) Point location

ii) Position reconstruction

iii) Interpolation weights calculation

```
SourceMesh.getCellsContainingPoints (
                    NodesCoordinates,
                    NumOfNodes,
                    Toll,
                    PosCells,
                    PosCellsIndex
                 );
for(int i=0; i<NumOfNodes; i++){
   int NumPosCells = PosCellsIndex->getIJ( i+1,0 ) - PosCellsIndex->getIJ( i,0 );
   for(int j=0; j<NumPosCells; j++){
      int CellId = PosCells->getIJ( PosCellsIndex->getIJ( i,0 ) + j,0 );
   }
}
```

Listing 2.1: MEDCoupling function to get list of possible source mesh cells containing target mesh nodes

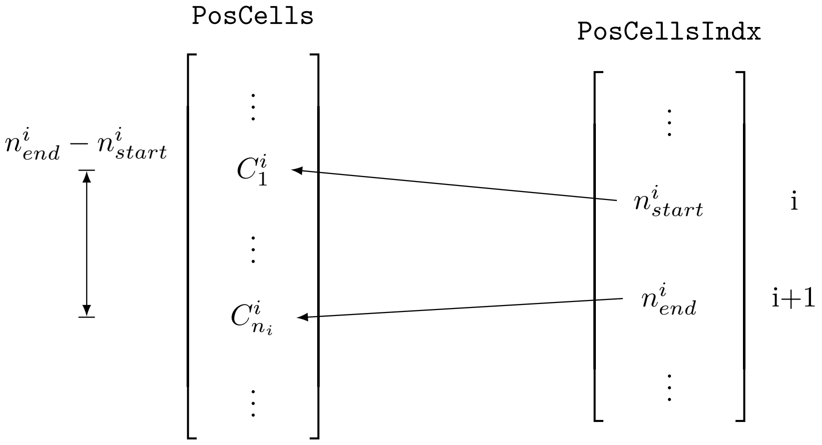


Figure 2.14: Reading ids of possible source mesh cells containing i-th target mesh node from PosCells and PosCellsIndex arrays of Listings 2.1.

Step i) is performed using MEDCoupling function getCellsContainingPoints as reported in Listings 2.1. SourceMesh is a MED mesh object and the functions takes as input: NodesCoordinates, an array containing the coordinates of the nodes whose position is to be searched, NumOfNodes, the total amount of nodes, so that the dimension of NodesCoordinates is equal to NumOfNodes multiplied by space dimension, Toll, a tolerance factor used to determine cell belonging and two arrays, namely PosCells and PosCellsIndex. The fist array contains all indices of possible source mesh cells containing the NumOfNodes target mesh nodes. For the general i-th node the total amount of possible cells is read from PosCellsIndex, in particular by accessing the values stored at i-th and $i + 1$-th position, as represented in Fig. 2.14. The total amount is the calculated as PosCellsIndex[i+1] - PosCellsIndex[i] and the indices are read from PosCellsIndex[i+1]-th and PosCellsIndex[i]-th positions of ar-

```
DataManipulator *Interpolator = new DataManipulator(
     SourceMesh,   // Source mesh in med format
     TargetMesh,   // Target mesh in med format
     DomainType    // Flag for volume or boundary interpolation
     );
P.createInterface(
      InterId,     // int value for interface indentification
      Group        // mesh group upon which interface is built
     );
MEDCouplingFieldDouble *SourceField = P.getValues(
      InterId,     //
      SystemName,  // name of equation system where solution is extracted
      nComp,       // number of components to extract
      FirstComp    // first component to extract
     );
MEDCouplingFieldDouble *TargetField = Interpolator->InterpolateField(SourceField);
```

Listing 2.2: Data Manipulator function for P2P2 interpolation

```
 MEDCouplingFieldDouble *TargetField = Interpolator->InterpolateField(
    SourceField,
    val
   );
```

Listing 2.3: Data Manipulator function for P2P2 interpolation with default value for non-overlapped target nodes

ray `PosCells`. The right cell is found by solving the reverse mapping algorithm described in Section 2.3.1. A set of coordinates ξ is determined: if the cell is the right one then ξ coordinates satisfy the constrain of admissible coordinate values in the canonical element, i.e.

$$\text{Rectangle based} \qquad \xi_i \in [-1-\delta, 1+\delta]\,, \forall i = 1, ..., dim\,, \tag{2.35}$$

$$\text{Triangle based} \qquad \xi_i \in [0, 1+\delta]\,, \forall i = 1, ..., dim\,, \tag{2.36}$$

where δ is a small tolerance factor used in the algorithm. Point reconstruction step is then used to determine the right source mesh cell containing the target node and to calculate the coordinates representing the target mesh node in the canonical element transformation of the source mesh cell. An example of the use of interpolation function is given in Listings 2.2. When DataManipulator class

```
 MEDCouplingFieldDouble *TargetField = Interpolator->InterpolateField(
    SourceField,
    TargetFieldB
   );
```

Listing 2.4: Data Manipulator function for P2P2 interpolation. For non-overlapped target nodes the values is set from another field.

```
DataManipulator Integrator;
P.createInterface(
      InterId ,       // int value for interface indentification
      Group           // mesh group upon which interface is built
     );
MEDCouplingFieldDouble *SourceField = P.getValues(
      InterId ,       //
      SystemName,     // name of equation system where solution is extracted
      nComp,          // number of components to extract
      FirstComp       // first component to extract
     );
double IntegralValue = Integrator.Integrate(SourceField , Method);
```

Listing 2.5: Data Manipulator function for calculation of solution integrals with Gauss method

is used to perform P2P2 interpolation then source and target meshes are given in order to build the projection operator $\boldsymbol{P}$. Meshes can then be updated if they change with time with a proper function. The function reported in Listings 2.2 is used when source and target mesh completely overlap. In the opposite case the function can be called with additional parameters, namely as defined in Listings 2.3 and 2.4. In the first case a default value val is set on those target mesh nodes that belong to no source mesh cell. In the latter case, for the same nodes, the written value is taken from field `TargetFieldB` that is thus defined on target mesh.

2.3.3 Gauss integration of P2 fields

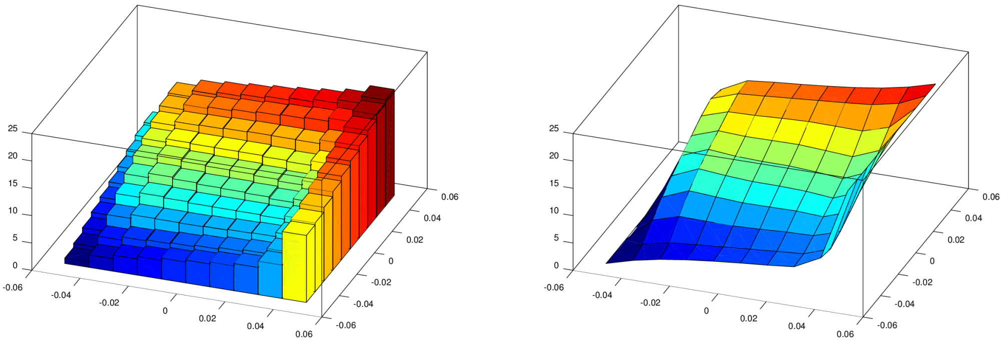
When a numerical coupling is realized between a multidimensional code and a system code, then the different dimensions of the equations is a problem to be solved, since the system code is based on a single coordinate along the axial position of the system. With defective coupling technique, source terms are calculated from multidimensional code solution to be used within system code equations in order to improve the code results [15, 16, 21, 23]. Source terms are usually calculated from mean integral values of the multidimensional code solution. When the solution is extracted in MED data format through a proper interface then the integral value can be easily calculated by following the operations described in the brief example reported in Listings 2.5. With the `Method` flag different types of integrals can be computed: integral value, mean integral value, L_2 norm, bulk value (if an additional field containing velocity field is given in input) but also the volume of the geometry defined by the grid of `SourceField`.

2.3.4 Generation of P0 fields

This routine has been developed to obtain piece-wise fields from point-wise fields. In Listings 2.6 an example is shown for the function usage, i.e. `GetCellField` from

```
DataManipulator Integrator;
P.createInterface(
      InterId,       // int value for interface indentification
      Group          // mesh group upon which interface is built
     );
MEDCouplingFieldDouble *SourceField = P.getValues(
      InterId,       //
      SystemName,    // name of equation system where solution is extracted
      nComp,         // number of components to extract
      FirstComp      // first component to extract
     );
MEDCouplingFieldDouble *P0Field = Integrator.GetCellField(SourceField);
```

Listing 2.6: Data Manipulator function for generating P0 fields from point-wise fields.



a)

b)

Figure 2.15: Generation of piece-wise field a) starting from a node-wise source field b).

class `Data Manipulator`. A field $\tilde{\Phi}$ is obtained starting from a source one Φ. The underlying mesh of the two fields is the same. For each cell Ω_j of the computational grid the piece-wise value $\tilde{\Phi}_j$ is calculated as mean integral value of Φ, so

$$\tilde{\Phi}_j = \frac{1}{\int_{\Omega_j} dx} \int_{\Omega_j} \Phi dx \,. \tag{2.37}$$

A graphical example of the obtained field is shown in Fig. 2.15. The routine can be used when coupling a finite element code with a finite volume code.

2.4 Natural convection in a squared cavity

In the present section the natural convection in a squared cavity is studied by means of a numerical code coupling between Finite Element (FEM) code FEMuS

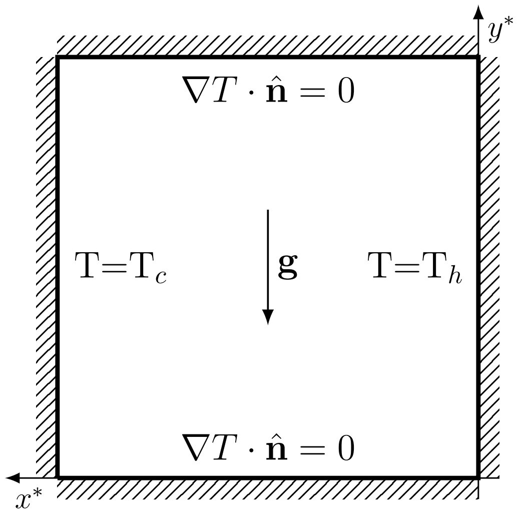

y^*
$\nabla T \cdot \hat{\mathbf{n}} = 0$
T=T$_c$
$\mathbf{g}$
T=T$_h$
$\nabla T \cdot \hat{\mathbf{n}} = 0$
x^*

Figure 2.16: Sketch of the simulated domain for the natural convection case in squared cavity.

and Finite Volume (FVM) code OpenFOAM. Since the simulated case is a simple and well studied case [24, 25], it allows to better understand how to realize the code coupling between FEMuS and OpenFOAM, in particular in terms of data manipulation. As will be later discussed, many strategies can be used to convert a quadratic node-wise solution from FEMuS grid to OpenFOAM one and vice-versa. Attention is paid to grid resolutions and to the effects of the coupling procedures on final solutions. For the natural convection case the coupling between the two codes is realized on the whole simulated domain. Laminar Navier Stokes system of equation is considered, with Boussinesq approximation for the buoyant term, namely

$$\boldsymbol{\nabla} \cdot \mathbf{u} = \mathbf{0}\,, \tag{2.38}$$

$$\begin{aligned}\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla}\mathbf{u}\right) &= -\boldsymbol{\nabla}P + \\ &+ \boldsymbol{\nabla} \cdot \left[\mu\left(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^{\mathbf{T}}\right)\right] + \underbrace{\rho\mathbf{g}\beta(\tilde{T} - T_{ref})}_{coupling\, term}\,,\end{aligned} \tag{2.39}$$

$$\rho C_p\left(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla}T\right) = k\nabla^2 T\,. \tag{2.40}$$

The system of equations (2.38–2.40) is solved with both FEMuS and OpenFOAM codes. As reported in (2.39), the temperature field used to calculate the buoyancy force acts as coupling term between the two different codes. Three different cases are considered, namely

Grid resolution	20×20	40×40	80×80
Color line	🟩	🟦	🟥

Table 2.1: Grid resolutions and relative line colors for natural convection results.

a) Uncoupled case

b) FEMuS temperature field in OpenFOAM buoyancy force term

c) OpenFOAM temperature field in FEMuS buoyancy force term

The simulated domain is sketched in Fig. 2.16. On wall surface located on the right a uniform temperature value T_h is set, while on left wall surface a temperature value T_c is imposed, with $T_c < T_h$. The remaining walls are considered adiabatic and no-slip boundary condition is applied on each side. Temperature difference is set in order to obtain a value of Rayleigh number $Ra = 1 \times 10^5$, with $Ra = g\beta L^3(T_h - T_c)/(\nu\alpha)$ and β being the thermal expansion coefficient, L the length of the square side, g the modulus of gravity vector ν and α kinematic viscosity and thermal diffusivity. Space coordinates x and y are made non-dimensional using side length L, $x^* = x/L$ and $y^* = y/L$, $x^*, y^* \in [0, 1]$ with $x^* = 0$ being the position of the hotter wall. Velocity values are made non-dimensional using L and α, so $u^* = uL/\alpha$, $v^* = vL/\alpha$, while for temperature non-dimensional field T^+ is calculated as $T^+ = (T - T_c)/(T_h - T_c)$. Reference results are taken from [24]. Attention is focused on velocity picks of vertical velocity component v^* along plane at $y^* = 0.5$.

2.4.1 Numerical procedure

The coupling procedure used for this study case has been previously described. We recall that OpenFOAM requires three-dimensional meshes even for simulating two dimensional problems. In order to deal with field interpolation and manipulation an additional mesh is considered, i.e. a two-dimensional mesh with same discretization three-dimensional OpenFOAM one along $x - y$ plane. This additional mesh is not required if the coupling is performed for simulating three-dimensional problems. Mesh generation for both OpenFOAM and FEMuS codes can be handled with SALOME platform: in the first case the generated mesh is exported only in .med format while in the latter it is exported in .unv and .med data format. MED meshes are used to create coupling interfaces, while .unv mesh is converted into OpenFOAM mesh format after execution of command

```
ideasUnvToFoam <mesh.unv>
```

Several meshes are considered, as summarized in Table 2.1, where grid resolutions and relative line colors, for the solutions that will be shown, are reported.

	v^*_{max} on A - B	
Grid size	FEMuS	OpenFOAM
20 × 20	73.51	67.99
40 × 40	73.48	73.93
80 × 80	73.48	73.98
160 ×160	-	73.81

Table 2.2: Maximum values of non-dimensional velocity v^*, along plane $y^* = 0.5$, for the uncoupled case.

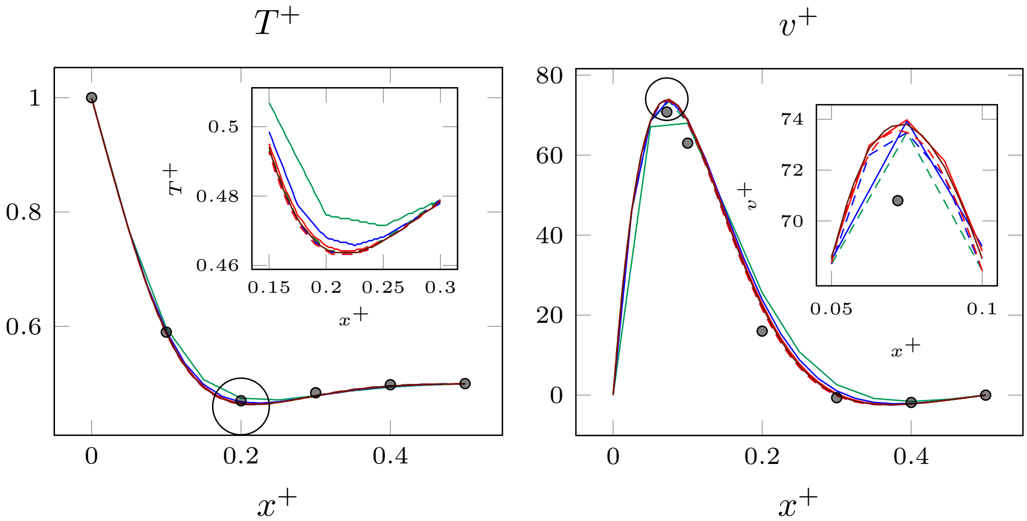


Figure 2.17: Uncoupled case: non-dimensional temperature T^+ and velocity v^+ profiles taken on plane $y^* = 0.5$. Solid lines are for OpenFOAM solutions while dashed ones are for FEMuS solutions. Dots represent reference values from [24].

Coupling case a)

Reference value of maximum non-dimensional velocity component v^* along y^* direction, on plane $y^* = 0.5$, is $v^* = 70.8$ [24]. For the uncoupled case, maximum values of v^*, v^*_{max}, are reported in Table 2.2. It can be seen that, with exception of coarsest grid case, OpenFOAM results predict higher values of v^*_{max} than FEMuS ones, and with both codes higher values than reference one are obtained. Non-dimensional temperature and velocity values along plane $y^* = 0.5$, from wall position to $x^* = 0.5$, are plotted in Fig. 2.17. Different effect of grid size can be seen from this Figure. Results obtained with FEMuS code with the coarsest are quite close to those obtained with finest one, as quadratic elements have been used. For OpenFOAM results the solutions of grid cases 80×80 and

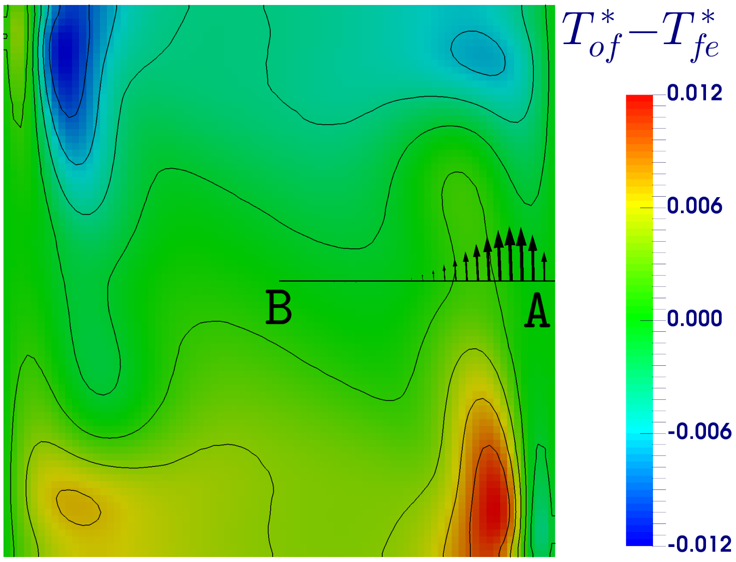


Figure 2.18: Coupling case a): non-dimensional temperature difference between OpenFOAM and FEMuS solutions with grid 80×80.

	v^*_{max}	
Grid size	FEMuS	OpenFOAM 80×80
20×20	73.51	73,241
40×40	73.48	73,189
80×80	73.48	73,168

Table 2.3: Maximum values of non-dimensional velocity v^*, along plane $y^* = 0.5$, for coupled case b2, OpenFOAM grid 80×80

160×160 are very close. Temperature profiles are in good agreement with reference values, while for velocity a general overestimation is observed in the range $x^{\in}[0.1, 0.3]$. What it is interesting to notice is the difference between OpenFOAM and FEMuS non-dimensional temperatures, for case 80×80, shown in Fig. 2.18. OpenFOAM predicts a higher temperature value in the region close to hot wall and a smaller temperature value in the region close to the cold wall, with respect of FEMuS results. This difference in the temperature fields enhances flow circulation in OpenFOAM results, explaining the higher obtained values of v^*_{max}.

Coupling case b)

Coupling case b is a one way coupling between FEMuS and OpenFOAM, with the temperature field of the first code being used for the calculation of OpenFOAM

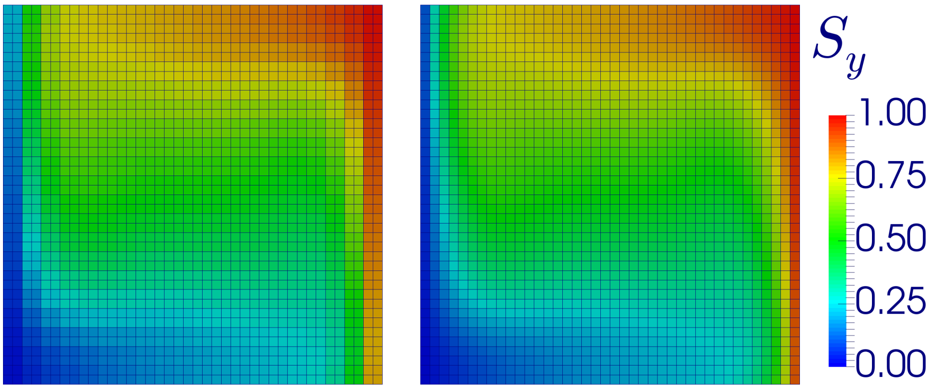


Figure 2.19: Coupling case b): buoyancy source term S_y obtained for coupling cases b1), on the left, and b2), on the right.

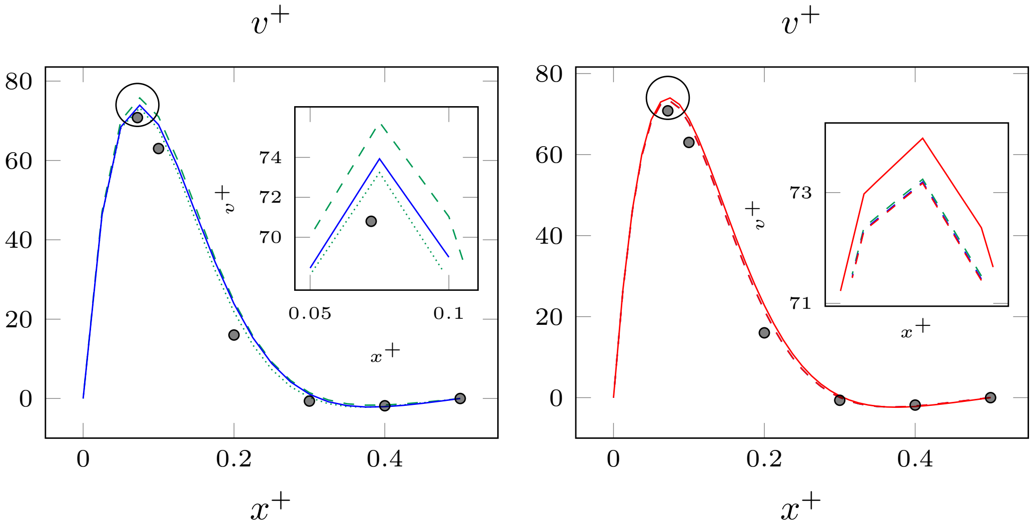


Figure 2.20: Non-dimensional velocity v^+ profiles taken on plane $y^* = 0.5$. On the left OpenFOAM results with grid 40×40 and coupling procedures b1 (dashed line) and b2 (dotted line) with FEMuS grid 20×20 . On the right results for OpenFOAM 80×80 and coupling procedure b2, with different FEMuS meshes, following color from Table 2.1. In both plots solid line refers to uncoupled case.

buoyancy force and with no feedback. From the observation of the results obtained for the uncoupled case, it is expected that OpenFOAM velocity field should have smaller maximum values v^*_{max} along $y^+ = 0.5$ plane. Two different strategies are considered. As already mentioned, FEMuS solution is obtained using quadratic finite elements, thus FEMuS grid is generally coarser than OpenFOAM one. Buoyancy force $\mathbf{S}$ can then be calculated on OpenFOAM grid with the following methods for temperature field projection

b1) A piece-wise field of T is generated on FEMuS grid and then interpolated with MED conservative method on OpenFOAM grid. This interpolation technique, for P0 fields, is formulated as follows

$$\Phi_j^t = \sum_{i=1}^{n_s} \omega_{j,i} \Phi_i^s \,. \tag{2.41}$$

For each target mesh cell the field value is calculated as a weighted sum of source field values, where interpolation weight $\omega_{j,i}$ is the volume fraction of j-th target mesh cell overlapped by i-th source mesh cell.

b2) Point-wise interpolation of source temperature field from FEMuS grid to OpenFOAM one, and then calculation of piece-wise temperature field on OpenFOAM support.

An example of resulting buoyancy force term $S_y = \mathbf{S} \cdot \hat{\boldsymbol{j}}$ is reported in Fig. 2.19, in particular on the left for method b1) and on the right for method b2), for coupling case with FEMuS grid 20×20 and OpenFOAM 40×40. With the first method, temperature field projected on OpenFOAM grid suffers from source grid resolution as target mesh cells belonging to the same source mesh cell share the same piece-wise temperature value. In the second case a piece-wise temperature field with target mesh resolution is obtained. The effect of the two methods on computed velocity field is displayed on the left of Fig. 2.20. Solid line stands for OpenFOAM solution of uncoupled case, while dashed line is for coupling method b1) and dotted line for method b2). On the contrary of what should be expected, a higher value of v^*_{max} is obtained for coupling method b1), while b2) obtained result behaves as expected. The obtained values are $v^{*,b1}_{max} = 75.76$ and $v^{*,b2}_{max} = 73.24$. On the right of Fig. 2.20 non-dimensional velocity values are reported for OpenFOAM grid 80×80 and FEMuS grids 20×20, 40×40 and 80×80. with method b2. Values of v^*_{max} are reported in Table 2.3. The results are all in agreement with the expected behavior and slight differences are obtained by increasing FEMuS grid resolution.

Coupling case c)

Coupling case c operates in the opposite direction than coupling case b. Temperature field calculated with OpenFOAM is given to FEMuS for the buoyancy force term calculation. Also in this case two different strategies are considered:

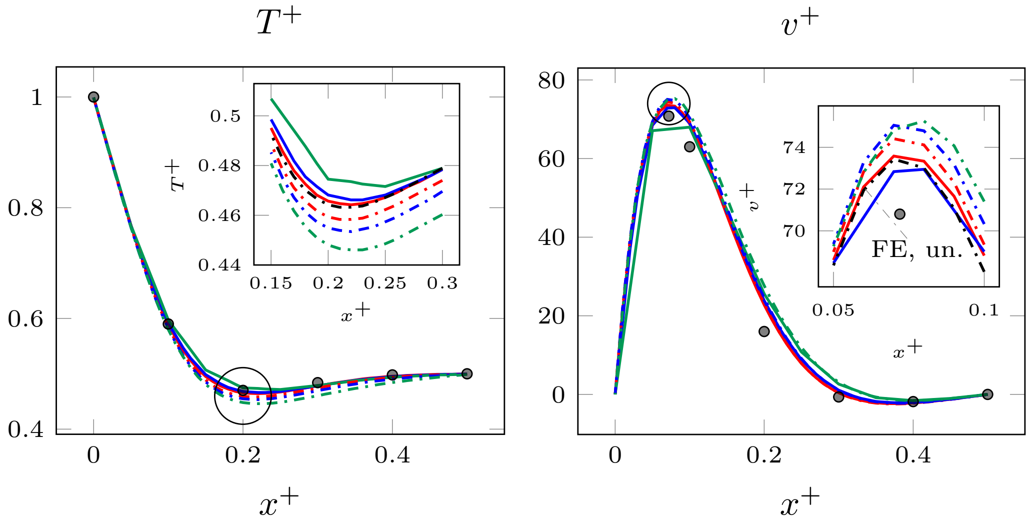


Figure 2.21: Coupling case c1: non-dimensional temperature T^+ and velocity v^+ profiles taken on plane $y^* = 0.5$. Solid lines stand for OpenFOAM solutions with different meshes while dash-dotted lines stand for FEMuS solution with grid 80×80 and OpenFOAM source term.

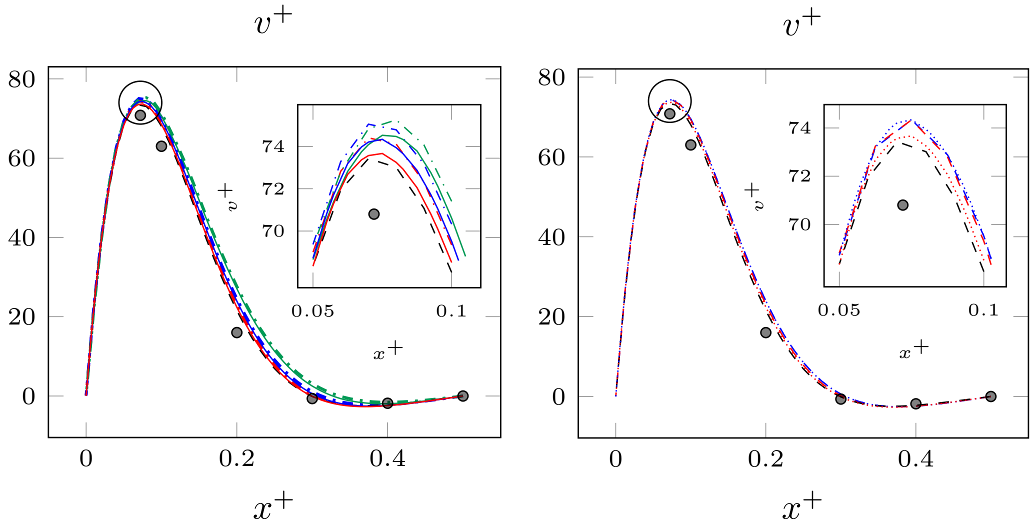


Figure 2.22: Coupling case c2: on the left comparison of v^* values obtained with c1 (dash-dotted lines) and c2 (solid lines) methods for FEMuS grid 80×80 and various OpenFOAM meshes. On the right comparison of c2 results for FEMuS grid 40×40 (dashed lines) and 80×80 (dash-dotted lines) and OpenFOAM grids 40×40 and 80×80.

	Case c2		Case c1
OpenFOAM Grid size	FEMuS 40×40	FEMuS 80×80	FEMuS 80×80
20×20	-	1.49058	1.50595
40×40	1.48677	1.48677	1.50266
80×80	1.48666	1.47338	1.48943

Table 2.4: Maximum values of non-dimensional velocity v^*, along plane $y^* = 0.5$, for coupled case b2, OpenFOAM grid 80×80

c1) Piece-wise temperature field of OpenFOAM solution is projected on FEMuS grid by using conservative MED mapping method. The buoyancy force is then calculated as a piece-wise field.

c2) As for case c1), temperature field on FEMuS grid is obtained through MED conservative mapping method. A node-wise temperature field is then calculated with Galerkin interpolation. The overall process is then conservative as both interpolation techniques are conservative.

From this coupling it is expected to obtain higher values of FEMuS velocity field along $y^* = 0.5$. This behavior is indeed observed, as shown on the right of Fig. 2.21, where results for FEMuS grid 80×80 and OpenFOAM grids 20×20, 40×40 and 80×80 are reported for coupling case c1. As OpenFOAM grid resolution increases the temperature field becomes more accurate, as seen on the left of Fig. 2.21, and FEMuS velocity field converges towards OpenFOAM solution. A comparison between c1 and c2 methods is shown on the left of Fig. 2.22 where non-dimensional velocity v^* profiles along $y^* = 0.5$ plane are plotted for the case of FEMuS grid 80×80 and OpenFOAM grids 20×20, 40×40 and 80×80. In both cases higher values are obtained with respect to uncoupled case. As OpenFOAM grid resolution increases the velocity values decrease and for a given OpenFOAM grid with c2 method slightly smaller velocity values are predicted. Galerkin interpolation of OpenFOAM piece-wise temperature field allows to obtain an increased resolution in the calculation of buoyancy force. On the right of Fig. 2.22 a comparison of c2 results is reported, for FEMuS meshes 40×40 and 80×80 and OpenFOAM grid resolutions 40×40 and 80×80. The change of OpenFOAM mesh resolution does not lead to a sensible change of v^*_{max} values for mesh FEMuS mesh grid 40×40, while a more sensible difference is found in the range $x^* \in [0.1, 0.3]$. For FEMuS grid 80×80 the increase of OpenFOAM mesh resolution from 40×40 to 80×80 leads to a general decrease of v^* values, v^*_{max} included. Finally all the values obtained of v^*_{max} for the coupling case c are reported in Table 2.4.

From the analysis of the obtained results for this simple case study some conclusions can be drawn on the optimal use of data manipulation for numerical code coupling of codes FEMuS and OpenFOAM. When using quadratic elements, an OpenFOAM mesh should be used with an additional level of refinement with respect to FEMuS grid, that means four times the number of elements for a two-dimensional case. When a FEMuS solution is given from FEMuS to OpenFOAM then it is better to perform a point-wise $P2P2$ interpolation on OpenFOAM support and then calculate a piece-wise field with routine presented in Section 2.3.1. For the data transfer in the opposite direction, the best results have been obtained by performing first the MED conservative $P0P0$ interpolation from OpenFOAM grid to FEMuS one and then a Galerkin interpolation in order to obtain a $P2$ field out of the piece-wise one.

Chapter 3

Immersed Boundary method

Over the last decades numerical simulation has become an increasingly widespread tool to investigate complex engineering applications. Numerous computational fluid dynamics (CFD) problems involve fluid flow around arbitrarily shaped objects, that can be either static or move with time. Such systems are characterized by internal boundaries between fluid and solid regions with dynamically changing geometry. In this regard, one of the challenges of devising numerical techniques is to track the moving boundary and simulate its evolving interaction with the system. Even though considerable efforts have been made, simulating moving boundaries is still not a trivial problem. Methods for handling such problems can broadly be classified into Lagrangian, Eulerian and combined formulations according to their computational grid representation. Lagrangian methods feature a moving computational grid that is dynamically adapted to the interface. Boundary conditions can be applied at the exact location of the interface since its evolution is explicitly tracked. The major drawback to this approach is the continuous grid rearrangement over the changing computational domain. Conversely, Eulerian methods do not track the interface explicitly but reconstruct it by means of appropriate field variables. This allows a fixed grid formulation where the interface is not treated as a sharp discontinuity, which implies smearing of boundary information. In Eulerian methods topological changes are easier to handle since they do not require mesh rearrangement work and thus they are particularly suited for free surfaces applications.

Recent years have seen a significant interest in solution algorithms for fluid flow around objects based on non-conforming grids, namely meshes that are not configured to conform to the fluid-solid interface. Several numerical models referred to as immersed boundary methods (IB) have received attention due to their remarkable advantages in simulating fluid flow around moving objects with complex boundaries. The IB method was originally introduced by Peskin as a mean to investigate cardiovascular flow patterns [26]. Although originally intended for biological fluid dynamics, the IB method has been successfully applied to a wide range of CFD problems [27, 28, 29]. Such methods share as a common core feature the ability to simulate the presence of boundaries without fitting the computational grid to the immersed body. The grid generation process is not altered and spacial discretization is performed over the whole computational domain, resulting in mesh elements being cut by the fluid-solid interface. IB methods constitute a promising alternative to conventional body-conformal grid approaches. The process of generating body-fitted grids requires extensive meshing work that may become prohibitively expensive for complex geometries. In order to handle moving boundaries problems, the grid needs to be regenerated at each time step, adding up to the computational costs. By contrast, IB methods allows simplified spatial discretization that results in more uniform computational grids obviating the need for mesh rearrangement. Because tackling complex moving boundaries is easier, non-conforming grid approaches experience less deterioration in grid quality which affects accuracy and convergence properties of the numerical scheme. Another common point of concern underlying IB methods is the imposition of boundary conditions on the fluid-solid interface. Conventional body-conformal meshes allow boundary conditions to be specified directly on the interface grid points with relative ease. Implementation of boundary condition is not as straightforward as in IB methods, which in turn may negatively impact the robustness and conservation properties of the solution algorithm. IB methods account for the fluid-solid interface and its effects through several different strategies, which are extensively reviewed in [30]. The presence of the immersed body is mimicked by means of additional source terms and modified boundary conditions which may be imposed both by indirect and direct techniques.

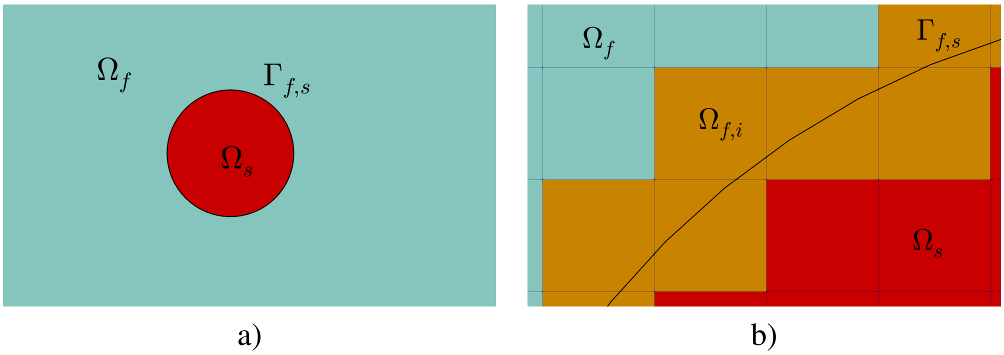
In the present work, a finite element based IB method intended for simulation of fluid flow around moving objects is discussed. Numerical solution of fluid flow is computed on a fixed grid which spans over the entire domain, including both fluid and solid regions, whereas moving boundaries are tracked by the motion of an additional grid defined on the solid body. The solid mesh accounts for the movement of the object without altering the fixed computational grid and consequently simplifying the discretization process. Coupling between fixed and moving mesh is tackled by appropriate projections of fields and boundary conditions. Thus, effects of moving objects on fluid flow may be imposed through indirect means by introducing an adequate projection algorithm. The proposed IB method succeeds in decoupling the equations for the fluid domain from those imposed on the moving solid and consequently no numerical solution is required for governing equations inside the immersed body.

An approach similar to Volume Of Fluid method for multiphase flows is here used to distinguish computational regions between fluid and solid domains [31]. A computational grid is used to model the solid body and a unitary field is defined upon it. An indicator function on the computational grid where equations are solved is then obtained as a result of node-wise P2P2 interpolation. The resulting function allows to represent the immersed body and its location in the fluid domain. Differently from the case of multiphase problems, where the indicator functions is initialized at the beginning of the simulation and then advected accordingly to solved velocity field, the indicator function is obtained with a projection operator and needs to be computed once or at every time-step depending on solid body law of motion. Additionally, a cell-wise volume fraction field is built to evaluate each cell volume fraction occupied by solid body. Galerkin interpolation scheme is used to retrieve a point-wise differentiable target field. This information allows to explicitly track the fluid-solid interface through linear reconstruction in arbitrarily shaped elements. The proposed method represent an efficient approach to interface reconstruction since it enables the computation of interface variables without being prohibitively complex and time consuming. Numerical solution for fluid flow around a moving object is then accurately computed on the fixed grid. At each time iteration the solid computational grid position is changed using an arbitrary law, involving a fixed movement or an interaction with fluid through stress calculation on fluid-solid interface. Consequently indicator function is interpolated and the volume fraction field updated on the target mesh (fluid computational grid). Numerical solution accuracy strictly relates to the robustness of the projection algorithm, thus it is worthwhile to investigate the interpolation process. We assess the performance of the proposed IB approach with two significant test cases with the intent of validating the numerical scheme.

As briefly described above, the proposed IB method heavily relies on interpolation techniques discussed in Chapter 2. The main advantage of this immersed boundary approach is that its formulation allows to handle both static and moving objects, without the need of computing variables such as a signed wall distance between fluid computational grid nodes and the immersed body surface. This quantity is computed to distinguish computational domain into fluid and solid regions and its computation can be a cumbersome task is complex shapes are involved [27, 32, 33].

3.1 Numerical Modeling

Let us consider a moving rigid body $\Omega_s(t)$ completely immersed in a fluid domain $\Omega_f(t)$ and separated by a time dependent interface $\Gamma_{f,s}(t) = \partial\Omega_f(t) \cap \partial\Omega_s(t)$, as sketched in Figure 3.1 a). Any topological changes in the solid region affect in turn the fluid domain $\Omega_f(t)$ in such a way that the entire computational domain $\Omega = \Omega_f(t) \cup \Omega_s(t)$ does not vary over time. The governing equations considered here are the unsteady Navier-Stokes equation for incompressible laminar flows,



a) b)

Figure 3.1: Sketch representation a) of a computational domain made of a fluid (Ω_f) and solid (Ω_s) sub domains which interact through a fluid-solid interface ($\Gamma_{f,s}$) and close-up view b) of computational grid cut by the fluid-solid interface.

namely

$$\nabla \cdot \mathbf{u} = 0\,, \tag{3.1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla P + \nu(\nabla^2 \mathbf{u}) + \mathbf{g}\,. \tag{3.2}$$

where $\mathbf{u}$ is the velocity vector, ρ the density, P the pressure, ν the kinematic viscosity and $\mathbf{g}$ the gravity acceleration vector.

3.1.1 Immersed boundary formulation

In the present formulation of Immersed Boundary algorithm with $\Omega_s(t)$ and $\Omega_f(t)$ the solid and fluid physical regions are respectively labeled, as sketched in Fig. 3.1 a). The union of these two regions will be denoted as Ω. It is remarked that Ω is not a function of time and that this region will be used as computational domain. When dealing with immersed boundary algorithms it is important to define a method for identifying fluid and solid regions. In literature many methods are proposed, involving, for example, level set methods based on wall distance between a mesh node and the fluid-solid interface [27, 28]. In the present work an indicator function $\chi(\boldsymbol{x}, t)$, defined as a multidimensional Heaviside function, is adopted. The field can be defined as

$$\chi(\boldsymbol{x}, t) = \int_{\Omega_s(t)} \delta(\boldsymbol{x}' - \boldsymbol{x})\, d\boldsymbol{x}' \quad \forall \boldsymbol{x} \in \Omega\,, \tag{3.3}$$

where δ is the Dirac delta function. The possible values of $\chi(\boldsymbol{x}, t)$ over Ω are then

$$\chi(\boldsymbol{x}, t) = \begin{cases} 1\,, & \forall \mathbf{x} \in \Omega_s(t)\,. \\ 0\,, & \forall \mathbf{x} \in \Omega_f(t)\,. \end{cases} \tag{3.4}$$

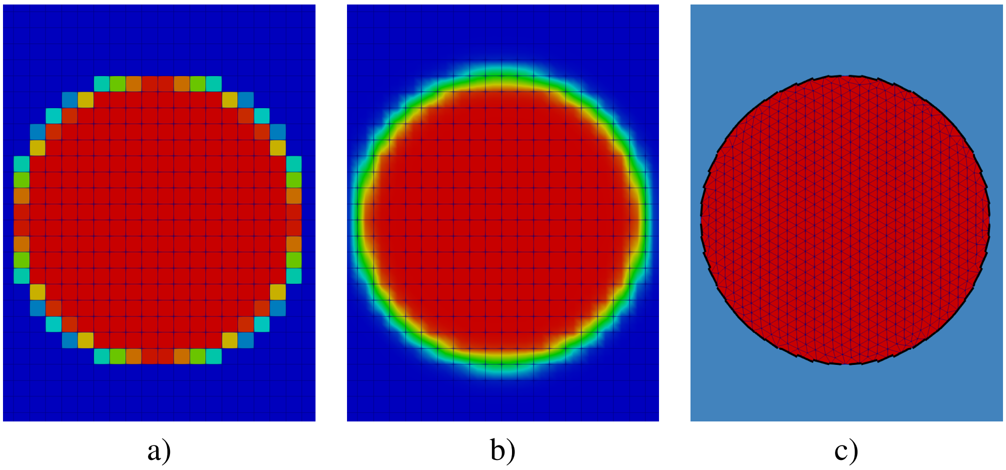


Figure 3.2: Example of discretized indicator function of volume fraction α a), interpolated volume fraction α' b) and fluid-solid interface reconstruction together with a mesh representation of solid body c).

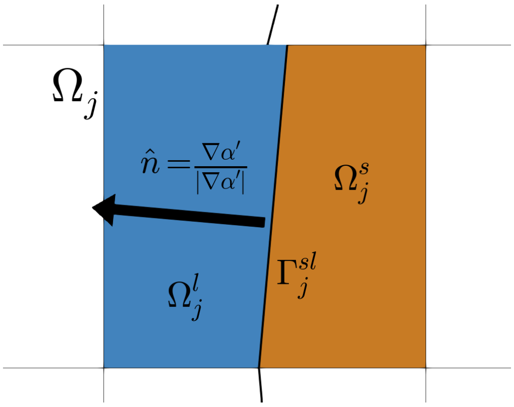


Figure 3.3: Close-up view of interface reconstruction on a generic mesh element.

A discretized indicator function $\chi^h(\mathbf{x}, t)$ is computed using the projection method described in Section 2.3.1. In particular a source field ψ_s is defined on the solid body mesh, with a uniform value $\psi_s(\mathbf{x}) = 1 \,\forall\, \mathbf{x} \in \Omega_s$. The field $\chi^h(\mathbf{x}, t)$ is then calculated as a projection of ψ_s over the computational domain, meaning $\chi^h = |P|\psi_s$. Alongside the continuous indicator function χ^h a cell-wise volume fraction field α is defined so that, for each computational cell, α indicates the fraction of volume occupied by the solid body. A continuous field, denoted as α' is finally introduced and calculated with a Galerkin projection of volume fraction α, namely

$$\int_\Omega (\alpha'\varphi + \gamma\boldsymbol{\nabla}\alpha' \cdot \boldsymbol{\nabla}\varphi)\, d\Omega = \int_\Omega \alpha\varphi d\Omega \quad \forall\varphi \in H^1(\Omega)\,, \tag{3.5}$$

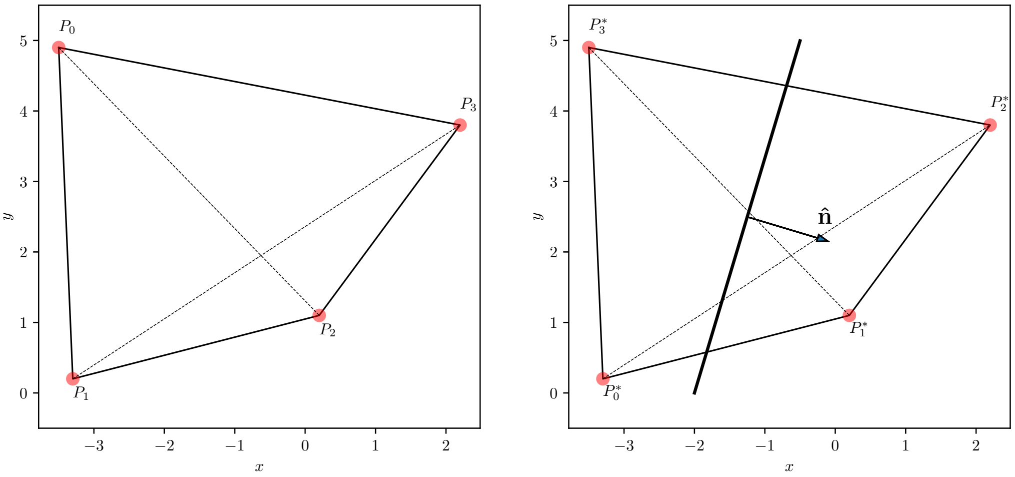


Figure 3.4: Interface reconstruction inside a generic shaped quadrangle. Original element, on the left, and element with reordered nodes, on the right, together with interface orientation represented, thick line, and unit normal vector $\hat{\mathbf{n}}$.

where γ is an arbitrary diffusion coefficient. Galerkin interpolation scheme (3.5) is obtained from general equation (2.22): source and target mesh, for this application, coincides and source field gradient values are null across each cell as source field α is a piece-wise field. Field α' can be used to perform a linear reconstruction of real solid-fluid interface $\Gamma_{f,s}$ as labeled in Fig. 3.1 a) and b). An example of obtained α and α' fields, together with a linear interface reconstruction are reported in Fig. 3.2 a), b) and c), respectively, for a case of circular solid body, whose mesh is shown together with reconstructed interface.

3.1.2 Interface reconstruction

The reconstructed fluid-solid interface can be used for evaluating quantities of interest, like the stress exerted from the fluid on solid body surface. A widely used method for interface calculation is based on level set method where a field $\psi(\boldsymbol{x})$ is introduced and defined as the distance between mesh nodes and the immersed body surface. The field $\psi(\boldsymbol{x})$ has positive values for mesh nodes belonging to fluid region while negative values characterize mesh nodes belonging to solid region. The solid-fluid interface is then defined as $\Gamma_i = \{\boldsymbol{x} \in \Omega : \psi(\boldsymbol{x}) = 0\}$ [27, 28, 34]. The construction of field ψ can be a cumbersome task for those applications where ψ can not be defined analytically and where the solid body moves.

The present method involves an interface reconstruction based on volume fraction field [35]. Since the volume fraction field calculation is handled by field projection routines, the interface reconstruction can be performed for arbitrarily

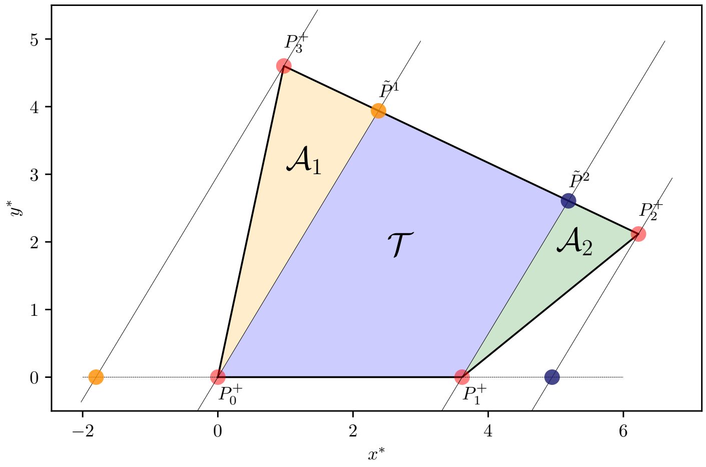


Figure 3.5: Interface reconstruction: decomposition of element Ω_e into two triangles $\mathcal{A}_1$, $\mathcal{A}_2$ and trapezoid $\mathcal{T}$ using lines parallel to interface.

shaped objects. As presented in Fig. 3.3, a generic mesh cell, labeled as Ω_j, having a volume fraction value $0 < \alpha_j < 1$, can be divided into sub-elements Ω_j^l and Ω_j^s by means of a straight edge. The interface normal unit vector $\hat{\mathbf{n}}$ is calculated as the ratio of local α' field gradient and the gradient modulus. The interface edge Γ_j^{sl} inclination is computed in accordance to $\hat{\mathbf{n}}$, while its position is determined from an area conservation equation, namely

$$\int_{\Omega_j^s} d\Omega = \alpha_j \int_{\Omega_j} d\Omega\,. \tag{3.6}$$

The algorithm for interface reconstruction has been developed to work with bi-dimensional grids made of arbitrarily shaped quadrilaterals, so that the fluid-solid interface can be reconstructed on non-regular meshes. For each intersected cell, i.e. having a value of α in the range $\alpha \in [0+\epsilon, 1-\epsilon]$ with ϵ being a tolerance factor, the intersection points between cell edges and linear interface are labeled as I^1 and I^2. Once the orientation of interface edge is calculated through the gradient of α', each intersected cell can be seen as the union of two triangles and one trapezoid, as shown in Fig. 3.5. A generic element Ω^e, made of 4 nodes P_0, P_1, P_2, P_3 ordered in counter-clockwise direction is considered, as represented on the left of Fig. 3.4. In order to simplify the computation of node I^1 and I^2 coordinates a new node numbering is defined, $P_0^*, P_1^*, P_2^*, P_3^*$, so that inclination of new diagonal $d_{0\to 2}$ is always higher than that of $d_{1\to 3}$, and P_0^* y coordinate value is smaller than P_2^* one. Starting from P_0^*, node numbering is assigned in counter-clockwise direction, as represented on the right of Fig. 3.4. In the Figure diagonals $d_{0\to 2}$ and $d_{1\to 3}$, connecting respectively nodes P_0^*-P_2^* and P_1^*-P_3^*, are drawn, together with a line denoting the orientation of interface to be reconstructed and the unit normal vector $\hat{\mathbf{n}} = -\boldsymbol{\nabla}\alpha'/|\boldsymbol{\nabla}\alpha'|$. A new reference frame x^*, y^* is defined as follows. First the reference frame is built in a way that axes origin coincides with node P_0^*, then, depending on the inclination of interface and of diagonals $d_{1\to 3}$ and $d_{0\to 2}$, the reference frame is rotated so that either node P_1^* or P_3^* will lay on x^* axis. In particular, by denoting with $d_{0,2}$, $d_{1,3}$ and d_I the angular coefficient of diagonals $d_{0\to 2}$, $d_{1\to 3}$ and of interface, reference frame is rotated so that P_1^* will lay on x^* axis if $d_I > d_{0\to 2}$ *or* $d_I < d_{1\to 3}$, otherwise so that P_3^* will lay on x^*. This procedure allows to use the family of lines parallel to the interface to decompose element Ω_e into two triangles and a trapezoid. For the sake of simplicity, a final node numbering $P_0^+, P_1^+, P_2^+, P_3^+$ is introduced. This numbering is determined so that P_0^+ is the node laying on x^* axis with minimum value of x^* coordinate and then proceeding in counter-clockwise direction, as shown in Fig. 3.5. Additional nodes $\tilde{P}^1$, $\tilde{P}^2$ are found as

- $\tilde{P}^1$ intersection of line parallel to interface, drawn from point P_0^+, with segment $\overline{P_3^+\,P_2^+}$ or with line parallel to interface, passing through P_3^+ and intersecting with segment $\overline{P_0^+\,P_1^+}$
- $\tilde{P}^2$ intersection of line parallel to interface, drawn from point P_1^+, with segment $\overline{P_3^+\,P_2^+}$ or with line parallel to interface, passing through P_2^+ and intersecting with segment $\overline{P_0^+\,P_1^+}$

For each of $\tilde{P}^1$ and $\tilde{P}^2$ two possible nodes are found, as shown in Fig. 3.5 with orange and blue colors respectively. The correct nodes are found by applying the constrain that both $\tilde{P}^1$ and $\tilde{P}^2$ must belong to Ω_e. Element Ω_e has been divided, in Fig. 3.5, into triangle A_1, with area $\mathcal{A}_1$, having vertices $\{P_0^+ , \tilde{P}^1 , P_3^+\}$, trapezoid T, with area $\mathcal{T}$ and vertices $\{P_0^+ , P_1^+ , \tilde{P}^2 , \tilde{P}^1\}$, and triangle A_2, with area $\mathcal{A}_2$ and vertices $\{P_1^+ , P_2^+ , \tilde{P}^2\}$. Triangle and trapezoid areas can be calculated with the following expression

$$S = \frac{1}{2}\left|\sum_{i=1}^{n} x_i(y_{i+1} - y_{i-1})\right| , \tag{3.7}$$

with n being the number of sides, (x, y) the coordinates of the vertices numbered in clockwise or counter-clockwise direction.

Depending on Ω_e α value, reconstructed interface can be placed in one of the three areas $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{T}$. Given Ω_e total area $\mathcal{A}$, cell area overlapped by the solid body is $\alpha\mathcal{A}$. Depending on the sign of x^* component of unit vector $\hat{\mathbf{n}}^*$, triangles and trapezoid areas can be ordered in the sequence $\{\mathcal{A}_i, \mathcal{T}, \mathcal{A}_j\}$ so that

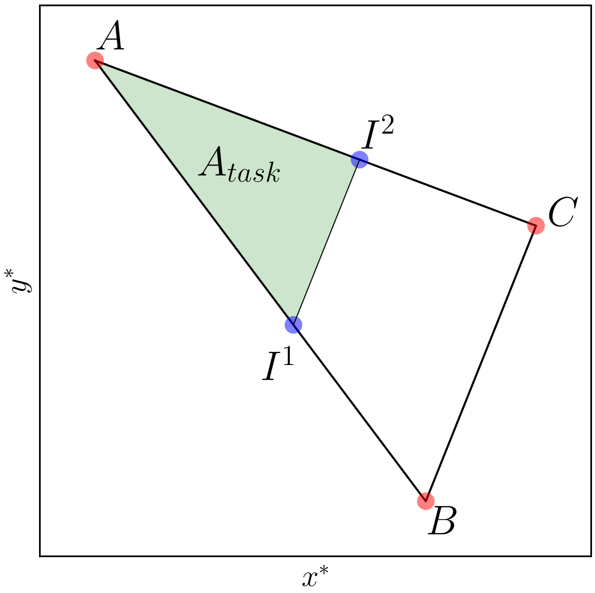


Figure 3.6: Interface reconstruction: case of interface points $\{I^1\,, I^2\}$ lying inside a triangle.

the following relations hold:

CASE A: if $\alpha\mathcal{A} \in [0, \mathcal{A}_i]$	then $\{I^1\,, I^2\} \in \mathcal{A}_i$
CASE B: if $\alpha\mathcal{A} \in [\mathcal{A}_i, \mathcal{A}_i + \mathcal{T}]$	then $\{I^1\,, I^2\} \in \mathcal{T}$
CASE C: if $\alpha\mathcal{A} \in [\mathcal{A}_i + \mathcal{T}, \mathcal{A}]$	then $\{I^1\,, I^2\} \in \mathcal{A}_j$

CASE A and CASE C

In the case of $\{I^1\,, I^2\}$ lying into a triangle, then coordinates of interface nodes can be calculated as follows. An area A_{task} is determined as

$A_{task} = \alpha\mathcal{A}$	for CASE A
$A_{task} = (1-\alpha)\mathcal{A}$	for CASE C

represented with green color in Fig. 3.6. Being A_{tr} the area of the triangle, $\mathcal{A}_i$ for CASE A or $\mathcal{A}_j$ for CASE B, the following relation valid for similar triangles can be used

$$\frac{\overline{A\,I^1}}{\overline{A\,B}} = \sqrt{\frac{A_{task}}{A_{tr}}}\,, \quad \frac{\overline{A\,I^2}}{\overline{A\,C}} = \sqrt{\frac{A_{task}}{A_{tr}}}\,, \tag{3.8}$$

to easily compute coordinate values for I^1 and I^2.

