

DEVELOPMENTS IN HYBRID APPROACHES

Vortex method with known separation location

Vortex method with near-wall Eulerian solver

RANS-LES coupling

GOÉRIC DAENINCK

Décembre 2006

Thèse présentée en vue de l'obtention du grade
de docteur en sciences appliquées

Composition du jury

Prof. J.-C. Samin (président)

Prof. G. Winckelmans (promoteur)

Prof. F. Dupret

Dr P. Geuzaine

Prof. V. Legat

Prof. P. Poncet

**Faculté des sciences
appliquées**

Université catholique de Louvain

To my grandfather Dr. Louis Daeninck

In the first place I would like to stress how much the years working on my dissertation have been exciting, instructive and fun. This is certainly due in part to my own interest in the subject matter and in research in general. Yet this work would not have been possible without the involvement and contributions of many people around me.

Above all, I would like to express my gratitude to my advisor Professor Grégoire Winckelmans. It has been a privilege to pursue my Ph.D. under his supervision. At critical moments, he found the right words to extract the best of me.

I would like to thank the members of my thesis jury for the time they devoted to it: Professors François Dupret and Vincent Legat who were also on my thesis committee and therefore had to survive numerous lengthy intermediate presentations; Professor Philippe Poncet who accepted to be part of the jury as a specialist in the field of Lagrangian methods; Doctor Philippe Geuzaine who showed great interest in the small world of vortex methods and hybrid approaches; and Professor Jean-Claude Samin, head of UCL's Mechanical Engineering Department, who accepted to chair the jury.

I also wish to thank the many colleagues I had the opportunity to meet and work with during my Ph.D. years at UCL: Hervé, Paul and François with whom I formed the early (and world-famous) "Winky Boys". Later the team expanded with the arrival of new doctoral students and the researchers on European projects. In particular, this last year was one of the richest for me in terms of interaction and mutual support. Roger, Laurent, Olivier, Matthieu, Timothée, Cédric and most recently Ivan made life at the TERM division highly enjoyable: a big thank you to all of you.

For my stay at Stanford University, I am grateful to the Belgian American Educational Foundation that provided me with the necessary funding, as well as Georgi Kalitzin and Gorazd Medic – my office mates for one year – with whom I had memorable debates on turbulence and many other topics.

Finally, I would like to thank my family and friends: my parents, who always encouraged and guided me; Yolande and Inès who often took great care of their scatterbrain brother; and my in-law family that supported me in particular during the last 12 months, when Marie-Catherine and I were living 9000 kilometers apart from each other. Most importantly, I thank Marie-Catherine for her love and sustained comfort all along these work-intensive years.

Contents

1	Introduction	1
1.1	Objectives	1
1.2	Outline	5
1.3	Vortex and boundary element method	7
2	Vortex methods with prescribed separation regions	13
2.1	Introduction	13
2.2	Infinitely thin boundary layers	14
2.3	“Under-resolved” DNS mode	18
2.4	A combined approach	24
3	A hybrid Eulerian-Lagrangian vortex method	51
3.1	Motivation	51
3.2	Global approach	53
3.3	The method	54

3.4	Results	61
3.5	Summary	70
4	Redistribution on hierarchically refined grids	73
4.1	Motivation	73
4.2	Conservative redistribution scheme on hierarchically refined grids	76
4.3	Cross level particle interactions	88
4.4	Validation	91
4.5	Summary	103
5	Flow past a hemisphere using a hierarchically refined grid	105
5.1	Introduction	105
5.2	Computational approach	106
5.3	Numerical setup	111
5.4	Results	113
5.5	Summary	121
6	LES calculations of spatially evolving wakes in ground effect	145
6.1	Introduction	147
6.2	Definition of the longitudinally developing wake	148
6.3	Numerical codes	152
6.4	Simulation configurations	156
6.5	Results	159
6.6	Conclusion	165
7	Investigation of a hybrid RANS-LES approach	191
7.1	Introduction	191
7.2	Numerical method and computational details	194
7.3	Code validation: from DNS to RANS and DES	196
7.4	On near-wall dynamic coupling of LES with RANS	209

7.5 Summary	235
8 Conclusion	237
Bibliography	241
A Flow past lifting surfaces using a lifting line model	251
B Mapping of the Physical Domain: 3-D Trumpet Mapping	259
C Multiple moving bodies	263
D List of symbols for channel flow	269
E Turbulence models	271
F Movie of the flow past a hemisphere at $Re=3,000$	273

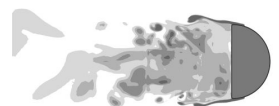
Introduction

1.1 Objectives

Predicting and understanding turbulent flows is still a partially open problem despite the huge effort devoted to it since more than a century. Turbulence also constitutes a very broad field. Here, we are specifically interested in incompressible external aerodynamics at high Reynolds numbers and often with massive separation regions. Such flows are typically encountered in technological fields such as the aircraft and automobile industries. Figure 1.1 shows a visualization of the flow past a sphere at $Re = 15,000$: an academical case which exhibits most of the essential flow features encountered in such flows (i.e., thin boundary layers, massive separation, shear layer with laminar to turbulent transition regions and fully turbulent regions).

Since the seventies, Computational Fluid Dynamics (CFD) has allowed to gain more insight into the physics of turbulence, through numerical simulation; yet much remains to be done. The goal of this work is to contribute to the development of an efficient numerical method that would be particularly efficient for the simulation of high Reynolds number turbulent flows past bluff bodies.

CFD can be subdivided in two distinctive areas which are at the same time closely related:



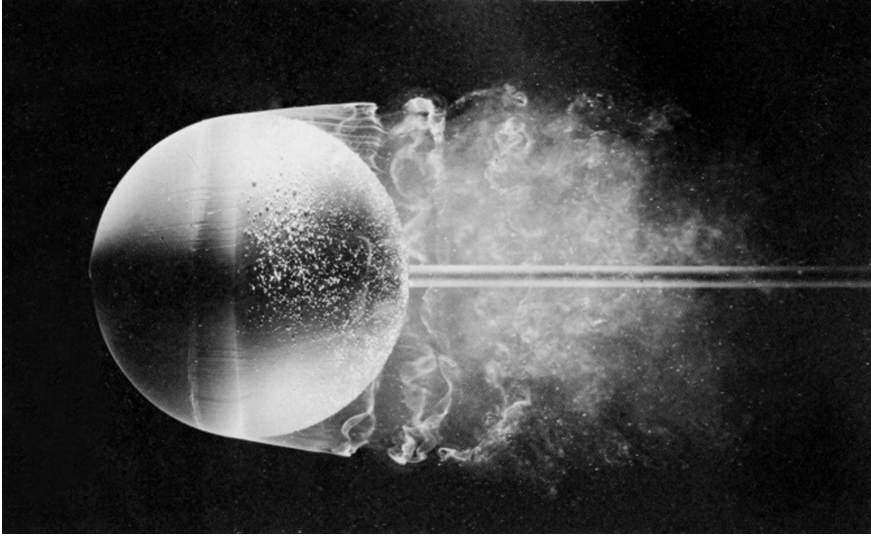


FIGURE 1.1 – Flow past a sphere at $Re = 15,000$. Dye in water shows the laminar boundary layer separating as a shear layer that remains laminar for almost one radius. It then becomes unstable and turbulent. *ONERA photograph, Werlé 1980* (reproduced from Van Dyke [94]).

Physical modeling

Obviously the basis of any CFD problem are the Navier-Stokes equations, supplemented by turbulence model equations to take the effect of unresolved scales (space and/or time) into account. Depending on the level of modeling, one refers to: Reynolds-Averaged Navier Stokes equations (RANS, where all the turbulent motions are modeled), Large-Eddy Simulation (LES, where a large fraction of the turbulence is resolved by the computation while the turbulent subgrid-scale stresses are modeled) and Direct Numerical Simulation (DNS, no model). As the full Navier-Stokes equations are by far too demanding (in terms of computational cost) to be used as such for most problems of practical interest, turbulence modeling remains one of the major fields in CFD.

Numerical solver

As important as the equations, is the numerical solver: the continuous differential equations have to be discretized in space and time in order to be used. A large variety of approaches exists. Because of efficiency and accuracy considerations, the numerical approach is usually closely

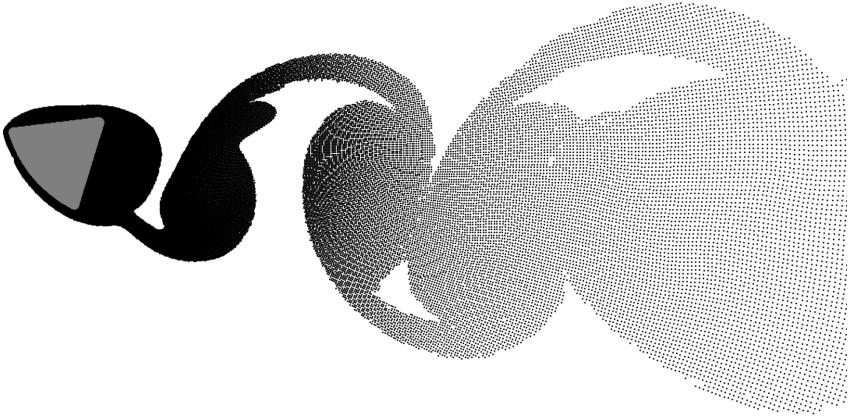
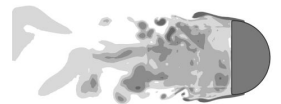


FIGURE 1.2 – Flow past an “Apollo” shape capsule at $Re = 1,500$ using the 2-D vortex element method. Visualization of the vortex particle locations. *Simulation by Ploumhans [76].*

related to the model equations to be solved. RANS solvers, for instance, use stabilizing differential operators along with fully implicit solvers to allow for large time steps (as time accuracy is not an issue, except for “unsteady RANS”, URANS). LES or DNS solvers, on the other hand, require energy conserving schemes (with respect to the convection term) and generally use explicit time integration.

We here focus on a particular numerical approach: the Lagrangian vortex element method. It greatly differs from the other “classical” numerical methods for several reasons. First, it is based on the discretization of the vorticity field: it solves the Navier-Stokes equations in their vorticity-velocity formulation, instead of the usual velocity-pressure formulation. The other major difference is that it solves the equations in the Lagrangian frame of reference: the computational elements are convected with the local flow field. Figure 1.2 shows the location of the computational elements for a simulation using the 2-D vortex element method.

The advantages of vortex methods usually put forward are the fact that: (i) computational elements are required only in regions where the vorticity field is non zero (which, for external flows is limited to the boundary layer regions and the wake), (ii) the computational elements automatically adapt to the flow requirements, (iii) the far field boundary conditions are enforced exactly



and, most importantly, (iv) the method transports flow structures without dispersion errors (as most grid-based methods do) or numerical diffusion (as some grid-based methods may do). There is also no CFL stability constraint on the convection; yet, of course, there are limitations related to accuracy. These advantages make vortex methods particularly well suited to efficiently simulate unsteady, unbounded, flows. In fact, for this type of flow configuration, vortex methods have practically no equivalent.

Of course, vortex methods also have a series of intrinsic drawbacks. The most consequential disadvantage is that the standard method uses radially symmetric computational elements: it is therefore not adapted to the “thin layer” scales found in boundary layers. Furthermore, boundary layer regions are usually close to being steady in the Eulerian frame: solving the equations in the Lagrangian frame of reference is thus much less beneficial there. The self adaptivity, mentioned above as an advantage, has in fact to be controlled for both accuracy and efficiency reasons: in order for vortex methods to converge to solutions of the Navier-Stokes equations, the particle cores should maintain a good spatial distribution (this calls for accurate and efficient “particle redistribution” schemes); it should also be possible to control the particle resolution depending on the local physics of the flow. Another important issue in three-dimensional vortex methods is related to the divergence-free constraint on the vorticity field, which is not naturally enforced or controlled by the method. Finally, while vortex methods require a limited number of elements, the computational cost per element is also much higher than a “grid point” of a grid based solver.

Thanks to its advantages and despite its drawbacks, past and recent developments allowed the Lagrangian vortex method to perform DNS of moderate Reynolds number bluff-body flows. The most important achievements in this domain being the DNS at $Re = 300$ and 500 of the flow past a sphere by Ploumhans *et al.* [78] as well as the DNS investigations of the flow past a cylinder at $Re = 400$ and 500 by Poncet *et al.* [81, 82]. These state of the art computations represent, in some sense, the upper limit in terms of Reynolds number for the Lagrangian vortex method. Obviously, an increase of computing power would allow to simulate higher Reynolds number flows, although not efficiently. This is essentially due to the limitation of the method to capture thin boundary layer regions.

The aim of this work is to further develop three-dimensional vortex methods to make them able to compute high Reynolds number flows in external

aerodynamics. To reach this objective we will first have to overcome the major drawback of vortex methods: its inability to efficiently handle boundary layer regions. Other issues such as a better control of the local spatial discretization and the problem of the vorticity divergence also need to be addressed. These challenges are essentially related to the numerical solver. However, no matter how efficient the numerical method used, there will always be the issue of turbulence modeling. An important part of this work will thus also be related to incorporating additional physical modeling in vortex methods.

1.2 Outline

Ch. 2 Vortex methods with prescribed separation regions

We first investigate the ability of the vortex and boundary element method to simulate high Reynolds number flows where the separation points are approximately known a priori (such as the flow past a hemisphere). The basic idea is that, for these flows, it should suffice to inject the wake vorticity along the separation line to model the flow correctly. We can thus investigate the capabilities of the vortex method to handle high Reynolds number turbulent wake flows without actually resolving the detailed evolution of the boundary layer, and still obtain the essential physical features of the flow. Three different approaches are investigated.

Ch. 3 A hybrid Eulerian-Lagrangian vortex method

We describe and illustrate a new hybrid Eulerian-Lagrangian approach for the incompressible Navier-Stokes equations. The method is based on a decomposition of the computational domain: the near-wall region is resolved using an Eulerian grid based solver, while the Lagrangian vortex element method is used for the wake region. The strengths and the weaknesses of both methods are indeed complementary in such a hybrid approach. The key feature of this new approach is the way the methods are coupled: this is done without iteration.

Ch. 4 Redistribution on hierarchically refined grids

Vortex methods automatically adapt to the flow field: computational elements are present only where they are required and the elements are “naturally” convected with the flow. Nevertheless, this self-adaptivity has to be controlled. In this chapter, a new redistribution approach is



presented where the particles are periodically redistributed on a hierarchically refined cartesian grid. This approach allows to arbitrarily refine the particle resolution where the flow physics require it.

Ch. 5 Flow past a hemisphere using a hierarchically refined grid

The capabilities of the redistribution method on hierarchically refined grids are demonstrated on the flow past a hemisphere at $Re = 3,000$ and using LES. At this Reynolds number, the boundary layer at separation is still laminar, and the transition to turbulence occurs further downstream, in the separated shear layers, via Kelvin-Helmholtz instabilities. A sufficiently fine grid is thus required to accurately capture the shear layers emanating from the hemisphere. However, inside the recirculation bubble and further downstream, where the flow has become fully turbulent, the grid requirements are less demanding. It is therefore a typical case where the ability to control the level of grid refinement is essential.

Ch. 6 LES calculations of spatially evolving wakes in ground effect

This chapter presents results obtained within a European Research project aimed at a better characterization and understanding of aircraft wake vortex turbulence. The case of a longitudinally developing wake generated by a wing is investigated at a moderate (yet significant) Reynolds number, and using LES. The same vortex wake is also investigated in a longitudinally uniform configuration (i.e., a temporal simulation with periodic boundary conditions). Finally, the 2-D (thus necessarily time developing) DNS of the same case is also performed, as a reference. The three simulations are analyzed and compared. These results show an application of vortex methods to a complex, turbulent, “real-life” problem.

Ch. 7 Investigation of a hybrid RANS-LES approach

This chapter corresponds to work performed as a BAEF (Belgian American Educational Foundation) research fellow visiting the Flow Physics and Computation division at Stanford University. This work was also done in collaboration with G. Kalitzin, G. Medic and J. Templeton.

In this chapter, we focus on turbulence modeling in the near-wall region (using a classical grid-based velocity-pressure numerical method). Even with the currently available computational resources, most engineering type applications are still out of reach for wall-resolved large-eddy simulation. Several approaches exist in order to reduce the cost of full LES. We here further investigate a novel concept for the near-wall treatment

of LES that was originally presented in [50, 63, 91]. In this approach a RANS-like eddy viscosity is imposed in the near-wall region and is corrected with the resolved turbulent stress obtained from the LES.

The next section provides a general introduction to the Lagrangian vortex method and the related tools used throughout this work. An excellent general overview of Lagrangian vortex methods and their theoretical background can be found in Cottet and Koumoutsakos [19] and Winckelmans [98].

1.3 Vortex and boundary element method

We here briefly describe the three-dimensional vortex and boundary element method for DNS of bluff-body flows as it was developed and implemented by Ploumhans *et al.* [78, 76]. For more details, refer to these articles and references therein. It is this approach and its implementation that was used as starting point for our work.

1.3.1 Basic equations

Three-dimensional incompressible flows are governed by the vorticity equation

$$\frac{D\boldsymbol{\omega}}{Dt} = \frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} = (\nabla \mathbf{u}) \cdot \boldsymbol{\omega} + \nu \nabla^2 \boldsymbol{\omega} , \quad (1.1)$$

where $\mathbf{u}(\mathbf{x}, t)$ is the velocity field, ν is the kinematic viscosity, and $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ is the vorticity. In viscous flows, one must satisfy the no-slip boundary condition on solid surfaces. In the vorticity formulation, Eq. (1.1), boundary conditions on the three components of the vorticity are needed: a Dirichlet condition on the normal component of $\boldsymbol{\omega}$:

$$\omega_n \triangleq \boldsymbol{\omega} \cdot \mathbf{n} = 0 , \quad (1.2)$$

(as the tangential derivatives of the velocity vanish at the wall), and a Neumann condition on the two tangential components of $\boldsymbol{\omega}$ (expressing the cancellation of the tangential components of the slip velocity at the wall). The velocity can be expressed as

$$\mathbf{u} = \nabla \times \boldsymbol{\psi} + \mathbf{U}_\infty , \quad (1.3)$$

with $\mathbf{U}_\infty$ the free stream velocity and $\boldsymbol{\psi}$ the stream function related to $\boldsymbol{\omega}$ by

$$\nabla^2 \boldsymbol{\psi} = -\boldsymbol{\omega} . \quad (1.4)$$



1.3.2 Vortex method for 3-D flows

The vorticity field is represented by a set of N Lagrangian, vector-valued, particles:

$$\tilde{\boldsymbol{\omega}}(\mathbf{x}, t) = \sum_{i=1}^N \zeta_i(\mathbf{x} - \mathbf{x}_i) \boldsymbol{\alpha}_i, \quad (1.5)$$

where the particles have positions $\mathbf{x}_i$, and vector strengths, $\boldsymbol{\alpha}_i$, that correspond to the integral of vorticity over the volume, V_i , associated to particle: $\boldsymbol{\alpha}_i = \int_{V_i} \boldsymbol{\omega} dV \triangleq \boldsymbol{\omega}_i V_i$. The distribution function, ζ_i , associated with each particle is defined by

$$\zeta_i(\mathbf{x}) = \frac{1}{\sigma_i^3} \zeta\left(\frac{|\mathbf{x}|}{\sigma_i}\right) \quad (1.6)$$

where σ_i is the smoothing parameter. The Gaussian distribution is here used throughout: $\zeta(\rho) = 1/(2\pi)^{3/2} \exp(-\rho^2/2)$. The smoothing parameter is set to $\sigma_i/h_i = 1$, where h_i is the local distance between particles (i.e., the local grid size at redistribution: see below).

The velocity field is computed from Eq. (1.5) as the curl of the stream function which solves $\nabla^2 \tilde{\psi} = -\tilde{\boldsymbol{\omega}}$. Defining $G(\rho)$ such that

$$\nabla^2 G(\rho) = -\zeta(\rho), \quad (1.7)$$

one obtains

$$\tilde{\psi}(\mathbf{x}, t) = \sum_{i=1}^N G_i(\mathbf{x} - \mathbf{x}_i) \boldsymbol{\alpha}_i, \quad (1.8)$$

where $G_i(\mathbf{x} - \mathbf{x}_i) = G(|\mathbf{x} - \mathbf{x}_i|/\sigma_i)/\sigma_i$. The velocity, $\mathbf{u}$, is obtained as:

$$\mathbf{u}(\mathbf{x}, t) = - \sum_{j=1}^N \frac{q(|\mathbf{x} - \mathbf{x}_j|/\sigma_j)}{|\mathbf{x} - \mathbf{x}_j|^3} (\mathbf{x} - \mathbf{x}_j) \times \boldsymbol{\alpha}_j, \quad (1.9)$$

with $q(\rho) = \int_0^\rho \zeta(s) s^2 ds$.

It is worth noting that the particle field, Eq. (1.5), is not generally divergence free. The issue of the non-zero divergence of $\tilde{\boldsymbol{\omega}}$, and of its reprojection onto a divergence free field, will be addressed several times throughout this work.

One time step of the method can be summarized as follows

1. Compute the velocity field, $\mathbf{u}$, and its gradient, $\nabla \mathbf{u}$, from $\boldsymbol{\omega}$, using a Green's functions approach (Biot-Savart, Eq. (1.9)). This is done ef-

ficiently using fast $\mathcal{O}(N \log N)$ tree algorithms based on multipole expansions [6, 42], with active error control [87, 101]. Furthermore, these algorithms have been implemented on parallel computers, thus making it possible to handle very large N [87, 101].

2. Update the particles' position and strength, for the time step, Δt ,

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{u}|_i \quad (1.10)$$

$$\frac{d\boldsymbol{\alpha}_i}{dt} = \nabla \mathbf{u}|_i \cdot \boldsymbol{\alpha}_i + \text{PSE}|_i \quad (1.11)$$

where the Particle Strength Exchange (PSE, see below) scheme is used to accurately treat the viscous diffusion by exchanging strength among neighbor particles [36, 100].

3. Enforce the no-slip condition at the wall by imposing a vorticity flux at each point on the solid surface (see below) [56, 55, 78].
4. Redistribute the particles, when necessary, using a high order scheme: the old set of particles is replaced by a new one in which the particles are on a regular lattice. This eliminates the time-developing errors of the Lagrangian method.

Particle strength exchange

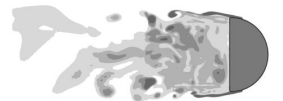
The treatment of viscous diffusion is based on the technique of particle strength exchange (PSE, Degond and Mas-Gallic [36]). In this algorithm, the laplacian operator ∇^2 is approximated by an integral operator,

$$\nabla^2 \omega(\mathbf{x}) \approx \frac{2}{\sigma^2} \int \eta_\sigma(\mathbf{x} - \mathbf{y}) (\omega(\mathbf{y}) - \omega(\mathbf{x})) d\mathbf{y} \quad (1.12)$$

with $\eta_\sigma(\mathbf{x}) = \eta(|\mathbf{x}|/\sigma)/\sigma^3$ and $\eta(s) = -\frac{1}{s} \frac{d}{ds} \zeta(s)$. The Gaussian smoothing is such that $\eta_\sigma = \zeta_\sigma$. The integral operator in Eq. (1.12) is discretized using the particles, and the evolution equation for the particle strengths becomes

$$\begin{aligned} \left. \frac{d\boldsymbol{\alpha}_i}{dt} \right|_{\text{PSE}} &= \frac{2\nu}{\sigma^2} \sum_{j=1}^N (V_i \boldsymbol{\alpha}_j - V_j \boldsymbol{\alpha}_i) \eta_\sigma(\mathbf{x}_i - \mathbf{x}_j) \\ &\simeq \frac{2\nu}{\sigma^2} \sum_{j \in \mathcal{P}_i} (V_i \boldsymbol{\alpha}_j - V_j \boldsymbol{\alpha}_i) \eta_\sigma(\mathbf{x}_i - \mathbf{x}_j). \end{aligned} \quad (1.13)$$

In practice, the right-hand side of Eq. (1.13) does not involve a contribution from all particles, as the rapid decrease of η_σ means that only the particles close



to particle i contribute significantly to $d\alpha_i/dt$. The subset of such particles is denoted $\mathcal{P}_i$.

Enforcement of the no-slip boundary condition

After the convection and diffusion step, there is a spurious slip velocity, $\Delta \mathbf{u}_{\text{slip}}$ at the boundary. The vortex sheet, $\Delta \gamma$, necessary to cancel this slip velocity is then computed using the boundary element method. The boundary surface is discretized using M boundary elements (i.e., “vortex sheet panels”) of uniform unknown strength. For each panel on the body, the mean slip velocity underneath that panel is taken as the average, over the panel, of the velocity induced by the freestream and all vortex particles.

Once the slip velocity has been evaluated for all panels, obtaining the panel strengths so as to cancel this slip velocity amounts to solving the following integral equation:

$$\frac{1}{2} \Delta \gamma(\mathbf{x}) \times \mathbf{n} + \frac{1}{4\pi} \int_S \frac{1}{|\mathbf{x} - \mathbf{x}'|^3} \cdot (\mathbf{x} - \mathbf{x}') \times \Delta \gamma(\mathbf{x}') d\mathbf{x}' = \Delta \mathbf{u}_{\text{slip}}, \quad (1.14)$$

where $\mathbf{n}$ points inside the fluid. The boundary element method transforms equation Eq. (1.14) in a linear system of the form $A \Delta \gamma = b$ where A is full since, in addition to inducing a tangential velocity underneath themselves, the panels induce a velocity on one another. The system is solved using an iterative method, with fast matrix-vector multiply at each iteration ($\mathcal{O}(M \log M)$ operations, using the fast multipole method) [101, 78].

The total flux to be emitted into the flow, for the other substep of the diffusion process, is then given by $\nu \frac{\partial \omega}{\partial n} = \frac{\delta \gamma}{\delta t}$. This flux must be emitted during a time δt . In effect, the vortex sheet $\delta \gamma$ must be distributed to neighbor particles by discretizing the Green’s integral for the inhomogeneous Neumann problem corresponding to the diffusion equation (see, [56, 55, 77] in 2-D and [79, 78] for its extension to 3-D).

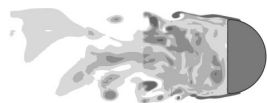
Particle redistribution

In accurate particle methods for viscous flows, one needs to maintain the condition that particle cores overlap. This calls for a particle redistribution scheme. It consists in replacing the distorted set of vortex particles by a new set where the particles are, again, located on an $h \times h \times h$ lattice.

The redistribution is performed using high order interpolation schemes, in such a way that a number of moments of the vorticity fields are conserved (usual up to order 2 or 3). The definition of usual redistribution schemes can be found in [19, 98].

The redistribution lattice needs not be uniform: it can stretch, as long as the stretching rate remains moderate. This is most useful, as it allows to tailor the spatial resolution of simulations. The redistributions schemes are then applied within the uniform index space. This does not conserve exactly the moments; yet, with moderate stretching rate, the behavior of the method is still acceptable (as redistribution is a local operator).

For the technique used to perform redistribution in the presence of solid boundaries that intersect the redistribution lattice in an arbitrary way, see also Ploumhans and Winckelmans [77] in 2-D and Ploumhans *et al.* [78] in 3-D.



CHAPTER 2

Vortex methods with prescribed separation regions

2.1 Introduction

In this chapter, we focus on the ability of the vortex and boundary element method to simulate the unsteady aerodynamics of high Reynolds number flows with massive separation. At higher Reynolds number, the boundary layers and the turbulent wake can no longer be resolved: modeling and large-eddy simulation need to be introduced.

Flows with massive separation can essentially be categorized as follows: at one end, we have flows where the separation points are approximately known a priori (such as the flow past a hemisphere), and, at the other end, we have flows where the evolution of the boundary layer has to be simulated in order to determine where the separation dynamically occurs (such as the flow past a sphere). We here investigate simplified approaches that are well suited for the first kind of flows. This allows to let aside, for the moment, one of the essential drawbacks of the vortex element method: its inability to handle efficiently boundary layer regions. The basic idea here is that, for these flows, it should



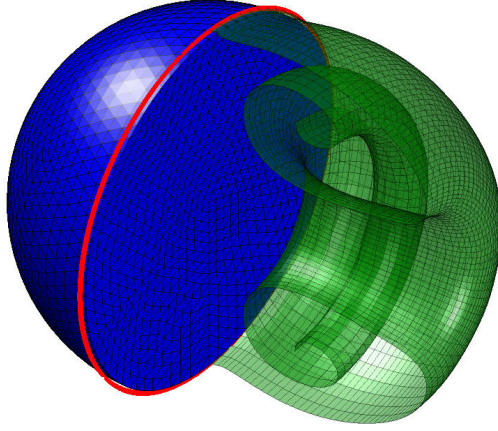


FIGURE 2.1 – Infinitely thin boundary layers. Attached vorticity (blue), prescribed separation line (red) and free wake vorticity (green).

suffice to inject the wake vorticity along the separation line to model the flow correctly. We can thus investigate the capabilities of the vortex method to handle high Reynolds number turbulent wake flows without actually resolving the detailed evolution of the boundary layer, and still obtain the essential physical features of the flow.

Three approaches are considered. In Section 2.2, the entire body is modeled using infinitely thin boundary layers and wake vorticity is explicitly injected in the flow along a user defined separation line. In Section 2.3 an opposite approach is investigated as, in this case, the boundary layers are under-resolved and artificially thick. For both cases, the method is applied to the flow past a hemisphere. Finally, in Section 2.4, a combination of those two approaches is used. This approach is used to investigate the flow past a simplified truck geometry.

2.2 Infinitely thin boundary layers

As we wish to simulate high Reynolds number flows, a natural approach is to introduce infinitely thin boundary layers: the attached vorticity remains captured by the vortex panels on the body surface, while vortex particles are used to capture the wake dynamics. The wake vorticity is produced along a

prescribed separation line by shedding the attached boundary layer vorticity into the flow. Figure 2.1 illustrates this idea for the flow past an impulsively started hemisphere.

The typical time step as described in Section 1.3 has to be modified as follows:

1. Compute the fields $(\mathbf{u}, \nabla \mathbf{u})$ induced by the free and the attached vorticity;
2. Convect and stretch the free vorticity elements (i.e., the vortex particles) according to the local velocity and velocity gradient;
3. Compute the attached vorticity in order to satisfy the boundary condition (i.e., use the BEM to obtain the vortex panel strength);
4. Shed the proper amount of vorticity into the flow along the prescribed separation line (see below).

