High order turbulence statistics of conjugate heat transfer

Incompact3d DNS code extension to solid wall conduction

A thesis submitted to The University of Manchester for the degree of
Master of Philosophy
in the faculty of Engineering and Physical Sciences

2014

Cedric Flageul

School of Mechanical, Aerospace and Civil Engineering

Contents

Word count : 17233

Abstract

This is the abstract of the thesis submitted in 2014 by Cédric Flageul to The University of Manchester for the degree of Master of Philosophy in the faculty of Engineering and Physical Sciences entitled : *High order turbulence statistics of conjugate heat transfer, Incompact3d DNS code extension to solid wall conduction.*

More than 40 years ago, DNS (Direct Numerical Simulation) started with the pioneering work of Orszag [25]. This powerful workbench to study turbulence is attracting rising numbers of researchers as the computational power available increases. However, few DNS have dealt with conjugate heat transfer issues. Tiselj et al. [31] were among the first to investigate this more realistic situation in a channel flow using pseudo-spectral methods. Kang et al. [12] performed Quasi-DNS with conjugate heat transfer of a heated cylinder in a channel using immersed boundaries and fully implicit LES solver based on unstructured collocated mesh.

Conjugate heat transfer is a major issue in industrial applications where cooling of hot components is often critical. In complex situations, RANS and LES simulations rely on wall-function modelling as the viscous sub-layer is not resolved. DNS is a valuable tool for understanding the flow physics of such complex phenomena and to provide fine data in order to improve RANS and LES approaches.

This document summarizes the developments performed in the open-source DNS code Incompact3d to investigate conjugate heat transfer. Budgets of second-order statistics for a channel flow with different thermal boundary conditions are presented (imposed temperature, imposed heat flux and conjugate heat transfer). To the author's knowledge, budgets for imposed heat flux and conjugate heat transfer are not available in the literature. Such data is valuable in the scope of assessing accuracy of wall-modeling.

Declaration

No portion of the work referred to in this dissertation has been submitted in support of an application for another degree or qualification of this or any other university or any other institute of learning.

Copyright Statement

To laughters and tears
To my family and friends
To my teachers and supervisors
Thank you.

Nick: Here's a thought. Why don't we get an egg and start our own chicken farm? That way we'd have all the eggs we could eat.
Fetcher: Right. We'll need a chicken, then.
Nick: No... no, we'll need an egg. You have the egg first, that's where you get the chicken from.
Fetcher: No, that's cobblers. If you don't have a chicken, where are you going to get the egg?
Nick: From the chicken that comes from the egg.
Fetcher: Yeah, but you have to have an egg to have a chicken.
Nick: Yeah, but you've got to get the chicken first to get the egg, and then you get the egg... to get the chicken out of...
Fetcher: Hang on. Let's go over this again?
...
Nick: Eggs from heaven.
Fetcher: No! From her bum.

Chicken Run

Chapter 1

Introduction

Fluid dynamics is a branch of physic dedicated to the motion of fluids. The evolution of a fluid flow is ruled by the conservation of mass, momentum and energy. The equation for the conservation of momentum is called Navier-Stokes equation from Claude-Louis Navier and Sir George Stokes. It is nearly 200 years old and the Clay Mathematics Institute has raised it as one of the 7 most important open problems in mathematic. In most situations, this non-linear equation leads the fluid flow to a chaotic behaviour called turbulence.

Many industrial applications use a fluid as the medium to exchange heat between hot and cold parts of a machinery. For instance, a Central Processing Unit is often cooled with a fan blowing air. A Pressurized Water Reactor uses water to transport the heat produced by the fission reaction. This water is also used to cool the Reactor Pressure Vessel if necessary. Any case involving heat-transfer between a fluid and a solid can be studied using conjugate heat-transfer.

The objective of the present study is to develop, on top of a existing Direct Numerical Simulation code, subroutines to solve the transport equation of a passive scalar inside the fluid and solid domain and to couple them.

1.1 Pressurized thermal shock

Pressurized Thermal Shock (PTS) is an important topic in the nuclear community, both related to safety and nuclear plant's life time. According to the IAEA [10], it needs a « multidisciplinary effort »and « involves thermal hydraulic analysis, neutron field calculations and structural analysis including fracture mechanics assessment and evaluation of material properties». A PTS event is characterized by a rapid injection in the primary coolant system of significantly colder fluid than normal conditions, usually with high level of primary system pressure.

For example, it can occur during an emergency core cooling when cold water is injected at high pressure to cool the core. During the injection, the reactor pressure vessel is subject to a severe thermal load. Repetition of such events can lead to fracture and this limits the plant's life time, as the steel vessel becomes increasingly brittle over a long time. Figure 1.1 illustrates a cold injection.

Figure 1.2 is a snapshot of the temperature field during a coolant injection's simulation in a reactor pressure vessel. The cold fluid is denser and forms an upside down plume. The rate of flow at the injection is not very high and the cold fluid cascades alongside the

outer wall. Injections are performed simultaneously at the 3 or 4 cold legs. The interaction between the plumes leads to an unsteady meandering flow.

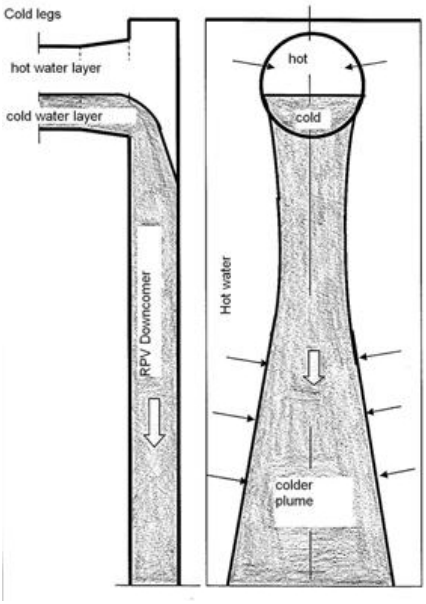

Figure 1.1: Cold injection plume [10]

PTS situations have been studied experimentally by many institutions and companies. Among them are EDF (France), MPA Stuttgart and HDR (Germany), JAPEIC (Japan), Prometey (Russia), AEA-Technology (United-Kingdom) and ORNL (USA). The snapshot in figure 1.2 comes from a RANS simulation. More recently, LES simulations have been performed on a PTS case: [23] and [22]. Full-scale PTS simulations cannot be fully resolved because the Reynolds number is too high for CFD meshes to capture the very thin viscous sub layer at the wall (even with millions of mesh points) so wall-modelling must be used. We aim to set up test cases of simplified PTS situations accessible for DNS. On this basis, RANS, LES simulations and wall-modelling can be compared with DNS simulation results that are free from any modelling hypothesis.

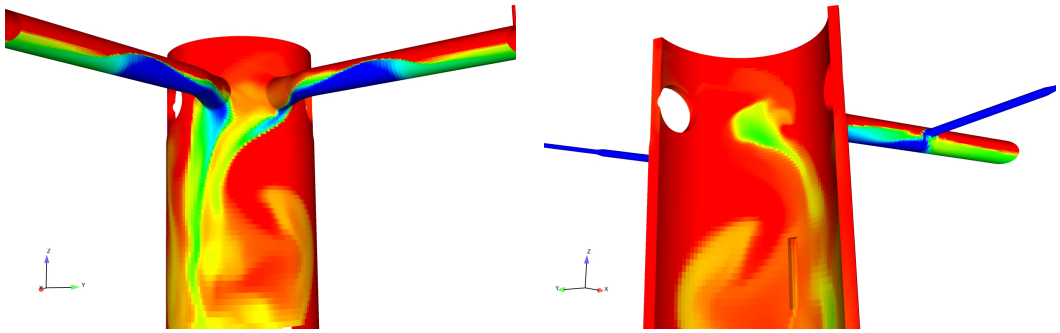

Figure 1.2: Temperature field in a RANS PTS Simulation, courtesy of EDF-R&D

1.2 Theoretical background

The conservation of momentum and energy is the root of fluid dynamic. From Neother's theorem, it is equivalent to the invariance of the law of physics when a translation in time or space happens. We restrict the topic to non-relativistic evolution and deduce the conservation of mass from the conservation of energy. Moreover, we work under the assumption of continuum mechanics: the matter is continuously distributed and fills the space. It means the Knudsen number K_n is much smaller than 1 (mean free molecule path much lower than the length of every turbulent structure).

In the following, an Eulerian specification of the flow field is used. Let $m(V, x, t)$ the mass in a volume V centred on the location x at time t. When the volume V goes to 0, the mass m also goes to 0 while the ratio m over V tends to $\rho(x, t)$. This is the mass per unit volume at location x and time t. The momentum $\rho\overrightarrow{u}$ and energy ρe per unit volume

are defined similarly.

The conservation of mass is expressed equation (1.1). When the flow is incompressible with an initially uniform density, the density remains constant and the equation simplifies to $\partial_i u_i = 0$.

$$\partial_t \rho + \partial_i (\rho u_i) = 0 \Rightarrow \partial_i u_i = 0 \text{ if } \rho \text{ is constant} \tag{1.1}$$

The conservation of momentum for an incompressible flow is expressed by equations (1.2) with an external forcing f_i (e.g. gravity, Lorentz force, ...). One defines the pressure p as minus the isotropic part of the Cauchy stress tensor σ: $p = -\frac{\sigma_{ii}}{3}$. Assuming the fluid is Newtonian, the deviatoric or shear stress tensor τ is related to the constant viscosity μ: $\tau_{ij} = \mu (\partial_i u_j + \partial_j u_i)$. The convective transport term (second one on the l.h.s) is non-linear and plays an important role in the chaotic behaviour while the viscous one (second one on the r.h.s) is responsible for the damping.

$$\begin{aligned} \rho (\partial_t u_i + \partial_j (u_i u_j)) &= \partial_j \sigma_{ij} + f_i \\ \rho (\partial_t u_i + \partial_j (u_i u_j)) &= -\partial_i p + \mu \partial_{j,j} u_i + f_i \end{aligned} \tag{1.2}$$

Derivation of the momentum equation in direction i and summing leads to the pressure Poisson equation (1.3). This is an elliptic PDE. Using Green's functions, we can write explicitly a solution, (1.4), to the Poisson equation (see [26] p.19). Modifying the velocity locally (therefore the velocity gradients) affects the pressure everywhere: the pressure is non-local and non-linear. In fact, the incompressibility condition is equivalent to assuming an infinite sound speed in the flow.

$$\partial_{ii} p = -\rho \partial_i u_j \partial_j u_i \tag{1.3}$$

$$p(\mathbf{x}, t) = p^{(h)}(\mathbf{x}, t) + \frac{\rho}{4\pi} \int\int\int (\partial_i u_j \partial_j u_i) \frac{\mathbf{dy}}{|\mathbf{x} - \mathbf{y}|} \text{ with } \partial_{ii} \left(p^{(h)} \right) = 0 \tag{1.4}$$

When the flow is not isothermal, evolution of temperature must be computed. We note H the total enthalpy per volume unit: $H = E_{total} + p$. From H we define h: $h = H - \rho e_c$ with ρe_c the local kinetic energy per volume unit. We assume the heat capacity at constant pressure (C_p) to be constant and $h = \rho C_p T$. T is the static temperature. We note κ the thermal conductivity. From the first law of thermodynamic, one finds that the evolution of h for an incompressible flow is ruled by a classic convection-diffusion equation with a source term linked to viscous dissipation (1.5) (see [1] p.204). The 3 last terms often lead to negligible temperature rise ($U^2 \ll C_p T$ and $U^2 \mu \ll \kappa T$).

$$\rho C_p (\partial_t T + \partial_j (T u_j)) = \kappa \partial_{j,j} T + \partial_t p + u_j \partial_j p + \frac{\mu}{2} (\partial_j u_i + \partial_i u_j)^2 \tag{1.5}$$

In some cases, a passive scalar is transported by the flow (e.g. the salt concentration or dye in experimental rigs). In this case, the scalar concentration per volume unit Γ is ruled by a classic convection-diffusion equation (1.6) where the diffusivity A is assumed constant.

$$\partial_t \Gamma + \partial_j (\Gamma u_j) = A \partial_{j,j} \Gamma \tag{1.6}$$

We have assumed ρ, μ, C_p and κ constant, the flow incompressible and the fluid Newtonian. On this basis, 5 equations have been derived to compute the evolution of 5 variables (pressure, velocity and temperature). This set of equations can produce chaotic solutions. Properties of chaotic systems are analyzed using statistical tools.

1.3 Dimensional analysis

One of the most basic (and important) matter in physic is dimensional analysis. Each quantity has a dimension: kilogram for mass, meter for space, second for time, ... The International System of Units gives seven base units which can be used to decompose any quantity. For example, a force of $1N$ (N for Newton) is equivalent to a mass of $1kg$ accelerating at $1m.s^{-2}$. Every mathematical relation connecting quantities must have balancing base units (see [34] or [3]). This is particularly relevant in convective heat transfer with at least 4 base units (time, length, mass, temperature) and many different dimensional parameters and variables.

At a first glance, dimensional analysis allows to check the consistency of a formula. It is also used to make rough estimations coming from number of experiments at many different scales. Let say we have an explosion at time $t = 0$, we take a picture at time t from which we deduce the radius of the blast R. The explosion has energy E and happens in a fluid of mass ρ. We assume $E \propto t^x R^y \rho^z$. E is in $J = kg.m^2.s^{-2}$, t is in s, R is in m and ρ is in $kg.m^{-3}$. Dimensional analysis leads to $z = 1$, $y = 5$ and $x = -2$. Assuming the relation is true whatever the explosion, using a reference small explosion and a picture of a large one, energy can be estimated.

Application of dimensional analysis is used to form dimensionless equations. Our base set of equations is (1.1), (1.2) and (1.5). We introduce some dimensionless numbers below. When there is no external forcing, the final set of equations reduces to (1.7).

- Re: Reynolds number, ratio between convection and diffusion for velocity, $Re = \frac{\rho U_0 X_0}{\mu}$
- Pr: Prandtl number, ratio between viscous and thermal diffusion rate, $Pr = \frac{C_p \mu}{\kappa}$

$$\begin{aligned} \partial_i u_i &= 0 \\ \partial_t u_i + \partial_j (u_i u_j) &= -\partial_i p + \frac{1}{Re} \partial_{j,j} u_i \\ \partial_t T + \partial_j (T u_j) &= \frac{1}{RePr} \partial_{j,j} T \end{aligned} \tag{1.7}$$

One of the most famous applications of dimensional analysis to turbulence was made by Kolmogorov [15]. Here, we note ϵ the dissipation rate in the flow: it is the energy dissipated per mass and time unit ($J.kg^{-1}.s^{-1}$). The hypothesis of Kolmogorov can be expressed in many ways, below are approximately the ones found in [26] p.185-186.

Kolmogorov's first similarity hypothesis. *In every turbulent flow at sufficiently high Reynolds number, the statistics of the small-scale motion have a universal form that is uniquely determined by ν and ϵ.*

Kolmogorov's second similarity hypothesis. *In every turbulent flow at sufficiently high Reynolds number, the statistics of the motion of scale l in the inertial subrange have a universal form that is independent of ν and uniquely determined by ϵ.*

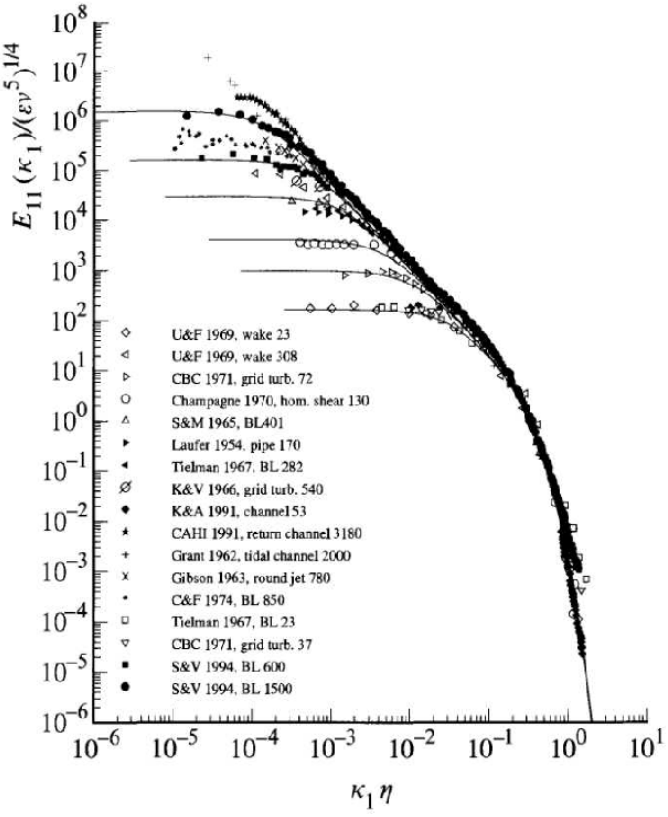

Figure 1.3: Velocity spectra from [26]

For the small-scale motion, using ϵ and ν, one can define the Kolmogorov scale for space η, velocity u_η and time τ_η: (1.8). The Kolmogorov scales characterize the very smallest and dissipative eddies.

$$\begin{aligned} \eta^4 &\propto \frac{\nu^3}{\epsilon} \\ {u_\eta}^4 &\propto \epsilon\nu \\ {\tau_\eta}^2 &\propto \frac{\nu}{\epsilon} \\ \epsilon &\propto \nu\left(\frac{u_\eta}{\eta}\right)^2 \end{aligned} \tag{1.8}$$

In the inertial sub-range, from the dissipation rate ϵ and the scale l, one can define a characteristic velocity $u(l)$ and time $\tau(l)$ for the motion at scale l: (1.9).

$$\begin{aligned} u(l)^3 &\propto \epsilon l \\ \tau(l)^3 &\propto \frac{l^2}{\epsilon} \\ \epsilon &\propto \frac{u(l)^2}{\tau(l)} \end{aligned} \tag{1.9}$$

We assume l to be in the inertial sub-range, we note k the wavenumber $\frac{2\pi}{l}$ and $E(k)$ the density of energy per mass unit contained in eddies of wave-number k. Energy of eddies with a wave-number between k and $k+dk$ is $E(k)dk$. Dimension of $E(k)$ is $J.m.kg^{-1} = m^3.s^{-2}$. Dimensional analysis (1.10) predicts the energy density dependence on the wavenumber k is a power $-\frac{5}{3}$ in the inertial sub-range. This dependency for the energy spectrum was observed experimentally for a huge number of flow configurations (figure 1.3) and made Kolmogorov's theory famous.

$$E(k) \propto \epsilon^{\frac{2}{3}} k^{-\frac{5}{3}} \tag{1.10}$$

Behind those excellent predictions lie strong assumptions. Turbulence is characterized

by intermittency: rare and violent events happening at small scales. Kolmogorov's theory does not explain those events and their magnitude.

Other theories are built on the basis of Kolmogorov's one and aim at a more complete description of turbulence. Some of the investigations assume ϵ is a random variable and depends on the wave-number k. See [30] for example. Turbulence is still an open area of research, and according to Richard Feynman, Nobel prize-winning physicist, *the most important unsolved problem of classical physics...*

1.4 Computational fluid dynamics

Computational fluid dynamics uses numerical methods to solve and analyze problems involving fluid flows. It is extensively used because for most of the situations, no exact solutions can be found. Depending on the complexity of the problem, on the computational power available and on the accuracy needed, one can choose over a broad range of methods. Hereafter are presented a few *physical* models to simulate turbulent flows. Each model can be adapted numerically in many ways.