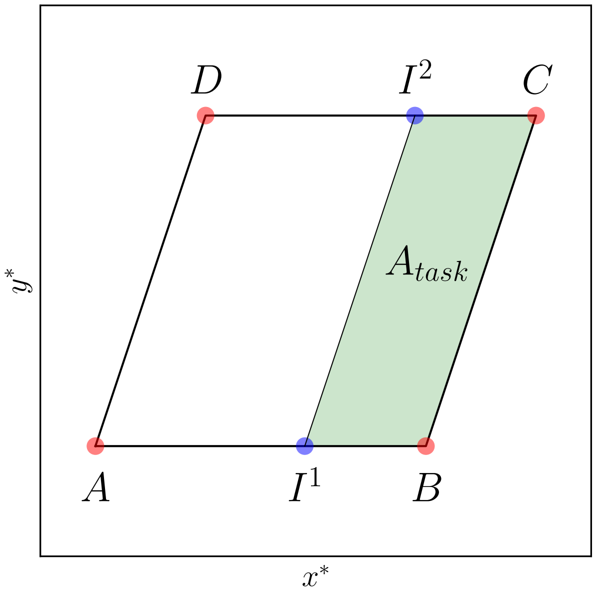


Figure 3.7: Interface reconstruction: CASE B1 of interface points $\{I^1\,, I^2\}$ lying inside the trapezoid, with limiting case of trapezoid being a parallelogram. Task area shown with green color.

CASE B

For CASE B of interface lying inside the trapezoid two variants are considered. The angular coefficient of the line passing through vertices P_2^+ and P_3^+ is labeled with $d_{2,3}$. The two cases are differentiated by $d_{2,3}$ value, namely the limiting case $d_{2,3} \simeq 0$ (CASE B1), for which the trapezoid is a parallelogram, and the more general case $d_{2,3} \neq 0$ (CASE B2). For CASE B1 a task area A_{task} can be calculated as

$$\begin{aligned} &\text{if } \hat{\mathbf{n}}^*(x) < 0 \qquad && A_{task} = \alpha\mathcal{A} - \mathcal{A}_i\,, \\ &\text{if } \hat{\mathbf{n}}^*(x) > 0 \qquad && A_{task} = \mathcal{T} + \mathcal{A}_i - \alpha\mathcal{A}\,, \end{aligned}$$

as shown in Fig. 3.7. Being dx the distance between nodes I^1 and B, and dy the height of the parallelogram, then $dx \cdot dy = 2A_{task}$, leading to an easy calculation of I^1 and I^2 coordinates. For the more general situation of CASE B2, an auxiliary node is used, P_{aux}, determined as the intersection between axes x^* and the line passing through vertices P_2^+ and P_3^+. As for precedent cases, it is useful to define a task area A_{task}. Following the trapezoid vertices numbering shown in Fig. 3.8, with $\mathcal{A}_{3,0,aux}$ we refer to the area of triangle with vertices $T_0\,, P_{aux}\,, T_3$. Task area A_{task}, shown in Fig. 3.8 with green color, is calculated as

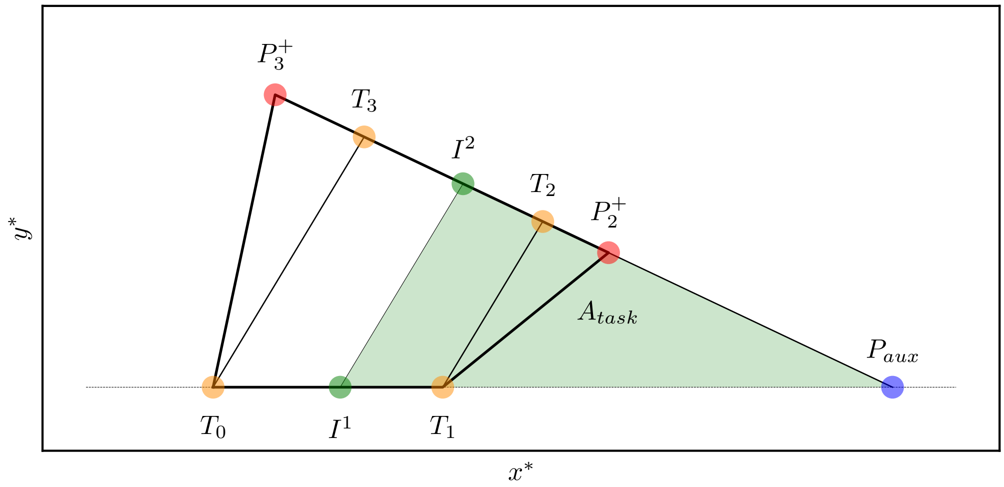


Figure 3.8: Interface reconstruction: CASE B2 general situation of interface points $\{I^1, I^2\}$ lying inside the trapezoid. Task area shown with green color and trapezoid vertices are highlighted with orange color and relative numbering.

- if $d_{2,3} > 0$

$$\begin{aligned} &\text{if } \hat{\mathbf{n}}^*(x) < 0 \qquad && A_{task} = \mathcal{A}_{3,0,aux} + \mathcal{T} + \mathcal{A}_i - \alpha\mathcal{A}\,, \\ &\text{if } \hat{\mathbf{n}}^*(x) > 0 \qquad && A_{task} = \mathcal{A}_{3,0,aux} - (\mathcal{A}_i - \alpha\mathcal{A})\,, \end{aligned}$$

- if $d_{2,3} < 0$

$$\begin{aligned} &\text{if } \hat{\mathbf{n}}^*(x) < 0 \qquad && A_{task} = \mathcal{A}_{3,0,aux} - (\mathcal{T} + \mathcal{A}_i - \alpha\mathcal{A})\,, \\ &\text{if } \hat{\mathbf{n}}^*(x) > 0 \qquad && A_{task} = \mathcal{A}_{3,0,aux} + (\mathcal{A}_i - \alpha\mathcal{A})\,. \end{aligned}$$

With a more compact expression, the general formula holds

$$A_{task} = \mathcal{A}_{3,0,aux} + f_1 \left[\frac{1}{2}\mathcal{T} - f_2 \left(\frac{1}{2}\mathcal{T} + \mathcal{A}_i - \alpha\mathcal{A}\right)\right]\,, \tag{3.9}$$

$$f_1 = \text{sign}(d_{2,3})\,, \quad f_2 = \text{sign}(\hat{\mathbf{n}}^*(x))\,. \tag{3.10}$$

By construction, task area A_{task} is the area of the triangle with vertices I^1, P_{aux} and I^2, with I^1 and P_{aux} lying on x^* axes. The area of the triangle can then be calculated as

$$A_{task} = \frac{1}{2}\left|I^2(y) \cdot (P_{aux}(x) - I^1(x))\right|\,. \tag{3.11}$$

Interface point I^2 is determined as the intersection of the line with angular coefficient d_I passing through I^1 and the one with angular coefficient $d_{2,3}$ passing through P_{aux}, so

$$I^2(y^*) = d_I(I^2(x^*) - I^1(x^*)) = d_{2,3}(I^2(x^*) - P_{aux}(x^*))\,. \tag{3.12}$$

It is obtained

$$I^2(x^*) = \frac{d_{2,3}P_{aux}(x^*) - d_I I^1(x^*)}{d_{2,3} - d_I}\,, \tag{3.13}$$

$$I^2(y^*) = \frac{d_I d_{2,3}}{d_{2,3} - d_I}(P_{aux}(x^*) - I^1(x^*))\,. \tag{3.14}$$

Writing equation (3.14) into (3.11), x^* coordinate of interface node I^1 can be finally calculated as

$$\begin{aligned} I^1(x^*) = P_{aux}(x^*) + \mathcal{I}\,, \quad h = \left|\sqrt{\frac{d_I d_{2,3}}{d_{2,3} - d_I}2A_{task}}\right| \\ \mathcal{I} \in \{-h\,, h\} : \mathcal{I} \in [T_0(x^*), T_1(x^*)]\,. \end{aligned} \tag{3.15}$$

Once that I^1 and I^2 coordinates have been calculated in the local reference frame $x^*\,, y^*$ an inverse change of coordinates system is applied in order to obtain I^1 and I^2 in global reference frame $x\,, y$. With the above illustrated algorithm it is possible to reconstruct fluid-solid interfaces in arbitrarily shaped quadrilaterals, if affine elements are used, since straight element sides are required. The combination of non-regular meshes and immersed boundary method allows to solve the system of equations on a computational grid that fits the physical domain and to model the presence of obstacles with immersed boundary method. This strategy is similar to the immersed boundary method based on curvilinear background grids CURVIB, where governing equations are solved on a background curvilinear boundary-fitted mesh and moving bodies are modeled with immersed boundary method [29, 36]. It could be stated that with the proposed method, a solid body fitted grid Π is used, so the real solid-fluid interface is already known. This is surely true, but its use could be not very straightforward. The reconstructed interface is sure an approximation but, as a great advantage, each interface segment lies within a unique interface cell, so the computation of variables such as drag coefficient can be carried with sustainable computational effort. When performing integration over real interface line, using Gauss method, it is not known a priori where each Gauss node belongs on computational grid Ω, in order to interpolate the desired variables upon it. A point-locator algorithm should then be used, with a great increase of the computational cost and without the certainty of an increased accuracy in the computed values. As an example of interface reconstruction, the case of a more complex

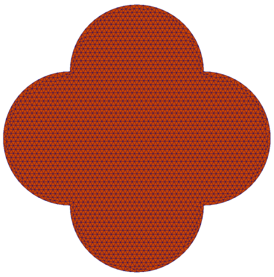


Figure 3.9: Test geometry for interface reconstruction.

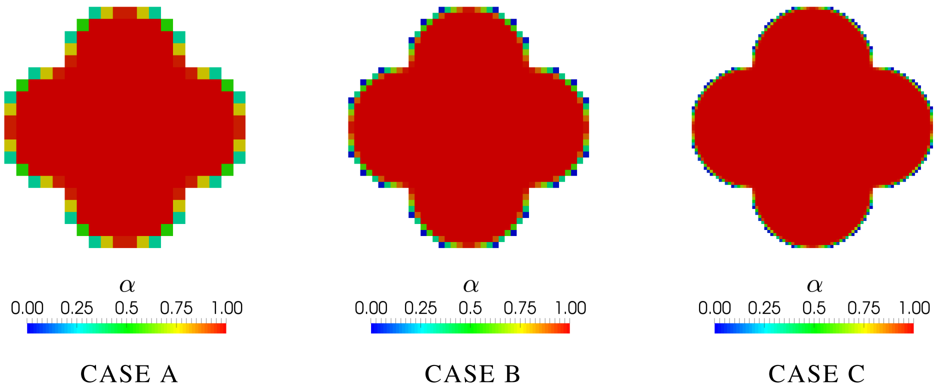


Figure 3.10: Interface reconstruction. Volume fraction field α, interpolated field α and reconstructed for three different target mesh grid resolutions.

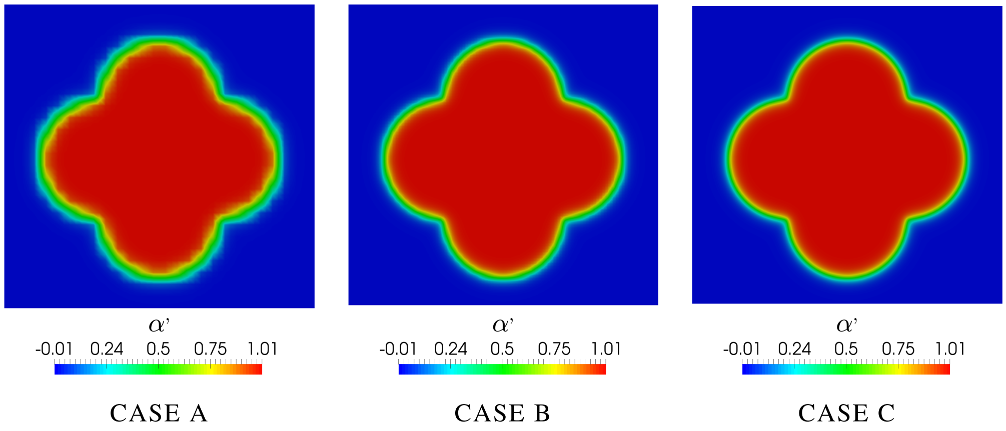


Figure 3.11: Interface reconstruction. Interpolated field α for three different target mesh grid resolutions.

object obtained from the union of 4 circles, represented in Fig. 3.9, is here consid-

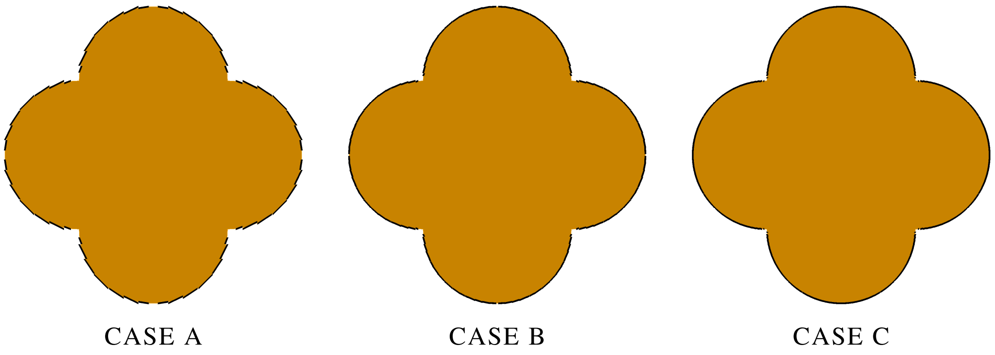


Figure 3.12: Interface reconstruction. Reconstructed interface compared with original geometry for three different target mesh grid resolutions.

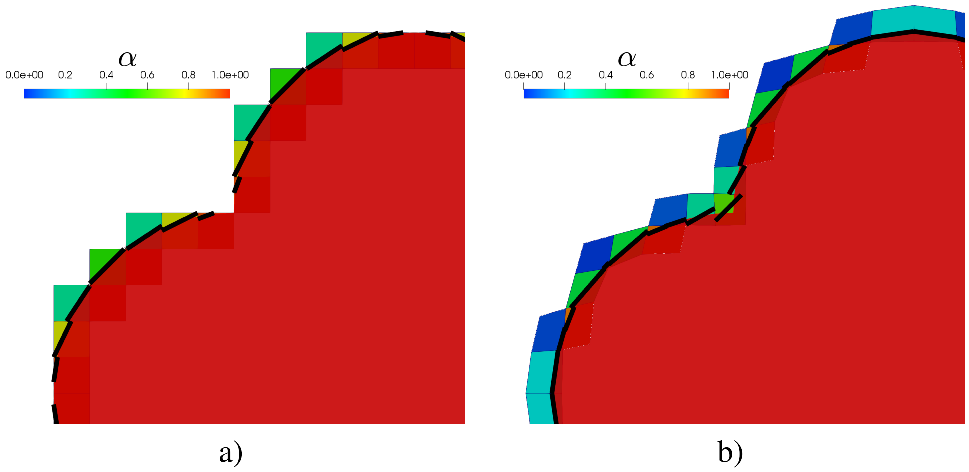


Figure 3.13: Interface reconstruction on a regular grid a) and on non-regular mesh b). With red color the original solid body, shown above α values of interface cells.

ered. Each circle has a diameter of 0.1m and the target grid is a squared domain with side $L = 0.4$m. Three different grid resolutions are considered, i.e. CASE A: 40×40, CASE B: 80×80 and CASE C: 160×160. For the three different cases it is obtained a resolution, expressed as number of target mesh cells per diameter, of 10, 20 and 40. The results are shown in Fig. 3.10, Fig. 3.11 and Fig. 3.12, in particular for volume fraction field α, the interpolated field α' and finally the reconstructed interface, compared with the original solid body shape. It can be seen that even for the coarsest case a good approximation of solid interface is obtained. As the grid resolution increases a more accurate reconstruction is obtained. It can be seen that interface near circle intersection points is not captured as for this simulated case the mesh grids are tangent to circle geometries in those points. An interface reconstruction of the same solid geometry is performed on a non-regular computational grid, as shown in Fig. 3.13 where a comparison on a close up view of reconstructed interfaces for CASE A is reported between results obtained with regular and deformed grid. The deformed mesh has been obtained starting from grid 40×40. A displacement vector field $\mathbf{S}$ has been applied to decrease the total amount of overlapped area of interface cells, that means

$$\mathcal{A}_o = \int_{\tilde{\Omega}^h} \alpha d\Omega \,, \quad \tilde{\Omega}^h = \left\{ \Omega_i^h \subseteq \Omega^h : \alpha_i \in [0+\epsilon, 1-\epsilon] \right\} \,. \tag{3.16}$$

It can be indeed observed from the values of α shown in Fig. 3.13 that the amount of overlapped interface cell area $\mathcal{A}_o$ is smaller for the case of deformed grid.

The Navier-Stokes system of equations is solved over the entire computational grid Ω. On mesh nodes that lay within the solid region the velocity field is imposed with a Dirichlet condition, so that momentum balance equation is not solved on those nodes. Incompressibility constraint is applied everywhere, with a modification for intersected cells expressed as follows. The weak formulation of the solved system of equation is obtained as

$$\begin{aligned}
&\int\limits_{\Omega} \frac{\partial \mathbf{u}}{\partial t} \cdot \boldsymbol{\varphi} \, d\Omega + \int\limits_{\Omega} (\mathbf{u} \cdot \boldsymbol{\nabla}) \mathbf{u} \cdot \boldsymbol{\varphi} \, d\Omega = \\
&\qquad = -\frac{1}{\rho} \int\limits_{\Omega} \boldsymbol{\nabla} P \cdot \boldsymbol{\varphi} \, d\Omega - \int_{\Omega} \nu (\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^T) : \boldsymbol{\nabla}\boldsymbol{\varphi} d\Omega + \\
&\qquad\quad + \int_{\Gamma} \left[\nu (\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^T) \cdot \mathbf{n} \right] \cdot \boldsymbol{\varphi} d\Omega \,, \qquad \forall \boldsymbol{\varphi} \in H_0^1(\Omega)
\end{aligned} \tag{3.17}$$

$$\begin{cases} \int_{\Omega_j} (\nabla \cdot \mathbf{u}) \, \psi \, d\Omega = 0 \,, & \forall \psi \in L^2(\Omega) \qquad \forall \Omega_j : \alpha_j < \tilde{\alpha} \\ \int_{\Omega_j} (\lambda \nabla^2 P - \nabla \cdot \mathbf{u}) \, \psi \, d\Omega = 0 \,, & \forall \psi \in L^2(\Omega) \qquad \forall \Omega_j : \alpha_j > \tilde{\alpha} \end{cases} \tag{3.18}$$

$$\mathbf{u}_i = \tilde{\mathbf{u}}_s \qquad \forall \mathbf{x}_i : \chi(\mathbf{x}_i, t) = 1 \,, \tag{3.19}$$

where λ is a constant value and $\tilde{\alpha}$ is a threshold value. In cells where $\alpha > \tilde{\alpha}$ we solve a pressure equation that is similar to the ones used in velocity projection methods [37]. It will be later shown that this equation allows to obtain a smoother pressure field. As reported in (3.19), the fluid velocity field in the solid-occupied computational region is modeled with the solid velocity field. In particular, the values $\tilde{\mathbf{u}}_s$ imposed in (3.19) are obtained by a point-wise projection of the velocity field $\mathbf{u}_s$ that is defined on the solid body mesh, namely $\tilde{\mathbf{u}}_s = |P|\mathbf{u}_s$. It can be seen from the solved system of equations, in the present immersed boundary formulation additional forces for the imposition of boundary conditions along the solid-fluid interface are not used. The algorithm used for the realization of the numerical simulation with the present immersed boundary method can be summarized, for a generic n-th time step, as:

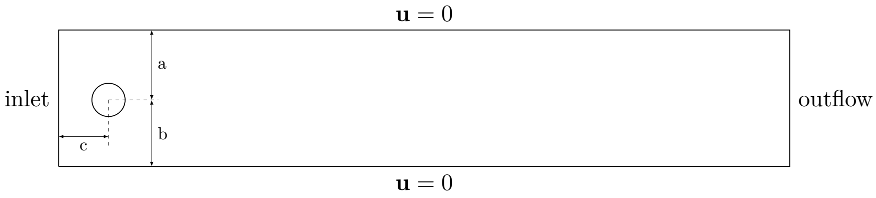


Figure 3.14: Sketch of the physical domain.

a $[m]$	b $[m]$	c $[m]$	d $[m]$	L $[m]$
0.21	0.2	0.2	0.1	2.2

Table 3.1: Geometrical parameters of the simulated domain.

1. Solid position on computational grid: creation of volume fraction α, projection of solid velocity $\tilde{\mathbf{u}}_s$ and indicator function χ^h
2. Solution of system of equations
3. Calculation of interpolated volume fraction α'
4. Reconstruction of solid-fluid interface $\Gamma_{f,s}$
5. Stress calculation
6. Update solid position and velocity field

We remark that the present algorithm can be used with arbitrarily moving solid objects as the indicator function is calculated using a field projection from the solid body mesh to the computational grid. Moreover, if the solid body position is fixed in time, in the above described algorithm steps (1), (3), (4) and (6) can be performed just once.

3.2 Results

In the present section we describe the results obtained for two different test cases, involving both a fixed and a moving obstacle.

3.2.1 Flow around a cylinder at Re=20

The present test consists of simulating a stationary fluid flow around a fixed cylinder. This benchmark case is well studied in [38]. Let U_0, d and ν be the mean inlet velocity, the cylinder diameter and the fluid kinematic viscosity, respectively. The resulting Reynolds number is $Re = U_0 d/\nu = 20$. A sketch of the simulated domain, together with the imposed boundary conditions, is reported in Fig. 3.14,

	REF0	REF1	REF2
$d/\delta x = d/\delta y$	7.2	14.5	29
ε	-1.5×10^{-5}	-6×10^{-6}	-1.4×10^{-6}

Table 3.2: Grid resolution and mass loss coefficient for the three different used computational grids.

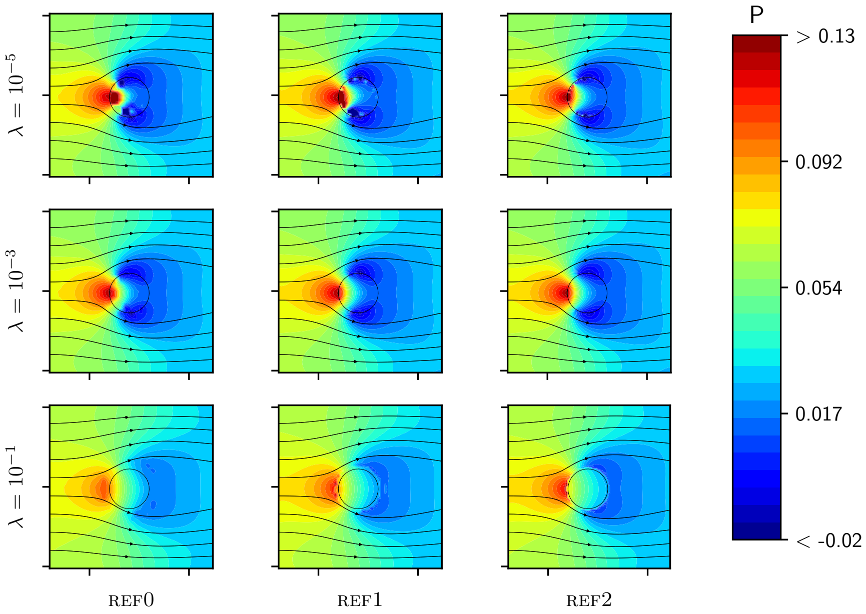


Figure 3.15: Influence of diffusion coefficient λ on pressure field and flow streamlines in the near solid region.

while the geometrical parameters are described in Tab. 3.1, where L is the length of the channel. A parabolic velocity profile is imposed on the inlet section, namely

$$U(x, y) = 4U_{max} \frac{(y_{max} - y)(y - y_{min})}{(y_{max} - y_{min})^2}, \tag{3.20}$$

where y_{max} and y_{min} are the channel maximum and minimum y coordinate values, and U_{max} is the maximum velocity value, $U_{max} = 1.5U_0$. One major concern when dealing with immersed boundary methods is mass conservation. For the present case we use three different computational grids with increasing levels of mesh refinements, in order to demonstrate that the proposed method allows to obtain a good mass conservation. The three different meshes are parameterized as a function of grid resolution with respect to the obstacle diameter and will be here-

		ΔP			c_d	
Case	10^{-5}	10^{-3}	10^{-1}	10^{-5}	10^{-3}	10^{-1}
REF0	0.1072	0.1122	0.0741	5.051	5.185	5.351
REF1	0.1105	0.1116	0.0671	4.897	5.014	5.224
REF2	0.1162	0.1120	0.0532	4.912	5.019	5.060
Reference		0.1175			5.579	

Table 3.3: Computed values of pressure difference ΔP between front and rear side of the cylinder and of drag coefficient c_d, together with reference data [38], ordered for increasing mesh refinements and values of diffusion coefficient λ.

after labeled as REF0, REF1 and REF2, where the number stands for the number of refinements. The grid resolution for the three different meshes is reported in Tab. 3.2 as a function of the ratio $d/\delta x$, where d is the obstacle diameter and δx is the mesh size along x direction. In order to evaluate the behavior of the proposed immersed boundary method regarding mass conservation, we report the values of mass loss fraction ε defined as $(U_0 - U_{m,out})/U_0$, where $U_{m,out}$ is the mean outlet velocity. It can be seen from the values reported in Tab. 3.2 that the mean outlet velocity is slightly higher than the mean inlet velocity value and mass loss fraction decreases by approximately an order of magnitude using a double mesh refinement. In the present test we studied the influence of the diffusion coefficient λ that is used in (3.17) for the pressure equation in the solid-occupied cells. In Fig. 3.15 we report a set of close-up views of the pressure fields obtained with the three different mesh refinements and with three different values of λ, namely 10^{-5}, 10^{-3} and 10^{-1}. We see that pressure spots arise in the vicinity of fluid-solid interface when using a small value of λ. Although λ value has not a sensible impact on mass conservation (the values of Tab. 3.2 refer to $\lambda = 10^{-3}$ but are quite the same for each of the tested values of λ), we see that λ has a great impact on fluid behavior in the near solid body region, as can be seen from the flow streamlines reported in Fig. 3.15. Finally we compare the obtained results for pressure difference ΔP between front and rear side of the cylinder and drag coefficient c_d with the reference ones, discussed in [38]. The computed values for all simulated cases, in terms of computational grid and λ values, are reported in Tab. 3.3 along with reference values. The drag coefficient is here calculated as

$$c_d = \frac{2F_x}{\rho d U_0^2} \quad \rightarrow \quad \mathbf{F} = \int_S \left[\mu\left(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^T\right) - p\mathbf{I}\right] \cdot \hat{n} dS\,, \tag{3.21}$$

where $\hat{n}$ is the normal to the solid-fluid interface, pointing from solid to fluid region, S is the area of solid-fluid interface and F_x the x component of the stress $\mathbf{F}$. In the present case the integral is calculated over the reconstructed interface.

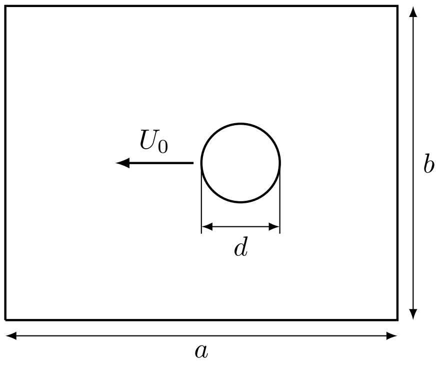
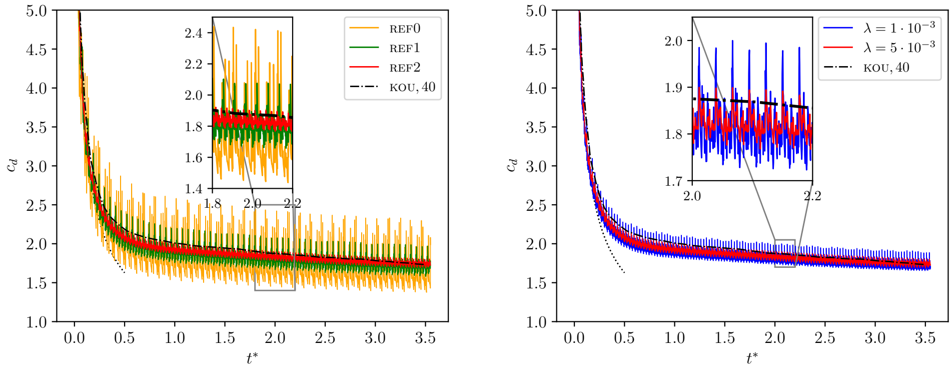


Figure 3.16: Sketch of the simulated domain for the case of flow around an impulsively started cylinder.



a) b)

Figure 3.17: Comparison of drag coefficient evolution, for cases REF0, REF1 and REF2 a), with analytical law (dotted line) [39] and numerical reference values (dash-dotted line) [40]. The same comparison is shown for case REF2 and two different values of λ b).

With small values of λ we observe a slight underestimation of pressure difference, ranging from 1 to 8 %, while with $\lambda = 0.1$ we obtain much higher errors (37 to 54 %). If we exclude the case $\lambda = 0.1$, the percentage difference between the computed drag coefficient and the reference one lays in the range 7 to 12 %.

3.2.2 Flow around an impulsively started cylinder

In the present test we simulate the case of a cylinder that is impulsively put in motion with a constant and uniform velocity $\mathbf{U}_0$. For this particular case an experimental study was performed in [43], as well as numerical simulations with

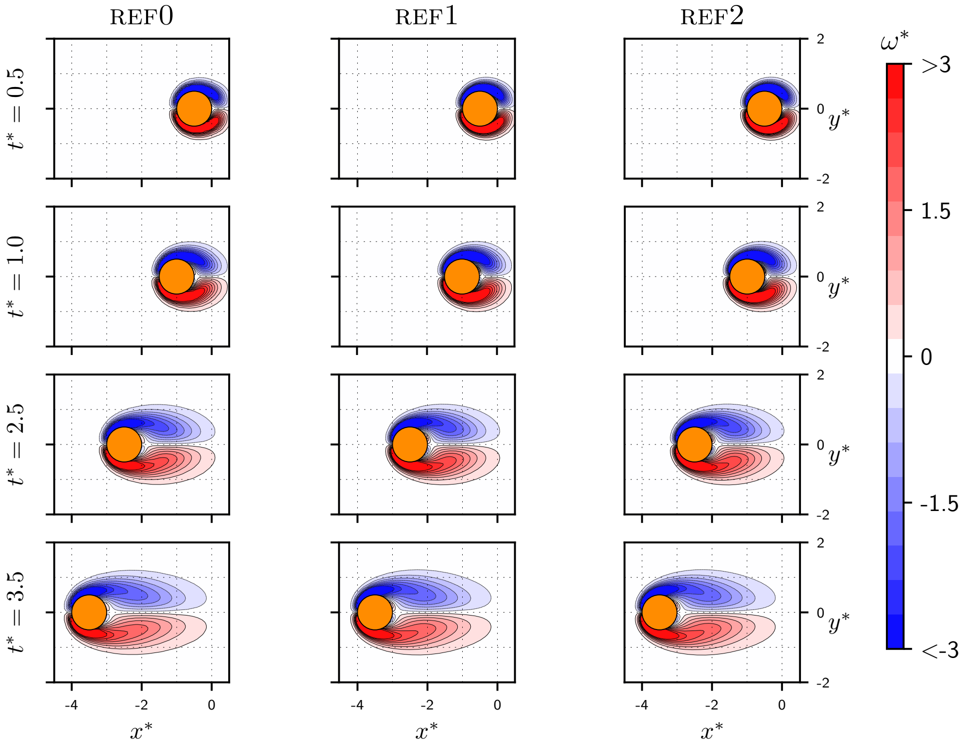
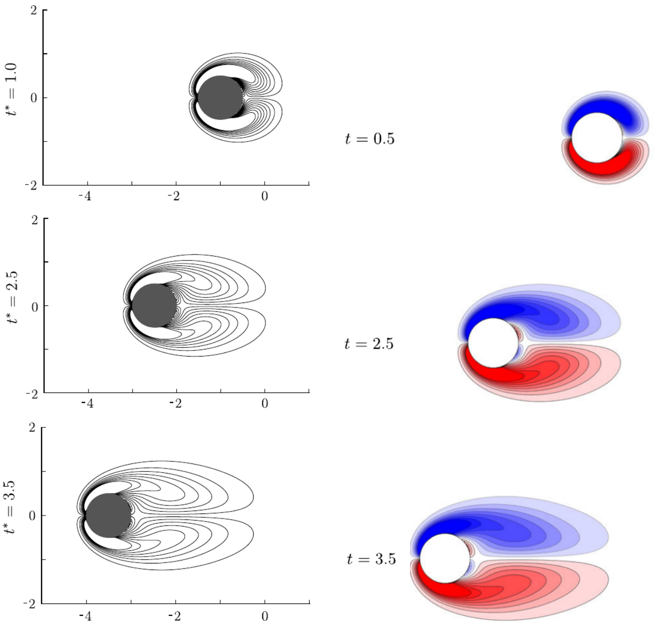


Figure 3.18: Equi-vorticity contours for the three simulated cases at three different non-dimensional time steps. Vorticity values range from -3 to 3 with a step of 0.4.

vortex method [39, 44, 40] and immersed boundary techniques [41, 42] for various Reynolds numbers. A sketch of the simulated domain is reported in Fig. 3.16, together with parameters a, b and d, denoting the domain width, height and cylinder diameter respectively. The computational box is characterized by the values $a = 12d$ and $b = 10d$. A no-slip boundary condition is imposed on all boundaries. The Reynolds number, based on cylinder velocity modulus U_0, cylinder diameter and fluid kinematic viscosity, is equal to 40. At time $t = 0$ the cylinder is put into motion with a constant velocity, moving from right to left. The simulated time interval is $t \in [0, 3.5\tilde{t}]$, where $\tilde{t}$ is the reference time constant, $\tilde{t} = d/U_0$. Numerical results will be presented as a function of non-dimensional time $t^* = t/\tilde{t}$, in order to compare them with literature values [39, 40, 41, 42]. We use two different computational grids that are characterized by the following mesh sizes δx and δy, along x and y direction: $\delta x = \delta y = d/9$ for case REF0, $\delta x = \delta y = d/18$ for case REF1 and $\delta x = \delta y = d/36$ for case REF2. In Fig. 3.17 we report the time evolution of the drag coefficient c_d for the three simulated cases and compare them with the reference numerical results [40] and with the analytical relation proposed



a) b)

Figure 3.19: Reference equi-vorticity contours for vorticity values ranging from -3 to 3 with a step of 0.4. Values at non dimensional times $t^* = 1, 2.5, 3.5$ a) and for non dimensional times $t^* = 0.5, 2.5, 3.5$ b). Pictures taken from: a) [41], b) [42].

in [39], namely

$$c_d = \frac{4\sqrt{\pi}}{\sqrt{t^*\ Re}} + \frac{\pi}{Re}\left(9 - \frac{15}{\sqrt{\pi}}\right), \tag{3.22}$$

which is valid for $t^* < 0.5$. As in the previous test, the drag coefficient is computed using (3.21) on the reconstructed solid-fluid interface. The comparison is made between results obtained with the three different mesh resolutions and $\lambda = 5 \cdot 10^{-3}$ in Fig. 3.17 a) and between values calculated with the finest grid and two different values of λ, namely $\lambda = 1 \cdot 10^{-3}$ and $\lambda = 5 \cdot 10^{-3}$. It can be seen that values of c_d are characterized by the presence of oscillations for each simulates case, with some peculiarities. From Fig. 3.17 a) it is seen that oscillations have higher peaks and occur with a larger frequence, for case REF0, while for REF2 they have smaller peaks and higher frequence. These oscillations are caused by the projection of solid velocity field onto the computational grid, altering the solved velocity field. As solid body moves, target mesh cells where velocity field is projected change with time. This fact leads to local lack of incompressibility, so that pressure peaks arise to restore the condition $\nabla \cdot \mathbf{u} = 0$. Since the computational grid has a homogeneous discretization, a pressure oscillation is observed every time that the

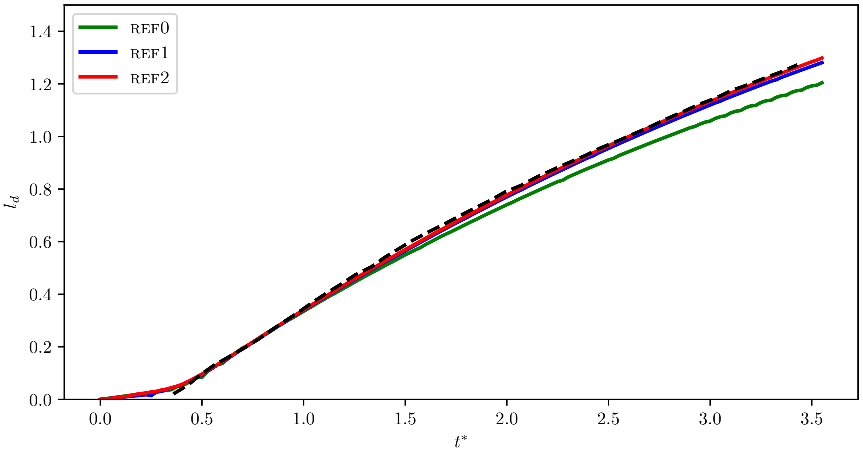


Figure 3.20: Time evolution of recirculation length l and comparison with reference data [43].

solid body changes its position by a quantity equal to grid size δx, i.e. with a period $\Delta t^* = \delta x/d$ expressed in non dimensional time. For coarser grids oscillations occur with higher period but when velocity field is projected on a new cell a more sensible modification is obtained, causing higher pressure peaks. In Fig. 3.17 b) the comparison of c_d time evolution is made between results obtained for case REF2 and two different values of λ. As expected, drag coefficient oscillations occur with the same frequency as the computational grid is the same for the two cases. Oscillation peaks are smaller the case of higher λ value, showing that a smoother pressure field is obtained. Drag coefficient values are slightly underestimated, with respect to the reference values [40], in the interval $t^* \in [0, 2]$, while for higher values of the non-dimensional time we observe a good agreement. A comparison between the simulated cases of the obtained vorticity ω values is reported in Fig. 3.18 for three different time steps, namely $t^* = 0.5$, $t^* = 1.0$, $t^* = 2.5$ and $t^* = 3.5$. In particular, non-dimensional vorticity ω^* contours are shown in a range $\omega^* \in [-3, 3]$ with increments of 0.4, in order to compare them with the reference results, shown in Fig. 3.19 [41, 42]. Comparing case REF0 and REF1 we see that grid spacing has a major impact on the evaluation of vorticity in the fluid area behind the cylinder and also in the estimation of a higher vorticity region. An increase of mesh resolution from REF1 to REF2 leads to less considerable changes in the predicted values of ω^*. We observe a very good agreement with the reference profiles even though the maximum mesh resolution used here is much smaller, namely $\delta x/d = 36$ instead of $\delta x/d = 50$ and $\delta x/d = 100$ [41, 42]. As a final test we study the evolution of the recirculation zone length l. Non-dimensional values of this variable, $l^* = l/d$, are plotted against non-dimensional time t^* in Fig. 3.20 for all the simulated cases and compared with reference values [43]. The length l^* is calculated by examining the velocity field $\mathbf{U} - \mathbf{U}_0$, i.e. from a reference frame defined on the moving cylinder. For the first part of the simulated time interval, namely $t^* < 1$, we observe a good agreement between our results and the reference ones. For higher values of non-dimensional time the results obtained from case REF0 increasingly underestimated l_d, while the ones of case REF1 and REF2 are still close to reference data.

With the results obtained from the two simulated cases some conclusions can be drawn. The present immersed boundary method is based on numerical fields projection, so on routines discussed in Chapter 2, proving the great versatility of the developed computational platform. Two computational grids are used: one for the solution of Navier-Stokes system of equations and one built on the solid body. This latter grid is particularly important because solid velocity field and additional fields are defined over it and then projected onto the first computational grid. Field projection allows to deal with static and moving objects, but also with arbitrarily shaped objects. An algorithm for solid-fluid interface reconstruction has been proposed to recover an approximated linear interface on fluid computational grid. The algorithm works with arbitrarily shaped quadrilateral affine elements so it can be used with non-regular grids bi-dimensional grids. Two different tests have been performed, to evaluate the accuracy of the present method with the cases of static and moving solid body. Obtained results have shown a good agreement with reference data [38, 39, 44, 40, 41, 42].

Future developments of the present method will involve an extension to three-dimensional problems. Interface orientation can be easily calculated, while interface reconstruction is a more difficult task, due to a greater number of possible intersection types [29]. Moreover the immersed boundary formulation can be extended to problems involving heat transfer and fluid turbulence. In particular the use of wall function boundary conditions can be an interesting strategy to deal with turbulence and immersed boundaries [45]. CFD simulations have been performed for this kind of geometries and mesh generation is not a trivial task [46, 47, 48, 49]. Complex geometries as wire wrapped nuclear reactor bundles can then be simulated using a mixed approach with body conforming mesh and immersed boundary method for modeling the wire geometry.

Chapter 4

Turbulence modeling

With the present PhD study a new logarithmic formulation of a four parameter turbulence model is proposed with the intent of providing a reliable tool for numerical study of heat transfer in low Prandtl number fluid turbulent flows. In the present Chapter the derivation of the four transport equations that are used in the four logarithmic parameter turbulence model is described. After a brief discussion on turbulent flow problems, the Reynolds Averaged Navier-Stokes system of equations (RANS) are introduced together with reference solutions that are used in the modeling process. The transport equations for turbulence representative variables are obtained from the RANS system and the modeling of eddy kinematic viscosity ν_t and turbulent thermal diffusivity α_t is explained.

Turbulent behavior of fluid flow is encountered in many everyday life situations. Since the first observations of Leonardo Da Vinci on fluid flow, with the drawings shown in Fig. 4.1, fluid turbulent behavior has been studied by scientists and researchers. For a given domain, with a representative space length l, with characteristic fluid velocity u and kinematic viscosity ν, the turbulent behavior is normally observed when the Reynolds number $Re = ul/\nu$ has a high value, being $Re > 10^3$ a good hint. Different from the laminar case, turbulent flows are irregular and chaotic with vortex structures that develop on a wide range of length scales. Increased diffusivity comes with the chaotic behavior, so friction and heat exchange are enhanced on solid walls. For Newtonian fluids, with density ρ and specific heat c_p, transient turbulent incompressible flows can be simulated using the Navier Stokes system of equations, consisting in a mass conservation equation,

Figure 4.1: Leonardo Da Vinci drawings from observations of water turbulent behavior.

a momentum balance equation and an energy balance equation as follows

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0\,, \tag{4.1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \boldsymbol{\nabla})\mathbf{u} = -\frac{1}{\rho}\boldsymbol{\nabla} \cdot \boldsymbol{\sigma} - \mathbf{F}_b\,, \tag{4.2}$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} T = \frac{1}{\rho c_p}\boldsymbol{\nabla} \cdot \mathbf{q}\,, \tag{4.3}$$

where (4.1) represents the constrain for incompressible flows, (4.2) the momentum balance equation and (4.3) the energy balance equation. The state variables, describing the fluid in this system of equations, are the velocity vector $\mathbf{u}$, temperature T, the stress tensor $\boldsymbol{\sigma}$ and the heat flux $\mathbf{q}$. The buoyancy force $\mathbf{F}_b$ is calculated with Boussinesq assumption as $\mathbf{F}_b = \mathbf{g}(1 - \beta(T - T_r))$, where β is the thermal expansion coefficient, $\mathbf{g}$ the gravity vector and T_r a reference temperature. We set T_r such that $\rho_r = \beta T_r$ with ρ_r the reference density. In order to close the system of equations, the stress tensor is modeled as

$$\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\tau}\,, \tag{4.4}$$

where p is the pressure field and $\boldsymbol{\tau}$ the viscous stress tensor that, for a Newtonian fluid, is

$$\boldsymbol{\tau} = 2\mu\mathbf{D} - \frac{2}{3}\mu\,(\boldsymbol{\nabla} \cdot \mathbf{u})\,\mathbf{I} = \mu\left(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^T\right) - \frac{2}{3}\mu\,(\boldsymbol{\nabla} \cdot \mathbf{u})\,\mathbf{I}\,. \tag{4.5}$$

In case of an incompressible fluid the viscous stress expression reduces to $\boldsymbol{\tau} = 2\mu\mathbf{D}$, where $\mathbf{D}$ is called as strain tensor. The heat flux vector $\mathbf{q}$ is modeled using Fourier's law

$$\mathbf{q} = -\lambda\boldsymbol{\nabla} T\,, \tag{4.6}$$

where λ is the fluid thermal conductivity. The system of equations is then transformed into

$$\nabla \cdot \mathbf{u} = 0\,, \tag{4.7}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p + \nabla \cdot \boldsymbol{\tau} - \mathbf{F}_b\,, \tag{4.8}$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \alpha \nabla^2 T\,. \tag{4.9}$$

The above system of equations may be used to simulate both laminar and turbulent flows if a proper grid resolution is adopted. In turbulent flows, eddies develop on a wide range of characteristic lengths. Larger eddies take place in the bulk flow region. Their structure is then stretched and distorted and the kinetic energy is transferred to smaller eddies. This process, called *energy cascade*, continues until the kinetic energy is dissipated to thermal energy by action of viscous forces. A balance equation for the mechanical energy can be written by multiplying (4.8) with velocity $\mathbf{u}$, obtaining

$$\frac{\rho}{2}\frac{\partial u^2}{\partial t} = -\frac{\rho}{2}\mathbf{u} \cdot \nabla u^2 - \nabla \cdot (p\mathbf{u}) + p\nabla \cdot \mathbf{u} + \nabla \cdot (\boldsymbol{\tau} \cdot \mathbf{u}) - \boldsymbol{\tau} : \nabla \mathbf{u} + \mathbf{F}_b \cdot \mathbf{u}\,. \tag{4.10}$$

On the right hand side of (4.10), the first is the convective term, the second represents the pressure work, the third takes into account the conversion of mechanical energy into internal energy due to expansion/contraction, the fourth and the last the viscous force and buoyancy force work, respectively. Since the viscous stress tensor is a symmetric one, the term $\varepsilon = \boldsymbol{\tau} : \nabla \mathbf{u}$ can be written as

$$\boldsymbol{\tau} : \nabla \mathbf{u} = \frac{1}{2}\boldsymbol{\tau}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) = \mu \mathbf{D} : \mathbf{D}\,, \tag{4.11}$$

and it is then always positive. Having a negative sign in (4.10), ε always acts as a dissipative term which transforms mechanical energy into thermal energy. This is related to velocity gradients, so it is higher where higher velocity gradients occur. The size of the smallest eddies can be estimated with the Kolmogorov characteristic lengths that are determined using fluid kinematic viscosity ν and the dissipation of kinetic energy ε, as dissipation occurs at the finer scales by action of viscous forces. Reference scales for velocity, space and time are then obtained as

$$u_k = (\nu\varepsilon)^{1/4}\,, \quad l_k = (\nu^3/\varepsilon)^{1/4}\,, \quad t_k = (\nu/\varepsilon)^{1/2}\,. \tag{4.12}$$

On a bulk scale, the reference velocity and length are labeled as u_0 and l_0 and are used to define the Reynolds number $Re = u_0 l_0/\nu$. During the energy process the kinetic energy is transferred from larger to finer eddies at a constant rate ε, so its order of magnitude can be estimated using bulk reference scales, $\varepsilon \simeq u_0^3/l_0$. The ratio bulk to Kolmogorov reference scales leads to the following values [50]

$$u_0/u_k = (\nu\varepsilon)^{-1/4}u_0 = (u_0 l_0/\nu)^{1/4} = Re^{1/4}\,, \tag{4.13}$$

$$l_0/l_k = (\nu^3/\varepsilon)^{-1/4}l_0 = (nu^3/u_0^3 l_0^3)^{-1/4} = Re^{3/4}\,, \tag{4.14}$$

$$t_0/t_k = (\nu/\varepsilon)^{-1/2}t_0 = (u_0 l_0/\nu)^{1/2} = Re^{1/2}\,. \tag{4.15}$$

Turbulent flows can be fully resolved with (4.7-4.9) if the grid resolution is smaller then Kolmogorov length scale, i.e. with a number of elements that is proportional to $Re^{9/4}$ for a three-dimensional simulation. For theses reasons complex systems require a huge amount of resources when studied with Direct Numerical Simulations (DNS).

Turbulence models have been developed with the aim of overcoming the computational requirements of DNS simulations still retaining the main physical features of the flow. Among them, Reynolds Averaged Navier Stokes systems of equations (RANS) are widely used. They are based on a decomposition of instantaneous variables as the sum of a mean and a fluctuating value

$$a = \overline{a} + a'\,, \tag{4.16}$$

where $\overline{(\cdot)}$ stands for the mean operator and $(\cdot)'$ for the fluctuation value. The RANS approach involves time averaging

$$\overline{a}(\mathbf{x}, t) = \frac{1}{\Delta t}\int\limits_{t}^{t+\Delta t} a(\mathbf{x}, t')dt'\,, \tag{4.17}$$

where the time interval Δt needs to be sufficiently large to establish an unique mean value $\overline{a}$. It is remarked that the average of an averaged value $\overline{a}$ is still equal to $\overline{a}$. Having this in mind, from equations (4.16) and (4.17) the following relation holds

$$\overline{a} = \overline{\overline{a} + a'} = \overline{a} + \overline{a'} \;\rightarrow\; \overline{a'} = 0\,, \tag{4.18}$$

so the mean value of the fluctuating component is always considered to be equal to zero. Some useful properties on time averaging can be recalled as follows [51]:

- independence from constant multiplication

  $$\overline{c\phi} = c\overline{\phi}$$

- addition has no effect on averaging

  $$\overline{\psi + \phi} = \overline{\psi} + \overline{\phi}$$

- commutation for differentiation and integration

$$\overline{\frac{\partial\phi}{\partial x}} = \frac{\partial\overline{\phi}}{\partial x}\,, \qquad \overline{\int \phi dx} = \int \overline{\phi} dx$$

The mean value of the product between fluctuations cannot be considered null, so the mean value of the product between to instantaneous properties a and b is equal to

$$\overline{ab} = \overline{(\overline{a} + a')(\overline{b} + b')} = \overline{\overline{a}\overline{b}} + \overline{\overline{a} \cdot b'} + \overline{a' \cdot \overline{b}} + \overline{a'b'} = \overline{a}\overline{b} + \overline{a'b'}\,.$$

4.1 The Reynolds Averaged Navier Stokes system of equations

The RANS system of equations is obtained by applying the mean operator to the system (4.7-4.9). For the continuity equation, incompressibility constrain must be satisfied by both instantaneous and mean velocity field, leading to

$$\begin{aligned} &\boldsymbol{\nabla} \cdot \mathbf{u} = 0 \\ &\overline{\boldsymbol{\nabla} \cdot \mathbf{u}} = \boldsymbol{\nabla} \cdot \overline{\mathbf{u}} + \boldsymbol{\nabla} \cdot \overline{\mathbf{u}'} = \boldsymbol{\nabla} \cdot \overline{\mathbf{u}} = 0\,. \end{aligned} \tag{4.19}$$

As a consequence, also the fluctuating velocity field $\mathbf{u}'$ is incompressible. The mean momentum balance equation is

$$\begin{aligned} \frac{\partial\overline{\mathbf{u}}}{\partial t} + \overline{\mathbf{u}} \cdot \boldsymbol{\nabla}\overline{\mathbf{u}} = -\frac{1}{\rho}\boldsymbol{\nabla}\overline{p} + \boldsymbol{\nabla} \cdot \left[\nu(\boldsymbol{\nabla}\overline{\mathbf{u}} + \boldsymbol{\nabla}\overline{\mathbf{u}}^T)\right] + \\ - \boldsymbol{\nabla} \cdot \overline{\mathbf{u}'\mathbf{u}'} + \beta\mathbf{g}(\overline{T} - T_r)\,. \end{aligned} \tag{4.20}$$

Equation (4.20) is very similar to (4.8), with the term $\boldsymbol{\nabla} \cdot \overline{\mathbf{u}'\mathbf{u}'}$ being the only exception. In particular this term, known as Reynolds stress tensor, represents a transport of momentum due to fluctuating velocities. In a similar way the mean energy balance equation is obtained as

$$\frac{\partial\overline{T}}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla}\overline{T} = \boldsymbol{\nabla} \cdot (\alpha\boldsymbol{\nabla}\overline{T}) - \boldsymbol{\nabla} \cdot \overline{\mathbf{u}'T'}\,. \tag{4.21}$$

As for the case of mean momentum balance equation, an unknown term is introduced in (4.21) as a consequence of convective term averaging process. This new term $\boldsymbol{\nabla} \cdot \overline{\mathbf{u}'T'}$ is known as turbulent heat flux and represents a transport of energy that combines fluctuating temperature and velocity fields. The averaging process gives a new system of equations that can be solved with a smaller amount of computational resources if compared to those required by DNS simulations. On the other hand, information on instantaneous fields are lost, as a consequence of the averaging process, and new unknown terms are introduced in the system (4.19-4.21), namely the Reynolds stress tensor $\boldsymbol{\tau}^r = -\rho\overline{\mathbf{u}'\mathbf{u}'}$ and the turbulent heat flux $\mathbf{q}^r = -\rho c_p\overline{\mathbf{u}'T'}$, which depends on fluctuating velocity and temperature fields. These terms are important and need to be carefully modeled since it is through them that the effect of turbulence on the mean flow can be studied. Unknown components of $\boldsymbol{\tau}^r$ and $\mathbf{q}^r$ can be calculated by solving a proper transport equation for each one of them, that consists in a system of nine additional equations for a three-dimensional problem. This approach increases the computational cost and requires an accurate modeling for the unknown term such as the $\boldsymbol{\tau}^r$ and $\mathbf{q}^r$ components. A widely used closure strategy consists of modeling Reynolds stresses and turbulent heat flux components with mean velocity and temperature gradient components and two turbulent diffusion coefficients, the eddy kinematic viscosity ν_t and eddy thermal diffusivity α_t

$$\tau^r_{i,j} = 2\nu_t\overline{D}_{i,j}\,, \tag{4.22}$$

$$q^r_i = \alpha_t\boldsymbol{\nabla}\overline{T}\cdot\hat{\mathbf{i}}\,, \tag{4.23}$$

where $\overline{\mathbf{D}}$ is the mean strain tensor and $\hat{\mathbf{i}}$ is the i-th unit vector direction. The new turbulent diffusion coefficients are modeled with the turbulent kinetic energy $k = 0.5\,\overline{u'_iu'_i}$ and characteristic time scales τ_{lk}. In the present work we use a four logarithmic parameter turbulence model, i.e. a model composed by a system of four transport equations that are solved to obtain turbulent characteristic quantities as a function of the turbulent kinetic energy k, its dissipation rate ε, mean squared temperature fluctuations $k_\theta = 0.5\,\overline{T'T'}$ and their dissipation rate ε_θ. Characteristic time scales for dynamical and thermal turbulence are calculated with these new variables, leading to the following simplified equations for ν_t and α_t

$$\nu_t = C_\nu k\tau_\nu(k,\varepsilon)\,, \quad \alpha_t = C_\alpha k\tau_\alpha(k,\varepsilon,k_\theta,\varepsilon_\theta)\,, \tag{4.24}$$

where C_ν and C_α are model constants and τ_ν and τ_α are two modeled time scales. In the following sections the exact transport equations for k, ε, k_θ and ε_θ are derived and discussed. The four logarithmic parameter turbulence model is obtained and, finally, the models for ν_t and α_t are presented. As it will be shown, the closure problem cannot be solved with the introduction of k, ε, k_θ and ε_θ exact transport equations because many unknown terms, i.e. higher order moments, appear in those equations. It is then necessary to model unknown terms in order to get a closure of the interested system of equations. For this reason some hypothesis are introduced [52]:

1. Turbulent diffusion of a variable φ is considered proportional to its gradient.
2. Small eddies are isotropic.

3. All turbulent quantities depend on $\overline{u_i'u_j'}$, $\overline{u_i'T'}$, k, ε, k_θ, ε_θ, $\mathbf{u}$, T, p, ν and α.
4. Turbulent scales are function of k, ε and ν, in particular (k, ε) for large eddies and (ν, ε) for small eddies.