Note that, in this approach, no redistribution is performed and no viscous diffusion is taken into account.

To introduce the separated wake vorticity into the flow, a shedding model has to be defined. We here use an approach similar to the method developed by Chua [13] for the flow past a square plate.

The separation line is first discretized using N “shedding edges”. At the beginning of the time step, the attached vortex sheet strength, γ_i , and the external velocity, $\mathbf{u}_i$, is evaluated at the location of each shedding edge i . During the time step, shedding is considered to have occurred and a vortex sheet has been convected into the flow. This is illustrated in Fig. 2.2.

At the end of the time step, the total vorticity (i.e., $\boldsymbol{\alpha}_i = \int \boldsymbol{\omega} dV$) shed into the flow, at shedding edge i , is given by:

$$\boldsymbol{\alpha}_i = \gamma_i h_i \frac{|\mathbf{u}_i|_{\text{norm}}}{2} \Delta t, \quad (2.1)$$

where, $|\mathbf{u}_i|_{\text{norm}}$ is the magnitude of the velocity components normal to the shedding edge i and, h_i is the length of the shedding edge i . This strength is assigned to a new vortex particle located at (Fig. 2.3):

$$\mathbf{x}_p = \mathbf{x}_i + \frac{\mathbf{u}_i}{2} \Delta t. \quad (2.2)$$

After the new wake particles are created, the solenoidal condition must be enforced (Fig. 2.3). To do this, attached vortex particles with opposed



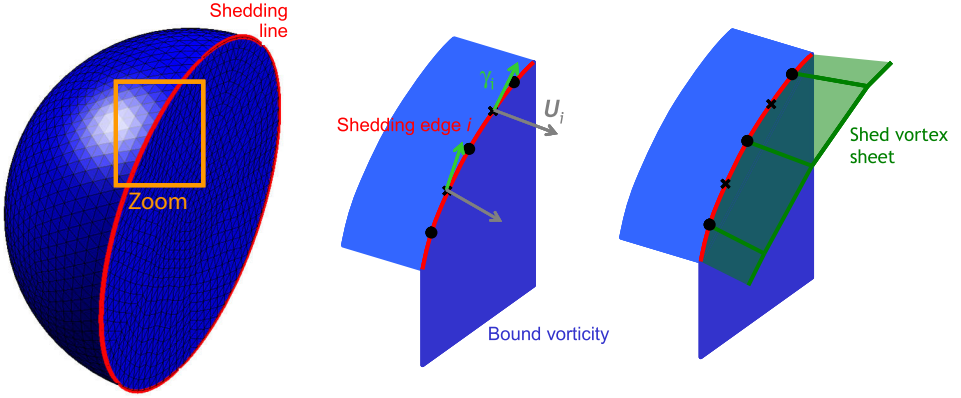


FIGURE 2.2 – Shedding model for the flow past a hemisphere. **Left:** Attached vortex panels (blue) and prescribed shedding line (red). **Middle:** Discretized shedding line with local external velocity, $\mathbf{u}_i$, and boundary layer strength, γ_i . **Right:** Attached vorticity (blue) and shed vorticity (green).

strength are placed at each shedding edge. To complete the loop, particles with orthogonal strengths to $\boldsymbol{\alpha}_i$ are shed into the flow. Their magnitude is reduced by d_i/h_i , where $d_i = |\mathbf{u}_i| \Delta t$ is the outward displacement at shedding location i over the time step. Finally, at each time step, the attached vortex particles from the previous step are lumped with the new wake particles shed into the flow.

Simulations of the flow past a hemisphere for short times after the impulsive start were performed. The time evolution for the case at 0° angle of attack is shown in Fig. 2.4. The number of shedding edges for that case is $N_{\text{shed}} = 80$ (thus $h_{\text{shed}}/D = 8.01 \cdot 10^{-2}$). The particle regularization length is here set to $\sigma = h_{\text{shed}}$. This determines the effective thickness of the shed wake vortex sheet. As can be seen from Fig. 2.4, the strength of the separated vortex sheet is initially high due to the strong flow acceleration past the sharp edge for the initial potential flow around the hemisphere. This strong separated vorticity region forms the core of the vortex ring that is generated behind the hemisphere. At the same time, the wake vorticity begins to interact with the back face of the hemisphere. These two events eventually lead to the numerical blow-up of the simulation, due to the absence of viscous diffusion in this approach.

For short time simulations, the method is able to reproduce the correct physical behavior, at least from a qualitative point of view. For longer times,

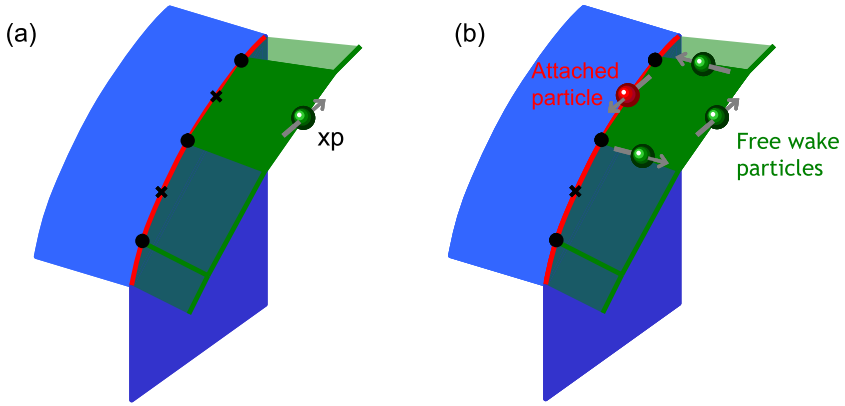
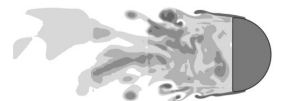


FIGURE 2.3 – Vorticity generation along shedding line. **Left:** Injection of shed vorticity by creating a new wake vortex particle. **Right:** Enforcement of solenoidal condition by creating closed vortex loops.

several intrinsic drawbacks of the method eventually lead to numerical blowup:

- At $T = tU_\infty/D = 4$, the downstream part of the recirculation region becomes unstable due to the stagnation flow there. The absence of viscous diffusion in this approach prevents viscous reconnection and thus dissipation at small scales.
- Free vortex rings hit the back face of the body, as can be seen at $T = 5$ in Fig. 2.4. In order to satisfy the boundary condition, bound vortex rings of opposite circulation are then created at the wall. In a “real” flow, due to the viscous interaction, these vortex rings would partially annihilate each other, through secondary separation. Our simple method cannot capture this phenomena.
- Nothing is done to prevent particle clustering and spreading. This could however be corrected through the use of a redistribution scheme.
- No subgrid-scale turbulence model is used.

The case at 10° angle of attack was also investigated and is shown in Fig. 2.5. Here, the initial vortex ring is ejected before strong interaction with the back face of the hemisphere occur. However, a second strong vortex structure is formed and it also eventually leads to numerical blow-up.



The shedding model was also applied to the case of lifting surfaces. In this case the detailed flow past the airfoil cross section is modeled using Prandtl’s lifting line theory. Results for the flow past a wind turbine and an airfoil are presented in Appendix A.

2.3 “Under-resolved” DNS mode

In this approach, the vortex and boundary element method, as described in Section 1.3 to perform DNS of bluff-body flows, is modified so as to work in under-resolved mode: the boundary layers are captured by a single layer of particles around the body, and the resolution in the wake is too coarse to perform DNS. However, even in this case, the vortex and boundary element method ensures that the correct amount of circulation is carried by the particles. Thus, while producing artificially thick boundary layers, the proper amount of vorticity is still shed at the main separation line. Furthermore, compared to the previous approach (Section 2.2), it has the advantage that it also allows for the wake vorticity to interact in a viscous way with the back of the body: an essential ingredient for long time and physical simulations.

In contrast to the DNS method, we here use a low order redistribution scheme (Λ_1): only the moments of order zero and one of the vorticity distribution are conserved (thus circulation and linear impulse). It can be shown that the Λ_1 scheme has a viscous-like behavior. The redistribution scheme is thus used as a (yet very crude) LES subgrid-scale (SGS) model. The diffusion effect introduced by a Λ_1 redistribution applied every n time steps ($n \Delta t$ being the time between two redistributions) is roughly equivalent to an effective viscosity:

$$\nu_{\text{eff}} \simeq \frac{h^2}{12 n \Delta t}, \quad (2.3)$$

where h is the local grid size and Δt the time step. The Λ_1 -redistribution can here thus be seen as a periodic filter which eliminates the energy that gets accumulated in the small scales captured by the computation. Notice that the dissipation scales with the local grid size (as any SGS model should do). On the other hand, this effective viscosity is not sensitive to the local flow structure: it is active everywhere (even in laminar regions) indeed a crude SGS model.

A new redistribution mapping was introduced, in order to have a smooth transition from a constant $h \times h \times h$ lattice to an exponentially growing lattice (Fig. 2.6). The definition of this mapping is given in Appendix B. This ensures

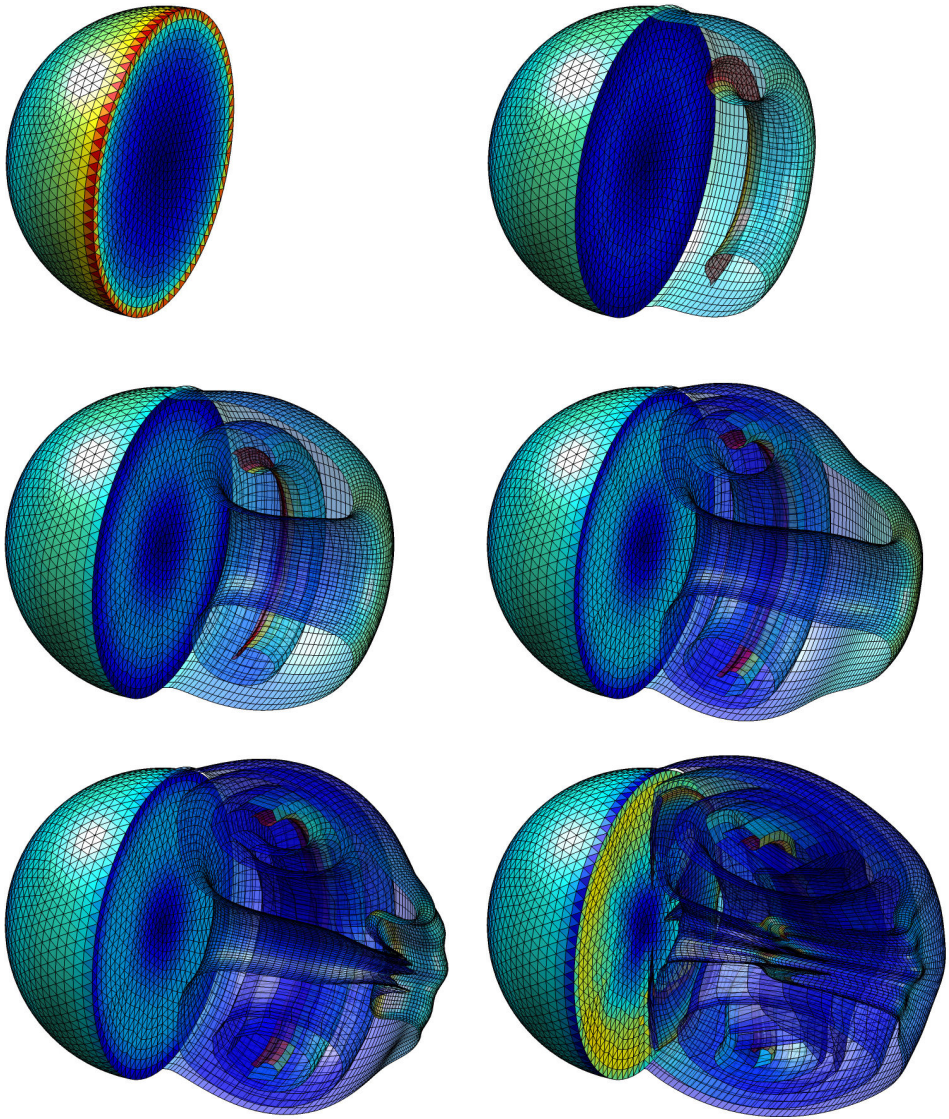


FIGURE 2.4 – Flow past an impulsively started hemisphere at 0° angle of attack. Visualization of the generated wake vortex sheet and the attached vorticity (colored by vortex sheet strength). **From left to right and top to bottom:** $T = \{0, 1, 2, 3, 4, 5\}$.



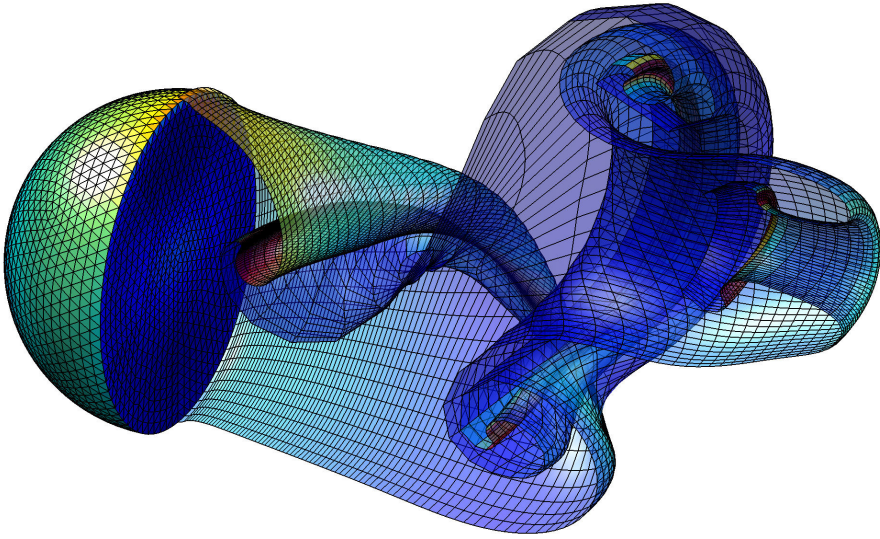


FIGURE 2.5 – Flow past an impulsively started hemisphere at 10° angle of attack. Visualization of the generated wake vortex sheet and the attached vorticity (colored by vortex sheet strength) at $T = 6$.

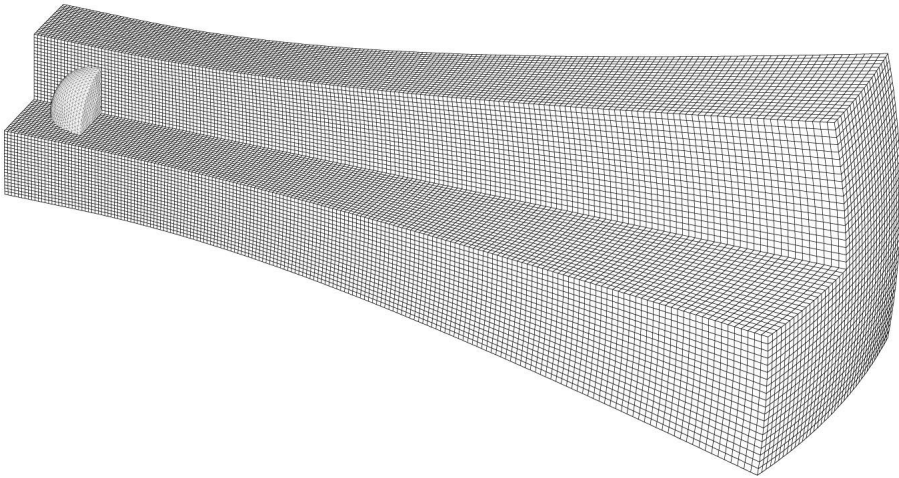


FIGURE 2.6 – Redistribution mapping and hemisphere discretization with $M = 3840$ triangles. The resolution shown is the same as that used in the computation.

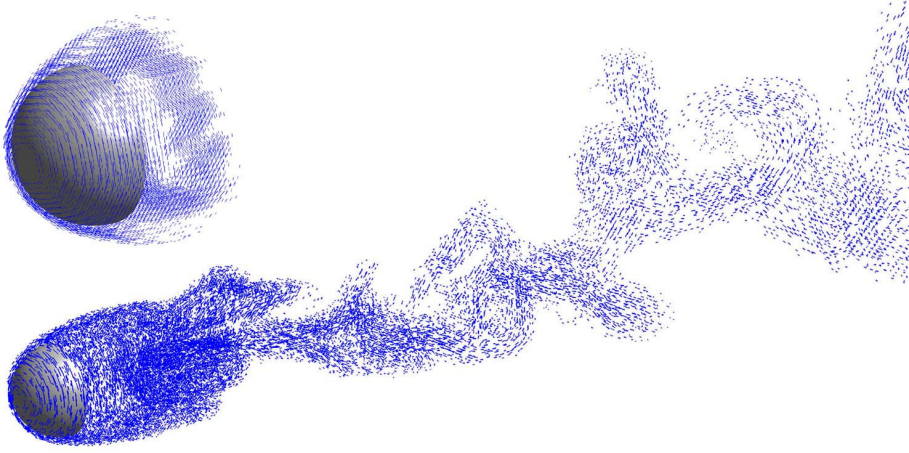
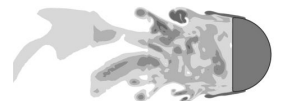


FIGURE 2.7 – Flow past a hemisphere with the DNS code in “under-resolved” mode. **Top:** view of the vortex particles and their strength at an early stage of the flow, $T = 6$ (about 8% of the total number of particles are shown). **Bottom:** view of the developed turbulent flow at $T = 70$ (about 12% of the total number of particles are shown).

that particles have a constant size in the vicinity of the body and it prevents a rapid drop of the resolution in the near wake. Several tests have also shown that the use of a “vibrating” lattice (at each redistribution, the center of the lattice is “moved” randomly around its nominal position) reduces the influence of the arbitrary intersection between the redistribution lattice and the hemisphere (see also [77] for a similar method used in DNS).

Although quite crude, the method enabled us to perform long time simulations ($T = tU_\infty/D = 70$) with a relatively low number of particles. The redistribution lattice and the body surface mesh used in these computations are shown in Fig. 2.6. The grid size in the vicinity of the hemisphere is $h/D \simeq 0.036$. The time step for the simulation was $\Delta T = 0.02$. Redistribution was performed every 5 time steps. The number of particles went from $\sim 13\,000$ ($T = 0$) to $\sim 440\,000$ ($T = 70$). The total run time was 120 hours on 8 processors of a DEC-Alpha ev67 system.

Defining a Reynolds number for these computations has little meaning. We can however mention that the explicit viscosity, taken into account by the PSE scheme, was $\nu/(U_\infty D) = 10^{-4}$ (thus $Re = 10^4$) while the SGS viscosity, introduced by the redistribution (acting as SGS model), was $\nu_{\text{eff}}/(U_\infty D) \approx$



10^{-3} in the vicinity of the body.

A visualization of the flow is provided in Fig. 2.7: particles of strength, $|\alpha_i|$, above an arbitrary threshold are shown along with their vorticity vector. During the early stage ($T = 6$), one can see the vortex ring becoming unstable. The developed turbulent wake exhibits hairpin like vortex structures, as also observed in experiments.

From a quantitative point of view, the mean drag coefficient was obtained from the time derivative of the linear impulse (Fig. 2.8) over the interval $T_1 = 20$ to $T_2 = 60$ (thus over a significant time after the wake is well developed):

$$C_D = \frac{2}{U_\infty^2 A} \frac{(I_x(T_2) - I_x(T_1))}{(T_2 - T_1)} = 0.57$$

Except for some approximate drag coefficient values found in engineering text books, there are very few detailed studies for the flow past the hemisphere (neither experimental, nor numerical) in the literature. Engineering text books give the drag of the hemisphere at $C_D \sim [0.4, 0.5]$ for $Re = 10^4..10^5$ (e.g., [46]). Hoerner [44] provides the approximate drag coefficient for several bluff bodies. The hemisphere is reported at $C_D = 0.42$ for $10^4 < Re < 10^6$. In Chapter 5, we use a high resolution grid along with an advanced SGS model for the flow past a hemisphere at $Re = 3,000$. The average velocity profiles obtained from this simulation are then in very good agreement with experimental results by Willemart [97] in our experimental facilities. The drag coefficient obtained from this simulation is $C_D = 0.50$. Finally, Cottin (private communication) measured $C_D = 0.434 \pm 8\%$ and $C_D = 0.469 \pm 6\%$ respectively at $Re = 55,000$ and $Re = 110,000$ in our low speed, open-section, wind tunnel.

Figure 2.8 provides averaged velocity defect profiles at different downstream stations. The first and second stations, one and two diameters downstream the back face of the hemisphere (top curves in Fig. 2.8), are located inside the recirculation bubble as the velocity defects on the centerline are greater than 1. Further downstream, the wake spreads and decays as expected. Recall that, as opposed to the mixing layer and the jet, a wake flow cannot become exactly self-similar: it only becomes self similar in the limit of the very far wake, i.e., when the velocity defect on the centerline (U_s) tends to zero, see also [47]. Typically, this condition requires $x/D > 50$ [47, 93]. In the present simulation, we could not establish statistically converged velocity profiles for $x/D > 10$. Self similarity was tested in the range $x = [5, 10]$: as expected, it is not realized yet. To further investigate this point, one would need to perform a simulation

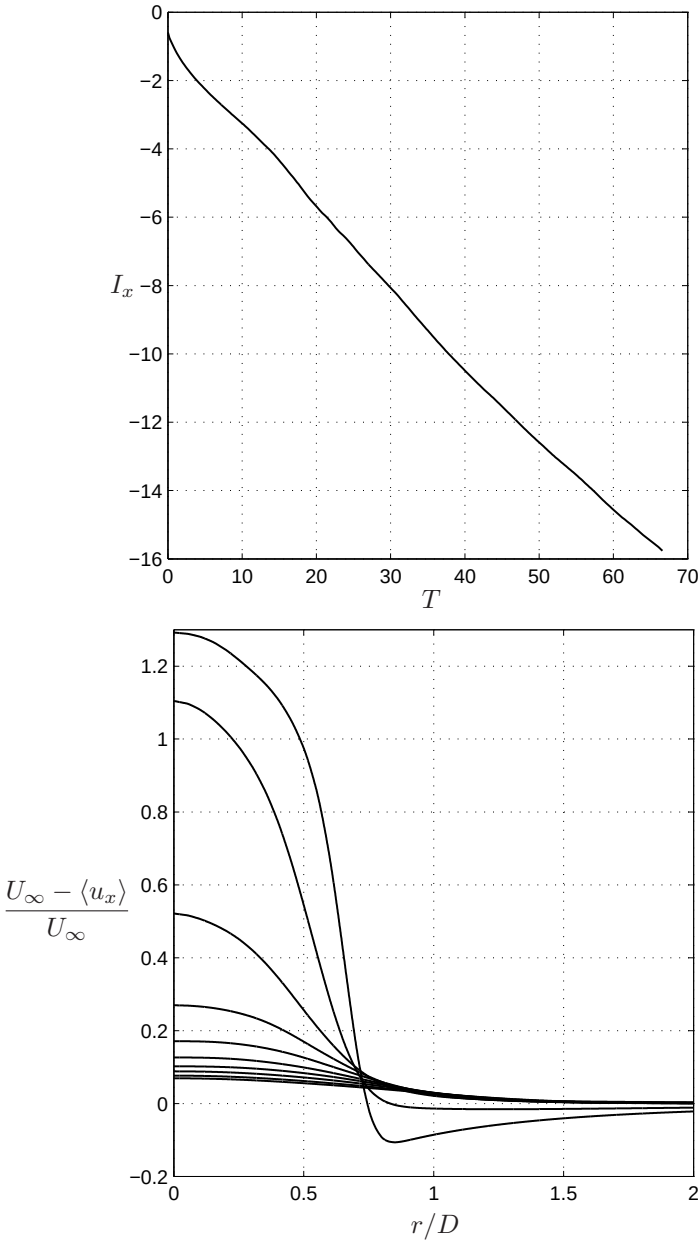
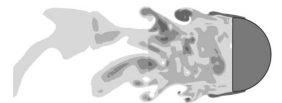


FIGURE 2.8 – **Left:** x -component of the linear impulse as a function of time (the developed turbulent wake corresponds to $T \geq 20$). **Right:** Mean velocity defect profiles (averaged over time and over azimuthal angle) behind the hemisphere at different downstream stations ($x/D = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$).



capturing sufficiently well the very far wake region (and also for sufficiently long times); this would be computationally quite demanding.

From a quantitative point of view, the usefulness of these results is limited: the mean drag coefficient is over-predicted by at least 10%. This is quite disappointing since such bluff body flows (with sharp back face and thus well-defined separation) have C_D independent of the Reynolds number when it is high enough (typically $Re \gtrsim 10^4$). The fact is that, due the low order redistribution scheme, the effective Reynolds of this simulation is much lower. The lack of quantitative agreement with actual global averaged statistics is further confirmed by the mean velocity profiles. Indeed, the recirculation length is over-predicted when compared to results presented in Chapter 5: we here obtain $x/D \simeq 2.1$, while $x/D = 1.47$ is obtained from the advanced numerical simulation at $Re = 3,000$ and $x/D \simeq 1.6$ from the measured mean velocity profiles at $Re = 55,000$.

Nevertheless, these results indicate that the simplified approach is able to capture wake dynamics with a reduced number of computational elements: a significant fraction of the turbulent and coherent large scale wake dynamics (including the hairpin like vortices, and interaction with the back face of the hemisphere) are already captured. These results also show that, in order to predict high Reynolds number flows with massive separation, good subgrid-scale models have to be introduced (this also calls for high order redistribution schemes) and the boundary layer regions have to be better captured. Finally, there is also a minimum spatial grid resolution that is required to resolve a sufficiently large fraction of the turbulent flow features in the wake. In this case, with about 30 grid points across the diameter of the hemisphere, this requirement is probably not met (no matter how good the SGS model or boundary layer modeling would have been).

2.4 A combined approach

In this section, a combined approach is investigated: the attached boundary layer regions are captured using infinitely thin vortex sheets (as in Section 2.2) while the Lagrangian vortex particles are used to capture the vorticity dynamics in the separation regions and in the wake, including the viscous interaction with the solid walls in the separated regions (as in Section 2.3). This approach was developed following a collaboration with Professor Leonard's group at the

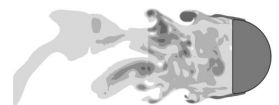
California Institute of Technology to investigate the flow past a generic and simplified “truck” shape, the Ground Transportation System (GTS, [26]). The GTS was designed as a geometry for the validation of numerical simulations versus experimental measurements.

Recently, several research groups have investigated this flow, using essentially Reynolds-Averaged Navier-Stokes (RANS) approaches. The conclusion is (not surprisingly) that the RANS approach performs well for the attached boundary layer regions but it does not capture well the details of the separated regions (i.e., the wake region behind the GTS). Even if the RANS simulations manage to produce a reasonable global drag coefficient, being able to capture the wake flow dynamics is still essential. This is critical when it comes to computing the potential effect of proposed drag reduction devices which operate in the wake region. This is also critical when one tries to characterize, by simulation, the impact of the unsteady wake vortical structures on a crossing or follower vehicle.

The inherent shortcomings of RANS approaches for these types of flows are, in principle, addressed by a Large-Eddy Simulation (LES) approach. The argument is that, in LES, the proper medium and large scale dynamics are well captured by the simulation, and that only the effect of the small scale turbulence (the subgrid scales: those not captured) needs to be modeled. However when it comes to LES of large-scale wall-bounded flows, the computational cost is still too high. This is because “wall-resolved” LES still requires a fine grid in the near-wall regions: fine in the normal direction (as in RANS) but also quite fine in the other two directions (and significantly finer than for RANS).

Nowadays, the research efforts are being put on developing and validating hybrid approaches: RANS near the walls, LES away from the wall. We here consider an even simpler approach: an “ad hoc” approach, of engineering type, for flows where the separation lines are determined a priori, and with the assumption that all boundary layers upstream are attached. This is thus, in some sense, the idea of Detached-Eddy Simulation (DES) pushed to the limit.

This section is organized as follows. First, we present the developments of VEM and BEM specific to the present approach. The method is then applied to the flow past the GTS and with realistic ground effects: the GTS is moving over the ground. This is a long time simulation, in order to obtain sufficiently converged mean velocity fields. The results are also compared with experimental results (wind tunnel) and RANS simulations. Unfortunately, those correspond to unrealistic ground effects: the GTS was fixed to the ground.



2.4.1 Computational approach

Attached and separation regions

The unsteady wake region is here captured using the vortex element method (VEM), while the vorticity generated at the wall is captured using the boundary element method (BEM, here vortex sheets discretized using panels): the BEM is used for the “attached” boundary layer regions and for the “diffusing/separating” boundary layer regions. Here, the user explicitly “tags” the regions where the flow is attached and the regions where separation and complex viscous interactions are allowed.

Let us consider a set of M panels, discretizing the body surface, and N vortex particles. Let $\mathcal{P}_{\text{att}}$ be the subset of “attached”-tagged vortex panels and $\mathcal{P}_{\text{diff}}$ the “diffusing”-tagged panels (i.e. where separation is allowed).

One time step then goes as follows:

1. Compute the velocity field, $\mathbf{u}$, and its gradient, $\nabla\mathbf{u}$, at particle locations from $\boldsymbol{\omega}$, the vorticity carried by the vortex particles and $\boldsymbol{\gamma}$, the vorticity carried by the subset of attached vortex panels, $\mathcal{P}_{\text{att}}$.
2. Update the particles’ position, $\mathbf{x}_i$, and strength, $\boldsymbol{\alpha}_i$, for the time step, Δt ,

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{u}|_i, \quad \frac{d\boldsymbol{\alpha}_i}{dt} = \nabla\mathbf{u}|_i \cdot \boldsymbol{\alpha}_i + \text{PSE}|_i, \quad (2.4)$$

where the Particle Strength Exchange (PSE) scheme is used to treat the viscous diffusion.

