UCL
Université
catholique
de Louvain

École polytechnique de Louvain (EPL)

Simulation of a low Prandtl shear layer

Dissertation presented by
Jérémy Salerno

for obtaining the Master's degree in
Mechanical Engineering

Supervisors
Philippe Chatelain, Yann Bartosiewicz, Grégoire Winckelmans

Reader
Jean-François Remacle

Academic year 2016-2017

Acknowledgments

At the end of my master thesis, I would like to thanks all the people that who made this work possible by helping me not only this year but in the last five years.

First of all, I would like to thanks all my supervisors the professors Philippe Chatelain, Yann Bartosiewicz and Grégoire Winckelmans for their guidance and their help that has been more than once needed.

I would also like to thanks the assistants or researchers whose help, pieces of advice and time were very welcome. I would like to particularly thanks Denis-Gabriel Caprace whose office door is never closed and always took the time to help. But also Matthieu Duponchel for its help at the understanding of the physics of the problem.

Also, finally, I would like to thanks my family and my fiancé who supported me and motivated me to do my best

Computational resources have been provided by the Consortium des Équipements de Calcul Intensif (CÉCI), funded by the Fonds de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under Grant No. 2.5020.11

Contents

List of Figures

List of Symbols

Symbols	**Units**	**Description**
Pr	$-$	Prandtl number
ν	m^2/s	Kinematic viscosity
α	m^2/s	Heat diffusivity
Re	$-$	Reynolds number
$\mathbf{u}$	m/s	Velocity
P	m^2/s^2	Reduced pressure
p	Pa	Pressure
ρ	kg/m^3	Density
$\boldsymbol{\omega}$	$1/s$	Vorticity
T	K	Temperature
c	$J/kg/K$	Heat capacity
$\mathbf{x}$	m	Position
V	m^3	Volume
$\boldsymbol{\alpha}$	m^3/s	Integral of vorticity over a volume
α_T	Tm^3	Integral of temperature over a volume
h	m	Grid size
δ_ω	m	Vorticity thickness
k_{KH}	$1/m$	Kelvin-Helmholtz mode
λ_{KH}	m	Kelvin-Helmholtz wave length
t	s	Time

Introduction and objectives

In Belgium, there are currently two active nuclear power plants for a total of seven reactors. However, these reactors were all constructed in the 1970s and 1980s. They are seriously starting to age and up to now, none of them is programmed be operated after the year 2025. Therefore, if we are unable to find an alternative solution to produce electricity, many of these reactors will need to be replaced by new ones. Since their construction, the design of nuclear reactors has widely changed.

One of the drawbacks of the old reactors is their inefficiency. Roughly only 30 percent of the heat released during the nuclear reaction is transformed into electricity. This poor efficiency is in part due to the relatively low temperature at which the nuclear reactors operate (300 c°). This limitation comes from the fact that they use pressurized water as coolant. Therefore, to operate at a higher temperature and thus, to have a better efficiency, several new concepts of reactors advocates to use liquid metals, such as Sodium or Lead Bismuth Eutectic (LBE), as coolant.

Concerning heat transfer, the main difference between liquid metals and pressurized water is the Prandtl number $Pr = \frac{\nu}{\alpha}$ where ν is the kinematic viscosity and α is the heat diffusivity. For the pressurized water, the Prandtl number is of the order of 1 but for liquid metals, i.e. LBE, it is of the order of 10^{-2}. This dramatically changes the behavior of the heat transfer and thus, the models used in Reynolds Averaged Navier-Stokes (RANS) code used to design the reactors needs to re-calibrated. These calibrations are often based on experimental results but Direct Numerical Simulation (DNS) results can also be used.

To this extent, this master thesis concerns the development of a computational tool allowing the study of unbounded low Prandtl number flows, such as the time-developing shear/mixing layer. The method implemented during the course of this thesis is based on the Vortex Particle-Mesh (Vpm), a numerical tool developed at UCL, and permits to simulate the DNS of high Reynolds number flows.

This work also involves the simulation of a time-developing mixing layer with a low Prandtl number to study unconfined heat transfer. In the literature, the case of time-developing mixing layer was already studied. In this document, the results obtained are compared with two of those studies. The first one study was conducted by Watanabe et al. [1] at the Reynolds number $Re = 380$ but for a high Schmidt number flow. The other one was conducted by Duponcheel and Bartosiewicz [6] at $Re = 700$ for Prandtl numbers between 1 and 10^{-2}.

The first chapter of this document is assigned to the presentation of the numerical tool implemented. First, it sets the model used, the Navier-Stokes equations. Then, it briefly introduces the driving concepts of Vpm. Follows the presentation of the code developed during the year of work and finally it stages the initial condition for the time-developing mixing layer simulations.

The second chapter is focused on the validation of the code implemented. In this optic, it presents a test scenario that possesses an analytical solution. The results obtained are compared with this solution.

The third chapter presents the results of the different simulations. First, it describes the simulation and their driving parameters. Then, it analyses the results obtained and compares them to the literature.

At the end of this document, conclusions are drawn on the obtained results and a perspective to improve the method is presented.

Chapter 1

Numerical method

1.1 The equations

The Navier-Stokes equations for an incompressible Newtonian fluid are used to model the behavior of the flow. The velocity-pressure formulation for these reads

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

$$\frac{D\mathbf{u}}{Dt} = -\nabla P + \nu \nabla^2 \mathbf{u} \tag{1.2}$$

where $\mathbf{u} = (u, v, w)$ is the velocity, $\frac{D}{Dt} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla$ is the material derivative, ν is the kinematic viscosity and $P = \frac{p}{\rho}$ is the reduced pressure where p is the pressure and ρ is the density.
The Navier-Stokes equations can also be written as

$$\frac{D\boldsymbol{\omega}}{Dt} = (\boldsymbol{\omega} \cdot \nabla)\,\mathbf{u} + \nu \nabla^2 \boldsymbol{\omega} \tag{1.3}$$

$$\boldsymbol{\omega} = \nabla \times \mathbf{u} \Rightarrow \nabla \times \boldsymbol{\omega} = \nabla^2 \mathbf{u} \tag{1.4}$$

where $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ is the vorticity. This is known as the velocity-vorticity formulation. In this one the pressure does not appear but the equation (1.4) has to be solved to find the velocity field needed to time integrate the equation (1.3). Moreover, from the definition of the vorticity, it respects the following property

$$\nabla \cdot \boldsymbol{\omega} = \nabla \cdot (\nabla \times \mathbf{u}) = 0 \tag{1.5}$$

which is called the solenoidal property.
In the optic of this work, the equation added to model the heat transfer is the energy equation for an incompressible flow

$$\frac{DT}{Dt} = 2\frac{\nu}{c}\underline{\underline{\mathbf{d}}} : \underline{\underline{\mathbf{d}}} + \alpha \nabla^2 T \tag{1.6}$$

where T is the temperature, $\alpha = \frac{\nu}{Pr}$ is the thermal diffusivity with Pr the Prandtl number, c is the heat capacity assumed constant and $\underline{\underline{\mathbf{d}}} = \frac{1}{2}\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^t\right)$ is the velocity strain tensor. The first term of the right hand side of the equation (1.6) is the dissipation work done by the shear stress and we made the assumption to ignore it for the rest of this work since it is negligible in high Reynolds number flows. Thus, the equation used to model the heat transfer reduces to

$$\frac{DT}{Dt} = \alpha \nabla^2 T \tag{1.7}$$

which is an convection-diffusion equation with no source term.
In addition, the two following assumptions were made:

1. there are no buoyancy forces;
2. the density remains constant.

Under these assumptions, the dynamic of the flow is not influenced by the temperature field. Therefore, VPM could be used to time integrate the Navier-Stokes equations and the addition of the temperature field does not interfere with it.
To summarize, the equations time integrated are equations (1.3) and (1.7) while equation (1.4) is solved to allow the time integration of the others.

1.2 The Vortex Particle-Mesh method

The method used here to simulate the flow is a hybrid vortex particle-mesh method (VPM [2]). It is an in-house code developed at UCL-EPL based on the Parallel Particles Mesh (PPM) libraries written in Fortran using Message Passing Interface (MPI) to allow fast parallel computations on multiple processors. The resolution of the flow is based on the Lagrangian discretization of the velocity-vorticity formulation of the Navier-Stokes equations (1.3) and (1.4).

This method combines the use of particles and of a mesh. Each particle is characterized by its position ($\mathbf{x}_p$), volume (V_p) and strength ($\boldsymbol{\alpha}_p$). This strength represents the integral of vorticity over the volume of the particle

$$\boldsymbol{\alpha}_p = \int_{V_p} \boldsymbol{\omega} d\mathbf{x} \triangleq \boldsymbol{\omega}_p V_p. \tag{1.8}$$

The particles are used as carriers for the vorticity. This means that the convective part of the Navier-Stokes equations is resolved by the time integration of the position of the particles. The strength of the particles also needs to be time integrated to take into account the diffusion and vortex stretching term. The time integration scheme used in VPM is a third-order low storage Runge-Kutta scheme. This leads to the equations

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}_p \tag{1.9}$$

$$\frac{d\boldsymbol{\alpha}_p}{dt} = \int_{V_p} \Big((\boldsymbol{\omega} \cdot \nabla)\, \mathbf{u} + \nu \nabla^2 \boldsymbol{\omega} \Big)\, d\mathbf{x} \tag{1.10}$$

where $\mathbf{u}_p = \mathbf{u}(\mathbf{x}_p, t)$ is the particle velocity. Lagrangian methods are good to simulate convection-dominated flow, such as high Reynolds flows, because they are not constraint by the Courant–Friedrichs–Lewy (CFL) condition while methods that discretize the convection term needs to respect it. However, for a method using only particles, the computations of $\mathbf{u}_p$ and operators such as ∇ are quite difficult and resources consuming.

To counteract these drawbacks VPM uses a mesh. In fact, solving the equation (1.4) can quickly be done on a mesh with the use of fast Fourier transforms (FFT). The mesh is also very useful to compute the vortex stretching and diffusion term to integrate the strength of the particles.

The mesh is also used to do a remeshing of the particles. Indeed, another drawback of the particles methods is that they suffer from inaccuracy due to the Lagrangian distortion. Therefore, to prevent this phenomenon, VPM implements a remeshing operation which consists on positioning the particles at their initial position. The mesh is then used to store the vorticity field during this operation. The figure 1.1 schematizes how the VPM works.

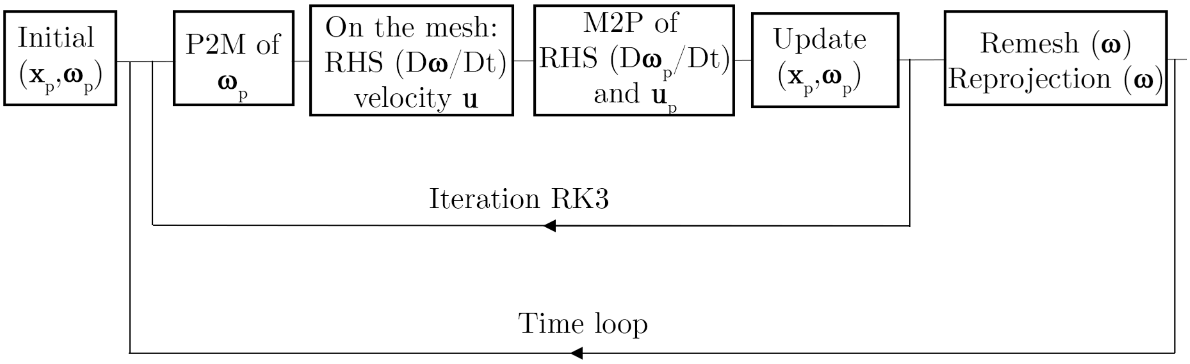

Figure 1.1: Summary of the VPM scheme of integration.

The particles have an initial state (position and vorticity). At each time sub-step, the vorticity is interpolated on the mesh (P2M). On this mesh, the velocity field is computed from the vorticity (equation (1.4)) and also the right hand side of equation (1.3). This right hand side is then interpolated onto the particles as well as their velocity (M2P). The state of the particles is updated (equations (1.9) and (1.10)). These three steps are included inside the third order Runge-Kutta scheme.

Futhermore, VPM also introduces a remeshing operation and a solenoïdal reprojection. The remeshing consists on interpolating the vorticity on the mesh, reinitializing the position of the particles and interpolating the vorticity back onto the particles. The solenoidal reprojection is done to ensure that the vorticity field remain at zero divergence to respect equation (1.5).

1.3 Modifications of Vpm for the treatment of temperature

As said in the previous section, VPM is only used to simulate the incompressible fluid dynamic and thus, does not incorporate a passive scalar field, such as the temperature. Therefore, the goal of this work is to implement in VPM the support to time integrate equation (1.7). As for the resolution of the Navier-Stokes equations, the choice was made to use the particles-mesh dual discretization:

- the temperature is added to the particles which will be convected by the local flow. This carry out the integration of the convective part of equation (1.7),
- the mesh is used to compute the diffusion of the temperature field. This is implemented both in a explicit way using the third order low storage Runge-Kutta scheme and implicitly using the Euler implicit scheme.

1.3.1 Convective part

Adding the temperature on the particles is done as follow: each particle instead of having just a vorticity ($\boldsymbol{\omega}_p$), also has a temperature (T_p). So each particle p has now a strength $\alpha_{T,p}$. Similarly as for the vorticity, the strength of a particle is given by the integral of the temperature over its volume

$$\alpha_{T,p} = \int_{V_p} T d\mathbf{x} \triangleq V_p T_p. \tag{1.11}$$

Therefore, as the particles are convected by the flow, the temperature field is also convected. This is done by the time integration of the position of each particle in the third order low storage Runge-Kutta scheme by Vpm

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}_p \tag{1.12}$$

where $\mathbf{u}_p$ is the velocity of the particle. This is handled by Vpm and no further work is required here.

The remeshing operation needs to implemented for the temperature of the particles. Therefore, to interpolate the temperature field on the mesh, the M_4' interpolation scheme is used

$$T(\mathbf{x}) = \sum_p \frac{\alpha_p}{h^3} M_4' \left(\frac{\mathbf{x} - \mathbf{x}_p}{h} \right) \tag{1.13}$$

where $\mathbf{x}$ is the position of the mesh point onto which the interpolation is done. The interpolation from the mesh onto the particles is also done using M_4'. This operation reads

$$\alpha_{T,p} = \sum_{\mathbf{x}} h^3 T(\mathbf{x}) M_4' \left(\frac{\mathbf{x} - \mathbf{x}_p}{h} \right). \tag{1.14}$$

The figure 1.2 summarizes how the convection of temperature was included inside the integration scheme of Vpm.

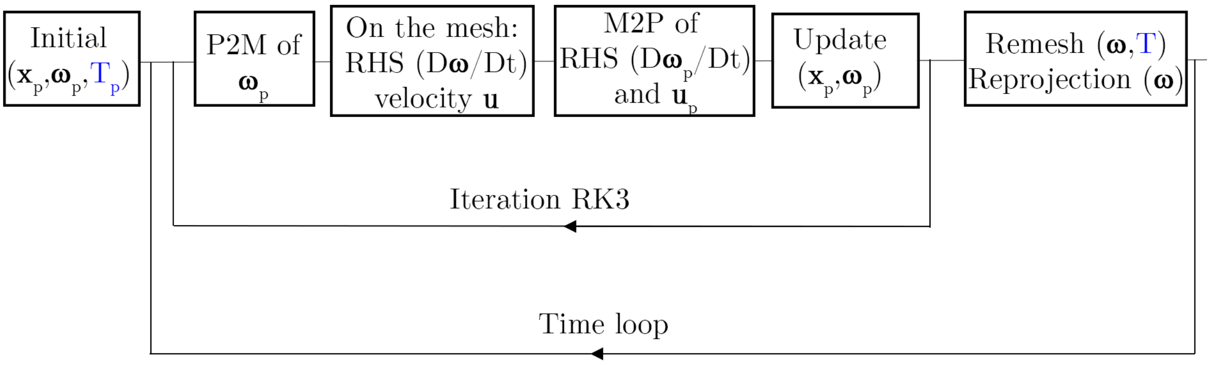

Figure 1.2: Summary of the Vpm scheme of integration with the addition of the convection of temperature (blue).

1.3.2 Diffusive part

As it is done for the vorticity of the particles, their temperature needs to be integrated with the diffusion term.