Among the above mentioned hypothesis, 2) implies that off-diagonal components of Reynolds stresses are zero and that diagonal components are all equal to $2/3k$ [53]. A clear explanation of the physical meaning of isotropic turbulence at small scales can be found in the work of Bradshaw [54].

4.1.1 Law of the wall

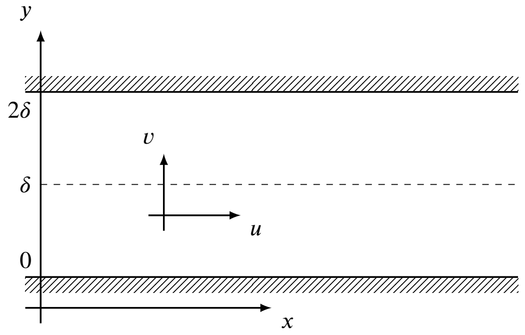


Figure 4.2: Sketch of a bi-dimensional channel, with x being the mean flow direction and δ half channel width along wall normal direction y.

A useful reference solution for the fully developed turbulent channel flow can be obtained from the non-buoyant version of (4.20). We consider a bi-dimensional domain Ω, as shown in Fig. 4.2, where x is the main flow direction and δ the half channel width in y direction. As a consequence of fully developed turbulent flow hypotheses, velocity gradient components along main flow direction vanish. From the incompressibility constrain and no-slip boundary conditions on the wall we obtain

$$\frac{\partial \overline{v}}{\partial y} = 0\,, \quad \overline{v}_w = 0 \;\rightarrow\; \overline{v} = 0 \;\forall (x, y) \in \Omega\,. \tag{4.25}$$

Wall shear stress is labeled as τ_w,

$$\tau_w = \mu \left.\frac{\partial \overline{u}}{\partial y}\right|_w\,, \tag{4.26}$$

and the friction velocity u_τ is defined as

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}}\,, \tag{4.27}$$

where ρ is the fluid density. Friction velocity and fluid kinematic viscosity ν are used as reference parameters for the calculation of non-dimensional variables

$$u^+ = \frac{\overline{u}}{u_\tau} \qquad y^+ = \frac{u_\tau}{\nu} y\,. \tag{4.28}$$

Friction Reynolds number $Re_\tau = u_\tau \delta / \nu$ is finally defined.

The steady equation (4.20) along x and y directions is then written as

$$\frac{\partial \overline{p}}{\partial x} - \frac{\partial}{\partial y}\left(\mu \frac{\partial \overline{u}}{\partial y} - \rho \overline{u'v'}\right) = \frac{\partial \overline{p}}{\partial x} - \frac{\partial \tau_{xy}^{eff}}{\partial y} = 0 \tag{4.29}$$

$$\frac{\partial \overline{p}}{\partial y} = 0\,, \tag{4.30}$$

where $\boldsymbol{\tau}^{eff}$ is the effective stress, i.e. $\boldsymbol{\tau}^{eff} = \boldsymbol{\tau} - \rho\overline{\mathbf{u}'\mathbf{u}'}$. Mean pressure field is a function of x, while effective stress component τ_{xy}^{eff} changes only along y direction as a consequence of fully developed turbulent flow hypothesis. Equation (4.29) is then satisfied only if

$$\frac{\partial}{\partial y}\left(\mu \frac{\partial \overline{u}}{\partial y} - \rho \overline{u'v'}\right) = C = \frac{\partial \overline{p}}{\partial x}\,, \tag{4.31}$$

where C is a constant value. By integrating (4.31) along y direction, from the wall up to an arbitrary y' coordinate, we obtain

$$\left[\mu \frac{\partial \overline{u}}{\partial y} - \rho \overline{u'v'}\right]_0^{y'} = y' \frac{\partial \overline{p}}{\partial x}\,. \tag{4.32}$$

At the wall effective stress component τ_{xy}^{eff} reduces to wall shear stress value τ_w as velocity fluctuations approach the no-slip boundary condition, so

$$\mu \frac{\partial \overline{u}}{\partial y} - \rho \overline{u'v'} = \tau_{xy}^{eff} = \tau_w + y' \frac{\partial \overline{p}}{\partial x}\,. \tag{4.33}$$

Integrating (4.31) from the wall to the channel half-width we obtain

$$\frac{\partial \overline{p}}{\partial x} = -\frac{\tau_w}{\delta}\,, \tag{4.34}$$

as the channel half-width is a symmetry plane both viscous and turbulent stress components τ_{xy} and τ_{xy}^R are null. By substituting (4.34) inside (4.33) the following expression for effective stress component τ_{xy}^{eff} can be written

$$\tau_{xy}^{eff} = \tau_w \left(1 - \frac{y}{\delta}\right) = \mu \frac{\partial \overline{u}}{\partial y} - \rho \overline{u'v'}\,, \tag{4.35}$$

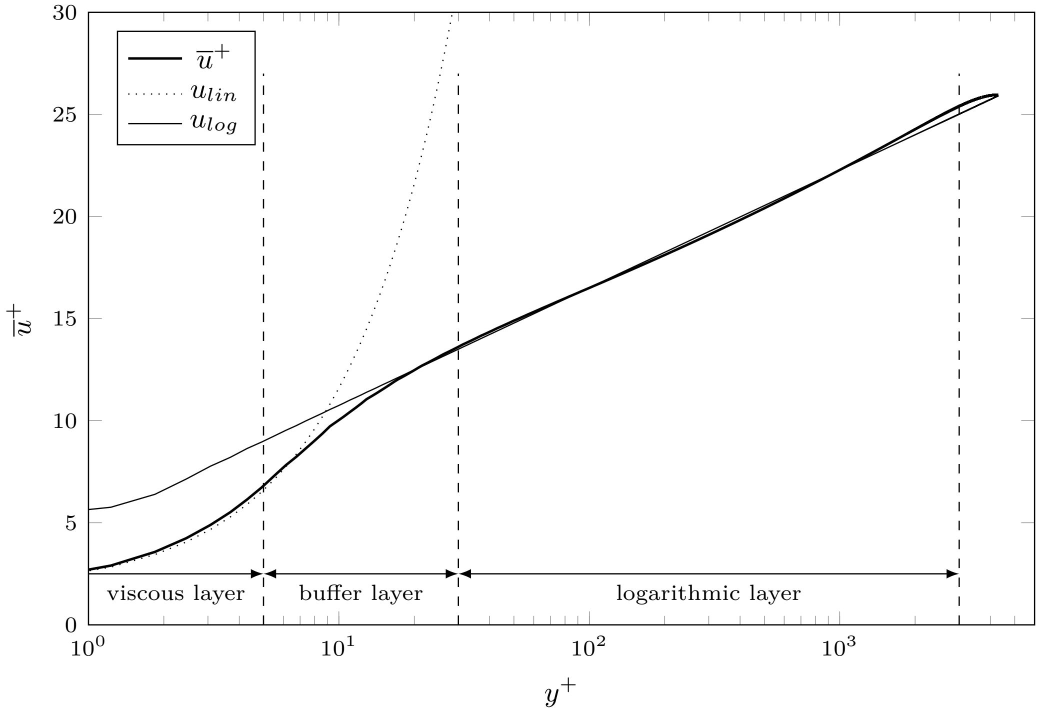


Figure 4.3: Non-dimensional velocity $\overline{u}^+$ profile plotted against logarithmic values of non dimensional wall distance y^+ for a fully developed turbulent flow in a plane channel with $Re_\tau = 4400$.

or, in non-dimensional form, as

$$\frac{\partial \overline{u}^+}{\partial y^+} - \overline{u'^+ v'^+} = 1 - \frac{y^+}{Re_\tau}\,. \tag{4.36}$$

It can be seen that effective stress component τ_{xy}^{eff} decreases linearly with wall distance y from the wall value τ_w. For small values of the ratio y^+/Re_τ, τ_{xy}^{eff} can be considered constant and equal to τ_w, so for higher friction Reynolds number values τ_{xy}^{eff} is approximately constant over a wider range of positions y^+. At the wall, from (4.36) we obtain

$$\frac{\partial \overline{u}^+}{\partial y^+} = 1 \quad \rightarrow \quad \overline{u}^+ = y^+ + \lambda\,, \tag{4.37}$$

where λ is a constant value and it is equal to zero from the no slip boundary condition. In the near wall region, non dimensional velocity u^+ increases linearly with non dimensional wall distance y^+. On the other hand, viscous contribution to effective stress is considered to be negligible in the region far from the wall and Reynolds stress component τ_{xy}^R to be constant and equal to wall shear stress value

$\tau_w = u_\tau^2$, so that u_τ can be used as a reference velocity value even in the region far from the wall. Derivative of mean velocity component $\overline{u}$ along y direction is then considered

$$\frac{\partial \overline{u}}{\partial y} = \frac{u_\tau}{\kappa y} \,, \tag{4.38}$$

where κ is a constant value. From Boussinesq hypothesis, where

$$-\overline{u'v'} = \nu_t \left(\frac{\partial \overline{u}}{\partial y} + \frac{\partial \overline{v}}{\partial x} \right) \,, \tag{4.39}$$

we obtain that eddy kinematic viscosity grows linearly with wall distance

$$\nu_t = \kappa u_\tau y \,. \tag{4.40}$$

From effective stress component equation we obtain

$$\tau^{eff} = \rho \left(\nu + \nu_\tau \right) \frac{\partial u}{\partial y} = \kappa u_\tau y \frac{\partial u}{\partial y} = \rho u_\tau^2 \,, \tag{4.41}$$

or, in non-dimensional form,

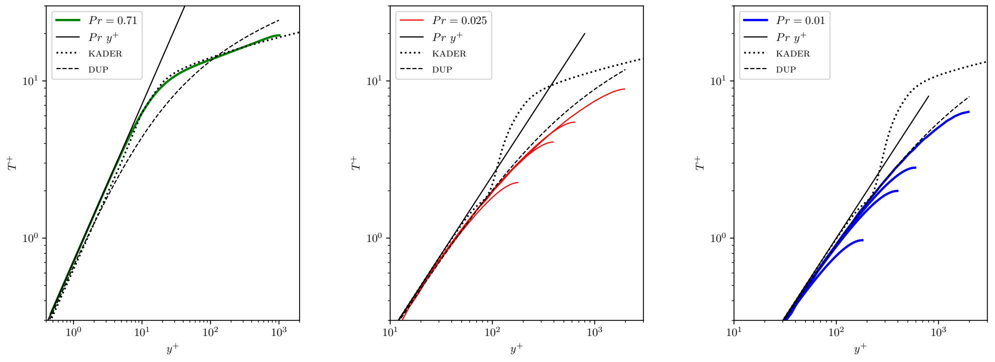
$$\frac{\partial u^+}{\partial y^+} = \frac{1}{\kappa y^+} \,. \tag{4.42}$$

From integration of (4.42) the well known logarithmic velocity behavior is obtained

$$u^+ = \frac{1}{\kappa} \ln(y^+) + B \,. \tag{4.43}$$

Constants κ, known as Von Karman constant, and B are equal to 0.41 and 5.2 respectively. The flow region can then be divided in two different zones: the first, closer to the wall, labeled as inner layer, and the second, labeled as outer layer, located far from the wall. Within the inner layer the linear and logarithmic velocity profiles can be observed in the range $y^+ \in [0; 5]$ and $y^+ \in [30; 300]$ respectively, as shown in Fig. 4.3, where the non dimensional velocity profile u^+ is plotted against non dimensional wall distance for the case of fully developed turbulent channel flow at $Re_t = 4400$. The layer $y^+ \in [5; 30]$ is defined as buffer layer and it is characterized by a velocity profile bending from linear to logarithmic.

Although the above velocity behavior, known as *law of the wall*, has been derived with simplifying assumptions, it has been observed in Direct Numerical Simulations of turbulent flows and is then used as reference behavior that must be reproduced by turbulence models.



a) b) c)

Figure 4.4: DNS temperature profiles for fully developed turbulent flows in plane channel. Molecular Prandtl number $Pr = 0.71$ a), $Pr = 0.025$ b) and $Pr = 0.01$ c). Profiles compared with linear law $Pr\ y^+$, Kader law and Duponcheel law.

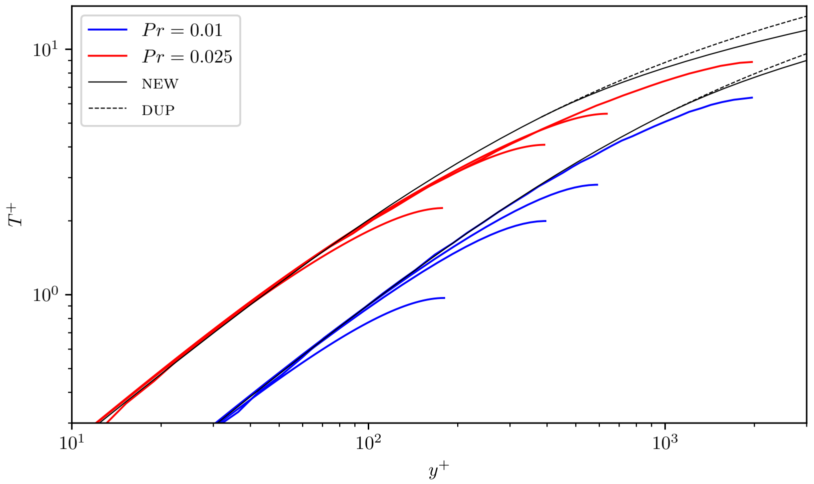


Figure 4.5: Comparison of DNS results for $Pr = 0.025$ and $Pr = 0.01$ with Duponcheel law and the newly proposed.

Law of the wall for temperature profile

For the case of a constant and uniform heat flux applied on solid boundary, near wall temperature profile exhibits a linear dependence on wall distance. In particular, non-dimensional temperature $T^+ = T/T_\tau$, where friction temperature T_τ is defined as $q/(u_\tau \rho c_p)$, is proportional to non-dimensional wall distance y^+ through molecular Prandtl number, i.e. $T^+ = Pr\ y^+$. The existence of a temperature logarithmic behavior, in the region far from the wall, has been questioned in literature [55]. For a simple flow configuration, as the fully developed turbulent flow in plane channel, the logarithmic behavior has been observed from DNS simulation results. A first logarithmic law has been proposed by Kader [56, 57]

$$\tilde{T}^+ = \frac{Pr_t}{\kappa} \ln\left(1 + y^+\right) + \beta(Pr) \, , \tag{4.44}$$

$$\beta(Pr) = \left(3.85 Pr^{1/3} - 1.3\right)^2 + \frac{Pr_t}{\kappa} \ln(Pr) \, . \tag{4.45}$$

The relation that includes both linear and logarithmic temperature behaviors contains exponential blending functions

$$T^+ = Pr\, y^+ e^{-G} + \left[\frac{Pr_t}{\kappa} \ln\left(1 + y^+\right) + \beta(Pr)\right] e^{-1/G} \, , \tag{4.46}$$

$$G = 0.01 (Pr\, y^+)^4 / (1 + 5 Pr^3 y^+) \, . \tag{4.47}$$

As can be seen from (4.44), the coefficient β is a function of the molecular Prandtl number, while for turbulent Prandtl number Pr_t the suggested value to use is 0.85. The correlation allows to fairly reproduce temperature profiles of fluids with a molecular Prandtl number close to 1, as represented in Fig. 4.4 a), but fails with low Pr fluids, Fig. 4.4 b) and c). A different formulation has been proposed by Duponcheel et al. [58], based on the heat flux conservation hypothesis in the near wall region. It is then obtained

$$\rho C_p \left(\alpha + \alpha_\tau\right) \frac{d\tilde{T}}{dy} = q \quad \to \quad \frac{d\tilde{T}}{dy} = \frac{q}{\rho C_p \left(\alpha + \alpha_\tau\right)} \, , \tag{4.48}$$

or, in non-dimensional form,

$$\frac{d\tilde{T}^+}{dy^+} = \frac{Pr}{1 + \alpha_\tau / \alpha} \, . \tag{4.49}$$

A linear behavior of α_t is assumed using molecular kinematic viscosity ν, non-dimensional wall distance y^+ and turbulent Prandtl number $Pr_t = 2.0$

$$\alpha_t = \nu \frac{\kappa}{Pr_t} y^+ \quad \to \quad \frac{d\tilde{T}}{dy^+} = \frac{Pr}{1 + \dfrac{\kappa\, Pr}{Pr_t} y^+} . \tag{4.50}$$

The above equation can be integrated over y^+

$$\tilde{T}^+ = \frac{Pr_t}{\kappa} \ln\left(1 + \frac{\kappa\, Pr}{Pr_t} y^+\right) \, . \tag{4.51}$$

For low values of y^+ equation (4.51) approximates very well the linear behavior, while for high values of y^+ it leads to a logarithmic law. When dealing with low

Prandtl number fluids, $Pr_t = 2.0$ can be a good assumption in the region close to the wall, while as wall distance increases the turbulent Prandtl number decreases. As it will be shown in Section 4.3 for the modeling of α_t, we assume a constant turbulent Prandtl number equal to $1.\overline{3}$, for the region far from the wall. We propose here a new logarithmic law for mean temperature profile, formulated as

$$\tilde{T}^+ = \frac{Pr_t}{\kappa} \ln\left(y^+\right) + C \,, \tag{4.52}$$

where C is a constant value that is determined through (4.51). New temperature law (4.52) is intended to be a correction of (4.51) for the region far from the wall and it is obtained assuming that the viscous contribution to effective heat flux is negligible. We calculate the non-dimensional coordinate $Pr\, y^+ = \tilde{y}^t$ for which the derivatives of (4.51) and (4.52) have the same value. Constant C is then determined so that the following function

$$\tilde{T}^+ = \begin{cases} \frac{Pr_t^1}{\kappa} \ln\left(1 + \frac{\kappa\, Pr}{Pr_t^1} y^+\right) \,,\; Pr_t^1 = 2.0 \quad \forall\, Pr\, y^+ < \tilde{y}^t \\ \frac{Pr_t^2}{\kappa} \ln\left(y^+\right) + C \,,\; Pr_t^2 = 1.\overline{3} \quad \forall\, Pr\, y^+ > \tilde{y}^t \end{cases} \tag{4.53}$$

is continuous. We obtain

$$\tilde{y}^t = \frac{Pr_t^1 Pr_t^2}{\kappa(Pr_t^1 - Pr_t^2)} \quad \rightarrow \quad C = \frac{Pr_t^1}{\kappa} \ln\left(1 + \frac{\kappa\, Pr}{2} \tilde{y}^t\right) - \frac{Pr_t^2}{\kappa} \ln\left(\tilde{y}^t\right) \,, \tag{4.54}$$

so the value of non dimensional coordinate $\tilde{y}^t$ is $\tilde{y}^t = 9.6833$ and constant C, for molecular Prandtl numbers $Pr = 0.01$ and $Pr = 0.025$ is $C_{0.01} = -16.97$, $C_{0.025} = -14$. A comparison of (4.51) and (4.53) with DNS results for $Pr = 0.01$ and $Re_\tau = 180$, 395, 590, $Pr = 0.025$ and $Re_\tau = 180$, 395, 640 is reported in Fig. 4.5.

4.1.2 Transport equation for turbulent kinetic energy

In this section we derive the exact transport equation for the turbulent kinetic energy k. We introduce the equation for the evolution of fluctuating velocity field as this equation will be used as a starting point for obtaining both k and ε transport equations. The Navier Stokes equation for instantaneous velocity field can be written as follows

$$\begin{aligned} &\frac{\partial \overline{\mathbf{u}}}{\partial t} + \frac{\partial \mathbf{u}'}{\partial t} + \left[(\overline{\mathbf{u}} + \mathbf{u}') \cdot \boldsymbol{\nabla}\right] (\overline{\mathbf{u}} + \mathbf{u}') = \\ &\quad = -\frac{1}{\rho} \boldsymbol{\nabla} \overline{p} - \frac{1}{\rho} \boldsymbol{\nabla} p' + \nu \nabla^2 \overline{\mathbf{u}} + \nu \nabla^2 \mathbf{u}' - \beta \mathbf{g}(T + T' - T_r) \,, \end{aligned} \tag{4.55}$$

where all solved variables have been decomposed as the sum of a mean and a fluctuating value. A transport equation for the fluctuating velocity field can be obtained by subtracting (4.20) from (4.55)

$$\begin{aligned} &\frac{\partial \mathbf{u}'}{\partial t} + \boldsymbol{\nabla} \cdot \overline{\mathbf{u}}\mathbf{u}' + \boldsymbol{\nabla} \cdot \mathbf{u}'\overline{\mathbf{u}} - \boldsymbol{\nabla} \cdot \overline{\mathbf{u}'\mathbf{u}'} = \\ &\qquad = -\frac{1}{\rho}\boldsymbol{\nabla} p' + \nu \nabla^2 \mathbf{u}' - \beta \mathbf{g}(T') \,. \end{aligned} \tag{4.56}$$

Equation (4.56) is more conveniently written in scalar form for a generic i-th component of $\mathbf{u}'$, obtaining

$$\begin{aligned} &\frac{\partial u_i'}{\partial t} + \frac{\partial u_i' u_j}{\partial x_j} + \frac{\partial u_j' u_i}{\partial x_j} + \frac{\partial u_j' u_i'}{\partial x_j} - \frac{\partial \overline{u_j' u_i'}}{\partial x_j} = \\ &\qquad = -\frac{1}{\rho}\frac{\partial}{\partial x_i} p' + \nu \frac{\partial^2 u_i'}{\partial x_j^2} - \beta g_i T' \,. \end{aligned} \tag{4.57}$$

Starting from (4.57), the exact transport equation for a generic component $\overline{u_i' u_j'}$ of Reynolds stress tensor can be obtained. We refer to Eq_{uj} as the transport equation (4.57) written for fluctuating velocity component u_j'. Transport equation for $\overline{u_i' u_j'}$ is obtained by multiplying Eq_{uj} with u_i' and Eq_{ui} with u_j'. Finally the mean operator is applied to the sum of the two equations, namely $\overline{u_i' \cdot Eq_{uj} + u_j' \cdot Eq_{ui}}$. The resulting equation is the following

$$\begin{aligned} &\frac{\partial \overline{u_i' u_l'}}{\partial t} + \underbrace{\overline{u_j} \frac{\partial \overline{u_i' u_l'}}{\partial x_j}}_{\text{I}} = \underbrace{- \frac{\partial}{\partial x_j}\left[\overline{u_i' u_l' u_j'} + \frac{\delta_{ij}}{\rho}\overline{u_l' p'} + \frac{\delta_{lj}}{\rho}\overline{u_i' p'} - \nu \frac{\partial \overline{u_i' u_l'}}{\partial x_j} \right]}_{\text{II}} \\ &\qquad \underbrace{- \frac{1}{\rho}\overline{p' \frac{\partial u_l'}{\partial x_i} + p' \frac{\partial u_i'}{\partial x_l}}}_{\text{III}} - \underbrace{\left[\overline{u_j' u_l'} \frac{\partial \overline{u_i}}{\partial x_j} + \overline{u_j' u_i'} \frac{\partial \overline{u_l}}{\partial x_j} \right]}_{\text{IV}} \\ &\qquad \underbrace{- 2\nu \overline{\frac{\partial u_i'}{\partial x_j} \frac{\partial u_i'}{\partial x_j}}}_{\text{V}} - \underbrace{\left[\beta g_i \overline{u_j' T'} + \beta g_j \overline{u_i' T'} \right]}_{\text{VI}} \,. \end{aligned} \tag{4.58}$$

The labeled terms represent I) the convective contribution, II) a diffusion term that depends on fluctuating pressure and velocity components and V) represents the molecular diffusion of the Reynolds stresses component. Term III) represents the interaction of fluctuating pressure with the fluctuating part of the strain tensor. This term is responsible for kinetic energy redistribution among the normal stresses, in an effort to recover an isotropic condition. The production term for

Reynolds stresses is represented by contribution IV) where Reynolds stresses interact with the mean velocity gradient components. Term V) is the dissipation term of the Reynolds stresses and term VI is a source depending on the interaction between gravity vector and turbulent heat flux components for buoyant flows. The exact transport equation for turbulent kinetic energy is obtained by setting subscript j equal to i in (4.58) and by performing a summation over i index, as, by definition

$$k = \frac{1}{2}\sum_{i}^{n} \overline{u'_i u'_i} \,.$$

The resulting equation is

$$\begin{aligned} &\frac{\partial k}{\partial t} + \overline{u_j}\frac{\partial k}{\partial x_j} = -\frac{\partial}{\partial x_j}\left[\frac{1}{2}\overline{u'_i u'_i u'_j} + \frac{\delta_{ij}}{\rho}\overline{u'_i p'} - \nu\frac{\partial k}{\partial x_j}\right] + \\ &- \overline{u'_j u'_i}\frac{\partial \overline{u_i}}{\partial x_j} - \nu\overline{\frac{\partial u'_i}{\partial x_j}\frac{\partial u'_i}{\partial x_j}} - \beta g_j \overline{u'_j T'} \,. \end{aligned} \tag{4.59}$$

As we see, the pressure strain term is not present in k equation as the trace of fluctuating strain tensor $\mathbf{D}'$ is equal to zero by effect of incompressibility constrain. The pressure strain term performs then a net redistribution of turbulent kinetic energy among the normal stresses, without altering the total value. The diffusive term related to pressure fluctuations is normally negligible and it is not considered here. The triple moment $\overline{u'_i u'_i u'_j}$ is modeled using a Simple Gradient Diffusion Hypothesis (SGDH) [59]

$$\frac{1}{2}\overline{u'_i u'_i u'_j} = -\frac{\nu_t}{\sigma_k}\frac{\partial k}{\partial x_j} \,, \tag{4.60}$$

where σ_k is a model constant and ν_t the eddy kinematic viscosity used to model the Reynolds stresses with Boussinesq assumption. The production term P^k

$$P^k = -\overline{u'_i u'_j}\frac{\partial \overline{u_i}}{\partial x_j} \,. \tag{4.61}$$

is modeled using (4.22), so

$$P^k = \left[\nu_t\left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i}\right) + \frac{2}{3}\delta_{ij} k\right]\frac{\partial \overline{u_i}}{\partial x_j} \,. \tag{4.62}$$

It is easy to see that $\delta_{i,j}\partial\overline{u_i}/\partial x_j$ is equal to zero for an incompressible flow. We introduce the anti-symmetric tensor $\boldsymbol{\Omega}$, whose generic component $\Omega_{i,j}$ is defined as

$$\Omega_{ij} = \frac{1}{2}\left(\frac{\partial \overline{u_i}}{\partial x_j} - \frac{\partial \overline{u_j}}{\partial x_i}\right) \,, \tag{4.63}$$

and obtain

$$P^k = 2\nu_t D_{ij}\left(D_{ij} + \Omega_{ij}\right) . \tag{4.64}$$

The production term is finally expressed as

$$P^k = 2\nu_t D_{ij} S_{ij} + 2\nu_t D_{ij}\Omega_{ij} = 2\nu_t D_{ij} D_{ij} . \tag{4.65}$$

The product $D_{ij}\Omega_{ij}$ vanishes as the dyadic product between a symmetric and an anti-symmetric tensor is equal to zero. The dissipation of turbulent kinetic energy is labeled with ε and it is not modeled as it is obtained from the solution of a proper transport equation. The transport equation for turbulent kinetic energy is then written as

$$\frac{\partial k}{\partial t} + \bar{\mathbf{u}} \cdot \boldsymbol{\nabla} k = \boldsymbol{\nabla} \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\boldsymbol{\nabla} k\right] + P^k - \varepsilon + \beta \mathbf{g} \cdot \mathbf{q}^r . \tag{4.66}$$

4.1.3 Transport equation for turbulent kinetic energy dissipation

Turbulent kinetic energy dissipation rate ε transport equation is needed for two main reasons: ε is an important term inside k equation as it rules the dissipation of turbulent kinetic energy at fine scales, so the solution of ε equation allows to obtain more accurate prediction of fluid flow instead of using a modeled ε. Moreover ε is chosen to model eddy kinematic viscosity by defining a characteristic time scale $\tau_u = k/\varepsilon$. The exact transport equation for turbulent kinetic energy dissipation ε, Eq_ε, is obtained by performing the following operation

$$Eq_\varepsilon := \sum_i \sum_k \overline{2\nu \frac{\partial u_i'}{\partial x_k} \cdot Eq_{ui}} . \tag{4.67}$$

With some subscript manipulation the resulting terms can be grouped to obtain the following equation

$$\begin{aligned}
\frac{\partial \varepsilon}{\partial t} + \overline{u_j}\frac{\partial \varepsilon}{\partial x_j} = & -\underbrace{2\nu\left(\overline{\frac{\partial u_i'}{\partial x_l}\frac{\partial u_k'}{\partial x_l}} + \overline{\frac{\partial u_l'}{\partial x_i}\frac{\partial u_l'}{\partial x_k}}\right)\frac{\partial \overline{u_i}}{\partial x_k}}_{\text{I}} + \\
& -\underbrace{2\nu \overline{u_k'\frac{\partial u_i'}{\partial x_j}}\frac{\partial^2 \overline{u_i}}{\partial x_j \partial x_k}}_{\text{II}} - \underbrace{2\nu \overline{\frac{\partial u_j'}{\partial x_k}\frac{\partial u_i'}{\partial x_k}\frac{\partial u_i'}{\partial x_j}}}_{\text{III}} - \underbrace{2\overline{\left(\nu\frac{\partial^2 u_i'}{\partial x_j \partial x_k}\right)^2}}_{\text{IV}} + \\
& + \underbrace{2\nu\beta g_i \overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial T'}{\partial x_k}}}_{\text{V}} - \underbrace{\nu\frac{\partial \overline{u_j'\varepsilon'}}{\partial x_j}}_{\text{VI}} - \underbrace{2\nu\frac{\partial}{\partial x_i}\overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial p'}{\partial x_k}}}_{\text{VII}} + \underbrace{\nu\frac{\partial^2 \varepsilon}{\partial x_j^2}}_{\text{VIII}} .
\end{aligned} \tag{4.68}$$

Terms I – V represent source contributions to ε transport equation. Under the assumption of isotropic turbulence, which means that out of diagonal components of $\boldsymbol{\nabla}\mathbf{u}'$ tensor are equal to zero, term I can be written as

$$2\nu\left(\overline{\frac{\partial u_i'}{\partial x_l}\frac{\partial u_k'}{\partial x_l}}+\overline{\frac{\partial u_l'}{\partial x_i}\frac{\partial u_l'}{\partial x_k}}\right)\frac{\partial \overline{u}_i}{\partial x_k}=2\varepsilon\delta_{ik}\frac{\partial \overline{u}_i}{\partial x_k}\,. \tag{4.69}$$

As a result of continuity equation, this contribution is null for incompressible flows. Term II) is related to mean velocity field second space derivatives. An estimation of its impact on ε can be performed through an order of magnitude comparison with terms III) and IV) which both depend on fluctuating velocity field gradients. Assuming that the reference length for mean flow gradients is l_0 and that l_k is the one for fluctuating velocity gradients, we obtain

$$P_{\varepsilon 2}=2\nu\overline{u_k'\frac{\partial u_i'}{\partial x_j}}\frac{\partial^2\overline{u_i}}{\partial x_j\partial x_k}\simeq\nu\frac{u_0u_k^2}{l_0^2l_k} \tag{4.70}$$

$$P_{\varepsilon 3}=2\nu\overline{\frac{\partial u_j'}{\partial x_k}\frac{\partial u_i'}{\partial x_k}\frac{\partial u_i'}{\partial x_j}}\simeq\nu\frac{u_k^3}{l_k^3} \tag{4.71}$$

$$\Gamma_\varepsilon=2\overline{\left(\nu\frac{\partial^2u_i'}{\partial x_j\partial x_k}\right)^2}\simeq\nu^2\frac{u_k^2}{l_k^4}=\nu\frac{u_k^3}{l_k^3}\frac{\nu}{u_kl_k}=\nu\frac{u_k^3}{l_k^3}\,, \tag{4.72}$$

as the Kolmogorov characteristic length based Reynolds number is equal to 1. From the ratio (4.70)/(4.71) we obtain

$$\frac{P_{\varepsilon 2}}{P_{\varepsilon 3}}=\frac{u_0}{u_k}\left(\frac{l_k}{l_0}\right)^2=Re^{-5/4}\,, \tag{4.73}$$

where the last equality has been obtained using (4.13) and (4.14). Term II) is then considered negligible as it is $Re^{5/4}$ times smaller than $P_{\varepsilon 3}$. Terms III) and IV) represent then the source and dissipation contribution to ε transport equation and are modeled using k equation production and dissipation terms, multiplied by the inverse of characteristic time scale $\tau_u = k/\varepsilon$. The buoyancy related source term V) is labeled as $P_{\varepsilon b}$. Diffusive contributions are represented by terms VI) – VIII), in particular and VI) – VII) are turbulent contributions, related to velocity and pressure fluctuations, while VIII) is the molecular diffusion of ε. With ε' we label the instantaneous dissipation rate of turbulent kinetic energy, i.e. $\varepsilon = \overline{\varepsilon'}$. The turbulent diffusion terms are modeled using a SGDH, leading to

$$-\nu\left(\overline{u_i'\varepsilon'}+2\overline{\frac{\partial u_i'}{\partial x_k}\frac{\partial p'}{\partial x_k}}\right)=\frac{\nu_t}{\sigma_\varepsilon}\frac{\partial\varepsilon}{\partial x_i} \tag{4.74}$$

The modeled transport equation for turbulent kinetic energy dissipation rate is then written as

$$\frac{\partial \varepsilon}{\partial t} + \overline{\mathbf{u}} \cdot \boldsymbol{\nabla} \varepsilon = \boldsymbol{\nabla} \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon} \right) \boldsymbol{\nabla} \varepsilon \right] + \frac{\varepsilon}{k} \left(c_{\varepsilon 1} P^k - c_{\varepsilon 2} \varepsilon \right) + P_{\varepsilon, b} \,, \tag{4.75}$$

where σ_ε, $c_{\varepsilon 1}$ and $c_{\varepsilon 2}$ are model constants that will be defined in following sections. Once that the expression for τ_ν is provided, the closure of the dynamic RANS system of equations, for non buoyant flows, is obtained.

4.1.4 Transport equation for mean squared temperature fluctuations

During the years many closure strategies have been developed to model Reynolds stresses, with an increasing level of complexity. Starting from zero equation models, like Prandtl mixing length model where an eddy viscosity is introduced and algebraically defined, n equation models have been developed, where n additional transport equations are solved. Two equations are introduced in the case of a k - ε model like the one described above, or six for a three dimensional simulation with Reynolds Stress Model where a transport equation is solved for each component of $\boldsymbol{\tau}^r$. Thermal turbulence closure issue is widely addressed with Reynolds analogy, where a similarity between velocity and temperature field is assumed. A constant and uniform turbulent Prandtl number $Pr_t \simeq 0.9$ is then introduced in order to calculate eddy thermal diffusivity α_t as $\alpha_t = \nu_t / Pr_t$. While this approach allows to obtain accurate predictions of turbulent heat transfer for fluids with a molecular Prandtl number $Pr \simeq 1$, non negligible overestimation of heat exchange is obtained when simulating low Prandtl number fluids. Transport equations for mean squared temperature fluctuations k_θ and their dissipation rate ε_θ are obtained in order to model eddy thermal diffusivity α_t as symbolically described in (4.24), which means by using a time scale τ_α that depends on both dynamical and thermal turbulence characteristic time scales.

Similarly to the case of velocity fluctuations, we introduce the balance equation for instantaneous temperature field, decomposed into its mean and fluctuating value

$$\frac{\partial \overline{T}}{\partial t} + \frac{\partial T'}{\partial t} + \overline{u_j} \frac{\partial \overline{T}}{\partial x_j} + u_j' \frac{\partial \overline{T}}{\partial x_j} + \overline{u_j} \frac{\partial T'}{\partial x_j} + u_j' \frac{\partial T'}{\partial x_j} = \boldsymbol{\nabla} \cdot \alpha \boldsymbol{\nabla} \overline{T} + \boldsymbol{\nabla} \cdot \alpha \boldsymbol{\nabla} T' \,. \tag{4.76}$$

The transport equation for fluctuating temperature field is obtained as subtraction of (4.21) from (4.76)

$$\frac{\partial T'}{\partial t} + u_j' \frac{\partial \overline{T}}{\partial x_j} + \overline{u_j} \frac{\partial T'}{\partial x_j} - \overline{u_j' \frac{\partial T'}{\partial x_j}} = + \boldsymbol{\nabla} \cdot \alpha \boldsymbol{\nabla} T' \,. \tag{4.77}$$

In the following we refer to $Eq_{T'}$ as the transport equation for T' (4.77). Exact transport equation for mean squared temperature fluctuations $Eq_{k\theta}$ is obtained as

$$Eq_{k\theta} := \overline{2T' \cdot Eq_{T'}}\,. \tag{4.78}$$

The resulting equation is

$$\frac{\partial k_\theta}{\partial t} + \overline{u_j}\frac{\partial k_\theta}{\partial x_j} = \underbrace{\alpha\frac{\partial^2 k_\theta}{\partial x_j^2}}_{\text{I}} \underbrace{-\frac{1}{2}\frac{\partial \overline{u'_j T'^2}}{\partial x_j}}_{\text{II}} - \underbrace{\overline{u'_j T'}\frac{\partial \overline{T}}{\partial x_j}}_{\text{III}} - \underbrace{\alpha\overline{\frac{\partial T'}{\partial x_j}\frac{\partial T'}{\partial x_j}}}_{\text{IV}}\,. \tag{4.79}$$

On the left hand side of (4.79) the time derivative and convective term are shown. On the right hand side I) represents the molecular diffusion term. Turbulent diffusion contribution II) is modeled using SGDH, leading to the following expression

$$-\frac{1}{2}\overline{u'_j T'} = \frac{\alpha_t}{\sigma_{k\theta}}\frac{\partial k_\theta}{\partial x_j}\,. \tag{4.80}$$

Production and dissipation terms III) and IV) will be labeled with $P^\theta = \mathbf{q}^r \cdot \boldsymbol{\nabla} T$ and ε_θ. The final expression of the modeled k_θ transport equation is

$$\frac{\partial k_\theta}{\partial t} + \overline{\mathbf{u}} \cdot \boldsymbol{\nabla} k_\theta = \boldsymbol{\nabla} \cdot \left[\left(\alpha + \frac{\alpha_t}{\sigma_{k\theta}}\right)\boldsymbol{\nabla} k_\theta\right] + P^\theta - \varepsilon_\theta\,. \tag{4.81}$$

In the present formulation of the four parameter turbulence model, for mean squared temperature fluctuations dissipation rate ε_θ a transport equation is derived, as shown in the following. In literature thermal turbulence models can be found involving a modeled ε_θ [60]. The thermal to dynamic characteristic time scale ratio R is introduced and defined as

$$R := \frac{\tau_\theta}{\tau_u} = \frac{k_\theta}{\varepsilon_\theta} \cdot \frac{\varepsilon}{k}\,. \tag{4.82}$$

If a value of R is fixed, then ε_θ is simply calculated with algebraic relation (4.82), circumventing the problem of modeling a proper transport equation. On the other hand, the validity of assuming $R = R_{fixed}$ should be questioned case by case, depending on the simulated geometry and on fluid properties.

4.1.5 Transport equation for mean squared temperature fluctuations dissipation

The balance equation $Eq_{\varepsilon\theta}$ governing the evolution of mean squared temperature fluctuations dissipation rate is obtained as

$$Eq_{\varepsilon\theta} := \sum_k \overline{2\alpha\frac{\partial T'}{\partial x_k}\frac{\partial Eq_{T'}}{\partial x_k}}\,. \tag{4.83}$$

After some algebraic manipulation the following equation can be written

$$\begin{aligned}\frac{\partial \varepsilon_\theta}{\partial t}+\overline{u_j}\frac{\partial \varepsilon_\theta}{\partial x_j} = & \underbrace{-2\alpha\left(\overline{\frac{\partial T'}{\partial x_j}\frac{\partial u_k}{\partial x_j}}\right)\frac{\partial \overline{T}}{\partial x_k}}_{\text{I}} \underbrace{-2\alpha\left(\overline{\frac{\partial T'}{\partial x_i}\frac{\partial T'}{\partial x_k}}\right)\frac{\partial \overline{u_i}}{\partial x_k}}_{\text{II}} + \\ & \underbrace{-2\alpha \overline{u_k\frac{\partial T'}{\partial x_j}}\frac{\partial^2 \overline{T}}{\partial x_k \partial x_j}}_{\text{III}} \underbrace{-2\alpha\overline{\frac{\partial u'_j}{\partial x_k}\frac{\partial T'}{\partial x_k}\frac{\partial T'}{\partial x_j}}}_{\text{IV}} \underbrace{-2\overline{\left(\alpha\frac{\partial^2 \overline{T'}}{\partial x_k \partial x_j}\right)}}_{\text{V}} + \\ & \underbrace{+\alpha\frac{\partial^2 \varepsilon_\theta}{\partial x_j^2}}_{\text{VI}} \underbrace{-\frac{\partial \overline{\varepsilon'_\theta u'_k}}{\partial x_k}}_{\text{VII}}\,. \end{aligned} \tag{4.84}$$

Terms I) – IV) represents different production terms for ε_θ transport equation. The first, labeled as $P_{\varepsilon\theta 1}$, is related to interaction with mean temperature gradient, while term II), labeled as $P_{\varepsilon\theta 2}$, depend on the interaction of mean velocity gradient. Term III) is a production term that is normally considered negligible. Terms IV) and V) represent respectively the main production $P_{\varepsilon\theta}$, depending only on velocity and temperature fluctuations, and dissipation $\chi_{\varepsilon\theta}$ contributions to ε_θ transport equation. The remaining terms are the molecular and turbulent diffusion contributions. The latter is modeled using SGDH, so

$$\overline{\varepsilon'_\theta u'_k} = -\frac{\alpha_t}{\sigma_{\varepsilon_\theta}}\frac{\partial \varepsilon_\theta}{\partial x_k}\,. \tag{4.85}$$

The modeled transport equation for ε_θ is then written as

$$\frac{\partial \varepsilon_\theta}{\partial t} + \overline{\mathbf{u}}\cdot\boldsymbol{\nabla}\varepsilon_\theta = \boldsymbol{\nabla}\cdot\left[\left(\alpha+\frac{\alpha_t}{\sigma_{\varepsilon_\theta}}\right)\boldsymbol{\nabla}\varepsilon_\theta\right] + P_{\varepsilon\theta 1} + P_{\varepsilon\theta 2} + P_{\varepsilon_\theta} - \chi_{\varepsilon\theta}\,. \tag{4.86}$$

We see that the final form of transport equations (4.66), (4.75), (4.81) and (4.86) are very similar and can be expressed in a general form as

$$\mathcal{T}_a + \mathcal{C}_a - \mathcal{D}_a - \mathcal{P}_a + \mathcal{S}_a = 0\,, \tag{4.87}$$

where, for the given variable a, the modeled transport equation consists of a balance between a temporal-derivative term $\mathcal{T}_a$, a convective term $\mathcal{C}_a$, a diffusion term $\mathcal{D}_a$ due to both molecular and turbulent diffusion, a production term $\mathcal{P}_a$ and a dissipative term $\mathcal{S}_a$.

As a completion we finally derive the transport equation Eq_{qi} for turbulent heat flux components $\overline{u'_i T'}$. We use both instantaneous velocity and temperature fluctuations transport equations as follows

$$Eq_{qi} := \overline{T' Eq_{ui} \;+\; u'_i Eq_{T'}}\,. \tag{4.88}$$

After some algebraic manipulation and applying incompressibility constrain, the final equation can be written as follows [50]

$$\begin{aligned}\frac{\partial \overline{u_i'T'}}{\partial t} + \frac{\partial \overline{u_k}\,\overline{u_i'T'}}{\partial x_k} = &- \underbrace{\left(\overline{u_i'u_j'}\frac{\partial \overline{T}}{\partial x_j} + \overline{u_j'T'}\frac{\partial \overline{u_i}}{\partial x_j}\right)}_{\text{I}} - \underbrace{\frac{1}{\rho}\overline{T'\frac{\partial p'}{\partial x_i}}}_{\text{II}} + \\ &- \underbrace{\frac{\partial \overline{u_j'u_i'T'}}{\partial x_j}}_{\text{III}} + \underbrace{(\nu+\alpha)\,\frac{\partial^2 \overline{u_i'T'}}{\partial x_j \partial x_j}}_{\text{IV}} - \underbrace{(\nu+\alpha)\,\frac{\partial u_i'}{\partial x_j}\frac{\partial T'}{\partial x_j}}_{\text{V}} - \underbrace{g_i\beta\overline{T'^2}}_{\text{VI}}\ .\end{aligned} \tag{4.89}$$

On the right hand side of (4.89), term I) is a production term due to interaction of Reynolds stresses and turbulent heat flux components with mean temperature and velocity gradients respectively. Term II) is a dissipation term due to an interaction between temperature fluctuations and fluctuating pressure gradient components [61]. Turbulent and molecular diffusion contributions are represented by terms III) and IV) respectively. The dissipation term of turbulent heat flux components $\varepsilon_{\theta i}$ is V), while VI) acts as a source term due to buoyancy. Most commonly (4.89) is used to derive an algebraic expression for turbulent heat flux components [62, 63, 64] by neglecting convective and diffusive contributions (both molecular and turbulent one), term II and also dissipation $\varepsilon_{\theta i}$. The time derivative is approximated as

$$\frac{\partial \overline{u_i'T'}}{\partial t} \simeq \frac{\overline{u_i'T'}}{c_\theta \tau_\theta}\,, \tag{4.90}$$

so it is assumed that turbulent heat flux components change on a characteristic thermal time scale $c_\theta\tau_\theta$. The algebraic equation is then written as

$$\overline{u_i'T'} = -c_\theta\tau_\theta\left(c_1\overline{u_i'u_j'}\frac{\partial \overline{T}}{\partial x_i} + c_2\overline{u_j'T'}\frac{\partial \overline{u_i}}{\partial x_j} + c_3\beta g_i\overline{T'^2}\right)\ . \tag{4.91}$$

The characteristic thermal time scale τ_θ is usually expressed as $\tau_\theta = R\tau_u$, where R is thermal to mechanical characteristic time scale ratio and τ_u is the mechanical characteristic time scale $\tau_u = k/\varepsilon$.

4.2 Calibration of the turbulence model

In previous section the exact transport equations for the system (k, ε, k_θ, ε_θ) have been derived starting from the equations for fluctuating velocity and temperature fields. In order to get a closure of the system, many terms need to be modeled and boundary conditions must be specified for all solved turbulence variables. It is indeed very important to accurately reproduce the correct behavior of turbulence variables in the near wall region, where viscous forces are more important and dissipation of turbulent kinetic energy takes place. In the present section exact boundary conditions are provided for all turbulence variables. Moreover all remaining terms of (4.66), (4.75), (4.81) and (4.86) are modeled.

4.2.1 Near wall behavior of turbulence variables

Exact near wall behavior expressions for turbulence variables can be obtained by performing Taylor series expansions in near wall region of mean and fluctuating velocity and temperature. In order to do this we refer to y as the wall distance From no slip boundary conditions we obtain

$$\begin{aligned} &\overline{u} = 0 \qquad \overline{v} = 0 \qquad \overline{w} = 0 \\ &u' = 0 \qquad v' = 0 \qquad w' = 0\,, \end{aligned} \tag{4.92}$$

while from incompressibility constrain the following equations must be satisfied.

$$\begin{aligned} &\frac{\partial \overline{u}}{\partial x} + \frac{\partial \overline{v}}{\partial y} + \frac{\partial \overline{w}}{\partial z} = 0 \\ &\frac{\partial}{\partial x} u' + \frac{\partial}{\partial y} v' + \frac{\partial}{\partial z} w' = 0\,. \end{aligned} \tag{4.93}$$

We refer to $\overline{v}$ and v' as mean and fluctuating velocity component along wall normal direction. As no slip boundary condition is valid along each wall position then

$$\begin{aligned} &\left.\frac{\partial \overline{u}}{\partial x}\right|_w = 0 \qquad \left.\frac{\partial \overline{w}}{\partial z}\right|_w = 0\,, \\ &\left.\frac{\partial u'}{\partial x}\right|_w = 0 \qquad \left.\frac{\partial w'}{\partial z}\right|_w = 0\,. \end{aligned} \tag{4.94}$$

From (4.93) and (4.94) we obtain

$$\left.\frac{\partial \overline{v}}{\partial y}\right|_w = 0 \qquad \left.\frac{\partial v'}{\partial y}\right|_w = 0\,. \tag{4.95}$$

We recall that the value of a generic variable $\mathcal{A}(x)$ can be calculated from a Taylor series expansion near position $x = \tilde{x}$ as

$$\mathcal{A}(x) = \mathcal{A}(\tilde{x}) + \sum_{n=1}^{\infty} \frac{1}{n!} \frac{\partial}{\partial x^n}^{n} \mathcal{A}\bigg|_{\tilde{x}} (x - \tilde{x})^n\,. \tag{4.96}$$

Near wall expansion approximations of mean and fluctuating velocity components are then the following

$$\begin{aligned}
\overline{u} &= A_1 y + A_2 y^2 + A_3 y^3 + \mathcal{O}(y^4)\,,\\
\overline{v} &= \qquad\quad\; B_2 y^2 + B_3 y^3 + \mathcal{O}(y^4)\,,\\
\overline{w} &= C_1 y + C_2 y^2 + C_3 y^3 + \mathcal{O}(y^4)\,,\\
u' &= a_1 y + a_2 y^2 + a_3 y^3 + \mathcal{O}(y^4)\,,\\
v' &= \qquad\quad\; b_2 y^2 + b_3 y^3 + \mathcal{O}(y^4)\,,\\
w' &= c_1 y + c_2 y^2 + c_3 y^3 + \mathcal{O}(y^4)\,,
\end{aligned} \tag{4.97}$$

where no slip boundary condition and incompressibility constrain have been taken into account. Near wall behavior of k and ε is obtained by recalling their constitutive expression, i.e.

$$k := \frac{1}{2}\sum_i \overline{u_i^2}\,, \quad \varepsilon := \nu \sum_{i,j} \overline{\left(\frac{\partial u_i'}{\partial x_j}\right)^2}\,. \tag{4.98}$$

Expressions for k and ε are truncated at first term, obtaining

$$k_{nw} = \frac{1}{2}(a_1^2 + c_1^2)y^2\,, \tag{4.99}$$

$$\varepsilon_{nw} = \nu(a_1^2 + c_1^2)\,, \tag{4.100}$$

where the subscript nw stands for near wall behavior. Turbulent kinetic energy has a null value along the wall and increase along wall normal direction as $\propto y^2$, while ε reaches a non null value at the wall. It is observed that $\varepsilon_{nw} = 2\nu k_{nw}/y^2$ and that coefficients a_1 and c_1 are related to velocity fluctuations and cannot be estimated a priori. As a consequence an exact Dirichlet boundary condition cannot be imposed on ε and the wall value must be estimated from k_{nw}. As it regards temperature field, similar expressions can be obtained

$$\begin{aligned}
\overline{T} &= T_w + H_1 y + H_2 y^2 + H_3 y^3 + \mathcal{O}(y^4)\,,\\
T' &= T_w' + h_1 y + h_2 y^2 + h_3 y^3 + \mathcal{O}(y^4)\,.
\end{aligned} \tag{4.101}$$

In the case of a constant wall temperature boundary condition, namely $T = T_w$, the condition must be fulfilled by both instantaneous and mean temperature values, so that $T' = 0$ along the wall. On the other hand, if a constant heat flux is applied then temperature fluctuations can either be considered null or not null. The constitutive expressions for mean squared temperature fluctuations k_θ and their dissipation rate ε_θ are

$$k_\theta := \frac{1}{2}\overline{T'^2}\,, \quad \varepsilon_\theta := \alpha \sum_i \overline{\left(\frac{\partial T'}{\partial x_i}\right)^2}\,. \tag{4.102}$$

The following near wall behavior relations can be obtained

$$k_{\theta,nw} = \frac{1}{2}T'^2_w + \frac{1}{2}(h_1^2 + 2T'_w h_2^2)y^2\,, \tag{4.103}$$

$$\varepsilon_{\theta,nw} = \alpha(h_1^2 + 2T'_w h_2^2)\,, \tag{4.104}$$

where a smooth change of k_θ near the wall is assumed, i.e. $\partial k_\theta/\partial y = 0$ and $\varepsilon_{\theta nw}$ is obtained from steady transport equation for k_θ with negligible convective and production terms, as described in [65]. For thermal turbulence system of equations we distinguish the following boundary conditions for the case of constant heat flux:

- Mixed type (MX): zero temperature fluctuations along the heated wall, then

  $$k_{\theta,nw} = \frac{1}{2}h_1^2 y^2\,, \quad \varepsilon_{\theta,nw} = \alpha h_1^2\,, \tag{4.105}$$

  As for the dynamical case, k_θ increases proportionally to the squared of wall distance while ε_θ has a non null value on the heated wall. This value depend on temperature fluctuations, it cannot be imposed exactly and so it is iteratively imposed as $\varepsilon_\theta = 2\alpha k_\theta/y^2$.

- Constant Heat Flux (CHF): temperature fluctuations are not considered null on the heated wall. Their value cannot be estimated *a priori*, so the following boundary conditions are set

  $$\left.\frac{\partial k_\theta}{\partial y}\right|_w = 0\,, \quad \left.\frac{\partial \varepsilon_\theta}{\partial y}\right|_w = 0\,. \tag{4.106}$$

In previous section eddy kinematic viscosity ν_t and eddy thermal diffusivity α_t were introduced to model Reynolds stresses and turbulent heat flux

$$-\overline{u'_i u'_j} = \nu_t\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right)\,, \quad -\overline{u'_i T'} = \alpha_t \frac{\partial \overline{T}}{\partial x_i}\,. \tag{4.107}$$

With he above expressions near wall behavior of ν_t and α_t can be calculated. For eddy viscosity we obtain

$$\nu_t = -\frac{\overline{u'v'}}{\dfrac{\partial \overline{u}}{\partial y} + \dfrac{\partial \overline{v}}{\partial x}} = -\frac{a_1 b_2}{A_1}y^3\,, \tag{4.108}$$

so ν_t grows with the third power of wall distance. For eddy thermal diffusivity

$$\alpha_t = -\frac{\overline{v'T'}}{\dfrac{\partial \overline{T}}{\partial y}} = -\frac{(b_2 y^2 + b_3 y^3)(T'_w + h_1 y)}{H_1} = \begin{cases} \propto y^2 & \text{CHF} \\ \propto y^3 & \text{MX} \end{cases} \tag{4.109}$$

We see that near wall behavior of eddy thermal diffusivity depends on applied boundary conditions for the thermal turbulence model. A proper model of these terms is then required to reproduce the correct behavior of both ν_t and α_t.

Before proceeding with the turbulence model calibration, we introduce the turbulent Reynolds number $R_t = k^2/(\nu\varepsilon)$, that is representative of the energy-containing turbulent eddies and the non-dimensional length R_d which is calculated as the ratio between the wall distance δ and the Kolmogorov characteristic length scale $l_k = (\nu^3/\varepsilon)^{1/4}$ [66]. These quantities will be widely used to model many transport equation terms. As we have already introduced, characteristic dynamical and thermal time scales are respectively computed as $\tau_u = k/\varepsilon$ and $\tau_\theta = k_\theta/\varepsilon_\theta$. Model calibration is performed in order to reproduce some limiting behaviors that can be encountered in special cases.