3. Enforce the no-slip boundary condition at the wall:
 - (a) Compute the slip velocity, $\mathbf{u}_{\text{slip}}$, underneath all vortex panels ($\mathcal{P}_{\text{att}} \cup \mathcal{P}_{\text{diff}}$) from the vorticity carried by the vortex particles and $\mathcal{P}_{\text{att}}$.
 - (b) Compute the vortex sheet, $\Delta\boldsymbol{\gamma}$, that cancels the slip velocity at the solid boundary.
 - (c) For panels belonging to $\mathcal{P}_{\text{att}}$, the new panel strength is set to $\boldsymbol{\gamma}_i + \Delta\boldsymbol{\gamma}_i$. For panels in $\mathcal{P}_{\text{diff}}$, the vortex sheet is equivalent to a vorticity flux that must be emitted into the flow for the time Δt : $\nu \frac{\partial \boldsymbol{\omega}}{\partial \mathbf{n}} = \frac{\Delta\boldsymbol{\gamma}}{\Delta t}$ (where ν is the molecular viscosity, and $\mathbf{n}$ is the local outward-pointing normal to the body surface). This amounts to modifying the strength of the particles close to these panels as in [78].

4. Redistribute the particles (when necessary): the old set of particles is replaced by a new one, in which the particles are on a regular lattice.

Progressively diffusing panels

When using “attached” panels, it is obvious that the flow experiences an abrupt transition as it goes from the “attached” vorticity region to the “separation” region. In this transition region, the boundary layer thickness varies from zero to a finite value (magnitude of the local particle size) over a very short distance.

In order to ensure a smooth transition zone, “partial” diffusing panels are used: a diffusion ratio, r , is attributed to each of these panels. The idea is to ensure that a fraction r of the total boundary layer vorticity, γ_{tot} , is carried by the vortex particles while the remaining fraction $(1 - r)$ is still attached to the vortex panels. The diffusion fraction is chosen such that there is a smooth transition from non-diffusing panels ($r = 0$) to totally diffusing panels ($r = 1$).

First, let's define γ_{tot} as the total boundary layer strength after the no-slip enforcement:

$$\gamma_{\text{tot}} = \gamma_{\text{panel}} + \gamma_{\text{part}} + \Delta\gamma, \quad (2.5)$$

where γ_{panel} is the current panel strength, γ_{part} the current strength of the boundary layer fraction discretized by vortex particles (see below) and $\Delta\gamma$ the vortex sheet that cancels the slip velocity at the solid boundary (Step 1b, in the general time step description above).

We can then compute $\Delta\gamma_{\text{part}}$ and $\Delta\gamma_{\text{panel}}$, respectively the variation of the vortex panel strength and the vorticity flux to be emitted onto the vortex particles, such that, for each panel:

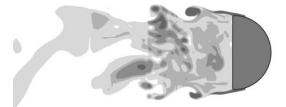
$$\gamma_{\text{part}} + \Delta\gamma_{\text{part}} = r \gamma_{\text{tot}} \quad (2.6)$$

$$\gamma_{\text{panel}} + \Delta\gamma_{\text{panel}} = (1 - r) \gamma_{\text{tot}}. \quad (2.7)$$

Notice that these equations are solved in the local panel coordinate system for the tangential vorticity components only (as the panel's strength normal component must be equal to zero).

For progressive diffusing panels, Step 3c of the general time step is modified as follows: the new panel strength is set to $\gamma_{\text{panel}} + \Delta\gamma_{\text{panel}}$; the vortex sheet $\Delta\gamma_{\text{part}}$ is the vorticity flux that must be emitted into the flow for the time Δt :

$$\nu \frac{\partial \omega}{\partial \mathbf{n}} = \frac{\Delta\gamma_{\text{part}}}{\Delta t}.$$



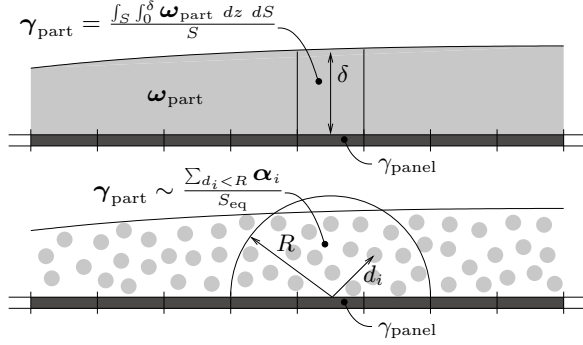


FIGURE 2.9 – Boundary layer strength carried by the vortex particles (γ_{part}) and attached to the vortex panels (γ_{panel}).

Even if γ_{part} , the boundary layer strength carried by the vortex particles above a given panel, is well defined conceptually: $\gamma_{\text{part}} = \int_0^\delta \omega_{\text{part}} dz$ (where δ is the local boundary layer thickness), its evaluation is somewhat arbitrary (Fig. 2.9). First, let's redefine γ_{part} as the averaged boundary layer strength (carried by the vortex particles) over a surface S (ideally the panel's surface):

$$\gamma_{\text{part}} = \frac{\int_S \int_0^\delta \omega_{\text{part}} dz dS}{S}$$

For each panel, we approximate this quantity as the discrete integral of ω_{part} over an hemisphere of radius R centered on the panel divided by an “equivalent” surface S_{eq} :

$$\gamma_{\text{part}} \sim \frac{\sum_{d_i < R} \alpha_i}{S_{\text{eq}}}$$

where d_i is the distance of a particle to the panel center, α_i is its strength, and S_{eq} was chosen equal to πR^2 . It is clear that R should be greater than the local boundary layer thickness, but also remain fairly low in order to obtain a good approximation of the local boundary layer strength.

Auto diffusing panels

If one wants to perform simulations where ground effects are taken into account, ground panels are also necessary in order to enforce the no-slip boundary condition there. Furthermore, the wake vorticity has to interact in a viscous way with the ground. This is true even if one is not interested in the wake-ground

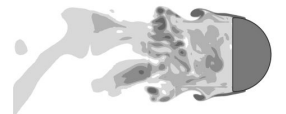
effect (as in the far wake for example), due to the fact that the interaction of wake vorticity with a non-viscous ground rapidly leads to numerical blowup. This observation leads to the conclusion that it is necessary to use “diffusing” ground panels over the whole wake region.

In order to avoid the extra cost of ensuring a few layers of particles above the whole wake region at all times (to capture the diffused panel vorticity), we introduce “automatic” diffusing panels: as soon as vortex particles travel above these panels, their vorticity is diffused onto them (they act as “diffusing panels”); otherwise the panel vorticity remains attached.

Improved panel solver

The vortex sheet $\Delta\gamma$, necessary on the body surface in order to cancel the slip velocity, is found as the solution of a boundary integral equation. The discretization of this boundary integral equation results in a system of $2M$ linear equations for the vortex panels (i.e. find $\Delta\gamma_i$ for each panel) which is solved iteratively. The use of multipole expansions for vortex panels reduces the cost of each iteration to $\mathcal{O}(M \log M)$. The efficiency of this iterative “multipole-based” solver was improved as follows:

- Choice of a good initial guess: the use of $\Delta\gamma_i = -2 \mathbf{n}_i \times \mathbf{u}_i^{\text{slip}}$ (exact solution for coplanar panels) has proved to lead to faster convergence compared to the former method where the previous time step solution is used. This is mainly because the vortex sheet is only a small correction at each time step dominated by high frequency variations as vortex particles come locally close to panels or not. These high frequency variations call for local corrections which are well estimated by the exact solution for coplanar panels.
- Let $\Delta\gamma^r$ be the corrective vortex sheet at iteration r . In order to evaluate $\Delta\gamma^{r+1}$ one must compute the slip velocity, $\mathbf{u}^{\text{slip}^r}$, induced by $\Delta\gamma^r$. Now, instead of computing the slip velocity induced by the whole vortex sheet $\Delta\gamma^r$ at each iteration we can compute the slip velocity induced by $(\Delta\gamma^r - \Delta\gamma^{r-1})$ and simply add it to $\mathbf{u}^{\text{slip}^{r-1}}$ (linear problem). This is advantageous because of the use of multipole expansions: as we converge to the solution, $(\Delta\gamma_i^r - \Delta\gamma_i^{r-1})$ gets smaller and thus each iteration takes less time to compute as we can make a more extensive use of multipole expansions.



These modifications reduced the global computational time for the panel solver by a factor 2 (up to 5 in some cases).

Total circulation

For a physical flow, one must ensure that the total vorticity remains equal to zero for all times. As the global time integration scheme is not conservative, the sum of the particle strengths is set back to zero at each time step, by distributing the difference over all particles.

However, when “attached” vortex panels are used, this simple scheme cannot be used anymore, as one must take the “attached” vorticity into account. Several approaches for enforcing the total vorticity were investigated.

- Our first approach was based on the fact that the vorticity flux at the wall must be divergence free. At a global level, this leads to the following constraint :

$$\int_{S_{\text{body}}} \Delta\gamma \, dS = 0.$$

We track the total amount of circulation diffused onto the vortex particles (Γ_{shed}) over time. The particle strengths are then adjusted to Γ_{shed} at each time step.

Despite the sound physical foundation of this approach, it performs very poorly in time. Actually, the basic hypothesis, $\int_{S_{\text{body}}} \Delta\gamma \, dS = 0$, is true only for an exactly divergence free vorticity field, which is not quite the case in practice. Practically, as we modify $\Delta\gamma$ in order to satisfy the total vorticity constraint, the vortex sheet no longer cancels the slip velocity and particles soon penetrate inside the body, which finally leads to numerical blowup.

- In the second approach, we do not place any constraint on the vortex sheet. We simply ensure, at each time step, that the total vorticity remains equal to zero by setting the sum of the particle strengths to minus the total “attached” vorticity, $-\int_{S_{\text{body}}} \gamma_{\text{attached}} \, dS$ (carried by the “attached” vortex panels).

This approach produces good results, even for long time simulations.

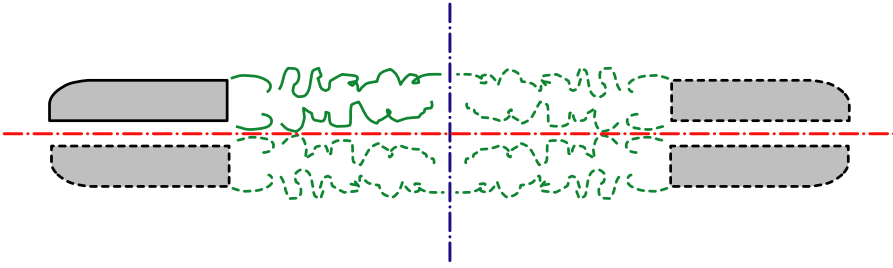


FIGURE 2.10 – Symmetrized vorticity field in order to impose no-through flow on the ground plane and a “natural” outflow condition in the far field.

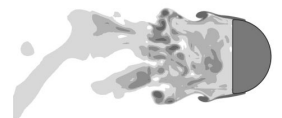
Vorticity flux at the wall for under-resolved boundary layers

When performing simulations where the boundary layers are no longer well resolved, the characteristic diffusion distance becomes small compared to the local particle size. This can become critical as the panel diffusion step must ensure that the total flux of vorticity is distributed to neighbor vortex particles. The use of the classical diffusion scheme with the correction for conservation leads to a very noisy distribution of the vorticity on the neighbor particles. This situation degrades even further for simulations with ground effect where a decreasing resolution is used in the wake.

Actually, in these under-resolved computations, one “simply” wants to transfer the whole vorticity flux to the first layer of particles above the wall. We therefore “relax” the local Reynolds number at the wall with respect to the local grid size to ensure that the vorticity flux can be “sufficiently well” captured. This scheme produces good results because of its diffusive nature, which helps remove the high-frequency noise present in the computed vorticity flux.

Boundary conditions for non-open domains

The vortex element method takes boundary conditions for open domain flows implicitly into account. Combined with the boundary element method, one can simulate external flows past bodies with the exact boundary condition satisfied at infinity. However, for practical simulations in external flow aerodynamics, we may want to impose:



- **Through flow plane.** The most straightforward way to handle an out-flow boundary condition is to discretize the entire wake. This amounts to keeping the vortex wake particles “forever”. For long time simulations however, even though the size of the particles increases downstream (as the size of the particle redistribution lattice is chosen to grow downstream), it leads to an additional and forever increasing computational cost. Another drawback arises from the fact that it can also lead to numerical errors precisely because of the very crude discretization of the far wake. The alternative is to eliminate far-wake particles and to impose a “convection” boundary condition on an outflow plane. This is done by enforcing symmetries on the flow. It leads to the following constraints on the vorticity field (here written for a $y - z$ plane located at $x = 0$):

$$\omega_n(-x, y, z) = -\omega_n(x, y, z) , \quad (2.8)$$

$$\omega_t(-x, y, z) = \omega_t(x, y, z) , \quad (2.9)$$

where ω_n and ω_t are respectively the normal and the tangential components of the vorticity field. This leads to $u_t(0, y, z) = 0$.

- **No-through flow plane.** A ground plane can be treated as a “normal” wall using the boundary element method as described above. However since a ground plane extends to infinity, it necessitates the discretization using vortex panels to extend very far from the body and from the wake. The no-through flow for an infinite plane can also be enforced using flow symmetries. It leads to the following constraints on the vorticity field (here written for a $y - z$ plane located at $x = 0$):

$$\omega_n(-x, y, z) = \omega_n(x, y, z) , \quad (2.10)$$

$$\omega_t(-x, y, z) = -\omega_t(x, y, z) . \quad (2.11)$$

This leads to $u_n(0, y, z) = 0$. Of course, in the regions of the ground plane where viscous interaction with the wake particles is allowed, the vorticity flux from the ground to the particles is still done properly, as explained previously.

Thus, using symmetries, we build a vorticity field such that there is no through flow on the ground plane and natural outflow on a plane far downstream, see Fig. 2.10. In practice, we do not build this symmetrized vorticity field explicitly: each time we make a velocity (and velocity gradient) evaluation at a point x in the computational domain, and due to the vorticity in the computational domain, we also make the evaluation at the three other symmetric

locations (sign must be adjusted depending on symmetry and component) and we simply add the contributions.

The subgrid-scale model

We here use LES, with effective subgrid-scale (SGS) viscosity. This can easily be implemented by using a modified PSE scheme:

$$\frac{d\boldsymbol{\alpha}_i}{dt} = \sum_{j \in \mathcal{N}_i} \frac{2}{\sigma_{ij}^2} (\nu + \nu_{e_{ij}}) (V_i \boldsymbol{\alpha}_j - V_j \boldsymbol{\alpha}_i) \eta_{ij}(|\mathbf{x}_j - \mathbf{x}_i|^2) , \quad (2.12)$$

where $\sigma_{ij}^2 = (\sigma_i^2 + \sigma_j^2)/2$, $\eta_{ij}(r) = \eta(r/\sigma_{ij})/\sigma_{ij}^3$ and $\eta(r)$ is the PSE kernel for the Laplacian operator. The effective viscosity is also symmetrized over the particles i and j : $\nu_{e_{ij}} = (\nu_{e_i} + \nu_{e_j})/2$. The symmetrized operations ensure that the operator remains conservative.

As closure model for the eddy viscosity, the Smagorinsky model was be used here: $\nu_e = C h^2 |S|$ where $|S| = \sqrt{2 S_{ij} S_{ij}}$ and where h is the particle size (recall that it is set by the size of the redistribution lattice).

In practice, the model is implemented as follows. We first evaluate the velocity and velocity gradient at particle locations as usual. The velocity gradient is then used to obtain the local eddy viscosity for all particles, using the norm of the strain rate tensor. A second traversal of the parallel tree code is then performed to evaluate Eq. (2.12) for all particles. The additional computational cost related to SGS modeling arises from this second tree traversal: in DNS-mode this operation is embedded (at almost no cost) in the tree traversal used for evaluating the velocity field.

Particle vorticity field divergence

The vortex particle method has no built-in control for enforcing the vorticity field to remain divergence free as time evolves. When the computation is well resolved, the divergence of the vorticity field tends to “naturally” remain at a fairly low value, even for long time simulations. This has been shown in various DNS results (e.g., for the flow past a sphere at moderate Reynolds number [78]). However, as soon as one wishes to push the resolution towards its limits, the divergence problem becomes a major issue.

In Section 2.3, a low order redistribution scheme was used as a form of SGS model. Due to its highly diffusive nature, it was sufficient to control



the vorticity field divergence. However, if one intends to use an “explicit” SGS model such as the Smagorinsky model used here, one has to use a high order, non-diffusive, redistribution scheme: such as the Λ_3 scheme, which is classically used in DNS computations. As the Λ_3 scheme does not have a viscous behavior, there also is no longer enough dissipation of the small-scale energy: this translates into a rapid increase of the vorticity field’s divergence and finally leads to numerical blowup.

In fact, one can state that an increasing divergence of the vorticity field is a sign that the computation is not well resolved locally and that energy is accumulating in small scales of the flow. It seemed that introducing a SGS stress model would resolve this issue. However, after some testing, this appears not to be the case: SGS modeling certainly improves the situation, but it does not completely remove the divergence problem. This is quite logical when one realizes that the effective “mesh Reynolds number” (here defined based on gradients of the velocity field) corresponding to the Smagorinsky model effective viscosity is, in fact, constant,

$$Re_h == \frac{|S| h^2}{\nu_e} = \frac{1}{C} \approx 40 , \quad (2.13)$$

and not that small. It thus appears that an LES is necessarily somewhat “under-resolved” with respect to its own effective viscosity. Thus, we also need to control the vorticity field divergence when doing LES.

Another simple way to deal with the divergence problem is to introduce a “relaxation scheme” where the particle strengths are set back to the curl of the velocity field evaluated at the particle locations (this field is divergence free). This relaxation procedure is however again a diffusive operation. It should be applied sufficiently scarcely, such that the global diffusive effect remains fairly low. This approach was used in the simulations presented in this section.

A better relaxation procedure would be a “projection” of the vorticity field onto a divergence free space, as it is done in incompressible velocity-pressure algorithms and also in vortex-in-cell methods (VIC), see [23, 24]. This requires solving a Poisson equation to project the vorticity field onto a divergence-free field:

$$\begin{cases} \nabla^2 F &= \nabla \cdot \boldsymbol{\omega}^{\text{old}} \\ \boldsymbol{\omega}^{\text{new}} &= \boldsymbol{\omega}^{\text{old}} - \nabla F \end{cases} \quad (2.14)$$

where $\nabla \cdot \boldsymbol{\omega}$ would here be computed using PSE like operators (as in [39]). The boundary conditions required by this elliptic equation would be imposed using

BEM and/or symmetries, as explained before. We would then obtain F (in fact ∇F) using the fast multipole method. As this projection operation is not diffusive, it could be applied after each redistribution. This approach is applied with great success in Chapter 5.

2.4.2 Flow past the GTS

Numerical setup

As can be seen in Fig. 2.11, most of the GTS was tagged as regions with attached (and infinitely thin) boundary layers. Progressively diffusing panels are used to ensure a smooth transition from the attached boundary layers to the regions where the panels are allowed to diffuse and separate. A close up view of the panel discretization is shown in Fig. 2.11. One can see that a variable panel size is used for computational efficiency: coarse panels can be used in the attached regions, whereas the diffusing (and progressively diffusing) panels' size is chosen equal to the local particle size. On the ground, the panels were only used to enforce the no-slip boundary condition: the no-through flow is enforced using symmetry. Ground panels were tagged as “diffusive” panels in the near wake region while the far wake uses “auto-diffusing” panels. The realistic ground effect was thus also simulated: the GTS is here travelling above a fixed, viscous ground. A pure “convection” outflow boundary condition was enforced at $x = 20W$.

The molecular viscosity was set to $\nu = 10^{-4} U_\infty W$ (U_∞ being the velocity of the GTS and W its width). The Reynolds number is thus 10,000, which is moderate yet sufficiently high for the wake flow to exhibit all characteristics of a turbulent flow. The C value in the LES model was set to the theoretical value derived from isotropic turbulence: $C = 0.025$ (which is consistent with a Kolmogorov constant of about $C_K = 1.6$).

We also use a redistribution mapping that smoothly goes from a uniform lattice (in the near body region) to an exponentially growing lattice (in the far wake). At each redistribution, the center of the mapping is “displaced” randomly around its nominal position: this reduces the influence of the arbitrary intersection between the redistribution lattice and the body. In the near body region, the particle discretization is $h = 0.02W$. The dimensionless time is defined as $T = tU_\infty/W$. The time step was taken as $\Delta T = 0.01$. Particle redistribution was performed every 5 times steps, and using the high order



Λ_3 scheme [19, 98]. The relaxation scheme (by reset of particle strengths to the curl of the local velocity field) was applied every 100 time steps (after a redistribution).

Results

The simulation was carried out for long time: up to $T = 165$. In regime (attained after $T = 40$), there are roughly 1.5 million particles in the simulation. Figures 2.12 to 2.14 show snapshots of the turbulent flow. These results feature much more small structures when compared (qualitatively) to simulations where the Λ_1 redistribution scheme was used (not shown here).

The statistics were accumulated in various planes (vertical and horizontal) at fixed distances downstream of the GTS. The present VEM results, with realistic ground effects (GTS moving above ground), are presented in Figs. 2.16 to 2.19. We also provide, for comparison, experimental results and RANS results, but for the case with unrealistic ground effects (GTS fixed to ground). The comparison cannot be made easily, as the ground condition has a significant effect on the the wake dynamics on thus on the obtained mean profiles. Moreover, the experimental results are only available in a very limited region (shown by the rectangular box and reported in all figures). It nevertheless appears that the recirculation region is shorter for the case with realistic ground effects. As to the amplitude of the max velocity defect in the recirculation region, it is about the same: roughly $1.2U_\infty$. In the case with unrealistic ground effects, it also appears that the recirculation region predicted by RANS is too large compared to the experiment.

We also illustrate, in Fig. 2.20, the effective viscosity of the Smagorinsky subgrid-scale model: instantaneous field and time averaged field. Due to the local $|S|$ scaling, it varies significantly in space. We also notice that, in the mean, it is larger in the high mean gradient regions: the boundary layers, the separating shear layers, and the downstream part of the recirculation region.

For further comparison, we also provide, in Fig. 2.21, a view of instantaneous vorticity fields for the GTS simulated with different ground conditions. The case with GTS fixed to the ground was thus also run using the VEM + BEM method; yet, it was not possible to perform a long time simulation (and thus obtain mean flow statistics), due to the difficulty of handling this case with the method. Problems in capturing the ground boundary layers over large distances downstream and sideways eventually lead to numerical blow up: the VEM +

BEM method is thus clearly far best suited to simulating the real case: that with realistic ground effects. Nevertheless, it is already seen, from Fig. 2.21, that the wake dynamics are quite different, depending on which ground effect is enforced. A ground boundary layer is necessarily present when the GTS is fixed to the ground, whereas none is present when the GTS moves above the ground. In absence of such boundary layer, the vortex wake tends to have a shorter mean recirculation region; its spreading rate and its dissipation are also much affected.

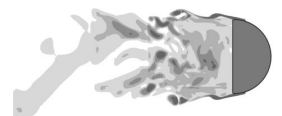
It would certainly be very useful if one were to conduct new experiments with realistic ground effects (e.g., using a wind tunnel with moving belts) and with a larger window of measurements.

2.4.3 Conclusion

The flow past the GTS, at moderate Reynolds number and using a Smagorinsky SGS model, was computed using a simplified version of the combined VEM + BEM method: a DES approach suitable for cases with prescribed separation lines. The realistic ground effects were also taken into account, with the GTS moving above the ground. A long time simulation was carried, and with long wake extend, to obtain mean velocity fields and mean streamlines. The obtained mean recirculation bubble appears shorter than in the case with unrealistic ground effects (where the GTS is fixed to the ground). The amplitude of the maximum velocity defect is roughly the same (about $1.2U_\infty$).

The wake dynamics were verified to be significantly different, depending on the ground conditions; this, in turn, could explain the differences in the size of the recirculation region and in the mean velocity profiles. The present results also call for conducting new experiments, with realistic ground effects (e.g., using a wind tunnel with moving belts) and with a larger window of measurements. They also suggest new RANS simulations.

The present simulation also illustrates the fact that LES requires long time simulations. This is especially true of flows without homogeneous directions, as in this case, the mean flow statistics can only be obtained by averaging over time. Here, the simulation was run up to $T = 165$: even then, the mean velocity fields are only “satisfactorily” converged (they are not fully converged). As to obtaining converged means of fluctuations (e.g., rms turbulence intensities), this would require even longer simulation times.



The present method, although simplified, already constitutes a useful engineering tool: one that can reproduce relevant unsteady vortex dynamics for flow past bluff bodies with prescribed separation regions. It should be further validated and enhanced (e.g. to obtain the unsteady forces). It is also expected that the more sophisticated relaxation scheme (based on the “projection algorithm” as used in Chapter 5) will constitute an improvement.

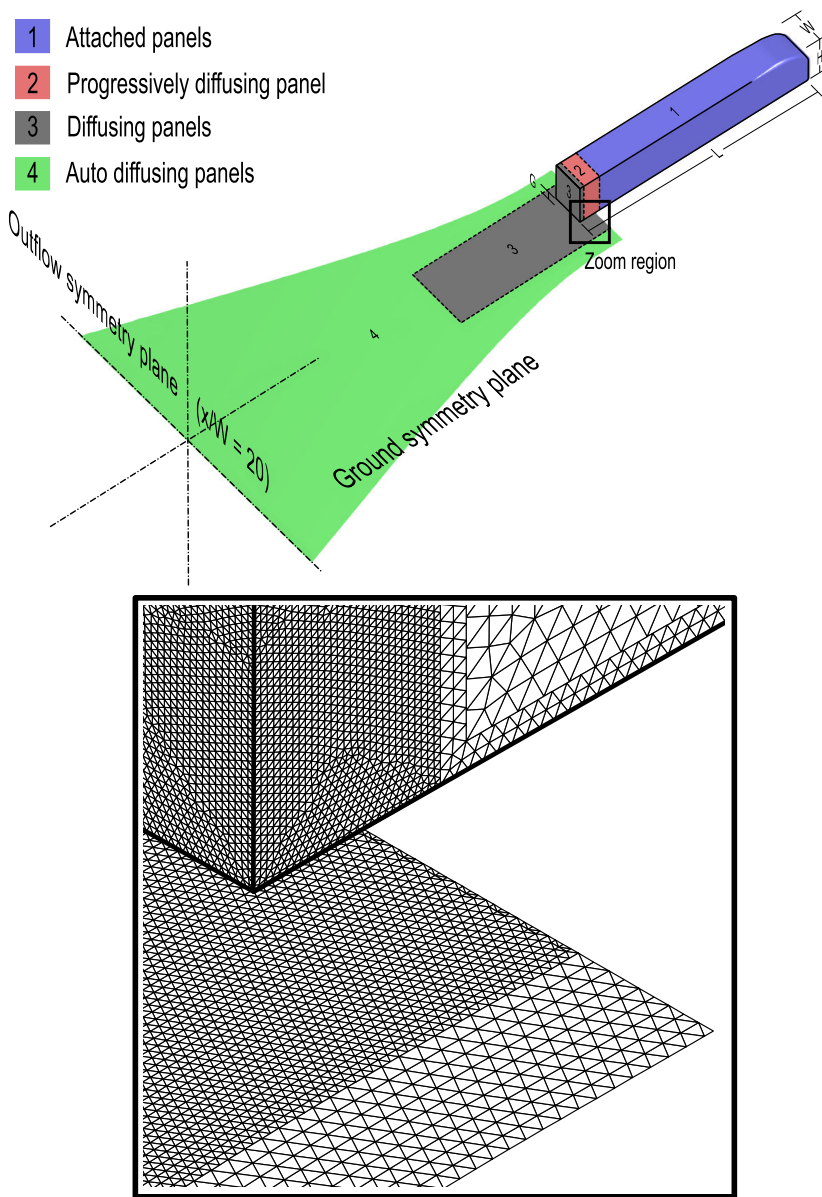
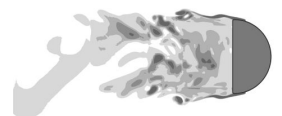


FIGURE 2.11 – Left: “Numerical setup” for the flow past the GTS. Dimensions: $L/W = 7.65$, $H/W = 1.375$, $G/W = 0.23$. The symmetry planes used to enforce the outflow and ground boundary conditions are shown. Right: Zoom on body surface and ground discretization (using vortex sheet panels).



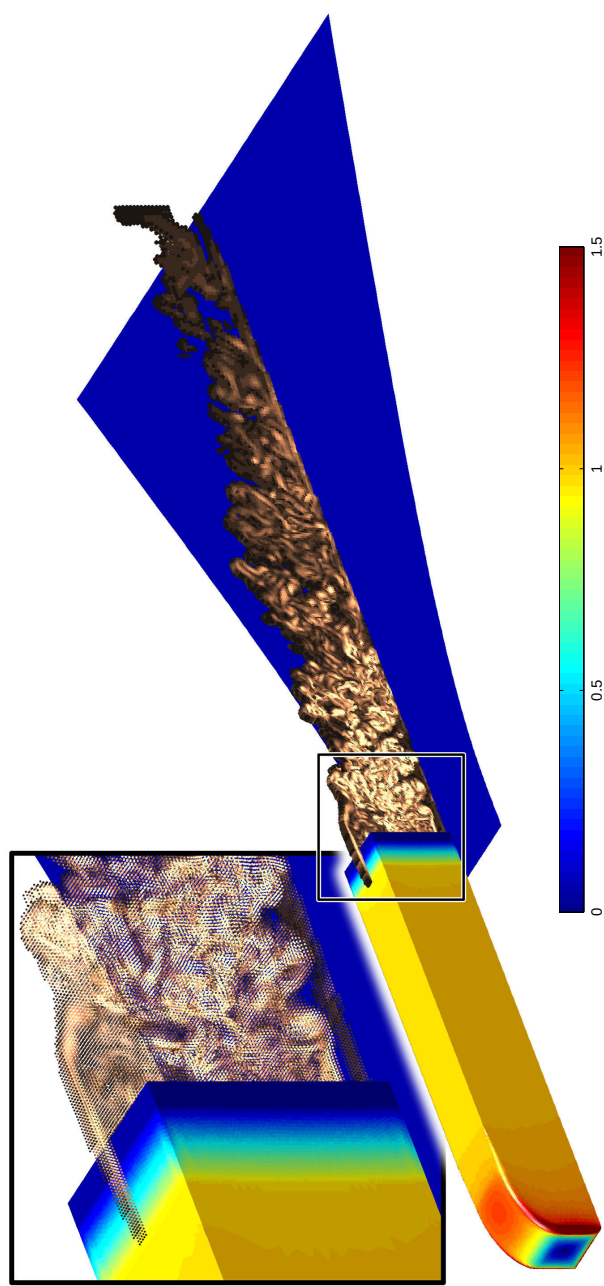


FIGURE 2.12 – Snapshot of the flow past the GTS. The vortex panels are colored according to the norm of their strength (which is equal to the local slip velocity above the panel). A “slice” cut of the particles through the wake is also shown; only particles with a strength above a threshold were kept (about 5% of the particles are shown). Particles are colored according to the norm of the vorticity.