To achieve efficiency, this term is computed on the mesh and interpolated on the particles. The equation (1.7) can be rewritten for a particle as

$$\frac{d\alpha_{T,p}}{dt} = \int_{V_p} \alpha \nabla^2 T d\mathbf{x}. \tag{1.15}$$

The first approach developed is to do the diffusion explicitly. This can be used to simulate flow at a Prandtl number $Pr \geq 1$ without imposing any supplementary constraint on the time step and thus, not slowing down VPM. As said previously, the explicit scheme implemented is the third order low storage Runge-Kutta scheme which is depicted below

$$\begin{aligned}
& z^{(0)} \leftarrow T^n \\
\text{Substep 1 : } & q^{(1)} = \Delta t f\left(z^{(0)}\right) \\
& z^{(1)} = z^{(0)} + \frac{1}{3} q^{(1)} \\
\text{Substep 2 : } & q^{(1)} = \Delta t f\left(z^{(1)}\right) - \frac{5}{9} q^{(1)} \\
& z^{(2)} = z^{(1)} + \frac{15}{16} q^{(2)} \\
\text{Substep 3 : } & q^{(3)} = \Delta t f\left(z^{(2)}\right) - \frac{153}{128} q^{(2)} \\
& z^{(3)} = z^{(2)} + \frac{8}{15} q^{(3)} \\
& z^{(3)} \rightarrow T^{n+1}
\end{aligned}$$

where $f(*) = \alpha \nabla^2 (*)$ is the diffusion operator, $z^{(*)}$ and $q^{(*)}$ are temporary storage arrays. This is a third order scheme and the spatial discretization of the diffusion operator is done using the fourth order centered cross pencil. In one direction it reads

$$\frac{\partial^2 T}{\partial x^2} \approx \frac{-T_{i+2} + 16T_{i+1} - 30T_i + 16T_{i-1} - T_{i-2}}{12h^2}. \tag{1.16}$$

These choices are made as this temporal scheme and pencil are already implemented in VPM for the diffusion of vorticity. The figure 1.3 summarizes the time integration scheme of VPM with this addition.

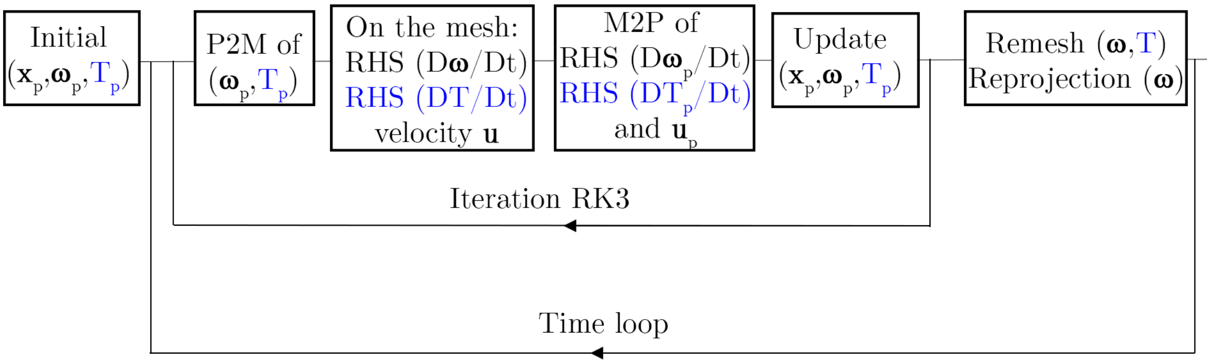

Figure 1.3: Summary of the VPM scheme of integration with the addition of the convection and explicit diffusion of temperature (blue).

However, as a goal of this work is to simulate flow at a low Prandtl number, it is unmanageable to use an explicit solver for the diffusion of temperature. The time step imposed by the numerical stability constraint (section **??**) would be very small and therefore, the computational cost (resources and time) would be very high.
So the choice was made to use the Euler implicit scheme. By being an implicit scheme, it does not impose any constraint on the time step. This one is then only determined by VPM just as if there is no temperature field. For the equation (1.7), at time n, the Euler implicit reads

$$\frac{T^{n+1} - T^n}{\Delta t} = \alpha \nabla^2 T^{n+1} \tag{1.17}$$

where T^n is the temperature after n time steps. Introducing

$$\Delta T^n = T^{n+1} - T^n \tag{1.18}$$

an Helmholtz equation for ΔT^n can be derived

$$\frac{\Delta T^n}{\Delta t} = \alpha \nabla^2 \underbrace{\left(T^{n+1} - T^n\right)}_{\Delta T^n} + \alpha \nabla^2 T^n \tag{1.19}$$

$$\nabla^2 \left(\Delta T^n\right) - \frac{1}{\alpha \Delta t} \Delta T^n = -\nabla^2 T^n. \tag{1.20}$$

Such equation can be solved with the use of FFT.
Taking the 3-D Fourier transform of the equation (1.20) leads to

$$-\left(k_x^2 + k_y^2 + k_z^2\right) \mathcal{F}\left(\Delta T^n\right) + \lambda^2 \mathcal{F}\left(\Delta T^n\right) = \mathcal{F}\left(-\nabla^2 T^n\right) \tag{1.21}$$

$$\Rightarrow \mathcal{F}\left(\Delta T^n\right) = \frac{\mathcal{F}\left(\nabla^2 T^n\right)}{k_x^2 + k_y^2 + k_z^2 - \lambda^2} \tag{1.22}$$

where $\lambda^2 = \frac{i^2}{\alpha \Delta t}$. Then ΔT^n is calculated by taking the inverse Fourier transform. Taking the Fourier transform on a discretized mesh implies periodicity in all the directions. To neutralize this effect, the solver used is a Fourier-based solver that can combine unbounded and periodic directions developed for the computation of the velocity field in VPM [3]. It revolves around splitting the Laplacian accordingly to the periodic and unbounded directions.
In this case the domain is periodic in two directions and unbounded in one, so the equation (1.20) becomes

$$\nabla_y^2 \left(\Delta T^n\right) + \frac{\partial^2 \left(\Delta T^n\right)}{\partial x^2} \frac{\partial^2 \left(\Delta T^n\right)}{\partial z^2} + \lambda^2 \Delta T^n = -f^n \tag{1.23}$$

where $f^n = \nabla^2 T^n$. A Fourier transform of the equation (1.23) in the periodic directions leads to

$$\nabla_y^2 (\widetilde{\Delta T^n}) + \left(\lambda^2 - k_x^2 - k_z^2\right) \widetilde{\Delta T^n} = -\tilde{f} \tag{1.24}$$

where $\widetilde{\Delta T^n}(k_x, y, k_z)$ is the Fourier transform of ΔT^n along the x-direction and the z-direction and $\tilde{f}(k_x, y, k_z)$ is the Fourier transform of f also along the x-direction and the z-direction. For a given couple (k_x, k_z), the equation (1.24) is an one-dimensional unbounded Helmholtz equation and the solution is given by the convolution between the source term and the fundamental solution to the Helmholtz equation

$$G_{\sqrt{\lambda^2-k_x^2-k_z^2}}(y) = \frac{ie^{i\sqrt{\lambda^2-k_x^2-k_z^2}y}}{2\sqrt{\lambda^2-k_x^2-k_z^2}}. \tag{1.25}$$

Therefore, the solution of the equation (1.24) is

$$\widetilde{\Delta T^n} = G_{\sqrt{\lambda^2-k_x^2-k_z^2}} \star \tilde{f} \tag{1.26}$$

but the implementations of algorithm capable of doing a convolution are much slower than the implementations of FFT.
Thus, the following procedure is used to emulate an unbounded domain:

1. the original domain is doubled in the unbounded direction (y-direction) and the source term is padded ($\tilde{f}$) with zeros. This zero padding eliminates the effect of periodic images;
2. the Helmholtz equation is solved in the wave space of the extended domain

   $$\widehat{\Delta T^n} = \hat{G}_{\sqrt{\lambda^2-k_x^2-k_z^2}}\hat{f} \tag{1.27}$$

   with $\hat{G}_{\sqrt{\lambda^2-k_x^2-k_z^2}}$ the Fourier transform in the y-direction of $G_{\sqrt{\lambda^2-k_x^2-k_z^2}}$ and $\hat{f}$ the Fourier transform in the y-direction of $\tilde{f}$. After the inverse transform in the y-direction the upper halves of the results are discarded.

Using this procedure $\widetilde{\Delta T^n}$ is computed and then the inverse Fourier transform is taken along the x-direction and the z-direction to compute ΔT^n on the mesh. Afterwards, this is interpolated onto the particles to do the time integration of the strength $\alpha_{T,p}$ of the particles (equation (1.15)).

A particular attention needs to be paid to the computation of the source term of the equation (1.24). Indeed, a Laplacian can either be computed in the physical domain or in the wave space. However, it is important to remember than before taking the Fourier transform in the y-direction the domain is doubled and the signal is zero padded. This implies that the support of the Fourier-transformed field is inside of the computational domain which is the case for the temperature field because it tends to two different values in the unbounded direction. Therefore, the Laplacian of T^n needs to be computed in the physical domain before taking the Fourier transform. This is operation is performed using the same fourth order centered cross pencil described previously. The procedure to ensure that the support of the Laplacian is contained in the computational domain is explained in the section 1.4.

To summarize, the procedure used to diffuse involves 4 steps:

1. interpolating the temperature on the mesh;
2. computing the Laplacian of the temperature;
3. solving the equation (1.20) by taking the Fourier transforms in both periodic directions then in the unbounded direction using the method described previously and then taking the inverse Fourier transforms in the inverse order;
4. interpolating the right hand side of the equation (1.15) on the particles and integrating their strength.

These operations are implemented after the third order low storage Runge-Kutta scheme as it is shown on the figure 1.4.

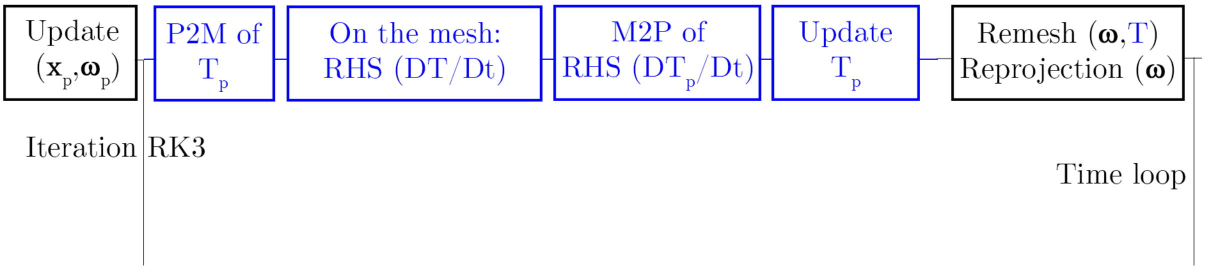

Figure 1.4: Summary of the implicit scheme of diffusion of temperature (blue).

It can be verified that for a well posed diffusion problem ($\alpha > 0$) the fundamental solution of equation (1.24) is always real. Using the definition of λ, the equation (1.25) becomes

$$G_{\sqrt{\lambda^2-k_x^2-k_z^2}}(y) = \frac{ie^{i^2\sqrt{-\lambda^2+k_x^2+k_z^2}y}}{2i\sqrt{-\lambda^2+k_x^2+k_z^2}} \tag{1.28}$$

$$= \frac{e^{-\sqrt{-\lambda^2+k_x^2+k_z^2}y}}{2\sqrt{-\lambda^2+k_x^2+k_z^2}} \tag{1.29}$$

where the radical of the square root is always positive and thus G is real. As λ depends on the time step and that VPM computes the time step dynamically, at each iteration the function G needs to be recalculated.

1.4 The dynamic domain

1.4.1 Management of the size of the domain

An advantage of VPM is that it is able to use a computational domain whose size can vary over time. Doing so, it reduces the computation time by keeping the domain as compact as the support of vorticity throughout the simulation. This feature can be used for 1, 2 or 3 unbounded directions. For this work, that function is used in the unbounded direction. VPM implements the following procedure:

1. it computes the dimensions of the smallest box that includes all particles whose vorticity magnitude is higher than a configurable threshold value;
2. if in an unbounded direction, one side of the box is closer to the boundary of the domain than the minimum distance allowed (configurable in the code), points are added to the mesh in that direction.

This procedure ensures that only particles whose vorticity is lower than the threshold are allowed to leave the domain. On the figure 1.5, the red and green circles represent particles whose magnitude of vorticity is respectively higher and lower than the threshold. The solid lines serve as the boundaries of the domain, the long dashed lines show the side of the box embodying all the particles that may not exit the domain and the short dash lines delimit the part of the domain where any particle is allowed without the need to extend the domain. So, in this example, the domain needs to grow because there are red particles outside of the domain delimited by the short dashed lines.

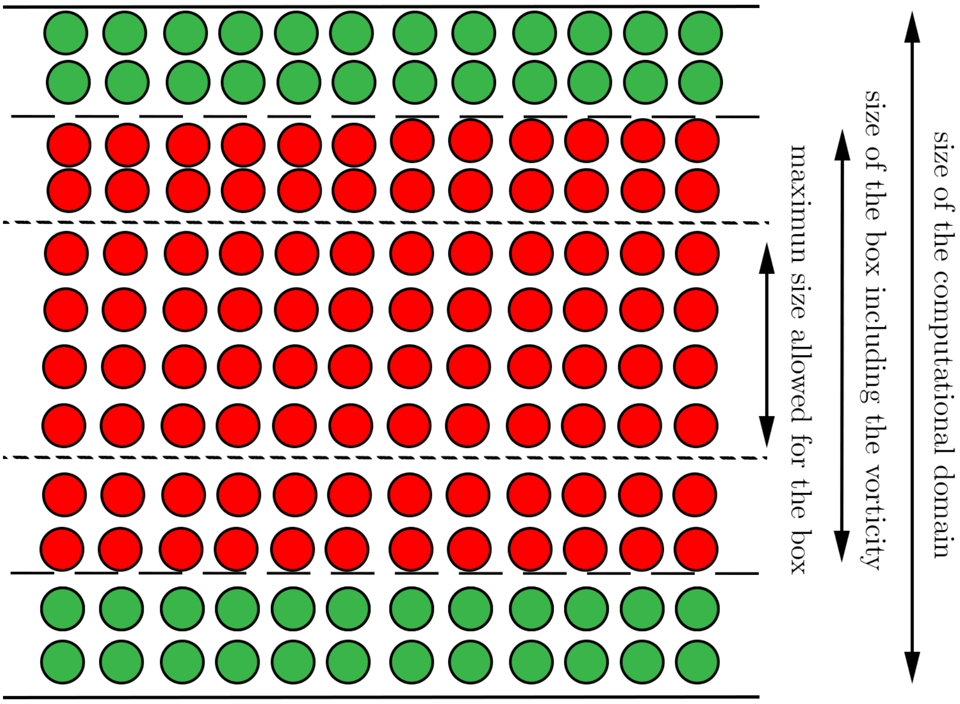

Figure 1.5: Illustration of the computations to manage the size of an unbounded domain based on the vorticity.

As the temperature was in VPM, the management of the dynamic domain needs to also take into account the temperature field. Indeed, the solver used to diffuse the temperature (section 1.3.2) can only give valid results if the support of the temperature Laplacian is contained in the domain. Beyond, for a low Prandtl number mixing layer, the size of the temperature mixing layer will likely be greater than the size of the vorticity shear layer and the growth of the mixing layer will force the growth of the domain.

Therefore, to ensure that outside of the computational domain the temperature Laplacien is zero, the same principle than the one presented for the vorticity is implemented.
To this extent, the goal of the procedure performed is to guarantee that when a particle whose temperature is not the same as the temperature outside of the mixing layer comes close to a boundary of the domain, it grows.

So, the size of the box containing all the particles whose $|T_p - T_{out}|$ is greater than a threshold needs to be computed (T_{out} is the temperature outside of the mixing layer). Thus, it is needed to determine with which side of the mixing layer the temperature of each particle should be compared and this choice is made in function of the coordinate of the particle in the unbounded direction. If it is higher or lower than the mid-point of the computational domain, T_{out} becomes respectively the temperature above the mixing layer or below.

Once the size of the box is computed, if a side of the box is closer to the boundary of the domain than the minimum distance allowed, points are added to the mesh. After adding the mesh points, for the vorticity field there is nothing more to do because the domain is extended on a zone with zero vorticity. But for the temperature field, the domain expands on regions with a certain temperature, thus, new particles need to be created to keep this information.