Direct Numerical Simulation

DNS started in 1971 for homogeneous isotropic turbulence of an incompressible flow with Orszag [25]. It was then extended to both anisotropic and compressible cases. It requires a resolution in time and space at least as fine as the scales predicted by Kolmogorov. In addition, the computation domain must be big enough to resolve the integral turbulence scale. It is therefore very expensive computationally. For a cubic domain with a volume L^3, the ideal resolution is at least $\left(\frac{L}{\eta}\right)^3 \propto Re^{\frac{9}{4}}$ grid nodes. In addition, the turbulence should be resolved in time. The ratio of characteristic time between the macroscopic eddies and the smallest turbulent one scales like $\frac{\tau_L}{\tau_\eta} \propto Re^{\frac{1}{2}}$. Therefore, the computational cost for ideally resolved DNS vary as $Re^{\frac{11}{4}}$ (see [4]). The Reynolds number accessible by DNS in 1987 was 10^3 ([14]). In 2006 , 19 years later, it was 150 times higher ([11]). If computational power continues to increase exponentially, as it has indeed followed Moore's law over the past 40 years, flows with a reasonably extended inertial range will soon be accessible by means of DNS. Having a large inertial range isolates the large energy-containing eddies from the small dissipative ones and is of particular interest as a universal range (i.e. present with similar characteristics in most of the *natural* and industrial flows).

The main advantage of DNS is the possibility to measure numerically any quantity at any location and time without influencing the flow. One should keep in mind that whatever the resolution in time and space is, the accuracy of results can still depend on user choices, e.g. the size of the statistical sample (number of time-steps) may be insufficient to obtain reliable high order moments which can be influenced by rare events, or the finite size of the computational domain might prevent very large structures from appearing. Deeper insight into DNS and the related numerical tools in section 2.1 page 17.

Large Eddy Simulation

When performing a LES, large scales are resolved while small scales are modelled. Therefore, LES are computationally less expensive compared to DNS. The resolved variables ($\underline{\overline{u_i}}$) are filtered turbulent variables and depend on the filter function G used, see equation (1.11). The over- and under-lines denote filtered values, unlike Reynolds averaged ones which will be noted by a single over-line. For the sake of simplicity, we assume the filter is homogeneous. The difference between turbulent and filtered variables is called residual field. Applying the homogeneous filter to incompressible Navier-Stokes equation leads to (1.11). Evolution of filtered momentum is influenced by the residual stress (τ_{ij}^R). It appears as a source term, and is linked to unresolved small scales that must be modelled.

$$\begin{aligned} \partial_t\left(\rho\ \underline{\overline{u_i}}\right) + \partial_j\left(\rho\ \underline{\overline{u_i}}\ \underline{\overline{u_j}}\right) &= -\partial_i\left(\underline{\overline{p}}\right) + \mu\partial_{j,j}\underline{\overline{u_i}} - \partial_j\left(\rho\tau_{ij}^R\right) \text{ with } \tau_{ij}^R = \underline{\overline{u_i u_j}} - \underline{\overline{u_i}} \times \underline{\overline{u_j}} \\ \underline{\overline{u_i}}(\mathbf{x},t) &= \int\int u_i(\mathbf{x'},t')G(\mathbf{x\text{-}x'},t-t')\mathbf{dx'}dt' \end{aligned} \tag{1.11}$$

One of the simplest model was proposed by Smagorinsky in 1963 [29], see (1.12). The first term on the r.h.s. can be incorporated in a modified pressure. The anisotropic part of the residual stress uses a linear eddy-viscosity model: it is proportional to the filtered rate of strain $\underline{\overline{S_{ij}}}$. The eddy-viscosity ν_r is modelled with a mixing-length hypothesis. l_s is the Smagorinsky length scale often expressed as: $l_s = C_s\Delta$. Here, Δ is the filter width and C_s the Smagorinsky coefficient. However, the Smagorinsky coefficient is not a constant: it depends on the type of flow and on the filter used. This model is soundly justified for high Reynolds number flow with a filter width in the inertial sub-range. It is known to have deficiencies in the near-wall region and at low Reynolds number (in both cases, the inertial range no longer exists) or when the filter width is inadequate (e.g. out of the inertial sub-range).

$$\tau_{ij}^R = \frac{1}{3}\tau_{ll}^R\delta_{i,j} - 2\nu_r\underline{\overline{S_{ij}}} \text{ with } \underline{\overline{S_{ij}}} = \frac{1}{2}\left(\partial_j\underline{\overline{u_i}} + \partial_i\underline{\overline{u_j}}\right) \text{ and } \nu_r = {l_s}^2\sqrt{2\underline{\overline{S_{ij}}}\ \underline{\overline{S_{ij}}}} \tag{1.12}$$

There are many possibilities for modelling the residual stress or the eddy-viscosity. For example, Germano proposed in 1991 [6] a dynamic subgrid-scale eddy-viscosity model with a refined mixing-length hypothesis. In addition, other approaches like the spectral eddy-viscosity [5] or using transport equation for the residual kinetic energy [7] are possible. In addition to the filter function and the residual stress model, the numerical implementation should be considered carefully: the discretization and schemes used may produce an additional *numerical* stress.

Reynolds-Averaged Navier-Stokes

We assume that the flow is stationary and decompose every variable in a temporal average and a fluctuating part: $u(x,t) = \overline{u}\,(x) + u'(x,t)$. After averaging in time the base set of equations [(1.1), (1.2), (1.5)], one finds [(1.13), (1.14), (1.15)]. This set of equations is the Reynolds-averaged Navier-Stokes equations for an incompressible flow and comes from Osborne Reynolds. The non-linearity connects the correlation between fluctuations $\overline{u_i'u_j'}$

to the temporal average $\overline{u_i}$. These correlations are called Reynolds stresses or turbulent stresses and have to be modelled.

$$\partial_i \overline{u_i} = 0 \tag{1.13}$$

$$\begin{aligned} \rho\left(\partial_t \overline{u_i} + \partial_j\left(\overline{u_i u_j}\right)\right) &= -\partial_i \overline{p} + \mu \partial_{j,j} \overline{u_i} + \overline{f_i} \\ \Leftrightarrow \rho\left(\partial_t \overline{u_i} + \partial_j\left(\overline{u_i}\ \overline{u_j}\right)\right) &= -\partial_i \overline{p} + \mu \partial_{j,j} \overline{u_i} + \overline{f_i} - \rho \partial_j\left(\overline{u_i' u_j'}\right) \end{aligned} \tag{1.14}$$

$$\begin{aligned} \partial_t \overline{\Gamma} + \partial_j\left(\overline{\Gamma u_j}\right) &= A \partial_{j,j} \overline{\Gamma} \\ \Leftrightarrow \partial_t \overline{\Gamma} + \partial_j\left(\overline{\Gamma}\ \overline{u_j}\right) &= A \partial_{j,j} \overline{\Gamma} - \partial_j\left(\overline{\Gamma' u_j'}\right) \end{aligned} \tag{1.15}$$

When the flow is stationary, temporal derivatives of averaged quantities are zero. Using an ensemble average instead of temporal average, one can find similar equations while keeping temporal derivatives. The obtained set is called URANS equations (Unsteady Reynolds-Averaged Navier-Stokes). We present in the following a few models used to express the Reynolds stresses.

Linear eddy-viscosity

Those models are the simpler ones. They are commonly used for engineering applications and are not very expensive computationally. The Reynolds stress tensor is decomposed in an isotropic part and a traceless symmetric part proportional to the mean strain rate. See (1.16) where k is the turbulent kinetic energy, which is combined with $\overline{p}$ in a modified pressure.

$$R_{i,j} = \overline{u_i' u_j'} = \frac{2}{3} k \delta_{i,j} - 2\nu_t S_{i,j} \text{ with } k = \frac{1}{2} R_{l,l} \text{ and } S_{i,j} = \frac{1}{2}\left(\partial_j \overline{u_i} + \partial_i \overline{u_j}\right) \tag{1.16}$$

Models differ in the expression of the turbulent eddy viscosity ν_t. From dimensional analysis, one finds ν_t is $m^2.s^{-1}$. Using Kolmogorov's second hypothesis one can express ν_t as a function of the turbulent kinetic energy k and the associated energy dissipation rate ϵ. If one defines ω as the turbulent kinetic energy dissipation rate per unit k ($\omega k = \epsilon$), another expression for ν_t is obtained (1.17).

$$\nu_t \propto \frac{k^2}{\epsilon} = \frac{k}{\omega} \tag{1.17}$$

This leads to the $k-\epsilon$ and $k-\omega$ family of models. The main difficulty is to determine k and ϵ or ω. This is achieved by solving 2 transport equations where some terms have to be modelled. They are omitted here and can be easily found in the vast literature, see [9] for a compilation.

Reynolds stress model

Reynolds stress models do not introduce a turbulent eddy viscosity. Instead, transport equations are solved for every component of the Reynolds stress tensor ($R_{ij} = \overline{u_i' u_j'}$) and for ϵ or ω. However, transport equations for Reynolds stresses are derived from Navier-Stokes equations and involve third order correlations ($\overline{u_i' u_j' u_k'}$), pressure-velocity correlations ($\overline{u_i' \partial_i p'}$) or velocity gradient correlations ($\overline{\partial_k u_i' \partial_k u_j'}$), as expressed equation (1.18). The left hand side is the convected Reynolds stress. The first line in the right hand side is the production rate of Reynolds stress from the mean velocity gradient. The second one is the turbulent diffusion. The third one is the velocity pressure-gradient correlation. The fourth one is the viscous diffusion term and the fifth one is the opposite of the dissipation rate. Some of those quantities are not known *a priori* and have to be modelled. Here again, details are omitted, the literature is vast, interested reader could start with [19] or [9].

$$\begin{aligned}
\partial_t R_{ij} + \overline{u_k} \partial_k R_{ij} &= -R_{ik} \partial_k \overline{u_j} - R_{jk} \partial_k \overline{u_i} \text{ : Production} \\
&- \partial_k \left(\overline{u_i' u_j' u_k'} \right) \text{ : Turbulent diffusion} \\
&- \overline{u_j' \partial_i p'} - \overline{u_i' \partial_j p'} \text{ : Pressure gradient correlation} \\
&+ \frac{1}{Re} \partial_{kk} R_{ij} \text{ : Viscous diffusion} \\
&- \frac{2}{Re} \overline{\partial_k u_i' \partial_k u_j'} \text{ : Dissipation}
\end{aligned} \tag{1.18}$$

In case of non-isothermal flow or in presence of a scalar, the same decomposition is applied. Computing the temperature or the scalar concentration is not straightforward. The budget of the turbulent heat fluxes ($\overline{u_i' T'}$) and for the temperature variance ($\overline{T'^2}$) are expressed equations (1.19) and (1.20). Here again, some terms have to be modelled.

$$\begin{aligned}
\partial_t \overline{u_i' T'} + \overline{u_k} \partial_k \overline{u_i' T'} &= -R_{ik} \partial_k \overline{T} - \overline{u_k' T'} \partial_k \overline{u_i} \text{ : Production} \\
&- \partial_k \left(\overline{u_i' T' u_k'} \right) \text{ : Turbulent diffusion} \\
&- \overline{T' \partial_i p'} \text{ : Pressure gradient correlation} \\
&+ \frac{1}{Re} \partial_{kk} \overline{u_i' T'} \text{ : Viscous diffusion} \\
&+ \frac{1}{Re} \left(\frac{1}{Pr} - 1 \right) \overline{u_i' \partial_{kk} T'} - \frac{2}{Re} \overline{\partial_k u_i' \partial_k T'} \text{ : Dissipation}
\end{aligned} \tag{1.19}$$

$$\begin{aligned}
\partial_t \overline{T'^2} + \overline{u_k} \partial_k \overline{T'^2} &= -2 \overline{u_k' T'} \partial_k \overline{T} \text{ : Production} \\
&- \partial_k \left(\overline{T'^2 u_k'} \right) \text{ : Turbulent diffusion} \\
&+ \frac{1}{RePr} \partial_{kk} \overline{T'^2} \text{ : Molecular diffusion} \\
&- \frac{2}{RePr} \overline{\partial_k T'^2} \text{ : Dissipation}
\end{aligned} \tag{1.20}$$

The budgets of the Reynolds stresses, turbulent heat fluxes and temperature variance

presented here were obtained in the channel flow configuration and are presented in the next chapter.

Wall effect

In addition to the previous considerations, special care should be taken when the flow is inhomogeneous.

For example, in the presence of a wall, when using the $k-\epsilon$ model it is advised to introduce a damping function and modify ν_t according to (1.21). It is similar for LES: a (Edward Reginald) Van Driest damping function is often used to account for the viscous sublayer near the wall. See [33] or [26] p. 434 and 598 for more information.

$$\nu_t \propto f_\mu \frac{k^2}{\epsilon} \text{ with } f_\mu = exp\left(\frac{-125}{50 + Re_L}\right) \text{ and } Re_L = \frac{k^2}{\epsilon\nu} \tag{1.21}$$

Chapter 2

Numerical Tools

Our numerical simulations are based on open source software. Incompact3d is adapted to massive parallel platforms and is used to perform DNS, Quasi-DNS and LES. The ability of DNS to produce results without turbulence modelling can be used to investigate the validity of assumptions often used in CFD. Readers interested in spectral methods could start with the software ChannelFlow (http://channelflow.org/) while the ones interested in finite differences could start with Incompact3d (http://code.google.com/p/incompact3d). Our objective is to perform DNS of conjugate heat transfer with Incompact3d to study turbulence and heat transfer in the vicinity of a wall.

2.1 Numerical tools for DNS

DNS are simulations resolving all the scales of turbulence but very expensive computationally. It is somehow a brute force solution to simulate a flow and most of the fluid-mechanicists would prefer a cheaper solution (e.g. RANS). As stated previously, turbulence modelling introduces approximations, limitations and errors. DNS and experiments are the only tools available to investigate the validity of a model. During early ages of DNS it only drew attention of academic researchers. As the computational power available increased, more and more people were interested. Even companies are getting attracted by DNS as a mean to validate complex models like the ones encountered in multiphase flows (oil drilling) or non-Newtonian fluids (polymers in the plastic industry). It is a powerful workbench to study turbulence.

First DNS dealt with homogeneous isotropic turbulence. One of the first DNS of a three-dimensional wall-bounded turbulent flow was performed by Kim, Moin and Moser in 1987 ([14]). It corresponds to the flow between two infinite plates with a bulk Reynolds number around 3000 corresponding to a Reynolds number based on friction velocity Re_τ around 180. More recently, Schlatter et al. ([27]) simulated a turbulent boundary layer at a Reynolds number available experimentally ($Re_\tau = 1300$), excellent agreement was found. DNS are even used to investigate the limitations of experimental probes and to design new less intrusive ones ([35]).

Numerically, DNS can be performed with any scheme, as long as it is consistent, stable and converges toward the physical solution of Navier-Stokes equations. However, computational power is limited and dedicated numerical methods, more efficient and accurate than their general counterparts are used. Some of them are introduced in the following.

Spectral methods

Spectral methods were the first tools used to perform DNS and are some of the most accurate: their rate of convergence is exponential rather than polynomial. Most of the time, periodic directions are treated with a Fourier transform and variables are defined in the spectral space, not in the physical one. Expressing variables in the physical space implies an inverse Fourier transform and global communications, which can become very time-consuming. If a direction is not periodic, several techniques are available. Most frequently, a Chebyshev pseudospectral approximation is used: variables are approximated using Lagrange interpolation on Chebyshev nodes. It is a convenient technique as Chebyshev interpolation polynoms convergence is both quadratic and uniform: Runge's phenomenom and spurious oscillations at the boundary are avoided. Applying spectral methods to non-linear equations is not-simple. E.g. consider equation (2.1).

$$\cos(k_1 x)\cos(k_2 x) = \frac{1}{2}\left(\cos\left(\left[k_1 + k_2\right]x\right) + \cos\left(\left[k_1 - k_2\right]x\right)\right) \tag{2.1}$$

From equation (2.1), a non-linear term brings harmonics with new frequencies. Recalling Fourier transform of a product $\widehat{fg}$ is a convolution product $\widehat{f} \bigotimes \widehat{g}$, one can deduce that the non-linear convective term in Navier-Stokes equations produces new frequencies. The process goes both ways: large structures degenerate in small ones while small ones may aggregate to form a bigger one (i.e. forward- and back-scatter of energy). Representation of such processes is limited in a simulation: the small structure may go out of the range of resolved scales. It is called aliasing and is a source of error. One solution is the "$\frac{3}{2}$" rule: $M = \frac{3}{2}(N+1) - 1$ nodes are used in the physical space while only N modes are retained for computation in spectral space.

Spectral methods remain extremely accurate and fitted for academic simulations. It is somehow the less flexible method and situations with complex geometry or obstacles are tough to handle. Finite differences are introduced in the following. While being less accurate than their spectral counterparts, they are more flexible.

Finite difference schemes

The basis of the following analysis (limited to $1D$) is inspired by an article from Lele published in 1991 called "Compact finite difference schemes with spectral-like resolution" [20]. We focus here only one the ones used in Incompact3d. Both collocated and staggered schemes are introduced. The versatility of such schemes is particularly evident for second derivative, see equation (2.2). Second derivatives are used to compute the viscous term. Therefore, there is a strong connection between the numerical scheme used for the second derivative and the energy dissipation during the simulation [18].

$$
\begin{aligned}
\alpha_2 f^{"}_{i-1} + f^{"}_i + \alpha_2 f^{"}_{i+1} &= a_2 \frac{f_{i-1} - 2f_i + f_{i+1}}{h^2} + b_2 \frac{f_{i-2} - 2f_i + f_{i+2}}{4h^2} + c_2 \frac{f_{i-3} - 2f_i + f_{i+3}}{9h^2} \\
a_2 + b_2 + c_2 &= 1 + 2\alpha_2 \text{ (Second order)} \\
a_2 + 2^2 b_2 + 3^2 c_2 &= \alpha_2 \frac{4!}{2!} \text{ (Fourth order)} \\
a_2 + 2^4 b_2 + 3^4 c_2 &= \alpha_2 \frac{6!}{4!} \text{ (Sixth order)} \\
a_2 + 2^6 b_2 + 3^6 c_2 &= \alpha_2 \frac{8!}{6!} \text{ (Eighth order)} \\
k^{"} &= \frac{2a_2(1-\cos(k)) + \frac{b_2}{2}(1-\cos(2k)) + \frac{2c_2}{9}(1-\cos(3k))}{1 + 2\alpha_2 \cos(k)}
\end{aligned} \tag{2.2}
$$

The stencil used in scheme (2.2) uses 3 points for the l.h.s. and 7 points for the r.h.s., which will be marked as a $(3;7)$ stencil. Enlarging the stencil on any of the sides reduces the truncation error. The scheme (2.2) can reach eighth order accuracy.

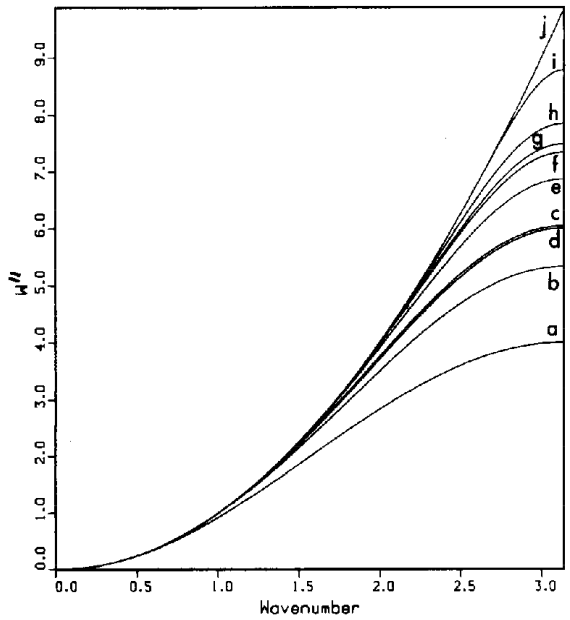

Figure 2.1: Modified wavenumber for second derivative from [20]. a, b, c, ... h denotes schemes described in r.h.s text.