The four parameter turbulence model is then made of the following system of equations

$$\frac{\partial k}{\partial t} + \bar{\mathbf{u}} \cdot \boldsymbol{\nabla} k = \boldsymbol{\nabla} \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_k}\right) \boldsymbol{\nabla} k\right] + P^k - \varepsilon + \beta \mathbf{g} \cdot \mathbf{q}^r \,, \tag{4.110}$$

$$\frac{\partial \varepsilon}{\partial t} + \bar{\mathbf{u}} \cdot \boldsymbol{\nabla} \varepsilon = \boldsymbol{\nabla} \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right) \boldsymbol{\nabla} \varepsilon\right] + \frac{\varepsilon}{k} \left(c_{\varepsilon 1} P^k - c_{\varepsilon 2} f_\varepsilon \varepsilon\right) + P_{\varepsilon b} \,, \tag{4.111}$$

$$\frac{\partial k_\theta}{\partial t} + \bar{\mathbf{u}} \cdot \boldsymbol{\nabla} k_\theta = \boldsymbol{\nabla} \cdot \left[\left(\alpha + \frac{\alpha_t}{\sigma_{k\theta}}\right) \boldsymbol{\nabla} k_\theta\right] + P^\theta - \varepsilon_\theta \,, \tag{4.112}$$

$$\begin{aligned} \frac{\partial \varepsilon_\theta}{\partial t} + \bar{\mathbf{u}} \cdot \boldsymbol{\nabla} \varepsilon_\theta = \boldsymbol{\nabla} \cdot \left[\left(\alpha + \frac{\alpha_t}{\sigma_{\varepsilon_\theta}}\right) \boldsymbol{\nabla} \varepsilon_\theta\right] + \\ + P_{\varepsilon\theta 1} + P_{\varepsilon\theta 2} + P_{\varepsilon_\theta} - \chi_{\varepsilon\theta} \,. \end{aligned} \tag{4.113}$$

Reynolds stresses and turbulent heat flux components have been modeled as

$$-\overline{u_i' u_j'} = \nu_t D_{ij} + \frac{2}{3}\delta_{ij} k \,, \quad -\overline{u_i' T'} = \alpha_t \frac{\partial \overline{T}}{\partial x_i} \,, \tag{4.114}$$

leading to the following forms of production terms

$$P^k = 2\nu_t D_{ij} D_{ij} \,, \quad P^\theta = \alpha_t \boldsymbol{\nabla}\overline{T} \cdot \boldsymbol{\nabla}\overline{T} \,, \quad P_b = \beta \mathbf{g} \cdot \mathbf{q}^r = -\beta \alpha_t \mathbf{g} \cdot \boldsymbol{\nabla}\overline{T} \,. \tag{4.115}$$

Production and dissipation terms in ε equation have been modeled using P^k and ε itself, i.e. source terms of k equation, multiplied by the inverse of characteristic time scale τ_u, two constants $c_{\varepsilon 1}$ and $c_{\varepsilon 2}$ and function f_ε. In a similar fashion $P_{\varepsilon b} = P_b/\tau_u$. Source terms of k_θ transport equation do not require further modeling, while for ε_θ equation we follow the approach used in [67]

$$P_{\varepsilon\theta 1} + P_{\varepsilon\theta 2} + P_{\varepsilon_\theta} - \chi_{\varepsilon\theta} = \frac{c_{p1}}{\tau_\theta} P^\theta + c_{p2} \frac{\varepsilon_\theta}{k} P^k - c_{d1} \frac{\varepsilon_\theta^2}{k_\theta} - c_{d2} \frac{\varepsilon_\theta \varepsilon}{k} \,. \tag{4.116}$$

Mechanical source terms are then introduced, each one multiplied by a model constant c and function f and by the ratio ε_θ/k in order to match their physical dimension with other terms. Source terms of k_θ equation are multiplied by the inverse of characteristic thermal time scale τ_θ.

From a near wall analysis of (4.111), in steady state condition, the convective term, the turbulent diffusion and the production terms can be neglected, obtaining the following equation

$$\nu\nabla^2\varepsilon = -c_{\varepsilon 2} f_\varepsilon \frac{\varepsilon^2}{k} \, . \tag{4.117}$$

Diffusive term on the left hand side is of order y^0, while the dissipation contribution on the right hand side is of order $f_\varepsilon y^{-2}$. It is then required f_ε to be of order y^2. The requirement is fulfilled by using the following expression

$$f_\varepsilon = \left(1 - e^{-R_d/3.1}\right)^2 \left(1 - 0.3e^{-R_t^2/42.25}\right) \, . \tag{4.118}$$

The value of constant $c_{\varepsilon 2}$ is inferred by examining (4.110) – (4.111) in the case of homogeneous turbulence. If x is the main flow direction, the decay law of homogeneous turbulence is given by $k \propto x^{-n}$, where n varies from $1 - 1.25$ in the first period and 2.5 in the final period. The system of equations states

$$\overline{u}\frac{dk}{dx} = -\varepsilon \, , \quad \rightarrow \quad \varepsilon = \overline{u}\, n\, x^{-(n+1)} \, , \tag{4.119}$$

$$\overline{u}\frac{d\varepsilon}{dx} = -c_{\varepsilon 2} f_\varepsilon \frac{\varepsilon^2}{k} \, , \tag{4.120}$$

leading to

$$c_{\varepsilon 2} f\varepsilon = \frac{n+1}{n} \, . \tag{4.121}$$

At first period, $n \simeq 1.1$ and $f\varepsilon = 1$, so $c_{\varepsilon 2} = 1.91$. At final period $n = 2.5$ and $f_\varepsilon = 0.7$ as $e^{-R_t^2/42.25} \simeq 1$. Constant $c_{\varepsilon 1}$ value is obtained by applying (4.110) - (4.111) in the constant stress layer [68]. In this region velocity profile exhibits a logarithmic behavior, so that it's gradient component $\partial u/\partial y$ is equal to $u_\tau/(\kappa y)$, with u_τ friction velocity. Moreover, turbulence production P^k is balanced by dissipation ε, so that turbulent kinetic energy has a constant value $k = u_\tau^2/C_\mu^{1/2}$. Convective terms can be neglected and, from (4.111), the following relation is obtained

$$c_{\varepsilon 1} = c_{\varepsilon 2} - \frac{\kappa^2}{\sigma_\varepsilon C_\mu^{1/2}} \, . \tag{4.122}$$

Constant σ_ε is assumed $\sigma_\varepsilon = \sigma_k = 1.4$, leading to $c_{\varepsilon 1} = 1.5$.

For the thermal turbulence model no further modeling is required in (4.112), while for ε_θ transport equation some constants need to be calibrated. A similar approach can be used by considering system (4.110) – (4.113) in the condition of homogeneous decaying turbulence and of constant stress - flux layer [69, 70]. In the first case the system of equations becomes

$$\overline{u}\frac{dk}{dx} = -\varepsilon\,, \tag{4.123}$$

$$\overline{u}\frac{d\varepsilon}{dx} = -c_{\varepsilon 2} f_\varepsilon \frac{\varepsilon^2}{k}\,, \tag{4.124}$$

$$\overline{u}\frac{dk_\theta}{dx} = -\varepsilon_\theta\,, \tag{4.125}$$

$$\overline{u}\frac{d\varepsilon_\theta}{dx} = -\frac{1}{2}c_{d1}\frac{\varepsilon_\theta^2}{k_\theta} - c_{d2}\frac{\varepsilon_\theta \varepsilon}{k}\,, \tag{4.126}$$

where the coefficient $1/2$ appears in front of thermal dissipation term of ε_θ as the original equation in [69] was obtained considering mean squared temperature fluctuations $\overline{T'^2}$ as state variable instead of $k_\theta = 0.5\overline{T'^2}$. In this flow condition it is observed that thermal to mechanical time scale ratio R has a constant value $R = 0.5$ and does not change along flow direction. Equation (4.124) can be rewritten by setting $\varepsilon = R\varepsilon_\theta k/k_\theta$ in the time derivative term. After substitution of (4.123) and (4.125) the two following equations are obtained

$$\overline{u}\frac{d\varepsilon_\theta}{dx} = \frac{\varepsilon\varepsilon_\theta}{k}(1 - c_{\varepsilon 2} f_\varepsilon) - \frac{\varepsilon_\theta^2}{k_\theta}\,, \tag{4.127}$$

$$\overline{u}\frac{d\varepsilon_\theta}{dx} = -2c_{d1}\frac{\varepsilon_\theta^2}{k_\theta} - c_{d2}\frac{\varepsilon_\theta \varepsilon}{k}\,, \tag{4.128}$$

leading to

$$c_{d2} = c_{\varepsilon 2} f_\varepsilon - 1\,, \qquad c_{d1} = 2\,. \tag{4.129}$$

We set $c_{d2} = \tilde{c}_{d2} f_{d2}$, where $\tilde{c}_{d2}$ is a constant and f_{d2} a function. Equation (4.129) is also valid for the initial period, so $f_\varepsilon = f_{d2} = 1$ and $\tilde{c}_{d2} = c_{\varepsilon 2} - 1 = 0.9$. In the latter case the following assumptions are made: production terms of k and k_θ are balanced by dissipative terms ε and ε_θ, both mean velocity and temperature exhibit a logarithmic behavior, namely $\overline{u} = u_\tau \ln(y^+)/\kappa$ and $\overline{T} = T_\tau \ln(y^+)/\kappa$, and eddy thermal diffusivity α_t is considered proportional to ν_t through a constant turbulent Prandtl number Pr_t. In this region $\alpha_t \gg \alpha$. From steady version of (4.113), when the convective term is negligible, we have

$$\frac{\partial}{\partial y}\left[\frac{\alpha_t}{\sigma_{\varepsilon_\theta}}\frac{\partial \varepsilon_\theta}{\partial y}\right] = \frac{c_{p1}}{2}\frac{\varepsilon_\theta}{k_\theta}P^\theta + c_{p2}\frac{\varepsilon_\theta}{k}P^k - \frac{c_{d1}}{2}\frac{\varepsilon_\theta^2}{k_\theta} - \tilde{c}_{d2}\frac{\varepsilon\varepsilon_\theta}{k}\,. \tag{4.130}$$

$c_{\varepsilon 1}$	$c_{\varepsilon 2}$	C_μ	σ_k	σ_ε	$\sigma_{k\theta}$	$\sigma_{\varepsilon\theta}$	c_{d1}	c_{p1}	c_{d2}	c_{p2}
1.5	1.9	0.09	1.4	1.4	1.4	1.4	1.1	1.025	1.9	0.9

Table 4.1: Alternative values of the model constants for (4.110) and (4.113).

With the above mentioned assumptions and by expressing k_θ as a function of R, the following relation is obtained

$$\frac{\kappa^2}{\sigma_{\varepsilon_\theta} Pr_t C_\mu^{1/2}} = \tilde{c}_{d2} - c_{p2} + \frac{c_{d1} - c_{p1}}{2R} \,. \tag{4.131}$$

Coefficient c_{p1} is taken equal to 1.9 [70], and $c_{p2} = 0.733$ is calculated using (4.131) with $R = 0.5$, $\sigma_{k\theta} = \sigma_{\varepsilon_\theta} = 1.4$ and $Pr_t = 1.5$. The above set of coefficient is obtained by considering reference cases, i.e. decay of homogeneous turbulence and constant stress-flux region, and some assumptions as logarithmic behavior of mean temperature profile and proportionality between α_t and ν_t. These assumptions can be too strong. Mean temperature field can exhibit a logarithmic behavior, but differently from the velocity field, where linear and logarithmic behaviors are encountered in well determined y^+ ranges, the non dimensional reference length is $Pr\ y^+$ instead of y^+. For turbulent flows with a Prandtl number close to 1 temperature and velocity logarithmic behaviors can occur in the same region, but as Pr decreases the two logarithmic layers occur at different distances from the wall. Good agreement with reference data has been obtained also with a different set of constant values, namely the ones reported in Table 4.1 [71, 72, 73].

4.3 Models for eddy kinematic viscosity and eddy thermal diffusivity

In previous sections we introduced variables k, ε, k_θ and ε_θ as key parameters for modeling eddy kinematic viscosity and eddy thermal diffusivity. In general ν_t and α_t are calculated as functions of k and two local characteristic time scales τ_{lu} and $\tau_{l\theta}$, the first for the dynamical turbulence and the latter for the thermal turbulence,

$$\nu_t = C_\mu k \tau_{lu} \,, \qquad \alpha_t = C_\theta k \tau_{l\theta} \,. \tag{4.132}$$

The local characteristic dynamical time scale τ_{lu} is modeled as sum of different contributions [10, 74, 70, 75]

$$\tau_{lu} = \left(f_{1\mu} A_{1\mu} + f_{2\mu} A_{2\mu} \right) \,, \tag{4.133}$$

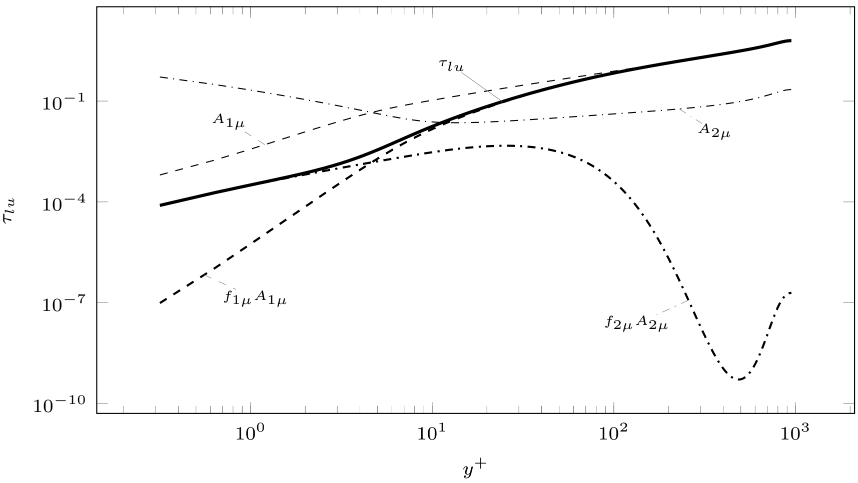


Figure 4.6: Representation of local characteristic dynamic time scale τ_{lu} for a fully developed turbulent flow with $Re_\tau = 950$. The time scale is plotted against the non-dimensional wall distance y^+ together with the terms $f_{1\mu}A_{1\mu}$ and $f_{2\mu}A_{2\mu}$ used to model τ_{lu}.

where

$$f_{1\mu} = (1 - e^{-R_d/14})^2 \,, \tag{4.134}$$
$$A_{1\mu} = \tau_u \,, \tag{4.135}$$
$$f_{2\mu} = f_{1\mu} e^{-2.5\times 10^{-5} R_t^2} \,, \tag{4.136}$$
$$A_{2\mu} = \tau_u \frac{5}{R_t^{3/4}} \,. \tag{4.137}$$

The term $A_{1\mu}$ is the bulk dynamical time scale while $A_{2\mu}$ is the correction term for the near wall behavior. The function $f_{2\mu}$ decreases rapidly as the wall distance increases, so the term $f_{2\mu}A_{2\mu}$ is significant only in the near wall region, as can be seen in Figure 4.6. In this Figure the time scale τ_{lu} and the terms $f_{1\mu}A_{1\mu}$ and $f_{2\mu}A_{2\mu}$ are reported as functions of the non-dimensional distance from the wall y^+ for a test case with $Re_\tau = 950$. In the near wall region, $y^+ < 3$, the value of τ_{lu} is almost equal to $f_{2\mu}A_{2\mu}$ while far from the wall, at $y^+ > 20$, the value of the local dynamical time scale is equal to the bulk term $f_{1\mu}A_{1\mu}$. With this modeling we can correctly reproduce the near-wall behavior of the eddy viscosity $\nu_t \propto \delta^3$.

The time scale $\tau_{l\theta}$ is modeled as [10, 74, 70, 11, 12, 75]

$$\tau_{l\theta} = (f_{1\theta}B_{1\theta} + f_{2\theta}B_{2\theta}) \,, \tag{4.138}$$

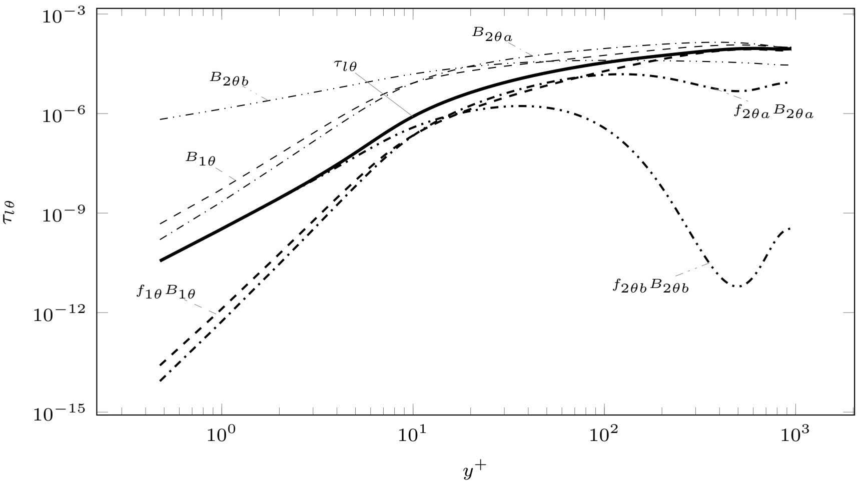


Figure 4.7: Representation of the local characteristic time scale $\tau_{l\theta}$, plotted against the non-dimensional wall distance y^+, together with the various terms used to model $\tau_{l\theta}$. The time scale is for the case of fully developed flow with $Re_\tau = 950$ in a plane channel.

where

$$f_{1\theta} = \left(1 - e^{\sqrt{Pr}R_d/19}\right)\left(1 - e^{R_d/14}\right) , \tag{4.139}$$

$$B_{1\theta} = \frac{\tau_u}{Pr_{t,\infty}} , \tag{4.140}$$

$$f_{2\theta}B_{2\theta} = \tau_u \left(f_{2a\theta}\frac{2R}{C_\gamma + R} + f_{2b\theta}\sqrt{\frac{2R}{Pr}}\frac{1.3}{\sqrt{Pr}R_t^{3/4}} \right) , \tag{4.141}$$

with

$$f_{2a\theta} = f_{1\theta}e^{-(R_t/500)^2} , \tag{4.142}$$

$$f_{2b\theta} = f_{1\theta}e^{-(R_t/200)^2} . \tag{4.143}$$

The time scale $\tau_{l\theta}$ is calculated as the sum of two terms: the first, $B_{1\theta}$, is a bulk term that is a function only of the dynamical time scale τ_u. This term is the most important in the region far from the wall. The second term, $B_{2\theta}$, is the sum of two contributions: a near wall term that is proportional to the time scale $\tau_u\sqrt{R} = \sqrt{\tau_u\tau_\theta}$ and a term that is proportional to the mixed time scale $\tau_m = \tau_u R/(C_\gamma + R)$. This last time scale is calculated as the harmonic mean between the time scales τ_u and τ_θ. In Figure 4.7 the plot of the local time scale $\tau_{l\theta}$ as a function of the non-dimensional distance from the wall y^+ is given together with the terms used to model the timescale itself. With $B_{2\theta a}$ and $B_{2\theta b}$ we refer to the following terms

$$\begin{aligned} B_{2\theta a} &= \tau_u \frac{2R}{C_\gamma + R}\,, \\ B_{2\theta b} &= \tau_u \sqrt{\frac{2R}{Pr}} \frac{1.3}{\sqrt{Pr} R_t^{3/4}}\,. \end{aligned} \tag{4.144}$$

We can see that in the near wall region where $y^+ < 4$ the main contribution to the local time scale is given by the term $B_{2\theta b}$. In the region far from the wall, $y^+ > 300$, the main term is $B_{1\theta}$ while in the intermediate region the time scale $\tau_{l\theta}$ is influenced by both $B_{1\theta}$ and $B_{2\theta a}$. With this formulation the model is able to reproduce the near wall behavior of the eddy thermal diffusivity, which is $\alpha_t \propto \delta^3$, when no thermal fluctuations are assumed on the wall surface, and $\alpha_t \propto \delta^2$, when thermal fluctuations are present on the wall surface [70, 74, 10]. After the introduction of ν_t and α_t, the final form of the RANS system of equations is the following

$$\nabla \cdot \mathbf{u} = 0\,, \tag{4.145}$$

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} &= -\frac{1}{\rho} \nabla P + \\ &+ \nabla \cdot \left[(\nu + \nu_t) \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right) \right] + \mathbf{g} \beta (T - T_r)\,, \end{aligned} \tag{4.146}$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \nabla \cdot \left[(\alpha + \alpha_t) \nabla T \right] + \frac{Q}{\rho C_p}\,. \tag{4.147}$$

4.4 The four logarithmic parameter turbulence model

In Section 4.1 the exact transport equations for turbulence variables k, ε, k_θ and ε_θ have been derived and subsequently modeled in Section 4.2. The four parameter turbulence model has been used in the past providing reliable results for several flow configurations [10, 11, 12], but its numerical stability, in particular for the thermal dissipation rate ε_θ, has been questioned, see [76] and reference therein. The purpose of this PhD work is to provide a new formulation of the turbulence model with an increased numerical stability. In the present section the four logarithmic parameter turbulence model is described. First of all, two new turbulence variables are introduces, ω and ω_θ, defined as the turbulent kinetic energy and the mean squared temperature fluctuations specific dissipation rates. The constitutive expressions of these two new variable are

$$\omega := \frac{\varepsilon}{C_\mu k}\,, \quad \omega_\theta := \frac{\varepsilon_\theta}{C_\mu k_\theta}\,. \tag{4.148}$$

The new turbulence model is obtained in two steps: first the k - ε - k_θ - ε_θ (KE) system of equations is transformed into a k - ω - k_θ - ω_θ (KW) turbulence model and finally into a logarithmic parameter model K - Ω - K_θ - Ω_θ (KLW). The double transformation is performed in order to obtain an increased numerical stability of the turbulence model. The KW model has been proposed with the intent of improving the numerical stability of the KE model by enabling the possibility to impose exact Dirichlet boundary conditions on ω and ω_θ over near-wall regions. From a combination of k, ε, k_θ and ε_θ near wall behaviors the following expressions are obtained

$$\omega_{nw} = \frac{\varepsilon_{nw}}{C_\mu k_{nw}} = 2\frac{\nu}{C_\mu y^2}\,, \tag{4.149}$$

$$\omega_{\theta,nw} = \frac{\varepsilon_{\theta,nw}}{C_\mu k_{\theta,nw}} = 2\frac{\alpha}{C_\mu y^2}\,. \tag{4.150}$$

As can be seen, both ω and ω_θ near wall behaviors (in the latter case for the condition of zero temperature fluctuations along the heated wall) do not depend on k and k_θ values. Exact Dirichlet boundary condition can then be imposed on solid boundaries as the value is exactly calculated using physical (ν and α) and geometrical (y) properties. The constrain of zero fluctuations on the wall, for both velocity and temperature, leads to infinite values of ω and ω_θ. Because of this singular behavior, the boundary is considered at a distance δ from the wall, in order to impose exact and non singular Dirichlet boundary conditions. Characteristic dynamical and thermal time scales τ_u and τ_θ become

$$\tau_u = \frac{k}{\varepsilon} = \frac{1}{C_\mu \omega}\,, \quad \tau_u = \frac{k_\theta}{\varepsilon_\theta} = \frac{1}{C_\mu \omega_\theta}\,, \tag{4.151}$$

so the thermal to dynamical time scale ratio R is simply ω/ω_θ. On solid boundaries, if temperature fluctuations are zero, R has a finite value equal to molecular Prandtl number. For simplicity of notation, we now refer to $\mathbf{u}$, T and p as mean velocity, temperature and pressure, and we label as ν^i_{eff} and α^j_{eff} the effective viscosity and thermal diffusivity for the turbulence variables transport equations, namely

$$\nu^i_{eff} = \nu + \frac{\nu_t}{\sigma_i}\,, \quad \alpha^j_{eff} = \alpha + \frac{\alpha_t}{\sigma_j}\,. \tag{4.152}$$

Transport equation for ω and ω_θ are obtained from those of ε and ε_θ by using the following rule

$$\frac{\partial \varepsilon}{\partial s} = \frac{1}{C_\mu}\left(k\frac{\partial \omega}{\partial s} + \omega\frac{\partial k}{\partial s}\right)\,, \quad \frac{\partial \varepsilon_\theta}{\partial s} = \frac{1}{C_\mu}\left(k_\theta\frac{\partial \omega_\theta}{\partial s} + \omega_\theta\frac{\partial k_\theta}{\partial s}\right)\,, \tag{4.153}$$

where s can either be a space coordinate or time. The k - ω - k_θ - ω_θ turbulence model is then made of the following system of equations

$$\frac{\partial k}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} k = \boldsymbol{\nabla} \cdot \left(\nu^k_{eff} \boldsymbol{\nabla} k\right) + \mathcal{P}_k - C_\mu \, k \, \omega + P_b \,, \tag{4.154}$$

$$\begin{aligned}\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \omega &= \boldsymbol{\nabla} \cdot \left(\nu^\omega_{eff} \boldsymbol{\nabla} \omega\right) + \frac{2}{k} \nu^\omega_{eff} \boldsymbol{\nabla} k \cdot \boldsymbol{\nabla} \omega + \\ &+ (c_{\varepsilon 1} - 1) \frac{\omega}{k} \mathcal{P}_k - C_\mu \left(c_{\varepsilon 2} f_\varepsilon - 1\right) \omega^2 + (c_b - 1) \frac{\omega}{k} P_b \,, \end{aligned}\tag{4.155}$$

$$\frac{\partial k_\theta}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} k_\theta = \boldsymbol{\nabla} \cdot \left(\alpha^{k\theta}_{eff} \boldsymbol{\nabla} k_\theta\right) + \mathcal{P}_{k\theta} - C_\mu \, k_\theta \, \omega_\theta \,, \tag{4.156}$$

$$\begin{aligned}\frac{\partial \omega_\theta}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \omega_\theta &= \boldsymbol{\nabla} \cdot \left(\alpha^{\omega\theta}_{eff} \boldsymbol{\nabla} \omega_\theta\right) + \frac{2}{k_\theta} \alpha^{\omega\theta}_{eff} \boldsymbol{\nabla} k_\theta \cdot \boldsymbol{\nabla} \omega_\theta + \\ &+ \omega_\theta \left[\frac{(c_{p1} - 1)}{k_\theta} \mathcal{P}_{k\theta} + \frac{c_{p2}}{k} \mathcal{P}_k - (c_{d1} - 1) \, C_\mu \omega_\theta - c_{d2} C_\mu \omega \right] \,. \end{aligned}\tag{4.157}$$

As a consequence of the change of state variables, in ω and ω_θ transport equations some additional diffusion terms are introduced, namely the cross diffusion contributions

$$\frac{2}{k} \nu^\omega_{eff} \boldsymbol{\nabla} k \cdot \boldsymbol{\nabla} \omega \,, \quad \frac{2}{k_\theta} \alpha^{\omega\theta}_{eff} \boldsymbol{\nabla} k_\theta \cdot \boldsymbol{\nabla} \omega_\theta \,. \tag{4.158}$$

Production and dissipation terms of ω and ω_θ are obtained as

$$\mathcal{P}_\omega = \frac{1}{k C_\mu} \mathcal{P}_\varepsilon - \frac{\omega}{k} \mathcal{P}_k \,, \qquad \mathcal{S}_\omega = \frac{1}{k C_\mu} \mathcal{S}_\varepsilon - \frac{\omega}{k} \mathcal{S}_k \,, \tag{4.159}$$

$$\mathcal{P}_{\omega\theta} = \frac{1}{k_\theta C_\mu} \mathcal{P}_{\varepsilon\theta} - \frac{\omega_\theta}{k_\theta} \mathcal{P}_{k\theta} \,, \qquad \mathcal{S}_{\omega\theta} = \frac{1}{k_\theta C_\mu} \mathcal{S}_{\varepsilon\theta} - \frac{\omega_\theta}{k_\theta} \mathcal{S}_{k\theta} \,. \tag{4.160}$$

It is important to notice that model constants $c_{\varepsilon 1}$, $c_{\varepsilon 2} f_\varepsilon$, c_{p1}, c_{d1} and c_b need to be greater than one to ensure that the same physical processes that act as production - dissipation for ε and ε_θ act in the same way also for ω and ω_θ.

Another improvement to this model is the possibility of maintaining the state variables and the modeled quantities, such as ν_t and α_t, always positive during the solution process, because negative values do not have physical meaning and lead to numerical instability. Ilinca et al. proposed the use of logarithmic variables as a way to naturally avoid this problem [77]. In the paper the authors used the change of state variables into their natural logarithm as a new method to increase numerical stability of existing turbulence models. As great advantages of this approach they showed that original variables are always kept positive, logarithmic variables have smoother profiles and that once the problem is formulated with logarithmic variables, all turbulence two-equation models can be written as a linear combination of different turbulence models. The approach of logarithmic variables is used in this thesis to provide an increased numerical stability of the four parameter turbulence model. In particular the transformation of state variables into logarithmic values is applied to the system (k,ω,k_θ,ω_θ) in order to take advantage of the benefits of both logarithmic and omega variables, ω and ω_θ. The system of equations can be transformed using the following general rule

$$\text{I} = \ln(i)\,, \quad \rightarrow \quad \frac{\partial \text{I}}{\partial s} = \frac{1}{i}\frac{\partial i}{\partial s}\,, \tag{4.161}$$

the four parameter turbulence model with logarithmic variables (K,Ω,K_θ,Ω_θ) is written as

$$\begin{aligned}
\frac{\partial \text{K}}{\partial t} + \mathbf{u}\cdot\boldsymbol{\nabla}\text{K} = \boldsymbol{\nabla}\cdot\left[\nu^{\text{K}}_{eff}\boldsymbol{\nabla}\text{K}\right] + \nu^{\text{K}}_{eff}\boldsymbol{\nabla}\text{K}\cdot\boldsymbol{\nabla}\text{K} +&\\
+\frac{P_k}{e^{\text{K}}} + c_b\frac{P_b}{e^{\text{K}}} - C_\mu\, e^{\Omega}\,,& \qquad (4.162)\\
\frac{\partial \Omega}{\partial t} + \mathbf{u}\cdot\boldsymbol{\nabla}\Omega = \boldsymbol{\nabla}\cdot\left[\nu^{\Omega}_{eff}\boldsymbol{\nabla}\Omega\right] + 2\nu^{\Omega}_{eff}\boldsymbol{\nabla}\text{K}\cdot\boldsymbol{\nabla}\Omega +&\\
+\nu^{\Omega}_{eff}\boldsymbol{\nabla}\Omega\cdot\boldsymbol{\nabla}\Omega + \frac{c_{\varepsilon 1}-1}{e^{\text{K}}}P_k +&\\
+\frac{c_b-1}{e^{\text{K}}}P_b - C_\mu\left(c_{\varepsilon 2}f_{exp}-1\right)e^{\Omega}\,,& \qquad (4.163)\\
\frac{\partial \text{K}_\theta}{\partial t} + \mathbf{u}\cdot\boldsymbol{\nabla}\text{K}_\theta = \boldsymbol{\nabla}\cdot\left[\alpha^{\text{K}_\theta}_{eff}\boldsymbol{\nabla}\text{K}_\theta\right] + \alpha^{\text{K}_\theta}_{eff}\boldsymbol{\nabla}\text{K}_\theta\cdot\boldsymbol{\nabla}\text{K}_\theta +&\\
+\frac{P_\theta}{e^{\text{K}_\theta}} - C_\mu\, e^{\Omega_\theta}\,,& \qquad (4.164)\\
\frac{\partial \Omega_\theta}{\partial t} + \mathbf{u}\cdot\boldsymbol{\nabla}\Omega_\theta = \boldsymbol{\nabla}\cdot\left[\alpha^{\Omega_\theta}_{eff}\boldsymbol{\nabla}\Omega_\theta\right] + 2\alpha^{\Omega_\theta}_{eff}\boldsymbol{\nabla}\text{K}_\theta\cdot\boldsymbol{\nabla}\Omega_\theta +&\\
+\alpha^{\Omega_\theta}_{eff}\boldsymbol{\nabla}\Omega_\theta\cdot\boldsymbol{\nabla}\Omega_\theta + \frac{c_{p1}-1}{e^{\text{K}_\theta}}P_\theta + \frac{c_{p2}}{e^{\text{K}}}P_k +&\\
-\left(c_{d1}-1\right)C_\mu e^{\Omega_\theta} - c_{d2}C_\mu e^{\Omega}\,.& \qquad (4.165)
\end{aligned}$$

Source terms can be written as

$$\mathcal{P}_{\text{I}} = \frac{1}{e^{\text{I}}}\mathcal{P}_i\,, \quad \mathcal{S}_{\text{I}} = \frac{1}{e^{\text{I}}}\mathcal{S}_i \tag{4.166}$$

In the system (4.162-4.165), the cross diffusion terms

$$\begin{aligned}
&\nu^{\text{K}}_{eff}\boldsymbol{\nabla}\text{K}\cdot\boldsymbol{\nabla}\text{K} \quad , \quad \nu^{\Omega}_{eff}\left(2\boldsymbol{\nabla}\text{K}+\boldsymbol{\nabla}\Omega\right)\cdot\boldsymbol{\nabla}\Omega\\
&\alpha^{\text{K}_\theta}_{eff}\boldsymbol{\nabla}\text{K}_\theta\cdot\boldsymbol{\nabla}\text{K}_\theta \quad , \quad \alpha^{\Omega}_{eff}\left(2\boldsymbol{\nabla}\text{K}_\theta+\boldsymbol{\nabla}\Omega_\theta\right)\cdot\boldsymbol{\nabla}\Omega_\theta\,.
\end{aligned}$$

can be seen as additional convective terms. We introduce the fictitious velocities $\mathbf{u}_K$, $\mathbf{u}_\Omega$, $\mathbf{u}_{K\theta}$, $\mathbf{u}_{\Omega\theta}$ and rewrite the cross diffusion terms as

$$\mathbf{u}_{\text{K}}\cdot\boldsymbol{\nabla}\text{K} \quad , \quad \mathbf{u}_\Omega\cdot\boldsymbol{\nabla}\Omega \quad , \quad \mathbf{u}_{\text{K}_\theta}\cdot\boldsymbol{\nabla}\text{K}_\theta \quad , \quad \mathbf{u}_{\Omega\theta}\cdot\boldsymbol{\nabla}\Omega_\theta\,.$$

Model	Wall	Model	MX	CHF
$k-\omega$	$\dfrac{\partial k}{\partial y}=\dfrac{2k}{y}$ $\omega=\dfrac{2\nu}{C_\mu y^2}$	$k_\theta-\omega_\theta$	$k_\theta=\dfrac{2k_\theta}{y}$ $\omega_\theta=\dfrac{2\alpha}{C_\mu y^2}$	$\dfrac{\partial k_\theta}{\partial y}=0$ $\dfrac{\partial \omega_\theta}{\partial y}=0$
$\mathrm{K}-\Omega$	$\dfrac{\partial \mathrm{K}}{\partial y}=\dfrac{2}{y}$ $\Omega=\ln\left(\dfrac{2\nu}{C_\mu y^2}\right)$	$\mathrm{K}_\theta-\Omega_\theta$	$\dfrac{\partial \mathrm{K}_\theta}{\partial y}=\dfrac{2}{y}$ $\Omega_\theta=\ln\left(\dfrac{2\alpha}{C_\mu y^2}\right)$	$\dfrac{\partial \mathrm{K}_\theta}{\partial y}=0$ $\dfrac{\partial \Omega_\theta}{\partial y}=0$

Table 4.2: Boundary conditions on solid walls for both KW and KLW turbulence models.

Boundary conditions for KLW turbulence model are summarized in Table 4.2. We again remark that ν_t values are limited in a natural way with the logarithmic formulation and the condition $\nu_t \geq 0$ is always satisfied. An upper limit is imposed from a physical realizability condition [78, 79]. The limits are derived as follows. The strain tensor $\mathbf{D}$, in principal axis coordinates, becomes diagonal and its non zero components lead to

$$\lambda_1^2+\lambda_2^2+\lambda_3^2=|\mathbf{D}|^2\,, \tag{4.167}$$

$$\lambda_1+\lambda_2+\lambda_3=0\,. \tag{4.168}$$

For a two-dimensional case $\lambda_a=\sqrt{|\mathbf{D}|^2/2}$, while for a three-dimensional case $\lambda_a\leq\sqrt{2|\mathbf{D}|^2/3}$. From the model of Reynolds stresses

$$\overline{u_a'^2}=-2\nu_t\lambda_a+\frac{2}{3}k\,, \tag{4.169}$$

so that if $\lambda_m=max(\lambda_a)$, then the following must hold

$$2\nu_t\lambda_m\leq\frac{2}{3}k \tag{4.170}$$

as normal Reynolds stresses are positive. From an estimation of the maximum admissible value of λ_m a local upper limit for ν_t can be calculated. For three-dimensional cases it is obtained that $\lambda_m\geq|\mathbf{D}|/\sqrt{6}$, leading to the following upper bound for ν_t

$$\nu_t\leq\sqrt{\frac{2}{3}}\frac{k}{|\mathbf{D}|}\,. \tag{4.171}$$

As turbulent kinetic energy production term is $P_k=2\nu_t|\mathbf{D}|^2$, an upper bound is also derived for P_k, namely

$$P_k\leq\sqrt{\frac{8}{3}}k|\mathbf{D}|\,. \tag{4.172}$$

For a two-dimensional simulation (4.169) can be re-written as

$$\overline{u_a'^2} = -2\nu_t \lambda_a + k\,, \tag{4.173}$$

so the limit to be satisfied is $2\nu_t \lambda_m \leq k$. The bounds for ν_t and P_k are easily obtained as

$$\nu_t \leq \frac{1}{\sqrt{2}} \frac{k}{|\mathbf{D}|}\,, \quad P_k = \sqrt{2} k |\mathbf{D}|\,. \tag{4.174}$$

Chapter 5

Fully developed turbulent flows

In this Chapter the results of fully developed turbulent flows are discussed. Performances of KLW and KE models are evaluated and compared by simulating a fully developed turbulent flow in plane channel geometry. The comparison is provided to demonstrate the increased numerical stability of the KLW turbulence model. A more detailed study of the plane channel, cylindrical pipe and hexagonal 19 pin nuclear reactor bundle is then reported. Different values of both Prandtl and Reynolds numbers are considered, in particular $Pr = 0.025$, that is representative of Lead-Bismuth-Eutectic (LBE) and $Pr = 0.01$, that is representative of liquid sodium. With these simulations the turbulence model performances can be evaluated with simple and well studied cases for which DNS results and experimental correlation are available. In the first case, a detailed evaluation of the model can be performed by reproducing the flow features obtained with DNS simulations. In the latter, the model can be validated on an integral scale, for example by calculating Nusselt number values for heat transfer evaluation.

For the study of fully developed flows some simplifying assumptions can be made. If a straight domain is considered and a constant heat flux is applied on a boundary wall, from the energy balance equation (4.147), it is well known that the temperature grows linearly along the axial coordinate in absence of volumetric source term Q. The temperature gradient along this direction can be calculated by integrating (4.147) on the whole domain, obtaining

$$\frac{dT}{dz} = \frac{4qf_h}{w_{mid}\rho C_p D_h}\,, \tag{5.1}$$

where z is the axial direction, w_{mid} is the cross section average axial velocity, D_h is the hydraulic diameter and f_h is the ratio between the heated and the wet cross section perimeter. In this case the temperature field T, solution of (4.147), is usually transformed into a cross section temperature field $\tilde{T}$ defined as

$$T(x,y,z) = T_{w0} - \tilde{T}(x,y) + \frac{dT}{dz}(z - z_{in})\,, \tag{5.2}$$

where T_{w0} is the reference inlet surface temperature. The temperature field $\tilde{T}(x,y)$, which is the temperature distribution on the transverse section of the domain, does not change with the axial coordinate. The energy balance equation, rewritten with the state variable $\tilde{T}$, becomes

$$\frac{\partial \tilde{T}}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \tilde{T} = \boldsymbol{\nabla} \cdot \left[(\alpha + \alpha_t) \, \boldsymbol{\nabla} \tilde{T} \right] + \mathbf{u} \cdot \hat{\mathbf{a}} \frac{4 q f_h}{w_{mid} \rho C_p D_h} \,, \tag{5.3}$$

where $\hat{\mathbf{a}}$ is the unit vector in the axial direction. The last term in (5.3) is a source which depends on the heat flux on the wall and on the fluid velocity and geometry. This change of variables allows us to impose an periodic boundary conditions also on the temperature field and [10, 72, 80]. The RANS system of equations is then modified as

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0 \,, \tag{5.4}$$

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \mathbf{u} = -\frac{1}{\rho} \boldsymbol{\nabla} P + \mathbf{F}_z + \\ + \boldsymbol{\nabla} \cdot \left[(\nu + \nu_t) \left(\boldsymbol{\nabla} \mathbf{u} + \boldsymbol{\nabla} \mathbf{u}^T \right) \right] + \mathbf{g} \beta (T - T_r) \,, \end{aligned} \tag{5.5}$$

$$\frac{\partial \tilde{T}}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \tilde{T} = \boldsymbol{\nabla} \cdot \left[(\alpha + \alpha_t) \, \boldsymbol{\nabla} \tilde{T} \right] + \mathbf{u} \cdot \hat{\mathbf{a}} \frac{4 q f_h}{w_{mid} \rho C_p D_h} \,, \tag{5.6}$$

where $\mathbf{F}_z$ is parallel to mean flow direction and represents the pressure gradient force that drives the fully developed turbulent flow. The magnitude of $\mathbf{F}_z$ can be set in order to obtain a certain Re_τ case. In particular, $|\mathbf{F}_z|$ can be computed from (4.37), for $y = \delta$ we obtain

$$-\frac{\partial P}{\partial x} = \frac{\tau_w}{\delta} = \frac{\rho u_\tau^2}{\delta} \,. \tag{5.7}$$

Friction velocity u_τ can be written as a function of friction Reynolds number Re_τ in the following form

$$-\frac{\partial P}{\partial x} = \frac{\rho Re_\tau^2 \nu^2}{\delta^3} \,. \tag{5.8}$$

Fully turbulent flows in several geometries are considered in this Chapter, namely plane channel, cylindrical pipe and 19 pin nuclear reactor bundle, in Sections 5.2, 5.3 and 5.4 respectively. Results for both dynamical and thermal fields are shown and compared with DNS results, whereas available, or with experimental correlations.

All the results reported in the following have been obtained from solution of the RANS and KLW system of equations using finite element method in FEMuS code. Time derivatives are discretized using backward Euler method. The algebraic system of equations is solved using the Generalized Minimal Residual method (GMRES) with Incomplete LU factorization (ILU) preconditioner [81].

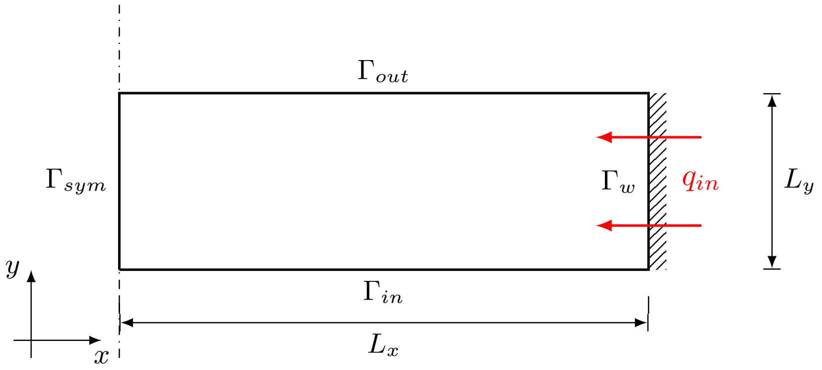


Figure 5.1: Sketch of the plane channel simulated domain.

Property	Symbol	Value		Unit
		$Pr = 0.01$	$Pr = 0.025$	
Viscosity	μ	$5.433 \cdot 10^{-4}$	$1.844 \cdot 10^{-3}$	Pa s
Density	ρ	914.6	10340	Kg/m^3
Thermal conductivity	λ	74.02	10.77	W/(m K)
Heat specific capacity	C_p	1362.3	146	J/(Kg K)

Table 5.1: Physical parameters for fluids with $Pr = 0.01$ and $Pr = 0.025$.

Var.	Γ_{sym}	$\Gamma_{in} \cup \Gamma_{out}$	Γ_w
u	$u = 0$	$\dfrac{\partial u}{\partial y} = 0$	$u = 0$
v	$\dfrac{\partial v}{\partial x} = 0$	$\dfrac{\partial v}{\partial y} = 0$	$\dfrac{\partial v}{\partial x} = \frac{\mu}{\delta}$
k	$\dfrac{\partial k}{\partial x} = 0$	$\dfrac{\partial k}{\partial y} = 0$	$\dfrac{\partial k}{\partial x} = \frac{2k}{\delta}$
Ω	$\dfrac{\partial \Omega}{\partial x} = 0$	$\dfrac{\partial \Omega}{\partial y} = 0$	$\Omega = \ln\left(\frac{2\nu}{C_\mu \delta^2}\right)$

Table 5.2: Boundary conditions imposed for the simulations of fully developed turbulent flows-dynamical fields in a plane channel.

5.1 Comparison between KLW and KE models

In the present Section results are shown to demonstrate the enhanced numerical stability of the four logarithmic parameter turbulence model KLW with respect to original KE formulation. The comparison is made with the KE formulation since this formulation has been more widely used, so it would be of great interest to increase the performances of existing model implementations [82, 83, 84, 85]. The

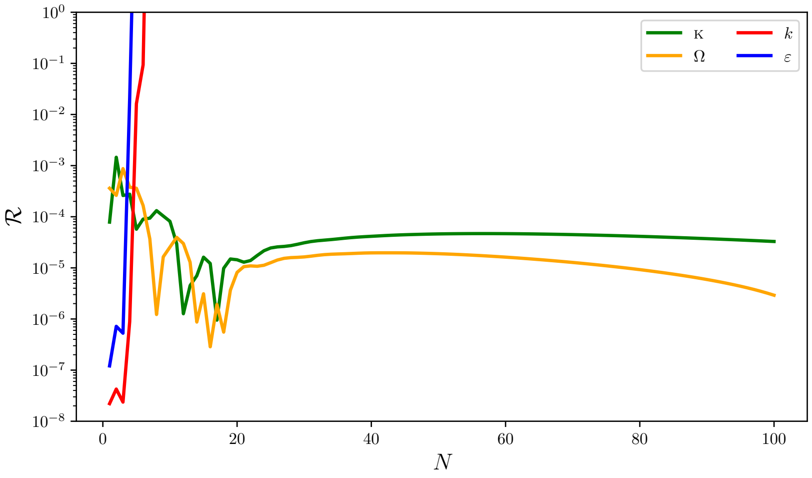


Figure 5.2: Comparison of $\mathcal{R}$ values obtained for k, ε, K and Ω for using same time discretization step.

fully developed turbulent flow in plane channel, with a friction Reynolds number $Re_\tau = 2000$ and fluid molecular Prandtl number $Pr = 0.01$ is chosen as a test case to investigate turbulence model performances. A more detailed study of fully developed turbulent flows in plane channel geometry is presented in the following Section. The simulated geometry is sketched in Fig. 5.1 and consists of two plates located at a distance $L = 0.0605$ m. The condition of fully developed turbulent flow is imposed with periodic boundary conditions on the inlet and outlet of the channel. The physical parameters used to obtain the present results are listed in Tab. 5.1. Because of the symmetry of the problem only half of the physical domain is simulated. We indicate with Γ_{in} the inlet surface, with Γ_{out} the outlet surface, with Γ_w the wall surface and with Γ_{sym} the surface on the symmetry plane. The boundary conditions we impose are reported in Table 5.2. Stationary solutions are obtained with a time marching scheme. Thermal fields, for this case, are passive scalars, so first the solution for velocity and dynamical turbulence variables is obtained and then the one for mean temperature and thermal turbulence variables. To better evaluate the performances of the two models we introduce variables $\mathcal{N}_{1,i}$ and $\mathcal{N}_{2,i}$ calculated as

$$\mathcal{N}_{1,i} := ||\Phi_i - \Phi_{i+1}||_{L_2} \,, \qquad \mathcal{N}_{2,i} := ||\Phi_i - \Phi_{steady}||_{L_2} \,. \tag{5.9}$$

The value of $\mathcal{N}_{1,i}$ represents the L_2 norm of the difference between two consecutive printed solutions, i and $i+1$, while $\mathcal{N}_{2,i}$ is the L_2 norm of the difference between the solution at times step i and the final one, labeled as Φ_{steady}. Another useful parameter to monitor is the sum of weighted residuals $\mathcal{R}$. For a generic partial differential equation Eq_i written in the form

$$Eq_i : \quad \mathcal{L}u - f = 0\,, \quad \text{on}\,\Omega\,, \tag{5.10}$$

the residual R of approximate solution u_h is calculated as

$$R = \mathcal{L}u_h - f\,. \tag{5.11}$$

The weighted residual is obtained from integration of equation 5.11 over domain Ω, after it has been multiplied by test function φ. For a given domain discretization Ω_h with $\mathcal{N}$ nodes, the sum of weighted residual $\mathcal{R}$ is calculated as

$$\mathcal{R} = \sum_{i=1}^{\mathcal{N}} \int_{\Omega} (\mathcal{L}u_h - f)\varphi_i d\Omega\,. \tag{5.12}$$

It is clear that when u_h tends to exact solution u, the sum of weighted residuals becomes zero. Differently from the values of norms $\mathcal{N}_1$ and $\mathcal{N}_2$, the value of $\mathcal{R}$ contains information on how approximated solution u_h is close to exact one u. The initial conditions for dynamical turbulence variables are provided as uniform fields over the domain, with equivalent values for k, ε and K, Ω, namely

$$k = k_{in}\,, \quad \varepsilon = \varepsilon_{in}\,, \tag{5.13}$$

$$\mathrm{K} = \ln(k_{in})\,, \quad \Omega = \ln\left(\frac{\varepsilon_{in}}{C_\mu k_{in}}\right)\,. \tag{5.14}$$

A fine mesh in the near-wall region is used, so that viscous layer can be resolved. Natural boundary conditions are set for k and K, from near-wall behavior

$$k_{nw} = \lambda\delta^2\,, \quad \rightarrow \quad \frac{\partial k}{\partial \delta} = \frac{2k_{nw}}{\delta}\,, \tag{5.15}$$

where δ is the wall distance. Essential boundary conditions are used for ε and Ω. In particular for ε the following defective scheme is used

$$\varepsilon_{nw}^{i+1} = \varepsilon_{nw}^{i} + \gamma\left(\frac{2\nu}{\delta^2}k_{nw}^{l} - \varepsilon_{nw}^{i}\right)\,, \tag{5.16}$$

in order to avoid huge oscillations in ε values. For Ω the near-wall value is constant in time since it depends only on kinematic viscosity ν and wall distance δ

$$\Omega = \ln\left(\frac{2\nu}{C_\mu\delta^2}\right)\,. \tag{5.17}$$

In Fig. 5.2 obtained values of $\mathcal{R}$, for k, K, ε and Ω are plotted against the number of iterations N performed. The evolution of $\mathcal{R}$ is here shown for the first

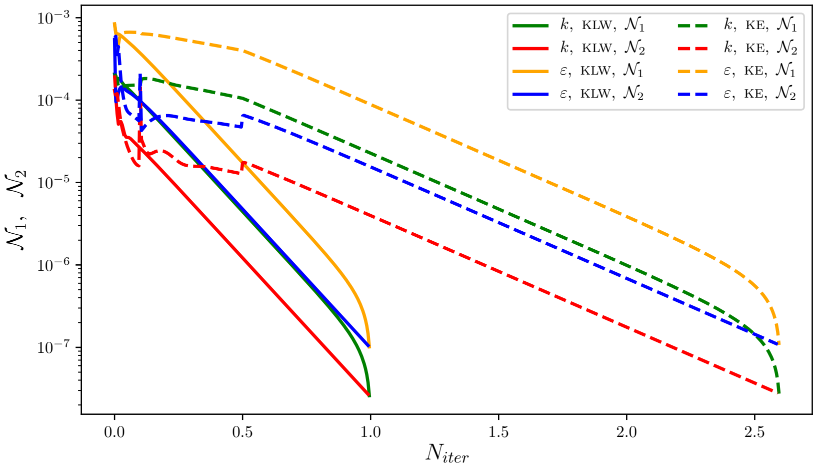


Figure 5.3: Comparison of L_2 norms $\mathcal{N}_1$ and $\mathcal{N}_2$ values obtained with KLW and KE. Values refer to time evolution of k and ε variables.

performed iterations, in order to make a comparison between the behaviors of the two turbulence models. The results have been obtained using the same fictitious time discretization step. For the KLW turbulence model it is observed that $\mathcal{R}$ present some oscillations, at the very first iterations, and then stabilizes, with Ω variable having a more rapidly decreasing value of $\mathcal{R}$ with respect to K. For KE turbulence model the initial values of $\mathcal{R}$ are smaller than those obtained with KLW but as soon as ε wall value begins to change $\mathcal{R}$ rapidly diverges. A smaller time step is then required for the solution of KE model, at least for an initial amount of iterations. Complete time evolution of $\mathcal{N}_1$ and $\mathcal{N}_2$ norm values is reported in Fig. 5.3. In this Figure norms are calculated on k and ε, so KLW model results have been transformed back to natural values of k and ε, in order to make a comparison between norms of the same variables. Time evolution is described as a function of normalized number of iterations N_{iter}, namely the number of iteration divided by the number of iterations to final convergence of KLW model. For KLW model the same time discretization step has been used for the entire simulation. After an initial solution guess, norms $\mathcal{N}_1$ and $\mathcal{N}_2$ rapidly decrease with the number of iterations. A smaller initial time step value is used with KE model. After some initial iterations, when norms $\mathcal{N}_1$ start to monotonically decrease, time step value can be increased. It can be seen that the solution with KE model required a total number of iterations approximately 2.5 times bigger than those required for the solution of KLW model. A comparison of the obtained results is shown on the left of Fig. 5.4, where non-dimensional value of v^+, k^+ and ε^+ are plotted against non-dimensional wall distance y^+. The models reproduce the same results, as expected, with only a small difference in the near wall value of ε.

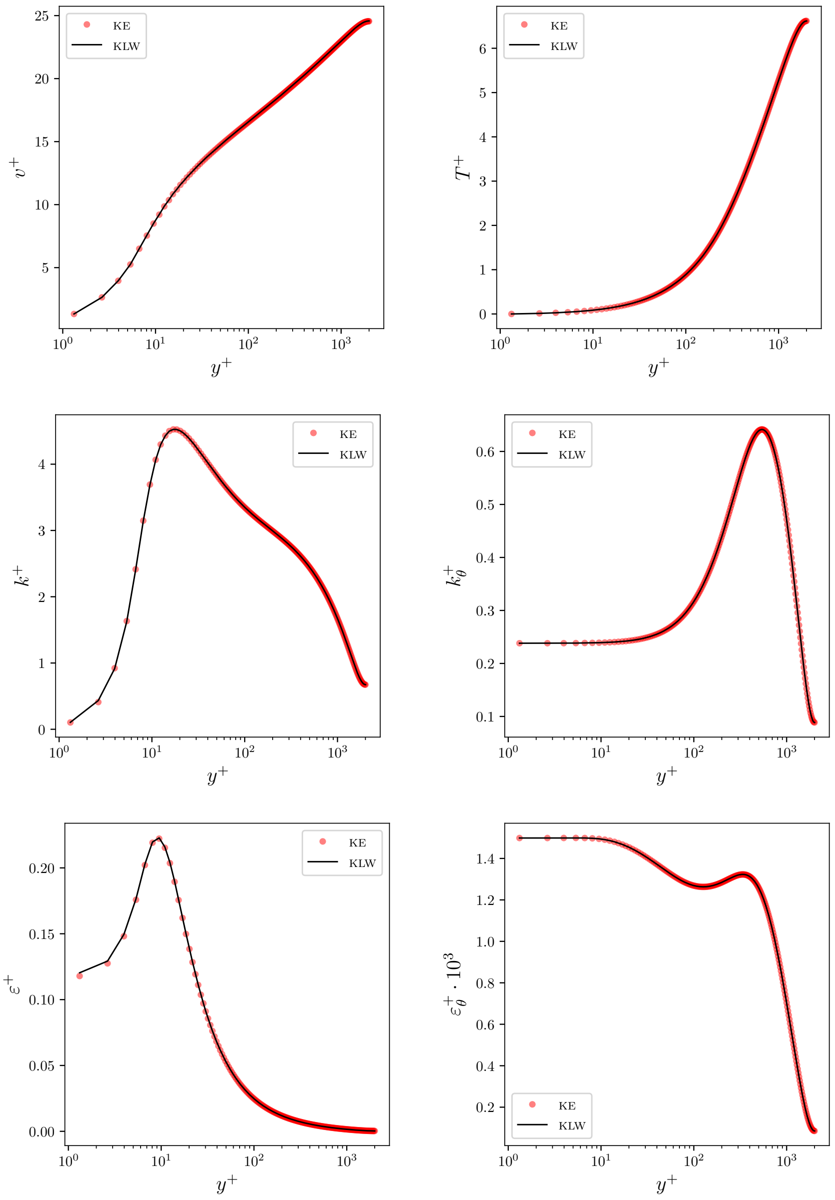


Figure 5.4: Comparison of results obtained with KE and KLW formulations of four parameter turbulence model. Plots are taken on channel cross sections and reported as a function of non dimensional wall distance y^+. On the left results for dynamical turbulence and on the right thermal field results for CHF boundary condition.