So, when the domain grows, the vorticity and the temperature of the particles are interpolated on the mesh. In the unbounded direction, the temperature of the mesh points outside the previous domain is overwritten with the value of the temperature outside of the mixing layer. It is also needed to overwrite a small layer of points in the old domain. The reason why is explained later in this dissertation. Finally, both fields are interpolated back onto a new set of particles. This ensures that the particles close to the borders of the domain always have the same temperature and therefore, the temperature Laplacien is included in the domain.

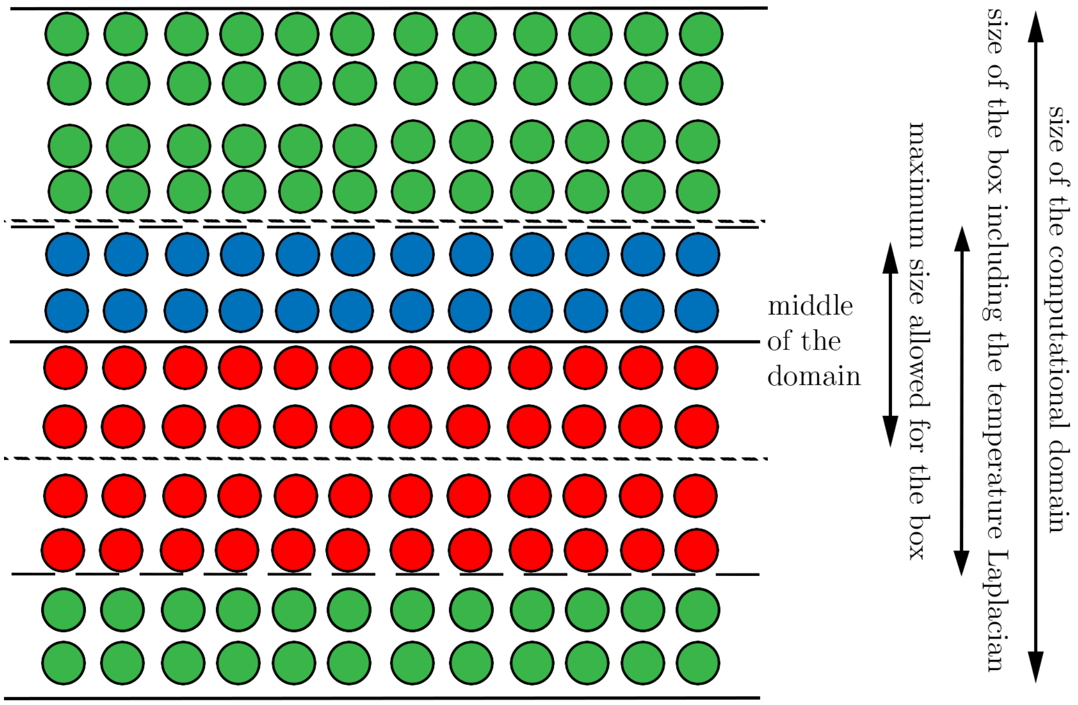

Figure 1.6: Illustration of the computations to manage the size of an unbounded domain based on the temperature.

An example of this is shown on the figure 1.6. The green circles represent particles whose temperature matches the outside of the mixing layer. The blue circles illustrate particles that are compared with the temperature above the mixing layer and that the absolute value of the difference is higher than the threshold. The red circles also represent the particles whose absolute value of the difference is higher than the threshold but compared with the temperature below the mixing layer. In this example, the domain needs to grow towards the bottom because there are particles whose temperature is different from the temperature outside of the shear layer close to the boundary of the domain.

1.4.2 Handling of the points at a boundary of the domain

Previously, it was said that it is needed to overwrite a small layer of the temperature field next to the boundaries of the domain in the unbounded direction. This has to be done every time the temperature is interpolated from the particles to the mesh. Indeed, the temperature field after interpolation is not exact near the boundaries of the domain in the unbounded direction. The figure 1.7 shows why and for the sake of simplicity, the phenomenon is explained in one dimension.

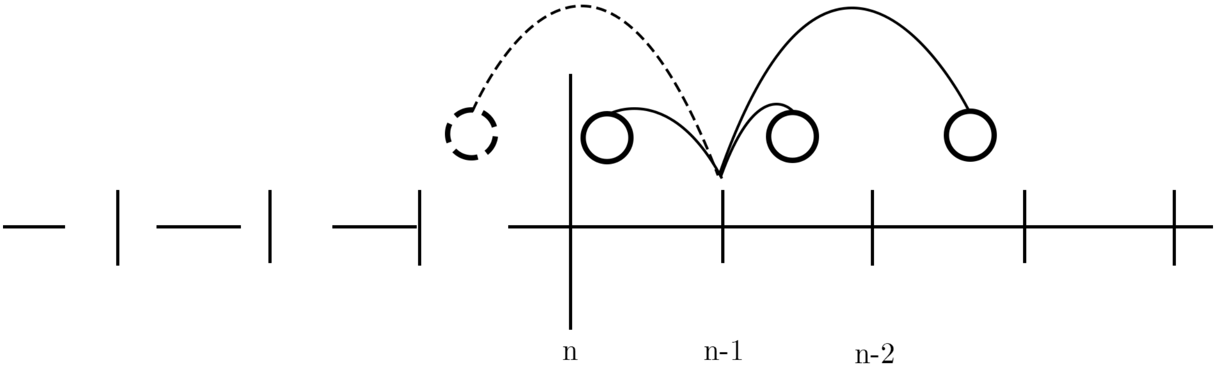

Figure 1.7: Illustration of the P2M operation near the boundary of the computational domain for a unidimensional problem.

The long vertical line represents the left frontier of the domain. So, as M_4' is used for the interpolation, a mesh point at position x is influenced by particles in the range $[x - 2h, x + 2h]$ where h is the mesh resolution. By example, for the mesh point $n - 1$, this range extends out of the computational domain. Therefore, the temperature of that mesh point is not correctly computed by the interpolation because the particle outside the domain (the dashed one in the figure) does not exist. Hence, it is needed to force the temperature of a small layer of at least 2 mesh points on the boundary. By safety, in the code, a layer of 4 mesh points is overwritten in case, since the last remeshing operation, two particles would have left the domain.

A final attention needs to be paid when computing the temperature Laplacien. Using a fourth order pencils means that on the sides of the domain 2 ghost points are needed. In the periodic directions, these are actually points from the other side of the domain because of the periodicity. On the other hand, in the unbounded direction, the value of these ghost points needs to be forced to the value outside of the mixing layer.

1.5 Initial conditions

The flow studied for this work is a time-developing mixing layer. The domain used is then periodic in the x-direction (the streamwise direction) and in the z-direction. It is unbounded in the y-direction.
The initial condition for the velocity profile is an hyperbolic profile

$$u(x, y, z) = \frac{\Delta u}{2} \tanh\left(\frac{2y}{\delta_{w,0}}\right) \tag{1.30}$$

where u is the streamwise velocity, Δu is the velocity difference through the shear layer and

$$\delta_{\omega,0} = (\Delta u) \,/\, \left.\frac{\partial u}{\partial y}\right|_{max} \tag{1.31}$$

is the initial thickness of the vorticity. A perturbation is added to velocity field. This is explained in section 3.2.
The initial temperature profile has the same thickness as the velocity profile

$$T(x, y, z) = \frac{T_1 + T_2}{2} + \frac{T_1 - T_2}{2} \tanh\left(\frac{2y}{\delta_{w,0}}\right) \tag{1.32}$$

where T_1 and T_2 are the temperatures respectively above and below the mixing layer. We define $\Delta T = T_1 - T_2$ the temperature difference across the mixing layer. No perturbation was added to the temperature field. The figure 1.8 shows those initial profiles.

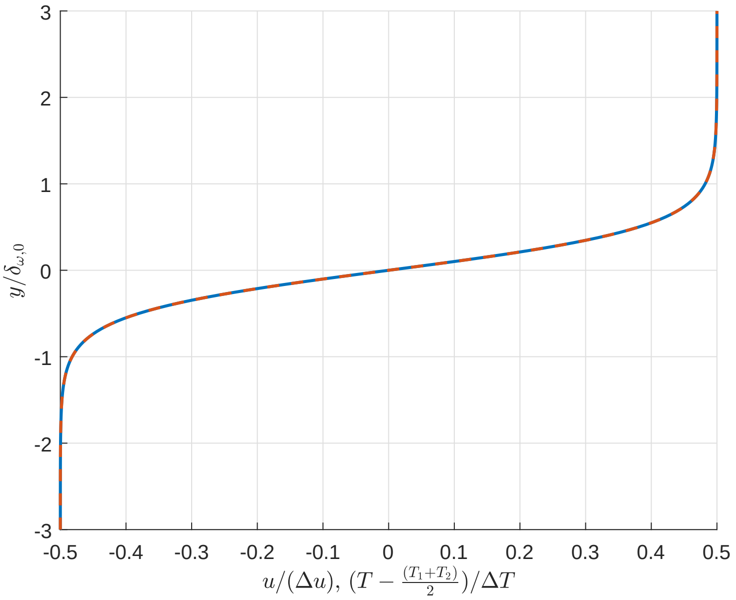

Figure 1.8: Initial streamwise velocity (blue solid line) and temperature (red dashed line) profile.

The initial condition needs to be given in vorticity and not in velocity thus the initial condition implemented in the code is

$$\omega_z(x, y, z) = -\frac{du}{dy} = -\frac{\Delta u}{\delta_{w,0}} \operatorname{sech}^2\left(\frac{2y}{\delta_{w,0}}\right) \tag{1.33}$$

where w_z is the vorticity in the z-direction. This profile is shown in the figure 1.9.

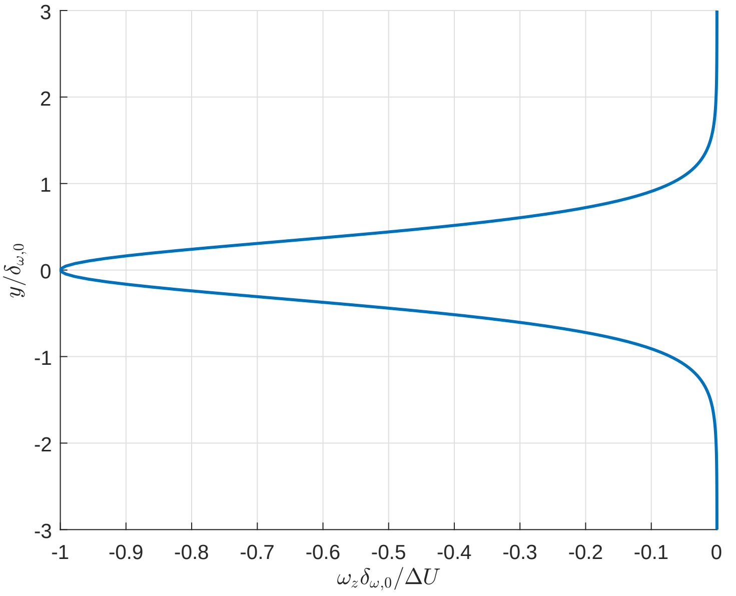

Figure 1.9: Initial vorticity profile.

Such flow is unstable and therefore the small perturbation added will turn the shear layer turbulent. The most unstable mode (k_{KH}) in the streamwise direction is the Kelvin-Helmholtz instability and is given by [4]

$$k_{KH}\delta_{\omega,0} \approx 0.44 \Rightarrow \lambda_{KH} = 7.15\delta_{\omega,0} \tag{1.34}$$

with λ_{KH} the wave length of the Kelvin-Helmholtz instability.

Chapter 2

Validation of the implicit diffusion scheme

Before using the implicit diffusion solver in an application, such as the time-developing mixing layer, it is needed to validate the exactness of the results produced by the method.
To this extent, the implicit solver is used on a test scenario and the results are compared with an analytical solution.

2.1 Description of the test scenario

Because the solver is implemented for a domain periodic in two directions while unbounded in the last one, the test scenario chosen is an one-dimensional diffusion problem

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial y^2} \tag{2.1}$$

with a hyperbolic profile as the initial condition (figure 1.8)

$$T_0(y) = \frac{T_1 + T_2}{2} + \frac{T_1 - T_2}{2} \tanh\left(\frac{2y}{\delta_{\omega,0}}\right). \tag{2.2}$$

In an unbounded domain, the solution can be found analytically by convolution

$$T_e(y,t) = G(t) * T_0 = \int_{-\infty}^{\infty} G\left(y-s,t\right) T_0\left(s\right) ds \tag{2.3}$$

where

$$G(y-s,t) = \frac{1}{\sqrt{4\pi\alpha t}} e^{\frac{-(y-s)^2}{4\alpha t}} \tag{2.4}$$

is the fundamental solution of the equation (2.1).

2.2 Stability analysis

This first validation test consists of running the implicit solver with different time steps and verifying that the solution is always numerically stable.

Comparisons between the exact solution and the numerical results at $t' = \frac{t\alpha}{\delta_{\omega,0}^2} = 25$ can be found in the figure 2.1.
For this test, in each periodic direction, the mesh is 4 points wide to simulate a domain with only one dimension while in the unbounded direction, the size is initially $6\delta_{\omega,0}$.

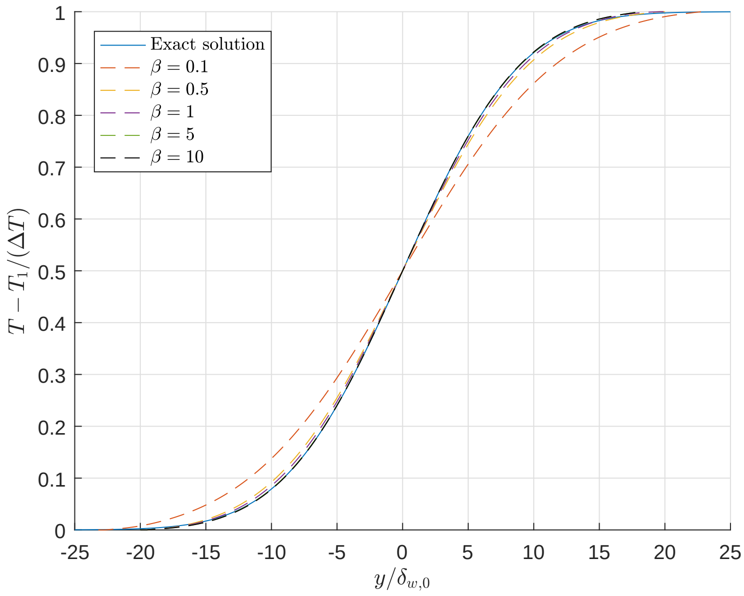

Figure 2.1: Analytical solution and results obtained for different β at $t' = 25$.

All the simulations run on the same mesh ($h = 0.1\delta_{\omega,0}$) but they differ by their time step and hence, they also differ by the factor

$$\beta = \frac{\alpha \Delta t}{h^2} \tag{2.5}$$

where Δt is the time step. For an explicit time integration scheme, such as the third order Runge-Kutta scheme, β must be smaller than $\beta_{max} = \mathcal{O}(0.1)$ to obtain stable results. On the other hand, an implicit time integration scheme does not have that constraint. So, the solver implemented can be run at $\beta > \mathcal{O}(0.1)$ and the results will be stable.

The first observation is that the method, as expected, is stable for all the time steps tested. The second observation, however, seems less logic. Indeed, the results obtained at the lowest β are also the results whose error is the largest. Theoretically, the error of a numerical method decreases when the time step gets smaller but here, the opposite is observed. When the time step becomes smaller and the mesh size does not change (so β gets smaller), this solver over-diffuses. Consequently, it will be important to monitor during the simulations that the time step imposed by VPM is not too small as the solver would over-diffuse the temperature field.

This could be explained by the fact that when β is small, the Green function of the equation (1.24) is not well approximated by its discretized version.
Let us take, for example to explain, the mode 0 in both periodic directions. Introducing $k_x = k_z = 0$ in the equation (1.25), it becomes

$$G_{\sqrt{\lambda^2}}(y) = \frac{e^{-\sqrt{-\lambda^2}y}}{2\sqrt{-\lambda^2}} \tag{2.6}$$

where $\lambda^2 = \frac{i^2}{\alpha \Delta t}$. Therefore, the equation can be rewritten as

$$G_{\sqrt{\lambda^2}}(y) = e^{-\frac{y}{\sqrt{\alpha \Delta t}}} \frac{\sqrt{\alpha \Delta t}}{2}. \tag{2.7}$$

In the code, the Green function is discretized on the mesh. So let $y_n = nh$ where n is a positive integer. The equation reads as follow

$$G_{\sqrt{\lambda^2}}(y_n) = e^{-\frac{nh}{\sqrt{\alpha \Delta t}}} \frac{\sqrt{\alpha \Delta t}}{2} \tag{2.8}$$

$$G_{\sqrt{\lambda^2}}(y_n) = e^{-\frac{n}{\sqrt{\beta}}} \frac{\sqrt{\alpha \Delta t}}{2}. \tag{2.9}$$

We can see that β appears on the denominator of the exponent. So when β is small, the exponential tends very quickly below the machine precision and then, is replaced by zero. Thus, the results given by the method are inaccurate as seen in figure 2.1. In the very extreme case where β is close to zero, the value of the discretized Green function could be $\frac{\sqrt{\alpha \Delta t}}{2}$ at $y_n = 0h$ and 0 everywhere else.