Following Lele ([20]), one can apply Fourier analysis on differencing schemes to measure the error when applied to the harmonic function $f = \cos(kx)$. While the exact second-derivative is $-k^2 f$, the finite-difference scheme produces an approximate solution $-k^{"} f$ where $k^{"}$ is called the modified wave-number. Figure 2.1 from [20] compares the modified wave-number obtained for several schemes alongside with the exact result. Schemes a to h are arranged in an increasing stencil and order: a is 2^{nd} order with a stencil $(1;3)$ while h is 10^{th} order with a stencil $(5;7)$. The quasi-spectral scheme i is only 4^{th} order with a stencil $(5;7)$: high-order conditions on the scheme's coefficients are discarded, leaving free parameters to optimize the modified wave-number.

Accordingly, Lamballais et al. in [18] proposed to relax the objective of very-high order accuracy in favour of modified wave-number optimization. It is the strategy implemented in Incompact3d: the numerical scheme used to compute the viscous term allows one to over or under dissipate small scales. This is achieved thanks to a singularity in the modified wave-number function at high-frequencies ($k \approx \pi$) when $\alpha_2 \approx \frac{1}{2}$.

$$
\begin{aligned}
\alpha_1 f'_{i-1} + f'_i + \alpha_1 f'_{i+1} &= a_1 \frac{f_{i+1} - f_{i-1}}{2h} + b_1 \frac{f_{i+2} - f_{i-2}}{4h} + c_1 \frac{f_{i+3} - f_{i-3}}{6h} \\
\alpha_{I1} f^{I'}_{i-0.5} + f^{I'}_{i+0.5} + \alpha_{I1} f^{I'}_{i+1.5} &= a_{I1} \frac{f_{i+1} - f_i}{h} + b_{I1} \frac{f_{i+2} - f_{i-1}}{3h} + c_{I1} \frac{f_{i+3} - f_{i-2}}{5h} \\
\alpha_I f^I_{i-0.5} + f^I_{i+0.5} + \alpha_I f^I_{i+1.5} &= a_I \frac{f_{i+1} + f_i}{2} + b_I \frac{f_{i+2} + f_{i-1}}{2} + c_I \frac{f_{i+3} + f_{i-2}}{2}
\end{aligned} \tag{2.3}
$$

Schemes (2.3) illustrate derivation or interpolation on a collocated and staggered grid. The last one, interpolation on a staggered-grid is of particular interest, see equation (2.4) and figure 2.2.

$$\begin{aligned}
\alpha_I f^I_{i-0.5} + f^I_{i+0.5} + \alpha_I f^I_{i+1.5} &= a_I \frac{f_{i+1}+f_i}{2} + b_I \frac{f_{i+2}+f_{i-1}}{2} + c_I \frac{f_{i+3}+f_{i-2}}{2} \\
a_I + b_I + c_I &= 1 + 2\alpha_I \text{ (Second order)} \\
a_I + 3^2 b_I + 5^2 c_I &= 2^3 \alpha_I \text{ (Fourth order)} \\
a_I + 3^4 b_I + 5^4 c_I &= 2^5 \alpha_I \text{ (Sixth order)} \\
a_I + 3^6 b_I + 5^6 c_I &= 2^7 \alpha_I \text{ (Eighth order)} \\
T^I(k) &= \frac{a_I \cos(\frac{k}{2}) + b_I \cos(\frac{3k}{2}) + c_I \cos(\frac{5k}{2})}{1 + 2\alpha_I \cos(k)}
\end{aligned} \tag{2.4}$$

The transfer function $T^I(k)$ is the error produced by the scheme for the harmonic function $f = \cos(kx)$. The highest resolvable frequency is $k_c = \pi$ (zig-zag pattern), for which every term on the r.h.s. is zero as $f_{i+1} + f_i$ is null.

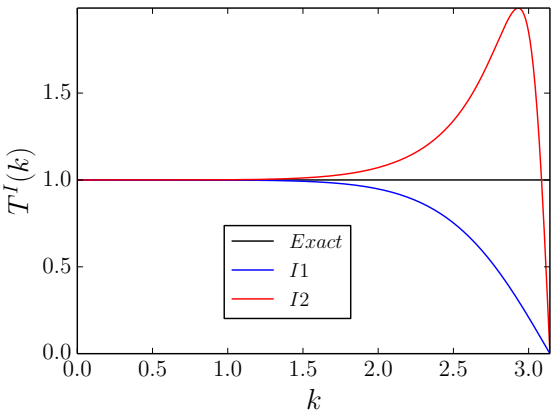

Figure 2.2: Staggered-interpolation

Apart from this special frequency, it is possible to apply the optimization mentioned previously. Figure 2.2 compares 2 interpolators, both are 6^{th} order. $I1$ is the one verifying $c_I = 0$ while $I2$ is the one verifying $\alpha_I = 0.49$. $I2$ interpolates at 6^{th} order but does not verify a min-max principle and amplifies some frequencies. After some testing, we decided to avoid any interpolator with a transfer function below 0 or above 1.

Increasing the stencil allows to maintain the order with more parameters available to optimize the scheme. The larger the stencil, the higher the accuracy. However, large stencils are hardly tractable at the boundary where 2 strategies can be used. If the parity of the function is known, ghost boundary points can be used. If there is no such information, skewed forward or backward schemes are necessary. They are more generic but create an additional numerical stress at boundary nodes compared to interior ones.

Finite differences are particularly well suited for Cartesian domains. It is possible to refine some blocks of the domain. It is also possible to use curvilinear grids. In such cases, a metric matrix connects the numerical directions (i, j, k) with the physical ones (x, y, z). Curvilinear grids are often used in presence of obstacles or for axisymmetric flows. However, the more complex the metric, the more complex the numerical implementation, see [28] for example. When using a rectilinear Cartesian grid, one can alternatively model obstacles using the Immersed Boundary Method. The literature on the topic is quite vast, interested readers may start with [24].

2.2 Incompact3d

Incompact3d is dedicated to DNS and LES of incompressible flows. It has been developed at *Université de Poitiers* and at *Imperial College London* over last 10 years. Turbulent flows are computed using high-order finite-difference compact schemes in the physical space and a spectral Poisson solver for the pressure. Thanks to parallel Fourier transforms and an efficient 2D domain decomposition, the code is known to scale well on thousands of processors (see [17]). Moreover, it can cope with scalars (both active and passive) and obstacles can be included using the immersed boundary method (see [16]). It is a powerful tool, briefly reviewed here. For more information, see Incompact3d website: http://code.google.com/p/incompact3d.

Numerical grid and domain decomposition

The physical domain is decomposed on a Cartesian grid. One direction can be stretched to give a refined grid at the centre of the domain, near one edge or near both edges. We assume $0 \leq y \leq L_y$, $0 \leq s \leq 1$ and define the metric h ($y = h(s)$). According to equation (2.5), a derivative in physical space is a classic product while a simple product in physical space is a convolution integral in spectral space. Here, $\widehat{f}$ is the Fourier transform of f and $f \bigotimes g$ the convolution product between f and g.

$$\begin{aligned} \frac{\partial f}{\partial y} &= \frac{ds}{dy}\frac{\partial f}{\partial s} \\ \Leftrightarrow \frac{\partial f}{\partial y} &= \frac{1}{h'}\frac{\partial f}{\partial s} \\ \Leftrightarrow \widehat{\frac{\partial f}{\partial y}} &= \widehat{\frac{1}{h'}} \bigotimes \widehat{\frac{\partial f}{\partial s}} \end{aligned} \tag{2.5}$$

For operations to be equivalent in physical and spectral space, one must avoid aliasing errors. Choosing $\frac{1}{h'}$ to depend only on few Fourier modes will discard aliasing errors from this convolution product and ensure faster computations in the spectral space. Accordingly, the metric used depends on 3 Fourier modes and is expressed equation (2.6) (here H is the Heaviside step function). Parameters α, β, γ and δ can be modified to have a refined mesh at the centre, at one boundary or at both. See [16] for more information.

$$\begin{aligned} \frac{1}{h'} &= \frac{1}{L_y}\left(\frac{\alpha}{\pi} + \frac{1}{\pi\beta} sin^2\left(\gamma s + \delta\right)\right) \\ h\left(s\right)\frac{\gamma\sqrt{\alpha}\sqrt{\alpha\beta+1}}{L_y\sqrt{\beta}} &= \tan^{-1}\left[\frac{\sqrt{\alpha\beta+1}\tan\left(\pi\left(\gamma s+\delta\right)\right)}{\sqrt{\alpha}\sqrt{\beta}}\right] \\ &+ \pi\left[H\left(s-\frac{1-2\delta}{2\delta}\right)+H\left(s-\frac{3-2\delta}{2\delta}\right)\right] \\ &- \tan^{-1}\left[\frac{\sqrt{\alpha\beta+1}\tan\left(\pi\delta\right)}{\sqrt{\alpha}\sqrt{\beta}}\right] \end{aligned} \tag{2.6}$$

Domain decomposition is a main issue affecting scalability of the code. Incompact3d uses the library 2decomp developed by NAG (see [21]). It decomposes the domain using 2d

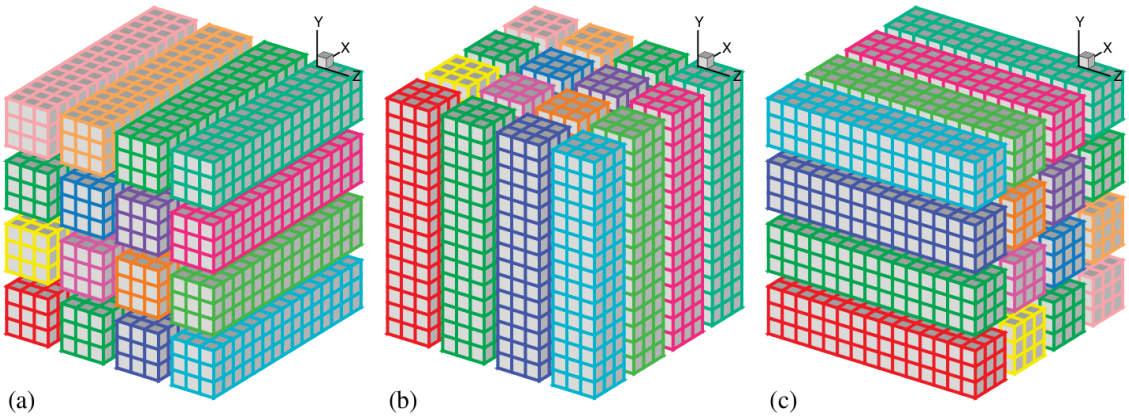

Figure 2.3: Domain decomposition. (a) x-pencil, (b) y-pencil, (c) z-pencil. [17]

	α	a	b	c
Collocated, ∂^2	$\frac{45\times4\pi^2-272}{90\times4\pi^2-416}$	$\frac{-3+24\alpha}{5}$	$\frac{6-9\alpha}{4}$	$\frac{2-11\alpha}{20}$
Collocated, ∂^1	$\frac{1}{3}$	$\frac{14}{9}$	$\frac{1}{9}$	0
Staggered, ∂^1	$\frac{9}{62}$	$\frac{63}{62}$	$\frac{17}{62}$	0
Staggered, interpolation	$\frac{3}{10}$	$\frac{75+70\alpha}{64}$	$\frac{-25+126\alpha}{128}$	$\frac{3-10\alpha}{128}$

Table 2.1: Numerical schemes for interior nodes

pencils. It has a good scalability and allows the code to run on $O\left(10^5\right)$ processors (see [17]). Figure 2.3 illustrates the domain decomposition and the 3 pencil configurations when 16 processors, using a mapping 4×4 are used. The number of cells in each direction constrains the mapping and the number of processor used. Interpolation or derivation in z-direction requires the data to be distributed over the processors in the z-pencil configuration. MPI communication allows shifting from a configuration to the other.

Compact finite difference schemes implemented

Compact finite difference schemes were introduced previously. Using the notation of equations (2.2) and (2.3), the implemented schemes are detailed table 2.1 for interior nodes, followed by a discussion around boundary nodes. Reader not interested in details of numerical methods may skip the section.

From table 2.1, one can see the compact coefficient α for the second derivative on a collocated grid is expressed as a function of $4\pi^2$. It corresponds to the value of the modified wave-number $k^{"}$ at the cut-off $k_c = \pi$. It is 4 times higher compared to the exact wave-number. It produces higher dissipation at small scales and is especially helpful for coarsely resolved simulations (see [18]).

If the boundary condition is periodicity or symmetry, the strategy chosen in Incompact3d is to use ghost points at the boundary. In case of periodicity, the value for ghost points at $x = 0$ is obtained from the nodes at $x = L_x$. In case of symmetry, the parity is variable-dependent: the normal velocity will be odd and the tangent one even. If a scalar variable is present, its parity is flow-specific.

If a velocity is prescribed at the boundary (no-slip condition for instance) and no parity assumption used, non-centred schemes are used at the boundary. The ones used in Incompact3d are detailed equation (2.7).

$$\begin{aligned}
f_1^{"} + 11 f_2^{"} &= \frac{1}{h^2}\left(13f_1 - 27f_2 + 15f_3 - f_4\right) \\
\frac{1}{10}f_1^{"} + f_2^{"} + \frac{1}{10}f_3^{"} &= \frac{6}{5h^2}\left(f_1 - 2f_2 + f_3\right) \\
\frac{2}{11}f_2^{"} + f_3^{"} + \frac{2}{11}f_4^{"} &= \frac{12}{11h^2}\left(f_2 - 2f_3 + f_4\right) + \frac{3}{11\left(2h\right)^2}\left(f_1 - 2f_3 + f_5\right) \\
f'_1 + 2f'_2 &= \frac{1}{h}\left(-\frac{5}{2}f_1 + 2f_2 + \frac{1}{2}f_3\right) \\
\frac{1}{4}f'_1 + f'_2 + \frac{1}{4}f'_3 &= \frac{3}{2}\frac{f_3 - f_1}{2h}
\end{aligned} \tag{2.7}$$

The boundary scheme at node $i = 1$ for the both derivatives is 3^{rd} order while it is 4^{th} order at node $i = 2$. Last but not least, at node $i = 3$, the second derivative is 6^{th} order accurate. One may notice the absence of staggered derivation or interpolation. Computations from the collocated grid (velocity) to the staggered one (pressure) are always performed with periodicity or parity assumptions (e.g. ghost points). It is coherent with the cosine expansion performed on the pressure when using symmetry or a prescribed velocity at the boundary. This treatment is second-order accurate in case of a no-slip condition and may deteriorate the rate of convergence in some cases, see [16] section 4.4 for more information.

Time stepping

Time advancement in Incompact3d can be performed with Adams-Bashforth (AB) or Runge-Kutta (RK) methods. Implemented AB schemes are 2^{nd} and 3^{rd} order while RK ones are 3^{rd} and 4^{th} order. Most of our simulations are done with the 2^{nd} order AB scheme. Time stepping is done with a prediction-correction algorithm (2.8). During the prediction step, the convective and diffusive terms are used to predict a velocity field u^* based on current and previous velocity fields (u^n, u^{n-1}, u^{n-2}). During the correction step, the pressure field p^{n+1} is computed and used to obtain the final, divergence-free velocity field u^{n+1}.

$$\begin{aligned}
\partial_t u_i + \partial_j\left(u_i u_j\right) &= -\partial_i p + \frac{1}{Re}\partial_{j,j} u_i \\
F_i(u^n) &= -\partial_j\left(u_i^n u_j^n\right) + \frac{1}{Re}\left(\partial_{xx}u_i^n + \partial_{yy}u_i^n + \partial_{zz}u_i^n\right) \\
\frac{u_i^* - u_i^n}{dt} &= \frac{3}{2}F_i(u^n) - \frac{1}{2}F_i(u^{n-1}) - \partial_i p^n \\
\frac{u_i^{**} - u_i^*}{dt} &= \partial_i p^n \\
\frac{u_i^{n+1} - u_i^{**}}{dt} &= -\partial_i p^{n+1} \\
dt\ \partial_{kk} p^{n+1} &= \partial_i u_i^{**}
\end{aligned} \tag{2.8}$$

Spectral Pressure solver

Solving the pressure correction equation of (2.8) in the physical space using compact finite difference schemes and an iterative procedure has a prohibitive computational cost. Moreover, the iterative procedure can converge slowly and brings uncertainty in choosing the target residual level. Solving it in the spectral space is simpler and accurate at machine precision. Applying Fourier transform to the divergence of the pressure-correction equation leads to (2.9).

$$-k^2\widehat{p^{n+1}}\left(k_x,k_y,k_z,t\right)=\widehat{\partial_i u_i^*}\left(k_x,k_y,k_z,t\right) \tag{2.9}$$

This is the analytical solution. However, the Laplacian in the physical place is different from the exact analytical Laplacian. Operations performed in spectral domain have to be identical to the ones performed in the physical domain. This is simply achieved by replacing k^2 with the modified wave-number $k"$ in (2.9). Pressure in physical space is then obtained with a simple division by $k"$ (inversion of a diagonal matrix). If the grid is stretched in one direction, the matrix is no longer diagonal. However, the metric was chosen so that the matrix has a sparse and simple band structure. Even if it makes solving the pressure equation three times longer, the total time used to compute the pressure does not exceed 30% of the computation time (iterative pressure solvers may consume 70% of the computational time).

2.3 Developments performed

Several numerical developments were undertaken. Some of them were mandatory (dedicated routines to compute high-order statistics, module for conjugate heat-transfer). Others improved the efficiency and the stability of the code.

Semi-implicitation of the viscous term

Thibault Dairay from Université de Poitiers, currently working with Incompact3d reported that using a semi-implicitation of the wall-normal viscous term $\partial_{yy}u$ enhances performance: time-step can be 3 times higher compared with fully explicit formulation. His developments concerned both the velocity and the thermal field but were restricted to a specific configuration. They were extended herein to handle generic boundary-condition and a stretched mesh and are detailed below.

$$\begin{aligned}\partial_{yy}f &= \left(\frac{1}{h'}\right)^2\partial_{ss}f-\frac{h"}{h'^2}\partial_s f\\ &= \left(\frac{1}{h'}\right)^2\partial_{ss}f-\frac{h"}{h'}\partial_y f\end{aligned} \tag{2.10}$$

In presence of a stretched mesh, we use a metric h to connect the physical grid $0\leq y\leq L_y$ to a regular homogeneous grid $0\leq s\leq 1$: $y=h(s)$. The viscous term $\partial_{yy}f$ depends on position y. Using the bijective metric h, it is expressed equation (2.10) as a function of variable s .

$$
\begin{aligned}
F_i(u^n) &= -\partial_j\left(u_i^n u_j^n\right) + \frac{1}{Re}\left(\partial_{xx}u_i^n + \partial_{zz}u_i^n - \frac{h"}{h'}\partial_y u_i^n\right) \\
\frac{u_i^* - u_i^n}{dt} &= \frac{\partial_{ss}u_i^* + \partial_{ss}u_i^n}{2Re\ h'^2} + \frac{3}{2}F_i(u^n) - \frac{1}{2}F_i(u^{n-1}) - \partial_i p^n \\
\left(1 - \frac{dt}{2Re\ h'^2}\partial_{ss}\right)u_i^* &= G_i(u^n, u^{n-1})
\end{aligned} \tag{2.11}
$$

From decomposition (2.10), the term $\partial_{ss}f$ is integrated with a Crank-Nicolson scheme while the other term is integrated with a AB scheme. The global time-integration scheme when using a 2^{nd} order AB is detailed equation (2.11).

$$
\begin{aligned}
\mathbf{A}\partial_{ss}f &= \mathbf{B}f \\
\partial_{ss}f &= \mathbf{A}^{-1}\mathbf{B}f
\end{aligned} \tag{2.12}
$$

The derivative $\partial_{ss}f$ is estimated with a compact finite difference scheme according to (2.2). It is expressed in a matrix form equation (2.12).

$$
\begin{aligned}
\left(\mathbf{A}h'^2 - \frac{dt}{2Re}\mathbf{B}\right)u_i^* &= \mathbf{A}h'^2 G_i(u^n, u^{n-1}) \\
\mathbf{M}u_i^* &= \mathbf{b}
\end{aligned} \tag{2.13}
$$

The metric ($y = h(s)$) used and the related derivatives depends on the s coordinate and does not commute with the matrix $\mathbf{A}$ or $\mathbf{B}$. The final expression for the time-integration is obtained equation (2.13). The right hand side is computed at each time-step. The left hand side matrix has a simple banded structure. The LU decomposition of the matrix is easily computed and stored at the beginning of the simulation and used every time step to compute u^*. The resulting algorithm is quite economical.