Once that the solution for dynamical variables has been obtained, the computation of thermal variables is set up. For each field an initial uniform value is set,

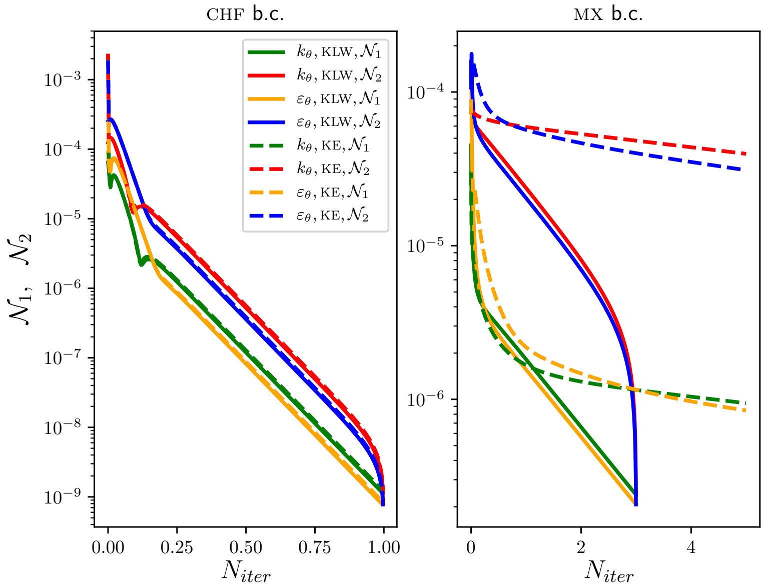


Figure 5.5: Comparison of L_2 norm $\mathcal{N}_1$ and $\mathcal{N}_2$ values obtained with KLW and KE models for simulation of turbulent heat transfer in plane channel, CHF boundary condition, $Re_\tau = 2000$ and $Pr = 0.01$.

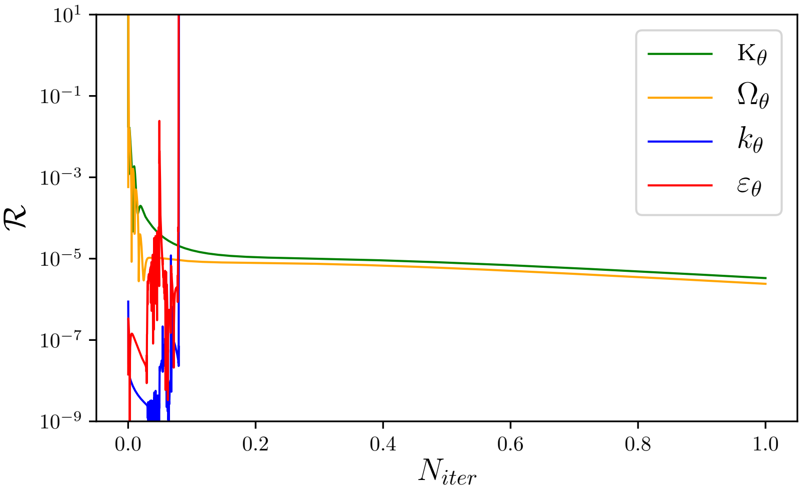


Figure 5.6: Comparison of $\mathcal{R}$ values obtained for k_θ, ε_θ, K_θ and Ω_θ for case MX and solution with same fictitious time step.

with equivalent values for k_θ, ε_θ and K_θ, Ω_θ, namely

$$k_\theta = k_{\theta,in}\,, \quad \varepsilon_\theta = \varepsilon_{\theta,in}\,, \tag{5.18}$$

$$\mathrm{K}_\theta = \ln(k_{\theta,in})\,, \quad \Omega_\theta = \ln\left(\frac{\varepsilon_{\theta,in}}{C_\mu k_{\theta,in}}\right)\,. \tag{5.19}$$

Both CHF and MX boundary conditions for thermal turbulence variables are investigated. For CHF case the boundary conditions are set for both systems of thermal turbulence variables $(k_\theta, \varepsilon_\theta)$ and $(\mathrm{K}_\theta, \Omega_\theta)$ by imposing a zero gradient value in the direction normal to interested boundary side. Thermal dissipation rate ε_θ does not suffer from a dependence on k_θ value, and KE and KLW models behave in the same way, as described in Fig. 5.5. Results of thermal fields are shown on the right of Fig. 5.4, where non-dimensional values of temperature T^+, mean squared temperature fluctuations k_θ^+ and their dissipation ε_θ^+ are plotted against non-dimensional wall distance y^+. One can see that the two different model formulations lead to the same results, as expected. Plots of norms $\mathcal{N}_1$ and $\mathcal{N}_2$ norms are reported as a function of variable N_{iter}. For this case N_{iter}, the ratio of number of iterations versus number of iterations to convergence for CHF case, is in the range of 0 to 1. Values of $\mathcal{N}_1$ and $\mathcal{N}_2$, for KLW model, have been calculated on $(k_\theta, \varepsilon_\theta)$ values obtained from $(\mathrm{K}_\theta, \Omega_\theta)$ in order to compare them with those obtained with KE model. For this case the same fictitious time step has been used for both turbulence models, leading to an equivalent time evolution of the solutions. A different behavior is observed with the setting of MX boundary conditions. Simulations are performed starting from steady solution obtained for CHF case. Natural boundary conditions are still set for k_θ and K_θ, since their near-wall gradient component along wall normal direction can be expressed as a function of wall distance δ. Essential boundary conditions are used for ε_θ and Ω_θ: for the first variable a Dirichlet boundary condition is set according to

$$\varepsilon_{\theta\, nw} = \frac{2\alpha}{\delta^2} k_\theta \,, \tag{5.20}$$

while for the logarithmic omega variable the following value

$$\Omega_{\theta\, nw} = \ln\left(\frac{2\alpha}{C_\mu \delta^2}\right) \tag{5.21}$$

is imposed. In order to prevent high oscillating wall values of ε_θ, Dirichlet boundary condition for ε_θ is updated with a defective method, i.e. as

$$\varepsilon_{\theta\, nw}^{i+1} = \varepsilon_{\theta\, nw}^{i} + \gamma\left(\frac{2\alpha}{\delta^2} k_{\theta\, nw}^{l} - \varepsilon_{\theta\, nw}^{i}\right) \,, \tag{5.22}$$

where $i+1$ is the actual time step, $k_{\theta\, nw}^{l}$ is the last updated near-wall value of k_θ and γ is a constant value smaller than one. If $\gamma = 0$ then the near-wall value of ε_θ is kept equal to the last computed value, i.e. the first imposed value if γ is not changed during the simulation, otherwise for $\gamma = 1$ the new value of ε_θ is completely determined on the last updated value of k_θ. In Fig. 5.5 the time evolution of norms $\mathcal{N}_1$ and $\mathcal{N}_2$ values is reported as a function of N_{iter} variable.

From the side of KLW model, MX boundary conditions introduce high gradients of Ω_θ field in the near-wall region, requiring a smaller fictitious time step than the one used for the solution of CHF case. Convergence is then obtained after the value of N_{iter} is greater than 1. For the KE model the used time step is equal to one tenth of that used with KLW, for a matter of stability as with higher dt values the solution diverges. This behavior can be easily seen from Fig. 5.6, where values of $\mathcal{R}$ have been plotted against N_{iter} values, for solutions of $(k_\theta, \varepsilon_\theta)$ and $(\text{K}_\theta, \Omega_\theta)$ using the same value of dt. $\mathcal{R}$ value for the system $(k_\theta, \varepsilon_\theta)$ are initially small, as the starting solution is the steady one of CHF case and ε_θ wall values are updated with (5.22). After some iterations $\mathcal{R}$ begins to oscillate until the solution diverges. For $(\text{K}_\theta, \Omega_\theta)$ system the correct wall value of Ω_θ is imposed at the first iteration, leading to initially high values of $\mathcal{R}$. After a few iterations, characterized by some oscillations, $\mathcal{R}$ stabilizes. Again from Fig. 5.5 it is clear that KE model requires a greater number of iteration to solution convergence.

From this simple test case it has be shown that the double turbulence model transformation, i.e. from ε variable to ω ones and from "natural" variables to logarithmic values, allows to obtain an increased numerical stability. In this light of view the above mentioned turbulence model transformation can be seen as a generic mathematical tool that can be applied to existing k-ε turbulence models to increase their numerical stability. The analysis has been performed by comparing the performances of KLW an KE models in simulating a steady fully developed turbulent flow. A major limitation of the KE model is represented by the boundary condition imposition for ε and ε_θ variables on wall boundaries. Small time steps are required to limit ε and ε_θ oscillations on wall boundaries while Ω and Ω_θ are not affected by this issue. Transient simulations have not been considered since in this thesis only steady simulations are performed. Anyway, while simulating transient situations, small time steps are required to reproduce phenomena occurring on small time scales. In this framework the KE model should suffer less of the boundary condition issue, but the logarithmic formulation is yet expected to be more stable due to the natural lower limitation of all solved variables. Original solved variables are in fact prevented from becoming negative so that lower threshold values do not need to be set in order to avoid numerical instabilities.

5.2 Plane channel

5.2.1 Dynamic turbulence

The plane channel is one of the simplest types of geometry that can be simulated and many Direct Numerical Simulations are available for this kind of flow. In particular for fully developed turbulent flows results of the dynamical turbulence

	k^+	ω^+	ν_t^+
Near wall	$\propto y^{+2}$	$\frac{2}{C_\mu y^{+2}}$	$\propto y^{+3}$
Const. stress	$\frac{1}{\sqrt{C_\mu}}$	$\frac{1}{\kappa\sqrt{C_\mu} y^+}$	κy^+

Table 5.3: Reference profiles, as a function of non-dimensional wall distance, for k^+, ω^+ and ν_t^+ in both near-wall and constant stress layer.

for the cases of friction Reynolds number $Re_\tau = u_\tau\delta/\nu \simeq$ 180, 395, 590, 640, 950, 2000 and 4200 are available [86, 87, 88, 89]. The simulation configuration is the same used for the performance comparison between KLW and KE models and described in the previous Section. In particular the physical domain is sketched in Fig. 5.1, while fluid physical properties and boundary conditions are reported in 5.1 and 5.2 respectively. Six test cases with Reynolds number $Re \simeq$ 5700 (A), 14000 (B), 24000 (C), 37000 (D), 86000 (E) and 198000 (F) are considered. They correspond to friction Reynolds numbers $Re_\tau \simeq$ 180, 395, 640, 950, 2000 and 4200. All simulations are performed using a spatial discretization of the physical domain in order to have the first mesh point at $y^+ < 1$.

Profiles of non-dimensional turbulent kinetic energy $k^+ = e^{\mathrm{K}}/u_\tau^2$, specific dissipation rate $\omega^+ = e^\Omega\nu/u_\tau^2$ and eddy kinematic viscosity $\nu_t^+ = \nu_t/\nu$ are shown in Fig. 5.7 for all the simulated cases. Reference behaviors for each variable are shown. In particular, these are the ones obtained in Section 4.2 and summarized in Table 5.3 for both near-wall and constant stress layer.

Quadratic dependence on y^+ and uniform value in the constant stress layer are perfectly reproduced by the solved K fields. Specific dissipation rate ω is proportional to the inverse of quadratic wall distance, in the near-wall region, and to the inverse of wall distance in the constant stress layer. The obtained results are in very good agreement with the reference behaviors. Finally eddy kinematic viscosity grows with the cubic power of wall distance, in the viscous region, and then in the logarithmic region. The cubic behavior is fairly reproduced and a better agreement is obtained with the linear behavior. In Figure 5.8 a comparison of non-dimensional velocity values is shown between the results obtained with the KLW turbulence model and those obtained from DNS simulations for the cases of $Re_\tau = 180$, 395, 640, 950, 2000 and 4200. The linear behavior $v^+ = y^+$ and the logarithmic one $v^+ = 1/0.41\ln(y^+) + 5$ are well reproduced in both cases and the results are in good agreement with DNS ones for each simulated case.

5.2.2 Thermal turbulence

Direct Numerical Simulations results of fully developed thermal turbulent flow for a fluid with $Pr = 0.025$ are available for the cases of friction Reynolds number

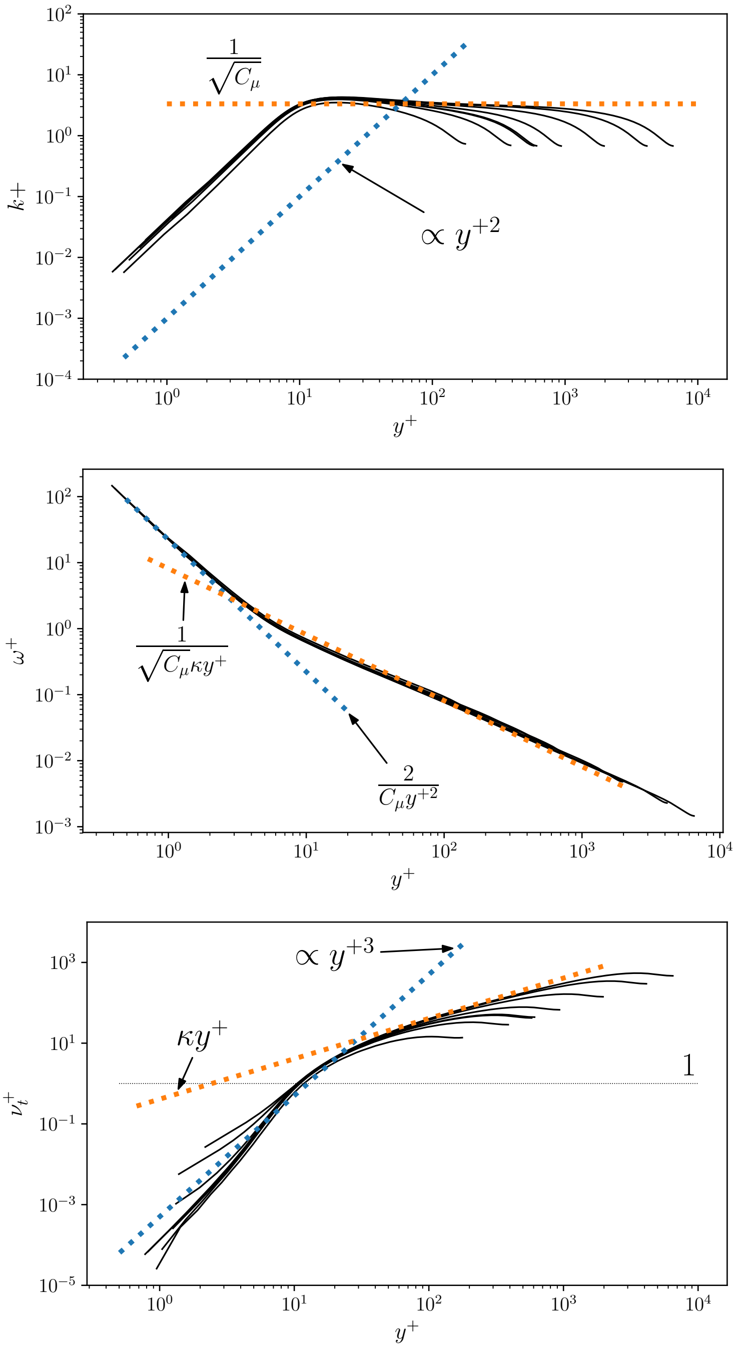


Figure 5.7: From top to bottom: non-dimensional turbulent kinetic energy k^+, specific dissipation rate ω^+ and eddy kinematic viscosity ν_t^+ as a function of non-dimensional wall distance y^+ for all the simulated cases.

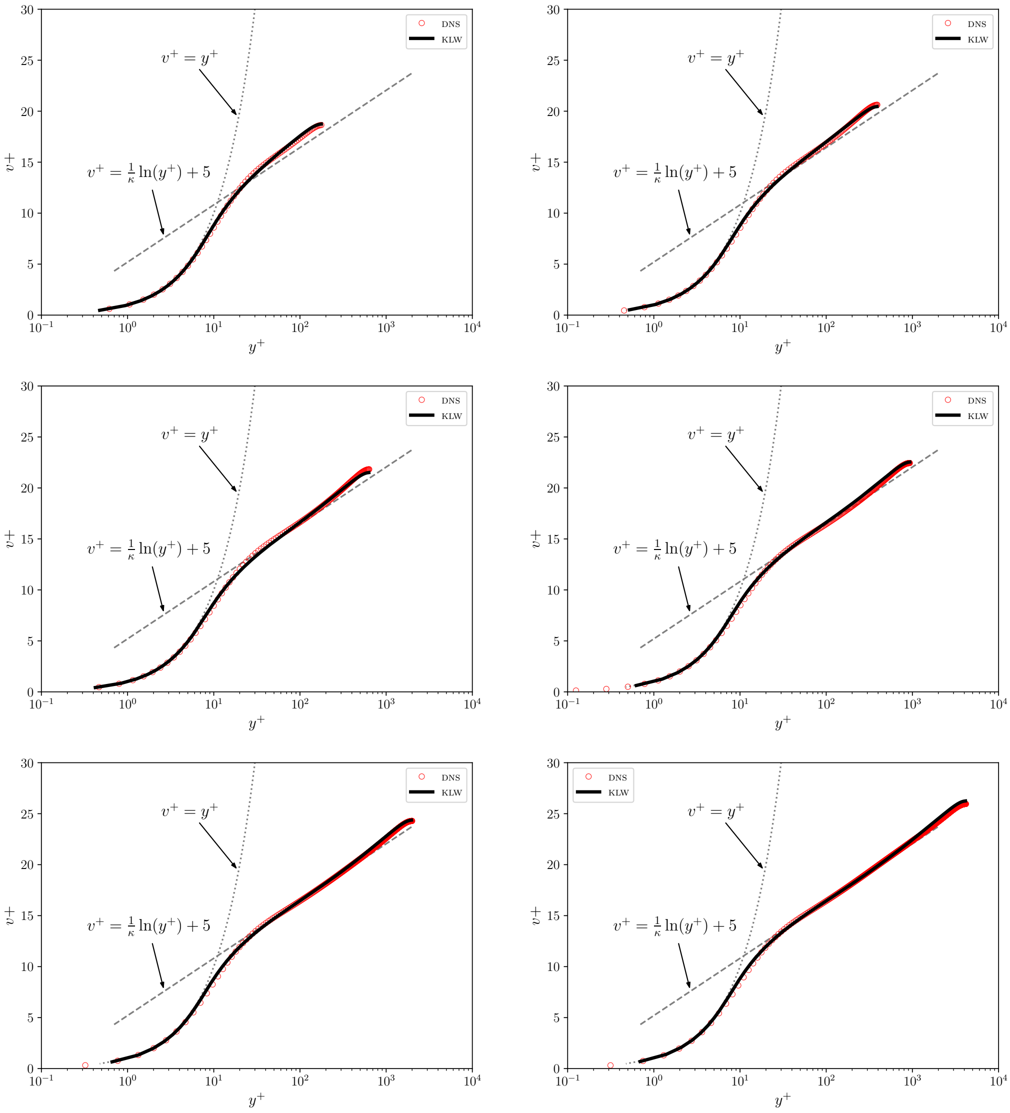


Figure 5.8: Non-dimensional velocity profiles plotted against non dimensional wall distance y^+ and compared with DNS results and characteristic linear and logarithmic behaviors.

Re_τ = 180, 395, 640, 1020 [86, 90]. For the case of $Pr = 0.01$, DNS results are provided for the study of conjugate turbulent heat transfer in plane channel with Re_τ = 180, 395, 590 [87]. Moreover Large Eddy Simulation results at $Re_\tau = 2000$ are described in [58] for both $Pr = 0.01$ and $Pr = 0.025$. The boundary conditions imposed to the solution of the thermal turbulence system of equations are reported in Tab. 5.4. On wall boundary Γ_w a uniform heat flux of 3.6×10^5 W/m^2 is applied and both MX and CHF boundary conditions are

Var.	Γ_{sym}	$\Gamma_{in} \cup \Gamma_{out}$	Γ_w
$\tilde{T}$	$\dfrac{\partial \tilde{T}}{\partial x} = 0$	$\dfrac{\partial \tilde{T}}{\partial y} = 0$	$\tilde{T} = 0$
K_θ	$\dfrac{\partial \mathrm{K}_\theta}{\partial x} = 0$	$\dfrac{\partial \mathrm{K}_\theta}{\partial y} = 0$	$\dfrac{\partial \mathrm{K}_\theta}{\partial x} = \frac{2}{\delta}\,,\ \dfrac{\partial \mathrm{K}_\theta}{\partial x} = 0$
Ω_θ	$\dfrac{\partial \Omega_\theta}{\partial x} = 0$	$\dfrac{\partial \Omega_\theta}{\partial y} = 0$	$\Omega_\theta = \ln\left(\frac{2\alpha}{C_\mu \delta^2}\right)\,,\ \dfrac{\partial \Omega_\theta}{\partial x} = 0$

Table 5.4: Boundary conditions imposed for the simulations of fully developed turbulent flows - thermal fields - in plane channel.

	k_θ^+		ω_θ^+		α_t^+	
B.c. type	MX	CHF	MX	CHF	MX	CHF
Near wall	$\propto y^{+2}$	$\propto y^{+0}$	$\frac{2}{C_\mu y^{+2}}$	$\propto y^{+0}$	$\propto y^{+3}$	$\propto y^{+2}$
Const. stress	$\frac{1}{2Pr_t\sqrt{C_\mu}}$		$\frac{2}{\kappa\sqrt{C_\mu} y^+}$		$Pr_t \kappa y^+$	

Table 5.5: Reference profiles, as a function of non-dimensional wall distance, for k_θ^+, ω_θ^+ and α_t^+, in near-wall and constant stress-flux layer, and for both MX and CHF boundary conditions.

considered for K_θ-Ω_θ.

Non-dimensional mean squared temperature fluctuations $k_\theta^+ = e^{\mathrm{K}_\theta}/T_\tau^2$, thermal specific dissipation rate $\omega_\theta^+ = e^{\Omega_\theta}\nu/u_\tau^2$ and eddy thermal diffusivity $\alpha_t^+ = \alpha_t/\nu$ are shown in Fig. 5.9 as a function of non-dimensional wall distance y^+. The profiles are obtained from simulations with both MX and CHF boundary conditions applied on the system K_θ-Ω_θ. The reference behavior, for near-wall and constant stress-flux regions, is summarized in Tab. 5.5 and plotted in 5.9. It is easy to see that for MX boundary conditions, quadratic behavior of k_θ is captured up to $y^+ \simeq 2$ while, for higher values of y^+, k_θ grows quickly between linear and quadratic order. The thermal specific dissipation rate ω_θ decreases, as wall distance increases, proportionally to the inverse of y^{+2}. The cubic growth of eddy thermal diffusivity is almost well captured up to $y^+ \simeq 10$. For the case of CHF boundary conditions, both k_θ^+ and ω_θ^+ are characterized by flat profiles in the whole viscous layer. The quadratic behavior of α_t^+ is well captured with modeling functions described in Section 4.3. Differently from the dynamic variables examined in Fig. 5.7 the computation of thermal turbulence variables in the bulk region is difficult because of the small Prandtl number. In Fig. 5.10 non-dimensional values of mean temperature $T^+ = T/T_\tau$, for cases with $Pr = 0.025$ and $Pr = 0.01$, and non-dimensional velocity are plotted against non-dimensional wall distance. The friction Reynolds number of the considered cases varies from $Re_\tau = 180$ to

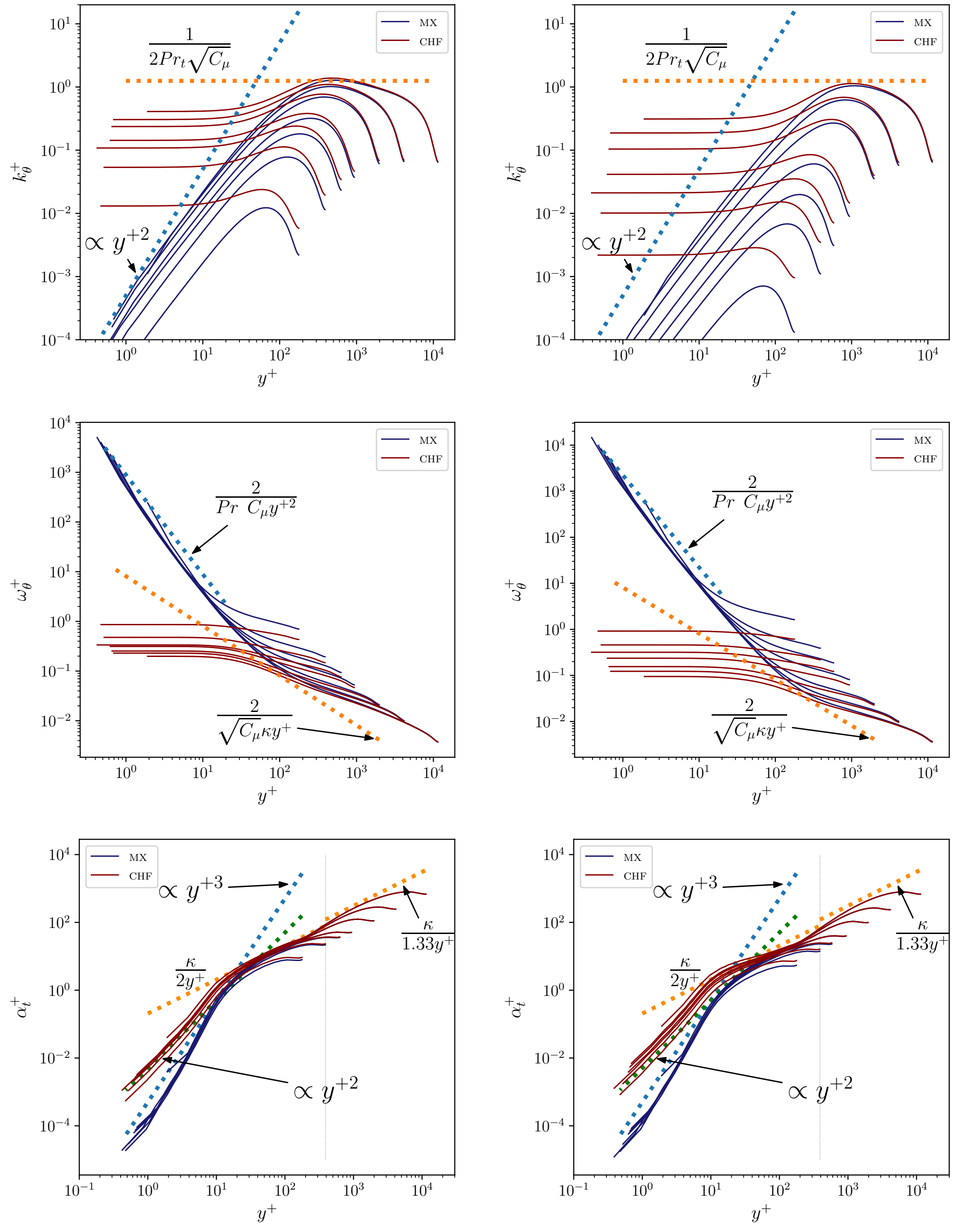


Figure 5.9: From top to bottom: non-dimensional mean squared temperature fluctuations k_θ^+, specific dissipation rate ω_θ^+ and eddy thermal diffusivity α_t^+ plotted against non-dimensional wall distance y^+. Results for $Pr = 0.025$ on the left and $Pr = 0.01$ on the right.

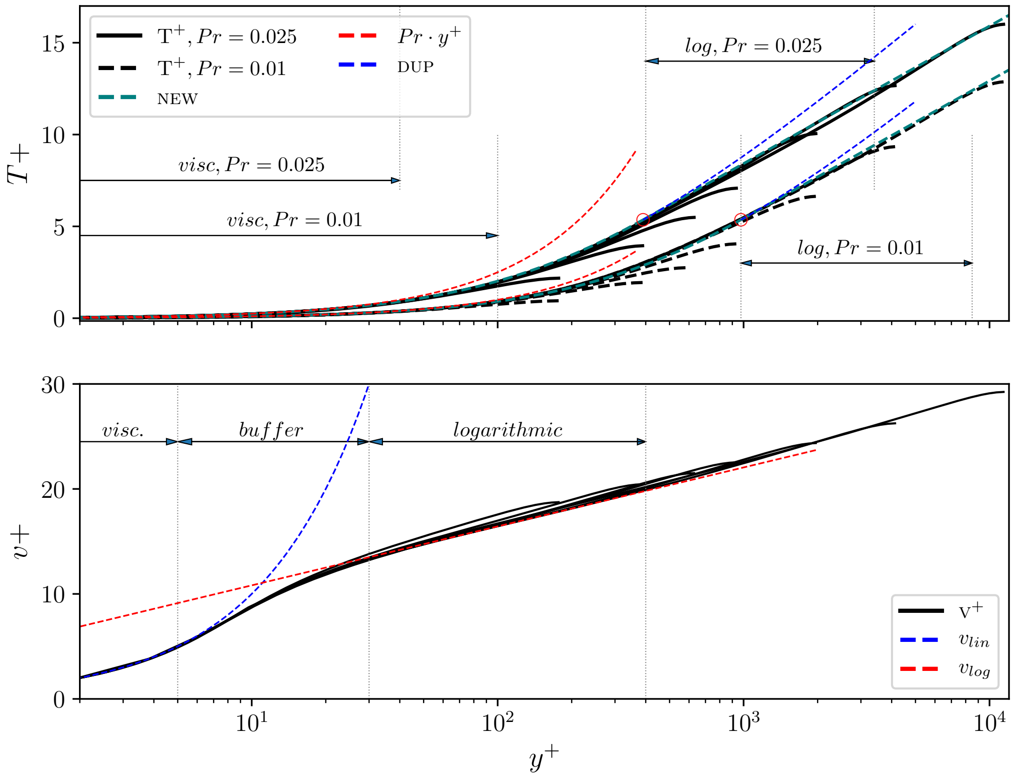


Figure 5.10: Plots of non-dimensional temperature values, for $Pr = 0.025$ and $Pr = 0.01$, and non-dimensional velocity profiles against non-dimensional wall distance. Viscous and logarithmic regions are shown for both temperature and velocity fields. Red circles show the points where the newly proposed temperature law (4.53) and Duponcheel's (4.51) meet.

$Re_\tau = 11600$. The comparison is proposed in order to notice the different y^+ ranges of T^+ and v^+ characteristic behaviors. For the velocity field the linear law is observed in the range $y^+ \in [0, 5]$ and the logarithmic one starting from $y^+ \simeq 30$ and up to an y^+ value that depends on Re_τ. Temperature values obtained with $Pr = 0.025$ show a departure from the linear behavior at about $y^+ = 40$. Temperature logarithmic behavior is obtained for cases $Re_\tau = 2000$, $Re_\tau = 4200$ and $Re_\tau = 11600$ from non-dimensional wall distance $y^+ = 400$. Duponcheel's law (4.51) for temperature approximates very well the viscous and buffer layer, while the newly proposed one (4.53) perfectly overlaps temperature profiles in the logarithmic region. From Fig. 5.9 one can see that eddy thermal diffusivity profiles are approximated quite well by the behavior

$$\alpha_t^+ = \frac{\kappa}{2y^+}\,, \qquad \alpha_t^+ = \frac{\kappa}{1.33y^+}\,, \tag{5.23}$$

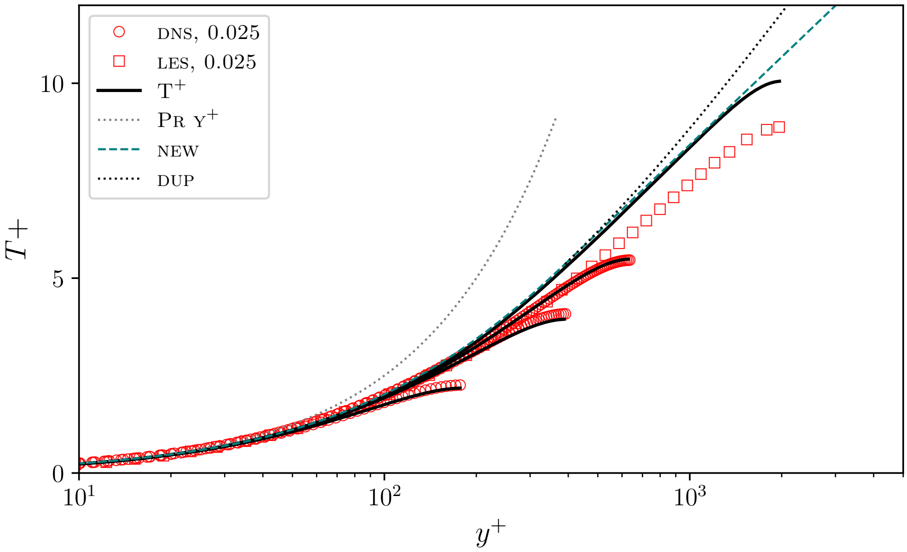


Figure 5.11: Comparison of non-dimensional temperature values, for $Pr = 0.025$, with DNS and LES results, as a function of non dimensional wall distance y^+ [86, 58].

upon which Duponcheel's relation and the newly proposed one are based, in the range $y^+ \in [20, 400]$ and $y^+ \in [400, 2000]$.

Source	Case [Re_τ]						
	180	395	640	950	2000	4200	11600
KLW	5.62	6.77	7.52	8.92	12.72	20.44	43.15
LES [58]			8.36		14.39		
CHENG & TAK	5.45	6.46	7.44	8.84	12.05	19.95	44.11

Table 5.6: Comparison of Nusselt number values for the plane channel simulations with LES and reference correlation values.

Results for $Pr = 0.025$

A comparison with DNS and LES results for non-dimensional temperature profiles, with $Pr = 0.025$, is shown in Fig. 5.11. Results for $Re_\tau = 180$, 395 and 640 are from [86], while LES solutions for $Re_\tau = 2000$ are from [58]. The results are in very good agreement with DNS ones, while a sensible difference is observed with LES results for case $Re_\tau = 2000$. These results were obtained in [58] using a combined LES - DNS approach, i.e., a LES simulation where a subgrid-scale stress

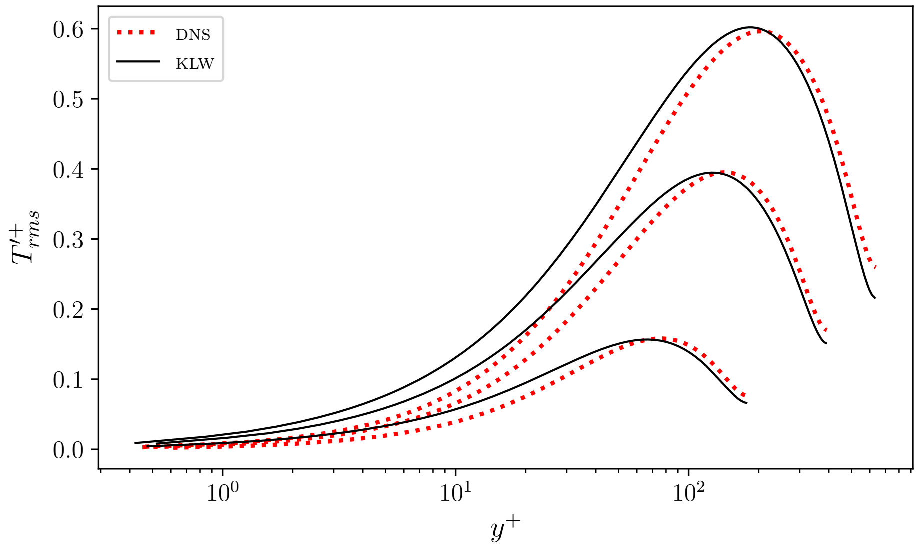


Figure 5.12: Comparison of non-dimensional root mean squared temperature fluctuations T'^{+}_{rms}, plotted against non dimensional wall distance y^+, with DNS values for $Pr = 0.025$ and cases $Re_\tau = 180$, 395 and 640 [86].

model had been used for the solution of the momentum equation while for the energy equation no subgrid-scale turbulent heat flux model had been used. The reason behind this approach is that for very low Prandtl number values the smallest thermal scales are yet much bigger than velocity ones, so that with a particular fine grid resolution the temperature field can be captured without a subgrid model. Following the procedure described in [58] the Nusselt number values are here calculated, in order to have a measure of heat transfer performance, and compared with the correlation of Cheng and Tak [91] that will be better introduced in the section of thermal results for cylindrical pipe geometry. Nusselt number values are reported in Tab. 5.6. It can be seen that the values obtained with KLW model are in good agreement with the correlation of Cheng & Tak on a very wide range of Peclet values, namely from $Pe = 144$ for $Re_\tau = 180$ to $Pe = 15500$ for $Re_\tau = 11600$. Results from LES simulations show a bigger discrepancy with reference correlation, in particular for the case $Re_\tau = 2000$. Results obtained with KLW model can then be considered rightful ones. Non-dimensional root mean squared temperature fluctuations $T'^{+}_{rms} = \sqrt{2e^{\mathrm{K}}_{\theta}}/T_\tau$ profiles are plotted in Figure 5.12 against non-dimensional wall distance y^+ and compared with DNS relative values [86]. With MX boundary conditions, temperature fluctuations increase with wall distance and reach a maximum value that, for high Re_τ values, occurs in the constant stress-flux region, as previously seen in Fig. 5.9. Temperature fluctuations are slightly overestimated, with respect to DNS values, in the region close to

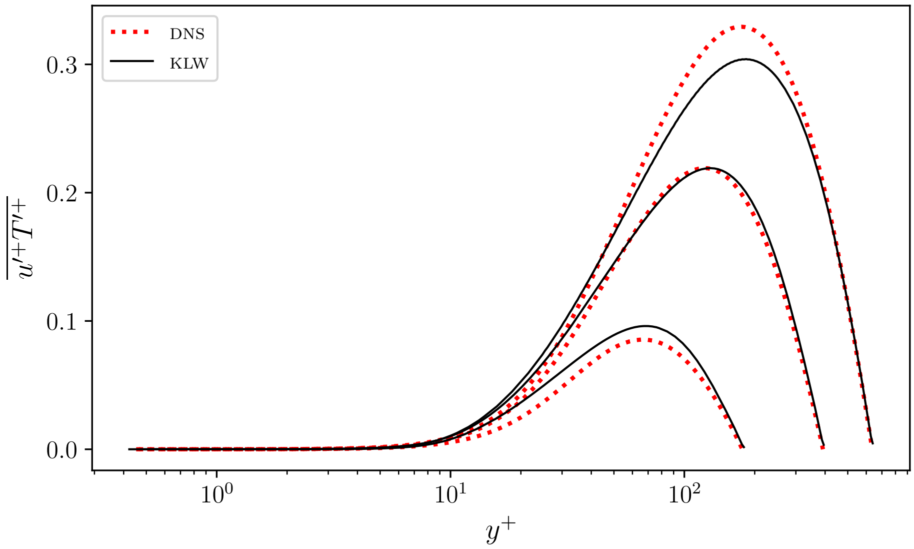


Figure 5.13: Profiles of non-dimensional wall normal turbulent heat flux component $\overline{u'^+T'^+}$ plotted against non dimensional wall distance y^+ and compared with relative DNS data for cases $Re_\tau = 180, 395$ and 640 and $Pr = 0.025$ [86].

the wall, while a better agreement is obtained as wall distance increases after k_θ peak value the positions. Profiles of non-dimensional turbulent heat flux component along wall normal direction

$$q_x^{R+} = \overline{u'^+T'^+} = -\alpha_t \frac{\partial T}{\partial x} \cdot \frac{1}{u_\tau T_\tau}, \qquad (5.24)$$

are plotted in Fig. 5.13 and compared with corresponding DNS values. An overall good agreement is obtained, with a slight overestimation of q_x^{R+} peak value for case $Re_\tau = 180$ and a slight underestimation for case $Re_\tau = 640$.

Results for $Pr = 0.01$

A DNS study on conjugate turbulent heat transfer for fluids with $Pr = 0.01$ has been discussed in [87] for cases $Re_\tau = 180, 395$ and 590. They consider the interaction between a solid, with thermal conductivity λ_s, and a fluid, with thermal conductivity λ_f. Conjugate heat transfer is studied by varying the ratio λ_f/λ_s, with two limiting cases, i.e. $\lambda_f/\lambda_s = 0$ and $\lambda_f/\lambda_s \to \infty$. The first case is representative of ideal non fluctuating temperature boundary condition, while the latter stands for ideal fluctuating temperature boundary condition, i.e. the boundary conditions that have been labeled here as MX and CHF respectively. The boundary conditions used for the present simulations are reported in Tab. 5.4. Values

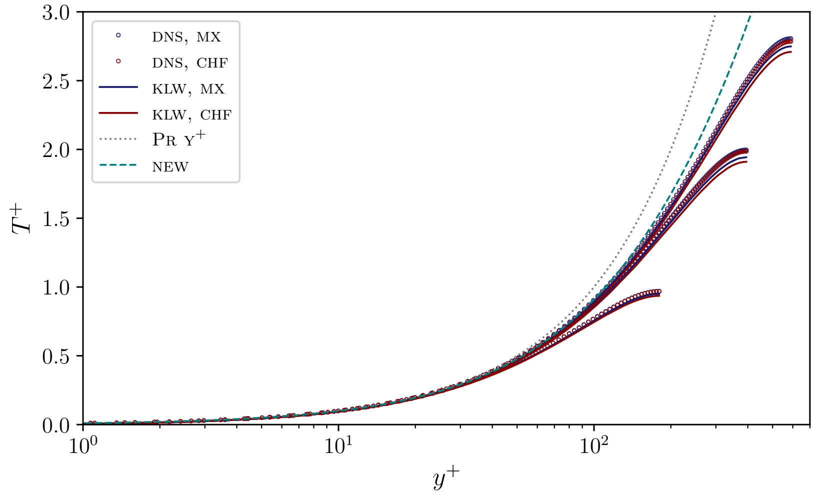


Figure 5.14: Comparison of non-dimensional temperature values, plotted against non dimensional wall distance y^+ and for $Pr = 0.01$, with DNS and LES results [86, 58].

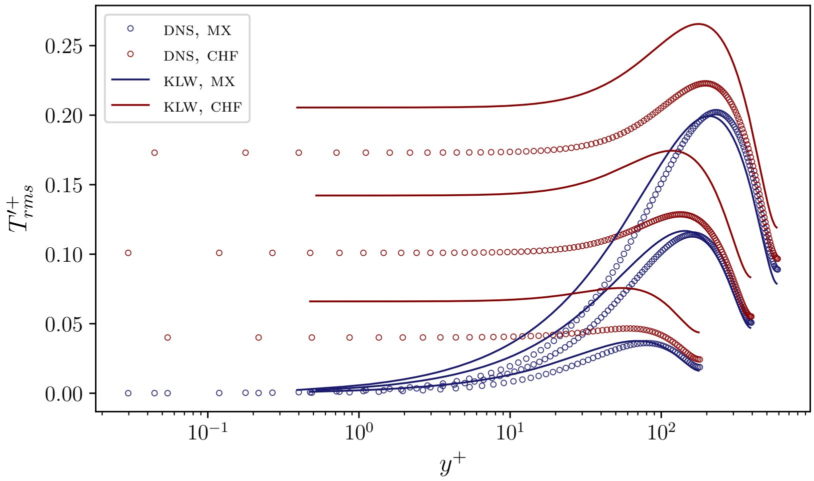


Figure 5.15: Comparison of non-dimensional root mean squared temperature fluctuations T'^+_{rms}, plotted against non dimensional wall distance y^+, with DNS values for $Pr = 0.01$ and cases $Re_\tau = 180$, 395 and 590 [87].

of non-dimensional temperature T^+ are reported in Fig. 5.14 and compared with DNS values, characteristic linear profile and relation (4.53). For these small values of Re_τ and very low value of molecular Prandtl number, characteristic logarithmic profile does not develop. For the case of free temperature fluctuations along the heated wall a slightly smaller mean temperature value is obtained. This behavior

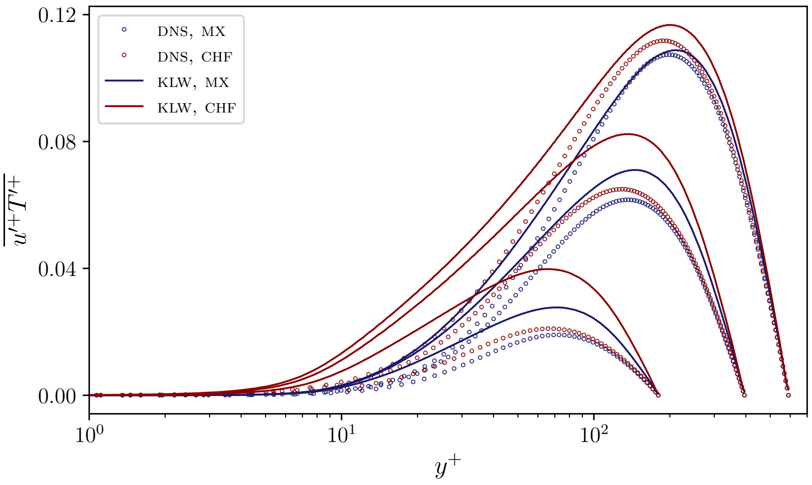


Figure 5.16: Profiles of non-dimensional wall normal turbulent heat flux component $\overline{u'^+T'^+}$, plotted against non dimensional wall distance y^+ and compared with relative DNS data for cases $Re_\tau = 180$, 395 and 590 and $Pr = 0.01$ [87].

is reproduced by the KLW model but a small underestimation of temperature is observed with respect of DNS data. Root mean squared temperature fluctuations are described in Fig. 5.15. For the case of zero temperature fluctuations along the heated wall a good agreement with DNS data is obtained, with T'^+_{rms} profiles similar to those obtained for $Pr = 0.025$. With free temperature fluctuations, almost flat profiles of T'^+_{rms} develop from wall position up to non-dimensional distance $y^+ \simeq 30$. This behavior is captured by the KLW model although temperature fluctuations are overestimated with respect to DNS data. In particular from DNS results it appears that the different boundary conditions have not a great impact on temperature fluctuations values in the bulk region, while sensible higher values are obtained with KLW model. A re-calibration of Ω_θ equation coefficient could lead to future improvement of the model in predicting free temperature fluctuations. Finally non-dimensional turbulent heat flux values are plotted in Fig. 5.16. For MX case a good agreement is obtained, with DNS values, especially for $Re_\tau = 590$ case. As a general behavior, higher values of turbulent heat flux are obtained, leading to higher temperature fluctuations.

5.2.3 Wall function approach

In this section results obtained with wall function approach are presented for case $Re_\tau = 2000$ and $Pr = 0.025$. Accurate flow prediction requires the use of highly refined computational grids near wall boundaries, in order to solve the viscous layer. This requirement can be very demanding even though a RANS simulation

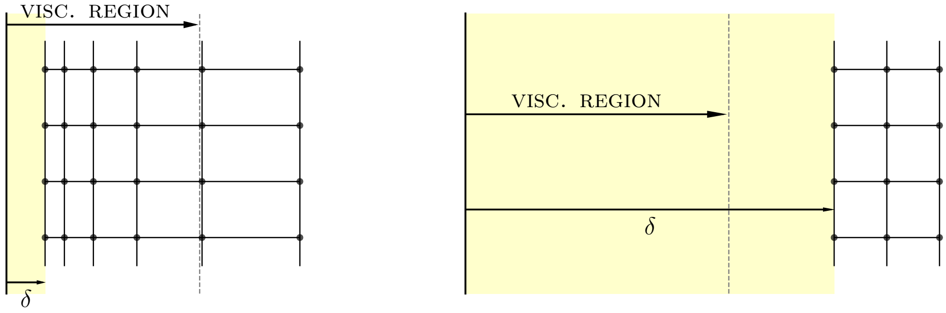


Figure 5.17: Sketch of computational grid for Low-Re approach, on the left, and for wall function approach, on the right. Yellow regions stand for non-solved physical part of the domain.

Variable	Boundary condition	
$\mathbf{u}$	$\nu \boldsymbol{\nabla} \mathbf{u}_t \cdot \hat{\mathbf{n}} = u_\tau^2$	
K	$y^+ < 10 :\ \boldsymbol{\nabla} \mathrm{K} \cdot \hat{\mathbf{n}} = 2/\delta$	$y^+ > 10 :\ \boldsymbol{\nabla} \mathrm{K} \cdot \hat{\mathbf{n}} = 0$
Ω	$y^+ < 5 :\ \Omega = \ln(2\nu/(C_\mu \delta^2))$	$y^+ > 5 :\ \Omega = \ln(u_\tau/(\kappa\sqrt{C_\mu}\delta))$

Table 5.7: Boundary conditions used for the dynamical sets of variable, i.e. $\mathbf{u}$, K and Ω.

Variable	Boundary condition	
T	$T = T_\tau 2 \cdot \ln(1 + \kappa Pr y^+/2)/\kappa$	
K_θ	$y^+ < 10 :\ \boldsymbol{\nabla} \mathrm{K}_\theta \cdot \hat{\mathbf{n}} = 2/\delta$	$y^+ > 10 :\ \boldsymbol{\nabla} \mathrm{K}_\theta \cdot \hat{\mathbf{n}} = 0$
Ω_θ	$y^+ < 5 :\ \Omega_\theta = \ln(2\alpha/(C_\mu \delta^2))$	$y^+ > 5 :\ \Omega_\theta = \ln(u_\tau/(R\kappa\sqrt{C_\mu}\delta))$

Table 5.8: Boundary conditions used for the thermal sets of variable, i.e. T, K_θ and Ω_θ.

is being performed instead of a LES or DNS one. Wall function approaches have been developed to soften the level of refinement needed for the computational grid, providing a strategy for more "economical" computational simulations of turbulent flows. Many wall function approaches have been developed, based on the hypothesis that inside the viscous layer and also in the logarithmic one, state variable behaviors can be predicted using universal laws [51]. In the present formulation wall function approach is used to "cut" part of the physical domain from the simulated one. In Section 4.4 it has already been introduced the fact that with the four logarithmic parameter turbulence model boundary conditions are placed at a distance δ from real wall boundaries because both logarithmic values of k and ω, but also k_θ and ω_θ for MX b.c., are not bounded on wall surfaces. Boundary conditions have then been provided using $\mathbf{u}$, T and κ-Ω-κ_θ-Ω_θ behaviors in the viscous layer. Different boundary conditions must be used when wall distance δ is high enough so that the first mesh point lays outside the viscous region, as sketched in Fig. 5.17. The boundary condition used for the dynamical set of variables are reported in Tab. 5.7 and are derived from the characteristic behaviors of $\mathbf{u}$, κ and Ω in both viscous and logarithmic regions that have been observed in Fig. 5.7. More difficult to determine are the boundary conditions for the thermal system of variables T, κ_θ and Ω_θ. A Dirichlet boundary condition is imposed for the temperature field by using (4.53) since this relation well approximates temperature profile over a wide range of y^+ values. For κ_θ zero gradient along wall normal direction is imposed while for Ω_θ a Dirichlet boundary condition is set using Ω value and the thermal to dynamical time scale ratio R, with $R = 0.5$. The boundary conditions are summarized in Tab. 5.8. A precise estimation of the friction velocity u_τ is important for setting correctly the boundary conditions with wall function approach. For a given value of wall distance δ, kinematic viscosity ν and tangential velocity boundary value $u_b = \left|\mathbf{u}_\Gamma \cdot \hat{\mathbf{t}}\right|$, friction velocity can be calculated from linear and characteristic law as follows

$$u_{\tau\, lin} = \sqrt{\frac{u_b \nu}{\delta}}\,, \tag{5.25}$$

$$u_{\tau\, log} = \frac{u_b}{\dfrac{1}{\kappa}\ln\left(\dfrac{u_{\tau\, log}\delta}{\nu}\right) + B}\,. \tag{5.26}$$

As already mentioned, linear and logarithmic behaviors are observed in region $y^+ \in [0, 5]$ and $y^+ > 40$ respectively. If the first mesh point lays in the intermediate region, $y^+ \in [5, 40]$, then a different relation than (5.25 - 5.26) must be used. For this reason the Musker law is introduced [92]. A velocity law is derived starting from a model for ν_t. Eddy kinematic viscosity is known to grow proportionally to cubic power of y^+, in viscous region, and as κy^+ in logarithmic region, while the behavior in the buffer layer is not known. A blending between cubic and linear laws is then proposed for the numerical evaluation of ν_t/ν as an harmonic mean between these two reference behaviors

$$\frac{\nu}{\nu_t} = \frac{1}{C{y^+}^3} + \frac{1}{\kappa y^+}\,, \tag{5.27}$$

where C is a constant value. Equation 5.27 can be written into 4.41, leading to the following non-dimensional relation

$$\frac{du^+}{dy^+} = \frac{\kappa + C{y^+}^2}{\kappa + C{y^+}^2 + \kappa C{y^+}^3}\,. \tag{5.28}$$

For the calculation of C equation (5.28) is integrated with respect to non-dimensional

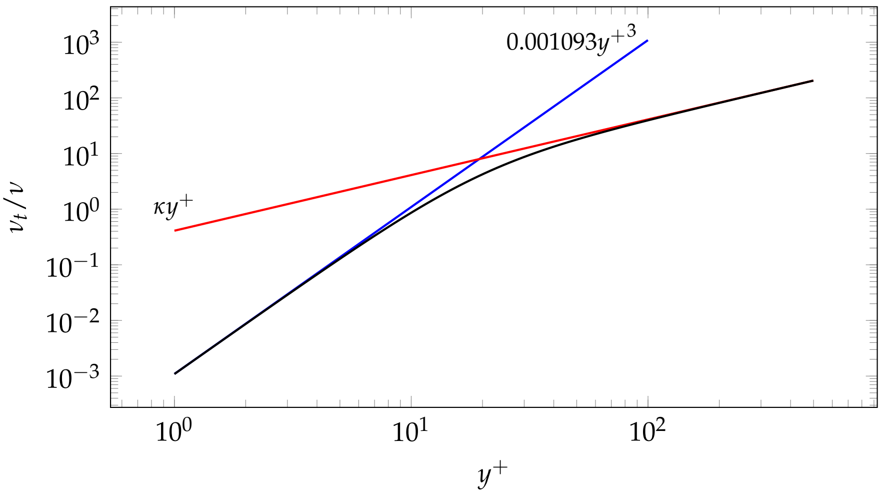


Figure 5.18: Modeled eddy viscosity ratio ν_t/ν with Musker relation (black line) compared with cubic and linear characteristic behaviors, blue and red lines respectively.