By looking at the figure 1.4, it can be seen that, at $t' = 25$, the length of the computational domain in the unbounded direction is over $40\delta_{\omega,0}$. This shows that the size of the domain is changing in line with the support of the temperature Laplacian during the simulation. Since the solution is valid, the growth of the domain was done without influencing the numerical solution. This confirms the choice made in the section 1.4 of imposing the temperature on layers at the borders of the computational domain in the unbounded direction.

2.3 Convergence analysis

The second test consists in a convergence analysis at constant β. A convergence analysis provides the rate of convergence of a numerical method. By example, the error for the Runge-Kutta 3 scheme is $\mathcal{O}(\Delta t^3)$. Therefore, we say that its rate of convergence is of the third order. The rate of convergence is given by the slope of the error in function of the time step if both axis use log scales.

The figure 2.2 shows the result of the convergence analysis for the Euler implicit scheme implemented. For this test the value of β is to 10 and the error is calculated as the least square error

$$R(t) = \sqrt{h \sum_n \left(T(t, y_n) - T_e(t, y_n)\right)^2} \tag{2.10}$$

at $t'_f = 0.032$.

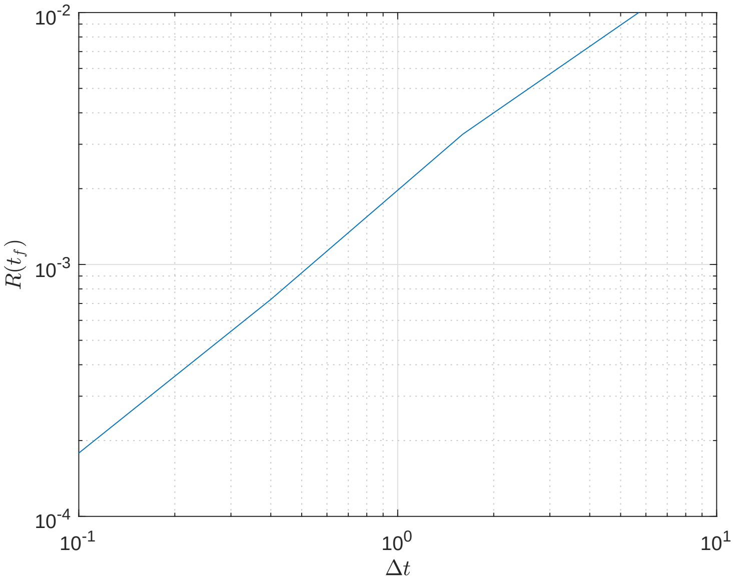

Figure 2.2: Convergence analysis of the Euler implicit scheme at $\beta = 10$.

The figure 2.2 shows that, as expected using the Euler implicit scheme, the method has a rate of convergence of the first order with respect to the time step ($R = \mathcal{O}(\Delta t)$).

In conclusion, the two tests realized confirm:

- the stability of the method and the absence of constraint on the time-step;
- the exactness of the method and the first order convergence of the Euler implicit scheme.

Therefore, the implicit diffusion solver can be used to simulate the diffusion for a low Prandtl number time-developing mixing layer.

Chapter 3

Analysis of the 3D mixing layer

3.1 Description of the simulations

The simulations performed are described hereafter. Each simulation is identified by an acronym used in the remainder of the document.

- **ML1** : first simulation at high Prandtl number.
  The Reynolds number is $Re = \frac{\Delta u \delta_{\omega,0}}{\nu} = 380$ and the Prandtl number is $Pr = 1$. The dimensions of the domain in the periodic directions are $9.5\lambda_{KH} \times 6.15\lambda_{KH}$ with λ_{KH} the wave length of Kelvin-Helmholtz instability (see the equation (1.34)). It corresponds to 680×440 points and the mesh size $h = 0.1\delta_{\omega,0}$. The initial perturbation is characterized by $u_{rms} = 0.01\Delta u$ like for every simulation except **ML2**.
- **ML2** : adjustment of the initial perturbation.
  All the parameters are the same as ML1 but the perturbation is different (see section 3.2)
- **ML3** : simulation at low Prandtl number.
  The dimensions of the domain and the number of points is identical to **ML1** but the Reynolds number is increased to $Re = 700$ and the Prandtl number is reduced to $Pr = 10^{-2}$.
- **ML4** : enlargement of the domain.
  The Reynolds number and the Prandtl number are the same than **ML3** but the dimensions of the domain in the periodic directions are $16\lambda_{KH} \times 10\lambda_{KH}$ with λ_{KH} the wave length of Kelvin-Helmholtz instability (see the equation (1.34)). The number of points is changed to 1144×760 in order to keep the same mesh size.
- **ML5** : adjustment of u_{rms}.
  Every parameter is identical to ML4 except for $u_{rms} = 0.02\Delta u$.

For ML1 and ML2, the diffusion of temperature is done explicitly because the Prandtl number is high. For all the others simulations, the diffusion of temperature is done implicitly using the solver developed for this work.

3.2 Creation of the initial perturbation

To force the flow to transition from the laminar to the turbulent regime, it is needed to add a small perturbation to the initial profile of velocity. Because a shear flow is inherently unstable, any small perturbation will grow in size and eventually become turbulence. Therefore, there are multiple types of perturbation possible: mathematical formulas such as sine waves, white noise, et cetera.

For this work, the choice is made to generate this perturbation in the wave space so its spectrum can be modulated. The procedure used to create the procedure is the following:

1. using a random number generator to create vorticity in the wave space ($\hat{\omega}\left(k_x, k_y, k_z\right)$), computing $\hat{\mathbf{u}}\left(k_x, k_y, k_z\right)$ and calculating the velocity field using FFT;
2. clipping the velocity field in the physical domain to ensure that the perturbation is initially only in the shear layer and setting the total momentum to zero;
3. computing the curl of the velocity, rescaling it to achieve the desired u_{rms} and adding it to the initial shear layer vorticity.

As the noise is initially generated in the wave space, it is possible to modulate its spectrum using a function of the wave number

$$\hat{\mathbf{u}}_{mod}\left(k_x, k_y, k_z\right) = \hat{\mathbf{u}}\left(k_x, k_y, k_z\right) f\left(k_x, k_y, k_z\right) \tag{3.1}$$

where f is the desired spectrum.
During the course of this work, two different functions were used:

1. the function constant $f\left(k_x, k_y, k_z\right) = 1$ to produce a non tuned perturbation;
2. the function $f\left(k_x, k_y, k_z\right) = |\mathbf{k}|^{-5/3}$ where $\mathbf{k} = \left(k_x, k_y, k_z\right)$ to produce a tuned perturbation.

The figure 3.1 shows the difference between the spectrum obtained with both functions. By using the modulated noise the energy contained in the large scales is greater than the energy contained in the small scales by several decades. On the opposite, if the spectrum of the noise is not modulated, the energy is almost distributed equally. This leads to a lot of the energy of the fluctuations being dissipated very quickly and thus, the laminar regime will last longer as it can be seen on the figure 3.2.

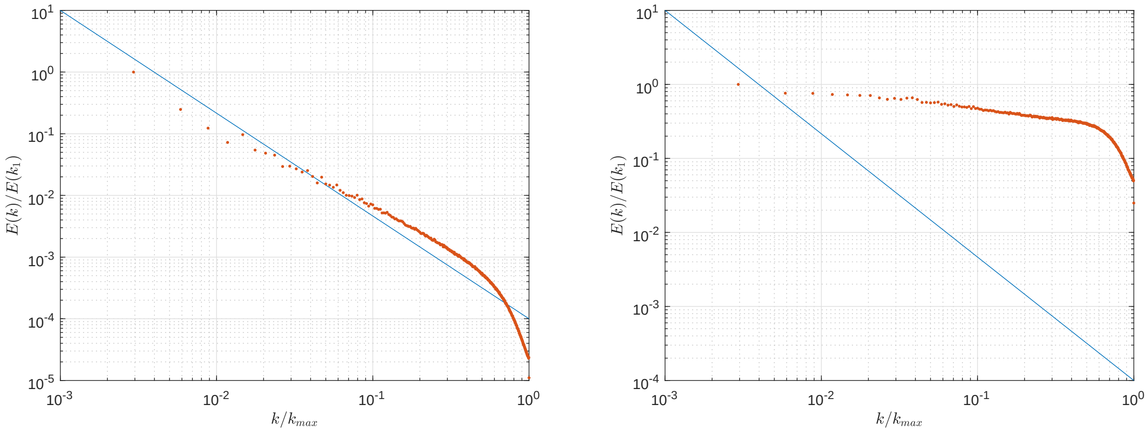

Figure 3.1: Spectrum in the streamwise direction of the initial condition (red) and $|\mathbf{k}|^{-5/3}$ (blue) (right, the spectrum is tuned and left, the spectrum is not tuned).

Out of the 5 simulations, ML1 is the only simulation using a non tuned perturbation.

3.3 Analysis of the results of ML1 and ML2

The results are compared to the results obtained by Watanabe et al. [1] and by Duponcheel and Bartosiewicz [5] at the same Reynolds and Prandtl numbers (Schmidt number).

Regarding the dimensions in the periodic directions of the domain, these are approximately the same as the dimensions used by Duponcheelsand Bartosiewicz [5]. However, compared to the simulation done by Watanbe et al. [1], the dimension is approximately the same in the streamwise direction but twice bigger in the spanwise direction.

It is important to note that the initial fluctuations are generated differently in the simulations of Watanabe et al. [1] and of Duponcheel and Bartosiewicz [5].
They used a diffused random white noise that is characterized by the desired length scale:

- Watanabe et al. [1] used a perturbation with an integral length scale of $\frac{\delta_{\omega,0}}{4}$ and $u_{rms} = 0.01\Delta u$.
- Duponcheel and Bartosiewicz [5] characterized the perturbation by an integral length scale $L = 0.25\delta_{\omega,0}$ or $L = 4.25\delta_{\omega,0}$ and $u_{rms} = 0.01\Delta u$.

In this comparison, the thickness of the momentum shear layer and the thickness of the temperature mixing layer are computed as follow

$$\delta_\theta = \frac{1}{(\Delta u)^2}\int_{-\infty}^{\infty} (u_1 - \langle u\rangle)(\langle u\rangle - u_2)\,dy \tag{3.2}$$

$$\delta_T = 2\left(y|_{\langle T\rangle = T_1 - 0.25\Delta T} - y|_{\langle T\rangle = T_2 + 0.25\Delta T}\right) \tag{3.3}$$

where $\langle * \rangle$ is the space-average over the periodic directions of the quantity $*$, u_1 and u_2 are the velocities respectively above and below the shear layer. The time is normalized using the velocity difference and the initial vorticity thickness of the shear layer

$$t^* = \frac{t\Delta u}{\delta_{\omega,0}}. \tag{3.4}$$

3.3.1 Comparison of the time-history

The figure 3.2 shows the comparison of the thickness of the velocity shear layer.

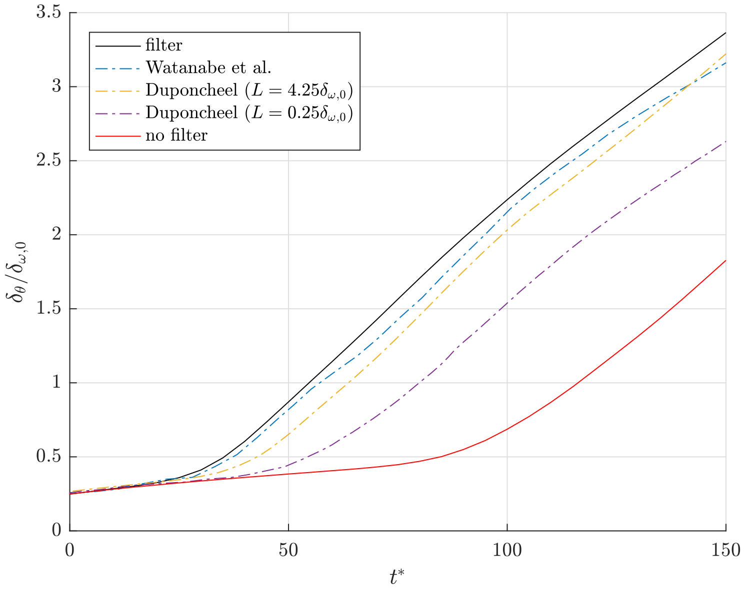

Figure 3.2: Time-history of the velocity shear layer thickness.

It confirms what has previously been stated about the perturbation. The time-history of the shear layer is strongly dependent on the spectrum of the initial perturbation. For ML1, the laminar regime lasts up to $t^* \cong 70$ then the growth rate increases. However, for ML2, the laminar regime only lasts up to $t^* \cong 30$ before the growth rate of the shear layer increases then decreases to become linear after $t^* \cong 100$. In both cases, after the laminar regime, the shear layers widen at the same rhythm. The time-history of the thickness of the simulation ML2 is very close to the ones obtained by Watanabe et al. [1] and by Duponcheel and Bartosiewicz [5] for $L = 4.25\delta_{\omega,0}$.

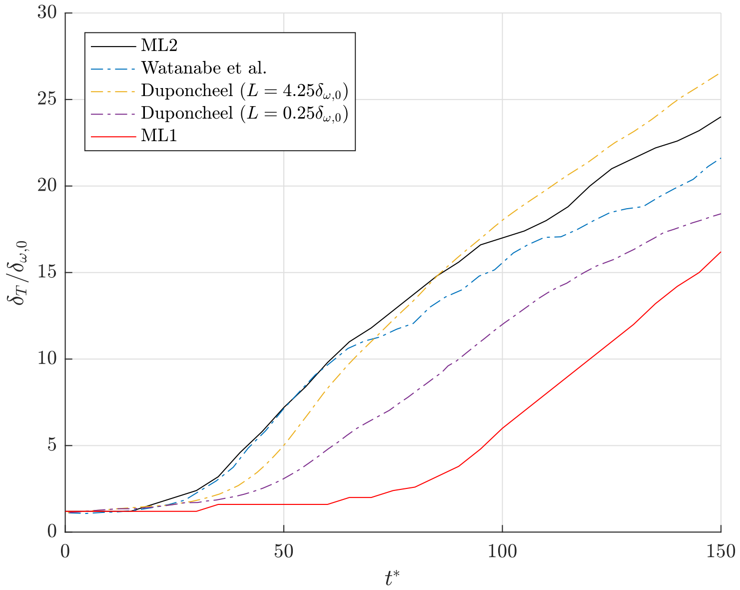

Figure 3.3: Time-history of the temperature mixing layer thickness.

The figure 3.3 shows the comparison of the thickness of the temperature mixing layer. It shows that, at $Pr = 1$, the thickness of the temperature is directly impacted when the laminar regime ends. However, after the laminar regime, the results from ML2 do not exactly match either the results by Watanabe et al. [1] or by Duponchee and Bartosiewiczl [5] but the growth rate of the temperature thickness is very similar for the three simulations.

3.3.2 Visualization of the simulations

The difference at $t^* = 100$ between the two simulations can also be seen by a visualization of the temperature. The figures 3.4 and 3.5 show both a vertical slice of the temperature field. In the figure 3.5, it can be observed that the turbulent mixing already took place as there many small scales present in the flow. Instead, for the other simulation, the flow is still dominated by the Kelvin-Helmhotlz instability and barely any turbulent mixing already occurred.

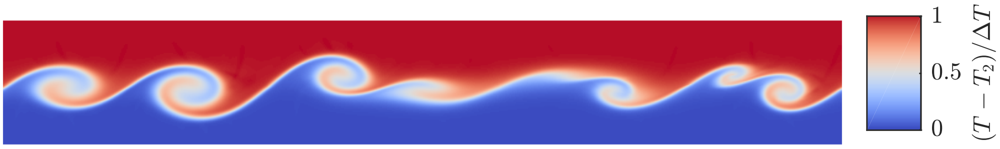

Figure 3.4: Vertical slice of the temperature field at $t^* = 100$ (ML1).