In case of no-slip condition, matrices $\mathbf{A}$ and $\mathbf{B}$ reproduces exactly the non-centred schemes at boundary nodes $i = 2$ and $i = 3$. The differencing scheme used at node $i = 1$ is replaced with a Dirichlet boundary condition. If the boundary condition is a symmetry, the left-hand side matrix depends on the velocity component: normal velocity does not behave like tangent one. In case of periodicity, the band structure of the matrix changes, it becomes cyclic. Dedicated routines were developed to handle those operations for every boundary condition as well as computation of $\mathbf{M}^{-1}\mathbf{b}$ without computation of $\mathbf{M}^{-1}$. Those developments were extended to handle scalar variables with a generic Robin type boundary condition. Thanks to a constant left hand side, the semi-implicitation is not computationally expensive and allows simulations with a time step 3 times larger.

Thermal conduction in a solid

The solver dedicated to the solid thermal field and developed in this project is presented here. The fluid domain is discretized on a Cartesian grid: $(0, 0, 0) \leq (x, y, z) \leq (L_x, L_y, L_z)$. It is similar for the solid domain except located on top and below the fluid domain: $-L_{s,1} \leq y \leq 0$ and $L_y \leq y \leq L_y + L_{s,2}$, see figure 2.4. Therefore, they share the

same homogeneous Cartesian grid in the x and z direction. The linear diffusive equation ruling solid temperature evolution is solved with a mixture of finite differences (x and z) and spectral methods (y).

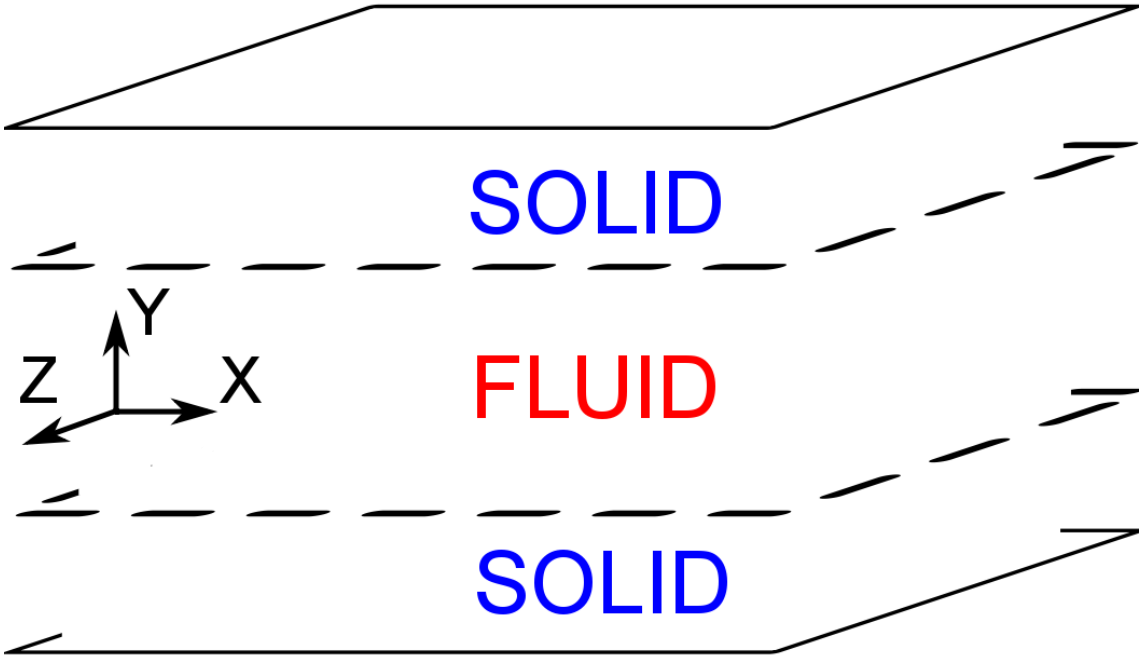

Figure 2.4: Sketch of the fluid and solid domain

$$\begin{aligned} \rho C_p \partial_t T_s &= \kappa \nabla_y^2 T_s + \kappa \nabla_{xz}^2 T_s \\ \alpha_a T_s(y=a) + \beta_a \partial_y T_s(y=a) &= g_a \\ \alpha_b T_s(y=b) + \beta_b \partial_y T_s(y=b) &= g_b \end{aligned} \tag{2.14}$$

Assuming homogeneous and constant solid properties, one finds the temperature is defined by equation (2.14) where the diffusive term is split in a wall-normal part and a wall-parallel one. The parameters (a, b) are used as general boundaries for $(-L_{s,1}, 0)$ or $(L_y, L_y + L_{s,2})$. The solid domain is decomposed in 2d pencils compatible with the fluid domain decomposition.

$$\frac{T_s^{n+1} - T_s^n}{dt} = \frac{\kappa}{\rho c_p} \left(\gamma \partial_{yy} T_s^{n+1} + (1-\gamma) \partial_{yy} T_s^n + \frac{3}{2} \Delta_{xz} T_s^n - \frac{1}{2} \Delta_{xz} T_s^{n-1} \right) \tag{2.15}$$

Wall-parallel diffusion is computed with non-compact finite differences to avoid global communications bottleneck. We use semi-implicitation for the wall-normal diffusive term, leading to equation (2.15). When γ is 0, we are fully explicit, 1 is fully implicit and $\frac{1}{2}$ is Crank-Nicolson.

$$y_i = \frac{a+b}{2} + \frac{b-a}{2}\cos\left(\frac{2(k-i)+1}{2k+2}\pi\right), 0 \leq i \leq k \tag{2.16}$$

The wall-normal direction is discretized with $k+1$ Chebyshev nodes (y_i). The general formula to compute the nodes in $[a, b]$ is (2.16).

$$\begin{aligned} l_i(y) &= \prod_{j=0, j\neq i}^{k} \frac{y - y_j}{y_i - y_j} \\ T_s(y) &= \sum_{i=0}^{k} T_s(y_i) l_i(y) \\ \partial_y T_s(y) &= \sum_{i=0}^{k} T_s(y_i) l_i'(y) \\ \partial_{yy} T_s(y) &= \sum_{i=0}^{k} T_s(y_i) l_i"(y) \end{aligned} \tag{2.17}$$

The temperature in the solid domain is obtained using Lagrange polynomial interpolation on the Chebyshev nodes (2.17). The main advantage of using Chebyshev nodes for the interpolation is the fast convergence in the uniform norm and the avoidance of any spurious oscillation at the boundaries (Runge phenomenon).

$$\begin{aligned} \mathbf{A}_{i,j} T_s^{n+1}(y_j) &= b_s(y_i),\ 0 \leq i \leq k \\ \alpha_a T_s(y_i) l_i(a) + \beta_a T_s(y_i) l_i'(a) &= g_a \\ \alpha_b T_s(y_i) l_i(b) + \beta_b T_s(y_i) l_i'(b) &= g_b \end{aligned} \tag{2.18}$$

The discretized set of equations is synthesised equation (2.18) where it is assumed $\mathbf{A}_{i,j} = \delta_{i,j} - dt\frac{\gamma\kappa}{\rho C_p} l_j"(y_i)$. While the number of unknown $T_s^{n+1}(y_i)$ is $k+1$, the number of equations is $k+3$. Using the boundary conditions, $T_s(y_0)$ and $T_s(y_k)$ can be expressed as functions of g_a, g_b and $T_s(y_i)$, see equation (2.19).

$$\begin{aligned} \begin{pmatrix} \alpha_a l_0(a) + \beta_a l_0'(a) & \alpha_a l_k(a) + \beta_a l_k'(a) \\ \alpha_b l_0(b) + \beta_b l_0'(b) & \alpha_b l_k(b) + \beta_b l_k'(b) \end{pmatrix} \begin{pmatrix} T_s(y_0) \\ T_s(y_k) \end{pmatrix} &= -\sum_{i=1}^{k-1} \begin{pmatrix} \alpha_a l_i(a) + \beta_a l_i'(a) \\ \alpha_b l_i(b) + \beta_b l_i'(b) \end{pmatrix} T_s(y_i) \\ &+ \begin{pmatrix} g_a \\ g_b \end{pmatrix} \end{aligned} \tag{2.19}$$

The matrix on the left hand side can be inverted for most of the boundary conditions and in particular when using Dirichlet or Neumann boundary condition. Assuming $\widetilde{T_s} = (T_s(y_1), ..., T_s(y_{k-1}))$ one can formulate (2.19) more concisely, see equation (2.20).

$$\begin{pmatrix} T_s(y_0) \\ T_s(y_k) \end{pmatrix} = \mathbf{H} \begin{pmatrix} g_a \\ g_b \end{pmatrix} + \mathbf{G}\widetilde{T_s} \tag{2.20}$$

To conclude, we modify (2.18): the dynamic for $i = 0$ and $i = k$ is replaced with the

boundary condition and we inject (2.20) in the dynamic for $1 \leq i \leq k-1$. The final set of equation is (2.21).

$$\begin{aligned}\left(\mathbf{A}_{i,j} + \begin{pmatrix}\mathbf{A}_{i,0} & \mathbf{A}_{i,k}\end{pmatrix}\mathbf{G}\right)\widetilde{T_s}^{n+1}(y_j) &= b_s(y_i) - \begin{pmatrix}\mathbf{A}_{i,0} & \mathbf{A}_{i,k}\end{pmatrix}\mathbf{H}\begin{pmatrix}g_a\\ g_b\end{pmatrix},\ 1 \leq (i,j) \leq k-1\\ \begin{pmatrix}T_s(y_0)\\ T_s(y_k)\end{pmatrix} &= \mathbf{H}\begin{pmatrix}g_a\\ g_b\end{pmatrix} + \mathbf{G}\widetilde{T_s} \end{aligned} \tag{2.21}$$

The matrix on left hand side of equation (2.21) does not vary during the simulation. Therefore, the inverse matrix is computed only once as the programme starts.

Conjugate heat-transfer

Conjugate heat-transfer is somehow similar to fluid-structure interaction: boundary conditions introduce a coupling between the fluid and the solid. Thermodynamic equilibrium implies continuity of temperature and heat flux at the fluid-solid interface. Two new dimensionless numbers: ratio of thermal diffusivity G and the ratio of thermal conductivity $\alpha = \frac{\lambda_f}{\lambda_s}$ are mandatory for conjugate heat transfer. Some authors express the ratio of thermal conductivity as a combination of G and the thermal activity ratio (which is the ratio of thermal effusivity...). The fact is that we need 2 additional dimensionless numbers to characterize the coupling (2.22).

$$\begin{aligned}\partial_t T_f + \partial_j\left(T_f u_j\right) &= \frac{1}{RePr}\partial_{j,j}T_f \text{ in } \Omega_f\\ \partial_t T_s &= \frac{1}{GRePr}\Delta T_s \text{ in } \Omega_s\\ T_s &= T_f \text{ in } \partial\Omega_f \cap \partial\Omega_s\\ \kappa_s\partial_n T_s &= \kappa_f\partial_n T_f \text{ in } \partial\Omega_f \cap \partial\Omega_s\end{aligned} \tag{2.22}$$

At time step n, both T_f^n and T_s^n are available. First, fluid temperature is updated with a Dirichlet condition and a θ-scheme: $T_f^{n+1} = \theta T_f^n + (1-\theta)T_s^n$ at $\partial\Omega_f \cap \partial\Omega_s$. Then, the solid temperature is updated with a Neumann condition: $\kappa_s\partial_n T_s^{n+1} = \kappa_f\partial_n T_f^{n+1}$ at $\partial\Omega_f \cap \partial\Omega_s$. Keeping the resolution simple led to this solution as solving the full coupled system is costly and difficult: solid temperature uses a spectral algorithm while fluid temperature uses finite differences. As a consequence of this strategy, the final heat flux is continuous at the interface while the temperature may not. The temperature gap at the interface is expected to be first order in time

The semi-implicitation allows simulations to be performed faster and can handle general boundary conditions. It is complementary with the thermal solid solver. Both developments were performed with a goal of simplicity and efficiency. Their convergence was tested against analytical solutions and their results compared with academic ones. Some of the tests performed are detailed below.

Chapter 3

Validation cases

This chapter presents test cases used to validate modifications in the code and post-processing that we have introduced.

3.1 Validation against analytical solutions

Some analytical solutions are available and stable for some Reynolds number. They were used to check the consistency and convergence of Incompact3d. Many more analytical solutions are available for the stand-alone solid thermal module, a few of them were also tested.

3.1.1 Fluid domain

The fluid and thermal solver used in the fluid domain were tested in different configurations. Convergence is measured with the quadratic norm L_2 and the uniform or infinite norm L_∞

Taylor-Green

The Taylor-Green solution allows checking the correct implementation of periodic and symmetric (free-slip) boundary conditions.

In a domain of size $(L_x, L_y, L_z) = (1, 1, 1)$, the analytical solution of 2d Taylor-Green vortices is detailed equation (3.1). In this case, the time-derivative compensates exactly the diffusion, for both velocity and temperature. Our test was done with $k = 2\pi$, $Re = 10^3$, $Sc = 0.2$, $dt = 10^{-4}$. The error was measured after 10^3 time steps: $T = 0.1$ at the end of the simulation. Time advancement was computed with a AB2 scheme. The streamlines are visible figure 3.1.

$$\begin{aligned} U_x(x,y,t) &= \exp^{-2k^2 \frac{1}{Re} t} \sin(kx)\cos(ky) \\ U_y(x,y,t) &= -\exp^{-2k^2 \frac{1}{Re} t} \cos(kx)\sin(ky) \\ T_f(x,y,t) &= \exp^{-2k^2 \frac{1}{RePr} t} \sin(kx)\sin(ky) \end{aligned} \tag{3.1}$$

Convergence in figures 3.1 and 3.2 is limited by the error related to the time scheme. Each time step, the 2^{nd}-order AB scheme produces an error $\propto dt^2$. The number of time

steps computed is $\propto \frac{T_f}{dt}$, leading to a final error $\propto dt$. This error of magnitude 9.10^{-10} can be observed for $nx = 64$ and 128 on regular grids after 1000 time steps. Our results highlight the error connected to the time-discretization while demonstrating order 6 convergence in space and no regression when using the semi-implicitation. Better convergence is obtained with a 3^{rd}-order AB scheme or a RK scheme.

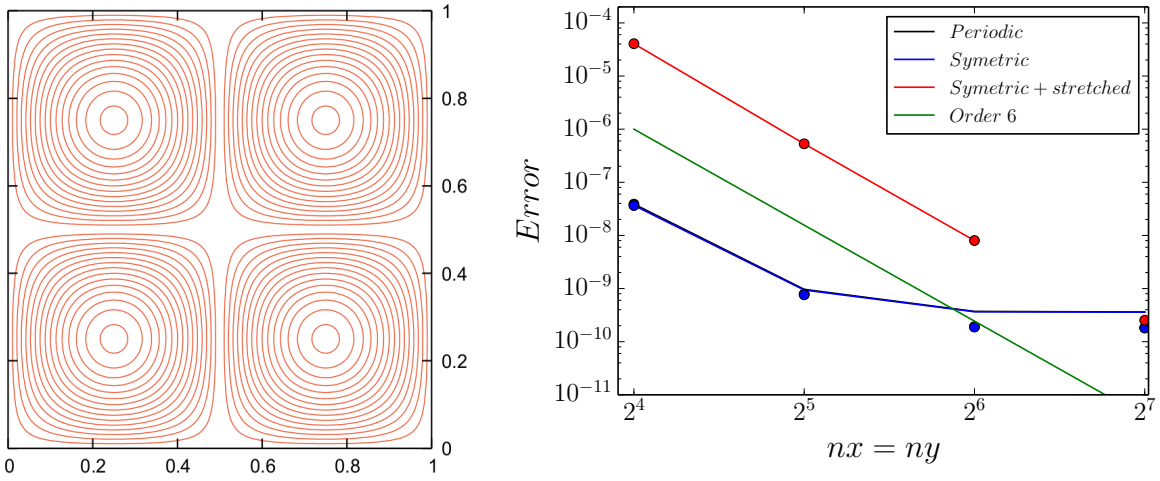

Figure 3.1: Left: Streamlines for Taylor-Green [16]. Right: Taylor-Green convergence for scalar. Lines: explicit. Symbols: implicit.

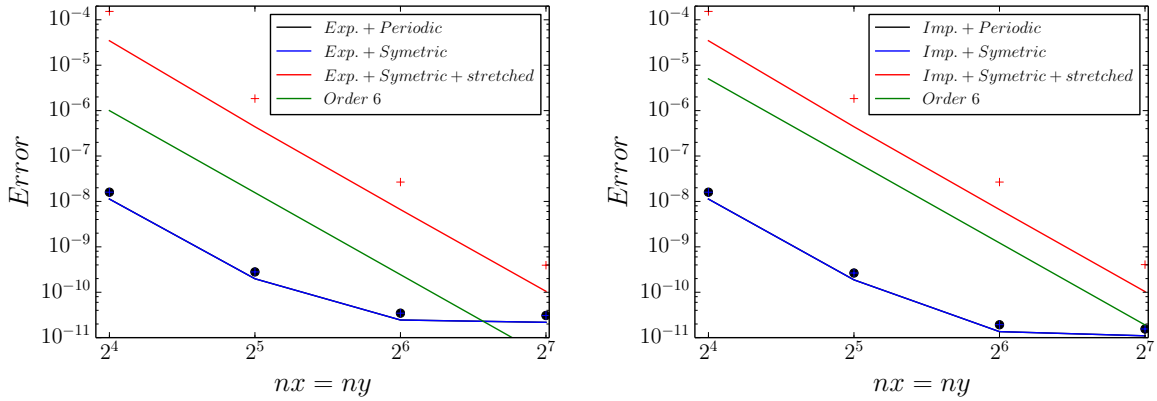

Figure 3.2: Taylor-Green convergence for velocity. Left: explicit. Right: implicit. Lines: L^2. Symbols: L^∞.

Identical results for periodic or symmetric boundary conditions and for both explicit and semi-implicit is a good point. Results on a stretched grid are not optimal and bring larger error when compared to a regular grid. However, order of convergence is not impacted, highlighting the compatibility of the semi-implicitation with the stretched grid. To conclude, the convergence for temperature figure 3.1 suggests slightly better results for the semi-implicit formulation when compared with the fully explicit one.