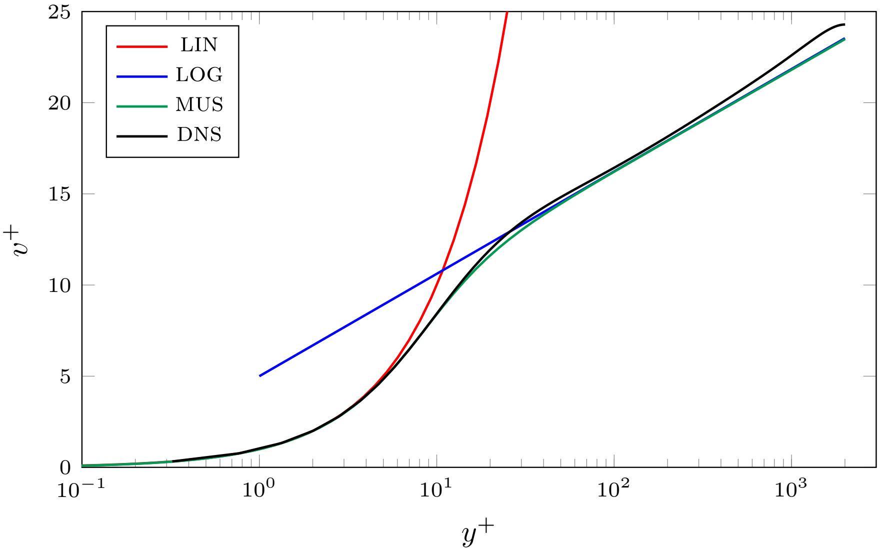


Figure 5.19: Non-dimensional velocity profile modeled with Musker law for ν_t compared with linear and logarithmic behaviors and DNS data for $Re_\tau = 2000$ [88].

wall distance y^+, requiring that for $y^+ \to \infty$ the logarithmic behavior is obtained

$$u^+ = \frac{1}{\kappa} \ln(y^+) + B \,, \qquad \kappa = 0.41 \,, \quad B = 5.0 \,, \tag{5.29}$$

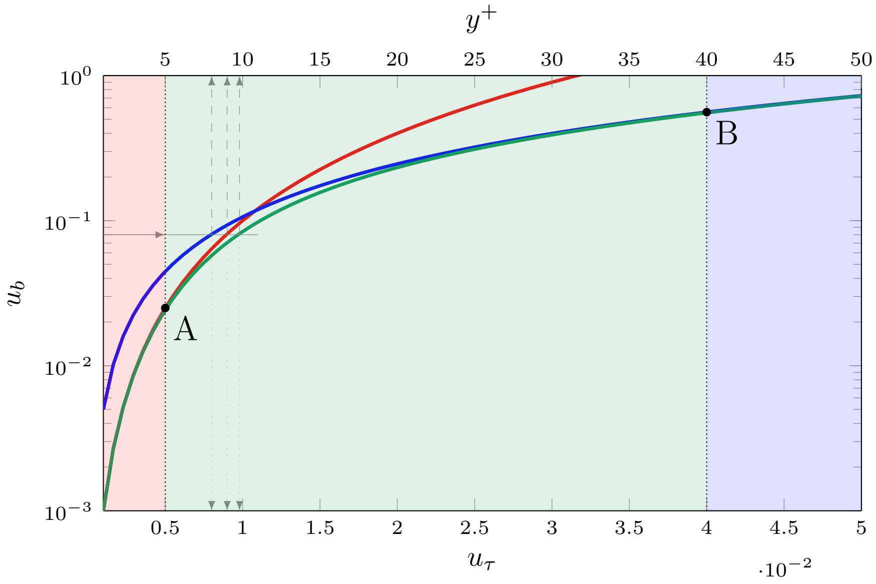


Figure 5.20: Calculation of u_τ using linear, logarithmic and Musker velocity laws, red, blue and green lines respectively.

leading to $C = 0.001093$ [92]. The final form of non-dimensional velocity u^+ is then obtained as

$$v^+ = 5.424 \arctan\left(\frac{2y^+ - 8.15}{16.7}\right) + 4.1693 \ln\left(y^+ + 10.6\right) + \\ - 0.8686 \ln\left({y^+}^2 - 8.15y^+ + 86\right) - 3.52\,. \tag{5.30}$$

Modeled eddy kinematic viscosity with (5.27) is plotted in Fig. 5.18, while velocity profile (5.30) is represented in Fig. 5.19 with characteristic linear and logarithmic laws and DNS results for case $Re_\tau = 2000$. It can be easily seen that Musker law allows to accurately model the velocity profile within the buffer layer. In Fig. 5.20 the results for u_τ are shown. On y-axis the value of velocity modulus u_b on boundary side is calculated as a function of u_τ, $\nu = 10^{-7}$ and $\delta = 10^{-4}$, with the following expressions

linear $$u_b = \frac{u_\tau^2 \delta}{\nu}\,, \tag{5.31}$$

Musker $$u_b = u_\tau \mathcal{M}(u_\tau, \delta, \nu)\,, \tag{5.32}$$

logarithmic $$u_b = u_\tau \left(\frac{1}{\kappa} \ln \frac{u_\tau \delta}{\nu} + B\right)\,, \tag{5.33}$$

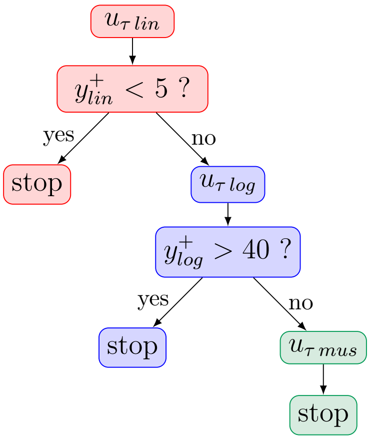


Figure 5.21: Algorithm for calculation of u_τ.

where $\mathcal{M}(u_\tau, \delta, \nu)$ represents equation (5.30). For a given value of u_b, three different values of u_τ can be calculated, i.e. (5.25-5.26) and

$$u_{\tau\, mus} = \frac{u_b}{\mathcal{M}(u_\tau, \delta, \nu)} \,. \tag{5.34}$$

Numerical evaluation of u_τ with (5.26-5.34) is iterative as many nonlinear terms are present and a direct calculation is not possible. It could be stated that Musker relation alone could be used to handle u_τ calculation, but because of its more complicate expressions it is used only when really needed. The algorithm used for u_τ is sketched in Fig. 5.21. Friction velocity $u_{\tau\, lin}$ is always solved and it is used as final u_τ value if the $y^+_{lin} = \delta u_{\tau\, lin}/\nu < 5$, condition represented with node A in Fig. 5.20. Otherwise, if $y^+_{lin} > 5$ then $u_{\tau\, log}$ is calculated and used as final value of u_τ only if $y^+_{log} = \delta u_{\tau\, log}/\nu > 40$, condition represented with node B in Fig. 5.20. Musker velocity law $\mathcal{M}(u_\tau, \delta, \nu)$ is used when $y^+_{lin} > 5$ and $y^+_{log} < 40$. Results obtained with the wall function approach are reported in Fig. 5.22 for v^+, k^+, ω^+, T^+, k^+_θ and ω^+_θ. One can refer to fully developed channel flow with $Re_\tau = 2000$ and $Pr = 0.025$. Several cases are reported, in terms of y^+ value on the first resolved mesh node, namely $y^+ = 16.4$, 32.3, 64, 124 and 180, and compared with reference solutions previously discussed, with $y^+ = 0.66$. The pressure gradient is determined from (4.37). If physical half channel width is labeled as h, then the following expression is obtained

$$-\frac{\partial P}{\partial x} = \frac{\rho Re_\tau^2 \nu^2}{h^2(h-\delta)} \,. \tag{5.35}$$

For the velocity profiles the reference DNS results are reported. It can be seen that friction velocity is well predicted for each of the simulated cases as velocity profiles start from reference values. The most problematic case is the one with

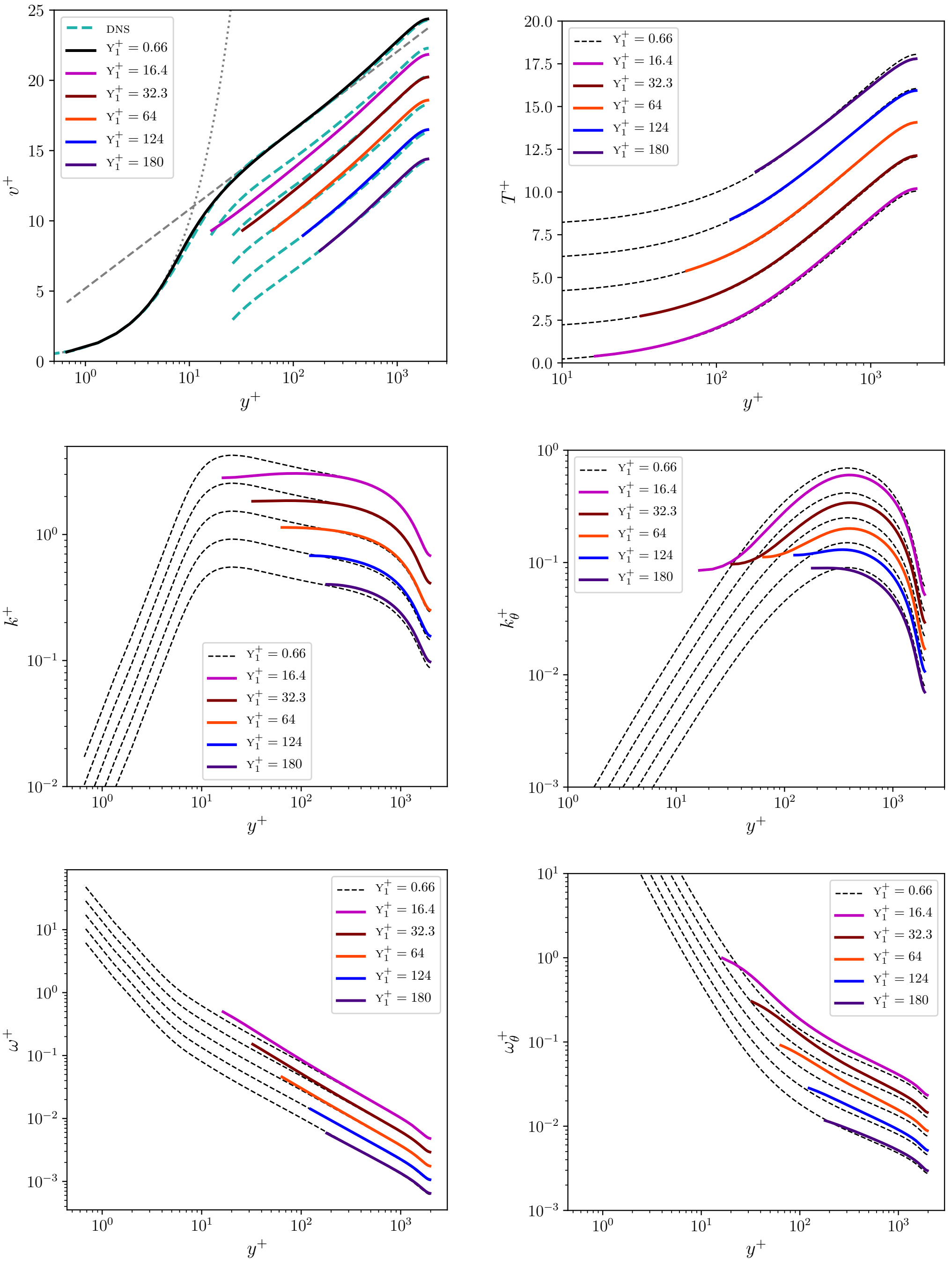


Figure 5.22: Results obtained with wall function approach for plane channel case $Re_\tau = 2000$ and $Pr = 0.025$. Variable profiles are translated to obtain better visibility.

$y^+ = 16.4$ as within the buffer layer k changes greatly from the parabolic behavior to the slightly decreasing one and ω cannot be predicted yet with its characteristic behavior of logarithmic region. The boundary condition imposed on ω is slightly overestimating ω value with respect to the reference case with $y^+ = 0.66$ and k values are under-predicted. Since y^+ value of the first mesh point increases, a better agreement is obtained between wall function approach and fully resolved viscous layer. Temperature equation is first considered. In order to solve a simplified case, temperature field has been transformed into a cross section temperature field $\tilde{T}$, hereinafter referred to as T, so that periodic boundary conditions can be imposed. As a result of the field transformation, a source term $\mathcal{S}_u$ has been added to temperature equation, namely

$$\mathcal{S}_u = \mathbf{u} \cdot \hat{\mathbf{a}} \frac{4 q f_h}{w_{mid} \rho C_p D_h} , \tag{5.36}$$

where w_{mid} is the averaged mean velocity along axial direction. Mean velocity w_{mid} should be calculated over the entire “physical” domain and not restricted on the computational domain. Musker velocity profile has been used to model the velocity field of the non-solved layer, so, once that u_τ has been calculated, w_{mid} is obtained as

$$w_{mid} = \frac{1}{h} \left[\nu \int_0^{y_1^+} \mathcal{M}(y^+) dy^+ + \int_\delta^h u dy \right] . \tag{5.37}$$

It is clear that for more complex geometries this approach is difficult to use. The results of the temperature field are in good agreement with reference values and the effect of zero gradient boundary condition can be seen on k_θ values. Close to the boundary k_θ reaches an almost flat profile and then recovers its behavior. Values of ω_θ are slightly overestimated over the entire simulated domain. Determination of ω_θ characteristic behavior is sure a key point for the realization of a more accurate wall function approach. Future studies should be performed in this direction, in order to provide a turbulence model that can be used over a wide range of flow configurations.

5.3 Cylindrical pipe

In the present section results obtained with the KLW model for the simulations of fully developed turbulent flows in a cylindrical pipe are reported. The cylinder has a diameter D = 0.0605 m and the cylinder axis is located at $r = 0$, where r is the radial coordinate, as shown in Fig. 5.23. The problem is axially symmetric

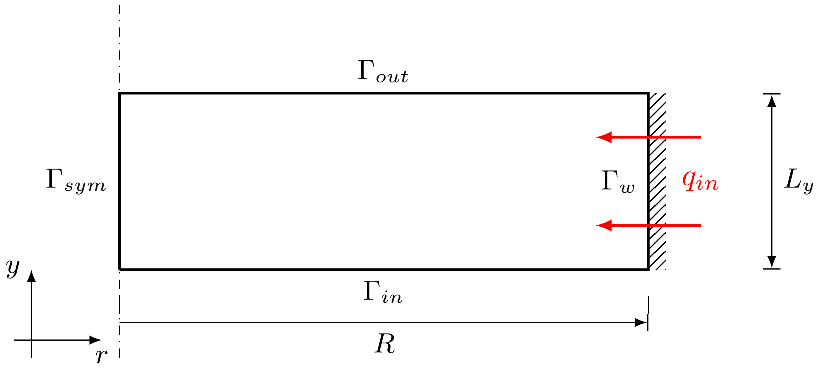


Figure 5.23: Sketch of the simulated domain for the case of fully developed flow in cylindrical pipe.

so the mean velocity, the mean temperature and all the turbulence model variables depend only on the radial coordinate. In the computational domain we identify with Γ_{sym} the symmetry axis, with Γ_w the wall surface and with Γ_{in} and Γ_{out} the inlet and outlet surfaces. Since turbulent thermal field DNS results for low Pr fluids are not available, results are validated on an integral scale by comparing obtained values of Nusselt number with reference correlations. Several fully developed turbulent flow cases have been simulated, in particular with $Re_\tau = 180$ (A), 360 (B), 550 (C), 1000 (D), 3580 (E), 5840 (F) and 6860 (G). The corresponding values of the Reynolds number for these cases are 5760, 12770, 20700, 41000, 165000, 286000 and 341000. For each simulated case both $Pr = 0.01$ and $Pr = 0.025$ values are considered. As in the plane channel case, all the simulations for the cylindrical pipe are performed refining the mesh in the near-wall region in order to have the first mesh point with $y^+ < 1$. A uniform heat flux is applied on wall boundary. Imposed boundary conditions for dynamical and thermal fields are analogous to those used for the plane channel simulations and reported in Tab. 5.2 and Tab. 5.4. For the thermal turbulence variables only MX b.c. have been used.

The Kays correlation has also been considered for modeling the turbulent heat transfer since it is generally suggested for RANS simulations of liquid metal flows [58, 93]. This model can be used to compute the eddy thermal diffusivity α_t by means of a turbulent Prandtl number Pr_t and of eddy viscosity ν_t. The dynamical turbulence model of KLW model is used to calculate ν_t while α_t is computed as

$$\alpha_t = \frac{\nu_t}{Pr_t}\,, \qquad\qquad Pr_t = 0.85 + \frac{0.7}{\frac{\nu_t}{\nu}} \tag{5.38}$$

where ν_t/ν is the eddy viscosity ratio and Pr_t is the turbulent Prandtl number modeled with Kays correlation. The main advantage of the model is that it does not require the solution of additional transport equations and that Pr_t is a function of local parameters, instead of being a uniform value as in classical Simple Eddy Diffusivity model. On the other hand, the major limitation of the model is represented by the fact that eddy thermal diffusivity α_t is still calculated as a function of only dynamical turbulence variables.

The heat transfer between a wall surface and a fluid flow is usually evaluated through Nusselt number. This is a non-dimensional parameter that takes into account the thermal conductivity λ of the fluid, the hydraulic diameter D_h of the transverse section of the pipe and the convective heat transfer coefficient h. The Nusselt number is then calculated as

$$Nu = \frac{hD_h}{\lambda} \,. \tag{5.39}$$

For the case of a constant heat flux per unit surface q applied on wall surfaces, the convective heat transfer coefficient can be expressed as

$$h = \frac{q}{T_w - T_b} \,, \tag{5.40}$$

where T_w is the wall surface temperature and T_b is a representative temperature of the fluid. Normally T_b is the bulk temperature of the fluid

$$T_b = \frac{\int_A T\mathbf{u} \cdot \hat{n} dA}{\int_A \mathbf{u} \cdot \hat{n} dA} \,, \tag{5.41}$$

where $\hat{n}$ is the unit vector normal to the transverse section A of the pipe. In literature many correlations based on experimental results of liquid metal flows in cylindrical geometry for the calculation of the Nusselt number are available. The general form of these correlations is

$$Nu = A + a\,Pe^n \,, \tag{5.42}$$

where Pe is the Peclet number and A, a and n are constant positive numbers. We report here some of the main correlations for the case of cylindrical pipe heated with constant heat flux with their range of validity expressed in terms of Reynolds number value or Peclet value.

$$Nu = 7.0 + 0.025 \left(\frac{Pe}{Pr_t}\right)^{0.8} \qquad 10^4 \leq Re \leq 5 \cdot 10^6 \,, \tag{5.43}$$

$$Nu = 4.82 + 0.0185 Pe^{0.827} \qquad 10^4 \leq Re \leq 5 \cdot 10^6 \,, \tag{5.44}$$

$$Nu = 6.3 + 0.0167 Pe^{0.85} Pr^{0.08} \qquad 10^4 \leq Re \leq 5 \cdot 10^6 \,, \tag{5.45}$$

$$Nu = 4.5 + 0.014 Pe^{0.8} \qquad 10^4 \leq Re \leq 5 \cdot 10^6 \,, \tag{5.46}$$

$$Nu = 3.6 + 0.018 Pe^{0.8} \qquad 88 \leq Pe \leq 4000 \,. \tag{5.47}$$

$$Nu = 4.5 + 0.018 Pe^{0.8} \qquad 10^4 < Re < 5 \cdot 10^6 \,. \tag{5.48}$$

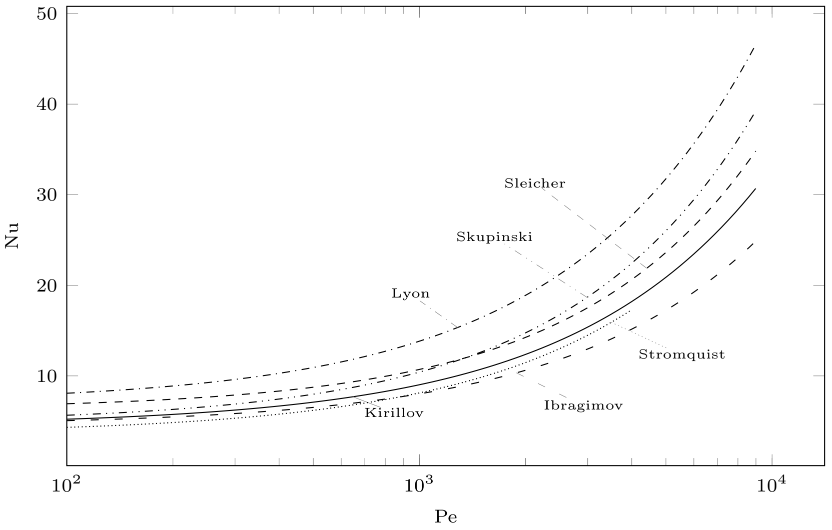


Figure 5.24: Cylindrical pipe heated with constant heat flux. Representation of the Nusselt number experimental correlations (5.43)–(5.48) as a function of the Peclet number.

All these correlations (5.43)–(5.48) are plotted in Fig. 5.24. The correlation proposed by Lyon (5.43) is one of the first correlations for liquid metals and uses the turbulent Prandtl number Pr_t [94, 95]. In Fig. 5.24 the Lyon's correlation is plotted with $Pr_t = 0.9$. As we can see the correlation gives values of the Nusselt number which are much greater than the ones of the other correlations. The Skupinski and Sleicher correlations (5.44, 5.45) were obtained using experimental heat transfer data of NaK [96, 97]. Ibragimov derived (5.46) using experimental data of LBE heat transfer while (5.47) is based on values of heat transfer obtained using mercury by Stromquist [98, 99]. Recently Kirillov proposed a new correlation based on a long analysis on existing correlations and experimental data as (5.48) [100].

As it can be seen from Figure 5.24 there is an overall disagreement between the different correlations that have been proposed for the Nusselt number. An explanation for these discrepancies is given in [91]. By a comparison with the only freely available experimental data present in literature [91, 101] it is shown that the Kirillov correlation is the one that fits better the experimental points in the low Peclet region ($Pe < 1000$), while in the high Peclet region ($Pe > 2000$) the Stromquist correlation shows a better agreement with the experimental values. For this reason the authors in [91] proposed a new correlation based on a fit of

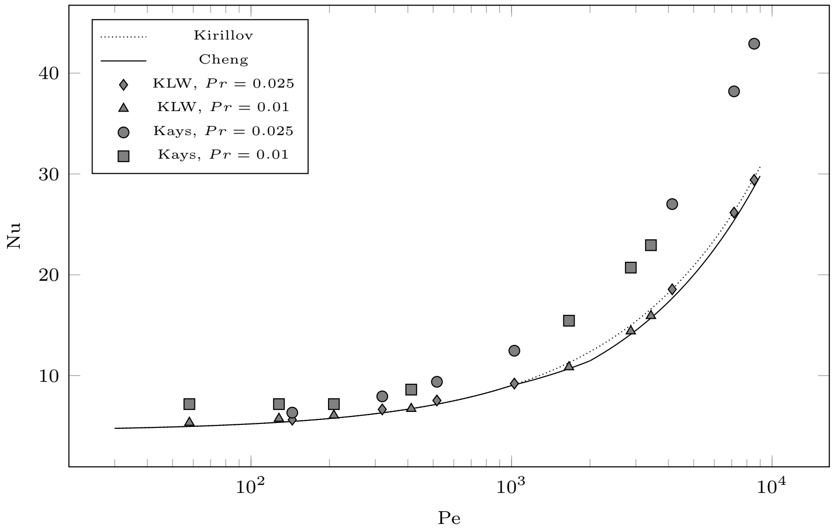


Figure 5.25: Cylindrical pipe. Nusselt number values for the simulations performed for fluids with $Pr = 0.025$ and $Pr = 0.01$ with KLW and Kays model. The values are compared with the Kirillov correlation and with the Cheng correlation.

these two correlations in the range of their best approximation,

$$\begin{aligned} &Nu = A + 0.018Pe^{0.8}\,, \\ &A = \begin{cases} 4.5 & Pe < 1000\,, \\ 5.4 - 9\times 10^{-4}Pe & 1000 \le Pe \le 2000\,, \\ 3.6 & Pe > 2000\,. \end{cases} \end{aligned} \tag{5.49}$$

This new correlation is equal to the Kirillov one in the low Peclet region and it is equal to the Stromquist in the high Peclet region, with an extension of its validity beyond $Pe = 4000$.

In Figure 5.25 the results obtained with the KLW and with Kays model for the simulations of fully developed turbulent flows of fluids with $Pr = 0.025$ and $Pr = 0.01$ are compared with the Kirillov correlation and the one reported in [91]. For a better interpretation of the results the obtained Nusselt number values are also reported in Table 5.9 along with those of Kirillov and Cheng correlations and compared for the different cases in terms of Reynolds number. As it can be seen, for the case of $Pr = 0.025$ the values of the KLW model are closer to the Kirillov correlation than to the Cheng correlation as it concerns $Pe > 1000$, while the values obtained for $Pr = 0.01$ are closer to the Cheng correlation. In general it is observed that the values obtained with the KLW model lay between the Kirillov and the Cheng correlations, in the high Peclet region, while in the low

Pr	Source	Reynolds number						
		$3.4 \cdot 10^5$	$2.85 \cdot 10^5$	$1.65 \cdot 10^5$	$4.1 \cdot 10^4$	$2.1 \cdot 10^4$	$1.27 \cdot 10^4$	$5.76 \cdot 10^3$
0.025	KLW	29.40	26.16	18.56	9.21	7.53	6.65	5.63
	Kays	42.91	38.18	27.01	12.40	9.38	7.94	6.33
	Kirillov	29.62	26.30	18.57	9.11	7.16	6.31	5.45
	Cheng	28.72	25.40	17.67	9.08	7.16	6.31	5.45
0.01	KLW	15.93	14.40	10.84	6.72	6.05	5.74	5.34
	Kays	22.93	20.71	15.44	8.61	7.17	6.50	5.71
	Kirillov	16.61	15.00	11.28	6.72	5.78	5.37	4.96
	Cheng	15.71	14.10	10.69	6.72	5.78	5.37	4.96

Table 5.9: Nusselt number values obtained with the KLW and Kays model for $Pr = 0.025$ and $Pr = 0.01$ compared with the Kirillov and the Cheng correlation.

Property	Symbol	Dimension	Value
Pitch to diameter ratio	χ	-	1.4
Rod diameter	d	[mm]	9.2
Bundle hydraulic diameter	$d_{h,bun}$	[mm]	7.7
Sub-channel hydraulic diameter	$d_{h,sch}$	[mm]	9.5
Minimum rod to adiabatic wall distance	δ_{min}	[mm]	1.716
Heated to wet perimeter ratio	f	-	0.733

Table 5.10: Geometrical parameters of the simulated 19 pin nuclear reactor bundle.

Peclet region they are slightly greater than the Kirillov correlation. As it regards the results obtained with Kays model, Nu values are generally higher than those obtained with KLW model and than reference ones, with bigger discrepancies as Peclet values increase. This fact can be explained by looking at the definition (5.38): if the eddy viscosity ratio becomes very high, as it happens for high Peclet numbers, the average turbulent Prandtl number of the channel as computed in (5.38) approaches 0.85 which is known to be not valid for liquid metal flows. Given the inability of Kays model to reproduce the integral heat transfer in this very simple geometry it is not recommended to use this model in the simulation of liquid metal turbulent flows.

5.4 Nuclear reactor bundle

In the present Section the KLW model is used to study a 19 pin nuclear reactor bundle whose cross section is sketched in Fig. 5.26 a). Many works involving numerical simulations of turbulent heat transfer in nuclear reactor bundles can be found in literature for triangular [11, 80] and square [12, 80] lattices. Hexagonal

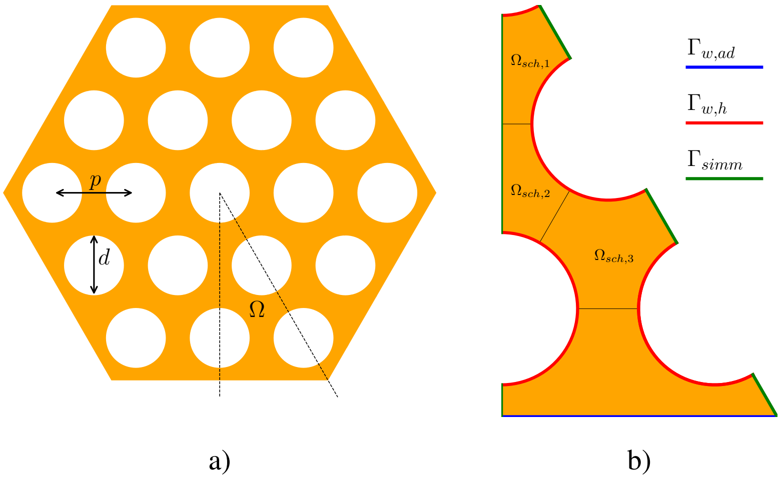


Figure 5.26: Schematic representation of whole bundle cross section a) and simulated domain, boundary definition and triangular sub-channels b).

Boundary	$\mathbf{u}$	K & Ω	$\tilde{T}$	K_θ & Ω_θ
$\Gamma_{w,h}$	no-slip	near-wall b	uniform heat flux q	near-wall b
$\Gamma_{w,ad}$	no-slip	near-wall b	null gradient	null gradient
$\Gamma_{w,simm}$	null gradient	null gradient	null gradient	null gradient

Table 5.11: Boundary conditions imposed on boundaries represented in Fig. 5.26 b).

bundle, in lattices of 19 or more rods, are studied in bare, wire-wrapped and with spacer grid configurations [102, 103, 104, 47]. Fully developed turbulent flows occurring in a bare rod lattice configuration are here considered, in order to provide results in a simplified geometry that can be used as a starting point for future analysis. The computational domain can then be reduced to a fraction of the whole geometry by taking advantage of the presence of symmetry planes, as shown in Fig. 5.26 b), where a sketch of simulated geometry is represented together with the different types of boundary surfaces, namely symmetry planes, heated walls and adiabatic wall. Rod lattices are usually classified using the pitch-to-diameter ratio χ, where pitch p is the distance between two adjacent rods center lines and diameter d is the rod diameter, as sketched in Fig. 5.26 a). Within the simulated geometry three different triangular sub-channels can be depicted, as done in Fig. 5.26 where they are labeled as $\Omega_{sch,1}$, $\Omega_{sch,2}$ and $\Omega_{sch,3}$. It is interesting to investigate the influence of adiabatic wall on sub-channel performances in terms of heat transfer. Numerical values of the bundle geometrical properties are reported in Table 5.10 and refer to experimental setup studied in [102, 47]. In particular rod

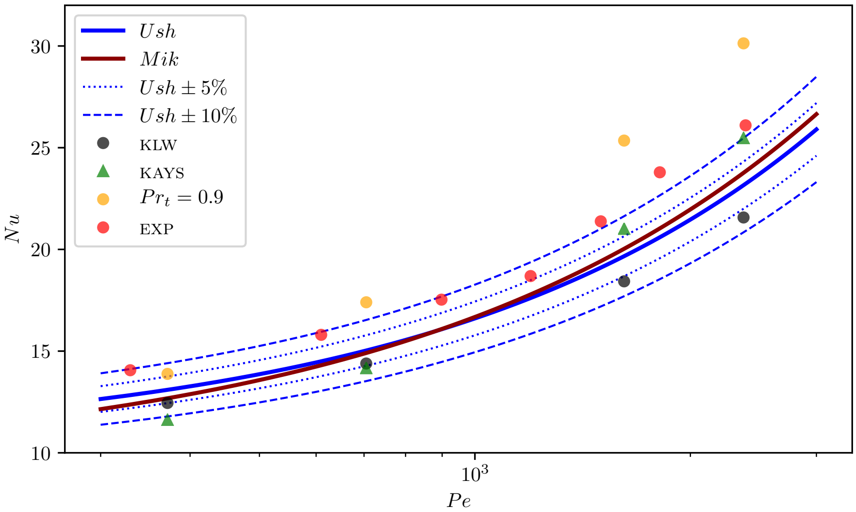


Figure 5.27: Comparison of computed Nusselt values with experimental correlations of Ushakov and Mikityuk, experimental values from [102] and values obtained with Kays and $Pr_t = 0.9$ models for turbulent Prandtl number.

diameter d is equal to 8.2 mm, with a pitch to diameter ratio $\chi = 1.4$. Adiabatic wall is located at a distance δ_m from heated rods, leading to a bundle hydraulic diameter $d_{h,bun} = 7.7$ mm, while the triangular sub-channel hydraulic diameter is $d_{h,sch} = 9.52$ mm. The ratio of heated to wet perimeter f is equal to 0.733 and it is used to calculate the source term in $\tilde{T}$ temperature equation. The simulated coolant fluid is Lead Bismuth Eutectic with a molecular Prandtl number $Pr = 0.025$.

5.4.1 Results

In order to compare obtained results with literature data, two different Reynolds numbers are introduced, namely the bulk Reynolds number $Re_b = u_b d_{h,bun} \rho/\mu$, based on bundle hydraulic diameter, and sub-channel Reynolds number $Re_{sch} = u_b d_{h,sch} \rho/\mu$, based on sub-channel hydraulic diameter. Turbulent heat transfer occurring inside the bundle geometry is studied from the solution of system of equations (5.4–5.6). The boundary conditions used for the present simulations are reported in Table 5.11, where “near-wall b” means that the imposed boundary conditions are in accordance with the near-wall behavior. Differently from the case of plane channel and cylindrical pipe, a Dirichlet boundary condition on temperature field $\tilde{T}$ cannot be imposed on heated walls as temperature field is not uniform. A Neumann boundary condition is then used, by imposing the heat flux q on heated walls. Resulting temperature field is obtained up to an arbi-

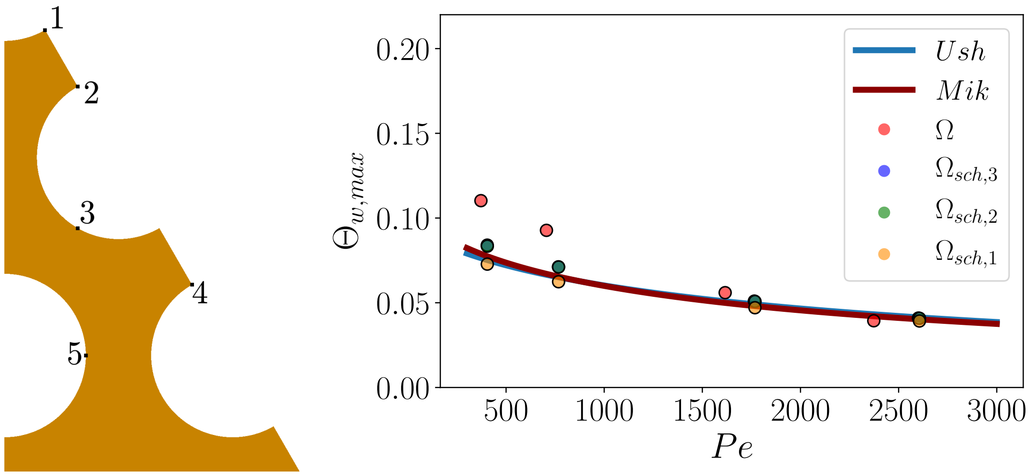


Figure 5.28: Maximum non-dimensional wall temperature values: thermocouple positions of [102], on the left, and comparison of values from triangular sub-channels, whole geometry and inverse value of Ushakov and Mikityuk correlations.

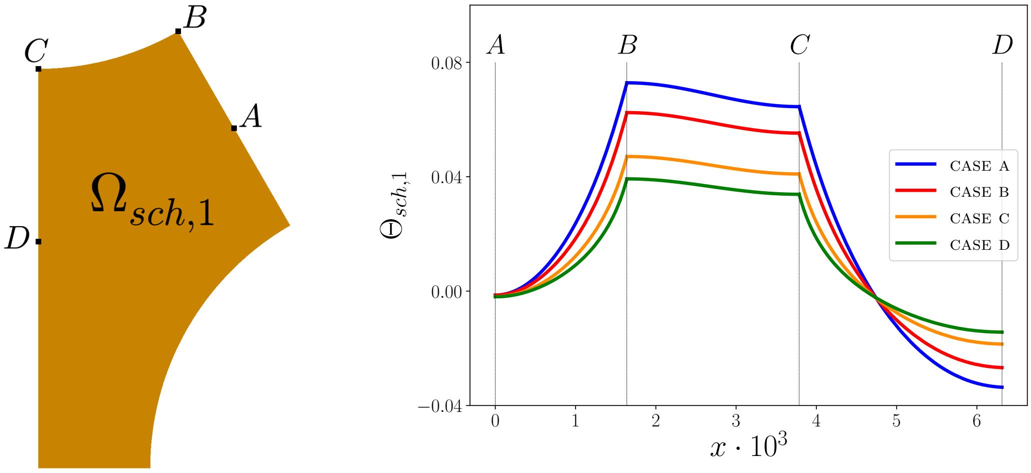


Figure 5.29: Triangular sub-channel $\Omega_{sch,1}$ with reference points A, B, C and D, on the left, and non-dimensional temperature field $\Omega_{sch,1}$ over path $A-B-C-D$ for the four simulated cases.

trary constant value. For thermal turbulence variables MX boundary conditions are imposed. Four different values of bulk Reynolds number are examined, namely CASE A : $Re_b = 1.4 \times 10^4$, CASE B : $Re_b = 2.2 \times 10^4$, CASE C : $Re_b = 3.1 \times 10^4$ and CASE D : $Re_b = 5.4 \times 10^4$. In order to compare the results with reference data we follow the methods used in [102, 47]. Nusselt number is thus calculated using sub-channel hydraulic diameter,

$$Nu = \frac{q d_{h,sch}}{\lambda(\langle T_w \rangle - T_b)} , \tag{5.50}$$

where λ is the fluid thermal conductivity and $\langle T_w \rangle$ is the bundle mean temperature over heated walls. Bulk temperature T_b is computed as

$$T_b = \frac{\int_{\mathcal{S}} T \mathbf{u} \cdot \hat{\mathbf{n}} \, dS}{\int_{\mathcal{S}} \mathbf{u} \cdot \hat{\mathbf{n}} \, dS} , \tag{5.51}$$

where $\mathcal{S}$ is the cross-section surface. Nusselt number values are compared with Ushakov experimental correlation [105], as done in [11]

$$Nu = 7.55\chi - 20\chi^{-13} + \frac{3.67}{90\chi^2} Pe^{0.56+0.19\chi} . \tag{5.52}$$

This correlation is used to predict Nusselt number values in regular rod lattices with pitch-to-diameter ratio values $\chi \in [1.3, 2.0]$ and for values of Peclet number smaller than 4000. Mikityuk correlation [106] is used in [102, 47] to compare experimentally obtained Nu values. The correlation is formulated as

$$Nu = 0.047(1 - e^{-3.8(\chi-1)})(Pe^{0.77} + 250) , \tag{5.53}$$

and is valid for Pe in the range $[30, 5000]$ and for a slightly wider range of χ, namely $[1.1, 1.95]$. Obtained results are shown in Fig. 5.27, where they are plotted against Peclet number based on sub-channel Reynolds number. In particular Pe values are $Pe = 372$ for CASE A, $Pe = 705$ for CASE B $Pe = 1616$ for CASE C and $Pe = 2373$ for CASE D. In the Figure, KLW results (with black markers) are compared with both Mikityuk and Ushakov correlations, experimental values and others obtained with two different turbulent Prandtl models, i.e. Kays model and the more common $Pr_t = 0.9$ model. Results obtained with KLW model show a well defined trend being approximately 5% smaller than Ushakov correlation. Kays results lay in the range of $\pm 10\%$ of Ushakov values, with an under estimation at low Pe values and an over estimation in high Pe region. Following the trend of obtained results, for higher Pe values Kays results would seem to lay outside $\pm 10\%$ range of Ushakov correlation. Turbulent Prandtl number equal to 0.9 clearly leads to a sensible overestimation of heat transfer, with obtained Nusselt number values much bigger than reference correlation. Experimental values from [102] predict higher Nu values. As explained by the authors, this could be related to measuring apparatus, as it was located on spacer grids. Having a solidity value $\epsilon = 0.29$, defined as the ratio between grid cross section to bundle cross section, the presence of spacer grids leads to a sensible fluid acceleration with consequent heat exchange enhancement. In [102] it has been seen that Mikityuk correlation can be used to estimate the maximum wall super-heat, i.e. the maximum value of non-dimensional temperature Θ,

$$\Theta = \frac{(T - T_b)\lambda}{d_{h,sch} q} \,. \tag{5.54}$$

In Fig. 5.28 the maximum wall super-heat is analyzed. In particular non-dimensional temperature Θ values are observed from points representing thermocouple locations of [102]. The maximum Θ value among these points is compared, with red dots, with inverse values of both Ushakov and Mikityuk correlations. Moreover for the triangular sub-channels three different non-dimensional Θ fields are also considered, namely $\Theta_{sch,1}$, $\Theta_{sch,2}$ and $\Theta_{sch,3}$, calculated using a sub-channel bulk temperature. In this way the sub-channels are analyzed as standalone channels of a triangular rod lattice. The influence of adiabatic wall presence can then be studied. Maximum values of non-dimensional temperature $\Theta_{sch,i}$ are obtained among wall temperature values of each sub-channel. In the small Pe values region sub-channels $\Omega_{sch,2}$ and $\Omega_{sch,3}$ have a similar maximum value of $\Theta_{sch,1}$, slightly higher than reference correlations, while sub-channel $\Omega_{sch,1}$ has a smaller value, denoting a more homogeneous temperature field for the inner sub-channel. For all investigated cases maximum values of $\Theta_{sch,1}$ are smaller then reference correlations. As Pe increases the maximum super-heat of sub-channel $\Omega_{sch,2}$ and $\Omega_{sch,3}$ decrease, converging to the value of $\Omega_{sch,1}$. It can be concluded that as Pe values increase the presence of adiabatic wall has a smaller influence on thermal behavior of triangular sub-channels. A similar trend is observed for Θ values computed on the whole geometry. For CASE A and CASE B obtained values are higher than reference correlations. These results have been obtained also in [47], with even higher super-heat values for low Pe, where simulations with code Star-CCM+ have been conducted using $Pr_t = 0.9$. For CASE C and CASE D the obtained super-heat values are below reference correlations and are close to those obtained for sub-channel $\Omega_{sch,1}$. Experimental values discussed in [102, 47] are well represented by Mikityuk correlation over the entire range of examined Pe values. In the present study buoyancy effects are not considered. In the low Pe region it is possible that they influence heat transfer by enhancing coolant fluid motion in the region close to heated walls. This would then lead to smaller wall temperature values and so to smaller super-heat values. A study of $\Omega_{sch,1}$ non-dimensional temperature $\Theta_{sch,1}$ is shown in Fig. 5.29. Four representative points are depicted on $\Omega_{sch,1}$ domain, A, B, C, D, as represented on the left of Fig. 5.29. Non-dimensional temperature $\Theta_{sch,1}$ along path A–B–C–D are plotted on the right of Fig. 5.29. Obtained values can be compared with results discussed in [11], where simulations of triangular rod lattices with $\chi = 1.2$, 1.3 and 1.5 have been performed with a four parameter turbulence model, and in [80], where triangular rod

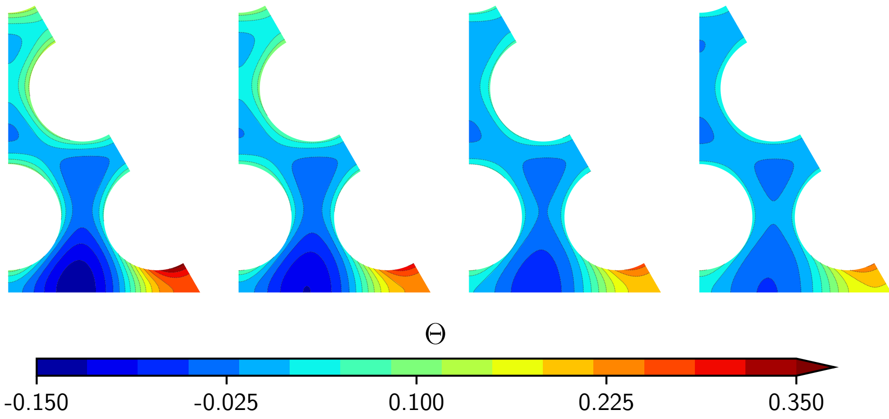


Figure 5.30: Non-dimensional temperature field Θ. From left to right simulated cases A, B, C and D.

lattices with $\chi = 1.2$ and 1.4 have been simulated with several different algebraic models for turbulent Prandtl number. As Pe increases a flatter temperature profile develops. From point A, where T value is close to bulk temperature, temperature increases moving towards the heated rod. A non uniform temperature field is observed along path B–C, with maximum super-heat occurring on point B. From wall position C towards sub-channel middle point D temperature decreases. Values of $\Omega_{sch,1}$ along the same path are shown in [80] for case $Pe = 1440$. Closest to this Pe value is CASE C having $Pe \simeq 1510$. Results are comparable, with smaller values for the case of KLW results, due also to slightly higher Pe value. Non-dimensional temperature field Θ is shown in Fig. 5.30 over the whole simulated domain for each simulated case. It can be seen that temperature measuring positions reported in Fig. 5.28 can be used for a comparison of super-heat values with literature data for triangular sub-channels but are not representative of bundle maximum super-heat locations. Maximum temperature values are in fact observed in the corner region, where cooling fluid velocity values are much smaller. In Fig. 5.31 turbulent Prandtl distribution on the bundle cross section are shown for each of the simulated cases. One can see that high Pr_t values are obtained in the region close to heated walls. Maximum values in those regions are in the range of $[2, 3]$ and tend to decrease as the Reynolds number increases. It is commonly suggested to set $Pr_t = 1.5$ [80] when using an algebraic model for turbulent Prandtl number. This assumption can be a good compromise as in the center of bundle cross section the computed Pr_t is approximately equal to 1.5, but it would lead to an under estimation in the near-wall region, predicting thus a higher turbulent heat exchange.

The present study has provided solutions for a simplified case of turbulent heat

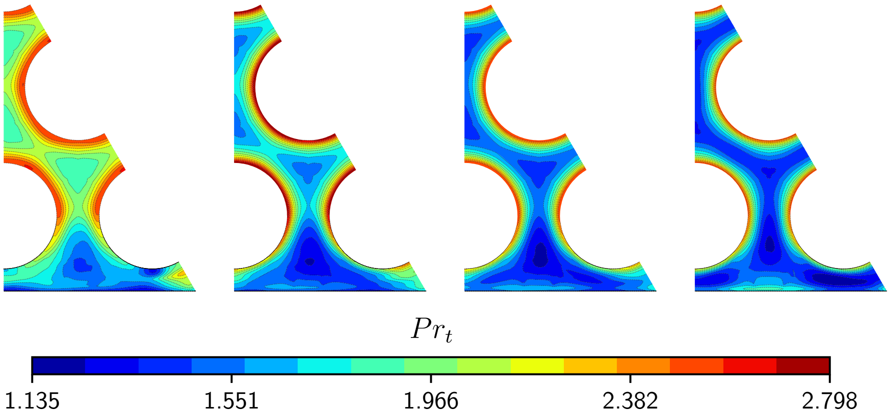


Figure 5.31: Turbulent Prandtl number distribution on the simulated domain for increasing Reynolds number values. From left to right CASE A, CASE B, CASE C and CASE D.

transfer occurring in a 19 pin hexagonal nuclear reactor bundle. As a major simplification fully turbulent flows are considered. The computational domain can be reduced to a fraction of the whole bundle geometry by taking advantage of the presence of symmetry planes. The simulated coolant fluid is Lead Bismuth Eutectic, with a molecular Prandtl number equal to 0.025. Four cases have been simulated with Peclet number values ranging from 370 to 2370, defined using triangular sub-channel hydraulic diameter. Results have been obtained using two algebraic turbulent Prandtl number models, namely $Pr_t = 0.9$ and Kays correlation, and the four logarithmic parameter KLW. Nusselt number values have been compared with reference correlations of Ushakov and Mikityuk, used respectively in [11, 102, 47] to compare results for triangular rod lattices and 19 pin hexagonal bundles. Results of KLW model are about 5% smaller then Ushakov correlation, for all simulated values of Pe. Kays correlation results lay in the range of $\pm 10\%$ of Ushakov correlation values, while model $Pr_t = 0.9$ predicts much higher heat transfer values. Super-heat phenomenon has been investigated with two approaches. In the first wall temperature values have been observed on some selected points of the simulated geometry, in particular on those positions where experimental temperature values have been measured in [102]. Among these values the maximum temperature has been used to calculate non-dimensional temperature $\Theta_{w,max}$ and compare it with reference correlations of Ushakov and Mikityuk, as done in [102]. Experimental temperature values lay close to reference correlation for all investigated Pe numbers. Super-heat values obtained with KLW model are in good agreement with reference correlation, for high Pe values, while for $Pe < 1000$ higher values are obtained. As second approach the three triangular sub-channels have been studied as standalone channels of a triangular rod lattice, by computing local non-dimensional temperature field $\Theta_{sch,i}$ defined with local bulk temperatures. This procedure allows to evaluate the influence of bundle adiabatic wall on heat transfer. As Pe value increases sub-channel super-heat values converge to values of inner sub-channel, showing a smaller influence of the adiabatic wall on sub-channel heat transfer. Future studies will consider the presence of buoyancy forces to see if they are important in the low Pe region. Finally the turbulent Prandtl number distribution obtained with KLW model has been shown. The results are everywhere grater than one, with maximum values $Pr_t \simeq 3$ occurring in the near heated wall region. A fixed Pr_t value equal to 1.5 appears to be a fair approximation only in the bulk region and for relatively high values of Pe.

A more detailed study of this case can be performed in future, in particular by considering the wire wrapped configuration or the presence of grid spacers. For these geometry configurations, the immersed boundary method presented in Chapter 3 can be a suitable method for modeling the wire geometry or the grid spacers, in an hybrid approach with a boundary fitted grid. This strategy would let to create the geometry model in a more simple way and to have a stronger control of mesh resolution close to wall boundaries.

Chapter 6

Turbulent heat transfer over vertical backward facing step

In the last few years, the turbulent flow over a backward facing step, a simple geometry that can be encountered in many engineering applications, has been extensively studied also for low Prandtl number fluids, in particular liquid sodium, in order to investigate the influence of buoyant forces on fluid behavior and heat exchange [107, 108, 76]. In the following, the flow regime will be defined for different values of the Richardson number $Ri = g\beta\Delta T h/U_b^2$, where g is the modulus of gravity acceleration, β the thermal expansion coefficient, ΔT a reference temperature difference, h the step height and U_b the bulk velocity. With $Ri = 0$ we refer to the pure forced convection flow regime. A DNS simulation with $Ri = 0$, is provided in [109], for a particular geometry where a constant heat flux is applied on the whole wall behind the step. For the same simulation case, i.e. geometry

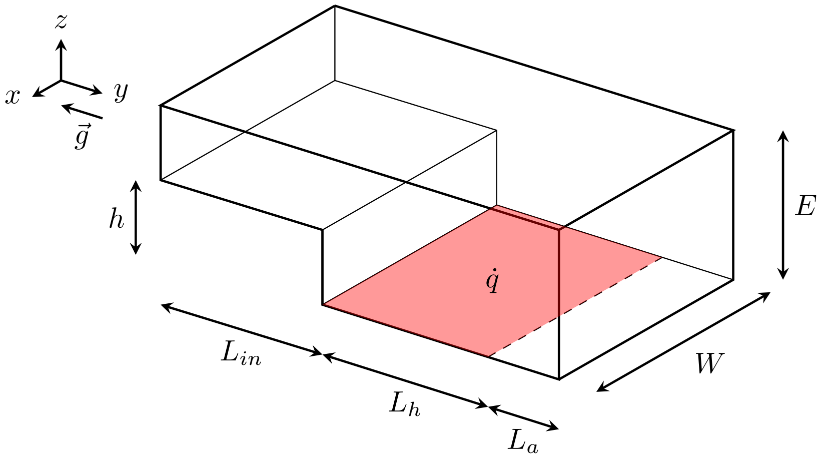


Figure 6.1: Sketch of backward facing step geometry.

configuration and Reynolds number Re, other studies are performed [82, 107, 76]. In [82] a comparison is provided between the results obtained from DNS simulation and from the solution of a Reynolds Averaged Navier Stokes (RANS) system of equations closed with various turbulence models. The results are obtained only for $Ri = 0$ case. It is shown that two equation heat transfer turbulence models, coupled with non linear expressions for Reynolds stresses, allow to improve the predictions of heat flux within the re-circulation zone. In [107] a DNS study is performed for the cases $Ri = 0$ and $Ri = 0.338$. In particular, for the mixed convection case, i.e. $Ri = 0.338$, a different domain configuration is considered as an adiabatic section is added behind the heated wall in order to minimize the influence of the outlet on the flow behavior over the heated wall. In [107] the effect of buoyancy results in a diminished re-circulation length, with respect to the case of pure forced convection, and in an increased heat exchange. The same study, in terms of Reynolds and Richardson numbers, is made in [76], where the turbulent flow is simulated using a linear k-ε-k_θ model with a novel algebraic formulation of turbulent heat flux, used for the first time for the simulation of a relatively more complicated case as the backward facing step. A quasi DNS method has been used to investigate the influence of molecular Prandtl number on forced convection turbulent heat transfer [110]. In particular the considered geometry is the same as above, while for molecular Prandtl number the values $Pr = 0.01$, 0.025, 0.1 and 1.0 are considered. It is shown that for large Pr values the position of maximum heat transfer is located downstream of reattachment point, while for small Pr values it coincides with bigger vortex reattachment point. Effect of buoyancy aided flow and adverse buoyancy force are studied in [111], where values of Richardson number $Ri = -0.04$, $Ri = 0.1$ and $Ri = 0.2$ are taken into account for the simulation of turbulent LBE flow, i.e. with $Pr = 0.025$. The work represents a first attempt to study adverse buoyancy force for low Prandtl number fluid turbulent mixed convection over a backward facing step. It is observed that as Ri increases the secondary vortex increases as well, until it leads to a complete detachment of the bigger vortex from the heated wall. With adverse buoyancy force both skin friction and Nusselt number over the heated wall decrease, on the contrary of what happens with $Ri > 0$. Many of simulations have been performed also for higher Reynolds number, namely $Re = 10000$, for forced and mixed convection cases [112, 108, 83, 84, 113]. In [112, 108, 83] DNS simulations are performed for various values of the Richardson number, ranging from 0 to 0.4, showing that re-circulation length decreases and heat transfer increases within the re-circulation zone as Ri increases. These studies provide useful data for the evaluation of turbulence models. In [83] Large Eddy Simulations are performed for higher Reynolds numbers, namely $Re_h = 20000$ and $Re_h = 40000$. Numerical simulations with RANS models are reported in [84, 113]. In the first work a comparison is performed between the results obtained using a four parameter turbulence model [10,

	L_{in}/h	L_h/h	L_a/h	E_r	W/h	Re_h	Method	Ri
[109]	2	20	0	1.5	4	4805	DNS	0
[82]	2	20	0	1.5	4	4805	DNS/RANS	0
[107]	2	20	10	1.5	4	4805	DNS	0, 0.338
[76]	2	20	10	1.5	4	4805	RANS	0, 0.338
[110]	2	20	0	1.5	-	4805	DNS	0
[111]	2	20	0	1.5	-	4805	DNS	-0.04, 0.1, 0.2
[112]	2	20	10	2	4	10000	DNS	0, 0.12, 0.2
[108]	2	20	10	2	4	10000	DNS	0, 0.12, 0.2, 0.4
[83]	2	20	10	2	1	10000 20000 40000	DNS LES LES	0, 0.2
[84]	4	20	20	2	-	10000	RANS	0, 0.2
[113]	4	20	20	2	-	10000	RANS	0, 0.12, 0.2, 0.4, 1

Table 6.1: Comparison of literature studies on sodium buoyant turbulent flows on backward facing step, as a function of geometrical parameters and of Reynolds and Richardson numbers.

11, 12, 47] and a two equation turbulence model [114], for modeling the Reynolds stresses, and the Kays correlation for modeling the eddy thermal diffusivity [93]. It is shown that similar results are obtained using the two different models, for that particular Reynolds number case. In the latter work a study on a wider range of Richardson number values is performed. Here the RANS system of equation is closed by using a two equation model and the Kays correlation. A comparison between the previously mentioned literature data is reported in Table 6.1, where the backward facing step geometry is parameterized by using the step height h and the length labels for inlet section L_{in}, heated section L_h, adiabatic section L_a, domain width W and downstream channel height E, as sketched in Figure 6.1. We refer to E_r as the expansion ratio, calculated as $E_r = E/(E - h)$. Following this line of research the present Chapter deals with the simulation of turbulent liquid sodium flow over a backward facing step by using the four logarithmic parameter turbulence model. The simulated case is the same studied in [82, 107, 76, 110]. The test case is used to evaluate the accuracy of the KLW model in simulating turbulent heat transfer in a more complex geometry, where recirculation occurs, for forced and mixed convection cases. Moreover a coupling with OpenFOAM is considered in order to use dynamical turbulence models different from the one of the KLW model. In this way, by using the computational platform, a more extensive set of dynamical turbulence models is made available, without the need of implementing it in FEMuS code. In Section 6.1 results are provided and discussed for simulations performed with FEMuS code while in Section 6.2 the coupling between OpenFOAM and FEMuS is discussed. Dynamical turbulence models that are present in OpenFOAM are tested, for the simulation of the backward facing step problem, and coupled with the two equation thermal turbulence model derived in Chapter 4. Two different code couplings are realized: for forced convection case a one way coupling is obtained while for the mixed convection case a two way coupling is built as dynamical turbulence field are transferred from OpenFOAM to FEMuS and temperature field is transferred in the opposite direction for buoyancy force calculation.