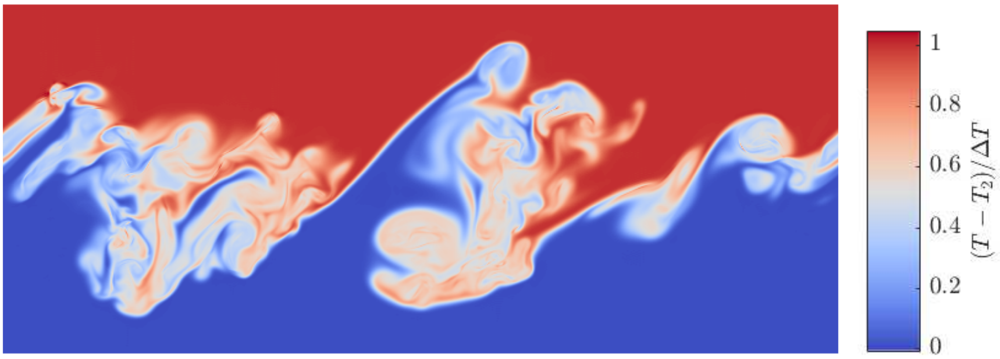

Figure 3.5: Vertical slice of the temperature field at $t^* = 100$ (ML2).

By observing the figure 3.5, it appears that there is a slight overshoot in the temperature. This comes from the fact that the interpolation scheme used for the P2M and M2P operations is M_4'. Indeed, the scheme creates overshoot/undershoot in regions where the flow stretches a steep temperature gradient. This problem was not addressed during the course of this work because at low Prandtl, it does not appear since steep temperature gradients are quickly diffused.

3.3.3 Comparison of the mean profiles

The figures 3.6 and 3.7 show the comparison between the average streamwise velocity profile ($\langle u \rangle$) and the temperature profile ($\langle T \rangle$) obtained at $t^* = 144$ for the two simulations and the data from Watanabe et al. [1].

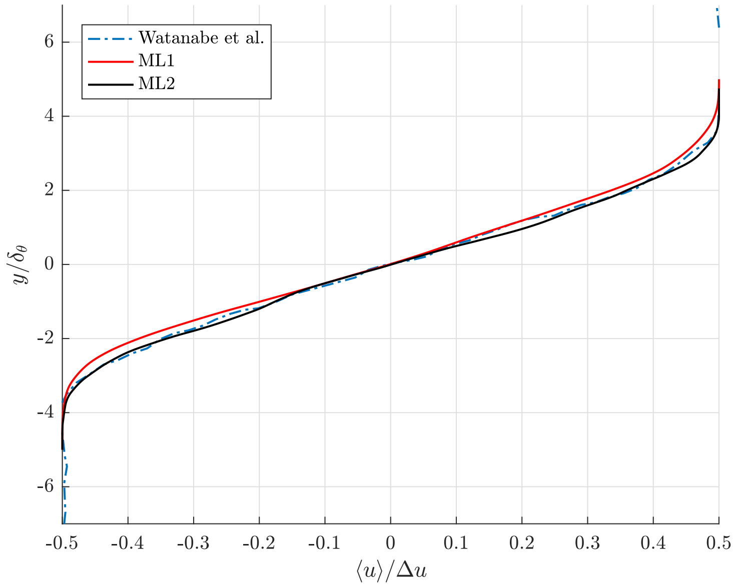

Figure 3.6: Mean velocity profile.

The figure 3.6 shows that the average velocity profile (ML1) fit reasonably well the data from Watanabe et al. [1]. The difference can be explained as a consequence the flow still not fully transitioned to turbulence. Yet, the average velocity profile (ML2) fits very well the data from Watanabe et al. [1]. This tends to show that the use of noise with a modulate spectrum is better to produce the turbulence. Similar conclusions can be drawn from the figure 3.7. Indeed, the average temperature profile (ML1) presents a double inflection and slightly deviates from the data obtained by Watanabe et al. [1]. In the other hand, the average temperature profile

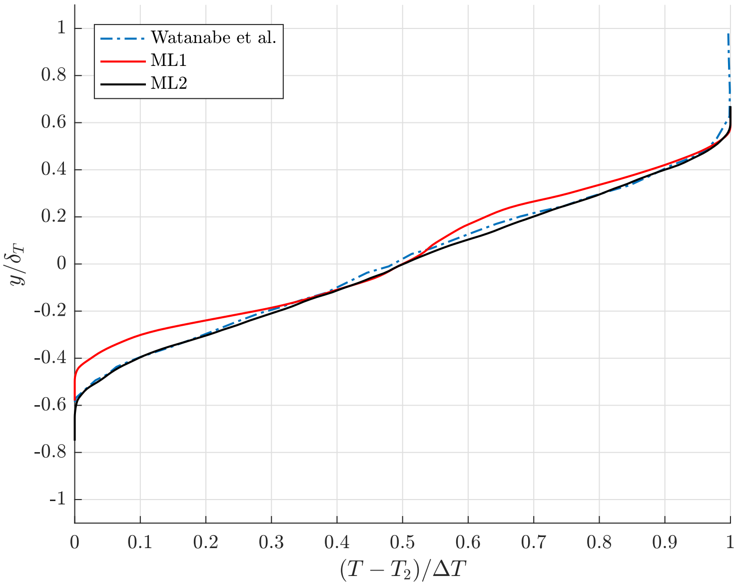

Figure 3.7: Mean temperature profile.

(ML2) fits well the reference curve from Watanabe et al. [1] with only very little disparities. Consequently, it is well-advised to use a modulated spectrum perturbation because the turbulent mixing is activated more quickly.

Therefore, we can draw the conclusion that the type of noise has an influence on the time-history of the shear layer thickness. Also, it is better to use a perturbation with a tuned spectrum for further simulations for two reasons. Firstly, the transition to turbulence happens quickly letting less time for the diffusion to happen in the laminar regime. Secondly, the perturbation does not really let the Kelvin-Helmholtz instability fully develop before becoming 3D turbulent and thus the temperature profile does not exhibit the double inflection of the Kelvin-Helmholtz instability. For these reasons, the initial fluctuations added to all the following simulations have a modulated spectrum.

3.4 Analysis of the results of ML3

The results of this simulation are compared with the results obtained by Duponcheel and Bartosiewicz [6] at the same Reynolds and Prandtl numbers. However, the domain dimensions in the periodic directions are different. The dimensions used by Duponcheel and Bartosiewicz [6] are $132\delta_{\omega,0} \times 90.6\delta_{\omega,0}$ which corresponds approximately to $18.5\lambda_{KH} \times 12.5\lambda_{KH}$. As a reminder, the dimensions of the domain used for ML3 are $9.5\lambda_{KH} \times 6.15\lambda_{KH}$.
Another difference is that the perturbation used by Duponcheel and Bartosiewicz [6] is the sum of a random white noise of amplitude $0.01\Delta u$ and a sine wave of wavelength $8.5\delta_{\omega,0}$ of amplitude $0.01\Delta u$.

From now on, in order to compare the results, the thickness of the mixing layer is computed using the equation (3.2), the following formula

$$\delta_\alpha = y|_{\langle u\rangle=\alpha\frac{\Delta u}{2}} - y|_{\langle u\rangle=-\alpha\frac{\Delta u}{2}} \tag{3.5}$$

which defines the α-percent thickness for the velocity and the α-percent layer thickness for the temperature computed as follows

$$\delta_{T,\alpha} = y|_{\langle T\rangle = T_1 - (1-\alpha)\frac{\Delta T}{2}} - y|_{\langle T\rangle = T_2 + (1-\alpha)\frac{\Delta T}{2}}. \tag{3.6}$$

3.4.1 Comparison of the time-history

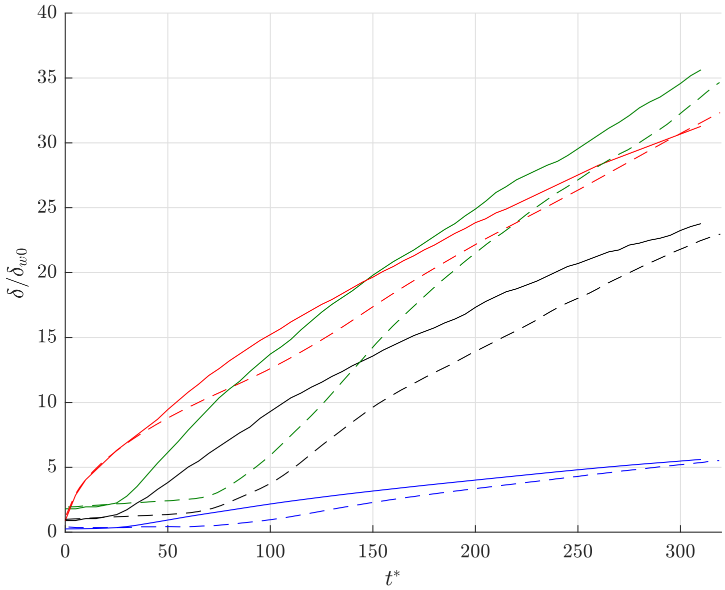

Figure 3.8: Time-history of the shear layer thickness from Duponcheel (dashed lines) and the simulation ML3 (solid lines): δ_θ (blue), $\delta_{T,75}$ (red), δ_{95} (green) and δ_{75} (black).

The figure 3.8 shows the time-history of the shear layer thickness. During the first part of the simulation the flow is laminar and the growth of the shear layer thickness is only due to the diffusion of either momentum or temperature. During the laminar regime, the thickness of the temperature expands much faster than the thickness of the velocity. For this simulation, the laminar regime ends at $t^* \cong 35$. However, the thickness of the temperature continues to widen only by diffusion up to $t^* \cong 50$. In fact, because the thickness of the temperature is much larger than the thickness of the velocity, the instabilities need to develop before impacting the expansion speed of the temperature mixing layer.

The first major difference with the simulation of Duponcheel and Bartosiewicz [6] is the time at which the laminar regime ends. In the simulation of Duponcheel and Bartosiewicz [6], the laminar regime lasts up to $t^* \cong 75$. This disparity is explained by the difference between the initial perturbations used in each simulation as we saw the influence of the initial pertubation on the time-history of the shear layer in section 3.3.

After the laminar regime, the flow enters in a phase of transition towards turbulence and the growth rate of the shear layer augments. It finally becomes linear after $t^* = 150$ and flow is considered turbulent. After $t^* = 150$, the growth rates of the temperature mixing layer and of the velocity mixing layer are the same (the red and black curves are parallel). Another difference is that in the turbulent regime, the shear layer does not widen at the same rate in this simulation and in the simulation from Duponcheel and Bartosiewicz [6].

3.4.2 Visualization of the simulation

The velocity and temperature are visualized at $t^* = 100$. At that moment, the flow is in transition towards turbulence.
The figure 3.9 shows that the velocity field is not laminar anymore. The figure also shows that the velocity magnitude outside of the shear layer is as expected $0.5\Delta u$. However contrary to the initial condition, this is not the absolute maximum velocity since in small regions the streamwise velocity reaches roughly $\pm 0.8\Delta u$.

The figure 3.10 shows the effect of the Prandtl number on the temperature. It clearly appears because the smallest fluctuations from the velocity do not appear in the temperature. The effect of the Prandtl number can also be seen when comparing the figure 3.10 with the figure 3.5. As already mentioned, there are no overshoot/undershoot in the temperature field because there are no steep temperature gradients.

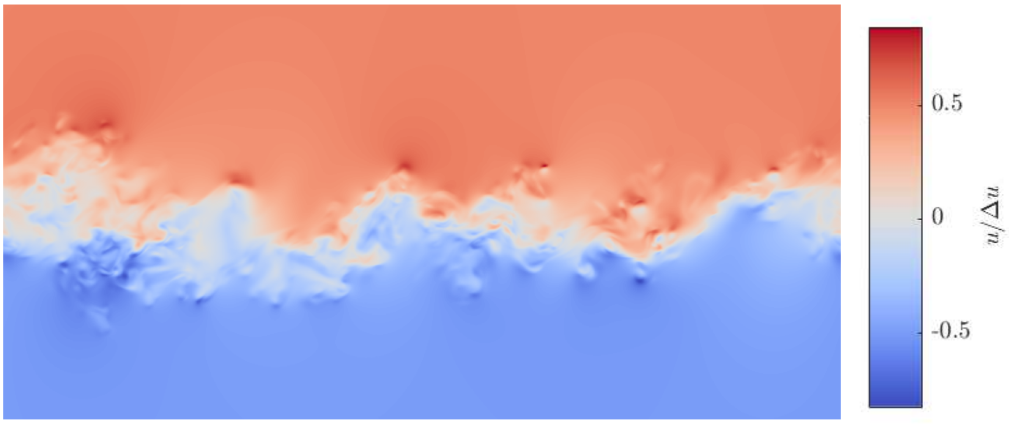

Figure 3.9: Vertical slice of the streamwise velocity at $t^* = 100$ (ML3).

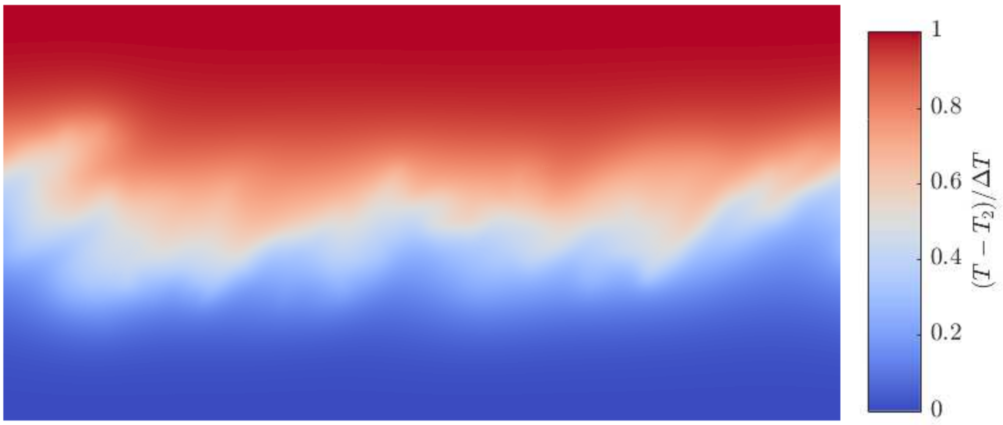

Figure 3.10: Vertical slice of the temperature field at $t^* = 100$ (ML3).

3.4.3 Definition of the turbulent quantities

In order to compare the different simulations, we compute several turbulent quantities:

- the root mean square fluctuations defined by

$$u_{rms} = \sqrt{\frac{2}{3}\langle k\rangle} \tag{3.7}$$

$$T_{rms} = \sqrt{\langle (T')^2\rangle} \tag{3.8}$$

  where $\langle k\rangle$ is the mean turbulent kinetic energy and $T' = T - \langle T\rangle$. The u_{rms} profile is a measure of the cinematic turbulence of the flow as for the T_{rms} profile, it shows how much the temperature deviates from its mean;

- the turbulent shear stress ($\langle\tau_t\rangle$) and the turbulent heat flux ($\langle q_t\rangle$) defined by

$$\langle\tau_t\rangle = -\langle u'v'\rangle \tag{3.9}$$

$$\langle q_t\rangle = -\langle T'v'\rangle. \tag{3.10}$$

  which represent the activeness of the turbulent transfers;

- the total shear stress (τ_{tot}) and the total heat flux (q_{tot}) defined below

$$\langle\tau_{tot}\rangle = \langle\tau_t\rangle + \nu\frac{d\langle u\rangle}{dy} \tag{3.11}$$

$$\langle q_{tot}\rangle = \langle q_t\rangle + \alpha\frac{d\langle t\rangle}{dy} \tag{3.12}$$

- the turbulent Prandtl number computed using the following formulas

$$\nu_t = \langle\tau_t\rangle / \frac{d\langle u\rangle}{dy} \tag{3.13}$$

$$\alpha_t = \langle q_t\rangle / \frac{d\langle T\rangle}{dy} \tag{3.14}$$

$$Pr_t = \frac{\nu_t}{\alpha_t} \tag{3.15}$$

  which represents the ratio between the turbulent viscosity and the turbulent diffusivity.

3.4.4 Comparison of the mean profiles

The mean profiles of the quantities listed in the section 3.4.3 are compared with the mean profiles obtained by Duponcheel and Bartosiewicz [6].
All the mean profiles presented hereafter are the results of space-averaging over the periodic directions and time-averaging between $t^* = 200$ and $t^* = 320$ at intervals $\Delta t^* = 5$. They are plotted in function of the position y is normalized using δ_{95} or $\delta_{T,95}$ the 95-percent thickness for that quantity.