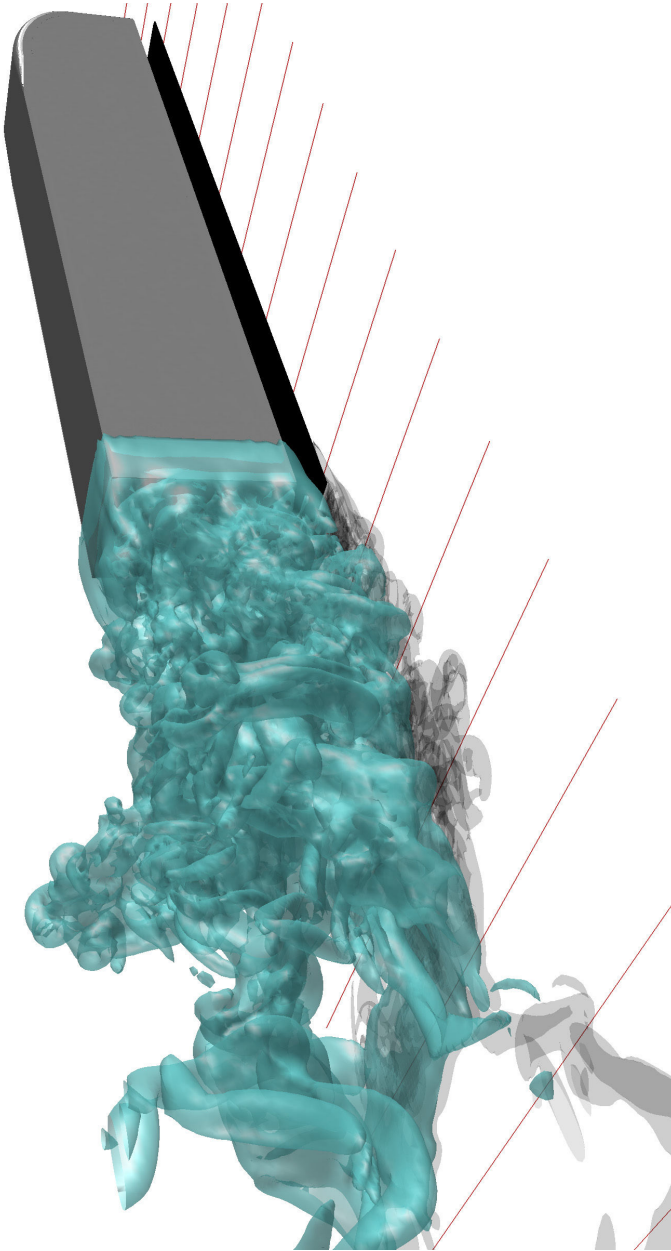
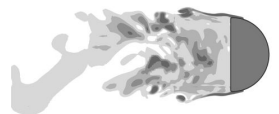


FIGURE 2.13 – Flow past the GTS with viscous ground effect in the wake. Isosurface of vorticity norm.



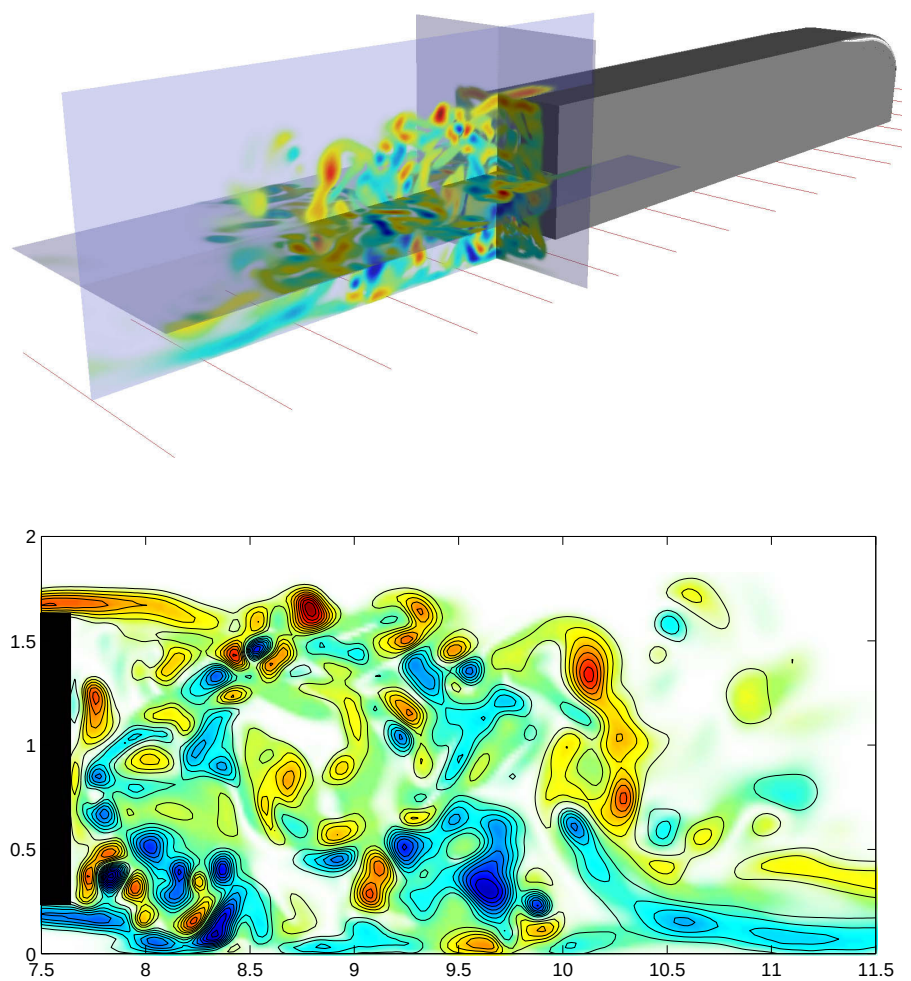


FIGURE 2.14 – Flow past the GTS with viscous ground effect in the wake. **Top:** The regions of non zero vorticity are shown in three slices behind the truck: each slice is colored by the y -component of vorticity. **Bottom:** Zoom on the vertical midplane slice behind the truck.

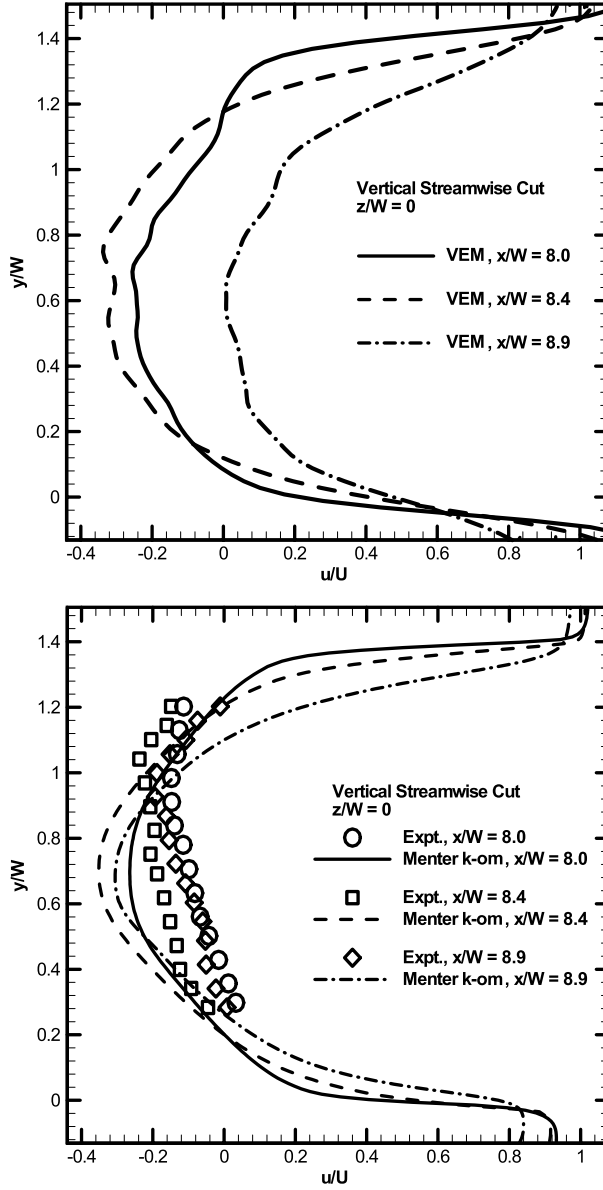
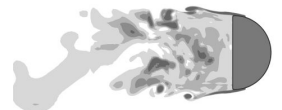


FIGURE 2.15 – Mean streamwise velocity profiles along three vertical lines located in the mid-plane, and at various positions behind the GTS ($x/W = 8.0, 8.4, 8.9$). **Top:** Our simulation with GTS moving above ground. **Bottom:** Experimental results (symbols) and RANS results by Menter (lines), both with GTS fixed to ground.



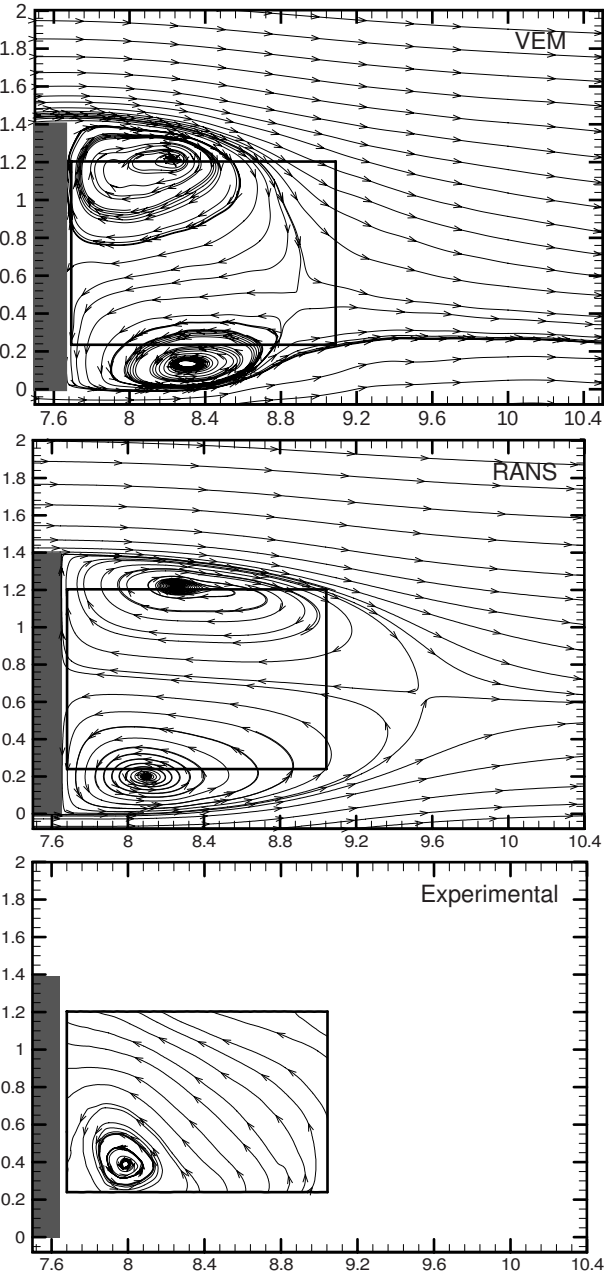


FIGURE 2.16 – Mean streamlines in the vertical midplane: our simulation with GTS moving above ground (**top**), RANS (**center**) and experiment (**bottom**) with GTS fixed to ground.

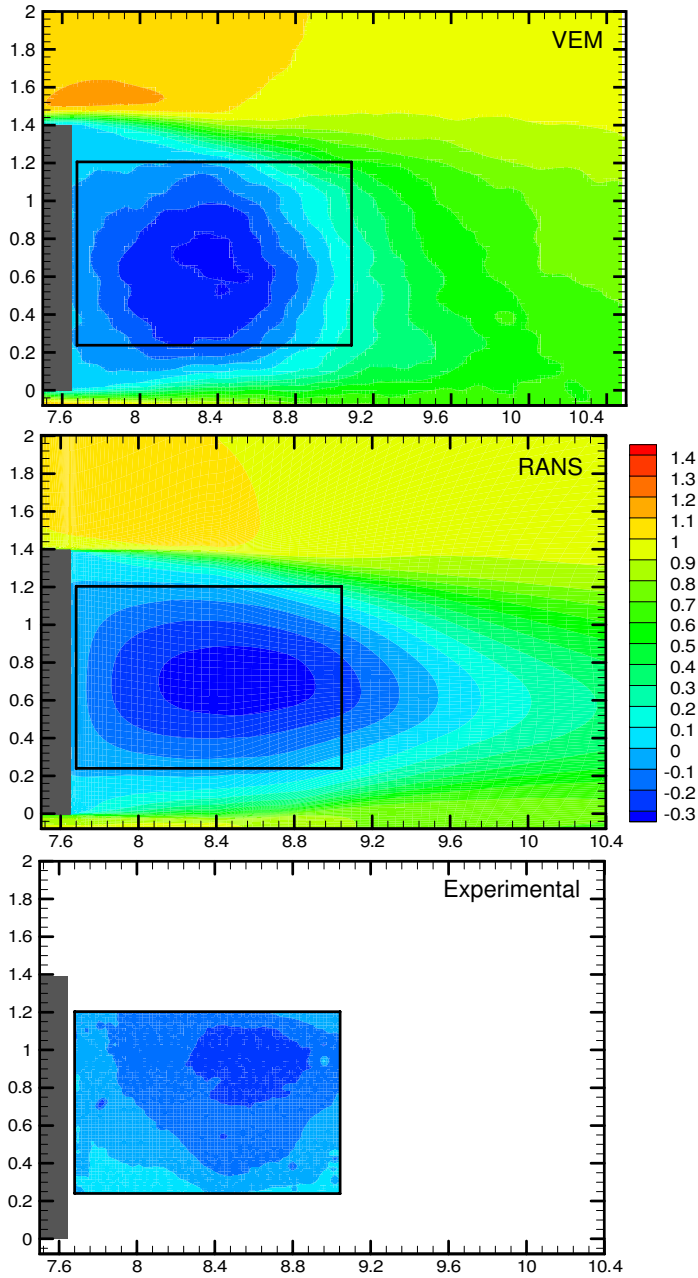
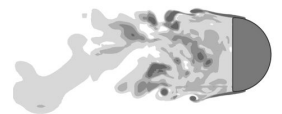


FIGURE 2.17 – Mean streamwise velocity component in the vertical midplane: our simulation with GTS moving above ground (**top**), RANS (**center**) and experiment (**bottom**) with GTS fixed to ground.



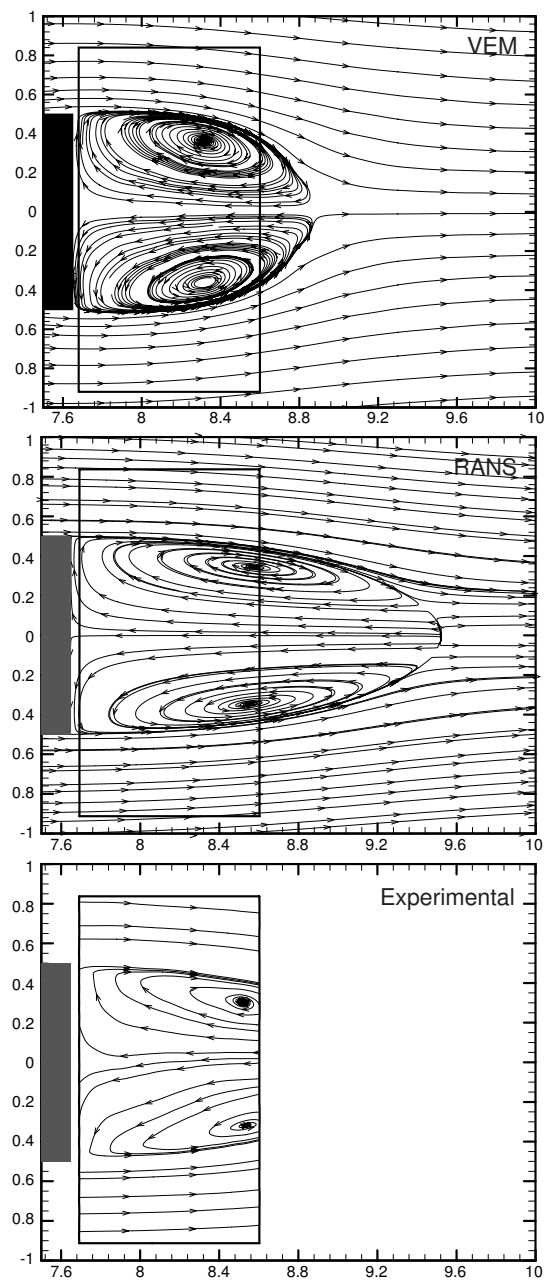


FIGURE 2.18 – Mean streamlines in a horizontal plane ($y = 0.7W$): our simulation with GTS moving above ground (**top**), RANS (**center**) and experiment (**bottom**) with GTS fixed to ground.

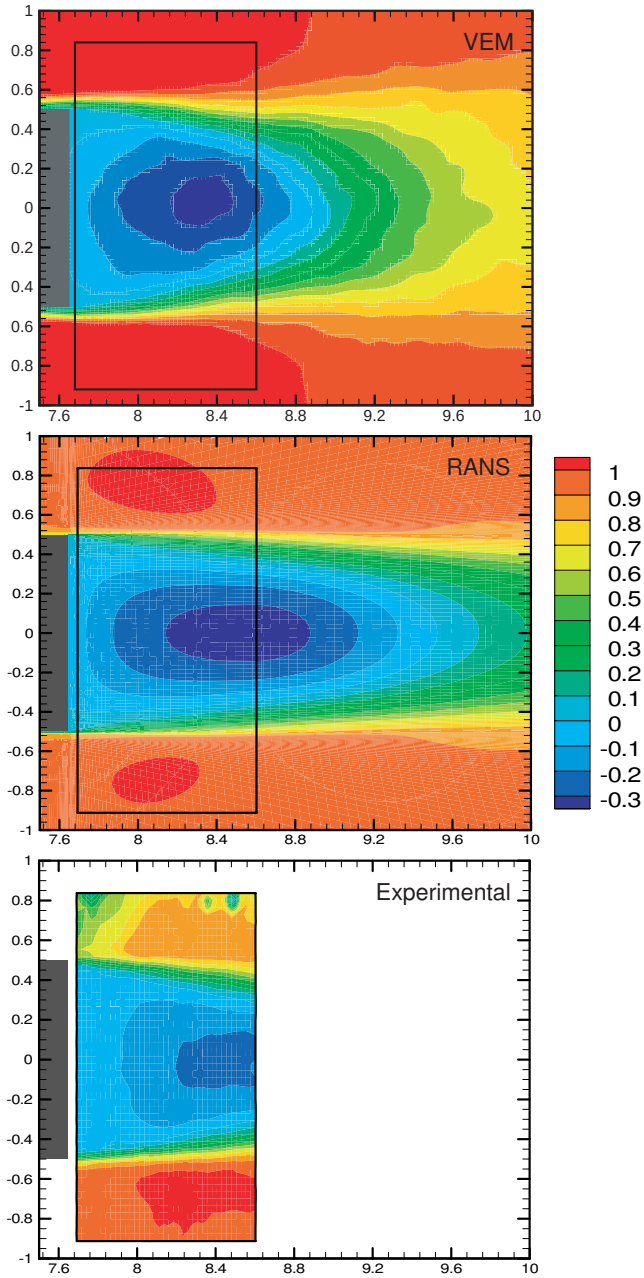
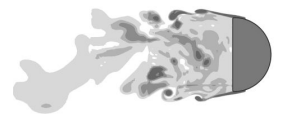


FIGURE 2.19 – Mean streamwise velocity component in a horizontal plane ($y = 0.7W$): our simulation with GTS moving above ground (**top**), RANS (**center**) and experiment (**bottom**) with GTS fixed to ground.



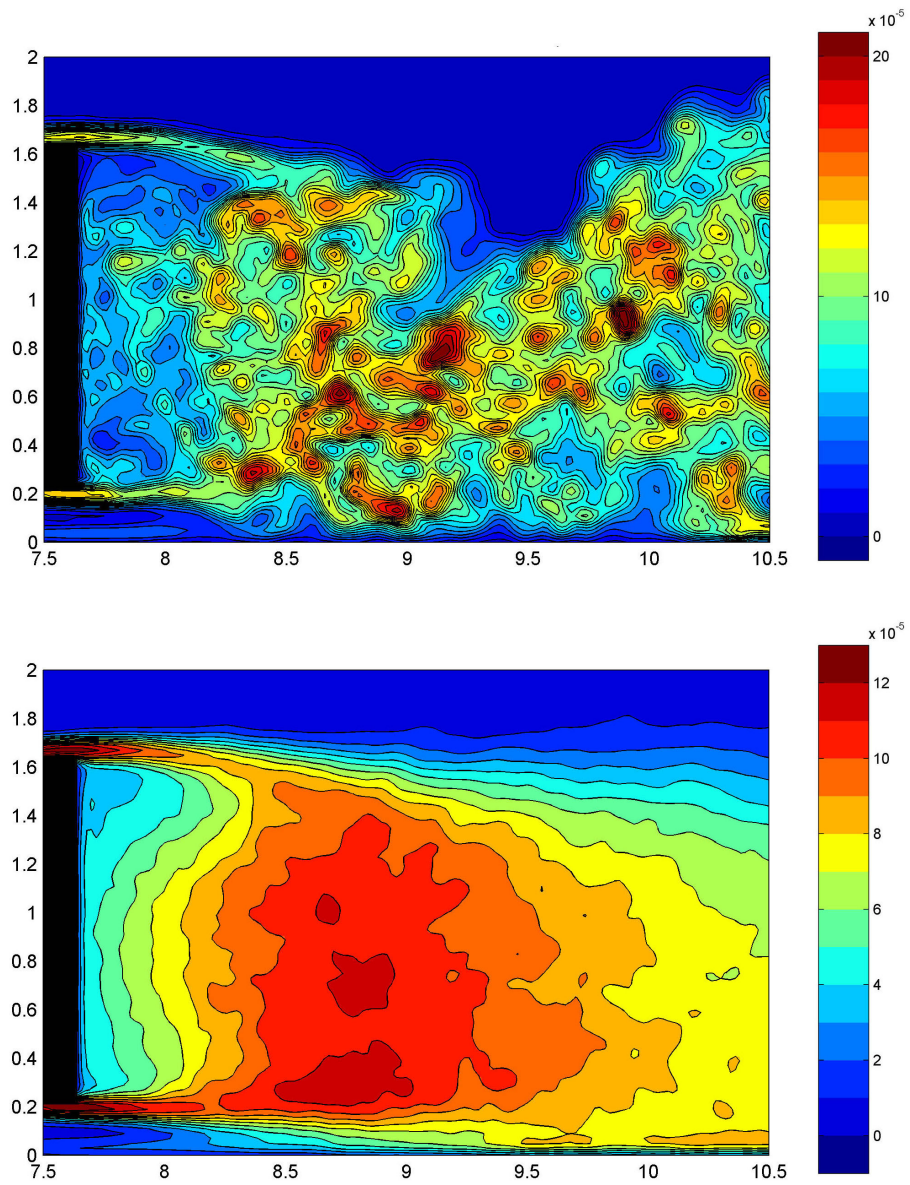


FIGURE 2.20 – Instantaneous (**top**) and mean (**bottom**) effective eddy viscosity in the vertical midplane for our simulation with GTS moving above ground.

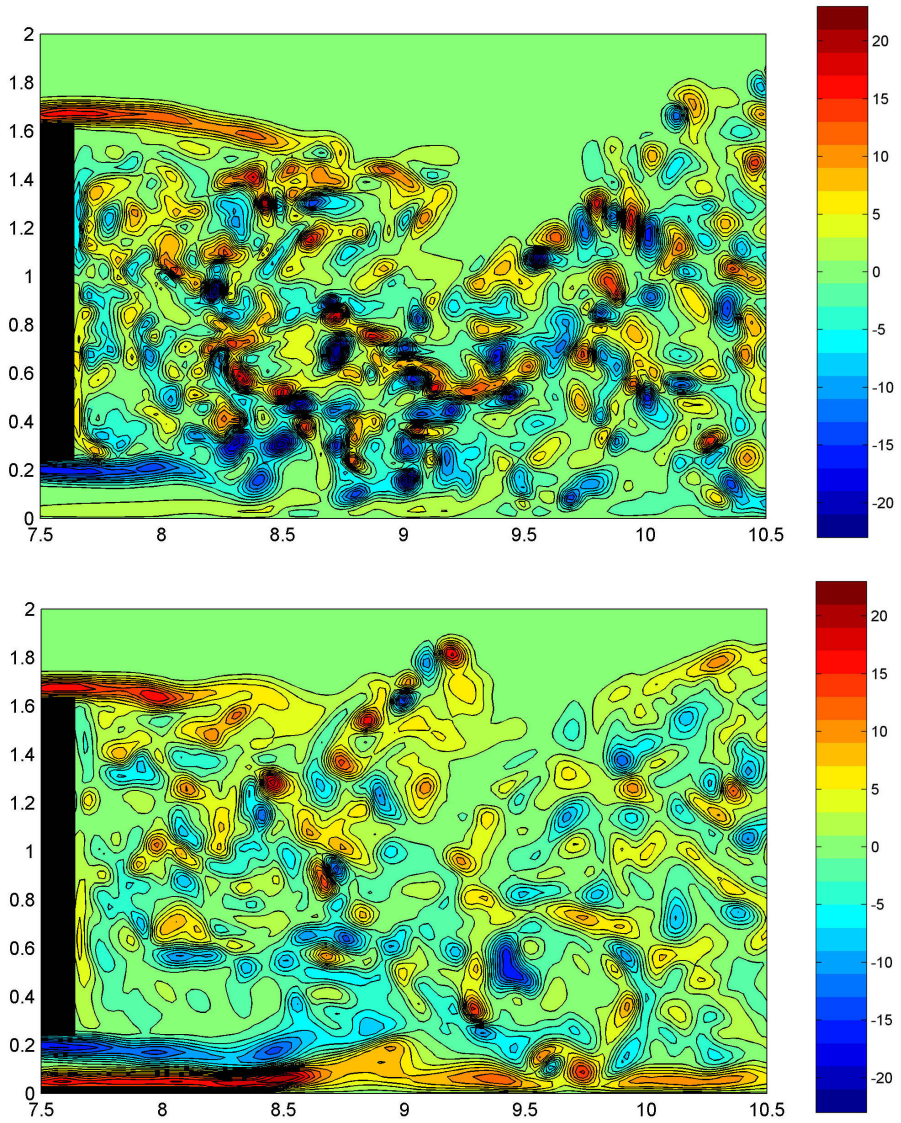


FIGURE 2.21 – Comparison of the flow past the GTS depending on ground condition: GTS moving above ground (**top**) versus vorticity field for GTS fixed to ground (**bottom**).



CHAPTER 3

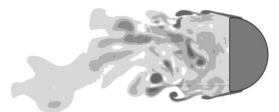
A hybrid Eulerian-Lagrangian vortex method

The content of this chapter was published as an article entitled “A new hybrid Eulerian-Lagrangian vortex method for flows with massive separation” [33]. Authors are G. Daeninck and G. Winckelmans. This article is essentially reproduced here as it was published with some additional developments and results.

We describe and illustrate a new hybrid Eulerian-Lagrangian approach for the incompressible Navier-Stokes equations. The method is based on a decomposition of the computational domain: the near-wall region is resolved using an Eulerian grid based solver, while the Lagrangian vortex element method is used for the wake region. The key feature of this new approach is the way the methods are coupled.

3.1 Motivation

The current work concerns with the development of tools for simulating 3-D unsteady flows with massive separation past bluff bodies. It has already been



shown that Lagrangian vortex element methods are well suited for this type of flows:

- Ploumhans *et al.* used a vortex and boundary element method for performing DNS of the 3-D flow past a sphere at $Re = 300, 500$ and $1,000$ [77, 32];
- Poncet *et al.* used a vortex in cell (VIC) method for simulating the 3-D flow past a circular cylinder [23, 81];
- Most recently, we used a vortex and boundary element method for modeling the flow past a simplified truck geometry at very high “Reynolds number” (in this approach boundary layers were modeled using infinitely thin vortex panels) [30].

Vortex element methods are based on the vorticity formulation of the Navier-Stokes equations and on the fact that, for incompressible flows, it is sufficient to follow the evolution of the vorticity field (the velocity can be recovered from the vorticity). For external flows, the vorticity is only present in the boundary layers and in the wake. As vortex methods only require particles to carry the vorticity, they require much less computational elements than grid-based methods. The Lagrangian treatment also eliminates the stability constraint on the treatment of the convection term (but not the accuracy constraint: $\omega \Delta t$ must remain $\mathcal{O}(1)$).

However, the VEM suffers from two drawbacks in the boundary layer regions, which prevent it from working efficiently in these regions when considering high Reynolds number flows:

- The VEM uses isotropic computational elements whereas, in the boundary layer regions, the strong gradients in the direction normal to the wall would allow to use highly anisotropic elements for computational efficiency.
- Although the VEM performs particularly well for flows dominated by convection (due to the treatment of the convective term), it is less suited to flexible and accurate treatment of the no-slip boundary condition.