6.1 Simulations with FEMuS code

6.1.1 Mathematical model

The liquid sodium turbulent flow over a backward facing step is simulated using a RANS set of equations. Assumptions of incompressible flow and Oberbeck-Boussinesq approximation for buoyant flows are considered. The system of equations consists then of the following

$$\nabla \cdot \mathbf{u} = 0\,, \tag{6.1}$$

$$\begin{aligned}\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\frac{1}{\rho}\nabla P + \\ + \nabla \cdot \left[(\nu+\nu_t)\left(\nabla\mathbf{u} + \nabla\mathbf{u}^T\right)\right] - \mathbf{g}\beta(T - T_{in})\,,\end{aligned} \tag{6.2}$$

$$\frac{\partial T}{\partial t} + \mathbf{u}\cdot\nabla T = \nabla \cdot \left[(\alpha + \alpha_t)\nabla T\right]\,. \tag{6.3}$$

Eddy kinematic viscosity ν_t and eddy thermal diffusivity α_t are modeled with the fields obtained from the solution of the four logarithmic parameter turbulence model

$$\begin{aligned}\frac{\partial \mathrm{K}}{\partial t} + \mathbf{u}\cdot\nabla\mathrm{K} = \nabla\cdot\left[\nu_{eff}^{\mathrm{K}}\nabla\mathrm{K}\right] + \nu_{eff}^{\mathrm{K}}\nabla\mathrm{K}\cdot\nabla\mathrm{K} + \\ + \frac{P_k}{e^{\mathrm{K}}} + c_b\frac{P_b}{e^{\mathrm{K}}} - C_\mu\, e^{\Omega}\,,\end{aligned} \tag{6.4}$$

$$\begin{aligned}\frac{\partial \Omega}{\partial t} + \mathbf{u}\cdot\nabla\Omega = \nabla\cdot\left[\nu_{eff}^{\Omega}\nabla\Omega\right] + 2\nu_{eff}^{\Omega}\nabla\mathrm{K}\cdot\nabla\Omega + \\ + \nu_{eff}^{\Omega}\nabla\Omega\cdot\nabla\Omega + \frac{c_{\varepsilon 1}-1}{e^{\mathrm{K}}}P_k + \\ + \frac{c_b - 1}{e^{\mathrm{K}}}P_b - C_\mu\left(c_{\varepsilon 2}f_{exp} - 1\right)e^{\Omega}\end{aligned} \tag{6.5}$$

$$\frac{\partial \mathrm{K}_\theta}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \mathrm{K}_\theta = \boldsymbol{\nabla} \cdot \left[\alpha_{eff}^{\mathrm{K}_\theta} \boldsymbol{\nabla} \mathrm{K}_\theta \right] + \alpha_{eff}^{\mathrm{K}_\theta} \boldsymbol{\nabla} \mathrm{K}_\theta \cdot \boldsymbol{\nabla} \mathrm{K}_\theta + \\ + \frac{P_\theta}{e^{\mathrm{K}_\theta}} - C_\mu \, e^{\Omega_\theta} \,, \tag{6.6}$$

$$\frac{\partial \Omega_\theta}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \Omega_\theta = \boldsymbol{\nabla} \cdot \left[\alpha_{eff}^{\Omega_\theta} \boldsymbol{\nabla} \Omega_\theta \right] + 2 \alpha_{eff}^{\Omega_\theta} \boldsymbol{\nabla} \mathrm{K}_\theta \cdot \boldsymbol{\nabla} \Omega_\theta + \\ + \alpha_{eff}^{\Omega_\theta} \boldsymbol{\nabla} \Omega_\theta \cdot \boldsymbol{\nabla} \Omega_\theta + \frac{c_{p1} - 1}{e^{\mathrm{K}_\theta}} P_\theta + \frac{c_{p2}}{e^{\mathrm{K}}} P_k + \\ - \left(c_{d1} - 1 \right) C_\mu e^{\Omega_\theta} - c_{d2} C_\mu e^{\Omega} \,, \tag{6.7}$$

where ν_{eff}^{K}, ν_{eff}^{Ω}, $\alpha_{eff}^{\mathrm{K}_\theta}$ and $\alpha_{eff}^{\Omega_\theta}$ are effective viscosity and thermal diffusivity for the turbulence variables transport equations, namely

$$\nu_{eff}^{i} = \nu + \frac{\nu_t}{\sigma_i} \,, \quad \alpha_{eff}^{j} = \alpha + \frac{\alpha_t}{\sigma_j} \,. \tag{6.8}$$

For the mixed convection case an additional source term P_b is considered [52]

$$P_b = -\beta \mathbf{g} \cdot \overline{\mathbf{u}' T'} = \alpha_t \beta \mathbf{g} \cdot \boldsymbol{\nabla} T \,. \tag{6.9}$$

Coefficient c_b is set equal to 1.44. As can be seen from (6.9), the P_b can assume both positive and negative values, depending on relative orientation between gravity vector and mean temperature gradient. Turbulence is then enhanced in the regions where mean temperature gradient is aligned with gravity vector.

Model for ν_t and α_t

Turbulent viscosity is modeled using the two time scale model derived in Chapter 4. For eddy thermal diffusivity three different models of α_t are considered. Kays turbulence model [93] is a simple algebraic model where α_t is calculated through a modeled turbulent Prandtl number Pr_t

$$Pr_t = Pr_{t,k} = 0.85 + \frac{0.7}{Pr \, \nu_t} \,. \tag{6.10}$$

The pros of this model are that α_t is not calculated with a constant and uniform Pr_t value, Pr_t is a function of local quantities, i.e. modeled turbulent viscosity, and it does not require the solution of transport equations for additional turbulence variables that are representative of thermal turbulence. Obviously, as cons, there is the fact that α_t is still modeled using only variables representative of dynamical turbulence. The multiple time scale formulation of $\alpha_t = C_\theta k \tau_{l\theta}$, presented in Chapter 4, is here recalled

$$\tau_{l\theta} = \left(f_{1\theta} B_{1\theta} + f_{2\theta} B_{2\theta} \right) \,, \tag{6.11}$$

L_{in}/h	L_h/h	L_a/h	E_r	W/h	Re_h	Method	Ri
2	20	0	1.5	0	4805	RANS	0 - 0.338

Table 6.2: Non-dimensional parameters for the classification of the studied cases.

4PMK	4PMC	4PCK	4PCC	2PK
- - - - -	———	- - - - -	———	———

Table 6.3: Simulated cases and relative line formatting for all the plots reported in the present work.

where

$$f_{1\theta} = \left(1 - e^{\sqrt{Pr}R_d/19}\right)\left(1 - e^{R_d/14}\right) \,, \tag{6.12}$$

$$B_{1\theta} = \frac{\tau_u}{Pr_{t,\infty}} \,, \tag{6.13}$$

$$f_{2\theta}B_{2\theta} = \tau_u \left(f_{2a\theta}\frac{2R}{C_\gamma + R} + f_{2b\theta}\sqrt{\frac{2R}{Pr}}\frac{1.3}{\sqrt{Pr}R_t^{3/4}} \right) \,. \tag{6.14}$$

Term $B_{1\theta}$ is used to model α_t in the region far from the wall, where it is assumed that turbulent transport is governed by velocity fluctuations and not by temperature ones. With this hypothesis a turbulent Prandtl number can be used and, in particular, both values $Pr_t = 1.33$ and $Pr_t = Pr_{t,k}$ are tested. By using Kays modeled turbulent Prandtl number it is expected to gain more accuracy with respect to the case of a constant and uniform value.

6.1.2 Results

In the present work we report the results obtained for the simulation of a vertical backward facing step case similar to those studied in [109, 82, 107]. The geometrical parameters of the simulated domain are reported in Table 6.2, in accordance with the classification proposed in the Table 6.1.

The system of equations (6.1-6.7) is discretized in FEMuS using quadratic Taylor-Hood finite elements and Stream-wise Upwind Petrov Galerkin stabilization [115]. Time derivatives are discretized using backward Euler method. The algebraic system of equations is solved using the Generalized Minimal Residual method (GMRES) with Incomplete LU factorization (ILU) preconditioner [81]. The basic mesh consists of 31600 cells with 120801 nodes. A general mesh refinement is performed near wall boundaries to obtain a non dimensional wall normal

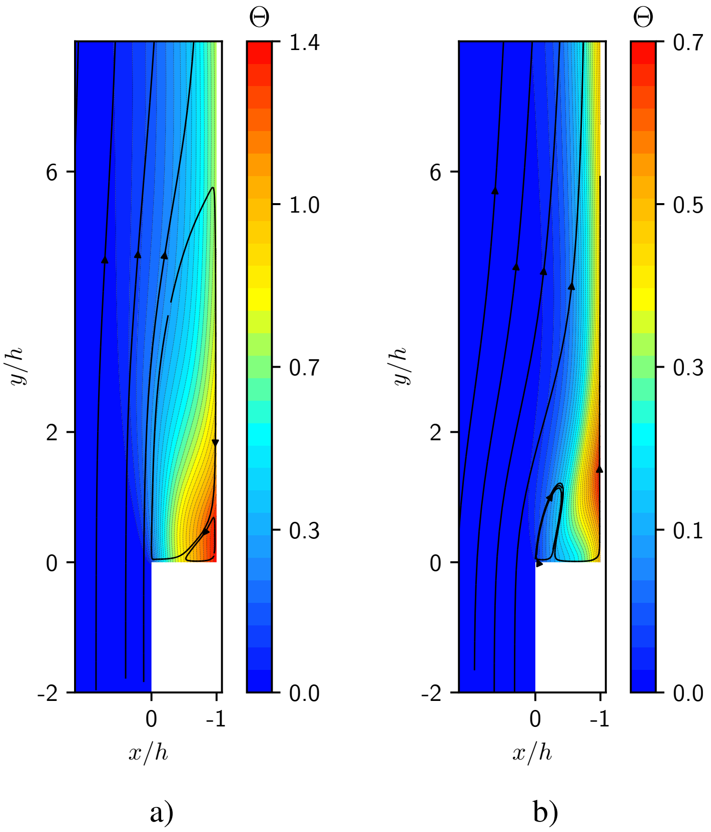


Figure 6.2: Velocity streamlines on a close-up view of the simulated domain, for forced a) and mixed convection b) cases. On the background non-dimensional mean temperature increment Θ fields are reported.

distance $y^+ = \delta u_\tau / \nu$ smaller than one on the first mesh point near wall boundaries, where δ is the wall distance, u_τ the friction velocity and ν the fluid kinematic viscosity.

On the inlet section of the domain we impose a velocity field and the turbulence variables obtained from the simulation of a fully developed turbulent channel flow having a friction Reynolds number $Re_\tau \simeq 300$. The obtained mean velocity leads to a Reynolds number $Re_h = 4805$. For temperature a uniform value equal to 150°C is set. The same temperature is used as reference value for the evaluation of the liquid sodium physical properties used in the system (4.1 - 4.165) through the correlations provided in [116]. We obtain a molecular Prandtl number equal to $Pr = 0.0088$.

On wall boundaries we impose no slip boundary condition for the velocity field, adiabatic boundary condition for the temperature field with the exception of the wall behind the step where a uniform heat flux $\dot{q}$ is imposed. For the dynamical turbulence variables we use boundary conditions in accordance with their near wall behavior. Both MX and CHF boundary conditions are tested on heated wall, for thermal turbulence variables, in order to see their influence on heat transfer, while on remaining walls only CHF boundary condition is used.

On the outlet section an outflow boundary condition is imposed on the velocity field, while for all the other variables we set a zero gradient. We study the cases of pure forced convection, i.e. $Ri = 0$ by setting $\beta = 0$, and the mixed convection case for $Ri = 0.338$ in order to compare the results with the ones obtained in [82, 107].

In Table 6.3 we report all the simulated cases and relative line formatting that is used for all the plots reported in the present work. The cases are labeled in the form *model - bound. cond. - bulk term*, where model can be 4P or 2P ($\mathrm{K} - \Omega - \mathrm{K}_\theta - \Omega_\theta$ or $\mathrm{K} - \Omega$), bound cond can be either M or C (MX or CHF boundary conditions) and bulk term is K or C (KAYS model or constant Pr_t value). For cases 4P M K and 4P C K the bulk correction term used in α_t modeling is calculated using KAYS correlation.

Qualitative comparison between forced and mixed convection cases

Velocity streamlines on a close-up view of the computational domain, are reported in Figure 6.2 a) and b) for $Ri = 0$ and $Ri = 0.338$, respectively. For the forced convection case two main vortices arise behind the step: a bigger one, rotating in clockwise direction, and a smaller one, near the corner between the step wall and the heated one, rotating in the opposite direction. In the mixed convection case, for this value of the Richardson number, the buoyancy force enhances the fluid motion in the region close to the heated wall. The clockwise rotating vortex is greatly reduced in size and it is completely detached from the heated wall. Buoyancy forces also enhance heat transfer, as can be seen from the non dimensional mean temperature increment $\Theta = (T - T_{in})/\Delta T$ field shown in Figure 6.2. In the mixed convection case Θ maximum value is reduced almost by a factor of two in comparison with $Ri = 0$ case.

Dynamical fields

In the present Section obtained results for dynamical fields are compared with relative DNS data. In particular skin friction values along the heated wall and plots of non dimensional velocity and turbulent kinetic energy on several channel cross sections are reported.

Skin friction For forced and mixed convection cases the skin friction $c_f = 2\tau_w/\rho U_b^2$ profile along the heated wall is shown in Figure 6.3 a) and b) and compared with DNS values [107]. As can be seen from the results with $Ri = 0$,

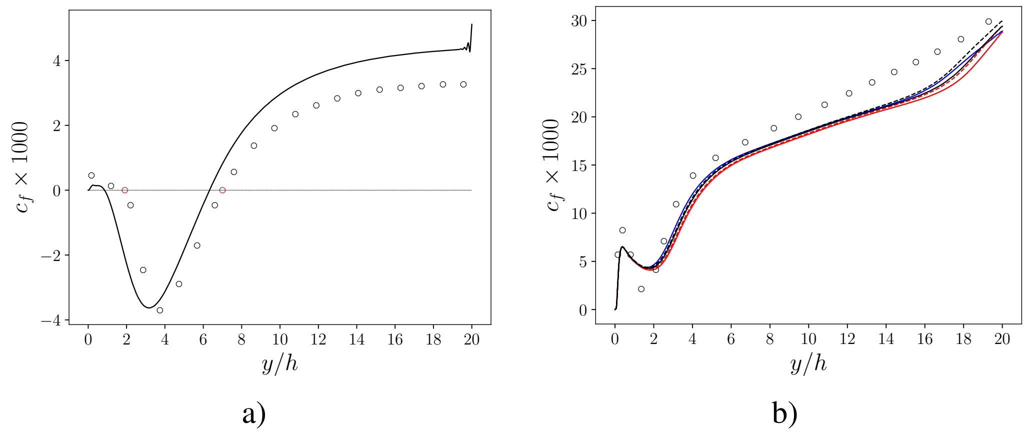


Figure 6.3: Skin friction coefficient c_f for forced a) and mixed convection b) cases along the heated wall. The results are compared with DNS data obtained from [107]. Line formatting in accordance to Tab. 6.3.

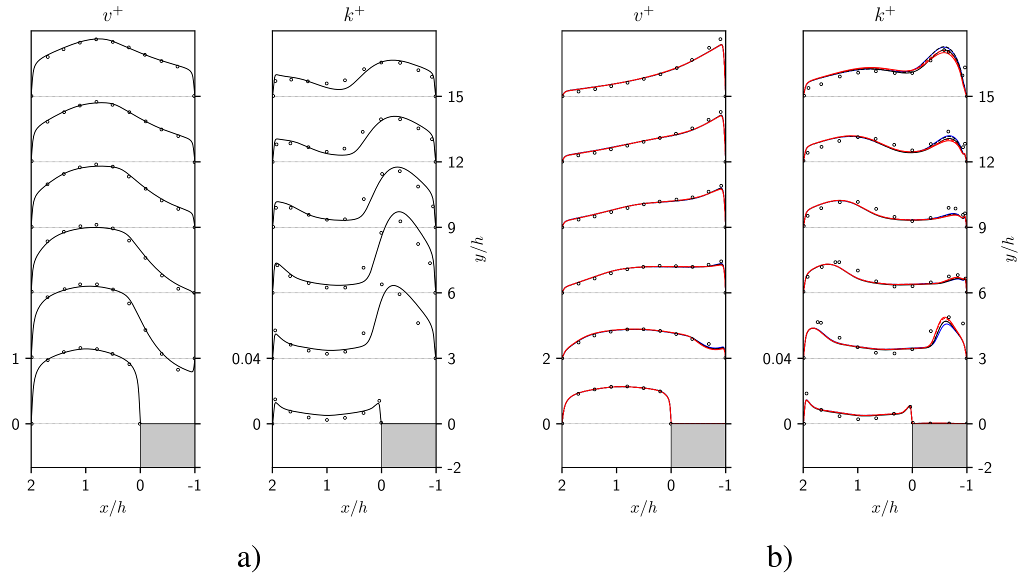


Figure 6.4: Plots of non dimensional velocity component v^+ and turbulent kinetic energy k^+ for forced a) and mixed convection b) cases along channel cross sections taken on several positions $\tilde{y}$. The results are compared with DNS data obtained from [107]. Line formatting in accordance to Tab. 6.3.

the c_f profile is subjected to a double change of sign, denoting the presence of two reattachment points. Their positions, expressed in terms of non-dimensional stream-wise coordinate $\tilde{y} = y/h$, are at $\tilde{y}_1 \simeq 0.85$ and $\tilde{y}_2 \simeq 6.32$. The DNS data give these points at $\tilde{y}_{1,\text{DNS}} \simeq 1.91$ and $\tilde{y}_{2,\text{DNS}} \simeq 7.01$.

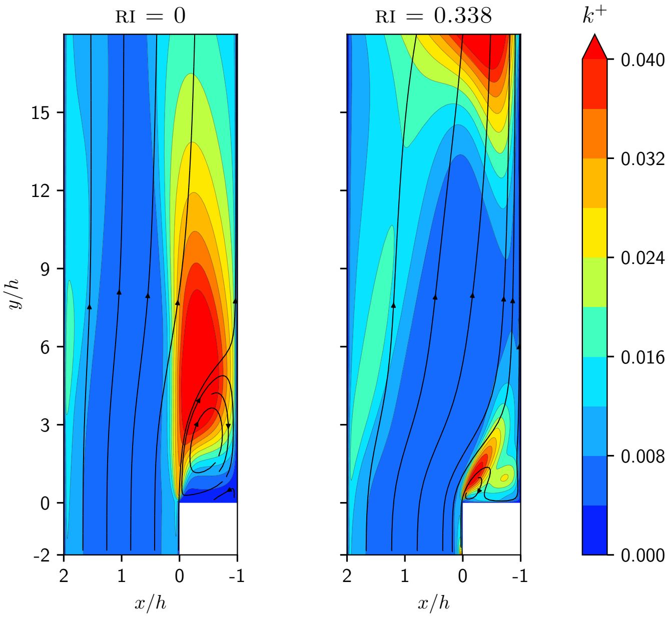


Figure 6.5: Non-dimensional turbulent kinetic energy field k^+, for the forced and mixed convection cases, together with flow streamlines.

As well known the diffusion model underestimates the sizes of the two vortices, and consequently the reattachment lengths, in comparison to DNS results, where reattachment points are shifted further downstream along the heated wall. For the mixed convection case, as reported in Figure 6.3 b), the c_f values are always positive along the heated wall: for this case no reattachment point is found on the heated wall since the clockwise rotating vortex is completely detached from the heated wall. Reference results show a c_f linear behavior for the range of stream-wise coordinates $\tilde{y} \in [6, 20]$.

Results obtained from the solution of RANS system of equations are slightly smaller than DNS ones, for all the tested models of α_t and boundary conditions imposed on the $\mathrm{K}_\theta - \Omega_\theta$ system of equations. These results are characterized by a linear behavior with two different slopes, for $\tilde{y} \in [6, 15]$ and $\tilde{y} \in [16, 20]$. When MX b.c. are used then higher c_f values are obtained. The same behavior is observed when $Pr_{t,\mathrm{KAYS}}$ is used as bulk correction term of α_t instead of the standard constant one. For 2P K simulated case c_f values are higher than those of the four parameter turbulence model for $\tilde{y} \in [2, 7]$ while for $\tilde{y} \in [16, 20]$ c_f shows a smaller linear increase than the one of all the 4P cases. Results obtained in [76] are closer to DNS ones, with a unique c_f linear slope increase along the heated wall. A similar behavior, with double linear slopes, is obtained in [84], although for a different case, namely $Re = 10000$, $Ri = 0.2$ and $E_r = 2$.

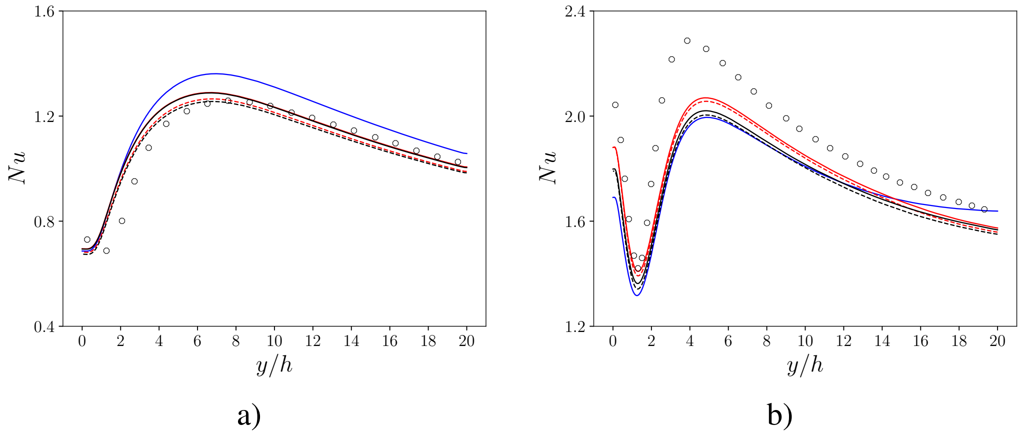


Figure 6.6: Nusselt number values for forced a) and mixed convection b) cases along the heated wall. The results are compared with DNS data [107]. Line formatting in accordance to Tab. 6.3.

Mean velocity and turbulent kinetic energy A more detailed comparison of the results with reference DNS data is given in Figure 6.4 where non-dimensional profiles of velocity and turbulent kinetic energy, taken on channel cross section planes, are reported for several stream-wise coordinate y/h values, for forced and mixed convection cases, respectively. For case $Ri = 0$, profiles of $v^+ = v/U_b$ and $k^+ = k/U_b^2$ are reported, in Figure 6.4, only for one of the simulated cases, because velocity field and dynamical turbulence quantities do not depend on the temperature field.

For $Ri = 0$ the presence of a re-circulation area can be seen from the plot taken at $\tilde{y} = 3$, where negative values of v^+ are present close to the heated wall. After the re-circulation region the flow evolves towards a classical turbulent channel flow. For the mixed convection case, a jet flow develops, in the near heated wall region, due to the presence of buoyancy forces leading to increasing velocity peak values along stream-wise direction. The comparison of velocity values with reference results shows an overall good agreement. The main discrepancies regard wall shear stress predictions and v^+ peak values in the jet flow region, for $Ri = 0.338$. In a comparison with reference results, on each channel cross section considered in Figure 6.4, the position of v^+ peak value is well predicted, but the value itself is slightly smaller than the DNS ones.

The buoyancy forces on fluid turbulence are important as one can be seen in Figure 6.5, where non-dimensional turbulent kinetic energy k^+ fields are reported for both $Ri = 0$ and $Ri = 0.338$. In the forced convection case the highest values of k^+ are found in the re-circulation area, in particular where the clockwise rotating vortex develops. Further downstream, for $\tilde{y} > 6$, k^+ values decrease along stream-wise direction. In the mixed convection case a peak of k^+ field is

			4P M K	4P M C	4P C K	4P C C	2P K	DNS
Forced conv.	Θ_{max}	val	1.484	1.442	1.467	1.439	1.457	1.45
		pos	0.26	0.24	0.28	0.22	0.22	1.01
	Θ_{min}	val	0.796	0.776	0.790	0.775	0.734	0.79
		pos	6.82	6.76	6.82	6.76	6.98	7.32
Mixed conv.	Θ_{max}	val	0.745	0.734	0.718	0.71	0.759	0.70
		pos	1.3	1.28	1.32	1.32	1.24	1.31
	Θ_{min}	val	0.499	0.495	0.486	0.483	0.501	0.44
		pos	4.9	4.86	4.88	4.84	4.94	3.93

Table 6.4: Non-dimensional temperature difference along heated wall for $Ri = 0$ and $Ri = 0.338$ cases. Maximum and minimum values, together with relative position, are compared with DNS data [107].

still found in the re-circulation area. The extent of the high k^+ region is sensibly reduced and its location is shifted closer to the step wall since the re-circulation area is reduced by the action of the buoyancy force. This can be seen from the fluid streamlines, in Figure 6.5, where their bending towards the heated wall is more significant for the $Ri = 0.338$ case. After the re-circulation area we have low k^+ values since the cold fluid is coming from the inlet section. Buoyancy force accelerates the fluid towards heated wall, generating higher wall shear stresses along stream-wise direction, as described from c_f profile. This enhances turbulent kinetic energy production as can be seen from increasing values of k^+ for $\tilde{y} > 9$. In the forced convection case the values of k^+ are slightly larger than the DNS ones on channel cross sections at $\tilde{y} = 3$ and $\tilde{y} = 6$ in the interval $\tilde{x} \in [-1, 0]$. The four parameter model predicts higher momentum exchange and, consequently, re-circulation lengths are smaller than reference ones. The agreement between the results and the DNS reference data improves as the distance from the re-circulation area increases. A good agreement with DNS results is also obtained for the mixed convection case. We remark that, in the mixed case, k^+ results show qualitative deviations at large y near the heated wall where DNS data are characterized by a local peak before going to zero at the wall.

Thermal fields

In this Section obtained results for thermal fields and heat transfer are compared with relative DNS data. As will be shown in the following, DNS results simulate the case of free temperature fluctuations along the heated wall. Both cases of free and zero temperature fluctuations along the heated wall have been considered, in the present study, to see how the four parameter model can reproduce DNS re-

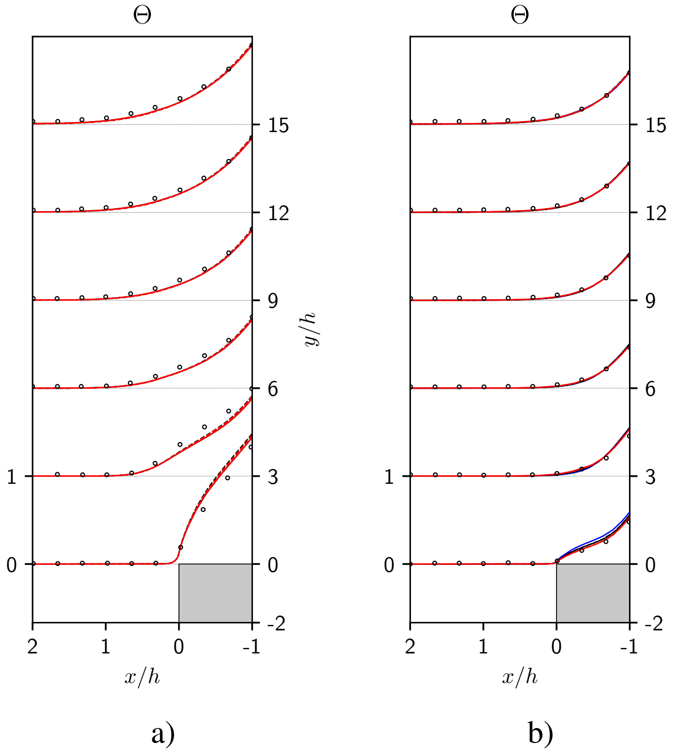


Figure 6.7: Plots of non-dimensional mean temperature increment Θ for forced a) and mixed convection b) cases along channel cross sections taken on several positions ($\tilde{y}$). The results are compared with DNS data [107]. Line formatting in accordance to Tab. 6.3.

sults but also to investigate the influence of different thermal turbulence boundary conditions on heat transfer and turbulent flow.

Nusselt number Nusselt number profiles along the heated wall, for forced and mixed convection cases, for different α_t models and boundary conditions, are shown in Figure 6.6 a) and b), respectively. The Nusselt number is computed as $Nu = \dot{q}h/(T - T_{in})\lambda$, where λ is the liquid sodium thermal conductivity calculated for $T = 150$°C. As one can see, for $Ri = 0$, the Nusselt number values are slightly greater than reference ones in the re-circulation region. For stream-wise positions $\tilde{y} > 9$ the values obtained with the four parameter turbulence model are closer to reference ones. In particular we observe that MX and CHF boundary conditions lead to the same results when a constant is used to model the bulk term in α_t. However when KAYS Pr_t is used as bulk term for α_t, then some differences are observed for different MX and CHF boundary conditions. For the 2PK simulated case, Nu values are slightly overestimated along all the stream-wise positions $\tilde{y}$.

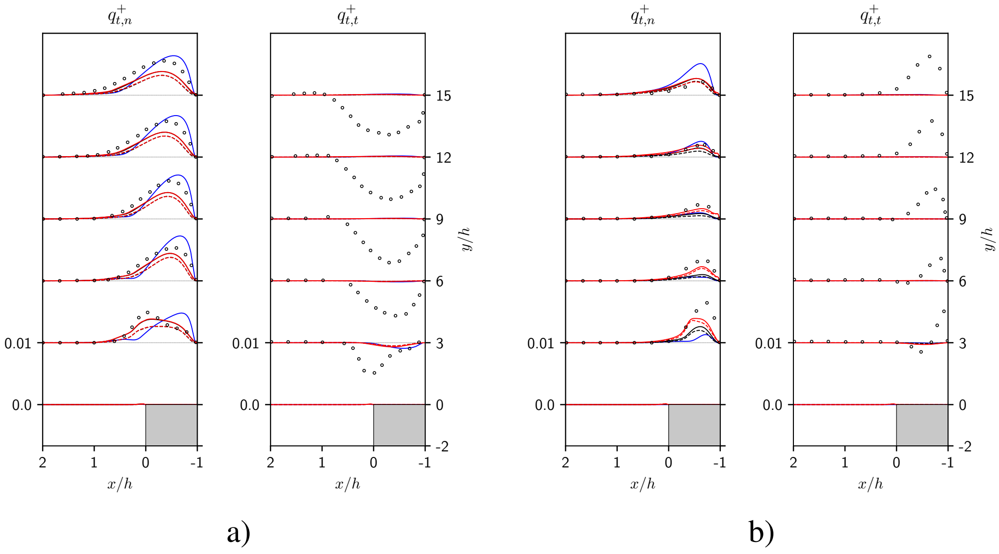


Figure 6.8: Non-dimensional components of turbulent heat flux for $Ri = 0$ a) and $Ri = 0.338$ b). In particular wall normal component, on the left, and stream-wise component, on the right. Reference results (dots) from [107]. Line formatting in accordance to Tab. 6.3.

For the mixed convection case a quite opposite behavior is observed for all models of α_t. In fact the Nu values are slightly smaller than the reference ones. The general behavior of Nu profile is well captured for all the simulated cases with four parameter turbulence model. The results obtained with the KAYS model show a different behavior that can be well defined by dividing the heated wall on three different intervals. In the first, for $\tilde{y} \in [0, 8]$, the Nu values are smaller than the ones obtained with four parameter turbulence model. In the second interval, $\tilde{y} \in [8, 15]$, the results are crossing the four parameters ones and in the third one, for $\tilde{y} > 15$, the Nu values are higher than those of 4P cases. It seems that, on a domain with a longer heated wall, KAYS results would be even higher than the DNS ones for $\tilde{y} > 20$. An over prediction of turbulent heat flux, in the jet flow region, is therefore obtained with the KAYS model, as one can see from the values of turbulent heat flux and eddy thermal diffusivity. In the first interval $\tilde{y} \in [0, 8]$ all the models are underestimating the Nusselt number values probably due to an underestimation of convective heat transfer. Boundary condition CHF leads to a slight increase of the turbulent heat transfer since Nu values are higher than the ones obtained with MX boundary conditions.

A better insight for the cases $Ri = 0$ and $Ri = 0.338$ is reported in Table 6.4 where maximum and minimum Θ values are shown together with the corresponding DNS values. For the $Ri = 0$ case, positions and values of Θ_{max} and Θ_{min} along the heated wall depend clearly on the flow pattern and on the choice of α_t model

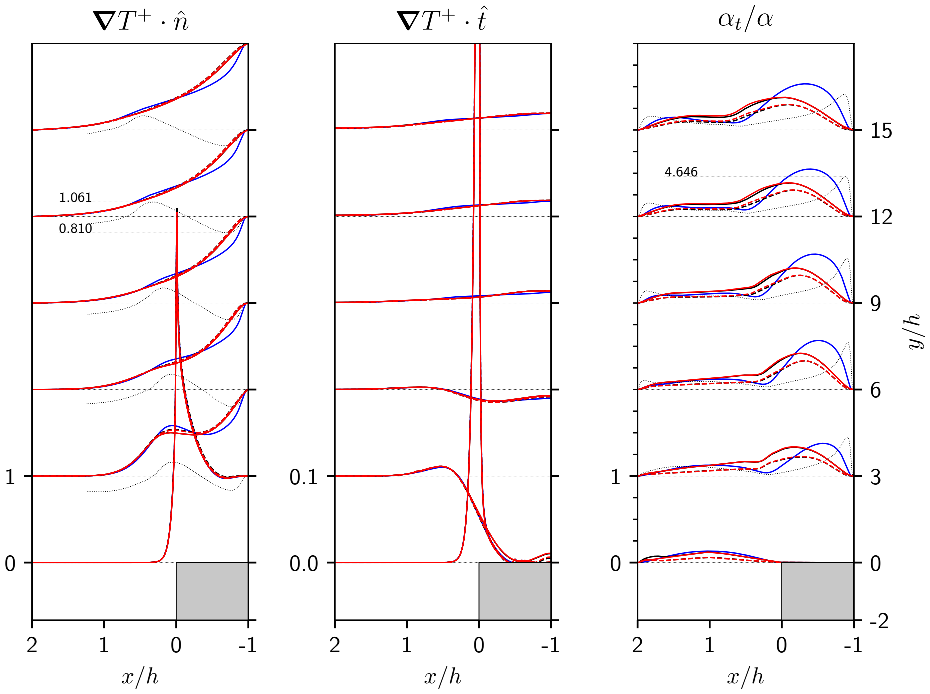


Figure 6.9: Non-dimensional components of mean temperature gradient, along wall normal and stream-wise directions, and non-dimensional eddy thermal diffusivity α_t/α profiles on channel cross sections taken at different stream-wise positions. Dotted lines represent the ratio between KAYS results and MX ones (with constant α_t bulk term). Line formatting in accordance to Tab. 6.3.

and boundary conditions for $\mathrm{K}_\theta - \Omega_\theta$ variables, as well. The Θ maximum values are located in the re-circulation area and are close to the DNS value. However their position is shifted along the upstream direction due to the underestimation of the smaller eddy reattachment point. Minimum values of Θ are obtained after the main vortex reattachment point. In particular we see that the position of Θ_{min} is determined by the choice of α_t bulk term: not only the cases 4PMK and 4PCK but also the cases 4PMC and 4PCC share the same location. The Θ_{min} values depend on the boundary conditions used for the thermal variables $\mathrm{K}_\theta - \Omega_\theta$. Maximum Θ value, obtained with KAYS model, is in good agreement with the reference one, but the minimum one is much smaller, as could be expected by the examination of the Nu profiles of Figure 6.6 a). As already mentioned, for the mixed convection case, both minimum and maximum Θ values are slightly overestimated, with respect to reference values. We see that both Θ maximum/minimum values and their positions depend not only on the turbulence model for α_t but also on the kind of boundary conditions. The best agreement with DNS results is obtained for the

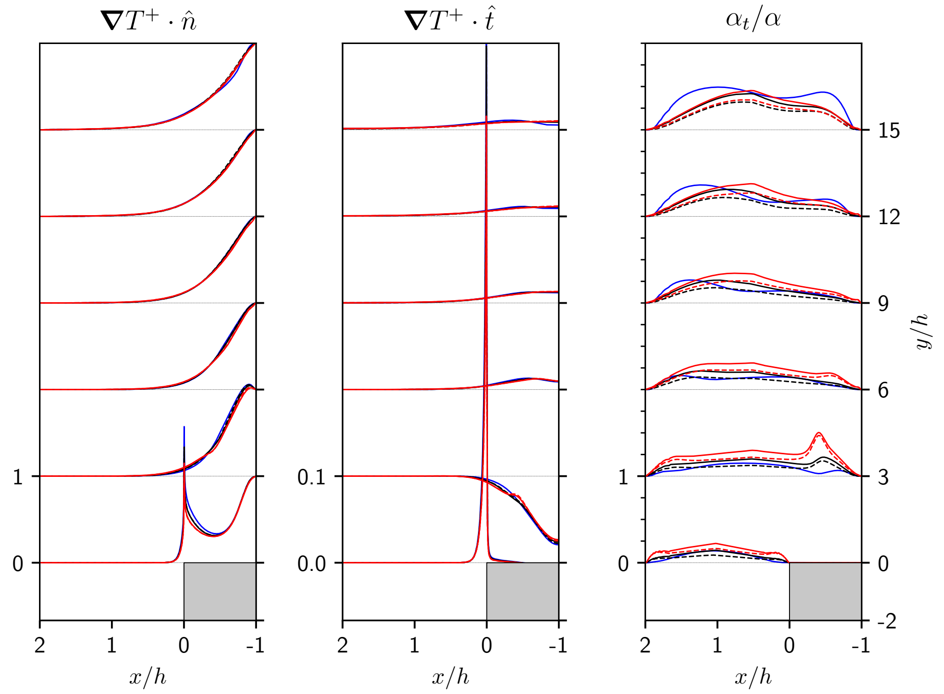


Figure 6.10: Non-dimensional components of mean temperature gradient, along wall normal and stream-wise directions, and non dimensional eddy thermal diffusivity α_t/α profiles on channel cross sections taken at different stream-wise positions. Line formatting in accordance to Tab. 6.3.

case 4PCC, in terms of maximum and minimum Θ values.

Mean temperature A detailed comparison of the results with reference DNS data is given in Figure 6.7 where non-dimensional profiles of temperature change, taken on channel cross section planes, are reported for various values of stream-wise coordinate y/h, for forced and mixed convection cases. An overall good agreement with reference results is obtained for both $Ri = 0$ and $Ri = 0.338$ cases. The major discrepancies with DNS values are found on the plots taken at $\tilde{y} = 0$ and $\tilde{y} = 3$, for the forced convection case, where an over and under prediction of Θ values is respectively obtained. As shown in Figure 6.6, for $\tilde{y} < 0.5$ where the smaller vortex develops, wall temperature values are very close to the maximum one while, for DNS results a slight increase of temperature at the wall is observed. This give an over-prediction of Θ on the plot at $\tilde{y} = 0$. Downstream of reattachment point $\tilde{y}_1$, both DNS and RANS wall temperatures decrease from a maximum value. From Figure 6.6 we see that the slope of the 4P and DNS Nu profiles is quite the same but since $\tilde{y}_1$ is underestimated the wall temperature

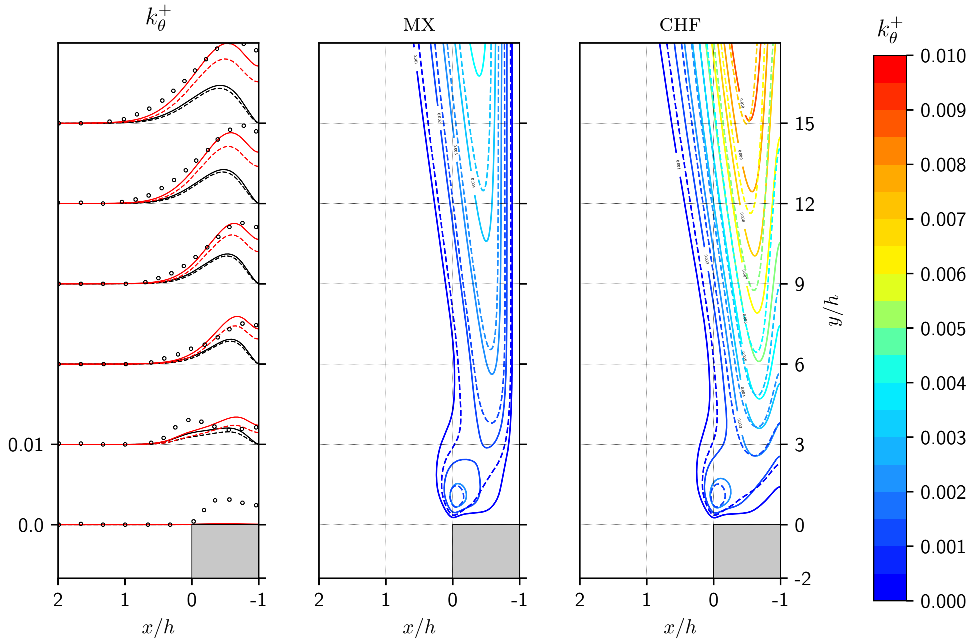


Figure 6.11: Non-dimensional mean squared temperature fluctuations k_θ^+. From right to left: k_θ^+ contours for CHF boundary condition, k_θ^+ contours for MX boundary condition and plots over channel sections (y/h). Reference results (dots) from [107]. Line formatting in accordance to Tab. 6.3.

Θ obtained from RANS simulation is consequently smaller than DNS one. The main difference is observed in the values of Θ along the step wall for the mixed convection case. In particular higher values are obtained in the 2PK simulated case.

Turbulent heat flux and mean temperature gradient In Figure 6.8 the turbulent heat flux components, along wall normal and stream-wise directions, are plotted over different channel cross sections and compared with DNS results for the cases $Ri = 0$ and $Ri = 0.338$ [107]. Modeling turbulent heat flux with the diffusive α_t coefficient and mean temperature gradient ∇T is not able to evaluate the stream-wise component of turbulent heat flux. The reason of this inconsistency to reproduce the stream-wise heat flux component is a model limitation. The mean temperature gradient along the stream-wise direction is small, as it can be seen from Figure 6.9 and 6.10 and DNS results, while turbulent heat flux components have about the same order of magnitude [107]. It is clear that a unique scalar diffusion coefficient is not suitable for predicting both turbulent heat flux components and an anisotropic heat flux modeling should be considered in the future.

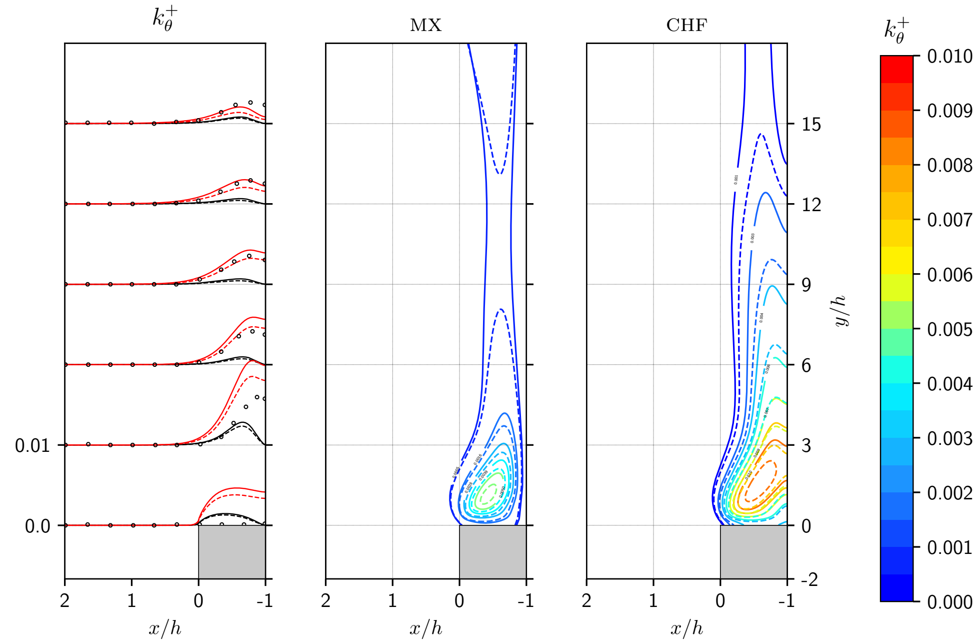


Figure 6.12: Non-dimensional mean squared temperature fluctuations k_θ^+. From right to left k_θ^+ contours for CHF boundary condition, k_θ^+ contours for MX boundary condition and plots over channel sections (y/h). Reference results (dots) from [107]. Line formatting in accordance to Tab. 6.3.

For the forced convection case in Figure 6.9 the values obtained with KAYS model are higher than those obtained with four parameters turbulence model and this is due to higher modeled values of α_t. The non-dimensional eddy thermal diffusivity $\alpha_t^+ = \alpha_t/\alpha$ profiles are reported for all the examined channel cross sections.

Mean temperature gradient component $\nabla T^+ \cdot \hat{n}$ is steeper near heated wall for the case 2P K. However when the ratio between KAYS and MX values of $\nabla T^+ \cdot \hat{n}$ reaches a minimum value close to 0.8 the ratio between KAYS and MX values of α_t^+ has a maximum value close to 5, in the region close to heated wall. The boundary condition choice for $\mathrm{K}_\theta - \Omega_\theta$ has a negligible influence on the turbulent heat flux. Smaller values are obtained when KAYS turbulent Prandtl number is used. With the four parameter turbulence model the estimated position of turbulent heat flux maximum values is in better agreement with DNS values, while KAYS model predicts peak values closer to heated wall. For the mixed convection case a very good agreement is achieved further downstream while some differences between DNS data and the numerical results are observed on the plot taken at $\tilde{y} = 3$.

As already discussed for the Nu profiles along the heated wall, higher t.h.f. values are obtained with KAYS model for $\tilde{y} > 5$, leading to higher Nu values with respect to those obtained with 4P simulations. We observe that the choice of boundary conditions between MX and CHF has a great impact on t.h.f. values and the influence is predominant in the region close to the step, for $\tilde{y} < 9$. The major effect of free temperature fluctuations, on the heated wall, is in the modeled values of eddy thermal diffusivity, as reported in Figure 6.10, where non-dimensional mean temperature gradient components and α_t^+ profiles are reported on several channel cross sections. Obtained results of mean temperature gradient $\boldsymbol{\nabla} T^+$ along wall normal direction are very similar for all simulated cases, while non dimensional eddy thermal diffusivity values are quite different. When using MX boundary conditions, temperature fluctuations specific dissipation, ω_θ, is set equal to $2\alpha/(C_\mu \delta^2)$ (δ is the wall distance) while with CHF boundary conditions a vanishing gradient is imposed along wall normal direction, $\boldsymbol{\nabla}\omega_\theta \cdot \hat{n} = 0$, obtaining much smaller values of ω_θ. For the turbulent kinetic energy specific dissipation ω a MX boundary condition is imposed, i.e. $\omega = 2\nu/(C_\mu \delta^2)$. The influence of the boundary condition used on $\mathrm{K}_\theta - \Omega_\theta$ is seen on the distribution of thermal-to-dynamical time scale ratio $R = \omega/\omega_\theta$ that is used for modeling α_t.

Mean squared temperature fluctuations Non-dimensional mean squared temperature fluctuations $k_\theta^+ = 2e^{\mathrm{K}_\theta}/\Delta T^2$ are examined in Figure 6.11 and Figure 6.12 for forced and mixed convection cases, respectively. As one can see in Figure 6.11, for the case $Ri = 0$, a comparison with DNS values is shown on several channel cross sections along stream-wise position $\tilde{y}$. Reference results clearly show the presence of non-vanishing temperature fluctuations along the heated wall. For MX b.c. temperature fluctuations are underestimated along each channel cross section. By using CHF b.c. we observe a good agreement with DNS results. As can be seen from k_θ^+ contours of Figure 6.11 when the KAYS correlation models the bulk correction term for α_t the temperature fluctuations are smaller than those obtained with a constant bulk correction term for α_t. As shown in Figure 6.11 this behavior is observed with both MX and CHF boundary conditions.

Mean temperature gradient values calculated with the four parameter turbulence model are quite independent from the choice of α_t bulk term and $\mathrm{K}_\theta - \Omega_\theta$ boundary conditions. The turbulent diffusivity α_t appears to be smaller when KAYS Pr_t values are used instead of a constant one. This fact leads to smaller values of the production term P_θ and consequently a reduction of k_θ. The temperature fluctuations presents a peak in the region close to step corner, where recirculation occurs, and growing values along stream-wise direction in the region close to heated wall.

For the mixed convection case we observe that DNS mean squared temperature fluctuations do not vanish near the heated wall and decrease along the stream-wise direction. Results obtained with CHF boundary condition are in good agreement with the reference ones. As already observed for the forced convection case they

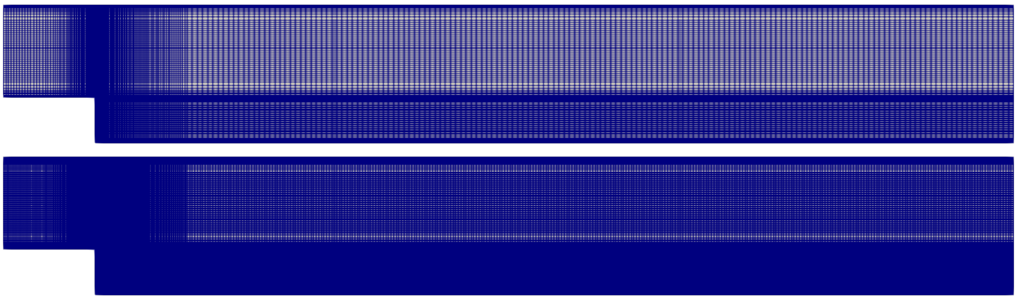


Figure 6.13: Computational grids used for the simulation of backward facing step geometry with OpenFOAM.

depends on α_t bulk term and decrease along stream-wise direction. Also for $Ri = 0.338$ case, the mean temperature gradient values are about the same for all chosen models of α_t, as shown in Figure 6.10, while modeled α_t values are smaller when KAYS turbulent Prandtl number is used instead of a constant one. As for the $Ri = 0$ case, smaller production term P_θ values are obtained and also k_θ values. By examining the k_θ contours, reported in Figure 6.12, we see that different boundary conditions lead to a different behavior of k_θ.

For forced convection case, as shown in Figure 6.11, k_θ profiles are quite similar, for both MX and CHF b.c., with a local peak close to the step corner and increasing values along stream-wise direction. For the mixed convection case we see that k_θ peak value is reached behind the step, within $\tilde{y} \in [0, 3]$ interval, and that for CHF boundary condition the peak position is shifted close to the heated wall. Values obtained with MX boundary conditions decrease along stream-wise direction and further downstream increase when $\tilde{y} > 12$. With CHF boundary condition k_θ just decreases as $\tilde{y}$ increases.

6.2 Simulations with coupled FEMuS & OpenFOAM codes

From the results discussed in the previous Section it has been seen that the under estimation of recirculation region size, for forced convection case, has a sensible impact on turbulent heat transfer occurring behind the step. Moreover it has been seen that the thermal turbulence modeling of the KLW model allows to obtain more accurate results than those obtained with Kays correlation for the eddy thermal diffusivity. A coupling between FEMuS and OpenFOAM codes is then realized, in particular using OpenFOAM extend Version 3.2. In OpenFOAM several dynamical turbulence model are present, in particular the Launder & Sharma low-Re k-ε

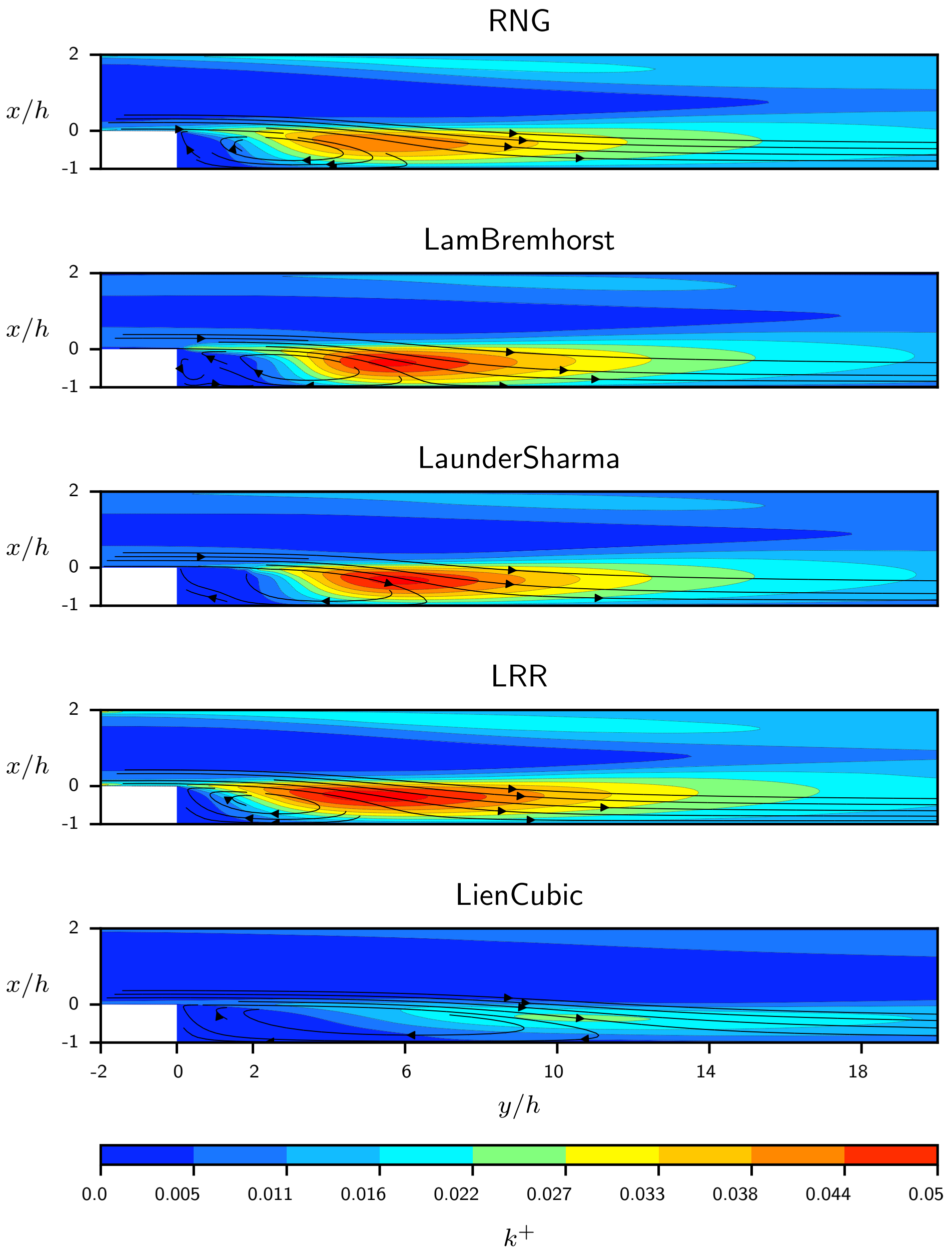


Figure 6.14: Non dimensional turbulent kinetic energy distributions k^+, for forced convection case with flow streamlines. Results obtained with OpenFOAM turbulence models RNG, LB, LS, LRR and LC.

(LS) [114], the Yakhot et al. Renormalization group k-ε (RNG) [117], the Lam & Bremhorst low-Re k-ε (LB) [118], the Lien cubic non-linear low-Re k-ε (LC) [119] and the Reynolds stress model (LRR) by Launder et al. [120]. The models can be

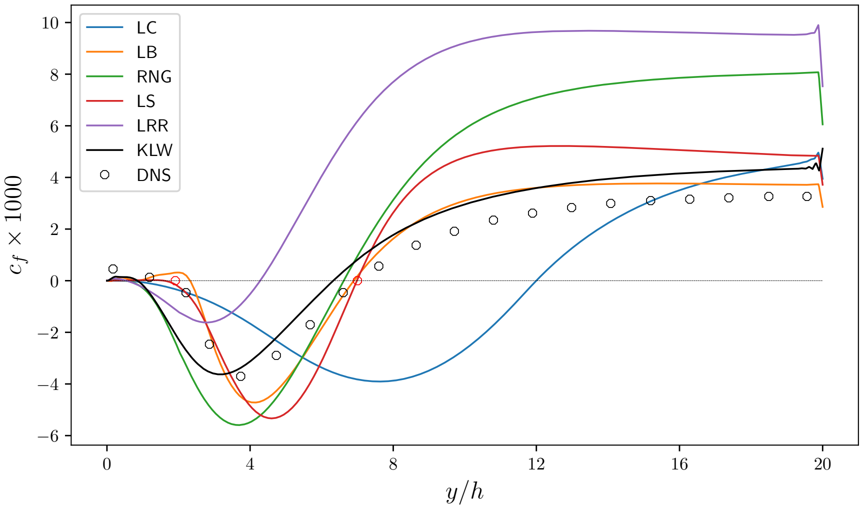


Figure 6.15: Skin friction c_f values, along the heated wall, obtained from simulations with OpenFOAM turbulence models. Results are compared with DNS data [107] and values obtained with KLW model.

tested in simulating the turbulent flow over the vertical backward facing step to see which one allows to improve the flow prediction with respect to the dynamical turbulence model of KLW. The great advantage of using the computational platform is that the dynamical turbulence models can be used without the need of implementing them in FEMuS code since they are already available in OpenFOAM. Results of the turbulent heat transfer over the vertical backward facing step can then be improved by using a dynamical turbulence model from OpenFOAM code and the thermal turbulence modeling of the FEMuS KLW model.

A first comparison of the results obtained with the above mentioned turbulence models is made by simulating the forced convection case that has been analyzed in previous Section. Boundary conditions have been set as follows:

- Inlet: fixed velocity profile and zero gradient for k, ε and Reynolds stresses
- Wall: no slip for velocity field, wall functions for turbulence variables
- Outlet: zero gradient on all solved variables

Two different grids are used, as shown in Fig. 6.13. The coarser one, which consists of approximately 32000 cells, has been used to compare the different turbulence models, while the finer one, consisting of approximately 130000 cells, has been used to couple OpenFOAM with FEMuS. Both grids have a fine discretization in the near wall region, allowing the resolution of viscous layer.