The figure 3.11 shows the two profiles collapse onto each other. Also, the temperature profile does not exhibit a double inflection which means that the flow quickly becames 3D turbulent. However, the velocity profile shows very slight deviations with respect to the temperature profile. It might be a consequence of the domain being too small.

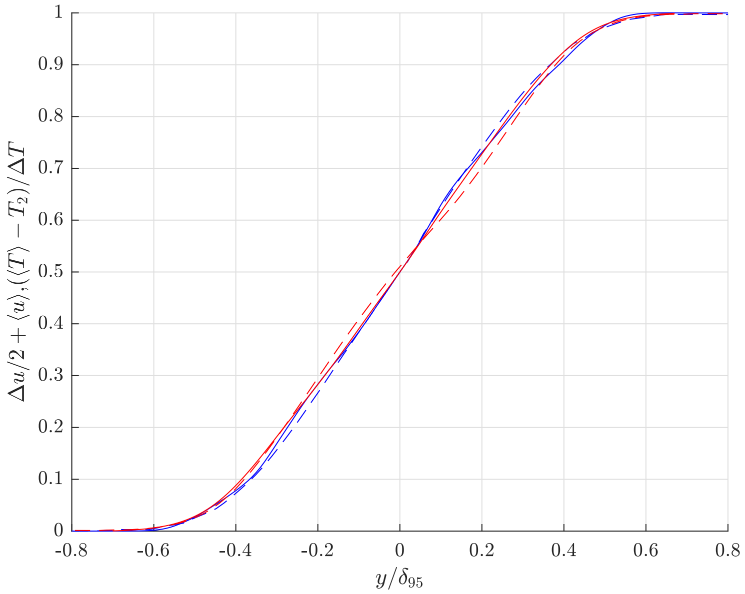

Figure 3.11: Mean profiles of velocity (blue) and temperature (red) from Duponcheel (dashed lines) and the simulation ML3 (solid lines).

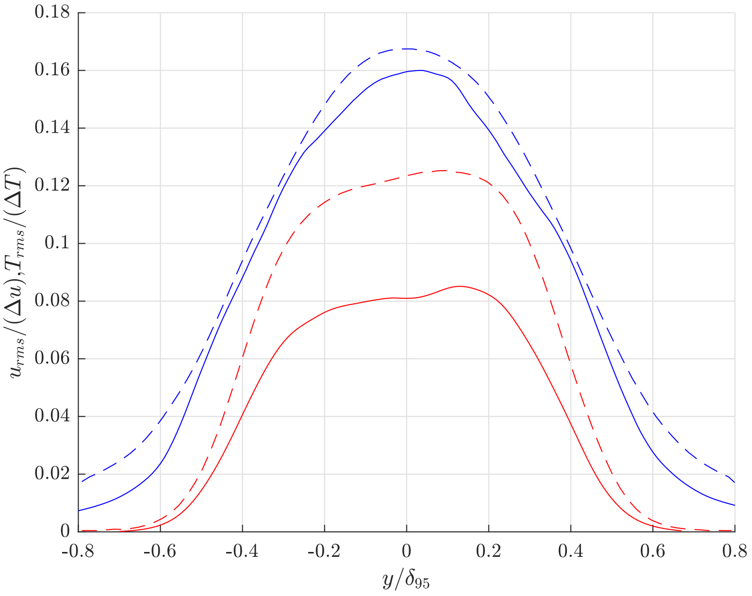

Figure 3.12: RMS profiles of velocity (blue) and temperature (red) from Duponcheel (dashed lines) and the simulation ML3 (solid lines).

The figure 3.12 shows that the u_{rms} profile of the simulation is of the same order of magnitude as the profile obtained by Duponcheel and Bartosiewicz [6]. Also both profiles exhibit the same trend: they have a maximum at the center of the shear layer and their value at $y=\pm0.8\delta_{95}$ is still significant $0.01\Delta u$. The shape of the profile also suggests that the distance at which $u_{rms}=0$ is outside of the shear layer.

It also shows that both T_{rms} profiles exhibit the same trends. They are almost flat between $y=-0.2\delta_{T,95}$ and $y=0.2\delta_{T,95}$ then falls towards zero at roughly $y=\pm0.6\delta_{T,95}$. Also, they are thinner with respect to $\delta_{T,95}$ than the u_{rms} profiles with respect to δ_{95}. However, the amplitude of the T_{rms} profile obtained by Duponcheel and Bartosiewicz [6] is approximately 50% larger than the amplitude of the profile obtained for this simulation.

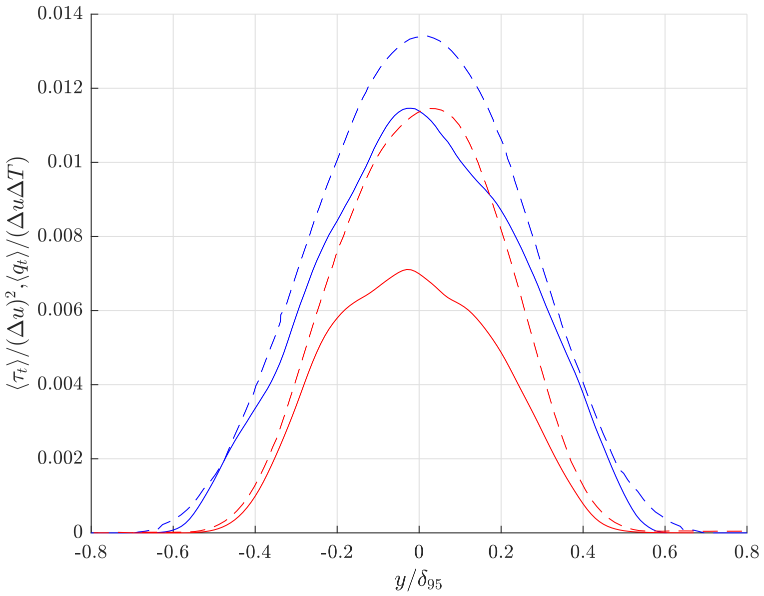

Figure 3.13: Profiles of the turbulent shear stress (blue) and the turbulent heat flux (red) from Duponcheel (dashed lines) and the simulation ML3 (solid lines).

The figure 3.13 shows that the profile of the turbulent shear stress obtained during this simulation is of the same of magnitude as the profile obtained by Duponcheel and Bartosiewicz [6]. Also both profiles are similar in terms of width but the turbulent shear stress profile is not symmetric and this might also be due to the size of the domain.

Regarding the profile of the turbulent heat flux, the amplitude of the one obtained during this simulation is roughly half of the amplitude obtained by Duponcheel and Bartosiewicz [6] but the two profiles tends to 0 roughly in the same region $y = \pm 0.6\delta_{T,95}$.

All the profiles have their maximum at the center of the shear layer which means that the center of the shear layer is most turbulent. The figure 3.13 also shows that the turbulent shear stress profile is narrower than the u_{rms} profile. The same can be said for the turbulent heat flux profile with respect to the T_{rms} profile.

The figure 3.14 shows that the total shear stress is almost equal to the turbulent shear stress in most of the shear layer except on the outer regions. In that manner, the results obtained by this simulation differ from the results obtained by Duponcheel and Bartosiewicz [6]. Furthermore, it is not normal that the ratio between the turbulent shear stress and the total shear stress falls off at $y = 0.6\delta_{95}$ so this might also indicate that the domain is too small.

On the other hand, the ratio between the turbulent heat flux and the total heat flux is of the same order of magnitude as the results from Duponcheel and Bartosiewicz [6] in the outer regions of the temperature mixing layer (outside of $y = \pm 0.3\delta_{T,95}$). At the center, the results obtained by Duponcheel and Bartosiewicz are roughly 10% larger than the results from the simulation but the turbulent heat flux still represents up to 60% of the total heat flux in the middle of the mixing layer. This value goes down to less than 10% at $y \cong \pm 0.5\delta_{T,95}$. The ratio follows the same shape than the turbulent heat flux. Also, the diffusion of temperature is not negligible anywhere.

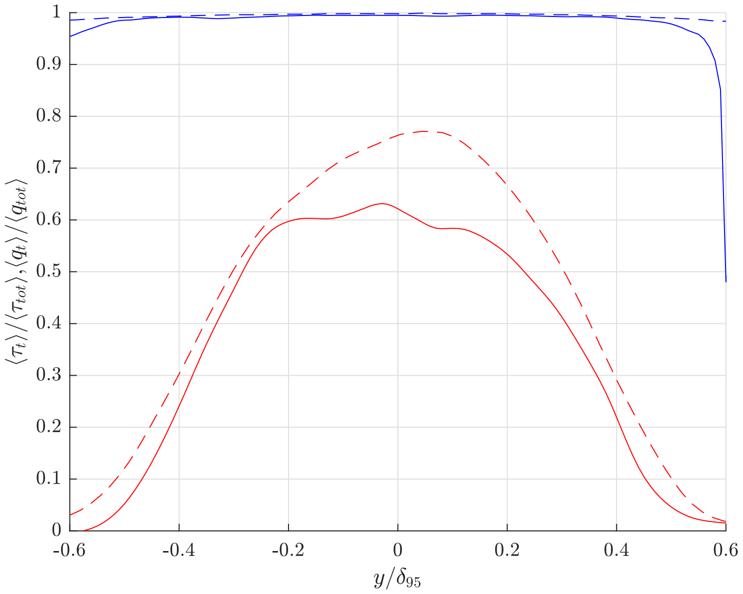

Figure 3.14: Profiles of $\langle\tau_t\rangle/\langle\tau_{tot}\rangle$ (blue) and $\langle q_t\rangle/\langle q_{tot}\rangle$ (red) from Duponcheel (dashed lines) and the simulation ML3 (solid lines).

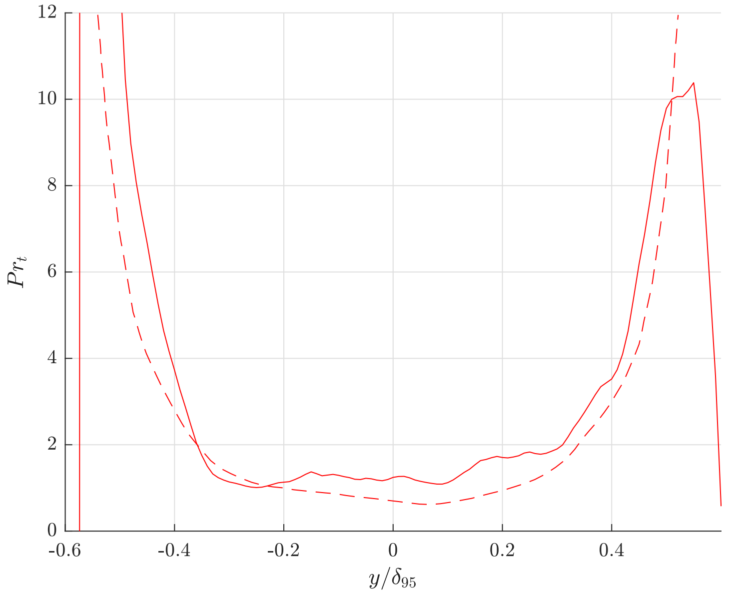

Figure 3.15: Profile of the turbulent Prandtl number from Duponcheel (dashed line) and the simulation ML3 (solid line).

By comparing the turbulent Prandtl number obtained in the simulation with the results obtained by Duponcheel and Bartosiewicz [6] (figure 3.15), it appears that the results are fairly similar between $y = -0.4\delta_{T,95}$ and $y = 0.4\delta_{T,95}$. The minimum values differ a little: 0.7 for Duponcheel and Bartosiewicz [6] and 1 for this simulation. However, the turbulent Prandtl number shows a non-natural behavior outside of $y = \pm 0.4\delta_{T,95}$ that might be due to the size of the domain.

In conclusion, the comparison with the data from Duponcheel and Bartosiewicz [6] shows two unexplained differences:

- the shear layer growth rate in the turbulent regime;
- the amplitude of the turbulent quantities associated to the heat.

These differences could possibly be explained by a couple factors. The first one is the size of the domain. As the domain used for this simulation is small, the results could be influenced by the periodic boundary conditions or some modes actually longer than the domain are supposed to activated but it can not happen in the small domain. The other factor could be that the amplitude of the initial perturbation. Each of these factor is studied in the following simulations.

3.5 Analysis of the results from ML4

This simulation was done to study the influence of the size of the domain on the results. These results are compared with the results obtained by Duponcheel and Bartosiewicz [6] at the same Reynolds and Prandtl numbers. But contrary to the simulation ML3, the domain dimensions in the periodic directions are similar to the dimensions used by Duponcheel and Bartosiewicz [6].

3.5.1 Comparison of the time-history

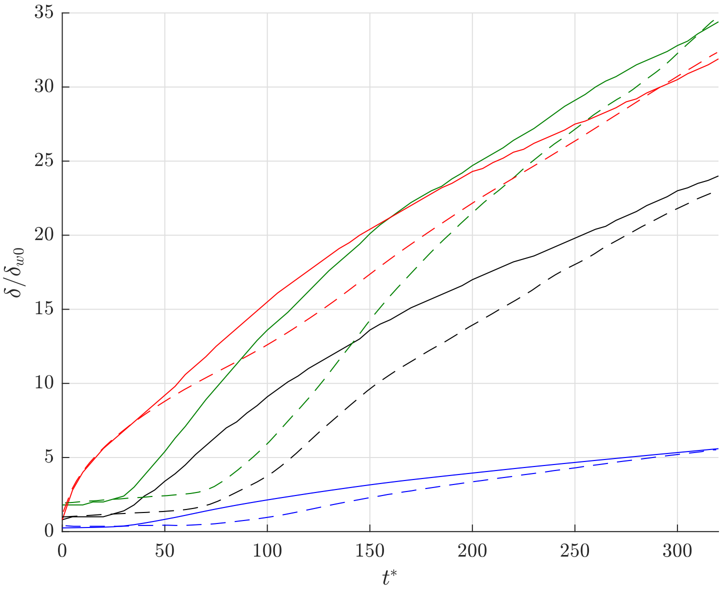

Figure 3.16: Time-history of the shear layer thickness from Duponcheel (dashed lines) and the simulation ML4 (solid lines): δ_θ (blue), $\delta_{T,75}$ (red), δ_{95} (green) and δ_{75} (black).

The figure 3.16 shows the time-history of the shear layer thickness and is very similar to the figure 3.8. As for the simulation ML3, the laminar regimes ends at $t^* \cong 35$ but the temperature thickness is only impacted at $t^* \cong 50$. This does not change significantly from ML3 because the initial perturbation is generated the same way as for the simulation ML3.

After the laminar regime, the flow enters its transition towards turbulence and the growth rate of the shear layer augments. It finally becomes linear after $t^* = 150$. Atfter, the growth rate of the temperature mixing layer and of the velocity mixing layer are the same (the red and black curves are parallel). However, the growth rate is the same as ML3 and augmenting the size of the domain does not influence that.

3.5.2 Visualization of the simulation

The velocity and temperature field are visualized at $t^* = 200$. At that moment the turbulence of the flow is established.
The figure 3.17 shows that the velocity field shows lots of deviations from the mean profile. The figure also shows that the velocity magnitude outside of the shear layer is, as expected, $0.5\Delta u$. However, contrary to the initial condition this is not the absolute maximum velocity as it is observable that the maximum velocity in some small regions is around $0.7\Delta u$.

The figure 3.18 shows the effect of the Prandtl number. It appears that the smallest fluctuations from the velocity once again do not appear in the temperature because of the effect of the high diffusivity.

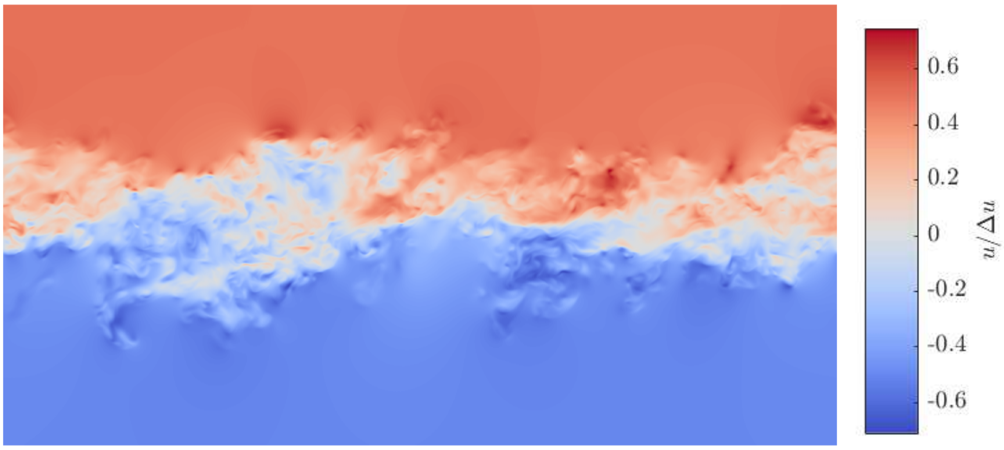

Figure 3.17: Vertical slice of the streamwise velocity field at $t^* = 200$ (ML4).