These drawbacks currently limit the applicability of the VEM to relatively low Reynolds number flows or flows where an excessive degree of modeling must be introduced in the boundary layer regions. Approaches along the lines of Detached-Eddy Simulation (DES), combining RANS methods (boundary layer

regions) and LES methods (separated regions), are required in order to perform simulations at much higher Reynolds numbers.

This leads naturally to the observation that Eulerian grid-based methods should be used to resolve the near-wall regions, where viscous effects are important, while using the Lagrangian VEM for the convection-dominated part of the flow. The strengths and the weaknesses of both methods are indeed complementary in such a hybrid approach.

3.2 Global approach

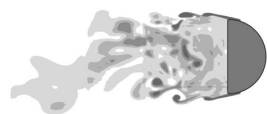
The idea of coupling grid-based methods and particle methods is not new. They are based on a decomposition of the computational domain into several subdomains where different solvers are used. The key feature of these methods is the coupling between the computational subdomains:

- to ensure consistant boundary conditions on each subdomain (Eulerian and Lagrangian);
- to perform accurate transfer of information between these subdomains.

Particle-grid domain decomposition methods have been formalized by Cottet *et al.* [18, 19, 75]. Their approach is based on the classical framework of domain decomposition techniques for partially overlapping domains (Fig. 3.1). The coupling between the subdomains relies on the Schwarz alternating method: the boundary conditions on each subdomain are determined iteratively. Cottet *et al.* were able to obtain good results (in 2-D) using this approach for several particle-grid decomposition configurations. They also showed the possibility of using the velocity-pressure formulation in the Eulerian domain (instead of the vorticity-velocity formulation as in the Lagrangian domain).

We here propose a new approach where the iterative Schwarz method for obtaining consistant boundary conditions is no longer needed. Our method arises from the following statements:

- Use the Lagrangian vortex method to solve the entire flow, although under-resolved in the near-wall region (do not waste computational elements in boundary layers where the Lagrangian vortex method is less efficient).
- Use a grid-based solver in the near wall region for capturing the detailed wall features with a much higher computational efficiency.



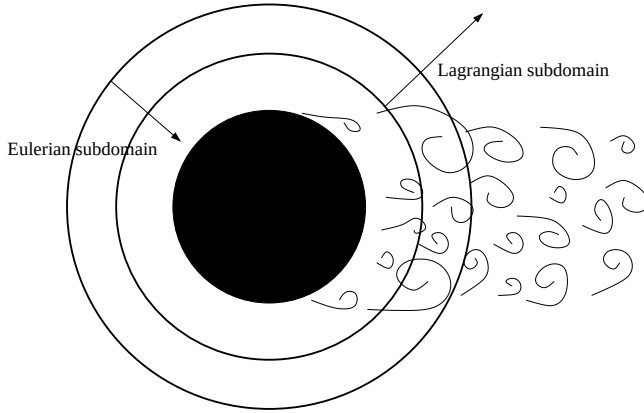


FIGURE 3.1 – Example of domain decomposition with partially overlapping domains as used by Cottet *et. al.*

- Correct the Lagrangian method in the near wall region using the Eulerian information.
- Use the Lagrangian domain to obtain the boundary condition on the Eulerian domain (the boundary condition can be obtained directly: no need for an iterative method).

In the end, our approach amounts to using a Lagrangian vortex element method which resolves the boundary layer regions by “calling” an Eulerian grid based solver. A typical domain configuration for our approach is shown in Figure 3.2.

For 2-D computations, one could argue that the computational gain provided by our method is only marginal compared to the approach used by Cottet *et al.*; however, in 3-D, we expect the gain to become significant and even critical as the Schwarz algorithm would then be very expensive. Furthermore, as we will see in Section 3.3, our approach allows to use different time steps in the Eulerian and Lagrangian subdomains (in a very simple way).

3.3 The method

As mentioned in Section 3.1, the aim of our work is the development of 3-D methods for flows with massive separation; however, for obvious reasons, we first developed and tested our new hybrid approach in 2-D, but keeping in mind

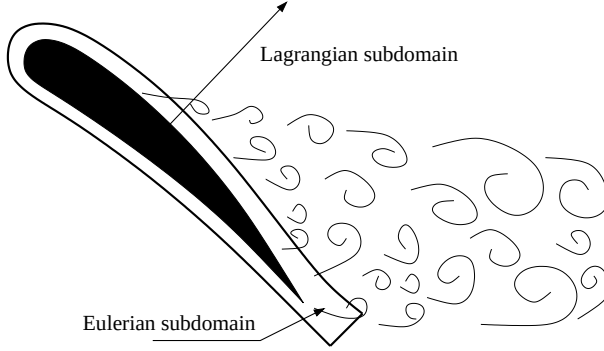


FIGURE 3.2 – Example of Eulerian and Lagrangian subdomains as used in our approach. The Lagrangian subdomain covers the whole computational domain: it captures the wake dynamics and uses the Eulerian subdomain as a near-wall region solver.

that the method should be easily extendable to the 3-D case. We first present the approach in the vorticity-velocity formulation for the Eulerian domain. However, we think that, in 3-D, the velocity-pressure formulation should be used. The extension of our method to the velocity-pressure pressure in the Eulerian subdomain is briefly presented in Section 3.3.4.

We now describe the numerical methods used in the Lagrangian and Eulerian subdomains and how the coupling is performed.

3.3.1 Lagrangian subdomain

The Lagrangian subdomain uses a modified version of the “classical” vortex element and boundary element method for simulating viscous wall-bounded flows.

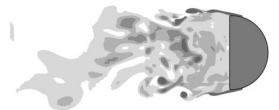
Let us first recall the vorticity-velocity formulation for the incompressible 2-D Navier-Stokes equations:

$$\frac{D\omega}{Dt} = \frac{\partial\omega}{\partial t} + \mathbf{u} \cdot \nabla\omega = \nu \nabla^2\omega, \quad (3.1)$$

$$\mathbf{u} = \nabla \times (\psi \mathbf{e}_z) + \mathbf{U}_\infty \quad (3.2)$$

$$\nabla^2\psi = -\omega \quad (3.3)$$

where $\mathbf{u}(\mathbf{x}, t)$ is the velocity field, $\psi = \psi \mathbf{e}_z$ is the associated stream function,



ν is the kinematic viscosity, and $\boldsymbol{\omega} = \nabla \times \mathbf{u} = \omega \mathbf{e}_z$ is the vorticity. Thus, in a Lagrangian framework, the evolution of the vorticity field is governed by viscous diffusion only. This system has to be supplemented with boundary conditions to enforce the no-slip boundary condition at the wall.

Eq. (3.1) is solved by discretizing the vorticity field using a set of N Lagrangian vortex particles:

$$\boldsymbol{\omega}(\mathbf{x}, t) = \sum_{i=1}^N \frac{1}{(2\pi\sigma_i)^2} \exp\left(-\frac{1}{2} \frac{|\mathbf{x} - \mathbf{x}_i|^2}{\sigma_i^2}\right) \boldsymbol{\alpha}_i, \quad (3.4)$$

where $\mathbf{x}_i$ is the position of particle i , and $\alpha_i = \int \omega d\mathbf{x} = \omega_i h_i^2$, is the particle strength. A value of $\sigma/h = 1$ is here used for good reconstruction/overlapping.

One time step of the method can be summarized as follows:

1. Compute the vortex sheet, γ , so as to cancel the slip velocity at the solid boundary. The sheet is discretized using M vortex panels, whose strength is found as the solution of a boundary integral equation, the discretization of which results in a system of M linear equations for the M vortex panels.
2. Compute the velocity field, $\mathbf{u}$, from ω (the vorticity carried by the vortex particles) and γ (the vorticity carried by the vortex panels), using a fast, $\mathcal{O}(N \log N)$, Green's functions approach (fast multipole method).
3. Update the particles' position and strength, for the time step, Δt ,

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{u}|_i \quad (3.5)$$

$$\frac{d\boldsymbol{\alpha}_i}{dt} = \text{PSE}|_i \quad (3.6)$$

where the Particle Strength Exchange (PSE) scheme is used to handle the viscous diffusion (this involves exchange of particle strengths, using only the near neighbor particles).

4. Redistribute the particles, when necessary, using a high order scheme: the old set of particles is replaced by a new one in which the particles are on a regular lattice.

The only difference with the “classical” vortex method as [77], is the way boundary conditions are handled. In the “classical” approach, the vortex sheet is computed at the end of a time step and is then diffused on the neighbor

particles in order to resolve the vorticity flux at the wall. Here, as the Lagrangian method is under-resolved at the wall, the vortex sheet is not diffused: it remains captured by the vortex panels. Then when particles are convected, the panel vorticity must also be taken into account.

Of course at this point, a question arises: how does vorticity get into the flow if there is no vorticity flux at the wall? Simply, through the near-wall region solver: before each time step the strength of the particles located in the near-wall region are interpolated from the Eulerian subdomain which resolves this region well.

The Lagrangian subdomain uses a second-order Runge-Kutta time stepping for the particle convection; a simple Euler explicit integration scheme is used for the diffusion. Only the diffusion part of the equation is subjected to a stability constraint.

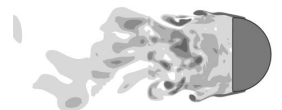
3.3.2 Eulerian subdomain

As we already mentioned, the Eulerian subdomain also uses the vorticity-velocity formulation. We thus solve the same set of equations as in the Lagrangian domain, but in an Eulerian framework: a finite difference approach in curvilinear coordinates is used, with second-order centered schemes (when necessary second-order decentered schemes are used).

At each time step (or substep in a Runge-Kutta scheme), the streamfunction is computed by solving Eq. (3.3) with the appropriate boundary conditions. At the wall, the no-trough flow condition is enforced through a homogeneous Dirichlet boundary condition. On the external boundary, the boundary condition is retrieved from the Lagrangian domain: we chose to enforce a Neumann condition (i.e. the tangential velocity) on this boundary, as it provides a stronger coupling between the methods.

Once the streamfunction is known, we can evaluate the velocities through Eq. (3.2), with, at the wall, the no-slip boundary condition.

Concerning the vorticity boundary condition at the wall, we used the following second-order scheme: after the velocities are computed (notice that this did not require the knowledge of the vorticity at the boundary points at any time), second-order one-sided finite differences are used to compute the vorticity at the boundary. On the external boundary, the vorticity is retrieved from



the Lagrangian domain. The vorticity on the boundaries is used as a Dirichlet boundary condition in the vorticity evolution equation.

We can now update the vorticity at all interior grid points using Eq. (3.1). The vorticity evolution equation is integrated in time using a fourth-order Runge-Kutta scheme. This is necessary for the centered scheme to be stable under a CFL condition.

From a computational point of view, most of the CPU time of this algorithm is obviously devoted to the resolution of the Poisson equation. We implemented an efficient Poisson solver in orthogonal curvilinear coordinates based on a public domain multigrid PDE solver (Mudpack, [2]). As we will see in Section 3.4.1, the global efficiency of the Eulerian domain solver is really the key which leads to the success of our approach. Furthermore, the general curvilinear coordinates approach allows us to refine the finite-difference grid where computational elements are most needed (e.g. in the boundary layers and the separation region).

3.3.3 Coupling

We here describe how both solvers are coupled in one global time stepping algorithm.

First, it is important to understand that, even if the Lagrangian domain is under-resolved in the boundary layer regions, it is able to provide the correct boundary condition on the Eulerian external boundary. This is because the vortex sheet enforcing the no-through flow at the wall is computed at each time step (or substep in the case of a Runge-Kutta time stepping scheme). However, it is clear that the dynamics in the near wall region cannot be captured, and therefore, the particle strengths are adjusted there using the Eulerian vorticity field.

The second point that needs to be stressed is that, at high Reynolds numbers, the stability constraint on the time step in the Eulerian domain is drastically more severe than that in the Lagrangian domain. Because one Lagrangian time step is much more expensive than one Eulerian time step (when the wake is fully developed), using the same time discretization in both domains really amounts to a waste of computational resources.

Let Δt_L and $\Delta t_E = \Delta t_L/k$ be respectively the time step in the Lagrangian and the Eulerian domain. The fields of both domains are known at the common

time t_n . Then, in order to advance both fields from time t_n to $t_n + \Delta t_L$, we proceed as follows:

1. Adjust the Lagrangian field in the near wall region from the Eulerian field (both fields are at time t_n): particle strengths are interpolated from the Eulerian grid.
2. Advance the Lagrangian field from time t_n to $t_n + \Delta t_L$ using a second-order Runge-Kutta algorithm as explained in Section 3.3.1 (the Lagrangian field is not corrected over this time step).
3. The Lagrangian field at time t_n is used to compute the Eulerian boundary conditions on ψ and ω at time t_n ; the Lagrangian field at time $t_n + \Delta t_L$ (which has not been corrected yet in the near-wall region) can be used to provide a very good prediction of the boundary condition on the Eulerian domain at time $t_n + \Delta t_L$.

At this point we have the boundary conditions for ψ and ω on the Eulerian domain at times t_n and $t_n + \Delta t_L$.

4. Advance the Eulerian field from time t_n to $t_n + \Delta t_L$ using k fourth-order Runge-Kutta time steps as explained in Section 3.3.2. The boundary conditions at each substep of the Runge-Kutta time steps are obtained by linear interpolation from the boundary at times t_n and $t_n + \Delta t_L$.

Now we have both fields at the new time step $t_{n+1} = t_n + \Delta t_L$; we can start over again.

We should mention that, in order to enhance the coupling between both methods in the most external region of the Eulerian subdomain, the grid vorticity in this region is interpolated from the Lagrangian field (this is done during substep 1, see Figure 3.3). We observed that this provides better results, as it allows a better vorticity transfer from the Lagrangian subdomain back to the Eulerian subdomain.

3.3.4 Extension to the velocity-pressure formulation

The extension to the case where the velocity-pressure formulation is used in the Eulerian subdomain is quite straightforward. The equations to be solved



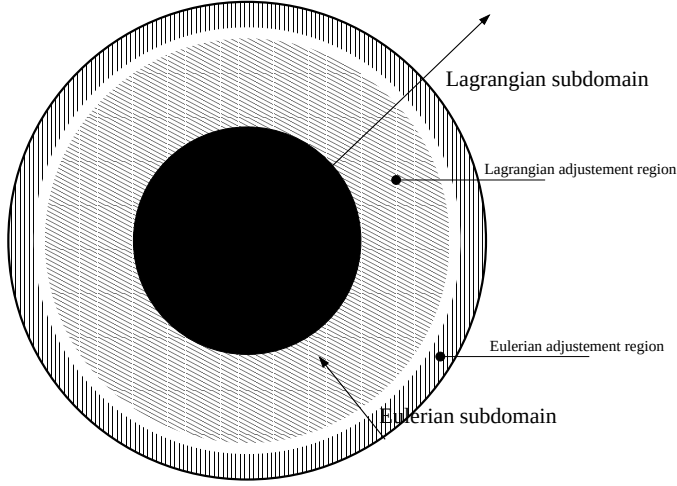


FIGURE 3.3 – Example of domain decomposition and interpolation regions.

in the Eulerian domain are:

$$\nabla \cdot \mathbf{u} = 0 \quad (3.7)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla P + \nu \nabla^2 \mathbf{u}, \quad (3.8)$$

where, $P = p/\rho$. In order to solve these equations, we have chosen to use the fractional step algorithm [12, 52]. This method splits the numerical operators and achieves the pressure-velocity coupling through solving a Poisson equation for pressure. The spatial discretization is here again achieved using second order central finite difference in curvilinear orthogonal coordinates.

For simplicity, we describe the algorithm using an explicit Euler temporal discretization (while in fact, in the Eulerian domain, all terms are treated using a fourth order Runge-Kutta scheme, as for the vorticity-velocity formulation). In this case, the intermediate velocity, $\mathbf{u}^*$, is obtained as follows:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -(\mathbf{u}^n \cdot \nabla) \mathbf{u}^n + \nu \nabla^2 \mathbf{u}^n \quad (3.9)$$

The intermediate velocity, $\mathbf{u}^*$ is then corrected to enforce mass conservation:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla P^{n+1} \quad (3.10)$$

where P^{n+1} is obtained by applying the divergence free constraint on $\mathbf{u}^{n+1}$,

which results in the pressure Poisson equation:

$$\nabla^2 P^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^* \quad (3.11)$$

$$\frac{\partial P^{n+1}}{\partial n} = \left(-\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} - (\mathbf{u}^n \cdot \nabla) \mathbf{u}^n + \nu \nabla^2 \mathbf{u}^n \right) \cdot \mathbf{n} \quad \text{on } \partial\Omega_E \quad (3.12)$$

The Neumann boundary condition is obtained by projecting Eq. (3.10) on the boundary normal and by replacing $\mathbf{u}^*$ by its expression from (3.9). This system thus requires that the velocity at time $n + 1$ is known on the boundary of the Eulerian domain: it can here be obtained directly from the Lagrangian solver. This approach is similar to the one developed by Ould-Salihi *et al.* [75], the difference being that we here do not need the iterative Schwartz algorithm for obtaining $\mathbf{u}^{n+1}$.

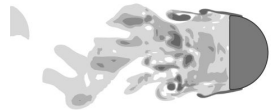
The global coupling algorithm is slightly modified (modifications shown in bold). The fields of both domains are known at the common time t_n . Then, in order to advance both fields from time t_n to $t_n + \Delta t_L$, we proceed as follows:

1. Adjust the Lagrangian field in the near wall region from the Eulerian field: **the vorticity field is obtained in the Eulerian grid by taking the curl of the velocity field, $\mathbf{u}^n$** . The particle strengths are interpolated from the Eulerian grid.
2. Advance the Lagrangian field from time t_n to $t_n + \Delta t_L$ using a second-order Runge-Kutta algorithm as explained in Section 3.3.1 (the Lagrangian field is not corrected over this time step).
3. The Lagrangian field at time t_n is used to **compute the Eulerian boundary conditions on $\mathbf{u}$** at time t_n and $t_n + \Delta t_L$.
4. Advance the Eulerian field from time t_n to $t_n + \Delta t_L$ **using the velocity-pressure fractional step algorithm** as explained above (using k Eulerian steps).

3.4 Results

3.4.1 Flow past a cylinder at $Re = 3,000$

We here present an application of the hybrid Eulerian-Lagrangian method (using the vorticity-velocity formulation in both domains) to the case of the flow past a cylinder at $Re = U_\infty D / \nu = 3,000$.



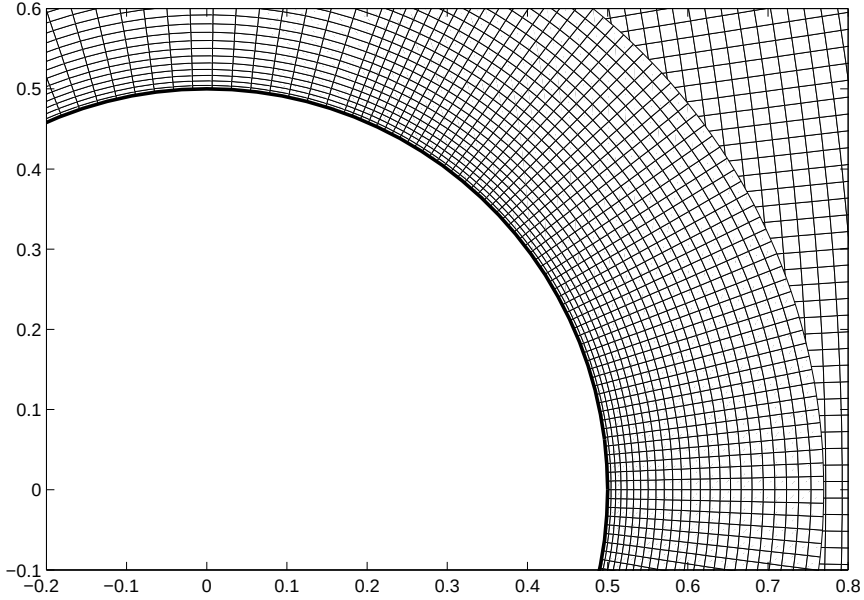


FIGURE 3.4 – Flow past a cylinder at $Re = 3,000$: zoom on the finite-difference grid and redistribution lattice in the vicinity of the body. The resolution shown is twice coarser in both directions compared to that used in the computation.

Figure 3.4 shows the finite-difference grid and the redistribution lattice for the Lagrangian subdomain. The curvilinear coordinate system was chosen such that the boundary layers regions are very well refined as well as the separation region behind the cylinder. Notice also that, in this region, the cells become isotropic as one moves away from the body. The Lagrangian redistribution lattice is chosen such that its resolution in the transition region is similar to the finite-difference grid. In the boundary layers, the finite-difference grid size in the normal direction is $h_n/D \sim 2 \times 10^{-3}$, while the particle size in this region is $h_{\text{part}}/D \sim 8 \times 10^{-3}$. The size of the finite-difference grid is 49×385 . The dimensionless time is defined as $T = U_\infty t/D$. The time step for the Lagrangian domain was $\Delta T_L = 5 \times 10^{-3}$, while the Eulerian domain used $\Delta T_E = 5 \times 10^{-4}$ ($k = 10$). The simulation was carried out up to $T = 50$. The number of particles in the Lagrangian domain goes from ~ 5000 at $T = 0$ up to ~ 85000 at the end of the simulation.

The computational times are provided in Table 3.1. As expected, the com-

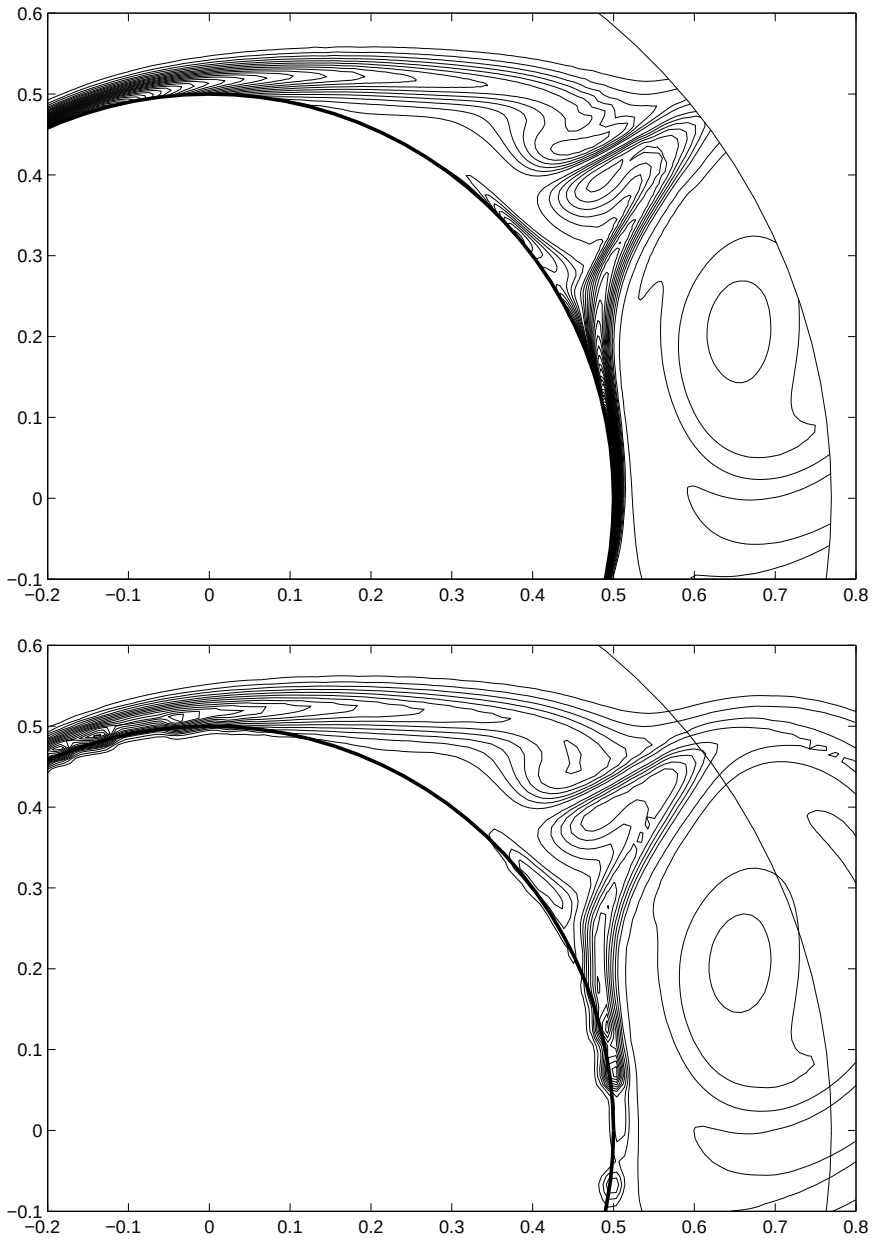


FIGURE 3.5 – Flow past a cylinder at $Re = 3,000$: isocontours of vorticity at $T = 34$. Levels are by steps of 4. Top: finite-difference subdomain. Bottom: Lagrangian domain.

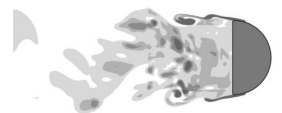


TABLE 3.1 – CPU times for Lagrangian and Eulerian solvers to perform one “global” time step (1 RK2 step in the Lagrangian solver, $k = 10$ RK4 steps for the Eulerian solver).

T	0	25	50
N particles	4188	54552	86857
Lagrangian solver	0.96 s	10.3 s	18.1 s
N grid points	18865	18865	18865
Eulerian solver	1.54 s	1.61 s	1.65 s

The computation ran on a linux PC with an Intel P4 processor at 2.4GHz

putational time for the Eulerian solver remains constant while the computational time for the Lagrangian solver increases with the number of particles. One should also notice that, if the same time step ($\Delta T_E = 5 \times 10^{-4}$) had been used in both domains, the computational time would have been multiplied by a factor $k = 10$ (since, once the wake is developed, the CPU time for the Eulerian solver becomes small compared to the CPU time for the Lagrangian solver). Furthermore, a purely Lagrangian computation with a resolution in the boundary layers equivalent to the current Eulerian grid would have needed 16 times more particles (and thus, at least 16 times more CPU time)!

In Figure 3.5, the vorticity field in the Eulerian domain is shown. One can see that the sharp vorticity gradients are very well captured by the finite difference grid, while the Lagrangian domain is clearly under resolved in the boundary layer regions. In the transition region, both subdomains are well resolved and, as one can see, both fields match almost perfectly there (see also Fig. 3.6).

3.4.2 Flow past a stalled airfoil at $Re = 5,000$

We here briefly present results obtained (using the vorticity-velocity formulation in both domains) for the flow past an airfoil at large angle of attack ($\alpha = 20^\circ$) and at $Re = U_\infty c/\nu = 5,000$, where c is the airfoil chord. Simulating this case in 2-D is somewhat artificial since, at this Reynolds number, the flow would rapidly become three-dimensional in the separated region and

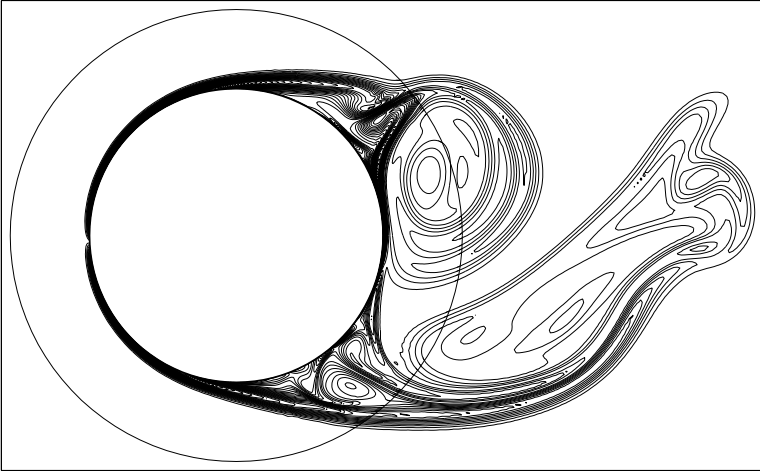
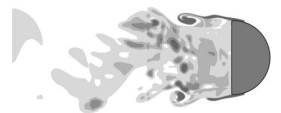


FIGURE 3.6 – Zoom on the near field for the flow past a cylinder at $Re = 3,000$: isocontours of vorticity at $T = 34$. Levels are by steps of 2. The vortex wake supported by the Lagrangian domain extends up to ~ 34 diameters.

in the wake. However, it suffices as proof of concept. As we will see, the 2-D simulation already produces interesting and complex dynamics due to massive separation along the leading and trailing edges.

The airfoil itself is a “regularized” Joukowski airfoil, in order to obtain a regular trailing edge (i.e., not infinitely thin). The Joukowski transformation is also used to obtain the Eulerian grid (Fig. 3.7). The Joukowski transformation being conformal, it allows to easily obtain an orthogonal grid, which is necessary for the present finite difference Eulerian solver. This grid is however not optimal in terms of efficiency: there is a very large ratio between the cell size at the trailing edge and the cell size along the intrados/extrados. Obviously, a *C*-type mapping would have been better suited for this type of configuration.

The grid and the computational domains are shown in Fig. 3.7. Figure 3.8 shows the Eulerian and Lagrangian domains along with the reconstructed vorticity fields from each domain. The figure also illustrates how well the hybrid method performs: the solution is very well-captured everywhere, and also in the Eulerian-Lagrangian transfer region. The zoom on the leading edge region shows that the Eulerian domain resolves the separating boundary layer very well. On the other hand, the Lagrangian domain is not sufficiently resolved



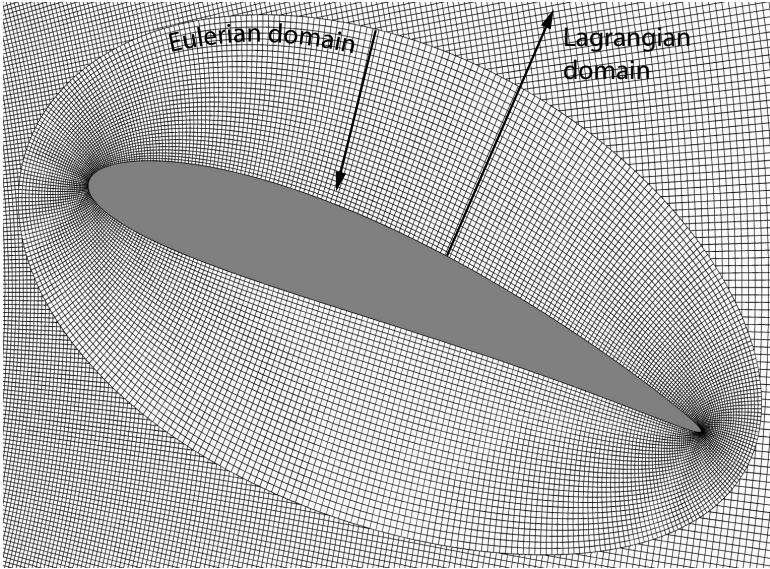


FIGURE 3.7 – Flow past a stalled airfoil at $Re = 5,000$: zoom on the finite-difference grid and redistribution lattice in the vicinity of the body. The resolution shown is twice coarser in both directions compared to that used in the computation.

there to accurately capture the flow dynamics on its own.