Burgraff flow

The Burgraff solution allows to check the correct implementation of no-slip boundary-condition. The flow is two-dimensional and stationary. While source terms are quite complex, the solution is simply a polynom (cf [16]), see equation (3.2). The global system

is detailed in equation (3.3).

$$\begin{aligned} U_x(x,y,t) &= 8\left(x^4-2x^3+x^2\right)\left(4y^3-2y\right) \\ U_y(x,y,t) &= -8\left(4x^3-6x^2+2x\right)\left(y^4-y^2\right) \end{aligned} \tag{3.2}$$

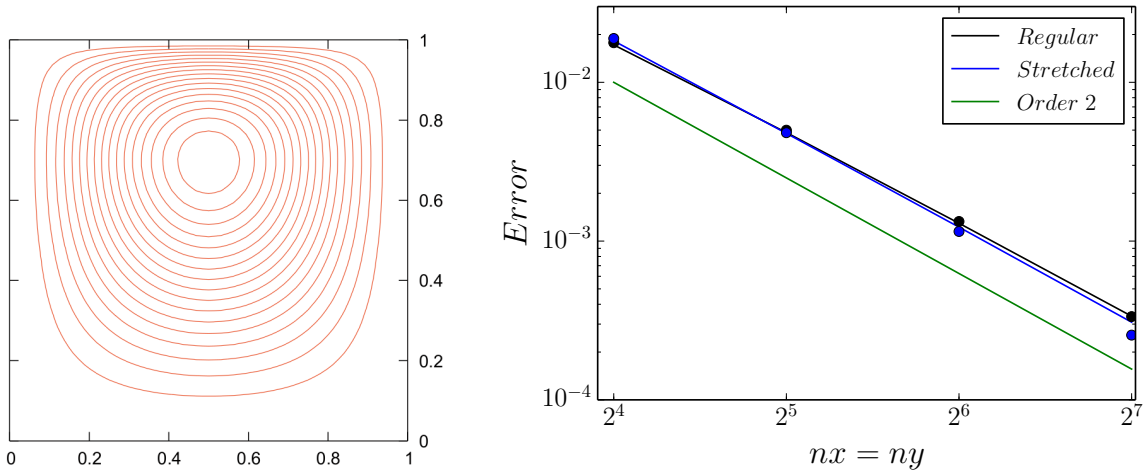

Figure 3.3: Left: Streamlines for the Burgraff flow [16]. Right: Convergence.

The test was done with $Re = 10$ and $dt = 10^{-4}$. The error was measured after 10^2 time steps: $T = 10^{-2}$ at the end of the simulation. Time advancement was computed with a AB2 scheme. According to figure 3.3, results are quasi-identical for explicit and semi-implicit time-integration. The convergence is 2^{nd} order because of the pressure cosine expansion with no-slip condition. Here again, no regression can be observed when using semi-implicitation, which is fully compatible with the stretched grid.

$$\begin{aligned} \partial_t u_i + \partial_j\left(u_i u_j\right) &= -\partial_i p + \frac{1}{Re}\partial_{j,j} u_i + f_i \\ \text{with } f_x &= -\frac{8}{Re}\left(24g + 2g^{"}h^{"} + g^{""}h\right) \\ &+ 64\left(\frac{g^{'2}}{2}\left(hh^{"} - h^{'}h^{"}\right) - hh^{'}\left(g^{'}g^{"} - g^{'2}\right)\right) \\ f_y &= 0 \\ g(x) &= \frac{x^5}{5} - \frac{x^4}{2} + \frac{x^3}{3} \\ h(y) &= y^4 - y^2 \\ U_x(x=0) = U_y(x=0) &= 0 \\ U_x(x=1) = U_y(x=1) &= 0 \\ U_x(y=0) = U_y(y=0) &= 0 \\ U_x(y=1) &= 16x^2\left(x-1\right)^2 \\ U_y(y=1) &= 0 \end{aligned} \tag{3.3}$$

Conduction in Fluid

The temperature solver of the fluid domain was tested on its own to check the correct implementation of Dirichlet and Neumann boundary conditions in case of no-slip condition and semi-implicitation. Therefore, convergence was checked for one-dimensional cases with a convective term forced to zero, see figure 3.4. The test was done with $RePr = 200$, $dt = 10^{-4}$. The error was measured after 10^3 time steps: $T = 0.1$ at the end of the simulation.

When using a Dirichlet boundary condition, the error decreases as dy^6, consistently with the 6-th order compact finite-difference schemes used. For Neumann boundary conditions, the error decreases as dy^4. It is coherent with the approximation used to estimate the derivative at the boundary in the semi-implicit viscous term: the stencil uses 4 points. If the convective term is not nil, the error on the velocity contaminates the temperature. Therefore, in a general configuration involving a no-slip condition, the error decreases as dy^2 as in the Burgraff flow (pressure cosine expansion).

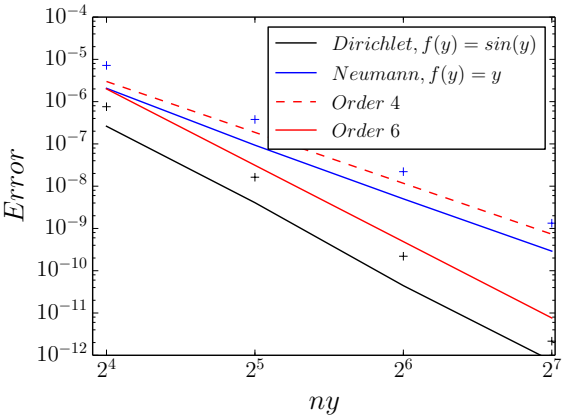

Figure 3.4: Thermal diffusion, fluid domain.

The error related to the time-scheme evocated for Taylor-Green is not relevant here: the only non-zero terms are the time-derivative and the wall-normal diffusive term, none of them being integrated through the AB time-scheme. Although Crank-Nicolson is second order in time, just like the 2^{nd} order AB scheme, one must keep in mind the convergence order is an exponent: for a given time step, there is no reason for a Crank-Nicolson and a 2^{nd} order AB scheme to produce the same absolute error values.

3.1.2 Solid domain

Error in the diffusive term

For simple cases, the analytical solution can be integrated through the time-scheme analytically, see equation (3.4) for example. We used such solutions to measure the spatial error in the diffusive term. From figure 3.5, one can see the exponential convergence of the spectral method used for the wall-normal diffusion: double precision maximal accuracy is quickly reached. Regarding the wall-parallel diffusion, our explicit finite-difference scheme converges like dx^6, in agreement with the 7 points stencil used in x and z direction to estimate the second derivative.

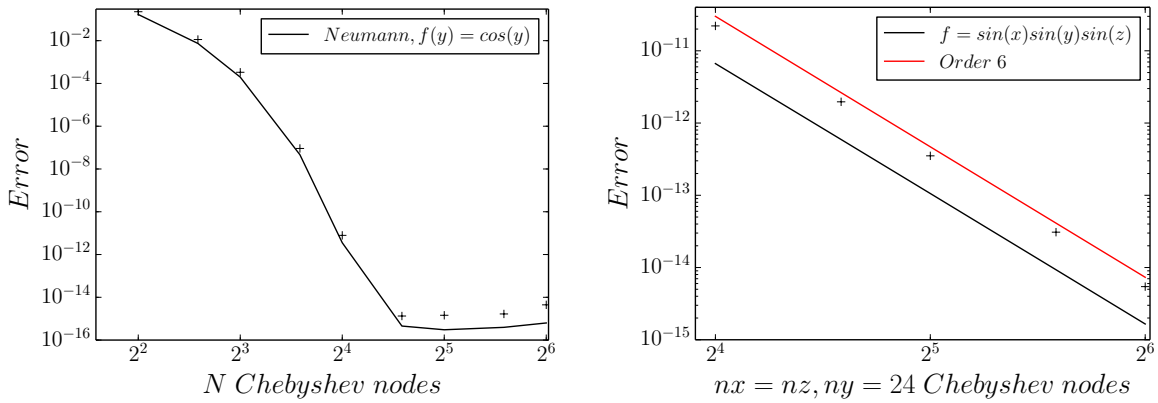

Figure 3.5: Thermal diffusion error in the solid. Left: 1D, right: 3D.

$$\begin{aligned} T_s(t=0) &= \cos\left(2\pi\frac{y}{L}\right) \\ \frac{T_s^{n+1}-T_s^n}{dt} &= \frac{\kappa}{\rho c_p}\left(\gamma\partial_{yy}T_s^{n+1}+(1-\gamma)\partial_{yy}T_s^n\right) \\ T_s(t=dt) &= T_s(t=0)\frac{1-\frac{dt\kappa}{\rho C_p}(1-\gamma)\left(\frac{2\pi}{L}\right)^2}{1+\frac{dt\kappa}{\rho C_p}\gamma\left(\frac{2\pi}{L}\right)^2} \end{aligned} \tag{3.4}$$

Time-varying boundary condition

A particular solution for the temporal diffusion in a solid is exhibited here, it is $1D$ and not stationary. The boundary condition on one side of the solid is adiabatic while the other one is a temperature varying as $\cos(\omega t)$. The analytical solution is exposed equations (3.5) and (3.6) for a unit thermal diffusivity.

From figure 3.6, exponential convergence of the spectral method can still be observed. However, the error quickly reaches an error-bound caused by time-integration. Temperature fluctuations in the solid domain decrease exponentially with $\frac{y}{\delta}$ where $\frac{1}{\delta^2}=\frac{\omega\rho C_p}{\kappa}$

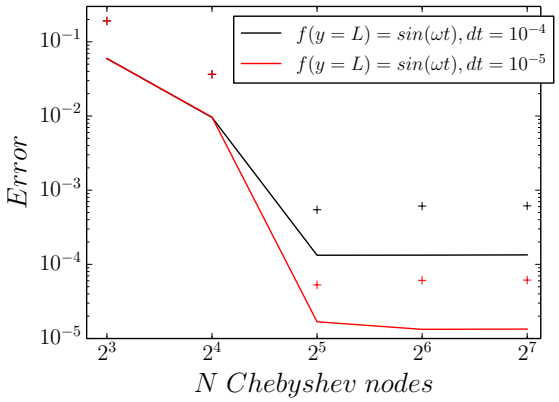

Figure 3.6: Error for a time-varying homogeneous boundary condition. Line: L^2. Symbol: L^∞

$$\begin{aligned} \partial_t T &= \partial_{yy}T \\ T(y,t) &= \left[C_\alpha \sin\left(\sqrt{2\omega}\frac{y}{2}\right)\sinh\left(\sqrt{2\omega}\frac{y}{2}\right) - C_\beta\cos\left(\sqrt{2\omega}\frac{y}{2}\right)\cosh\left(\sqrt{2\omega}\frac{y}{2}\right)\right]\cos(\omega t) \\ \partial_y T(y=0) &= 0 \\ T(y=L,t) &= cos(\omega t) \\ T(t=0) &= C_\alpha \sin\left(\sqrt{2\omega}\frac{y}{2}\right)\sinh\left(\sqrt{2\omega}\frac{y}{2}\right) - C_\beta\cos\left(\sqrt{2\omega}\frac{y}{2}\right)\cosh\left(\sqrt{2\omega}\frac{y}{2}\right) \end{aligned} \tag{3.5}$$

$$
\begin{aligned}
C_\alpha &= \frac{-2\sin\left(\sqrt{2\omega}\frac{L}{2}\right)\left(-1+\exp\left(-\sqrt{2\omega}L\right)\right)\exp\left(-\sqrt{2\omega}\frac{L}{2}\right)}{-2\exp\left(-\sqrt{2\omega}L\right)+4\cos^2\left(\sqrt{2\omega}\frac{L}{2}\right)\exp\left(-\sqrt{2\omega}L\right)+\exp\left(-2\sqrt{2\omega}L\right)+1} \\
C_\beta &= \frac{-2\cos\left(\sqrt{2\omega}\frac{L}{2}\right)\left(1+\exp\left(-\sqrt{2\omega}L\right)\right)\exp\left(-\sqrt{2\omega}\frac{L}{2}\right)}{-2\exp\left(-\sqrt{2\omega}L\right)+4\cos^2\left(\sqrt{2\omega}\frac{L}{2}\right)\exp\left(-\sqrt{2\omega}L\right)+\exp\left(-2\sqrt{2\omega}L\right)+1}
\end{aligned} \tag{3.6}
$$

In case of non-homogeneous boundary condition $T(x, y = L, t) = \cos(\omega t + k_x x)$, temperature fluctuations still decrease exponentially with $\frac{y}{\delta_x}$ where

$$\frac{1}{\delta_x^4} \propto k_x^4 + \left(\frac{\omega \rho C_p}{\kappa}\right)^2$$

From those results, one can conclude that temperature fluctuations at the solid boundary with high temporal or spatial frequency do not penetrate very deeply in the solid. We assume to be running a simulation with a time-step dt and a homogeneous Cartesian grid in x and z. Using $\omega \propto \frac{1}{dt}$ and $k_x \propto \frac{1}{\min(dx,dz)}$, one can estimate the resolved scale in the solid.

3.2 Verification against turbulent academic cases

Validation of thermal activity in Incompact3d was done on several test cases which are variations of the classic channel flow. The first test case is a channel with a temperature increasing linearly at the wall [13]. The second one is a channel with a constant heat flux imposed at the wall [31]. To conclude the section, preliminary results for conjugate heat transfer in a channel flow are presented.

3.2.1 Channel flow with passive scalar

Walls with linearly increasing temperature

As the temperature is linearly increasing, periodicity of the flow is obtained through $T = \phi - ax$ where the real temperature ϕ is not periodic while the modified temperature T is periodic.

$$\partial_t T + \partial_i \left(T u_i\right) = \frac{1}{Re\ Pr}\partial_{i,i} T - a u_x \tag{3.7}$$

The coefficient a corresponds to a source term in the temperature equation (3.7) that compensates exactly heat transfer from the wall. See [13] for more information. The modified temperature satisfies a Dirichlet condition at the boundary with a value of 0 imposed.

Table 3.1 summarizes the main differences between our simulation and the reference one. Our domain is slightly smaller in the streamwise and spanwise direction. Our simulation is also much shorter in time compared to the reference one. The finite duration of the simulation and the finite size of the computational domain produce statistical uncertainty. This uncertainty is higher in our simulation compared with the reference. In our case, the mean output which is the friction or Re_τ is underestimated by 1%. The 1^{st} and 2^{nd}

	Present	Ref. [13]
Domain	$[12.8; 2; 4.26]$	$[5\pi; 2; 2\pi]$
Grid	$[256; 193; 256]$	$[128; 97; 128]$
Re_τ	148.8	150
dy^+ $[min, max]$	$[0.49; 4.8]$	$[0.08; 4.9]$
$[dx^+, dz^+]$	$[7.4; 2.5]$	$[18.4; 7.36]$
dt^+ $(\frac{\nu}{u_\tau^2})$	2.10^{-4}	?
Final time	160	2100

Table 3.1: Simulation's parameters, linearly increasing temperature

moments are of similar ordre of accuracy and differences in graphs are only visible for the second moment budgets.

For both simulations, the Reynolds number based on bulk velocity is 2280 and the Prandtl number is 0.71. Note that $Re_\tau = 150$ is close to the lower limit for a turbulent flow and many 2^{nd} order industrial / commercial codes tend to relaminarise at this Re_τ. Figure 3.7 shows the time and space averaged velocity and modified temperature. For both of them, the relative difference in the middle of the channel is around 1%, within the bound of statistic uncertainty.

Velocity fluctuations figure 3.8 and temperature fluctuations figure 3.9 are also quite good, both in the middle of the channel (left) and at the vicinity of the wall (right).

To conclude, budgets of the fluctuations are available figure 3.10 and 3.11. For each of them, the sum of every term in the budget is theoretically zero. The statistical uncertainty leads the sum to be non null, both for our simulation and for the reference one. In our case, it is 2 times higher than the reference error, which is quite good in view of our more limited statistic sampling in time. Term-by-term error is a little higher. Correlation between pressure and velocity often exhibits the largest difference. This may comes from the prediction-correction scheme and is discussed section 4.2 page 47. Qualitatively, the results remains quite good considering our statistic sampling and the different ordre of finite-differences and spectral schemes.

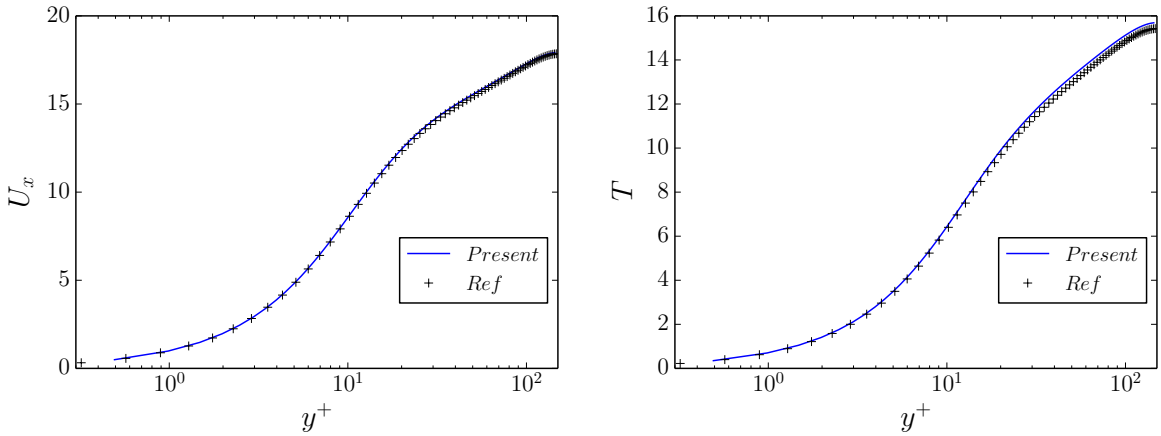

Figure 3.7: Left: Averaged velocity. Right: Averaged temperature, iso-thermal.