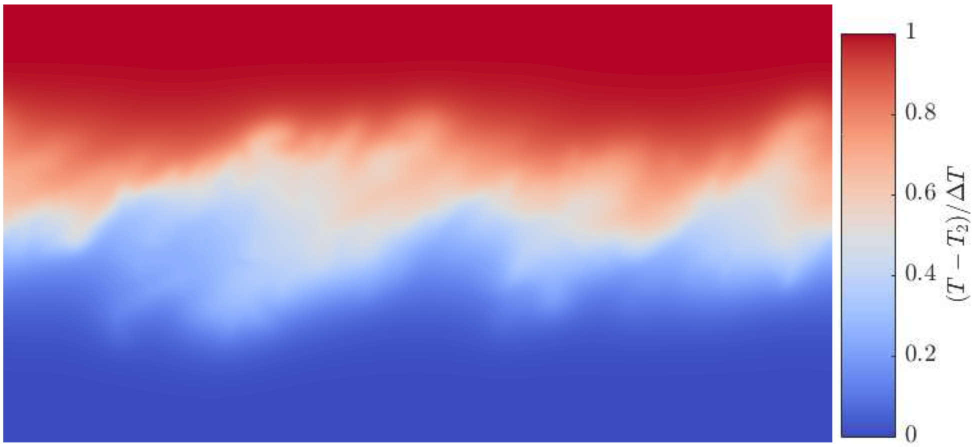

Figure 3.18: Vertical slice of the temperature field at $t^* = 200$ (ML4).

3.5.3 Comparison of the mean profiles

The mean profiles of the quantities listed in section 3.4.3 are compared with the mean profiles obtained by Duponcheel and Bartosiewicz [6]. All the mean profiles presented after are the results of space-averaging over the periodic directions and time-averaging between $t^* = 200$ and $t^* = 320$ at intervals $\Delta t^* = 5$. The position y is normalized using δ_{95} or $\delta_{T,95}$ the 95-percent thickness for that quantity. .

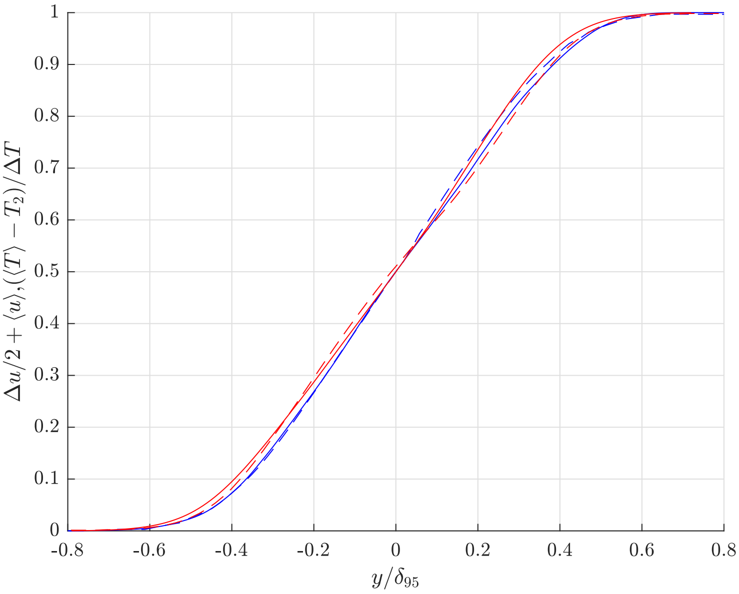

Figure 3.19: Mean profiles of velocity (blue) and temperature (red) from Duponcheel (dashed lines) and the simulation ML4 (solid lines).

The figure 3.19 shows that the profiles are very similar to the results from Duponcheel and Bartosiewicz [6]. The temperature profile exhibits small deviations with respect to the velocity profile in the outer region of the temperature mixing layer. This is possibly due to the fact that in this region the heat flux is mainly due to the diffusion.

The figure 3.20 shows that the u_{rms} profile of the simulation is of the same order of magnitude as the profile obtained by Duponcheel and Bartosiewicz [6]. The same observations that were made for the u_{rms} profile of the simulation ML3 can be made for this simulation as their profiles are similar.

It also shows that both T_{rms} profiles exhibit the same trends. They are almost flat between $y = -0.2\delta_{T,95}$ and $y = 0.2\delta_{T,95}$ then falls towards zero at roughly $y = \pm 0.6\delta_{T,95}$. Also, they are thinner with respect to $\delta_{T,95}$ than the u_{rms} profiles with respect to δ_{95}. The amplitude max of the T_{rms} profile of this simulation is larger than the amplitude obtained from the simulation ML3 which is due to the domain size change.

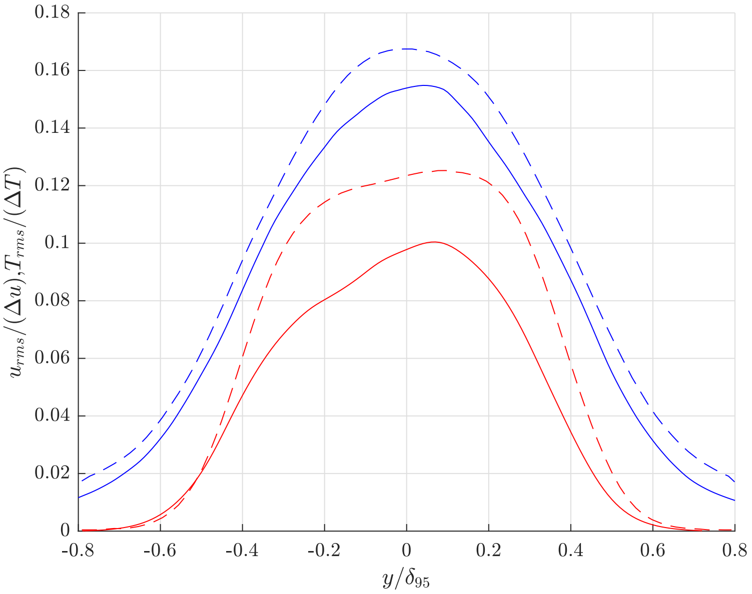

Figure 3.20: RMS profiles of velocity (blue) and temperature (red) from Duponcheel (dashed lines) and the simulation ML4 (solid lines).

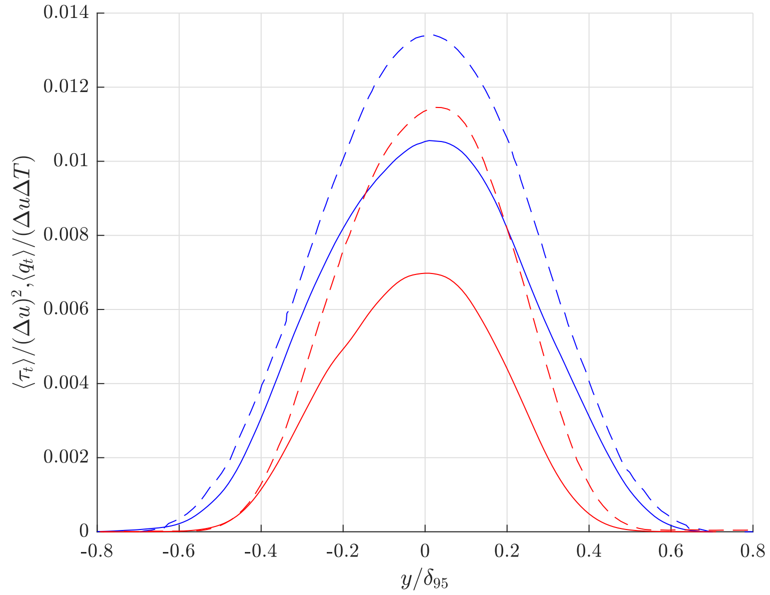

Figure 3.21: Profiles of the turbulent shear stress (blue) and the turbulent heat flux (red) from Duponcheel (dashed lines) and the simulation ML4 (solid lines).

On the figure 3.21, the amplitudes of the turbulent shear stress and of the turbulent heat flux are smaller for this simulation than the results from Duponcheel and Bartosiewicz [6]. However, both have a maximum at the center of the shear layer which means that the center of the shear layer is most turbulent. The figure also shows that the turbulent heat flux is narrower with respect to $\delta_{T,95}$ than the turbulent shear stress with respect to δ_{95}. This is a consequence of the thickness of the temperature layer being bigger than the thickness of the velocity mixing layer. The profiles obtained for this simulation are symmetric which was not the case in the simulation ML3.

The figure 3.22 shows that the total shear stress is almost equal to the turbulent shear stress. This is expected for a high Reynolds flow because the viscous diffusion of momentum is negligible compared to the diffusion done by the turbulence of the flow. Also, the turbulent shear stress accounts for almost 100% of the total shear stress across the entire shear layer even at $y = \pm 0.6\delta_{95}$ where the turbulent shear stress is more than 20 times less important than in the middle of the shear layer.

On the other hand the observations are different for the ratio between the turbulent and the total heat fluxes. Although the amplitude is smaller than the results by Duponcheel and Bartosiewicz [6], the turbulent heat flux represents up to 60% of the total heat flux in the middle of the mixing layer. But this value goes down to roughly 10% at $y \cong \pm 0.5\delta_{T,95}$. Thus, it is clear that the diffusion of temperature is not negligible anywhere. Moreover, the ratio follows the same shape than the turbulent heat flux.

The figure 3.23 shows the comparison of the turbulent Prandtl number with the results obtained by Duponcheel and Bartosiewicz [6], it appears that the profiles are lightly different. The minimum value is not exactly the same: 0.7 for Duponcheel and Bartosiewicz [6] against 1 for this simulation. Nonetheless, the trend shown is the same and the turbulent Prandtl is not constant across the mixing layer. This is the consequence that the turbulent heat flux falls off faster in the outer region of the mixing layer faster than the turbulent shear stress.

In conclusion, augmenting the size of the domain makes the results closer to the results of Duponcheel and Bartosiewicz [6]. At this stage, we can conclude that the domain used for the simulation ML3 was most likely too small. However, as seen in the figure 3.16, the growth rate of the shear layer in the turbulent regime is still slower than the results obtained by Duponcheel and Bartosiewicz [6]. So, the next section presents the analysis of the simulation ML5 done to determine the influence of the amplitude of the initial perturbation.

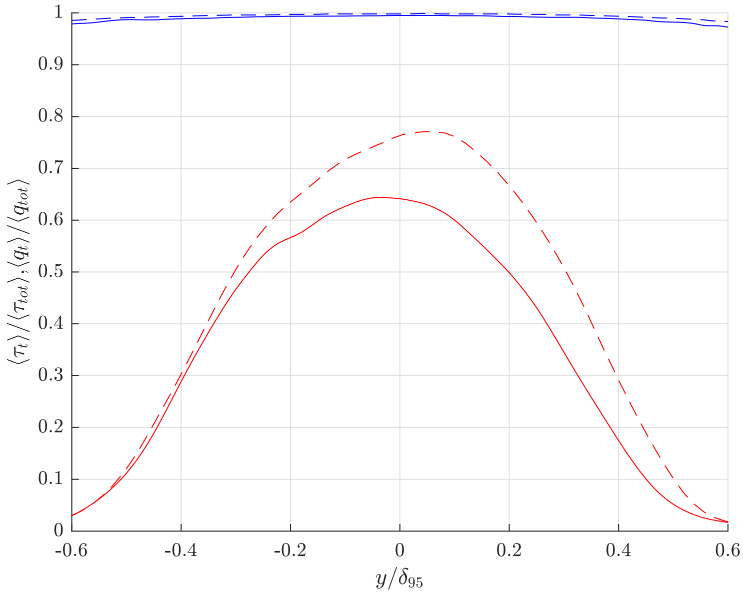

Figure 3.22: Profiles of $\langle\tau_t\rangle/\langle\tau_{tot}\rangle$ (blue) and $\langle q_t\rangle/\langle q_{tot}\rangle$ (red) from Duponcheel (dashed lines) and the code (solid lines).

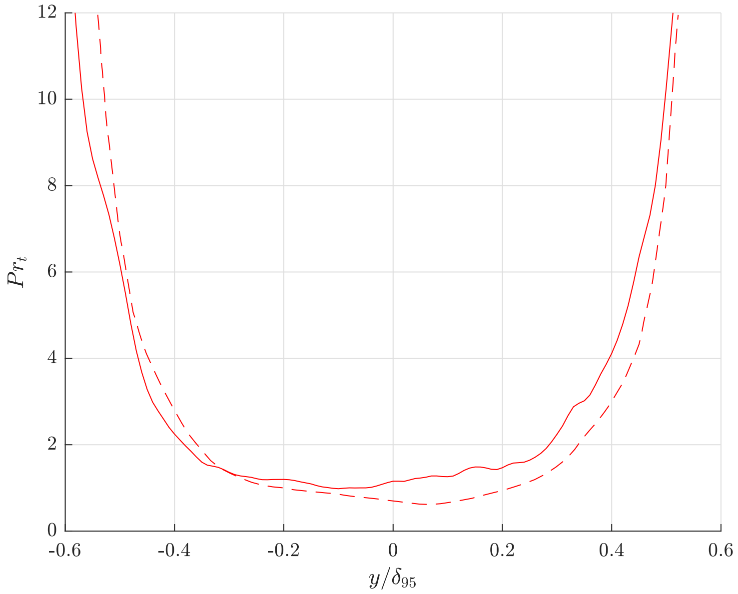

Figure 3.23: Profile of the turbulent Prandtl number from Duponcheel (dashed line) and the simulation ML4 (solid line).

3.6 Analysis of the results from ML5

This simulation permits to study the influence of the u_{rms} of the initial perturbation. The results of this simulation are compared with the results obtained by Duponcheel and Bartosiewicz [6] at the same Reynolds and Prandtl numbers.

3.6.1 Comparison of the time-history

The figure 3.24 shows the time-history of the shear layer thickness. Like for the simulations ML3 and ML4, the laminar regime ends at $t^* \cong 35$. As expected, the duration of the laminar regime is not significantly influenced by the amplitude of the initial perturbation.

However, there is a major difference between the results from this simulation and the results from the simulations ML3 and ML4. After $t^* = 150$, the expansion rate of the shear layer does not drop significantly like in the simulations ML3 and ML4. So, it appears that the u_{rms} of the initial noise impacts the growth rate of the mixing layer in the turbulent regime and using u_{rms}, this simulation almost reproduces the same widening speed in the turbulent regime as the simulation by Duponcheel and Bartosiewicz [6].

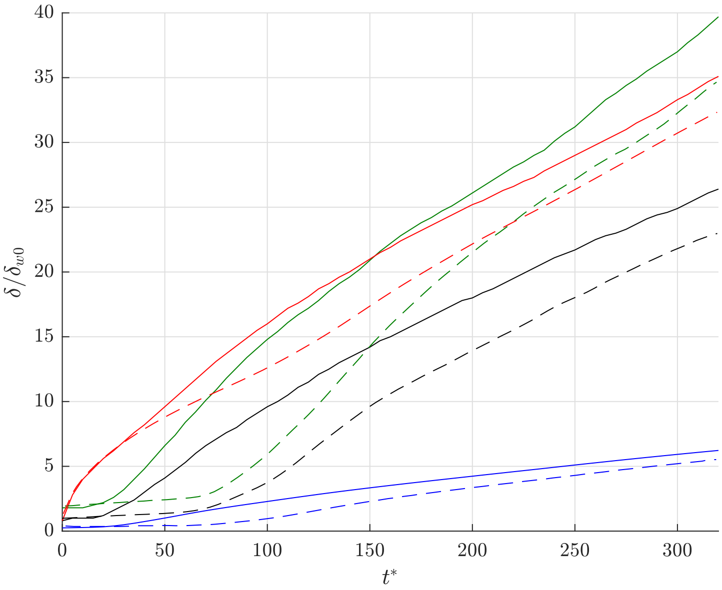

Figure 3.24: Time-history of the shear layer thickness from Duponcheel (in dashed lines) and the simulation ML5 (in solid lines): δ_θ (blue), $\delta_{T,75}$ (red), δ_{95} (green) and δ_{75} (black).