Model	LC	LB	RNG	LS	LRR	KLW	DNS
y_1	0.4	2.32	0.52	1.6	0.55	0.85	1.91
y_2	12	6.9	6.61	6.98	4.28	6.32	7.01

Table 6.5: Comparison of reattachment point locations for smaller vortex (y_1) bigger vortex (y_2) obtained with OpenFOAM turbulence models, KLW model and reference DNS values.

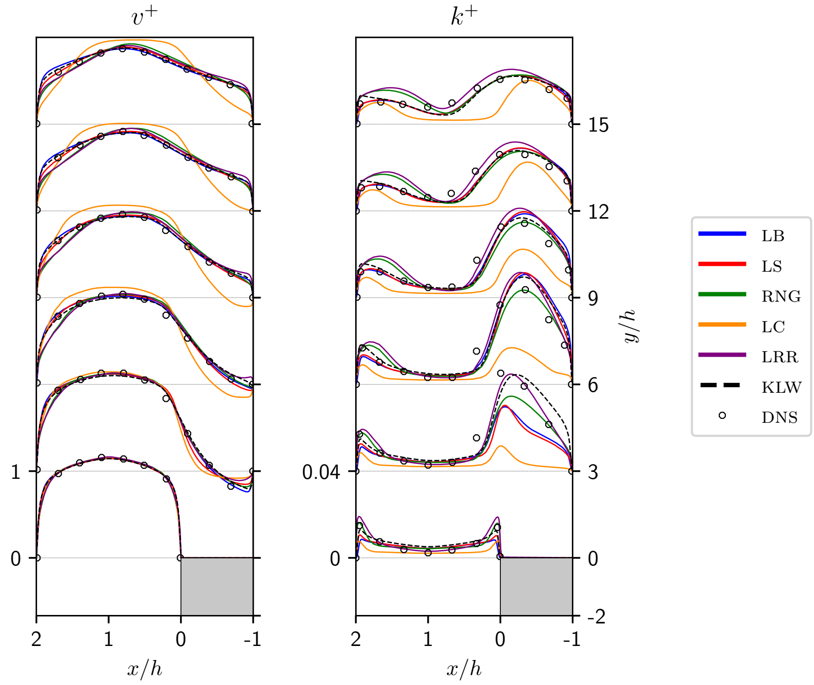


Figure 6.16: Plots of non dimensional velocity component v^+ and turbulent kinetic energy k^+ obtained with OpenFOAM turbulence models along channel cross sections taken on several positions $\tilde{y}$. The results are compared with DNS data obtained from [107] and KLW model.

Preliminary evaluation of OpenFOAM dynamical turbulence models

Non dimensional values of turbulent kinetic energy k^+ obtained with the different models are reported in Fig. 6.14, together with flow streamlines. From this qualitative comparison it can be seen that LB and LS models provide similar results. Reynolds stress models predicts a higher momentum exchange, as can be seen from high values of k^+ obtained in the region behind the step. Recirculation area appears to be much smaller than the one obtained with LB and LS models. This is probably due also to incorrect use of wall function boundary condition for Reynolds stresses as first layer of cells, close to wall boundary, should lay in $y^+ \in [30, 60]$ while for the considered case $y^+ < 1$ [121]. Reynolds stresses, in fact, are not forced to vanish in the near wall region, as they should, and high turbulent kinetic energy values are observed also close to the wall. Smaller values of k^+ are obtained with RNG and LC models, with a particular sensible under-prediction of k^+ in the latter case.

Skin friction values obtained with the above mentioned turbulence models are reported in Fig. 6.15, where they are plotted against non dimensional stream-wise coordinate y/h along the heated wall. The values are compared with reference DNS data and with those obtained with the KLW model. It can be seen that very different values of skin friction coefficient are obtained, with LS and LB model results being closer to DNS reference values. Reattachment point locations are reported in Table 6.5. As it can be seen from Fig. 6.14, LC model predicts a much smaller momentum exchange in the recirculation area that is then much bigger than the reference one, with bigger vortex reattachment point at about $y/h = 12$. Renormalization k-ε turbulence model under predicts the size of the smaller vortex, with $y_1 = 0.52$, and the reattachment point location y_2 is closer to DNS value than that obtained with KLW model. Downstream of y_2 location much bigger values of skin friction coefficients are observed for both RNG and LRR models. Models LS and LB perform better than the previous ones. Both of them predict the presence of a third small vortex, in the corner region, with a reattachment point located at $y_3 = 0.03$ and $y_3 = 0.16$ respectively. The presence of such small vortex is also observed in [110] with a y_3 value equal to 0.06.

As a final comparison between the different models, non dimensional values of velocity and turbulent kinetic energy are plotted on several channel cross sections in Fig. 6.16. The results are compared with KLW values and reference DNS data. Non linear k-ε model LC under-predicts k^+ on the whole domain and the resulting velocity field does not reproduce the DNS results. At stream-wise positions $\tilde{y} = 0$ and $\tilde{y} = 3$ RNG model well approximates DNS values of k^+, while further downstream turbulent kinetic energy is over-predicted, leading to an underestimation of recirculation area. This fact can be seen also from the $\tilde{y} = 6$ plot of non dimensional velocity, near heated wall, where other turbulence models predict a small negative velocity value, differently from LRR model. Velocity profiles obtained with RNG model show a behavior similar to those of LRR in the region close to adiabatic wall although smaller turbulent kinetic energy is predicted. Close to heated wall the model fairly reproduces DNS values of k^+, while an underestimation is obtained moving from k^+ peak position towards channel center line. Turbulence models LS and LB show similar values of k^+. Within the recirculation region they both underestimate k^+, and a better agreement with DNS data is obtained at positions $\tilde{y} = 12$ and 15, close to heated wall. For positions $\tilde{y} > 6$ obtained values of k^+ are similar to those obtained with KLW model, while for non

FORCED		MIXED	
UNCOUPLED	COUPLED	UNCOUPLED	COUPLED
- - - - -	———	- - - - -	———

Table 6.6: Simulated cases and relative line formatting for the results of the coupled FEMuS–OpenFOAM case.

dimensional velocity small differences are observed near the adiabatic wall, with LB values being slightly higher than KLW and LS models.

Turbulence model LB has been chosen to realize the code coupling between OpenFOAM and FEMuS codes for both forced and mixed convection cases. Because of the thermal coupling involved in the mixed convection case, forced and mixed convection simulations require different code coupling strategies. For forced convection case a one way coupling is realized from OpenFOAM to FEMuS, so that a steady OpenFOAM solution can be obtained and then the thermal fields can be solved with FEMuS code. For the mixed convection case a thermal feedback is given to OpenFOAM, through the buoyancy term with temperature field solved with FEMuS code. The coupling procedure is realized by following the steps described in Chapter 2. For OpenFOAM a finer grid than FEMuS one is used, so that a similar number of unknowns is solved by both OpenFOAM and FEMuS codes. FEMuS fields are interpolated on OpenFOAM grid with the P2P2 field interpolator and then piece-wise fields are computed. Numerical fields from OpenFOAM are interpolated on FEMuS mesh using MED conservative piece-wise field projector and then point-wise fields are recovered from the solution of Galerkin projection. For both forced and mixed convection cases, obtained results are compared with those obtained from uncoupled KLW model. Line formatting is described in Table 6.6. The reported results are only for the case of free temperature fluctuations along the heated wall.

Dynamical fields

As already mentioned the simulation of forced convection case is performed as a one way coupling from OpenFOAM to FEMuS. In particular k, ε and ν_t values computed in OpenFOAM are used to model eddy thermal diffusivity α_t and source terms for K_θ–Ω_θ system of equations. For the mixed convection case, at each iteration FEMuS solved mean temperature field is used in OpenFOAM, for the calculation of buoyancy force, and dynamical turbulence fields are sent from OpenFOAM to FEMuS as for the forced convection case.

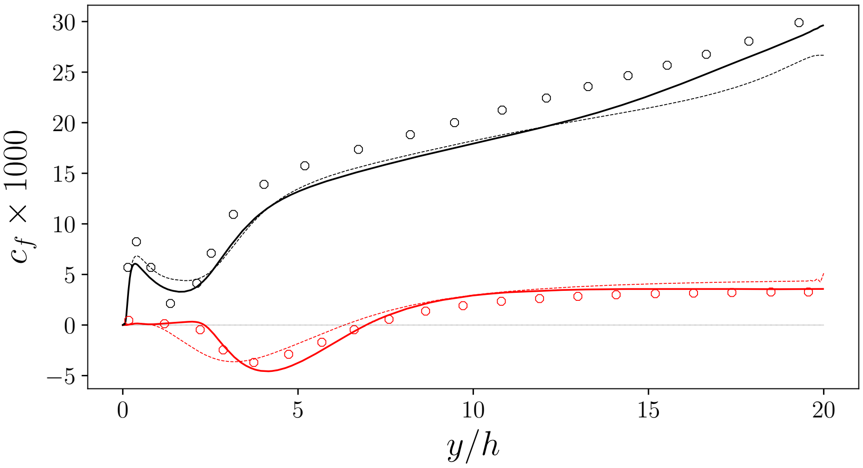


Figure 6.17: Comparison of skin friction values along heated wall obtained from uncoupled FEMuS and coupled OpenFOAM–FEMuS simulations, for both forced and mixed convection cases. The results are compared with DNS data [107]. Line formatting in accordance with Tab. 6.6.

Skin friction Skin friction profiles of both forced and mixed convection cases are shown in Fig. 6.17. As already seen before, with Lam-Bremhorst turbulence model the size of recirculation area is better predicted for the forced convection case, so that an improvement has been obtained from the coupled code solution. The change of turbulence model, for mixed convection case, has not a well defined impact on skin friction values. Close to the step, for $y/h < 2.5$, coupled solution c_f values are smaller than those obtained with KLW model, while higher values are obtained for $y/h > 12.5$.

Non dimensional velocity and turbulent kinetic energy In Fig. 6.18 non dimensional stream-wise velocity component v^+ and turbulent kinetic energy k^+ profiles are shown on several channel cross sections. As it regards the velocity field component, for forced convection case the main difference is observed on plot taken at $y/h = 6$, where for coupled solution results negative v^+ values are observed on a wider section than that obtained for KLW case. For the mixed convection case, from coupled simulation results a small under-prediction of v^+ peak value in the region close to heated wall is observed together with higher values near the adiabatic wall at $x/h = 2$. For forced convection case, the main differences in k^+ values are observed on plots at stream-wise position $y/h = 3$ where Lam-Bremhorst predicted turbulent kinetic energy is sensibly smaller than both KLW and reference DNS values. After the reattachment point, KLW and coupled case solutions do not differ very much. For mixed convection case slightly higher

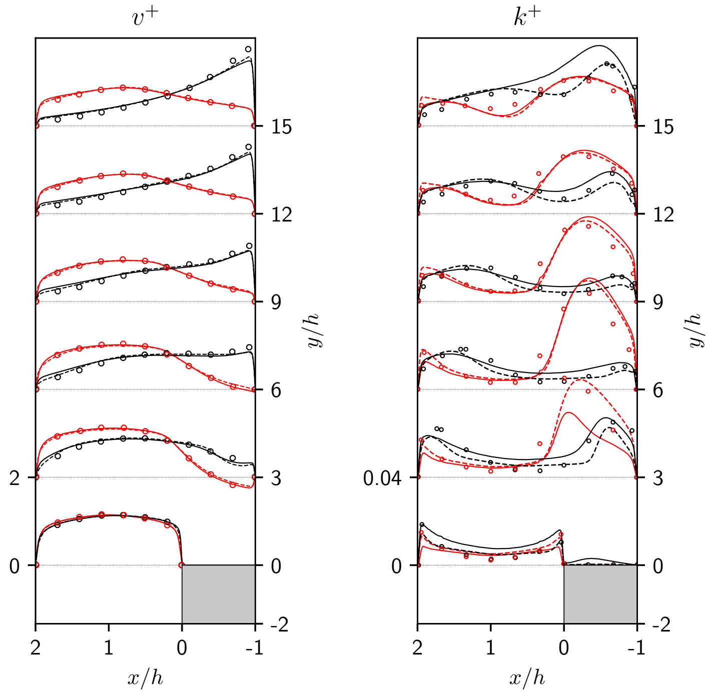


Figure 6.18: Comparison of non dimensional stream-wise velocity component v^+ and turbulent kinetic energy k^+ obtained with OpenFOAM–FEMuS solutions, for both forced and mixed convection cases. Results are compared with uncoupled case and reference values [107]. Line formatting in accordance with Tab. 6.6.

values of k^+ are observed on each channel cross section. Local k^+ peak DNS values are almost well captured, from coupled simulation, at y/h = 3, 6, 9 and 12, but higher momentum exchange is predicted in the central channel region.

Thermal fields

Nusselt number The effect of a better recirculation area prediction, for forced convection case, is noticeable in the profiles of Nusselt number values along the heated wall, as reported in Fig. 6.19. A great improvement in heat transfer prediction has been obtained in the interval $y/h \in [0, 7]$, while further downstream Nusselt number values obtained with KLW and coupled simulation are identical. For mixed convection case the Lam-Bremhorst k-ε turbulence model does not lead to a more accurate Nu prediction, along heated wall, as in the range $y/h \in [0, 1.5]$ Nu values are smaller than those obtained with KLW model, that means a higher discrepancy with DNS values, and on the remaining part of the heated wall they are comparable to uncoupled case results.

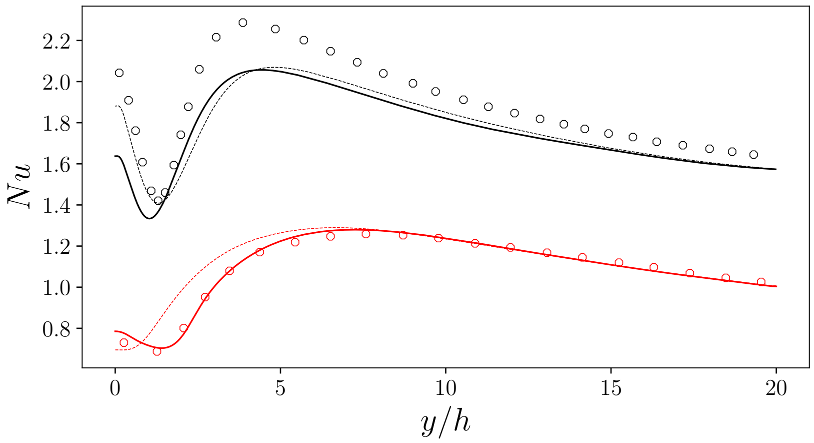


Figure 6.19: Nusselt number values along heated wall obtained with coupled OpenFOAM–FEMuS simulations for both forced and mixed convection cases. Results are compared with uncoupled case and DNS data [107]. Line formatting in accordance with Tab. 6.6.

Turbulent and advective heat fluxes In Fig. 6.20 non dimensional turbulent heat flux component along wall normal direction, $q^+_{t,n}$, and mean advective heat flux components $v^+\theta$ and $u^+\theta$ profiles are shown on several channel cross sections. Stream-wise component values of turbulent heat flux are not shown since also for the coupled case they are not captured. As it regards the values of wall normal component $q^+_{t,n}$, for $Ri = 0$ case, with the new model a better estimation is obtained at stream-wise position $y/h = 3$. Further downstream coupled and uncoupled KLW predicted values of turbulent heat flux are almost equivalent, with an under-prediction with respect to reference values. For mixed convection case $q^+_{t,n}$ values at $y/h = 3$ are smaller than KLW ones and for increasing values of downstream position y/h less sensible differences are observed. With Lam-Bremhorst turbulence model a better prediction of advective heat flux component $v^+\theta$ is obtained at $y/h = 3$, where the effect of flow recirculation can be seen on the change of $v^+\theta$ values. A very good agreement is observed with DNS results, for both $Ri = 0$ and $Ri = 0.338$ cases. The main discrepancy with reference values is observed, for mixed convection case, in the peak value of $v^+\theta$, which is slightly smaller then the reference one. Another improvement obtained from the coupled code solution can be seen in wall normal component values of mean advective heat flux $u^+\theta$, in particular at $y/h = 3$, where KLW model does not predict the change of sign of wall normal velocity component. As for previously examined results, coupled code results are about the same of KLW ones, along the remaining channel cross sections, and for all mixed case results. A general good agreement with DNS results is observed.

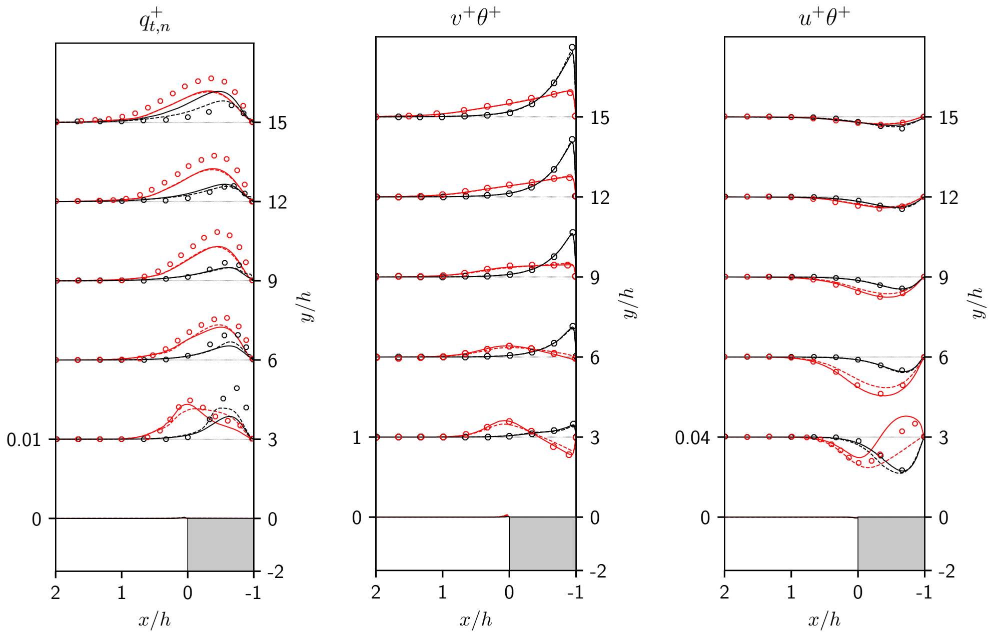


Figure 6.20: Profiles of non dimensional turbulent heat flux component along wall normal direction $q^+_{t,n}$ and of mean advective heat fluxes $v^+\theta$, $u^+\theta$ components on several channel cross sections, obtained with coupled OpenFOAM–FEMuS simulations. Results are for both forced and mixed convection cases, compared with uncoupled case and DNS data [107]. Line formatting in accordance with Tab. 6.6.

Mean temperature and temperature fluctuations For non dimensional temperature values the main differences with uncoupled case, for $Ri = 0$, are observed on profiles taken on channel cross sections at stream-wise locations $y/h = 0$, 3 and 6, as reported in Fig. 6.21. A better agreement with DNS data is here obtained, in particular at $y/h = 0$ and 3, where uncoupled case results respectively overestimated and underestimated reference values as a consequence of incorrect prediction of recirculation area. For mixed convection case sensible differences are not observed, apart from the plot taken at $y/h = 0$, where coupled simulation results predict higher θ values, as expected from smaller Nu values observed in Fig. 6.19. Profiles of non dimensional mean squared temperature fluctuations along several channel cross sections are plotted on the right of Fig. 6.21 and compared with results from uncoupled simulation and DNS values. For mixed convection case, results of coupled simulation predicts higher mean squared temperature fluctuations on the whole domain. Forced convection case results from coupled simulation are smaller than respective results from uncoupled simulation. On channel cross section at $y/h = 3$ the results obtained with coupled OpenFOAM-FEMuS simula-

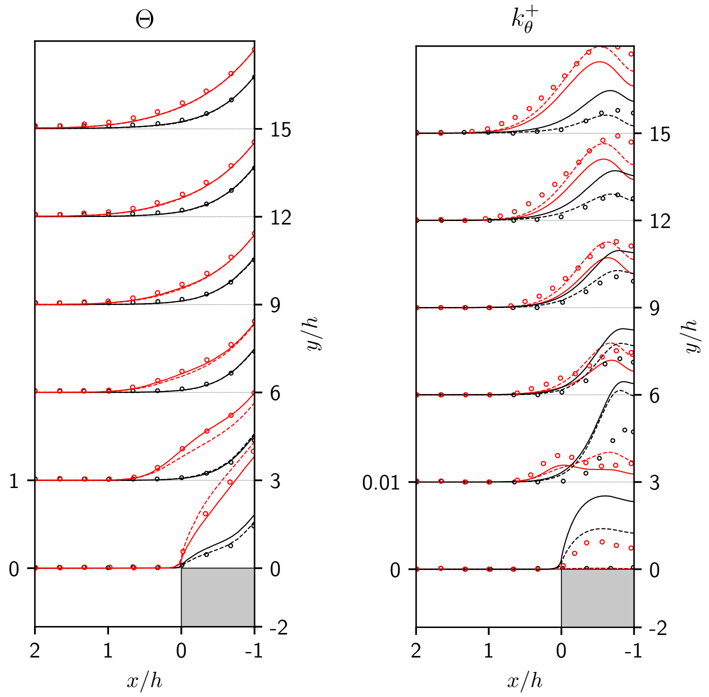


Figure 6.21: Non dimensional temperature and mean squared temperature fluctuations values, on several channel cross sections, obtained with coupled OpenFOAM–FEMuS solutions. Results are for both forced and mixed convection cases, compared with uncoupled case and DNS data [107]. Line formatting in accordance with Tab. 6.6.

tions better predicts the position of maximum k_θ^+ value. In Fig. 6.22 a comparison between k_θ^+, from coupled and uncoupled simulations of forced convection case, is reported together with plots of non dimensional eddy thermal diffusivity on several channel cross sections. The behavior of shown k_θ^+ fields is similar: a local peak of k_θ^+ values is observed close to the step corner, with a position slightly shifted downstream for the case of coupled simulation. Higher values of k_θ^+ then develop starting from the center of the recirculation region and moving downstream in the near heated wall region. Within the recirculation region smaller k_θ^+ values are obtained for the case of coupled simulation, due to smaller values of eddy thermal diffusivity, so of k_θ^+ production term, as can be seen from the plots reported on the right of Fig. 6.22. After the reattachment point the obtained values of α_t^+ are almost the same for coupled and uncoupled simulations. Smaller values of k_θ^+ are then obtained, on coupled simulation results, as a consequence of smaller values encountered in the recirculation region and then advected in the near hot wall region.

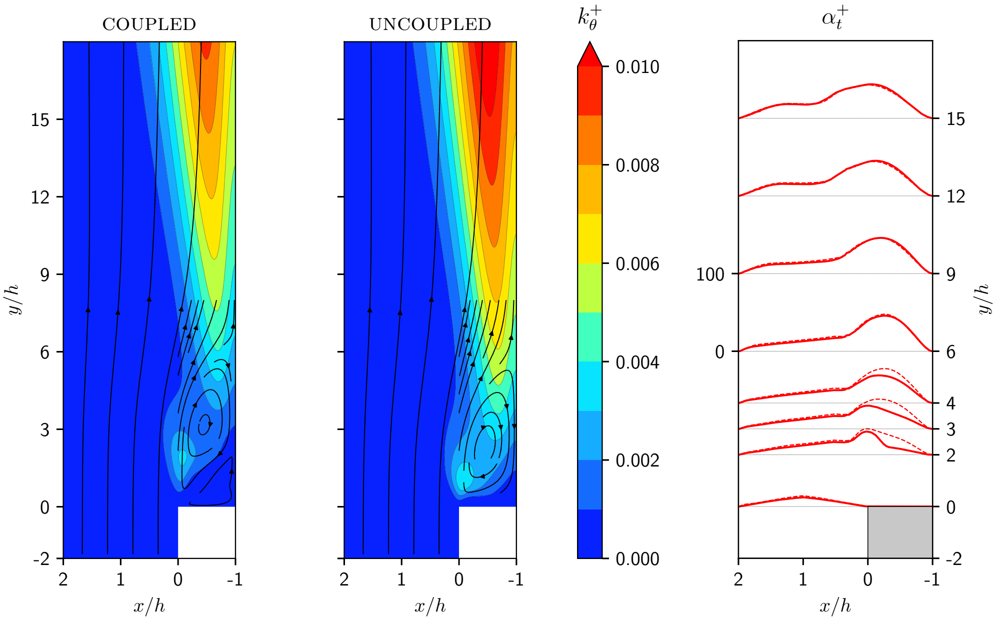


Figure 6.22: Forced convection case: comparison of non dimensional mean squared temperature fluctuations, for CHF boundary condition, between coupled and uncoupled simulations , on the left, and of non dimensional eddy thermal diffusivity, on the right. Line formatting in accordance with Tab. 6.6.

Overall conclusions The simulation of liquid sodium turbulent flow over vertical backward facing step has been performed for both forced and mixed convection cases. This is an interesting case since a very low Prandtl number has been considered ($Pr = 0.0088$) but also a more complex geometry, involving fluid recirculation. Due to the presence of DNS data, an extensive evaluation of the obtained results has been performed by comparing them with reference ones. Simulations have been performed using FEMuS code, with the KLW model, and using a numerical code coupling between FEMuS and OpenFOAM.

From the KLW model performances point of view, for both forced and mixed flow regimes the average temperature and velocity profiles match the DNS values. For the forced convection case, the model underestimates the length of the recirculation region, with an influence on obtained Nusselt number values along the heated wall. Nu values show in fact some main discrepancies with DNS data, in the recirculation region, while downstream of the reattachment point a very good agreement with reference values is obtained. For the mixed regime the Nusselt number and skin friction values along the heated wall are both slightly smaller than the reference ones. As expected, the turbulence model almost captures the values of wall normal turbulent heat flux component while it fails in reproducing the stream-wise component since the used isotropic model, with a unique scalar diffusion coefficient, is not suitable for predicting both turbulent heat flux components. For this very low Pr value, heat transfer is conduction dominated and almost accurate results can be obtained even if stream-wise turbulent heat flux component is not captured. An anisotropic model for turbulent heat transfer, capable of reproducing both wall normal and stream-wise components, should be used in applications where turbulent heat flux has a greater influence on global heat transfer.

From the coupled simulations it has been seen that Lam-Bremhorst k-ε model allows to obtain a better prediction of the flow recirculation occurring behind the step, for the forced convection case. A better agreement with DNS results is then obtained for all observed variables in the recirculation region, while downstream of the reattachment point LB and KLW models results do not show sensible differences. In the presence of buoyancy forces, the change of dynamical turbulence model does not lead to a sensible improvement of obtained results since, for the investigated $Ri = 0.338$ value, the recirculation region is greatly reduced in size. The computational platform has then proven to be a useful tool since better results have been obtained, for the forced convection case, without the need of implementing the LB model in FEMuS or the $\mathrm{K}_\theta - \Omega_\theta$ model in OpenFOAM. Once that the coupling routines have been developed, the approach used to simulate the backward facing step with OpenFOAM and FEMuS can be used in general.

Conclusions

Turbulent heat transfer is a very complex phenomenon where phenomena at different physical-space scales contribute to determine the fluid motion and its heat transport. Thermal turbulence modeling for low Prandtl number fluids is a challenging task required in engineering applications where liquid metals are used as operating fluids, e.g. concentrating solar power plants and IVth generation fast nuclear reactors. Turbulence models provided by commercial codes are not adequate for the simulation of low Prandtl number fluids, and more accurate and numerically stable models are needed to help the design process of these complex systems.

This PhD study has been realized following two main objectives: the investigation of a four parameters turbulence model capabilities and the development of a computational platform where more accurate simulations of low Pr turbulent flows can be performed. For the first objective, the four parameter turbulence model has been considered since it has already been validated with simulations of fully developed turbulent flows in straight geometries, with $Pr = 0.025$ [10]. A new formulation of the model, with logarithmic variables, has been proposed in this thesis and implemented in FEMuS code, with the intent to enhance the numerical stability of the original model. For the latter objective, a computational platform has been developed using the open-source software SALOME and the MED library. In particular, SALOME platform is used for geometry and mesh generation processes. The MED library is used as a standard data format for storing numerical fields and meshes in computer memory, allowing to perform data exchange between different codes within computer memory and not with much slower read/write operations on files. Several routines, based on the MED data format, have been developed as C++ classes to facilitate numerical code coupling, i.e. data exchange and manipulation. In particular, the routines have been developed to perform numerical field interpolation between different meshes and numerical field integration and can be generally used, alongside the other routines provided by MED library, since they are not code specific. With the computational platform, complex systems can be simulated using several codes, depending on the physical phenomena that need to be taken into account. For the simulation of low Pr turbulent flows, finite volume OpenFOAM and finite element FEMuS codes have been coupled, with the purpose of taking advantage of the wide set of dynamical turbulence models implemented in OpenFOAM and of the thermal turbulence model implemented in FEMuS.

Interpolation routines have been successfully used to define a new immersed boundary method. In this new method an indicator function is used to distinguish solid and fluid computational regions. Two grids are used, one for solving equations, and one for modeling the solid body. Indicator function field is obtained as a result of projection of a uniform field defined on solid body mesh to the computational grid. The method can then handle moving bodies with arbitrarily shapes. Two cases of fluid flow around static and moving objects, modeled with immersed boundary method, have been studied. Obtained results have been compared with reference data, showing a good agreement.

The case of fully developed turbulent flows in plane channel has been used to show the increased numerical stability of newly proposed logarithmic model with respect to original k-ε formulation. For the same geometry, validation against DNS results has been made for a wide range of Reynolds numbers and for both $Pr = 0.025$ and $Pr = 0.01$.

Fully developed turbulent flows in cylindrical pipes have also been simulated, over a wide range of Reynolds numbers, for both $Pr = 0.025$ and $Pr = 0.01$. Since DNS results for thermal turbulence fields at this low Prandtl number values are not available, obtained results have been compared with experimental correlations in the form of Nusselt number values. A very good agreement with reference Kirillov correlation has been obtained over the entire range of investigated Peclet values.

A fully developed turbulent flow of Lead-Bismuth-Eutectic has been simulated in the geometry of 19 pin hexagonal nuclear reactor bundle, using the four parameter turbulence model and also with constant turbulent Prandtl number $Pr_t = 0.9$ and Kays models. Results of Nusselt number and super-heat values have been compared with experimental values. Obtained Nusselt number values are close to Ushakov experimental correlation, in the range $\pm 5\%$, while Kays results lay in $\pm 10\%$ and those for $Pr_t = 0.9$ are much greater than reference correlation. A good agreement of super-heat values has been obtained for simulated cases at higher Peclet number values, while for $Pe < 700$ obtained results are higher than experimental ones. This fact could be related to effect of buoyancy forces, which is more sensible with small velocities and that has not been taken into account in the present simulations.

Simulations of liquid sodium turbulent flows over a vertical backward facing step geometry (BFS), with a $Pr = 0.0088$, have been performed for both forced and mixed convection cases, since DNS values are available for model validation. For this test case FEMuS, as standalone code, and coupled FEMuS-OpenFOAM codes have been used. From results of the uncoupled case one can see that the

KLW model leads to an underprediction of the recirculation length behind the step, with consequent slight discrepancies between obtained results and DNS ones in the recirculation region. Results have been improved with the coupled simulation, where Lam-Bremhorst $k - \varepsilon$ dynamical turbulence model, implemented in OpenFOAM, has been coupled with the thermal turbulence model implemented in FEMuS. With the coupled simulation a better estimation of the recirculation area has been observed, improving the accuracy of obtained results with respect to DNS values. Far from the recirculation region, similar results have been obtained between coupled and uncoupled simulations, with an overall good agreement with reference data. In the mixed convection case, due to the presence of buoyancy forces, the recirculation region is greatly reduced in size and no significant improvement has been observed from the coupled simulation. Again the obtained results are in satisfactory agreement with reference solutions.

With this PhD study the four parameter turbulence model capabilities and limitations have been investigated. For all simulated cases a good agreement with reference data, i.e. DNS results and experimental correlations, has been obtained. The isotropic turbulence hypothesis used to model the turbulent heat flux represents a limitation of the model since a single turbulent diffusion coefficient is not sufficient to reproduce both wall normal and stream-wise components of turbulent heat flux, as observed with BFS simulations. An anisotropic model for turbulent heat flux is planned to be investigated in order to overcome this issue. Thanks to the increased numerical stability and to the accuracy of the results, the KLW could be used with an anisotropic formulation of turbulent heat flux. The computational platform has then proven to be a useful tool since better results have been obtained, for the BFS forced convection case, without the need of implementing the Lam-Bremhorst $k - \varepsilon$ model in FEMuS or the $\mathrm{K}_\theta - \Omega_\theta$ model in OpenFOAM. The approach used to simulate this problem with coupled OpenFOAM and FEMuS codes can be used in general, also by adding additional codes to the platform, allowing the possibility to simulate very complex systems.

List of Figures

List of Tables

Bibliography

[1] S. Brenner and L. Scott. *The Mathematical Theory of Finite Element Methods*. Springer New York, 2002.

[2] G. Folland. *Real analysis: modern techniques and their applications*. Pure and applied mathematics. Wiley, 1999.

[3] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics, 2002.

[4] *OpenFOAM project*. URL: `https://www.openfoam.com/`.

[5] *Code TrioCFD*. URL: `http://www-trio-u.cea.fr`.

[6] *Code Saturne*. URL: `https://www.code-saturne.org/cms/`.

[7] *Code Aster*. URL: `https://www.code-aster.org/spip.php?rubrique2`.

[8] *Dragon and Donjon codes*. URL: `https://www.polymtl.ca/merlin/version5.htm`.

[9] *FEMuS code*. URL: `https://github.com/FemusPlatform/femus`.

[10] S. Manservisi and F. Menghini. "A CFD four parameter heat transfer turbulence model for engineering applications in heavy liquid metals". In: *Int. J. Heat Mass Tran.* 69 (2014), pp. 312–326.

[11] S. Manservisi and F. Menghini. "Triangular rod bundle simulations of a CFD k-ε-k_θ-ε_θ heat transfer turbulence model for heavy liquid metals". In: *Nucl. Eng. Des.* 273 (2014), pp. 251–270.

[12] S. Manservisi and F. Menghini. "CFD simulations in heavy liquid metal flows for square lattice bare rod bundle geometries with a four parameter heat transfer turbulence model". In: *Nucl. Eng. Des.* 295 (2015), pp. 251–260.

[13] *Computational platform*. URL: `https://github.com/FemusPlatform/NumericPlatform`.

[14] A. Ribes and C. Caremoli. “Salome platform component model for numerical simulation”. In: *31st Annual International Computer Software and Applications Conference (COMPSAC 2007)*. Vol. 2. July 2007, pp. 553–564.

[15] D. Cerroni. “Multiscale multiphysics coupling on a finite element platform”. PhD thesis. University of Bologna, 2015.

[16] D. Cerroni, R. Da Vià, S. Manservisi, F. Menghini, and R. Scardovelli. “CFD and Neutron codes coupling on a computational platform”. In: *Journal of Physics: Conference Series*. Vol. 796. 1. 2017.

[17] O. C. Zienkiewicz, R. L. Taylor, O. C. Zienkiewicz, and R. L. Taylor. *The finite element method*. Vol. 3. McGraw-hill London, 1977.

[18] J. N. Reddy. *An introduction to the finite element method*. Vol. 2. McGraw-Hill New York, 1993.

[19] G. H. Silva, R. Le Riche, J. Molimard, and A. Vautrin. “Exact and efficient interpolation using finite elements shape functions”. In: *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique* 18.3-4 (2009), pp. 307–331.

[20] D. Cerroni, R. Da Vià, and S. Manservisi. “A projection method for coupling two-phase VOF and fluid structure interaction simulations”. In: *J. Comput. Phys.* 354 (2018), pp. 646–671.

[21] A. Chierici, L. Chirco, R. Da Vià, S. Manservisi, and R. Scardovelli. “A multiscale numerical algorithm for heat transfer simulation between multidimensional CFD and monodimensional system codes”. In: *Journal of Physics: Conference Series*. Vol. 923. 1. 2017.

[22] P. Farrell and J. Maddison. “Conservative interpolation between volume meshes by local Galerkin projection ”. In: *Computer Methods in Applied Mechanics and Engineering* 200.1–4 (2011), pp. 89–100. ISSN: 0045-7825.

[23] D. Cerroni, R. Da Vià, S. Manservisi, and F. Menghini. “Multiscale simulation of a power-cooling system”. In: *AIP Conference Proceedings*. Vol. 1863. 2017.

[24] D. C. Wan, B. S. V. Patnaik, and G. W. Wei. “A new benchmark quality solution for the buoyancy-driven cavity by discrete singular convolution”. In: *Numer. Heat Tr. B-Fund.* 40.3 (2001), pp. 199–28.

[25] G. De Vahl Davis. “Natural convection of air in a square cavity: A bench mark numerical solution”. In: *Int. J. Numer. Meth. Fluids* 3.3 (1983), pp. 249–64.

[26] C. S. Peskin. “Flow patterns around heart valves: a numerical method”. In: *Journal of computational physics* 10.2 (1972), pp. 252–271.

[27] F. Ilinca and J.-F. Hétu. “A finite element immersed boundary method for fluid flow around rigid objects”. In: *Int. J. Numer. Methods Fluids* 65.7 (2011), pp. 856–875.

[28] A. Jendoubi, D. Yakoubi, A. Fortin, and C. Tibirna. “An immersed boundary method for fluid flows around rigid objects”. In: *Int. J. Numer. Methods Fluids* 75.1 (2014), pp. 63–80.

[29] F. Sotiropoulos and X. Yang. “Immersed boundary methods for simulating fluid-structure interaction”. In: *Prog. Aerosp. Sci.* 65 (2014), pp. 1–21.

[30] R. Mittal and G. Iaccarino. “Immersed boundary methods”. In: *Annu. Rev. Fluid Mech.* 37 (2005), pp. 239–261.

[31] G. Tryggvason, R. Scardovelli, and S. Zaleski. *Direct numerical simulations of gas–liquid multiphase flows*. Cambridge University Press, 2011.

[32] F. Ilinca and J.-F. Hétu. “Solution of flow around complex-shaped surfaces by an immersed boundary-body conformal enrichment method”. In: *Int. J. Numer. Methods Fluids* 69.4 (2012), pp. 824–841.

[33] F. Ilinca and J. F. Hétu. “An immersed boundary-body conformal enrichment method for thermal flow problems”. In: *Comput. Fluids* 88 (2013), pp. 616–628.

[34] F. Ilinca and J. F. Hétu. “A finite element immersed boundary method for fluid flow around moving objects”. In: *Comput. Fluids* 39.9 (2010), pp. 1656–1671.

[35] J. Dolbow, S. Mosso, J. Robbins, and T. Voth. “Coupling volume-of-fluid based interface reconstructions with the extended finite element method”. In: *Comput. Methods Appl. Mech. Eng.* 197 (2007), pp. 439–447.

[36] “A numerical method for solving the 3D unsteady incompressible Navier-Stokes equations in curvilinear domains with complex immersed boundaries”. In: *J. Comput. Phys.* 225.2 (2007), pp. 1782–1809.

[37] J. Guermond, P. Minev, and J. Shen. “An overview of projection methods for incompressible flows”. In: *Comput. Methods Appl. Mech. Eng.* 195.44 (2006), pp. 6011–6045.

[38] M. Schäfer et al. “Benchmark Computations of Laminar Flow Around a Cylinder”. In: *Flow Simulation with High-Performance Computers II: DFG Priority Research Programme Results 1993–1995*. Ed. by E. H. Hirschel. Wiesbaden: Vieweg+Teubner Verlag, 1996, pp. 547–566.

[39] M. Bar-Lev and H. T. Yang. “Initial flow field over an impulsively started circular cylinder”. In: *J. Fluid Mech.* 72.4 (1975), pp. 625–647.

[40] P. Koumoutsakos and A. Leonard. “High-Resolution simulations of the flow around an impulsively started cylinder using vortex methods”. In: *J. Fluid Mech.* 296 (1995), pp. 1–38.

[41] K. Taira and T. Colonius. “The immersed boundary method: A projection approach”. In: *J. Comput. Phys.* 225.2 (2007), pp. 2118–2137.

[42] C. Shang-Gui, O. Abdellatif, F. Julien, and H. Yannick. “Moving immersed boundary method”. In: *Int. J. Numer. Methods Fluids* 85.5 (2017), pp. 288–323.

[43] M. Coutanceau and R. Bouard. “Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 2. Unsteady flow”. In: *J. Fluid Mech.* 79.2 (1977), pp. 257–272.

[44] M. Chloe, G. Federico, C. Georges-Henri, and M. Iraj. “Vortex penalization method for bluff body flows”. In: *Int. J. Numer. Methods Fluids* 79.2 (2015), pp. 55–83.

[45] G. Kalitzin and G. Iaccarino. “Turbulence modeling in an immersed-boundary RANS method”. In: *Center for Turbulence Research Annual Research Briefs* (2002), pp. 415–426.

[46] S. Jeffrey G., B. Bruce R., P. W. David, and F. Paul F. “Effects of Mesh Density and Flow Conditioning in Simulating 7-Pin Wire Wrapped Fuel Pins”. In: *16th International Conference on Nuclear Engineering*. (Orlando, Florida). 2008.

[47] J. Pacio et al. “Heat transfer to liquid metals in a hexagonal rod bundle with grid spacers: experimental and simulation results”. In: *Nucl. Eng. Des.* 290 (2015), pp. 27–39.

[48] B. Landon, C. Lane, M. Elia, and H. Yassin. “CFD Investigation of Wire-Wrapped Fuel Rod Bundle Inner Subchannel Behavior and Dependency on Bundle Size”. In: *24th International Conference on Nuclear Engineering*. (Charlotte, North Carolina). 2016.

[49] J. Pacio, T. Wetzel, H. Doolaard, F. Roelofs, and K. V. Tichelen. “Thermal-hydraulic study of the LBE-cooled fuel assembly in the MYRRHA reactor: Experiments and simulations”. In: *Nucl. Eng. Des.* 312 (2017). 16th International Topical Meeting on Nuclear Reactor Thermal Hydraulics, pp. 327–337.

[50] L. Davidson. *Fluid mechanics , turbulent flow and turbulence modeling*. 2018.

[51] K. Hanjalic and B. Launder. *Modelling Turbulence in Engineering and the Environment*. Cambridge Books Online. Cambridge University Press, 2011.

[52] C. Chen. *Fundamentals Of Turbulence Modelling*. Combustion (New York). Taylor & Francis, 1997.

[53] W. D. McComb. *The physics of fluid turbulence*. Oxford Science Publications, 1990.

[54] P. Bradshaw. *An Introduction to Turbulence and its Measurement*. Pergamon, 1971.

[55] P. Bradshaw and G. P. . Huang. "The Law of the Wall in Turbulent Flow". In: *Proceedings: Mathematical and Physical Sciences* 451.1941 (1995), pp. 165–188.

[56] B. A. Kader and A. M. Yaglom. "Heat and mass transfer laws for fully turbulent wall flows". In: *Int. J. Heat Mass Tran.* 15.12 (1972), pp. 2329–2351.

[57] B. A. Kader. "Temperature and concentration profiles in fully turbulent boundary layers". In: *Int. J. Heat Mass Tran.* 24 (9 1981), pp. 1541–1544.

[58] M. Duponcheel, L. Bricteux, M. Manconi, G. Winckelmans, and Y. Bartosiewicz. "Assessment of RANS and improved near wall modeling for forced convection at low Prandtl numbers based on LES up to $Re_\tau = 2000$". In: *Int. J. Heat Mass Tran.* 75 (2014), pp. 470–482.

[59] J. Bredberg. *On Two-equation Eddy-Viscosity Models*. Tech. rep. Göteborg: Chalmers University of Technology, 2001.

[60] K. Hanjalic, S. Kenjeres, and F. Durst. "Natural convection in partitioned two-dimensional enclosures at higher Rayleigh numbers". In: *Int. J. Heat Mass Tran.* 39.7 (1996), pp. 1407–1427.

[61] N. Shikazono and N. Kasagi. "Second-moment closure for turbulent scalar transport at various Prandtl numbers". In: *Int. J. Heat Mass Tran.* 39.14 (1996), pp. 2977–2987.

[62] Y. Lai and R. So. "Near-wall modeling of turbulent heat fluxes". In: *Int. J. Heat Mass Tran.* 33.7 (1990), pp. 1429–1440.

[63] T. Sommer and R. So. "Wall-bounded buoyant turbulent flow and its modeling". In: *Int. J. Heat Mass Tran.* 39.17 (1996), pp. 3595–3606.

[64] R. So and T. Sommer. "An explicit algebraic heat-flux model for the temperature field". In: *Int. J. Heat Mass Tran.* 39.3 (1996), pp. 455–465.

[65] Y. Nagano. “Modelling Heat Transfer in Near-Wall Flows”. In: *Closure Strategies for Turbulent and Transitional Flows*. Cambridge University Press, 2002, pp. 188–247.

[66] H. Hattori, Y. Nagano, and M. Tagawa. “Analysis of Turbulent Heat Transfer under Various Thermal Conditions with Two-Equation Models”. In: *Engineering Turbulence Modelling and Experiments*. Ed. by W. Rodi and F. Martelli. Oxford: Elsevier, 1993, pp. 43–52.

[67] Y. Nagano and C. Kim. “A Two-Equation Model for Heat Transport in Wall Turbulent Shear Flows”. In: *J. Heat Transfer* 110.3 (1988), pp. 583–589.

[68] Y. Nagano, M. Kondoh, and M. Shimada. “Multiple time-scale turbulence model for wall and homogeneous shear flows based on direct numerical simulations”. In: *Int. J. Heat Fluid Fl.* 18.97 (1997), pp. 346–359.

[69] M. Youssef, Y. Nagano, and M. Tagawa. “A two-equation heat transfer model for predicting turbulent thermal fields under arbitrary wall thermal conditions”. In: *Int. J. Heat Mass Tran.* 35.11 (1992), pp. 3095–3104.

[70] K. Abe, T. Kondoh, and Y. Nagano. “A new turbulence model for predicting fluid flow and heat transfer in separating and reattaching flows-II. Thermal field calculations”. In: *Int. J. Heat Mass Tran.* 38.8 (1995), pp. 1467–1481.

[71] F. Menghini. “Advanced computational fluid dynamics models for liquid metal flows”. PhD thesis. University of Bologna, 2015.

[72] R. Da Vià, S. Manservisi, and F. Menghini. “A k–Ω–k_θ–Ω_θ four parameter logarithmic turbulence model for liquid metals”. In: *Int. J. Heat Mass Tran.* 101 (2016), pp. 1030–1041.

[73] R. Da Vià and S. Manservisi. “Numerical simulation of forced and mixed convection turbulent liquid sodium flow over a backward facing step with a four parameter turbulence model”. In: *Int. J. Heat Mass Tran.* Under Review (2018).

[74] Y. Nagano and M. Shimada. “Development of a two equation heat transfer model based on direct simulations of turbulent flows with different Prandtl numbers”. In: *Phys. Fluids* 8.12 (1996), pp. 3379–3402.

[75] K. Abe, T. Kondoh, and Y. Nagano. “A new turbulence model for predicting fluid flow and heat transfer in separating and reattaching flows – I. Flow field calculations”. In: *Int. J. Heat Mass Tran.* 37.1 (1994), pp. 139–151.

[76] A. De Santis and A. Shams. “Application of an algebraic turbulent heat flux model to a backward facing step flow at low Prandtl number”. In: *Annals of Nuclear Energy* 117 (2018), pp. 32–44.

[77] F. Ilinca, J. Hetu, and D. Pelletier. “A unified finite element algorithm for two - equation models of turbulence”. In: *Comput. Fluids* 27.3 (1998), pp. 291–310.

[78] P. A. Durbin. “On the k-ϵ stagnation point anomaly”. In: *Int J. Heat Fluid Fl.* 17.1 (1996), pp. 89–90.

[79] C. H. Park and S. O. Park. “On the limiters of two-equation turbulence models”. In: *Int. J. Comput. Fluid D.* 19.1 (2005), pp. 79–86.

[80] Z. Ge, J. Liu, P. Zhao, X. Nie, and M. Ye. “Investigation on the applicability of turbulent-Prandtl-number models in bare rod bundles for heavy liquid metals”. In: *Nucl. Eng. Des.* 314 (2017), pp. 198–206.

[81] Y. Saad and M. Schultz. “GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems”. In: *SIAM Journal on Scientific and Statistical Computing* 7.3 (1986), pp. 856–869.

[82] T. Schumm, M. Niemann, F. Magagnato, L. Marocco, B. Frohnapfel, and J. Fr. “Numerical prediction of heat transfer in liquid metal applications”. In: 2000 (2015), pp. 1–12.

[83] W. Jaeger et al. “Thermo-hydraulic flow in a sudden expansion”. In: *IOP Conf. Ser. Mater. Sci. Eng* 228.1 (2017).

[84] T. Schumm, B. Frohnapfel, and L. Marocco. “Numerical simulation of the turbulent convective buoyant flow of sodium over a backward facing step”. In: *J. of Phys.: Conf. Series* 745 (2016), p. 032051.

[85] L. Marocco, A. A. di Valmontana, and T. Wetzel. “Numerical investigation of turbulent aided mixed convection of liquid metal flow through a concentric annulus”. In: *Int. J. Heat Mass Tran.* 105 (2017), pp. 479–494.

[86] H. Kawamura, H. Abe, and Y. Matsuo. “DNS of turbulent heat transfer in channel flow with respect to Reynolds and Prandtl number effects”. In: *Int. J. Heat Fluid Fl.* 20.3 (1999), pp. 196–207.

[87] I. Tiselj and L. Cizelj. “DNS of turbulent channel flow with conjugate heat transfer at Prandtl number 0.01”. In: *Nucl. Eng. Des.* 253 (2012), pp. 153–160.

[88] S. Hoyas and J. Jiménez. “Reynolds number effects on the Reynolds-stress budgets in turbulent channels”. In: *Physics of Fluids* 20.10 (2008).

[89] A. Lozano-Durán and J. Jiménez. “Effect of the computational domain on direct simulations of turbulent channels up to $Re_\tau = 4200$”. In: *Physics of Fluids* 26.1 (2014).

[90] H. Abe, H. Kawamura, and Y. Matsuo. “Surface heat-flux fluctuations in a turbulent channel flow up to $Re_\tau = 1020$ with Pr=0.025 and 0.71”. In: *International Journal of Heat and Fluid Flow* 25.3 (2004), pp. 404–419.

[91] X. Cheng and N.-i. Tak. “Investigation on turbulent heat transfer to lead–bismuth eutectic flows in circular tubes for nuclear applications”. In: *Nuclear Engineering and Design* 236.4 (2006), pp. 385–393.

[92] A. J. Musker. “Explicit expression for the smooth wall velocity distribution in a turbulent boundary layer”. In: *AIAA Journal* 17 (1979), pp. 655–657.

[93] W. M. Kays. “Turbulent Prandtl Number - Where Are We?” In: *J. Heat Transfer* 116 (Apr. 1994), pp. 284–295.

[94] R. Lyon. “Liquid metal heat transfer coefficients”. In: *Chemical Engineering Progress* 47 (1951), pp. 75–79.

[95] R. Lyon. *Liquid-Metals Handbook*. Atomic Energy Comm., 1952.

[96] E. Skupinski, J. Tortel, and L. Vautrey. “Determination des coefficients de convection d’un alliage sodium-potassium dans un tube circulaire”. In: *Int. J. Heat Mass Tran.* 8 (1965), pp. 937–951.

[97] C. A. Sleicher, A. S. Awad, and R. H. Notter. “Temperature and eddy diffusivity profiles in NaK”. In: *Int. J. Heat Mass Tran.* 16 (1973), pp. 1565–1575.

[98] M. Ibragimov, V. Subbotin, and P. Ushakov. “Investigation of heat transfer in the turbulent flow of liquid metals in tubes”. In: *Atomnaya Energiya* 8 (1) (1960), pp. 54–56.

[99] W. K. Stromquist. “Effect of wetting on heat transfer characteristics of liquid metals”. In: *ORO-93* (1953). University of Tennessee.

[100] P. Kirillov and P. Ushakov. “Heat transfer to liquid metals: specific features, methods of investigation, and main relationships”. In: *Thermal engineering* 48.1 (2001), pp. 50–59.

[101] H. Johnson, J. Hartnett, and W. Clabaugh. “Heat transfer to molten lead–bismuth eutectic in turbulent pipe flow”. In: *Journal of Heat Transfer* (1953), pp. 1191–1198.

[102] J. Pacio et al. “Heavy-liquid metal heat transfer experiment in a 19-rod bundle with grid spacers”. In: *Nucl. Eng. Des.* 273 (2014), pp. 33–46.

[103] D. Martelli, R. Marinari, G. Barone, I. di Piazza, and M. Tarantino. “CFD thermo-hydraulic analysis of the CIRCE fuel bundle”. In: *Ann. Nucl. Energy* 103 (2017), pp. 294–305.

[104] J.-H. Jeong, J. Yoo, K.-L. Lee, and K.-S. Ha. “Three-dimensional flow phenomena in a wire-wrapped 37-pin fuel bundle for SFR”. In: *Nucl. Eng. Des.* 47.5 (2015), pp. 523–533.

[105] P. Ushakov, A. Zhukov, and N. Matyukhin. “Heat transfer to liquid metals in regular arrays of fuel elements”. In: *High Temp. (USSR)* 15 (Mar. 1978), pp. 1027–1033.

[106] K. Mikityuk. “Heat transfer to liquid metal: Review of data and correlations for tube bundles”. In: *Nucl. Eng. Des.* 239.4 (2009), pp. 680–687.

[107] M. Niemann and J. Fröhlich. “Buoyancy-affected backward-facing step flow with heat transfer at low Prandtl number”. In: *Int. J. Heat Mass Tran.* 101 (2016), pp. 1237–1250.

[108] M. Niemann and J. Fröhlich. “Turbulence Budgets in Buoyancy-affected Vertical Backward-facing Step Flow at Low Prandtl Number”. In: *Flow Turbul. Combust.* 99.3-4 (2017), pp. 705–728.

[109] M. Niemann and J. Fröhlich. “Direct Numerical Simulation of turbulent heat transfer behind a backward-facing step at low Prandtl number”. In: *Proc. Appl. Math. Mech* 14 (2014), pp. 659–660.

[110] P. Zhao, Z. Ge, J. Zhu, J. Liu, and M. Ye. “Quasi-direct numerical simulation of forced convection over a backward-facing step: Effect of Prandtl number”. In: *Nucl. Eng. Des.* 335.August (2018), pp. 374–388.

[111] P. Zhao, C. Wang, Z. Ge, J. Zhu, J. Liu, and M. Ye. “DNS of turbulent mixed convection over a vertical backward-facing step for lead-bismuth eutectic”. In: *Int J. Heat Fluid Fl.* 127 (2018), pp. 1215–1229.

[112] M. Niemann and J. Fröhlich. “Buoyancy Effects on Turbulent Heat Transfer Behind a Backward-Facing Step in Liquid Metal Flow”. In: *Direct and Large - Eddy Simulation IX*. Springer International Publishing, 2018, pp. 513–519.

[113] T. Schumm, B. Frohnapfel, and L. Marocco. “Investigation of a turbulent convective buoyant flow of sodium over a backward-facing step”. In: *Heat Mass Transf* (2017), pp. 1–11.

[114] B. Launder and B. Sharma. “Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc”. In: *Letters in Heat and Mass Transfer* 1.2 (1974), pp. 131–137.

[115] A. N. Brooks and T. J. Hughes. “Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations”. In: *Comput Methods Appl Mech Eng* 32.1 (1982), pp. 199–259.

[116] V. Sobolev. *Database of thermophysical properties of liquid metal coolants for GEN-IV*. 2011.

[117] V. Yakhot, S. A. Orszag, S. Thangam, T. B. Gatski, and C. G. Speziale. “Development of turbulence models for shear flows by a double expansion technique”. In: *Physics of Fluids A* 4 (July 1992), pp. 1510–1520.

[118] C. Lam and K. Bremhorst. “A Modified Form of the k-ε Model for Predicting Wall Turbulence”. In: 103 (1981), pp. 456–460.

[119] F. Lien, W. Chen, and M. Leschziner. “Low-Reynolds-Number Eddy-Viscosity Modelling Based on Non-Linear Stress-Strain/Vorticity Relations”. In: *Engineering Turbulence Modelling and Experiments*. Ed. by W. Rodi and G. Bergeles. Vol. 3. Elsevier Series in Thermal and Fluid Sciences. Oxford: Elsevier, 1996, pp. 91–100.

[120] B. Launder, G. J. Reece, and W. Rodi. “Progress in the Development of a Reynolds Stress Turbulence Closure”. In: 68 (Apr. 1975), pp. 537–566.

[121] A. Karvinen and H. Ahlstedt. “Comparison of Turbulence Models in Case of Three-Dimensional Diffuser”. In: *Open Source CFD International Conference*. (Berlin, Germany). 2008.