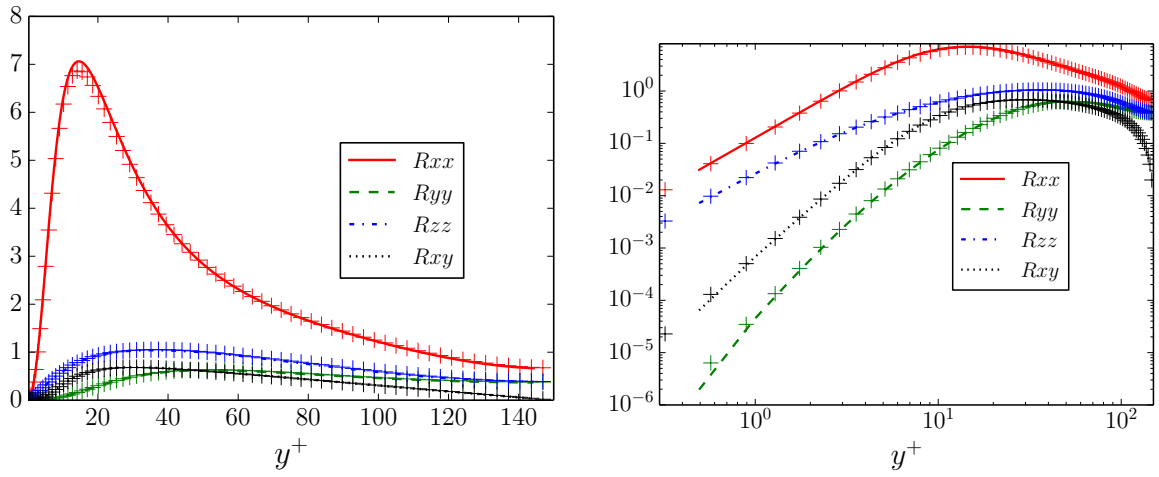

Figure 3.8: Velocity fluctuations. Left: linear scale. Right: log scale. Line: present. Symbol: ref [13]

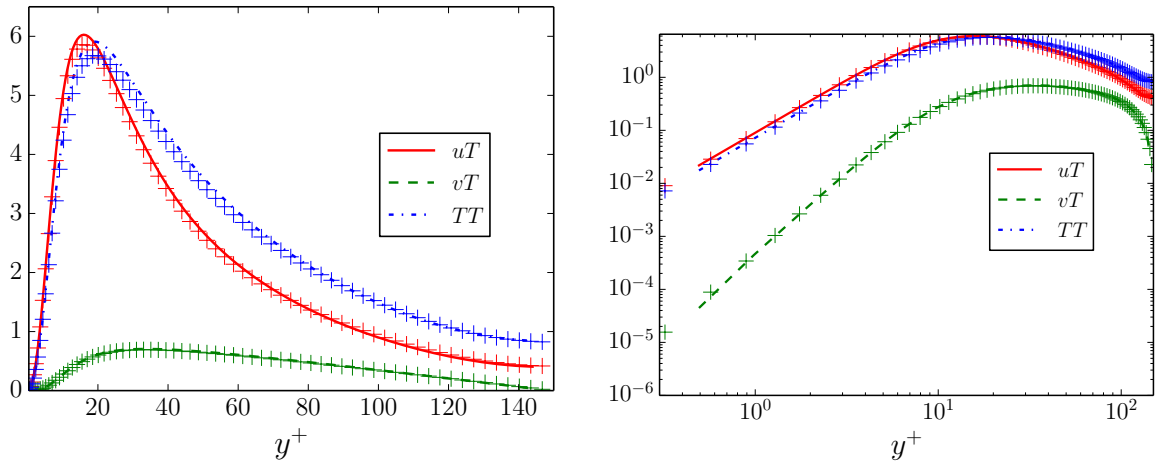

Figure 3.9: Turbulent heat fluxes and variance of temperature. Left: linear scale. Right: log scale. Line: present. Symbol: ref [13]

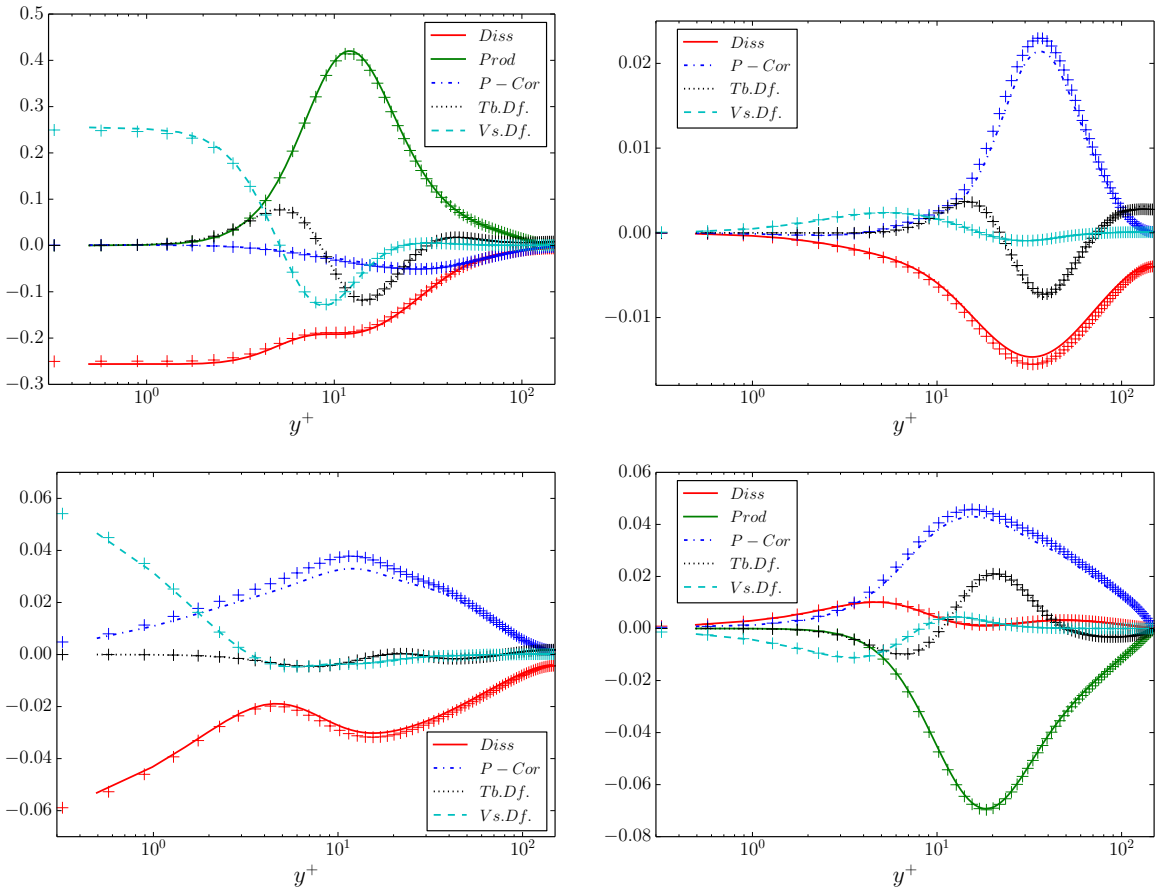

Figure 3.10: Budget of Reynolds Stresses. Clockwise from top-left: R_{xx}, R_{yy}, R_{xy}, R_{zz}. Line: present. Symbol: ref [13]

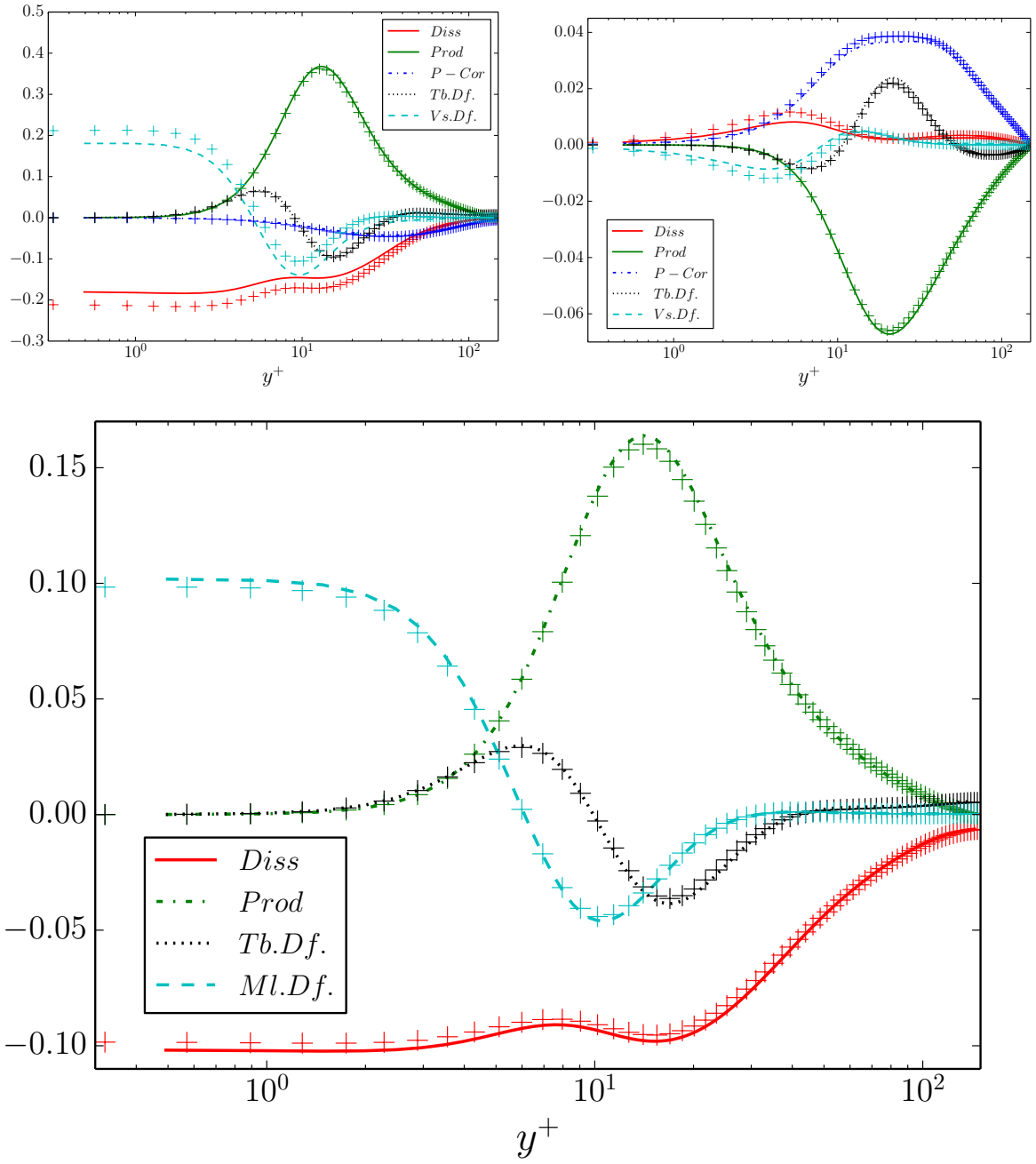

Figure 3.11: Budget of turbulent heat fluxes and temperature variance. Clockwise from top-left: uT, vT, TT. Line: present. Symbol: ref [13]

Constant heat flux

The exact same flow is computed with a Neumann boundary condition on temperature. As there is no longer a constraint on the temperature value at the boundary, a temporal drift may occur. We introduce a new modified temperature: $T = \phi - ax - bt$ where the real temperature ϕ is neither periodic nor stationary while the modified temperature T is both.

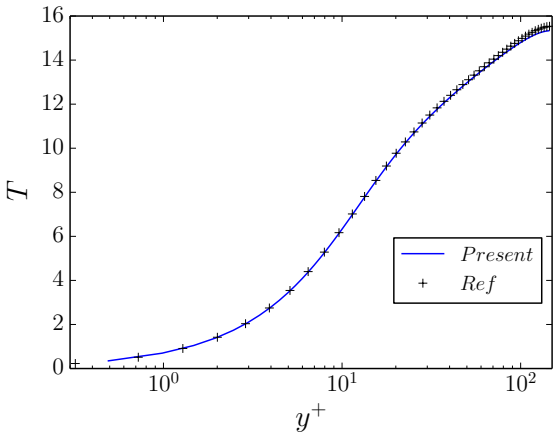

Figure 3.12: Averaged temperature, iso-thermal. Line: present. Symbol: ref [31]

$$\partial_t T + \partial_i \left(T u_i\right) = \frac{1}{Re\ Pr} \partial_{i,i} T - a u_x - b \tag{3.8}$$

The coefficient b corresponds to a source term in the temperature equation (3.8) that compensates a possible temporal drift of temperature. It is computed at every time step to ensure that the bulk temperature is null.

The flow parameters are exactly the ones available in table 3.1. The flow dynamic being identical, only temperature-related statistics are shown. The averaged temperature is presented figure 3.12. The fluctuations are available figure 3.13. Our reference simulation is [31]. Budgets of the fluctuations are available figure 3.14. Such data were not available in the literature, therefore, our iso-flux budgets are compared with iso-thermal ones from [13].

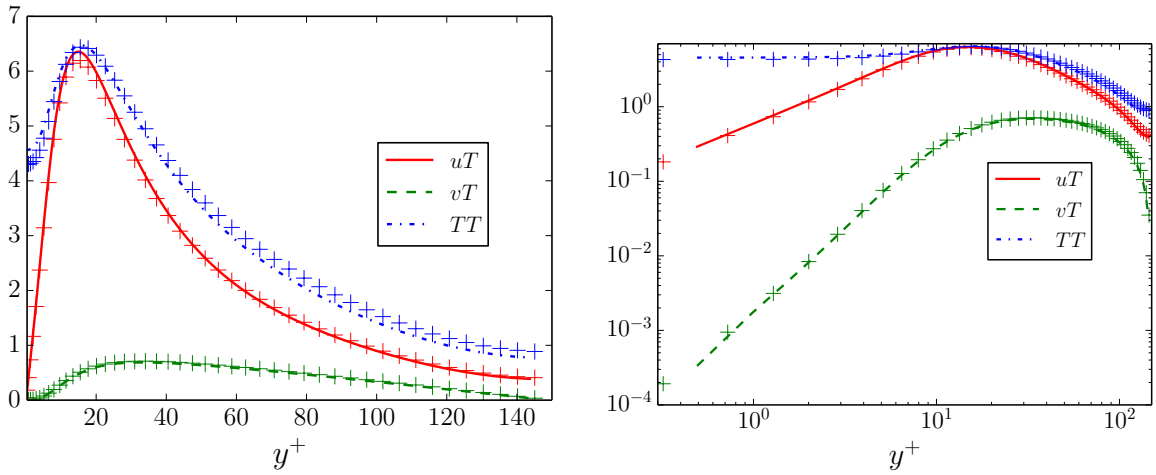

Figure 3.13: Turbulent heat fluxes and variance of temperature. Iso-flux. Left: linear scale. Right: log scale. Line: present. Symbol: ref [31]

From figure 3.13, the temperature variance at the wall is not null: the Neumann boundary condition imposed on the temperature allows fluctuations at the wall. It also enhances the turbulent heat fluxes in the viscous layer compared with the iso-thermal simulation, in agreement with the reference iso-flux simulation from [31].

Budgets available figure 3.14 are more revealing. Slight differences can be now observed in the budget of vT. While production and turbulent diffusion are almost identical, the

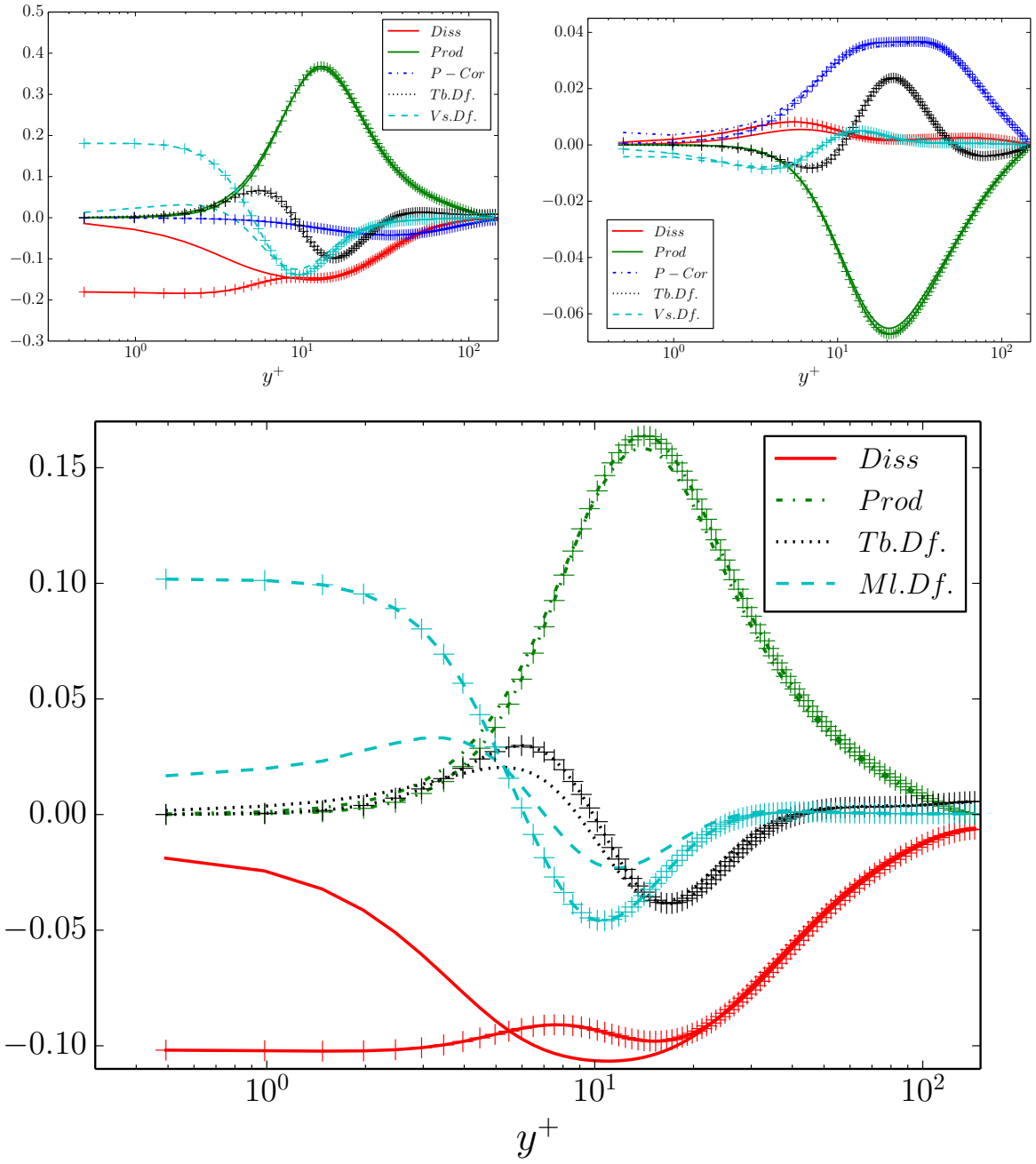

Figure 3.14: Budget of turbulent heat fluxes and temperature variance. Clockwise from top-left: uT, vT, TT. Line: iso-flux, present. Line+symbol: iso-thermal, present

dissipation, viscous diffusion and temperature-pressure gradient correlation seem to behave differently near the wall ($y^+ \leq 10$). While those differences are not obvious and remain within the statistical uncertainty, they are in good agreement with the results of Tiselj et al. [32] for a turbulent flume. From the budgets of uT and TT, large differences between iso-flux and iso-thermal statistics can be observed, especially dissipation and viscous diffusion near the wall. This allows to discuss the asymptotic instantaneous temperature at the wall, starting from equation (3.9).

$$\begin{aligned}
\partial_t T + \partial_j \left(T u_j\right) &= \frac{1}{RePr}\partial_{j,j} T - A u_x \\
T\left(x,y,z,t\right) &= a(x,z,t) + b(x,z,t)y + cy^2 + dy^3 + ey^4 + \cdots \\
u_x &= b_x y + c_x y^2 + d_x y^3 + \cdots \\
u_y &= c_y y^2 + d_y y^3 + \cdots \\
u_z &= b_z y + c_z y^2 + d_z y^3 + \cdots \\
\frac{\partial_{xx} a + 2c + \partial_{zz} a}{RePr} &= \partial_t a \; order\ 0 \\
-Ab_x + \frac{\partial_{xx} b + 6d + \partial_{zz} b}{RePr} &= \partial_t b + \partial_x(ab_x) + 2ac_y + \partial_z(ab_z) \; order\ 1 \\
-Ac_x + \frac{\partial_{xx} c + 12e + \partial_{zz} c}{RePr} &= \partial_t c + \partial_x(ac_x + bb_x) + 3(ad_y + bc_y) + \partial_z(ac_z + bb_z)
\end{aligned} \tag{3.9}$$

Averaging in time and along the periodic directions x and z leads to the temperature variance equation (3.10) where $T' = T - \overline{T}$ and $\overline{T}$ is the average of T.

$$\overline{T'^2} = \overline{a'^2} + 2\overline{a'b'}y + (\overline{b'^2} + 2\overline{a'c'})y^2 + \cdots \tag{3.10}$$

The iso-thermal case constrains the instantaneous temperature at the wall: $a = 0$, leading to $c = 0$ from the 0^{th} order condition. The temperature variance simplifies: $\overline{T'^2} = \overline{b'^2}y^2 + \cdots$. From the boundary layer theory, b is not zero ($T^+ = Pry^+ + \cdots$). The molecular diffusion behaves exactly like the second derivative of the temperature variance, up to a scaling factor. Therefore, the development obtained for the temperature variance gives a non-null molecular diffusion at the wall, in agreement with the iso-thermal simulation.

From figure 3.14, the iso-flux case is different: the molecular diffusion at the wall is much lower. The Neumann condition imposed on the temperature does not restrain the instantaneous temperature at the wall: $a \neq 0$. However, the temperature derivative is imposed: $b = \overline{b}$. The temperature variance simplifies: $\overline{T'^2} = 2\overline{a'c'}y^2 + \cdots$. From 0^{th} order condition, multiplication with a and averaging gives $2\overline{a'c'} = \overline{(\partial_x a)^2} + \overline{(\partial_z a)^2} > 0$. This suggests a non-null molecular diffusion at the wall for the temperature variance, which is not evident in our results.