3.6.2 Comparison of the mean profiles

The mean profiles of different quantities listed in section 3.4.3 are compared with the mean profiles obtained by Duponcheel and Bartosiewicz [6]. All the mean profiles presented hereafter are the results of space-averaging over the periodic directions and time-averaging between $t^* = 200$ and $t^* = 320$ at intervals $\Delta t^* = 5$. The position y is normalized using δ_{95} or $\delta_{T,95}$ the 95-percent thickness for that quantity.

The major improvement with the simulations ML3 and ML4 is the bigger amplitude of the fluctuations profiles. Indeed, the RMS profiles obtained by this simulation fits exactly the results obtained by by Duponcheel and Bartosiewicz [6].

In comparison with the figure 3.20, it shows that the T_{rms} profile has the same width. Even though, the amplitude of the T_{rms} profile at the center of the shear layer is 20% bigger than the results from the simulation ML4, the value of T_{rms} is still almost 0 at $y = \pm 0.6\delta_{T,95}$. This characteristic is indeed not dependent of the amplitude of the fluctuations but is exhibited because the temperature thickness is wider than the velocity thickness.

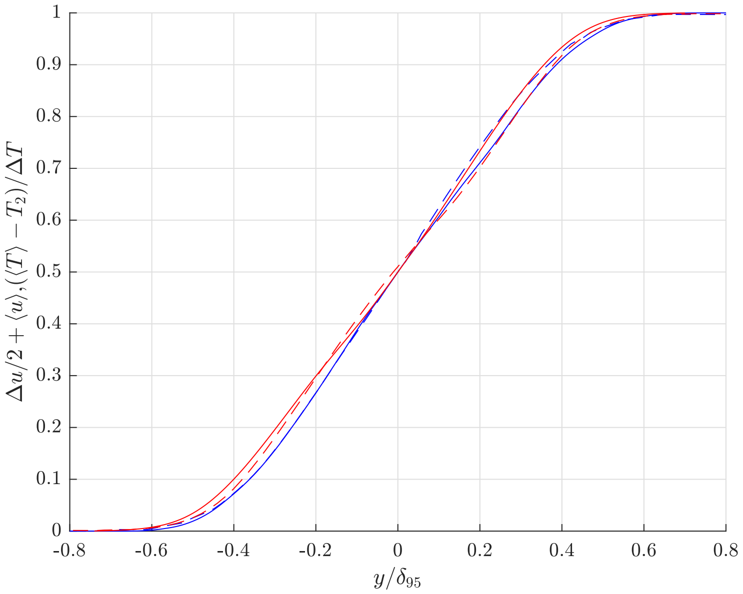

Figure 3.25: Mean profiles of velocity (blue) and temperature (red) from Duponcheel (dashed lines) and the simulation ML5 (solid lines).

The figure 3.25 shows that the profiles obtained for this simulation are similar to the ones obtained by the simulation ML4. So in the outer region of the temperature mixing layer, the temperature profile deviates from the mean streamwise velocity profile.

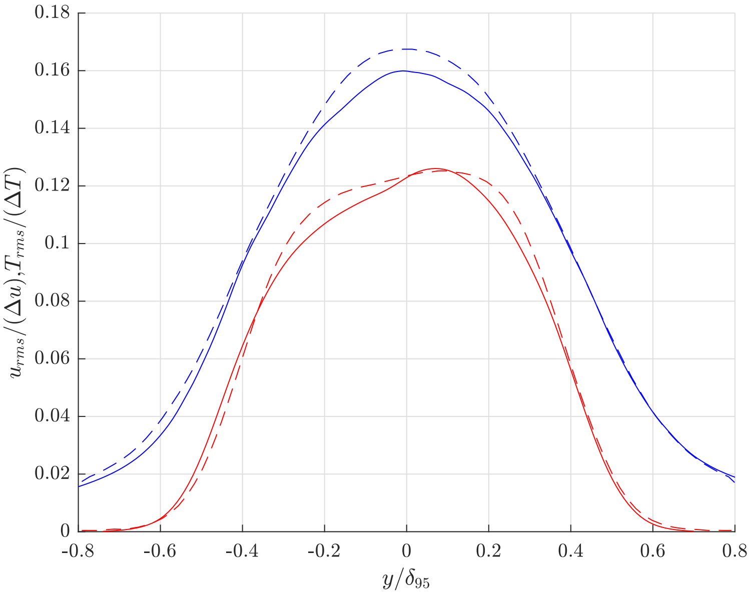

Figure 3.26: RMS profiles of velocity (blue) and temperature (red) from Duponcheel (dashed lines) and the simulation ML5 (solid lines).

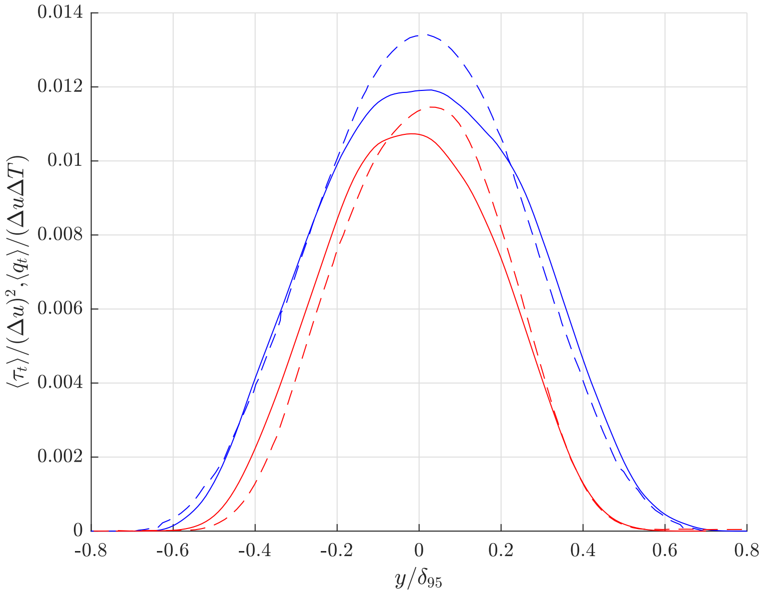

Figure 3.27: Profiles of the turbulent shear stress (blue) and the turbulent heat flux (red) from Duponcheel (dashed lines) and the simulation ML5 (solid lines).

The figure 3.27 shows that as a results of the higher fluctuations, the turbulent shear stress and turbulent heat flux are higher than for the simulations ML3 and ML4. The results obtained by this simulation are of the same order of magnitude than the results by Duponcheel and Bartosiewicz [6]. However, there appears to be a more significant difference regarding the turbulent shear stress than the turbulent heat flux.

Regarding the width of the mean turbulent heat flux profile, it is similar to the width of the mean turbulent heat flux profile obtained in the simulation ML4. Also, the same can be said about the turbulent shear stress. So, the amplitude of the initial perturbation affects the amplitude of the turbulent profiles but not their relative width with respect to either δ_{95} or $\delta_{T,95}$.

The figure 3.28 shows, as expected, that the turbulent shear stress accounts for almost 100% of the total shear stress across the entire shear layer. Moreover, the results of this simulation show that the turbulent heat flux is more significant than in the simulation ML4. At the center of the shear layer, it accounts for more than 70% of the total heat transfer. But on the side of the temperature mixing layer, the ratio falls off similarly to the results of the simulation ML4.

The figure 3.29 shows that the turbulent Prandtl reaches a minimum value of 0.7 as in the results by Duponcheel and Bartosiewicz [6]. However, the profile obtained here is slightly wider than the results by Duponcheel and Bartosiewicz [6] . This difference could be explained by the fact that the laminar regime lasts a shorter time in this simulation compared to the simulation by Duponcheel and Bartosiewicz [6] (see figure 3.25 and thus, the difference of thickness between the temperature mixing layer and the velocity mixing layer is smaller than for the results by Duponcheel and Bartosiewicz [6].

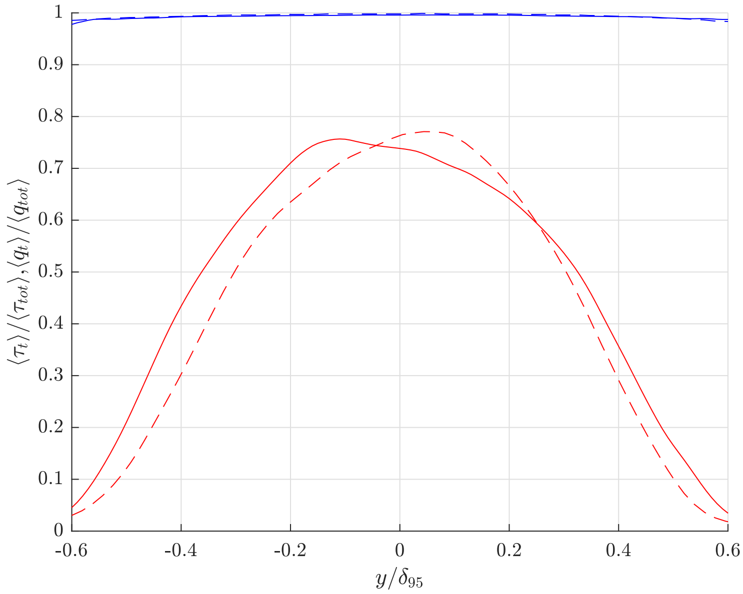

Figure 3.28: Profiles of $\langle\tau_t\rangle/\langle\tau_{tot}\rangle$ (blue) and $\langle q_t\rangle/\langle q_{tot}\rangle$ (red) from Duponcheel (dashed lines) and the simulation ML5 (solid lines).

In conclusion, this simulation shows results very similar to the one obtained by Duponcheel and Bartosiewicz [6] and also that the amplitude of the initial perturbation has a long lasting influence on the flow as it modifies that expansion rate of the shear layer in the turbulent regime and augments the amplitude of the turbulent profiles related to the heat.

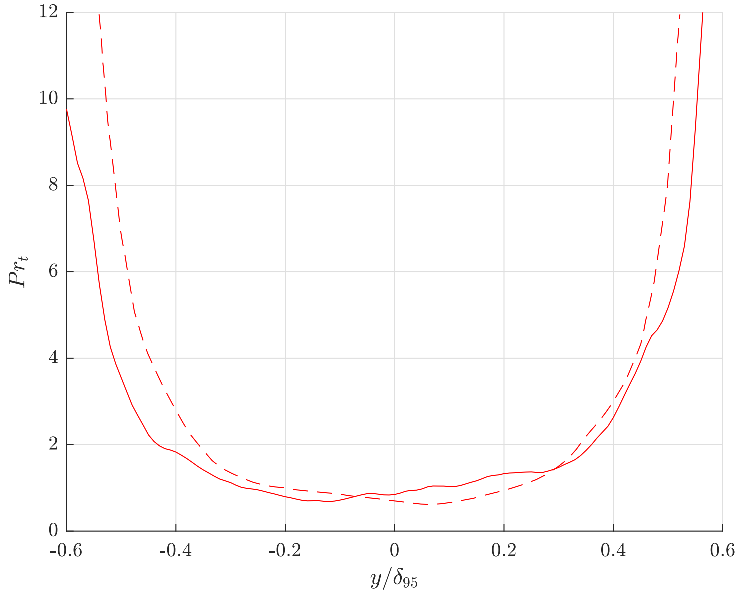

Figure 3.29: Profile of the turbulent Prandtl number from Duponcheel (dashed lines) and the simulation ML5 (solid lines).

Conclusions and perspectives

Conclusions

The goals of this master thesis were to develop a numerical tool capable of doing the DNS of a time-developing mixing layer at low Prandtl number and to study the heat transfer of such flow.

To this extent, the choice was made to modify an existing code VPM. To modify VPM, first it was needed to well understand its operation. Therefore, VPM was presented in the first chapter of this document. Then, the strategy used to add the support to solve the thermodynamics of the flow was detailed in that same chapter.

The steps of validation for the implicit solver were detailed in the second chapter and we saw the convergence and the stability of this implicit scheme. We also came across a very interesting property that the method does not give valid results for a too small time-step.

Finally, the third chapter contains the study of the time-developing mixing layer with multiple simulation results. In this regard, we were able to compare the results obtained with references from the literature and match them pretty well.

We were also able to examine the influence that certain parameters can have on a simulation:

1. the size of the computational domain which can influence the results obtained due to undesired periodicity effects;
2. the spectrum of the initial perturbation influences the time-history of the shear layer and more precisely the duration of the laminar regime;
3. the amplitude of the initial perturbation can influence the growth rate of the shear layer in the turbulent regime as well as the amplitude of the turbulent heat transfer.

From the different simulations, we were able to draw conclusions on the time-developing mixing layer at a low Prandtl number:

1. we saw that because of the diffusion, only the big scales of the flow affect the temperature field;
2. we computed the turbulent Prandtl number which was not constant across the mixing layer as contrary to the results when $Pr = 1$ [6].

Perspectives

Firstly, it would be interesting to simulate a time-developing mixing at a higher Reynolds number to see if, starting from a certain Reynolds number, the turbulent heat flux accounts for $\approx 100\%$ of the total heat flux in the center of the mixing layer. It would also be interesting to investigate more into the influence of the initial perturbation on the turbulent profiles.

Secondly, the method implemented is not perfect. Indeed, a drawback of the method developed for this master thesis is that we use the same particles and domain for the vorticity and the temperature. At a low Prandtl number, the thickness of the temperature is wider than the thickness of the vorticity.

Therefore, this method creates a domain which is bigger than needed for the vorticity. This augments the computation cost in two fashions:

1. there are many particles that actually have zero vorticity and thus, could be ignored when interpolating the vorticity on the mesh;
2. the velocity needs to be computed on a bigger domain than if there was no temperature field .

However, we can not just ignore the temperature and keep the domain as compact as the support of the vorticity.

So, to improve the method, we could split vorticity and temperature particles. We would use Vpm to time integrate the Navier-Stokes equations. On top of which, we would add the new set of temperature particles. Nevertheless, it would still be a hybrid Particle-Mesh solver as the diffusion would be done on a grid for the temperature. This grid would grow according to the support of the temperature Laplacian. Hence, two unbounded domains would be defined with independent sizes.

A special attention needs to be paid to the computation of the velocity of the temperature particles. Indeed, the domain defined by the temperature will be bigger than the support of the vorticity. Therefore, some temperature particles will exist outside the domain defined by the vorticity. But the velocity would only be computed on that domain. Fortunately, we showed during this master thesis that the velocity fluctuations far from the center of the mixing layer do not induce a turbulent heat flux. So, the easiest solution would be to simply set the velocity of the temperature particles outside of the vorticity domain to the velocity outside of the shear layer.

This method would have several advantages over the method currently implemented. Firstly, the size of the domain in Vpm would not be influenced by the temperature thickness. In consequence, the domain in Vpm would grow according to the support of vorticity and the velocity would be computed on a smaller domain than currently. Secondly, we saw in the visualizations of the simulations at a low Prandtl number that the temperature does not exhibit as small scales as the velocity. Thus, the method currently implemented uses a grid finer than needed to capture all the scales of the temperature. Splitting the vorticity and the temperature would allow us to use grids of different size for the vorticity and the temperature.

Bibliography

[1] K. Nagat Y. lto T. Watanabe, Y. Sakai and T. Hayase. Turbulent mixing of passive scaler near turbulent and non-turbulent interface in mixing layers. *Physics of fluids*, 2015.

[2] M. Bergdorf D. Rossinelli W. Andreoni P. Chatelain, A. Curioni and P. Koumoutskos. Billion vortex particle direct numerical simulations of aircraft wakes. *Computer methods in applied mechanics and engineering*, 2008.

[3] P. Chatelain and P. Koumoutsakos. A fourier-based elliptic solver for vortical flows with periodic and unbounded directions. *Journal of Computational Physics*, 2010.

[4] V. Legat and G. Winckelmans. Mécanique des fluides et transferts 1. Course notes, 2015.

[5] M. Duponcheel and Y. Bartosiewicz. Dns and les of a time developing mixing layer. Presentation slides for MYRTE Progress Meeting, March 2017.

[6] M. Duponcheel and Y. Bartosiewicz. Low prandtl turbulent heat transfer in unconfined or wall-bounded configurations: Dns of a mixing layer and of an impinging jet. 2017.

Rue Archimède, 1 bte L6.11.01, 1348 Louvain-la-Neuve **www.uclouvain.be/epl**