Figure 3.9 shows the time evolution of the wake generated by the impulsively started airfoil. Initially, a startup vortex is produced. At $T = tU_\infty/c = 1.0$, the boundary layer on the extrados is still attached but it rapidly separates. It is interesting to observe that the separation occurs almost immediately after the leading edge. This is due to the large angle of attack, and the fact that the upstream boundary layer is laminar.

The subsequent unsteady dynamics are relatively complex. Large negative vortices are formed through the rollup of the separated shear layer at the leading edge. Every now and then, these large structures are ejected into the wake. When a negative vortex is convected downstream of the trailing edge, an intense positive shear layer separates from the trailing edge (this can be seen at $T = 5$ and $T = 8$). This shear layer then rolls up and forms a large positive vortex which is then ejected into the flow. The process can then start over again. When the positive shear layer rolls up, Kelvin-Helmholtz instabilities appear. This can be seen from $T = 7.5$ to 8.5 . These dynamics are similar

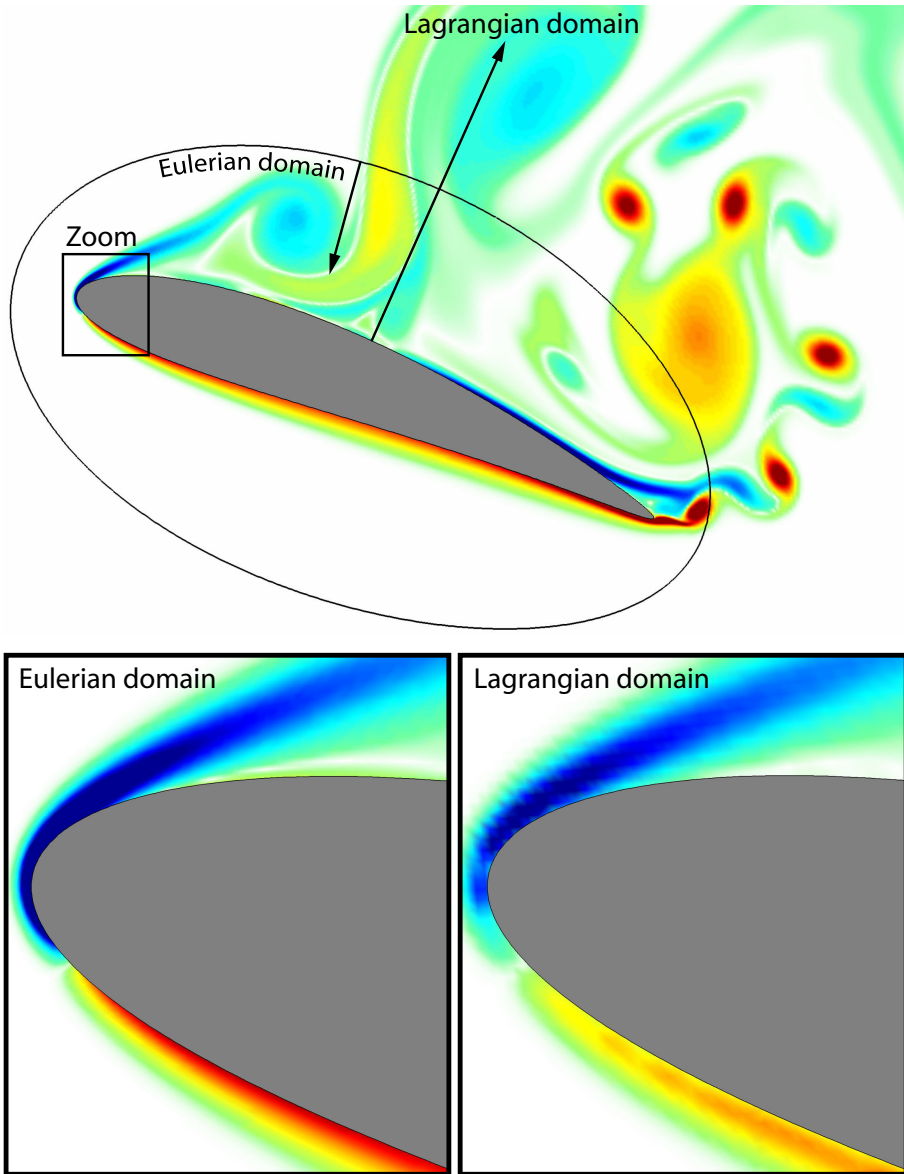
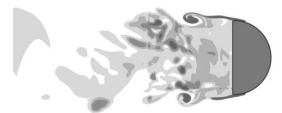


FIGURE 3.8 – Flow past a stalled airfoil at $Re = 5,000$. Top: Computational domains with reconstructed vorticity fields from both domains. Bottom: Zoom on the leading edge region in each domain.



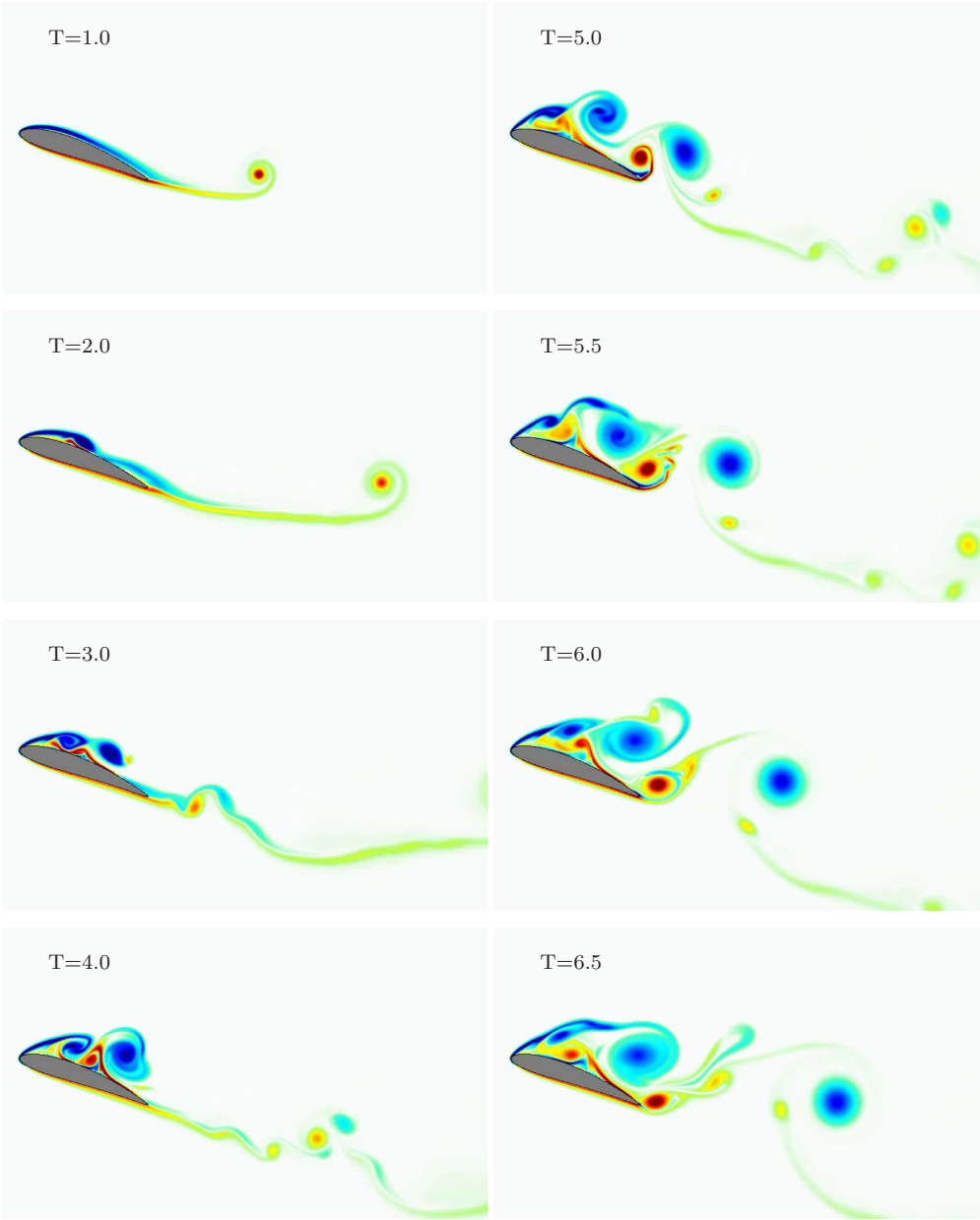


FIGURE 3.9 – Flow past a stalled airfoil at $Re = 5,000$. Time evolution of the vorticity field.

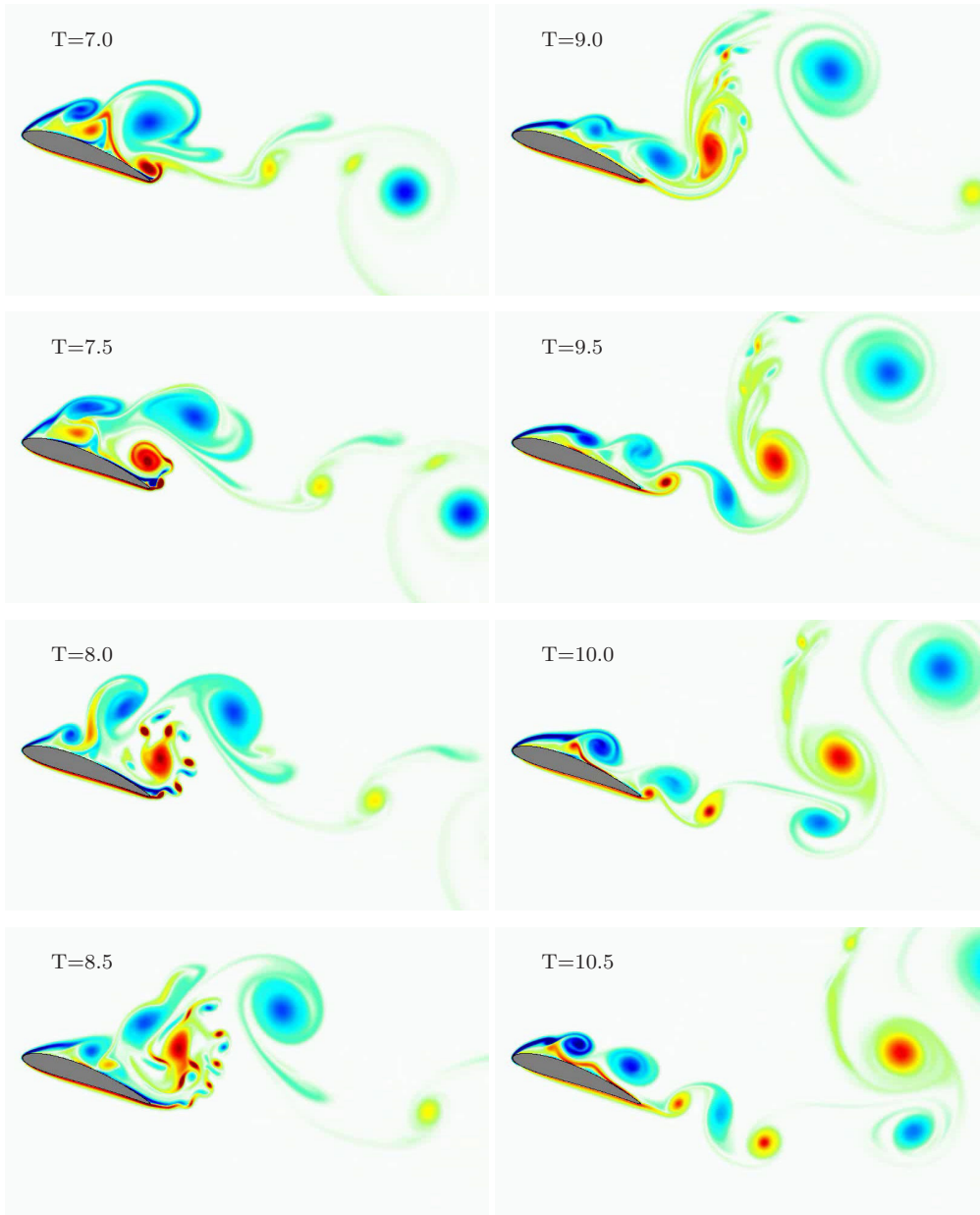
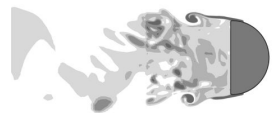


FIGURE 3.9 – (cont.) Flow past a stalled airfoil at $Re = 5,000$. Time evolution of the vorticity field.



to those observed for the wake flow past a flat plate [57]. There is, in fact, shedding of opposite sign vortices from the leading edge region, corresponding to vorticity from the lower (stronger) and upper (weaker) boundary layers (see also Fig. 3.8). This is thus a wake flow combined with a shear layer flow, with Kelvin-Helmoltz instabilities.

3.4.3 Flow past a cylinder at $Re = 1,000$

We now show the application of the hybrid Eulerian-Lagrangian method using the velocity-pressure formulation in the Eulerian domain. This case is clearly more challenging as we solve for different primitive variables in the Eulerian and Lagrangian domains. The test case is again the flow past a circular cylinder, here at $Re = 1,000$.

Figure 3.10 shows the vorticity field reconstructed from information in the Eulerian and Lagrangian domains. The pressure field in the Eulerian domain is also shown. The transition from the Eulerian to the Lagrangian domain slightly more visible than when the vorticity-velocity formulation is used in both domains. It is nevertheless very good: especially if one remembers that the coupling in the Eulerian domain is done at the velocity level, and that we are here looking at a derivative (i.e., the vorticity).

3.5 Summary

We have presented a new approach that combines a grid-based solver for the near-wall regions and a particle solver for the wake. The innovation comes from the fact that the Lagrangian domain covers the whole computational domain but is under-resolved in the near-wall region. The Eulerian grid-based solver is used to capture the dynamics of the boundary and separation regions. Due to the fact that the Lagrangian domain covers the whole computational domain, the boundary conditions for the Eulerian domain can be retrieved easily. A dual time stepping method was also introduced; this is the best way to obtain an efficient global algorithm. Finally, a very good behavior of the algorithm was observed in terms of robustness and information transfer between the domains, for long time simulations.

The coupling was first applied to the case where both the Lagrangian and the Eulerian domain use the vorticity-velocity formulation. It was then ex-

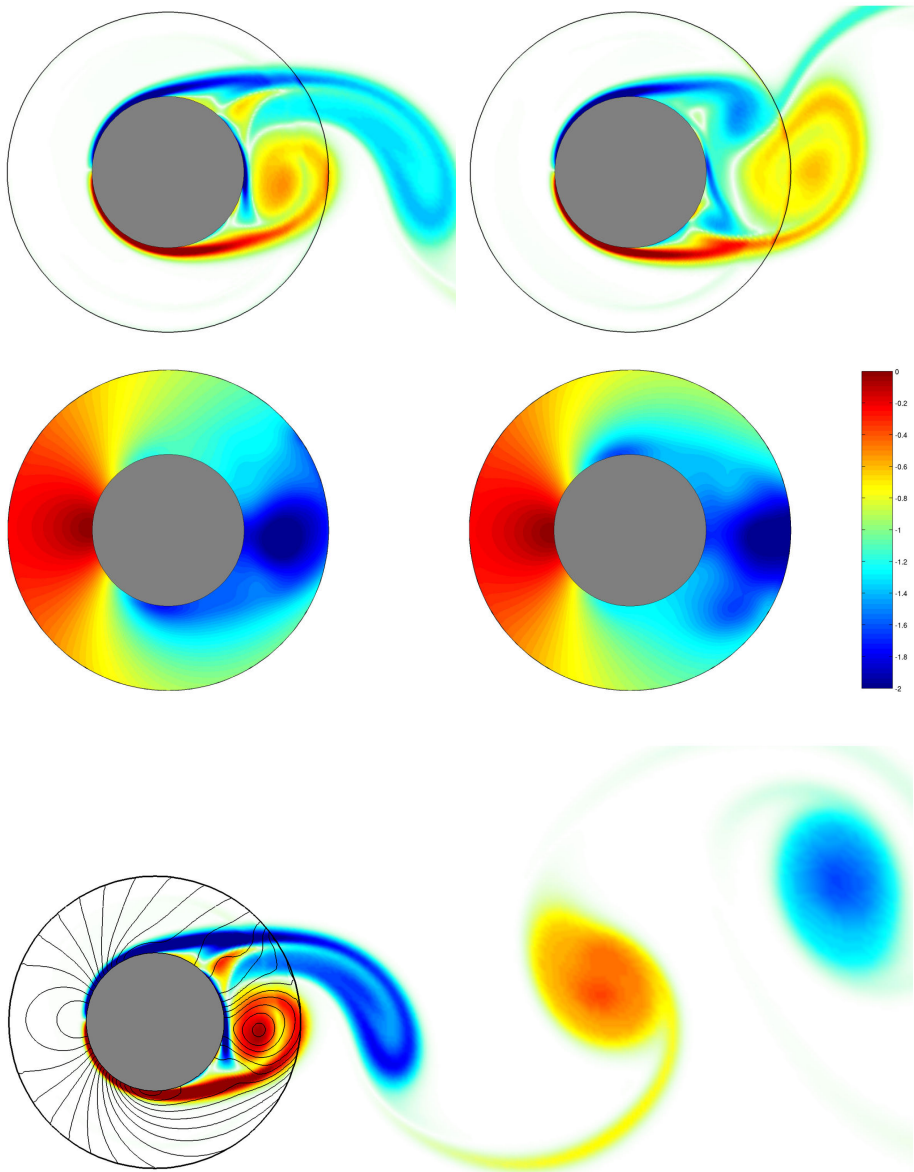


FIGURE 3.10 – Flow past a cylinder at $Re = 1,000$. The velocity-pressure approach is used in the Eulerian domain. Top: vorticity field reconstructed from information in the Eulerian and Lagrangian domains. Middle: pressure field in the Eulerian domain. Bottom: reconstructed vorticity field (Eulerian and Lagrangian domains) and pressure contours in Eulerian domain.



tended to the case where the velocity-pressure formulation is used in the Eulerian domain (while the Lagrangian domain is obviously still in the vorticity-velocity formulation).

The extension of the method to 3-D is, in principle, straightforward. It requires however that one has access to an Eulerian solver for the incompressible Navier-Stokes equations. A good candidate is the solver developed by CENAERO. However, at the time when this solution was considered, the CENAERO code only existed as a compressible solver and could therefore not be used. An incompressible version of the code was developed, in the mean time, by Marchandise *et al.* [61] and is now available. Due to time limitations, it could however not be used in the frame of this work.

An alternate solution was to use the finite volume (DNS) solver for the incompressible Navier-Stokes equations developed by Wu *et al.* [102] (see Section 7.2). This code was adapted and used for investigations of a RANS-LES approach in Chapter 7. Communication tools were implemented in order to couple the Eulerian code (written in Fortran and using OpenMP parallelization) with our Lagrangian code (written in C and using MPI parallelization). While working on this coupled approach, it appeared that the Lagrangian domain would benefit from a better control of the redistribution grid size. This led to the development of the hierarchical redistribution method, presented in Chapter 4. The 3-D hybrid Eulerian-Lagrangian approach could thus not be tested yet, but we are convinced of its success and we hope that we will be able to prove it in the near future.

CHAPTER 4

Redistribution on hierarchically refined grids

4.1 Motivation

Lagrangian vortex methods have the advantage that they automatically adapt to the flow field: computational elements are present only where they are required (i.e., where the vorticity field is non zero) and the elements are “naturally” convected with the flow.

Nevertheless, this self-adaptivity has to be controlled. The first reason arises from accuracy considerations: in order for vortex methods to converge to solutions of the Navier-Stokes equations, the particle cores should overlap. Since the flow strain causes local clustering and depletion of particles, it is necessary to control the particle distribution. Beyond this particularity of Lagrangian methods, there is another consideration that is related to both accuracy and efficiency: it should be possible to control the particle resolution depending on the local physics of the flow. For example, bluff body flow simulations require a very fine resolution in the boundary layers and separating shear layers while the wake regions, where the vorticity gradients are much lower, only require a coarser particle size.



Several approaches exist to maintain the spatial regularity of the particles, but the most effective one is probably the particle redistribution approach [19, 98]. This strategy consists in replacing the distorted set of vortex particles with a new set of particles at fresh locations on a regular grid. In practice, the strength of each particle in the old set is locally redistributed onto a cartesian grid (the new particles) using high-order interpolation kernels so as to ensure the conservation of a certain number of moments.

In order to achieve a varying spatial resolution, a simple approach is to use a mapping of the physical domain during the redistribution stage [59, 21, 77, 19, 98]:

- The old set of particles in the physical domain is transformed to the mapped domain;
- The particles are redistributed on a cartesian grid in the mapped domain;
- The new set of particles in the mapped domain is transformed back to the physical domain using the inverse mapping function.

By choosing a particular mapping function, one can thus control the spatial resolution of the computation (e.g., a polar mapping can be used to ensure a decreasing resolution in the wake for an external flow). The mapping should however satisfy the following properties:

- The resulting particle size should vary smoothly in space. This condition is required to maintain the second-order accuracy of the PSE [36, 11, 22];
- The mapping should be conformal (i.e., conserve the angles) in order to conserve the higher-order moments during the redistribution;
- The mapping should be invertible.

These constraints reduce the flexibility of a single global mapping function. It is therefore not well suited for situations where a more localized “grid” refinement is required, such as the flow past multiple bodies.

In order to address this flexibility issue, domain decomposition approaches were developed [18, 59, 27]: each sub-domain has a local redistribution mapping that corresponds to different grid-size requirements in the flow. The increased flexibility in these approaches is however at the expense of a (complex) global domain decomposition algorithm.

Bergdorf *et al.* [7] have recently investigated two novel adaptive approaches for Lagrangian particle methods. Both approaches are developed with the idea

that they dynamically adapt the refinement to the flow field requirements. It thus goes one step beyond the non-automatic adaptivity discussed in this chapter.

Their first approach uses an adaptive global mapping that is dynamically adapted to the flow field (using a discrete representation of the mapping) in the same spirit as r -adaptivity used in finite element methods. Because the discrete mapping is not invertible, the governing equations need to be transformed and solved in the mapped space. This dynamic mapping adaptivity suffers from similar drawbacks as the Lagrangian particle method itself: it is necessary to maintain a certain regularity of the mapping (this can be done by applying smoothing operators) but, at some point, the distortion of the mesh becomes too important and the mapping has to be “reset”. The adaptivity problem is thus pushed one level further, but is not really solved.

The second approach is in the spirit of Adaptive Mesh Refinement: the idea is to define blocks of uniform grid-sizes (that are defined dynamically based on error estimates). Each block has its own set of particles. The block overlapping and the particle remeshing in the overlapping zones is used to transfer vorticity between blocks: it plays the role of vorticity boundary conditions on each block in the same way as domain decomposition techniques as in [18, 59, 27].

In [7], both approaches are illustrated on a 2-D inviscid flow of an elliptical vortex and seem to provide good results. However, in the AMR approach, the velocities at the location of fine-size particles are evaluated using coarse-size particles. The same remark seems applicable to the adaptive global mapping approach as the velocity field is evaluated in the physical domain with probably a coarse regularization. This means that the Navier-Stokes equations are not actually solved at the fine level: it essentially is a method that transports (very accurately) a vorticity field with strong gradients but with a low-pass filtered velocity field (and not with the actual corresponding “fine-level” velocity field). The real use of such an approach is therefore questionable.

In this chapter, we will develop a new redistribution approach where the particles are redistributed on a locally refined cartesian grid. This approach will allow to arbitrarily refine the particle resolution where the flow physics require it. So far, it is not a dynamically adapting method; but some refinement criterion could be integrated to do so in the future. One important difference with the AMR method developed by Bergdorf *et al.* [7] is that there is no domain decomposition involved: at all times, a single set of particles is used. It therefore does not require any modification to the global algorithm of a



classical lagrangian vortex method. Of course, this novel approach also has its limitations: we will see what they are and where they arise from.

4.2 Conservative redistribution scheme on hierarchically refined grids

The objective of this section is to develop a procedure to interpolate an old vorticity field described by a distorted particle set on a new set placed on a hierarchically refined grid. The aim being to obtain an accurate and smooth representation on the new particle set, by also conserving the flow quantities (total circulation, linear and angular impulse) both globally and locally. Before presenting the procedure for hierarchically refined grids, we recall the redistribution procedure on uniform grids.

4.2.1 Classical scheme

The classical redistribution procedure [19, 98] is quite simple. Let's first consider the one-dimensional problem. When a distorted particle set, that was originally on a grid with uniform spacing (h), is redistributed on the same uniform grid, the weights of the new particles are obtained as follows:

$$\alpha_i = \sum_p \tilde{\alpha}_p W\left(\frac{x_i - \tilde{x}_p}{h}\right) \quad (4.1)$$

where, x_i ($\tilde{x}_p$) and α_i ($\tilde{\alpha}_p$) are respectively the location and the weight of the new (old) particles and $W(x)$ is an interpolation kernel. A good choice for the interpolation kernel has proven to be the so called M'_4 scheme developed by Monaghan [68, 67]; it will be used here. This kernel is continuously differentiable and ensures that when used in (4.1) on a uniform grid, the first three moments of the particle distribution are conserved (total circulation, linear and angular impulse). The M'_4 kernel is defined by:

$$M'_4(x) = \begin{cases} 1 - \frac{5}{2}|x|^2 + \frac{3}{2}|x|^3 & \text{if } |x| \leq 1 \\ \frac{1}{2}(2 - |x|)^2(1 - |x|) & \text{if } 1 < |x| \leq 2 \\ 0 & \text{if } |x| > 2 \end{cases} \quad (4.2)$$

As can be seen from this formula, the M'_4 kernel has a width 4: in 1-D, an old particle is thus redistributed onto four new particles. Another commonly used

interpolation kernel is the Λ_3 kernel: it has the same width, conserves the first four moments of the particle distribution but is not continuously differentiable and has therefore a slightly less smooth behavior.

The extension of the classical redistribution procedure to 2-D and 3-D is easily performed by applying the tensor product of the 1-D schemes in each direction: $W(\mathbf{x}) = \prod_{k=1}^d W(x_{(k)})$, where d is the dimension of the problem and (k) indicates the k^{th} coordinate.

4.2.2 General scheme on hierarchical grids

We now consider redistribution grids that are constructed from a hierarchical representation of space: a binary tree in 1-D, a quadtree in 2-D and an oct-tree in 3-D. This allows for a flexible yet structured grid representation. The redistribution grid is constructed as follows (we here consider the 2-D case):

- Let $h^{(0)}$ be the particle size at the coarsest level, $l = 0$, and $h^{(l)} = h^{(0)} 2^{-l}$ the particle size at level l ;
- a square box of size $b = h^{(0)} 2^L$ is placed in the computational domain, where L is the smallest integer such that the box covers all the particles;
- the grid representation is obtained by recursively subdividing the initial box into four quadrants;
- at each level, the resulting quadrants are inspected: if a uniform particle size $h^{(l)}$ can be used over a box it is tagged (as a region refined at level l) and the recursive process is stopped for that box, otherwise the box is further subdivided;
- In order to obtain a consistent discrete representation of the vorticity field, new particles are located at the center of each grid cell (as shown in Fig. 4.1): the size, $h_p = h^{(0)} 2^{-l}$, and the surface, $S_p = h_p^2$, associated with each particle is well defined even for particles located next to the interface between two levels. This would not be the case if the particles were defined at cell summit locations.

Once the hierarchical grid structure is built, each particle location is examined with respect to the grid. The target level and target size of an old particle are respectively defined as the level of refinement and the corresponding size of the grid (at the old particle location). The neighboring grid points (the target particles) that will be affected by the redistribution procedure are then



identified. These operations can be performed efficiently using the grid tree data structure. The weights of the target particles are then determined through a conservative redistribution scheme for hierarchically refined grids. This new scheme is developed below.

The redistribution operator being linear, we can develop the general redistribution procedure for a single particle of unitary weight. We define the following symbols:

$\tilde{\mathbf{x}}$	Position of old particle
$\tilde{l}$	Level of old particle
$\tilde{l}^*$	Target level of old particle (i.e., new level at old particle position)
$\tilde{h}$	Size of old particle
$\tilde{h}^*$	Target size of old particle (i.e., new particle size at old particle position)
$\mathcal{P}$	Subset of new particles affected by redistribution of old particle
n	Number of new particles affected by redistribution of old particle
i	Local particle index in subset $\mathcal{P}$
$\mathbf{x}_i$	Position of new particle i
l_i	Level of new particle i
$l_{\max}$	Maximum (finest) level of new particles in subset $\mathcal{P}$
$l_{\min}$	Minimum (coarsest) level of new particles in subset $\mathcal{P}$
h_i	Size of new particle i
α_i	Redistributed weight of new particle i
β_i	Target weight of new particle i
$l^\bullet$	Effective level at which the redistribution kernel will be evaluated
$h^\bullet$	Size of grid at effective level at which the redistribution kernel will be evaluated

Figure 4.1 shows an example of a 2-D hierarchical grid. In this example the particle size at the coarsest level is $h^{(0)} = 1$ and the upper right quadrant is refined at level $l = 1$ and thus $h^{(1)} = 0.5$. A coarse particle ($\tilde{l} = 0$) is located in the coarse region of the grid. Its target level is thus $\tilde{l}^* = 0$. There are $n = 28$ neighboring grid points that will be affected by the redistribution procedure. The target particles are both at the coarse level $l_{i \in [1,12]} = 0$ and at the fine level $l_{i \in [13,28]} = 1$.