To conclude this analysis, the importance of the source term is highlighted. If one were to compute the flow between infinite plates at different temperature with a Dirichlet condition on both plates, the source term is no longer necessary and the constant A vanishes. The resulting asymptotic averaged temperature would be $\overline{T} = \overline{b}y + \overline{e}y^4 + \cdots$ with $\overline{e} = \frac{1}{4}RePr\overline{bc_y}$. Similarly, a Neumann condition can be imposed on both plates. The resulting asymptotic temperature would be $\overline{T} = \overline{b}y + \overline{d}y^3 + \cdots$ with $\overline{d} = \frac{1}{3}RePr\overline{ac_y}$. The

asymptotic averaged temperature behaves differently but for both cases, there is a strong connection between averaged temperature and wall-normal heat flux. Integration in the wall-normal direction of the averaged temperature equation leads to equation (3.11) which confirms the asymptotic analysis.

$$\partial_y \overline{T} - RePr\overline{Tu_y} = \partial_y T(y=0) + ARePr \int_{s=0}^{y} \overline{u_x}\,(s)\,ds \tag{3.11}$$

3.2.2 Conjugate heat transfer

The exact same flow is computed with conjugate heat transfer. This first simulation is performed with a unit ratio of thermal diffusivity G and a unit ratio of thermal conductivity $\alpha = \frac{\lambda_f}{\lambda_s}$. The solid domain's size is $[12.8; 1; 4.26]$ while the fluid domain's size is still $[12.8; 2; 4.26]$. The solid is present on top and bottom of the fluid (e.g. $-1 \leq y \leq 0$ and $2 \leq y \leq 3$). The heat flux is imposed at the external boundary of the solid domain. The transient part of the simulation was unexpectedly long but a satifactory convergence was obtained.

From figure 3.15, heat fluxes and temperature variance for conjugate heat transfer are in-between iso-thermal and iso-flux results. From [31], the ratio of temperature RMS at the wall between the iso-flux and conjugate simulation should be around 0.5, we have obtained 0.54.

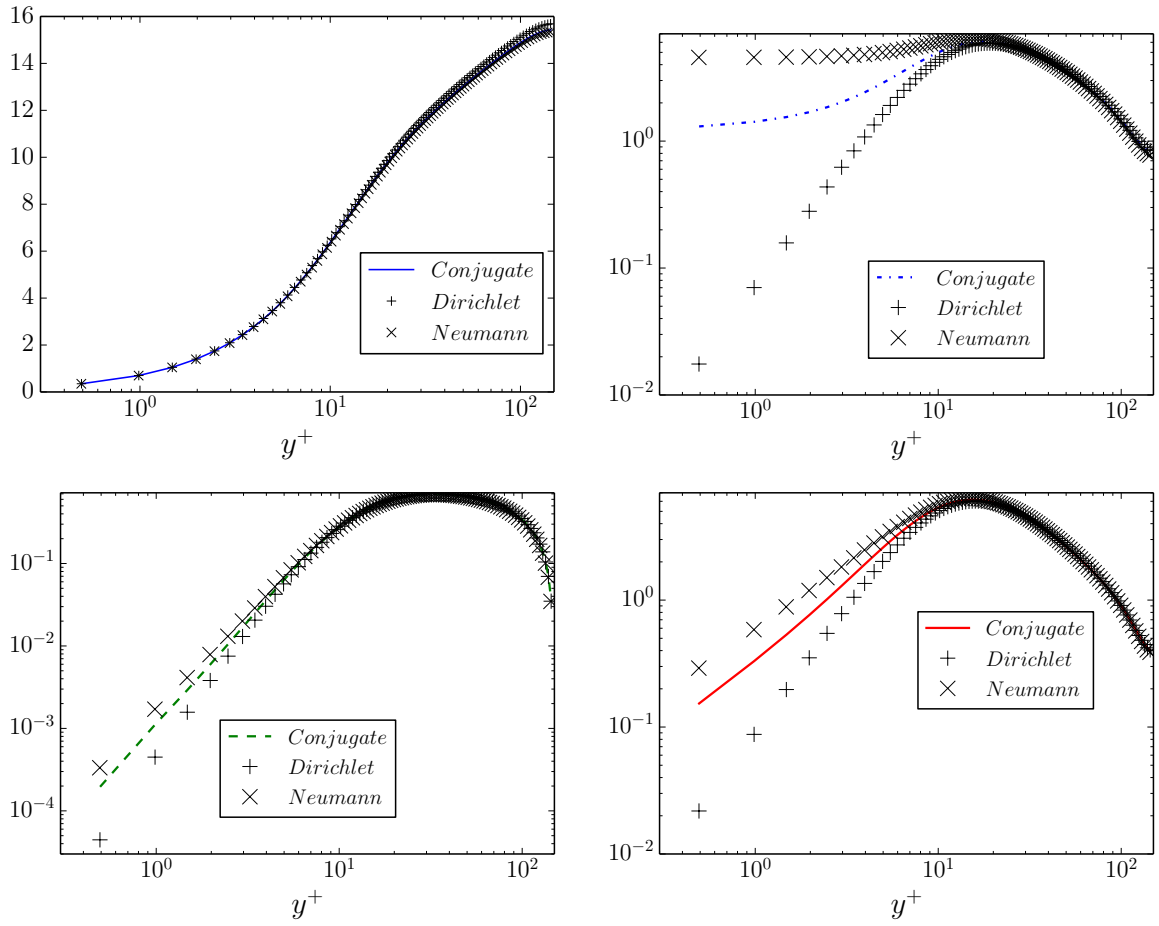

Figure 3.15: Statistics for conjugate heat transfer. Clockwise from top-left: T, TT, uT, vT. Line: conjugate. Symbol: iso-thermal and iso-flux

From figure 3.16, budgets of turbulent heat fluxes and temperature variance are presented. Some differences between iso-thermal and iso-flux simulations were noticed. Results obtained for conjugate heat transfer are in-between. While present results are only qualitative, the impact of the thermal boundary condition on the viscous layer is clear. On one hand, statistics related to temperature variance and axial heat-flux exhibit a strong dependency on the thermal boundary condition. On the other hand, statistics related to the wall-normal heat-flux are remarkably insensitive to the thermal boundary condition.

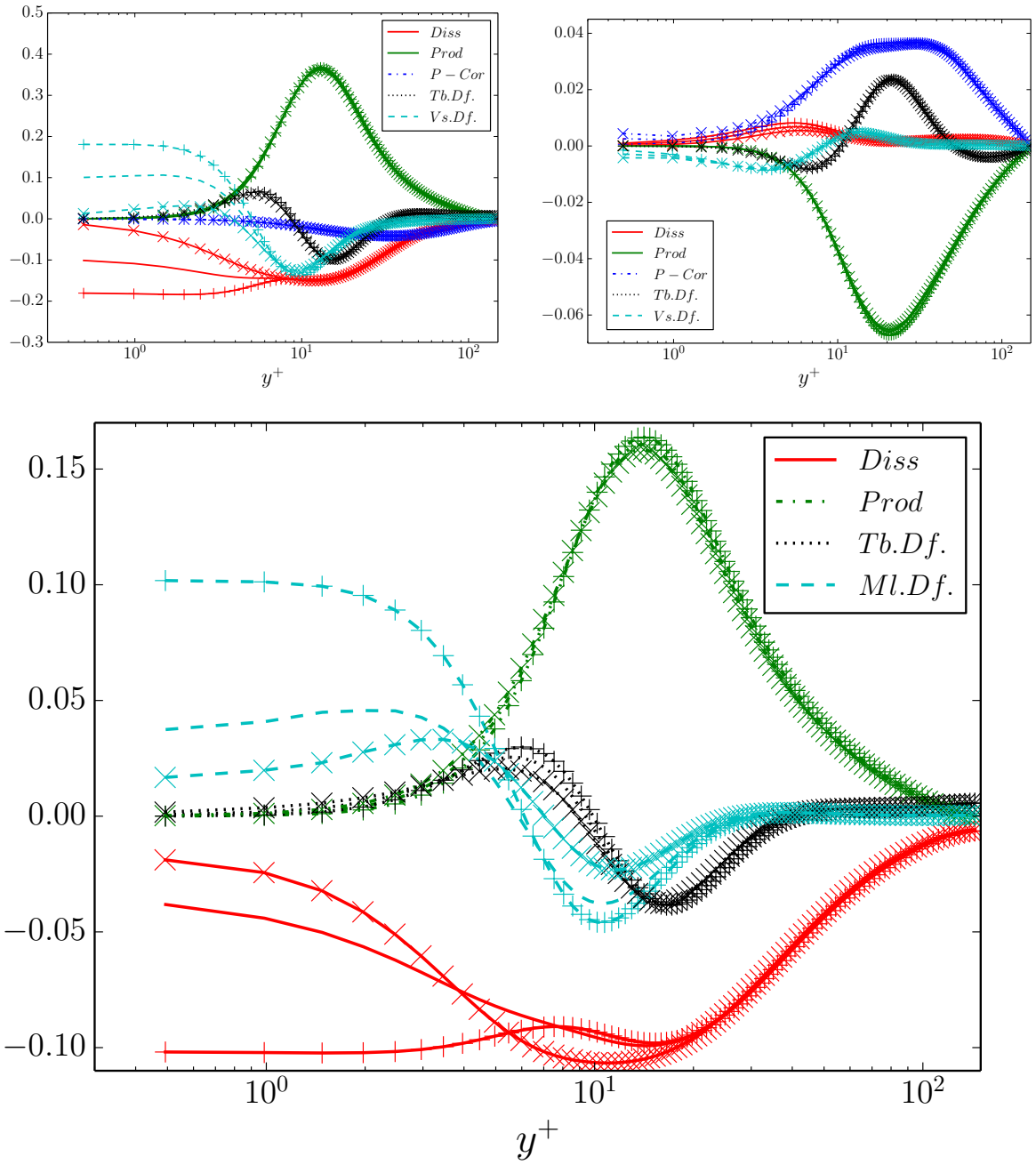

Figure 3.16: Budget of turbulent heat fluxes and temperature variance. Clockwise from top-left: uT, vT, TT. Line: conjugate. Line+symbol: iso-thermal, iso-flux

Statistical convergence is quite good. However, the transient was unexpectedly long for the solid domain. The solid acts as a buffer and damps temperature fluctuations. The more distant the fluid, the more important the initial temperature. Therefore, the solid

domain close to the fluid is directly affected by the turbulence and reaches a stationary state faster while convergence is much slower for the external part of the solid.

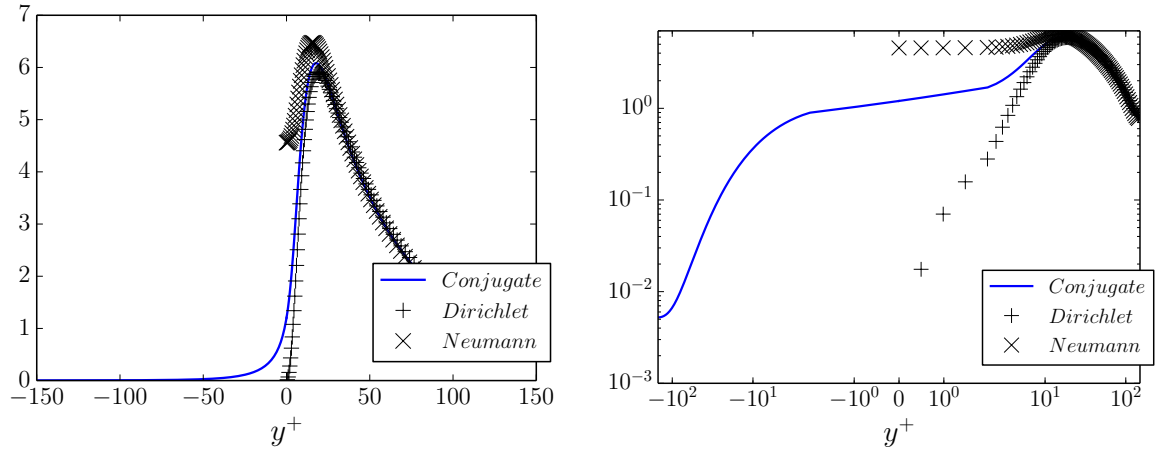

Figure 3.17: Temperature variance for conjugate heat transfer. Left: linear scale. Right: logarithmic scale. Line: conjugate. Symbol: iso-thermal and iso-flux

Chapter 4

Numerical experiences, future work

4.1 Interpolation, a crucial operation

Some numerical experiences were performed to investigate the role of the interpolator and are described here.

Numerical experience

Finite difference schemes are versatile: both the stencil and the coefficients can be tuned to mold the properties of the scheme. During the pressure correction step, some variables are interpolated from the velocity grid to the pressure grid and reciprocally. The stencil of the interpolator was increased and a new set of coefficients was tested in order to improve the correction step. The coefficients are available table 4.1 and the transfer functions expressed equation (4.1) are plotted figure 4.1. The ones called *classic* and *QS6* are 6^{th} order accurate while *Od4: 0.49* is 4^{th} order accurate. The stencil for *classic* and *Od4: 0.49* is $(3;4)$ while it is $(3;8)$ for *QS6*.

$$\begin{aligned} \alpha_I f^I_{i-0.5} + f^I_{i+0.5} + \alpha_I f^I_{i+1.5} &= a_I \frac{f_{i+1}+f_i}{2} + b_I \frac{f_{i+2}+f_{i-1}}{2} \\ &+ c_I \frac{f_{i+3}+f_{i-2}}{2} + d_I \frac{f_{i+4}+f_{i-3}}{2} \\ T^I(k) &= \frac{a_I \cos(\frac{k}{2}) + b_I \cos(\frac{3k}{2}) + c_I \cos(\frac{5k}{2}) + d_I \cos(\frac{7k}{2})}{1 + 2\alpha_I \cos(k)} \end{aligned} \quad (4.1)$$

	α_I	a_I	b_I	c_I	d_I
Classic	$\frac{3}{10}$	$\frac{75+70\alpha}{64}$	$\frac{-25+126\alpha}{128}$	$\frac{3-10\alpha}{128}$	0
QS6	0.461658	$\frac{75+70\alpha-320d}{64}$	$\frac{-25+126\alpha+1152d}{128}$	$\frac{3-10\alpha-640d}{128}$	0.00293016
Od4: 0.49	0.49	$\frac{139}{80}$	$\frac{97}{400}$	0	0

Table 4.1: Numerical schemes for interpolation

The transfer function of scheme *Od4* may look odd. However, it produces accurate low-order statistics on coarse grids. From figure 4.1, a zigzag pattern is clearly visible

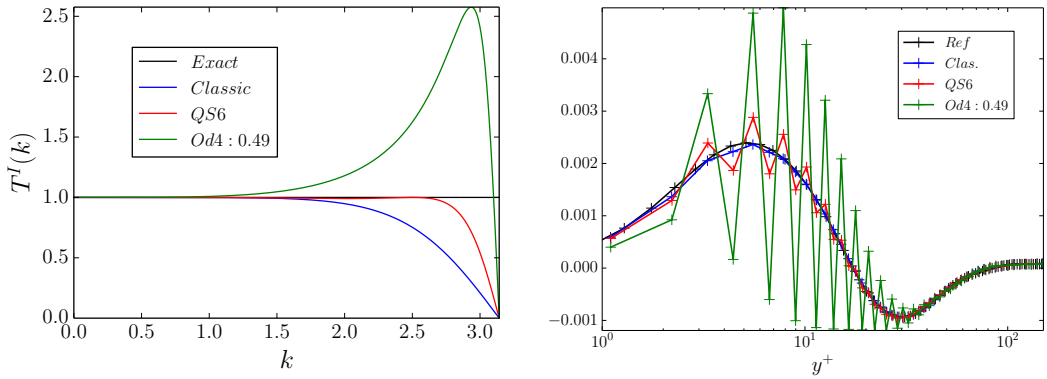

Figure 4.1: Left: Transfer function. Right: Viscous diffusion associated with R_{yy}. Ref: [13]

for $Od4$ in the near-wall region on high-order statistics. First guess for the spurious oscillation was the transfer function, which has a maximum higher than one. Iterating the interpolator produces a sharper and higher peak. This behaviour was suspected to trigger some instabilities. Therefore, interpolator $QS6$ was tested. The transfer function does not exceed 1 and the stencil was extended to improve accuracy at high wavenumbers. The zigzag pattern for $QS6$ is less visible but still present.

Kernel of the Poisson solver

The zigzag pattern observed on the second derivative of the wall normal Reynolds stress is a node-to-node oscillation. It does not affect low-order statistics and seems to be only a numerical issue. Interpolation and staggered derivation are used to compute the input ($\partial_i u_i^*$) and output ($\partial_i p$) of the pressure Poisson solver. The theoretical and numerical kernel of the Poisson solver is discussed here.

For the sake of simplicity, we assume to be working on a homogeneous grid and with a unit time step. As already discussed, interpolation and derivation have a spectral equivalent using the transfer function and the modified wave-number. Staggered interpolation in direction x of a function f can be expressed in spectral space $\widehat{f^{Ix}} = T_x \widehat{f}$. Similarly, staggered derivation can be expressed in spectral space $\widehat{f'} = k_x \widehat{f}$. The transfer function of the staggered interpolation is null only at the cut-off frequency (interpolation of a node-to-node zigzag) while the modified wave-number of the staggered derivation is null only for the null frequency (derivation of a constant), as expected anatically.

$$\begin{aligned} \widehat{\partial_i u_i^*} &= k_x T_y T_z \widehat{u_x^*} + T_x k_y T_z \widehat{u_y^*} + T_x T_y k_z \widehat{u_z^*} \\ \widehat{p} &= \frac{\widehat{\partial_i u_i^*}}{k_x^2 T_y^2 T_z^2 + T_x^2 k_y^2 T_z^2 + T_x^2 T_y^2 k_z^2} \end{aligned} \tag{4.2}$$

The divergence of the predicted velocity and the resulting pressure are expressed equation (4.2). The kernel of the Poisson solver corresponds to a null denominator in the 2^{nd} equation. The inevitable mode is the constant one ($k_x = k_y = k_z = 0$). The $3D$ checkerboard mode occurs at the cut-off in all directions ($T_x = T_y = T_z = 0$). The other modes of the kernel are $2D$ checkerboard when 2 of the 3 transfer functions are null.

$$
\begin{aligned}
\widehat{\partial_x p} &= 0 \\
\widehat{\partial_y p} &= \frac{T_x^2 k_y^2 T_z^2}{T_x^2 k_y^2 T_z^2} \widehat{u_y^*} \\
\widehat{\partial_z p} &= 0
\end{aligned} \tag{4.3}
$$

The pressure gradient obtained at the cut-off frequency in the y direction ($T_y = 0$) is expressed equation (4.3). Among the modes at the cut-off frequency in the y direction, some of them are out of the kernel. For such modes, the pressure gradient equals the predicted velocity. Therefore, any zigzag pattern present on the predicted velocity and out of the kernel of the pressure solver is supposed to be annihilated during the pressure-correction step.

Numerical analysis of the Poisson solver was performed with the library Lapack and the kernel was extracted both for homogeneous and stretched grids. In some cases, part of the kernel was absent. While periodic directions are handled as expected, it is not the case for symmetric and no-slip ones. For non-periodic directions, the solver does not account for the cut-off frequency. As the solver is supposed to cut them down, a simple workaround would be to filter u^* before computing the divergence. For a non-periodic direction i, one would simply cancel the projection of u_i^* on the cut-off frequency harmonics.