Developing a conservative redistribution procedure for hierarchically refined grids is more complex because different configurations can occur depending on

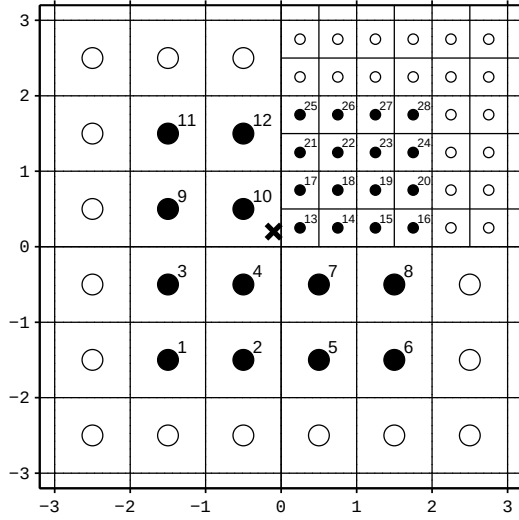


FIGURE 4.1 – 2-D redistribution of a particle on a hierarchically refined grid. In this example the old particle ($\times$) will be redistributed on 28 new particles ($\bullet$). The particles not affected by the redistribution of the particle are also shown ($\circ$).

the respective levels of the original and the new particles. We first investigate the cases where the old particle is redistributed on a uniform grid, possibly at a different level from the original level of the old particle (the equations are written in 1-D):

- (i) The old particle and the new particles in $\mathcal{P}$ are at the same level (Fig. 4.2a):

$$\begin{cases} \tilde{l} &= \tilde{l}^* &= l_{\max} = l_{\min} \\ \tilde{h} &= \tilde{h}^* &= h_{i \in [1, n]} \end{cases}$$

This is the simplest and most frequent case, the effective level is equal to the original particle level, $l^\bullet = \tilde{l}$, and thus $h^\bullet = \tilde{h}$. The classical redistribution scheme can be used:

$$\alpha_i = W\left(\frac{x_i - \tilde{x}}{h^\bullet}\right)$$

- (ii) The new particles in $\mathcal{P}$ are at the same level, but the old particle is at a finer level (Fig. 4.2a):

$$\begin{cases} \tilde{l} &> \tilde{l}^* &= l_{\max} = l_{\min} \\ \tilde{h} &< \tilde{h}^* &= h_{i \in [1, n]} \end{cases}$$



The old particle is simply “shifted” to the coarse level and is redistributed there with the classical scheme. This is done by setting the effective level to the target redistribution level, $l^\bullet = \tilde{l}^*$, and thus $h^\bullet = \tilde{h}^*$. The redistributed weights are then again given by:

$$\alpha_i = W\left(\frac{x_i - \tilde{x}}{h^\bullet}\right)$$

This ensures that the scheme is still conservative. It can also be seen as applying a low-pass filter (of width $h^\bullet$, the size of the coarse target level) on the fine particle in order to avoid aliasing errors that occurs when a function is “down-sampled”.

- (iii) The new particles in $\mathcal{P}$ are at the same level, but the old particle is at a coarser level (Fig. 4.2b):

$$\begin{cases} \tilde{l} < \tilde{l}^* & = l_{\max} = l_{\min} \\ \tilde{h} > \tilde{h}^* & = h_{i \in [1, n]} \end{cases}$$

When a coarse particle has to be redistributed on a fine grid, the classical interpolation scheme can be used at the original particle level, $l^\bullet = \tilde{l}$, and thus $h^\bullet = \tilde{h}$. The number of particles affected by the redistribution scheme is multiplied by a factor $2^{l^\bullet - \tilde{l}^*} = h^\bullet / \tilde{h}^*$, the weight redistributed to each particle has thus to be reduced accordingly:

$$\alpha_i = \frac{\tilde{h}^*}{h^\bullet} W\left(\frac{x_i - \tilde{x}}{h^\bullet}\right)$$

By using, $h^\bullet = \tilde{h}$, we ensure that the width of the redistribution stencil reflects that of the original coarse particle size.

These three cases can be generalized as follows (also generalized for d dimensions):

$$\begin{cases} l^\bullet & = \min(\tilde{l}, \tilde{l}^*) \\ h^\bullet & = \max(\tilde{h}, \tilde{h}^*) \\ \alpha_i & = \left(\frac{\tilde{h}^*}{h^\bullet}\right)^d W\left(\frac{\mathbf{x}_i - \tilde{\mathbf{x}}}{h^\bullet}\right) \end{cases} \quad (4.3)$$

As can be seen, the three cases collapse to a very simple set of equations that can easily be programmed. This set of equations is also at the base of the general “mixed” case where target particles at different levels are involved.

When it comes to the general “mixed” case, such as the 2-D example in Fig. 4.1 and the 1-D example in Fig. 4.2c, the classical redistribution kernels

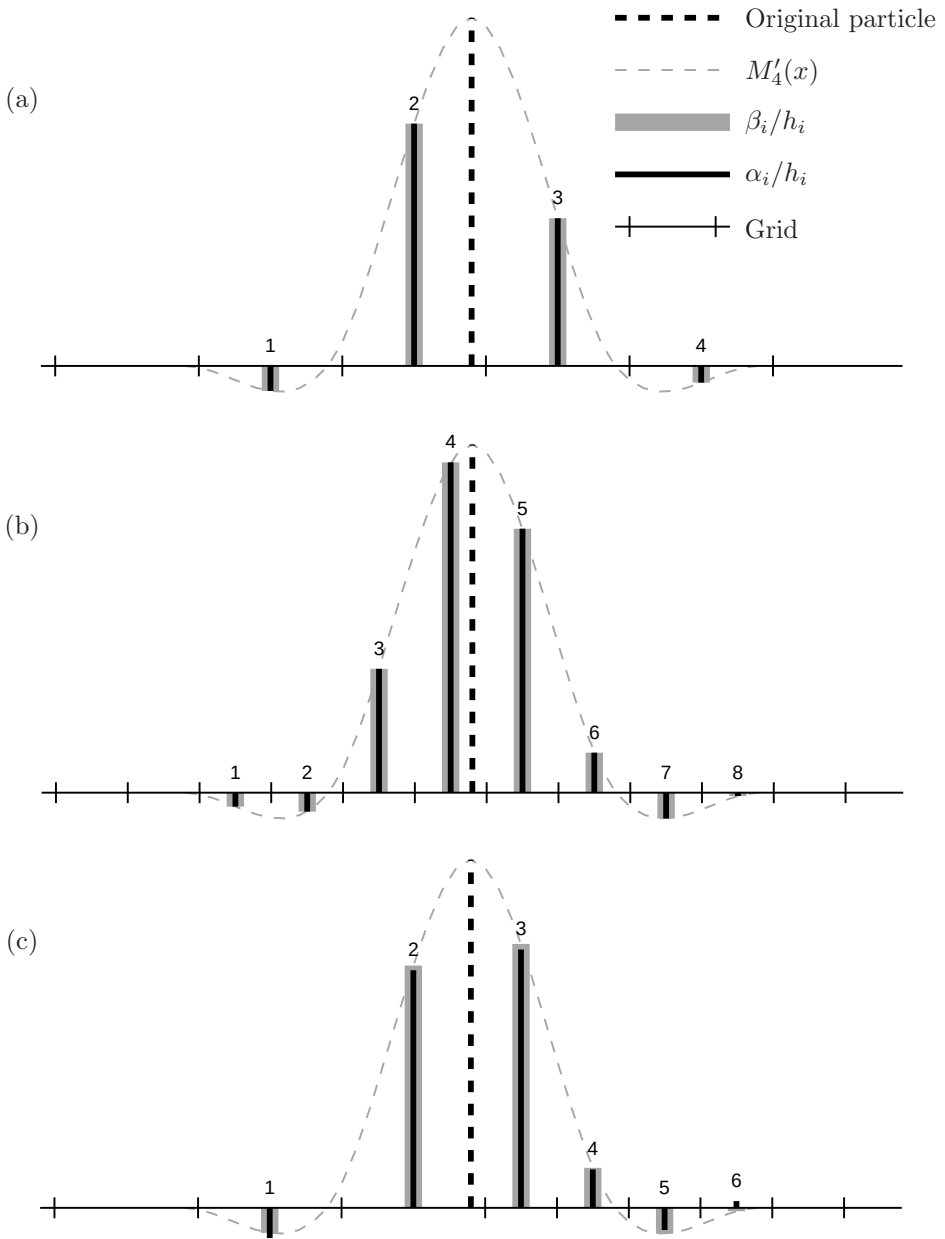
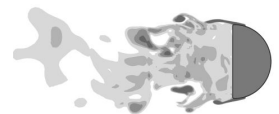


FIGURE 4.2 – 1-D redistribution of a particle on three different grids. The target weights, β_i , (gray thick) and the conservative weights, α_i , (black thin), are shown. **Top:** Uniform coarse grid; **Middle:** Uniform fine grid (a coarse particle is redistributed on a fine grid); **Bottom:** Mixed grid, in this case the target and conservative weights differ (slightly).



cannot be used anymore: they would not be conservative as they were developed for uniform grid spacings only. We thus have to develop new redistribution schemes such as to obtain an accurate and smooth representation on the new particle set and by also conserving the flow quantities. The problem being that a very large number of canonical configurations exist and one clearly cannot develop a particular scheme for each of them.

The idea is to build and to solve a linear system for each configuration in order to determine the weights of the target particles, $\alpha_{i \in [1, n]}$, such as to satisfy:

$$\begin{cases} \sum_{i=1}^n \alpha_i &= 1 & (1 \text{ constraint}) \\ \sum_{i=1}^n x_{(k), i} \alpha_i &= 0 & (d \text{ constraints}) \\ \sum_{i=1}^n x_{(k), i}^2 \alpha_i &= 0 & (d \text{ constraints}) \end{cases} \quad (4.4)$$

If these $m = 2d + 1$ equations are satisfied it ensures that the scheme conserves total circulation, linear and angular impulse. However, if one applies this system to the 1-D example shown in Fig. 4.2c, one has $m = 3$ equations with $n = 6$ unknowns: the system is under-determined. The 2-D example in Fig. 4.1 is even worse: there are $m = 5$ equations for $n=28$ unknowns!

In order to raise the underdetermination, we add the constraint that the redistributed field has to be “a smooth representation” of the original field. This constraint being rather subjective, it does not provide any additional equations as such. Let’s define a particle weight distribution, β_i , such that it provides a “smooth representation” of the particle to be redistributed (but not necessarily conservative). A natural choice is to use the following definition based on Eq. (4.3):

$$\begin{cases} l^\bullet &= \min(\tilde{l}, \tilde{l}^*) \\ h^\bullet &= \max(\tilde{h}, \tilde{h}^*) \\ \beta_i &= \left(\frac{h_i}{h^\bullet}\right)^d W\left(\frac{\mathbf{x}_i - \tilde{\mathbf{x}}}{h^\bullet}\right) \end{cases} \quad (4.5)$$

The only difference with Eq. (4.3) being that the local particle size, h_i , is used instead of $\tilde{h}^*$. For uniform grids, the Eqs. (4.3) and (4.5) are thus identical. For non uniform grids (and evidently for uniform grids), Eq. (4.5) leads to a smooth distribution of β_i/h_i^d :

$$\frac{\beta_i}{h_i^d} = \left(\frac{1}{h^\bullet}\right)^d W\left(\frac{\mathbf{x}_i - \tilde{\mathbf{x}}}{h^\bullet}\right)$$

This quantity represents the actual value of the transported field (recall that the particle strength is the integral of the field over the particle volume) and will thus have the same smoothness properties as the interpolation kernel $W(x)$.

At this point we have a set of estimated weights, β_i , that should provide a smooth redistributed field and has the additional property of falling back to the previously developed scheme when applied to uniform grids. Figure 4.2 shows the resulting estimated weights for three cases: a coarse (or fine) particle redistributed on a coarse grid, a fine particle redistributed on coarse grid and a coarse particle redistributed on a mixed grid. For the first two cases, the estimated weights and the conservative weights are identical. For the last case, the estimated redistributed field β_i/h_i is still “smooth” even if a mixed grid is used.

The idea is now to obtain the conservative weights, α_i , by minimizing the difference with the estimated weights, β_i , in a least mean square sense and enforcing the constraints given by Eq. (4.4) exactly. The minimization function is thus:

$$\min_{\alpha_i \in \mathbb{R}} \left(\sum_{i=1}^n \left(\frac{\alpha_i - \beta_i}{p_i} \right)^2 \right), \quad (4.6)$$

where p_i is a penalization parameter. We here chose the following definition:

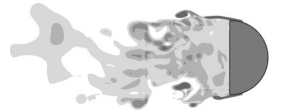
$$p_i = \frac{h_i^d}{1 + |\mathbf{x}_i|/|\mathbf{x}|_{\max}}, \quad (4.7)$$

where, $|\mathbf{x}|_{\max} = \max(|\mathbf{x}_i|)$, $i \in \mathcal{P}$. Introducing h_i^d ensures that we minimize the error on the actual field values and not on the particle weights. We also introduce a penalty related to the distance of the particles to avoid large corrections on the outer particles.

Introducing m Lagrange multipliers for the linear constraints, we write the Lagrangian function, F , as

$$\begin{aligned} F = & \sum_{i=1}^n \left(\frac{\alpha_i - \beta_i}{p_i} \right)^2 + \lambda_0 \left(\sum_{i=1}^n \alpha_i - 1 \right) \\ & + \sum_{k=1}^d \left(\lambda_{(k),1} \sum_{i=1}^n x_i \alpha_i \right) + \sum_{k=1}^d \left(\lambda_{(k),2} \sum_{i=1}^n x_i^2 \alpha_i \right). \end{aligned} \quad (4.8)$$

The solution of the constrained least square minimization problem leads to the following linear system of $N = n + m$ equation with $N = n + m$ unknowns:



$$\left\{ \begin{array}{l} \alpha_i + \lambda_0 \frac{p_i^2}{2} + \sum_{k=1}^d \left(\lambda_{(k),1} \frac{p_i^2 \mathbf{x}_i}{2} \right) \\ \quad + \sum_{k=1}^d \left(\lambda_{(k),2} \frac{p_i^2 \mathbf{x}_i^2}{2} \right) = \beta_i, \quad (n \text{ equations}) \\ \sum_{i=1}^n \alpha_i = 1, \quad (1 \text{ equation}) \\ \sum_{i=1}^n x_{(k),i} \alpha_i = 0, \quad (d \text{ equations}) \\ \sum_{i=1}^n x_{(k),i}^2 \alpha_i = 0, \quad (d \text{ equations}) \end{array} \right. \quad (4.9)$$

This system can become rather large: the 2-D example in Fig. 4.1 leads to $N = 33$ (the corresponding 3-D case would lead to $N = 127$). Recalling that this system has to be solved for each particle involved in a mixed scheme, it could end up having prohibitive computational cost. However, the system can actually be reduced due to its particular structure. Let's define, the following matrices ($\mathbf{S}$ by its elements s_{ij} , and $\mathbf{C}$ by its lines $\mathbf{c}_i$) and vectors ($\boldsymbol{\lambda}$ and $\mathbf{r}$):

$$\begin{aligned} s_{ij} &= \delta_{ij} p_i & \text{thus, size}(\mathbf{S}) &= [n \times n] \\ \mathbf{c}_i &= \left\{ 1, x_{(k),i}, x_{(k),i}^2 \right\} & \text{thus, size}(\mathbf{C}) &= [n \times m] \\ \boldsymbol{\lambda} &= \left\{ \lambda_0, \lambda_{(k),1}, \lambda_{(k),2} \right\}^T & \text{thus, size}(\boldsymbol{\lambda}) &= [m \times 1] \\ \mathbf{r} &= \left\{ 1, 0_{(k)}, 0_{(k)} \right\}^T & \text{thus, size}(\mathbf{r}) &= [m \times 1] \end{aligned} \quad (4.10)$$

With these definitions, we rewrite the system of equations (4.9) in matrix-vector notation:

$$\left(\begin{array}{c|c} \mathbf{I} & \frac{1}{2} \mathbf{S} \mathbf{C} \\ \hline [n \times n] & [n \times m] \\ \hline \mathbf{C}^T & \mathbf{0} \\ [m \times n] & [m \times m] \end{array} \right) \left(\begin{array}{c} \boldsymbol{\alpha} \\ \hline [n \times 1] \\ \hline \boldsymbol{\lambda} \\ [m \times 1] \end{array} \right) = \left(\begin{array}{c} \boldsymbol{\beta} \\ \hline [n \times 1] \\ \hline \mathbf{r} \\ [m \times 1] \end{array} \right) \quad (4.11)$$

Notice that $\mathbf{S}$ is a diagonal matrix and $\mathbf{I}$ is the unit matrix. By inspection, one can easily show that the system is actually equivalent to the following matrix identities:

$$\mathbf{D} = \frac{1}{2} \mathbf{C}^T \mathbf{S} \mathbf{C} \quad (4.12)$$

$$\mathbf{D} \boldsymbol{\lambda} = \mathbf{C}^T \boldsymbol{\beta} - \mathbf{r} \quad (4.13)$$

$$\boldsymbol{\alpha} = \boldsymbol{\beta} - \frac{1}{2} \mathbf{S} \mathbf{C} \boldsymbol{\lambda} \quad (4.14)$$

where Eq. (4.13) is the only system of m equations by m unknowns to invert. The computational cost of the generalized scheme is thus perfectly tractable even if one has to solve a small linear system for a large number of particles involved in mixed schemes.

If we examine again Fig. 4.2, we see that there are only small differences between the conservative weights, α_i , and the estimated weights, β_i . Nevertheless the adjusted weights, α_i , ensure that the scheme is conservative, while still being very close to the estimated “smooth” target weights.

In the end, the general conservative redistribution procedure of a particle on a hierarchically refined grid can be summarized as follows:

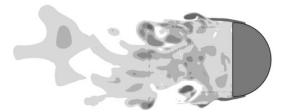
- Determine old particle target level, $l^\bullet$;
- Determine $\mathcal{P}$, the subset of particle affected by the redistribution of the old particle;
- Compute β_i , the estimated target weights, using Eq. (4.5);
- **If** all target particles are at the same level, **then** $\alpha_i = \beta_i$ (the estimated target weights do not need to be adjusted),
else compute the adjusted weights, α_i , using Eqs. (4.12) to (4.14).

Notice that, when all target particles are at the same level, if Eqs. (4.12) to (4.14) were used to compute adjusted weights (even if in that case the estimated weights are already conservative), one would obtain $\alpha_i = \beta_i$. This is actually not mandatory, but shows somehow that we have developed a consistent general scheme.

4.2.3 Correction procedure

Despite the careful development of a conservative smooth redistribution scheme in the previous section, some high frequency oscillations are introduced at the fine level for new particles that were involved in mixed schemes. These oscillations are related to the inherent discontinuity of the grid itself: the particle size goes from h to $h/2$ over one grid cell.

Figure 4.3 shows a 1-D example of this oscillation phenomenon: a set of coarse uniformly spaced particles with constant weights is redistributed on a hierarchically refined grid. The resulting redistributed weights present a large oscillation on the fine particles next to the fine-coarse interface. Notice that



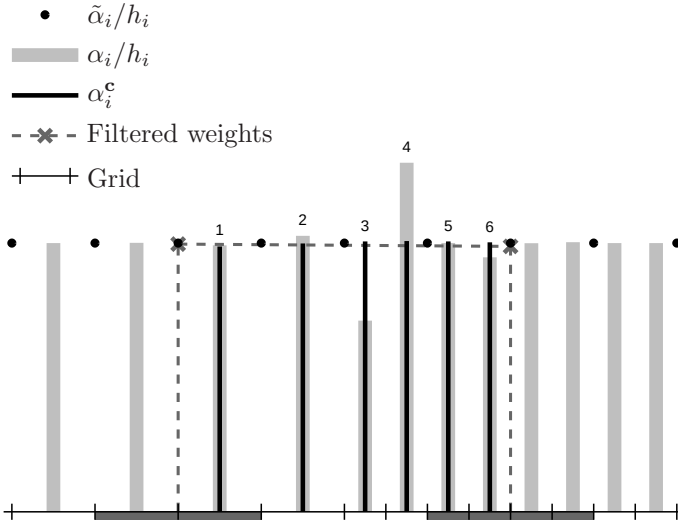


FIGURE 4.3 – 1-D example of the correction procedure for 6 particles involved in a mixed scheme. A set of uniformly spaced particles with constant weights, $\tilde{\alpha}_i$, is redistributed on a hierarchically refined grid. The resulting redistributed weights, α_i , present a large oscillation next to the fine-coarse interface. The “filtered” weights are obtained by linear interpolation from the averaged weights over the dark gray zones. The corrected weights, α_i^c , do not oscillate while still being conservative.

away from the interface, the redistributed weights are constant again, for the coarse particles as for the fine ones.

In order to correct this spurious oscillation, the particles involved in the mixed scheme are tagged during the redistribution procedure. Once all particles from the old set have been redistributed, the new particles are checked. For each “mixed” particle, p , a correction procedure is applied.

The first step is to identify the subset of neighboring particles, $\mathcal{P}_p$, that straddles across the fine-coarse interface (as shown in Fig. 4.4). Once again, this can be performed efficiently using the tree data structure. The next step is to define for each particle in $\mathcal{P}_p$ a target “filtered” weight, α_i^f : the filtering is such that the oscillations are removed, but it is not necessarily a conservative operation. The filtered weights are obtained by linearly interpolating the averaged weights obtained at the summits of the correction zone (as shown in

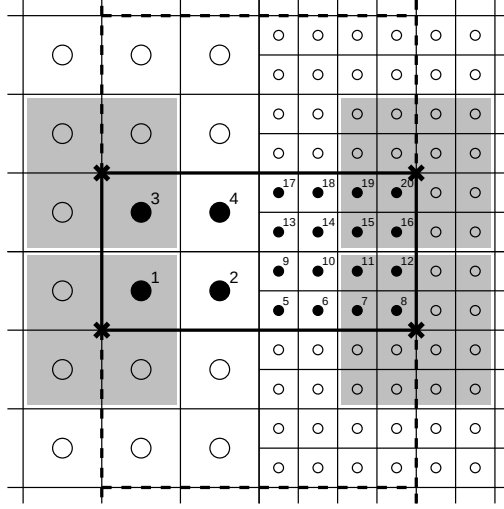


FIGURE 4.4 – Adjustment of 20 particles involved in a mixed scheme. The “filtered” weights, computed at the summits (×) of the adjustment zone (plain thick rectangle), are obtained by averaging over the gray zones. The filtered target weight, α_i^f , at each particle location (●) is obtained by bilinear interpolation. Other adjustment zones are also indicated (dashed thick rectangles).

Fig. 4.4). The final step consists in the correction of the redistributed weights:

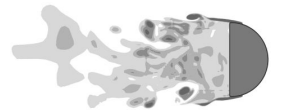
$$\alpha_i^c = \alpha_i + \varepsilon_i, \quad (4.15)$$

where ε_i is the weight correction. We will obviously require the corrected weights, α_i^c , to be conservative. This is obtained by imposing that,

$$\begin{cases} \sum_{i=1}^n \varepsilon_i = 0 & (1 \text{ constraint}) \\ \sum_{i=1}^n x_{(k),i} \varepsilon_i = 0 & (d \text{ constraints}) \\ \sum_{i=1}^n x_{(k),i}^2 \varepsilon_i = 0 & (d \text{ constraints}) \end{cases} \quad (4.16)$$

Once again, the idea is to obtain the corrected conservative weights, α_i^c , by minimizing the difference with the filtered target weights, α_i^f , in a least mean square sense and enforcing the constraints given by Eq. (4.16) exactly. The minimization function is thus:

$$\min_{\varepsilon_i \in \mathbb{R}} \left(\sum_{i=1}^n \left(\frac{(\alpha_i + \varepsilon_i) - \alpha_i^f}{p_i} \right)^2 \right) \quad (4.17)$$



where, p_i is defined by (4.7). Obviously this constrained minimization problem is exactly the same as the one defined by Eqs. (4.4) and (4.6) in the previous section: one just has to replace α_i by ε_i , β_i by $(\alpha_i^f - \alpha_i)$ and, in the right hand side of the first constraint, 1 by 0. The solution for ε_i can thus be obtained by Eqs. (4.12) to (4.14).

If we now turn again to Fig. 4.3, we see that the adjusted weights, α_i^c , essentially eliminate the oscillatory behavior (while still conserving the first three moments of the particle distribution). The correction procedure thus acts as a low pass filter: it actually introduces some diffusion when information is transferred from coarse level to fine level regions. Also, the correction operator is not idempotent: if the operator is applied again, an additional correction will be introduced.

Finally, it should be mentioned that the particles in subset $\mathcal{P}_p$ are tagged as “corrected” after the correction procedure. When scanning the full particle set for the next “mixed” particle, they will be skipped. The structured way of defining the correction regions (based on the hierarchical grid representation) also ensures that two correction regions do not overlap (Fig. 4.4).

4.3 Cross level particle interactions

As already mentioned, the present method has the advantage of using a single set of particles without the need for domain decomposition approaches, where each level of refinement would be represented by its own particles. However, when it comes to field evaluations, some care has to be taken for the direct interactions between particles that are at different levels of refinement.

We first consider the treatment of the diffusion. It is based on the technique of particle strength exchange (PSE) [36, 11, 22]. In this algorithm, the Laplacian operator is approximated by an integral operator:

$$\nabla^2 \omega(\mathbf{x}) = \frac{2}{\sigma^2} \int (\omega(\mathbf{y}) - \omega(\mathbf{x})) \eta_\sigma(\mathbf{x} - \mathbf{y}) d\mathbf{y} + \mathcal{O}(\sigma^k) \quad (4.18)$$

where, $\eta_\sigma(\mathbf{x}) = \eta(\mathbf{x}/\sigma)/\sigma^d$, σ is the kernel radius, and $\eta(\mathbf{x})$ is a kernel constructed to approximate the given differential operator to a certain order of accuracy, k [39, 98]. In practice however, one always uses $k = 2$ as higher order accuracy requires oscillating kernels. As already mentioned in Chapter 1, we here use the Gaussian kernel, $\eta(\mathbf{x}) = 1/(2\pi)^{d/2} \exp(-|\mathbf{x}|^2/2)$, which provides a

second order approximation of the Laplacian operator. Equation (4.18) is discretized by a midpoint quadrature over the particles, resulting in the following operator:

$$\nabla_h^2 \boldsymbol{\omega}(\mathbf{x}_p) = \frac{2}{\sigma^2} \sum_q (\boldsymbol{\omega}(\mathbf{x}_q) - \boldsymbol{\omega}(\mathbf{x}_p)) \eta_\sigma(\mathbf{x}_p - \mathbf{x}_q) V_q \quad (4.19)$$

where V_q is the particle volume. In practice, due to the rapid decrease of $\eta(\mathbf{x})$, the sum only involves a contribution from a subset, $\mathcal{P}_p$, of particles close to $\mathbf{x}_p$. The quadrature approximation of the integral operator induces an additional error. Provided that $\sigma/h > \sqrt{2}/2$ (h being the distance between particles) and that the particle grid distortion is small, it leads to second order accuracy at least [39].

Finally, the evolution for the particle weights, $\boldsymbol{\alpha}_i = \boldsymbol{\omega}_i V_i$, due to viscous diffusion is given by:

$$\frac{d\boldsymbol{\alpha}_p}{dt} = \frac{2\nu}{\sigma^2} \sum_{q \in \mathcal{P}_p} (V_p \boldsymbol{\alpha}_q - V_q \boldsymbol{\alpha}_p) \eta_\sigma(\mathbf{x}_p - \mathbf{x}_q) \quad (4.20)$$

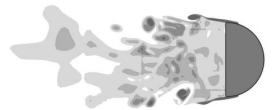
If Eq. (4.20) is summed over all the particles, the right hand side cancels due to symmetry: the PSE operator thus conserves the total circulation, as it should.

When it comes to hierarchically refined grids, one can still use Eq. (4.19) and thus Eq. (4.20): in the vicinity of level of refinement interfaces, the space quadrature will simply be more or less accurate depending on the particle level. In order to converge, the kernel radius has to be such that $\sigma/h_{\max} > \sqrt{2}/2$, where $h_{\max}$ is the maximum particle size.

Until now, we did not relate the kernel radius, σ , to the local distance between particles, h . It is however clear that one would like to adapt it to the local particle resolution, in order to also increase the accuracy on the PSE operator. This can be done by replacing σ by:

$$\sigma_{\max,p} = \beta \max(h_q) \quad \text{for } q \in \mathcal{P}_p \quad (4.21)$$

In our simulations, $\beta = 1$. Using the maximum of the neighboring particle sizes ensures that the quadrature still converges for fine particles located near an interface with a coarse level region. Well inside the fine region, this approach will lead to a fine size kernel and thus provide a higher accuracy, as expected. The drawback is that Eq. (4.20) is no longer exactly conservative along the interfaces where fine particles are involved in coarse and fine size



kernel interactions at the same time: in that case, Eq. (4.20) loses its symmetry property. This localized conservation error is however small and can be corrected by distributing the variation in total circulation over all the particles.

The same approach can be used for any other field evaluation such as the velocity field, the stream function, etc. The idea is thus to perform direct interactions for a given particle p using a constant kernel size, $\sigma_{\max,p} = \beta \max(h_q)$. The maximum is computed among the particles q whose kernel regularization actually affects particle p (typically, particles such that $|\mathbf{x}_q - \mathbf{x}_p| < 5\beta h_q$).

In order to verify the convergence properties of our approach, we investigate the error on the Laplacian and velocity field evaluations for a single Gaussian vortex:

$$\omega(\mathbf{x}) = \frac{\Gamma}{2\pi a^2} \exp\left(-\frac{|\mathbf{x}|^2}{2a^2}\right),$$

For each grid size, h_c , three grid configurations are considered:

- (i) a uniform grid with $h = h_c$;
- (ii) a hierarchically refined grid with $h = h_c$ for $x \leq h_c/2$ and $h = h_c/2$ for $x > h_c/2$, such that the first layer of coarse particles is centered on $x = 0$;
- (iii) a hierarchically refined grid with $h = h_c$ for $x \leq -h_c/4$ and $h = h_c/2$ for $x > -h_c/4$, such that the first layer of fine particles is centered on $x = 0$.