4.2 Pressure prediction-correction

The global scheme used in Incompact3d for time-advancement with semi-implicitation of the wall-normal viscous term is recalled equation (4.4) when a 2^{nd} order AB scheme is used. The boundary conditions are also detailed equation (4.5) for a channel flow configuration: x and z are periodic and a no-slip condition is applied at $y = 0$ and $y = L_y$. In the following, the order of accuracy of the global scheme in time is discussed.

$$
\begin{aligned}
\frac{u_i^* - u_i^n}{dt} &= \frac{3}{2}\left[\partial_j\left(u_i^n u_j^n\right) + \frac{1}{Re}\left(\partial_{xx} u_i^n + \partial_{zz} u_i^n - \frac{h"}{h'}\partial_y u_i^n\right)\right] \\
&- \frac{1}{2}\left[\partial_j\left(u_i^{n-1} u_j^{n-1}\right) + \frac{1}{Re}\left(\partial_{xx} u_i^{n-1} + \partial_{zz} u_i^{n-1} - \frac{h"}{h'}\partial_y u_i^{n-1}\right)\right] \\
&- \partial_i p^n + \frac{1}{2Re}\left(\frac{1}{h'}\right)^2 \left(\partial_{ss} u_i^* + \partial_{ss} u_i^n\right) \\
\frac{u_i^{**} - u_i^*}{dt} &= \partial_i p^n \\
\frac{u_i^{n+1} - u_i^{**}}{dt} &= -\partial_i p^{n+1} \\
\frac{1}{dt}\partial_i u_i^{**} &= \Delta p^{n+1}
\end{aligned} \tag{4.4}
$$

$$
\begin{aligned}
u_i^* &= 0 \text{ for } y = 0 \text{ and } y = L_y \\
u_x^{**} &= dt\partial_x p^n \text{ for } y = 0 \text{ and } y = L_y \\
u_y^{**} &= 0 \text{ for } y = 0 \text{ and } y = L_y \\
u_z^{**} &= dt\partial_z p^n \text{ for } y = 0 \text{ and } y = L_y \\
\partial_y p^{n+1} &= 0 \text{ for } y = 0 \text{ and } y = L_y
\end{aligned} \tag{4.5}
$$

Prediction step

The predicted velocity u^* is second-order accurate. However, according to section 6.4 of [2], the wall-normal no-slip condition on u^* causes a numerical stress, which can produce a numerical boundary layer. Therefore, our prediction scheme may have a low accuracy near the wall.

Correction step

During the correction scheme, the pressure gradient is computed assuming the derivative is null at the wall. However, from Navier-Stokes equation, one finds the pressure gradient at the wall compensates the viscous friction. In our simulation, the pressure gradient at the wall does not fulfil this role. Our correction scheme is not faithful in the physical sense. Therefore, the global prediction-correction scheme discussed may have a low accuracy in the near-wall region.

Sketched improvements

Guermond et al. in [8] proposed a *rotational incremental pressure-correction scheme*. The error on the velocity is bounded by dt^2 while it is $dt^{1.5}$ for the pressure. According to their analysis, the splitting error manifests itself only in the form of an inexact tangential boundary condition on the velocity.

$$
\begin{aligned}
\frac{u_i^* - u_i^n}{dt} &= \frac{3}{2}\left[\partial_j\left(u_i^n u_j^n\right) - \partial_i p^n\right] \\
&- \frac{1}{2}\left[\partial_j\left(u_i^{n-1} u_j^{n-1}\right) - \partial_i p^{n-1}\right] \\
&+ \frac{1}{2Re}\left(\partial_{jj} u_i^n + \partial_{jj} u_i^*\right) \\
u_i^* &= 0 \text{ for } y = 0 \text{ and } y = L_y \\
\frac{u_i^{n+1} - u_i^*}{dt} &= -\partial_i \phi^{n+1} \\
\partial_i u_i^{n+1} &= 0 \\
u_y^{n+1} &= 0 \text{ for } y = 0 \text{ and } y = L_y \\
\phi^{n+1} &= \frac{p^{n+1} + p^n}{2} - \frac{3}{2}p^n + \frac{1}{2}p^{n-1} + \frac{1}{2Re}\partial_j u_j^* \\
\partial_n \phi^{n+1} &= 0 \text{ for } y = 0 \text{ and } y = L_y
\end{aligned} \tag{4.6}
$$

A slightly modified version of the scheme proposed in [8] is given equation (4.6). The

curl of a gradient is null and the correction step gives $\nabla \times u^* = \nabla \times u^{n+1}$. Using this relation, summing the prediction and the correction gives equation (4.7).

$$\begin{aligned}
\frac{u_i^{n+1} - u_i^n}{dt} &= -\frac{\partial_i p^{n+1} + \partial_i p^n}{2} + \frac{3}{2}\partial_j \left(u_i^n u_j^n\right) - \frac{1}{2}\partial_j \left(u_i^{n-1} u_j^{n-1}\right) \\
&+ \frac{1}{2Re}\left(\partial_{jj} u_i^n + \partial_{jj} u_i^* - \partial_i \partial_j u_j^*\right) \\
\partial_{jj} u_i^* - \partial_i \partial_j u_j^* &= \partial_{jj} u_i^{n+1} \\
\frac{u_i^{n+1} - u_i^n}{dt} &= -\frac{\partial_i p^{n+1} + \partial_i p^n}{2} + \frac{3}{2}\partial_j \left(u_i^n u_j^n\right) - \frac{1}{2}\partial_j \left(u_i^{n-1} u_j^{n-1}\right) \\
&+ \frac{1}{2Re}\left(\partial_{jj} u_i^n + \partial_{jj} u_i^{n+1}\right)
\end{aligned} \tag{4.7}$$

One can try to adapt this to our scheme where only the wall-normal part of the viscous term is semi-implicit.

$$\begin{aligned}
\frac{u_i^* - u_i^n}{dt} &= \frac{3}{2}\left[\partial_j \left(u_i^n u_j^n\right) - \partial_i p^n + \frac{\partial_{xx} + \partial_{zz}}{Re} u^n\right] \\
&- \frac{1}{2}\left[\partial_j \left(u_i^{n-1} u_j^{n-1}\right) - \partial_i p^{n-1} + \frac{\partial_{xx} + \partial_{zz}}{Re} u^{n-1}\right] \\
&+ \frac{1}{2Re}\left(\partial_{yy} u_i^n + \partial_{yy} u_i^*\right) \\
\frac{u_i^{n+1} - u_i^n}{dt} &= -\frac{\partial_i p^{n+1} + \partial_i p^n}{2} + \frac{3}{2}\partial_j \left(u_i^n u_j^n\right) - \frac{1}{2}\partial_j \left(u_i^{n-1} u_j^{n-1}\right) \\
&+ \frac{3}{2Re}\left(\partial_{xx} + \partial_{zz}\right) u^n - \frac{1}{2Re}\left(\partial_{xx} + \partial_{zz}\right) u^{n-1} \\
&+ \frac{1}{2Re}\left(\partial_{yy} u_i^n + \partial_{yy} u_i^* - \partial_i \partial_j u_j^*\right)
\end{aligned} \tag{4.8}$$

The predicted velocity and the sum are modified according to equation (4.8). One can try to rearticulate the sum.

$$\begin{aligned}
\partial_{yy} u_i^* - \partial_i \partial_j u_j^* &= \partial_{jj} u_i^* - \partial_i \partial_j u_j^* - \partial_{xx} u_i^* - \partial_{zz} u_i^* \\
&= \partial_{jj} u_i^{n+1} - \partial_{xx} u_i^* - \partial_{zz} u_i^* \\
\frac{u_i^{n+1} - u_i^n}{dt} &= -\frac{\partial_i p^{n+1} + \partial_i p^n}{2} + \frac{3}{2}\partial_j \left(u_i^n u_j^n\right) - \frac{1}{2}\partial_j \left(u_i^{n-1} u_j^{n-1}\right) \\
&+ \frac{1}{2Re}\left(\partial_{jj} u_i^n + \partial_{jj} u_i^{n+1}\right) \\
&- \frac{1}{2Re}\left(\partial_{xx} + \partial_{zz}\right)\left(u_i^* - 2u^n + u^{n-1}\right)
\end{aligned} \tag{4.9}$$

From equation (4.9), one can see the scheme proposed by Guermond et al. [8] cannot be directly adapted to our semi-implicit formulation of the viscous term. Some deeper modifications may allow second-order accuracy but would require tedious analysis of the scheme. A simpler solution seems to be a semi-implicit formulation of the whole viscous term, allowing to use directly the scheme (4.6). Considering the $2D$ stencil configuration of the code, the alternating direction implicit method or a Douglas-Gunn procedure seems fitted.

To conclude on the zigzag pattern observed on high-order statistics, analysis of a instantaneous field was performed. The pressure field was extracted at a random x and z location, see figure 4.2. The pressure field is smooth but derivative at the wall is not null. Staggered derivation assuming a null derivative at the boundary on such a field is vowed to fail and seems responsible for the zigzag pattern observed on some statistics.

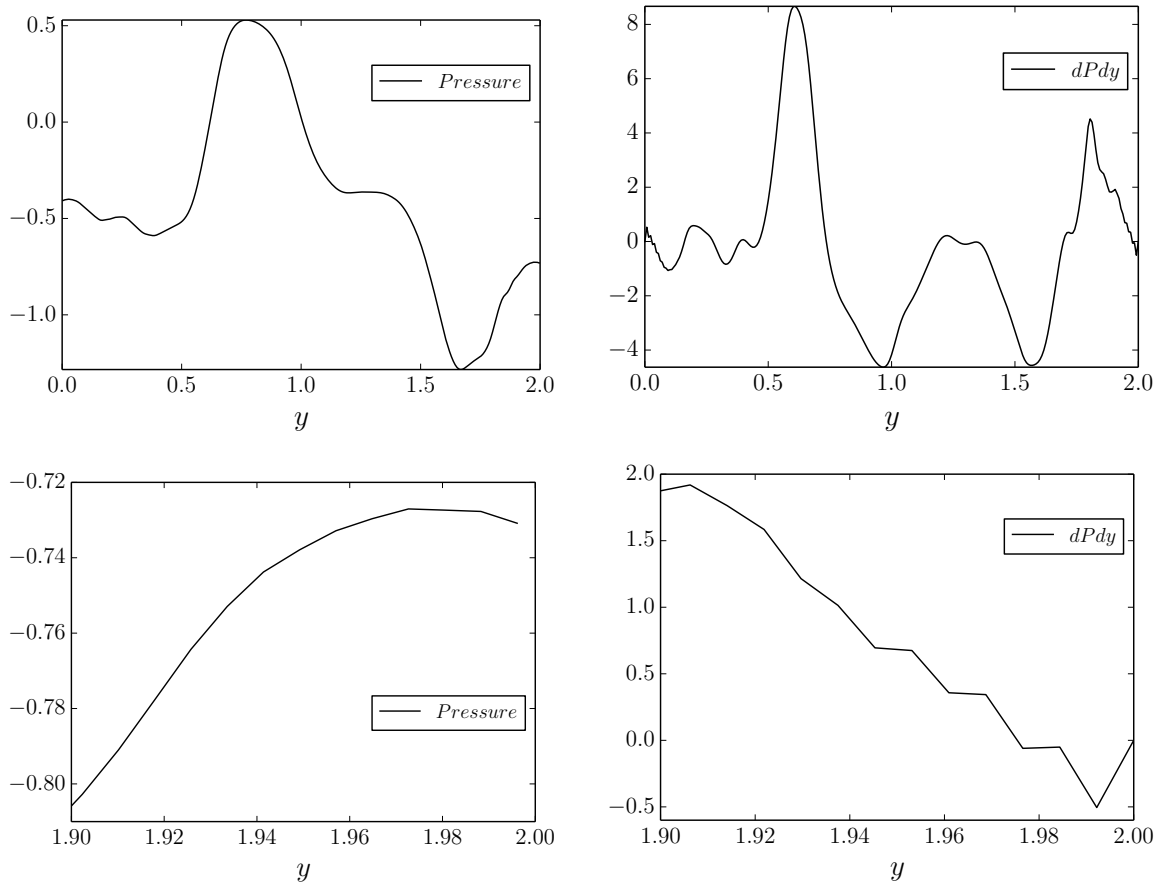

Figure 4.2: Left: Pressure. Right: Wall-normal pressure gradient. Top: global. Bottom: zoom

Chapter 5

Conclusion

Our main objective was to develop a conjugate heat transfer module within the DNS code Incompact3d and test its suitability for generating a database of second-order moments, i.e. of Reynolds stresses, heat fluxes, temperature fluctuations and eventually their budgets, which had not been done before with this code.

In a first step, the semi-implicitation of the wall-normal viscous diffusion was adapted to a generic configuration. This allowed us to use generic boundary conditions for the temperature solver of the fluid domain. Following this adaptation, a temperature solver for a solid domain on top and below the fluid domain was developed.

Both developments were checked against analytical solutions and academic turbulent flows. The ability of the code to compute high-order statistics is now established. On this basis, more complex configurations can be simulated and assumptions used to model turbulence close to the wall assessed.

From the numerical analysis in the final section, numerical oscillations near the solid walls have been observed on the wall normal Reynolds stress second derivative. As this term is balanced by pressure strain term at the wall limit, the suspect is deficiencies in the wall-normal pressure-velocity coupling in the near wall limit.

Deeper investigations have been performed to understand their source. The prediction-correction scheme adapted from [8] seems promising but may require additional developments: semi-implicitation of the full viscous term.

It was also noted that in conjugate heat transfer case the mean temperature in the solid needs a very long time to reach equilibrium, i.e. longer than that usually required to collect turbulence statistics. However this physical feature could be circumvented by initializing the temperature field from a precursor refined RANS simulation.

Bibliography

[1] DA Anderson, JC Tannehill, and RH Pletcher. Computational fluid mechanics and heat transfer. 1984.

[2] David L Brown, Ricardo Cortez, and Michael L Minion. Accurate projection methods for the incompressible navier–stokes equations. *Journal of Computational Physics*, 168(2):464–499, 2001.

[3] E. Buckingham. On physically similar systems; illustrations of the use of dimensional equations. *Phys. Rev.*, 4:345–376, Oct 1914.

[4] Olivier Cadot. *Introduction à la turbulence*. 2008. Course textbook.

[5] Jean-Pierre Chollet and Marcel Lesieur. Parameterization of small scales of three-dimensional isotropic turbulence utilizing spectral closures. *Journal of Atmospheric Sciences*, 38:2747–2757, 1981.

[6] Massimo Germano, Ugo Piomelli, Parviz Moin, and William H Cabot. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A: Fluid Dynamics*, 3:1760, 1991.

[7] Sandip Ghosal, Thomas S Lund, Parviz Moin, and Knut Akselvoll. A dynamic localization model for large-eddy simulation of turbulent flows. *Journal of Fluid Mechanics*, 286(1):229–255, 1995.

[8] JL Guermond, Peter Minev, and Jie Shen. An overview of projection methods for incompressible flows. *Computer methods in applied mechanics and engineering*, 195(44):6011–6045, 2006.

[9] Kemal Hanjalić and Brian Launder. *Modelling turbulence in engineering and the environment: second-moment routes to closure*. Cambridge University Press, 2011.

[10] Radim Havel. *Guidelines on Pressurized Thermal Shock Analysis for WWER nuclear power plants. Revision 1.* IAEA-EBP-WWER-08 (Rev. 1). International Atomic Energy Agency, January 2006.

[11] Sergio Hoyas and Javier Jiménez. Scaling of the velocity fluctuations in turbulent channels up to re= 2003. *Physics of fluids*, 18:011702, 2006.

[12] Seongwon Kang, Gianluca Iaccarino, and Frank Ham. Direct numerical simulation of buoyancy-dominated turbulent flows on a bluff body using the immersed boundary method. *Journal of Computational Physics*, 228(9):3189–3208, 2009.

[13] N Kasagi, Y Tomita, and A Kuroda. Direct numerical simulation of passive scalar field in a turbulent channel flow. *ASME Transactions Journal of Heat Transfer*, 114:598–606, 1992.

[14] John Kim, Parviz Moin, and Robert Moser. Turbulence statistics in fully developed channel flow at low reynolds number. *Journal of Fluid Mechanics*, 177(1):133–166, 1987.

[15] Andrey Nikolaevich Kolmogorov. The local structure of turbulence in incompressible viscous fluid for very large reynolds numbers. In *Dokl. Akad. Nauk SSSR*, volume 30, pages 299–303, 1941.

[16] Sylvain Laizet and Eric Lamballais. High-order compact schemes for incompressible flows: A simple and efficient method with quasi-spectral accuracy. *Journal of Computational Physics*, 228(16):5989–6015, 2009.

[17] Sylvain Laizet and Ning Li. Incompact3d: A powerful tool to tackle turbulence problems with up to o (100k) computational cores. *International Journal for Numerical Methods in Fluids*, 67(11):1735–1757, 2011.

[18] Eric Lamballais, Véronique Fortuné, and Sylvain Laizet. Straightforward high-order numerical dissipation via the viscous term for direct and large eddy simulation. *Journal of Computational Physics*, 2011.

[19] Brian E. Launder and Neil D. Sandham. *Closure strategies for turbulent and transitional flows*. Cambridge University Press, 2002.

[20] Sanjiva K Lele et al. Compact finite difference schemes with spectral-like resolution. *Journal of Computational Physics*, 103(1):16–42, 1992.

[21] Ning Li and Sylvain Laizet. 2decomp&fft-a highly scalable 2d decomposition library and fft interface. *Cray User Group 2010*, 2010.

[22] MS Loginov, EMJ Komen, and T Höhne. Application of large-eddy simulation to pressurized thermal shock: Assessment of the accuracy. *Nuclear Engineering and Design*, 241(8):3097–3110, 2011.

[23] MS Loginov, EMJ Komen, and AK Kuczaj. Application of large-eddy simulation to pressurized thermal shock problem: A grid resolution study. *Nuclear Engineering and Design*, 240(8):2034–2045, 2010.

[24] Rajat Mittal and Gianluca Iaccarino. Immersed boundary methods. *Annu. Rev. Fluid Mech.*, 37:239–261, 2005.

[25] Steven A Orszag. Numerical simulation of incompressible flows within simple boundaries: I-galerkin (spectral) representations. *J. Fluid Mech*, 49(1):75–112, 1971.

[26] Stephen B Pope. *Turbulent flows*. Cambridge university press, 2000.

[27] Philipp Schlatter, Ramis Örlü, Qiang Li, Geert Brethouwer, Jens HM Fransson, Arne V Johansson, P Henrik Alfredsson, and Dan S Henningson. Turbulent boundary

layers up to re= 2500 studied through simulation and experiment. *Physics of fluids*, 21:051702, 2009.

[28] Scott E Sherer and James N Scott. High-order compact finite-difference methods on general overset grids. *Journal of Computational Physics*, 210(2):459–496, 2005.

[29] Joseph Smagorinsky. General circulation experiments with the primitive equations: I. the basic experiment. *Monthly weather review*, 91(3):99–164, 1963.

[30] KR Sreenivasan. Fractals and multifractals in fluid turbulence. *Annual Review of Fluid Mechanics*, 23(1):539–604, 1991.

[31] Iztok TISELJ, Robert BERGANT, Borut MAVKO, Ivan BAJSIC, and Gad HETSRONI. Dns of turbulent heat transfer in channel flow with heat conduction in the solid wall. *Journal of heat transfer*, 123(5):849–857, 2001.

[32] Iztok Tiselj, Elena Pogrebnyak, Changfeng Li, Albert Mosyak, and Gad Hetsroni. Effect of wall boundary condition on scalar transfer in a fully developed turbulent flume. *Physics of Fluids (1994-present)*, 13(4):1028–1039, 2001.

[33] Edward R Van Driest. On turbulent flow near a wall. *Journal of the Aeronautical Sciences (Institute of the Aeronautical Sciences)*, 23(11), 1956.

[34] Aimé Vaschy. Sur les lois de similitude en physique. In *Annales télégraphiques*, volume 19, pages 25–28, 1892.

[35] Petar V. Vukoslavčević and James M. Wallace. Using direct numerical simulation to analyze and improve hot-wire probe sensor and array configurations for simultaneous measurement of the velocity vector and the velocity gradient tensor. *Physics of Fluids*, 25(11), 2013.