The error is then measured on the field evaluations for the subset of particles, $\mathcal{P}_e$, located at $x = 0$ and $-5 < y/a < 5$, thus along the interface for configurations (i) and (ii):

$$e_{|\nabla^2\omega|} = \left(\sum_{i \in \mathcal{P}_e} h_i \left(\nabla^2 \omega_i^h - \nabla^2 \omega(\mathbf{x}_i) \right)^2 \right)^{1/2} \quad (4.22)$$

$$\begin{aligned} e_{|\mathbf{u}|} &= \left(\sum_{i \in \mathcal{P}_e} h_i \left(u_{(1),i}^h - u_{(1)}(\mathbf{x}_i) \right)^2 \right. \\ &\quad \left. + h_i \left(u_{(2),i}^h - u_{(2)}(\mathbf{x}_i) \right)^2 \right)^{1/2} \end{aligned} \quad (4.23)$$

As shown in Fig. 4.5 the second order convergence is maintained (this is not the case if the adapted particle cross level interaction is not used).

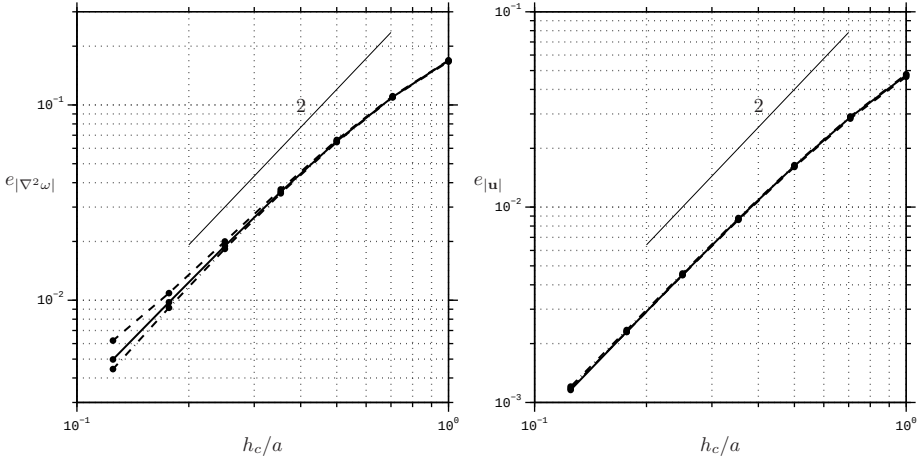


FIGURE 4.5 – Grid convergence of PSE scheme (**left**) and of velocity (**right**) for hierarchically refined grids. Error as a function of particle size is shown for three configurations: uniform grid (plain), error on coarse particles along interface for a refined grid (dash), error on fine particles along interface for a refined grid (dash-dot).

4.4 Validation

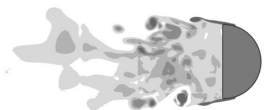
In order to validate and illustrate the redistribution procedure on hierarchically refined grids, two cases are investigated. The first one is the convection of a single Gaussian vortex: this very simple case allows to precisely investigate the behavior of the new redistribution scheme. The second test case is that of the so called “triangular vortex”: this particular well defined flow has the property of undergoing complex unsteady non-linear dynamics and producing fine scale structures. This case will essentially provide a qualitative illustration of the method for more complex flows.

4.4.1 Gaussian vortex

We define the following initial vorticity field:

$$\omega(\mathbf{x}) = \frac{\Gamma}{2\pi a^2} \exp\left(-\frac{|\mathbf{x} - \mathbf{x}_{c,0}|^2}{2a^2}\right),$$

where, $\mathbf{x}_{c,0}/a = \{-5, 0\}$. An external uniform velocity is also added to flow: $U_\infty a/\Gamma = 0.1$: this ensures that the velocity induced by the vortex itself is



significant compared to the external convection velocity. The viscosity is set to zero. Three grid configurations are investigated:

- (i) A uniform grid with $h/a = 0.2$;
- (ii) A hierarchically refined grid with $h/a = 0.2$ for $x \leq 0$ and $h/a = 0.1$ for $x > 0$. The vortex will thus be convected from the coarse grid to the fine grid;
- (iii) A hierarchically refined grid with $h/a = 0.1$ for $x \leq 0$ and $h/a = 0.2$ for $x > 0$. The vortex will thus be convected from the fine grid to the coarse grid.

For the mixed grid configurations, two simulations were performed in order to investigate the effect of the conservative correction procedure: one simulation with correction (**wc**) and one simulation without correction (**w/o c**). Five different simulations are thus carried out. The dimensionless time is defined as $T = tU/a$. The time step for all simulations is $\Delta T = 0.1$. The particles are redistributed every 5 time steps.

To illustrate a typical redistribution procedure on a hierarchically refined grid, Fig. 4.6 shows the particle locations before and after being redistributed. As can be seen, a lot of different configurations occur: particles being redistributed on the same size uniform grid, coarse particles redistributed on a uniform fine grid, particles involved in a mixed grid redistribution scheme, etc.

Figure 4.7 compares vorticity field contours for the uniform grid simulation (i) with the two simulations –with (**wc**) and without correction (**w/o c**)– carried out on grid (ii) (i.e., were the vortex is convected from the coarse grid to the fine grid region). Two distinctive times for each simulation are shown: when the vortex is entering the fine grid region and, at the end of the simulation, when the vortex is entirely inside the fine grid region. Comparing the vorticity contours for case (ii) **wc** with the uniform grid case (i), one can see that, when the vortex initially enters the fine grid region, contours are somewhat deformed along the interface. At $T = 10$, the refined simulation with correction almost recovers a perfectly axisymmetric shape. The high vorticity region is slightly larger than the uniform grid indicating that it is also slightly less diffused. On the other hand, the simulation without correction exhibits high frequency oscillations at $T = 10$ in the fine grid regions; the oscillations are reduced at the end of the simulation but still significant.

Figure 4.8 compares the uniform grid results with, this time, the results for the fine-to-coarse grid simulations. Interestingly, there is essentially no difference between the three simulations at $T = 10$. This is because, at the end of the simulation, all solutions are represented on the same grid and, of course, because the redistribution procedure accurately transfers information through the fine-coarse interface. At $T = 10$, one can see small differences between the simulation with and without correction: the solution with correction being slightly smoother there, as expected. The fact that there are no further oscillations on the coarse grid, when no correction is applied, is because those oscillations are immediately “filtered out” when they are transferred on the coarse grid.

We now turn to a more quantitative evaluation of the hierarchical redistribution procedure. We therefore define the following errors functions on the vorticity and the velocity fields, evaluated at particle locations:

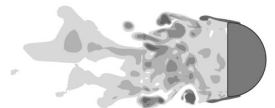
$$e_{|\omega|} = \left(\sum_{i \in \mathcal{P}_e} h_i^2 \left(\frac{\alpha_i}{h_i^2} - \omega(\mathbf{x}_i - \mathbf{x}_c) \right)^2 \right)^{1/2} \quad (4.24)$$

$$e_{|\mathbf{u}|} = \left(\sum_{i \in \mathcal{P}_e} h_i^2 \left(u_{(1),i}^h - u_{(1)}(\mathbf{x}_i - \mathbf{x}_c) \right)^2 + h_i^2 \left(u_{(2),i}^h - u_{(2)}(\mathbf{x}_i - \mathbf{x}_c) \right)^2 \right)^{1/2} \quad (4.25)$$

where, $\mathcal{P}_e$ is the subset of particles such that $|\mathbf{x}_i - \mathbf{x}_c| \leq 5a$ and $\mathbf{x}_c = (\mathbf{x}_{c,0} + U t)$ is the center of the vortex at time t . The velocity error is computed on the separate components of the velocity, in order to also take into account errors on the direction of the velocity vector.

Figure 4.9 shows the error on the vorticity field and on the velocity field as a function of time. Concerning the errors on the vorticity field, the results confirm the observations made for the vorticity contours:

- when the vortex is convected through on a hierarchically refined grid, the error on the vorticity field increases when the vortex passes through the interface and then decreases;
- for the fine-to-coarse grid case, the error is much larger when no correction is applied;
- for the coarse-to-fine grid case, the correction has little influence on the vorticity error.



In any case, at the end of the simulation, the error on the vorticity field is larger for the hierarchically refined grids than for the uniform case. This error is essentially due to high frequency “noise” in the vorticity field.

Now if we look at the error on the velocity field, we see the opposite: there is a strong decrease of the error when the vortex enters the fine domain. Furthermore, the velocity error for the case without correction is the same as for the case with correction. This might seem odd at a first sight, but one has to remember that the velocity is an integral field with respect to the vorticity (Biot-Savart): the high frequency oscillation are “smoothed” by the integral operator. Looking at the error on the vorticity field is thus much more “challenging”.

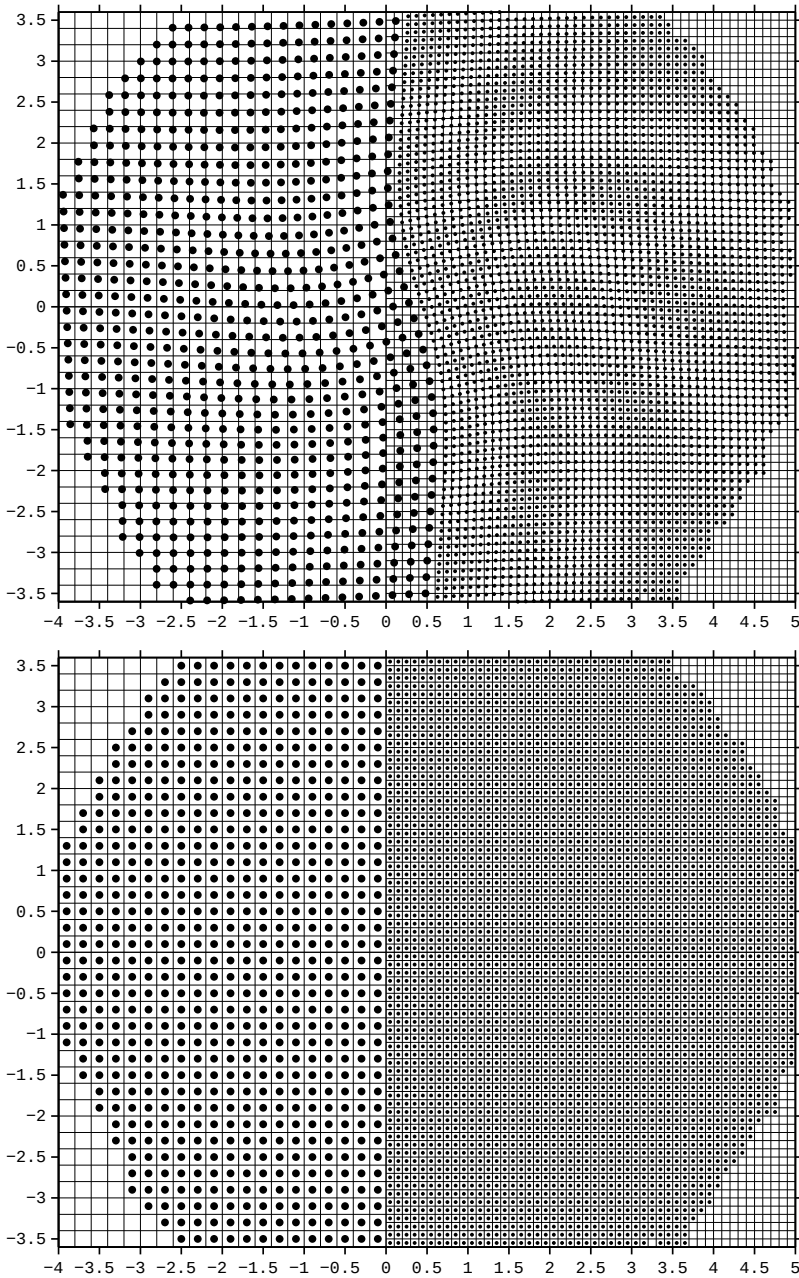
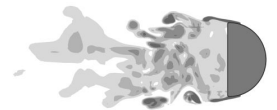


FIGURE 4.6 – Convection of a gaussian vortex: grid and particle locations for configuration (ii) **wc** at $T = 5.5$. **Top:** before redistribution; **Bottom:** after redistribution.



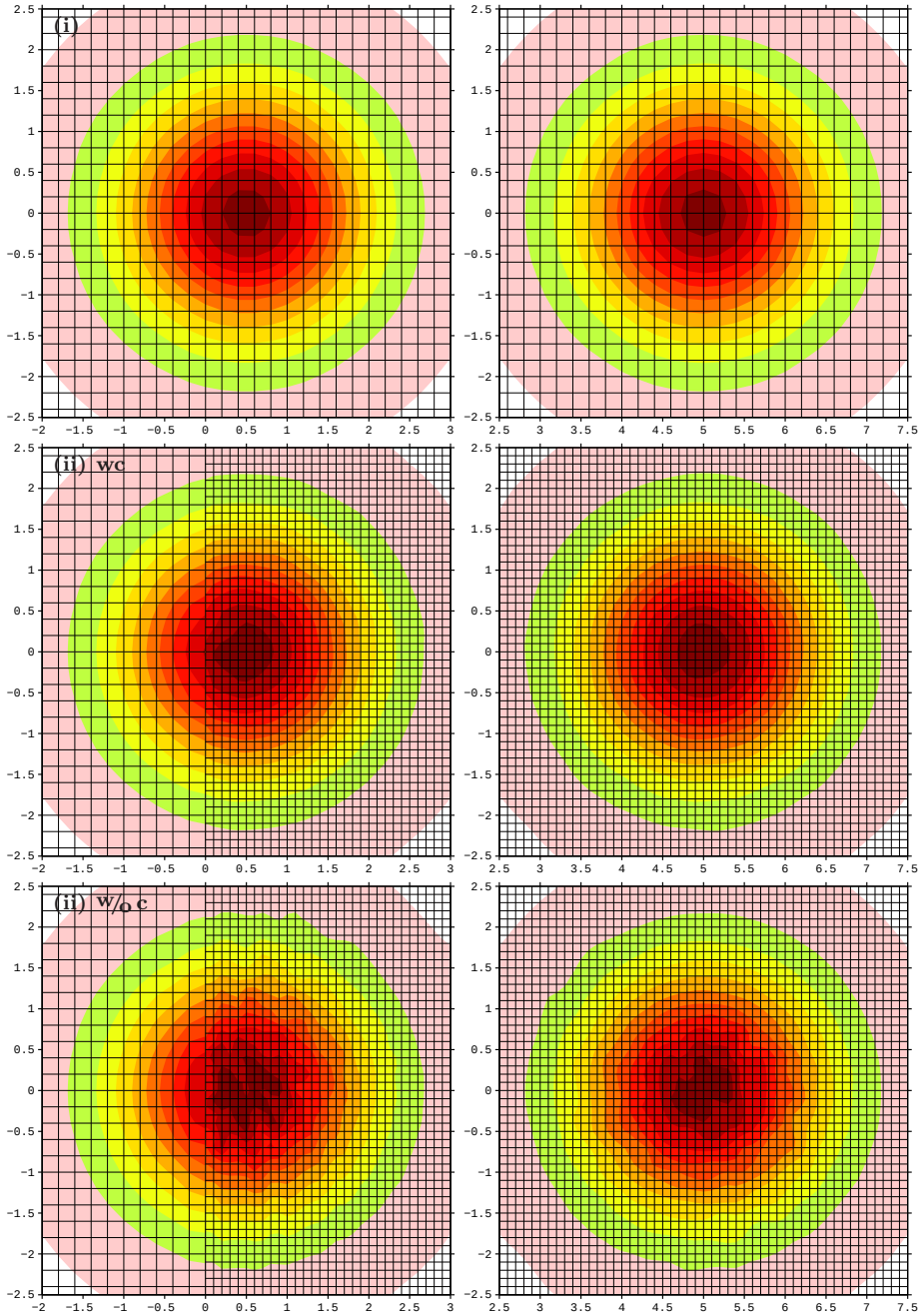


FIGURE 4.7 – Convection of a gaussian vortex: contours of vorticity at $T = 5.5$ (left) and $T = 10$ (right). **Top:** uniform grid; **Middle:** hierarchically refined grid with correction; **Bottom:** hierarchically refined grid without correction.

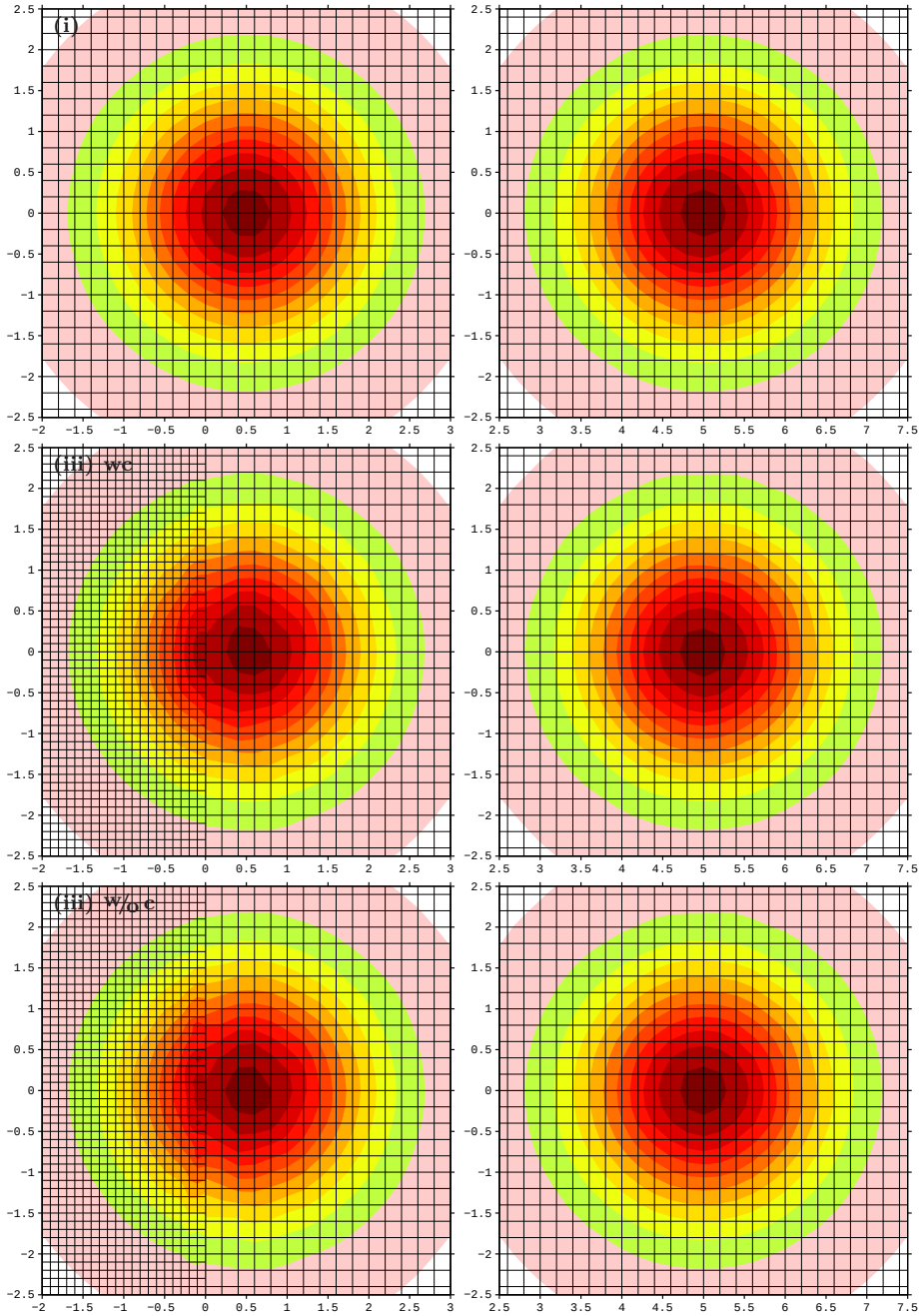
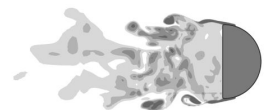


FIGURE 4.8 – Convection of a gaussian vortex: contours of vorticity at $T = 5.5$ (left) and $T = 10$ (right). **Top:** uniform grid; **Middle:** hierarchically refined grid with correction; **Bottom:** hierarchically refined grid without correction.



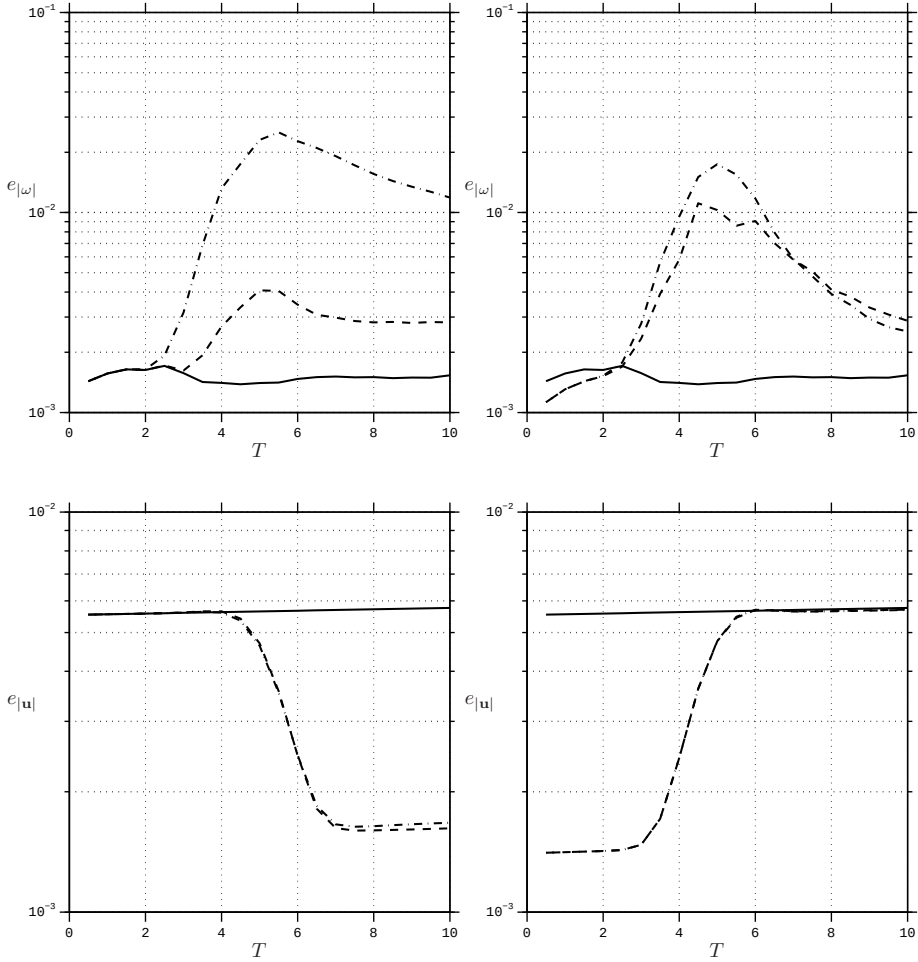


FIGURE 4.9 – Convection of a gaussian vortex: error as a function of time, T , on the vorticity field, $e_{|\omega|}$, (**Top**) and on the velocity field, $e_{|u|}$, (**Bottom**). Each graph shows the error for the uniform grid (plain), the hierarchically refined grid with correction (dash) and without correction (dash-dot). **Left:** convection from coarse to fine grid. **Right:** convection from fine to coarse grid.

4.4.2 Triangular vortex

In this section, we undertake the validation of the hierarchical redistribution procedure for a complex 2-D flow. The case is that of a circular vortex patch having a zero net circulation. The dimensionless velocity and vorticity profiles are respectively given by [9, 54]:

$$\begin{aligned}\frac{u_\theta(\rho)}{\Omega_0 a} &= -\frac{1}{2} \rho \exp(-\rho^\alpha), \\ \frac{\omega(\rho)}{\Omega_0} &= \left(\frac{1}{2} \alpha \rho^\alpha - 1 \right) \exp(-\rho^\alpha),\end{aligned}$$

where, Ω_0 is the core center vorticity, a is the vortex core size (more precisely, the inflection point in the velocity profile) and $\rho = |\mathbf{x}|/a$. These profiles define an axisymmetric flow with an essentially constant negative vorticity core surrounded by an annular region of positive vorticity. As in [38], we will study the $\alpha = 7$ case. At this parameter value, the vortex is unstable to 2-D perturbations with azimuthal wavenumbers of $k_\theta = 2, 3$ and 4. An initial perturbation is imposed to trigger the most unstable mode, $k_\theta = 3$, by modifying the initial vorticity field according to:

$$\omega^*(\mathbf{x}) = \omega(|\mathbf{x}^*|/a), \quad (4.26)$$

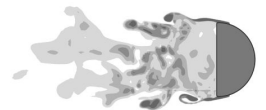
$$\mathbf{x}^*(\mathbf{x}) = \mathbf{x} (1 + \epsilon \sin(3\theta(\mathbf{x}))), \quad (4.27)$$

where, θ is the angular position and $\epsilon = 10^{-2}$ is the perturbation amplitude. The Reynolds number is defined as $Re = \frac{\Omega_0 a^2}{\nu}$ and was set to $Re = 10^5$. The dimensionless time is defined as $T = t\Omega_0$.

During the initial phase, the imposed perturbation grows and deforms the annular vorticity region. Vorticity lobes are then formed and they eventually lead to a new equilibrium state in the form of a triangular vortex patch: three positive vorticity regions rotating around a central negative vortex core. During the transitional phase, strong vortex filamentations occur, producing small scale dynamics.

In order to demonstrate the capabilities of the hierarchical redistribution method, two grid configurations are investigated:

- (i) a uniform grid with $h/a = 10^{-2}$, this grid is fine enough to capture the dynamics of the flow and will be used as reference.
- (ii) a hierarchically refined grid with $h/a = 10^{-2}$ and refined at $h/a = 5 \cdot 10^{-3}$ in the upper right quadrant (i.e., for $x > 0$ and $y > 0$).



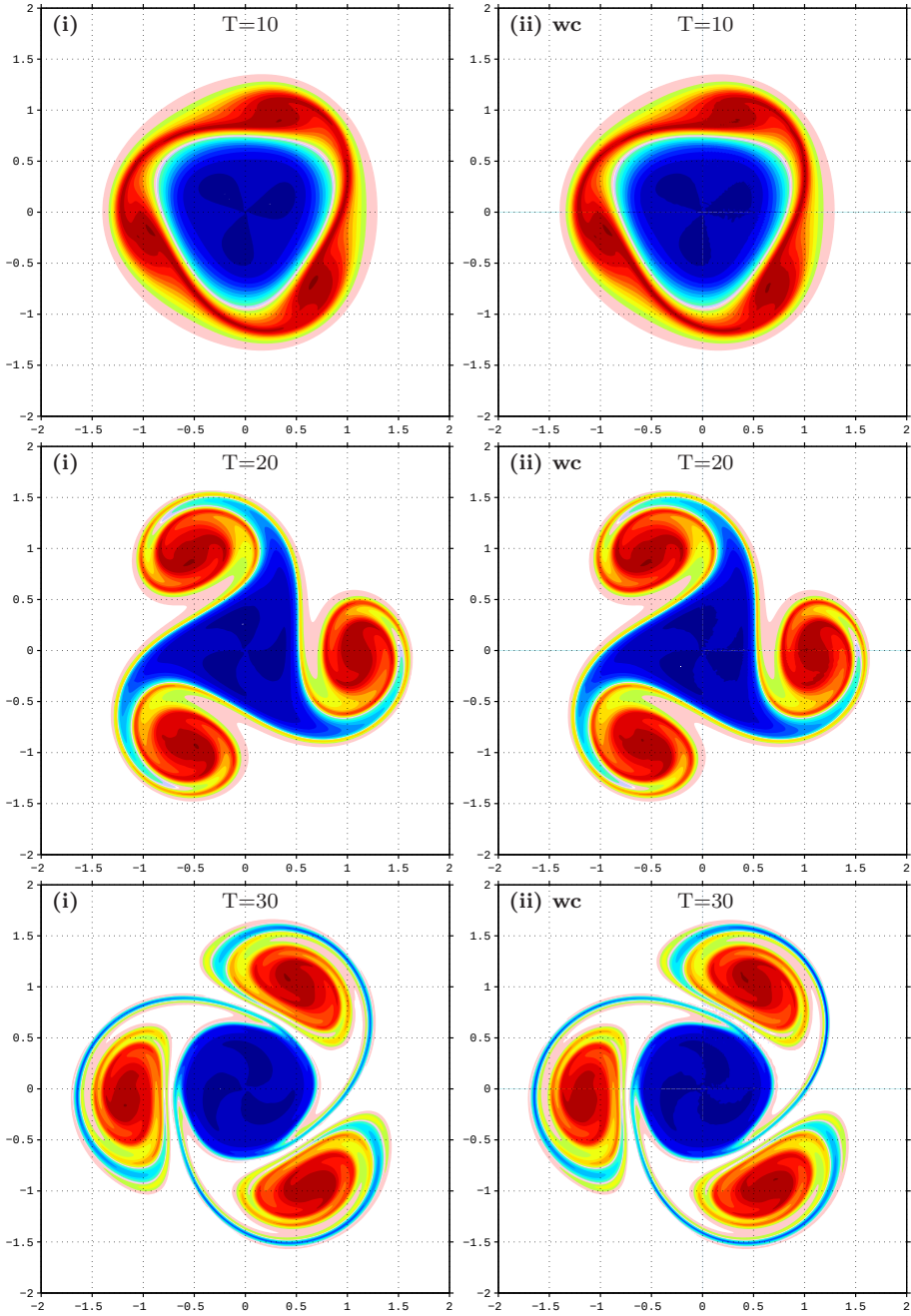


FIGURE 4.10 – Triangular vortex: contours of vorticity at $T = [10, 20, 30]$. **Left:** uniform grid; **Right:** refined grid with adjusted scheme.

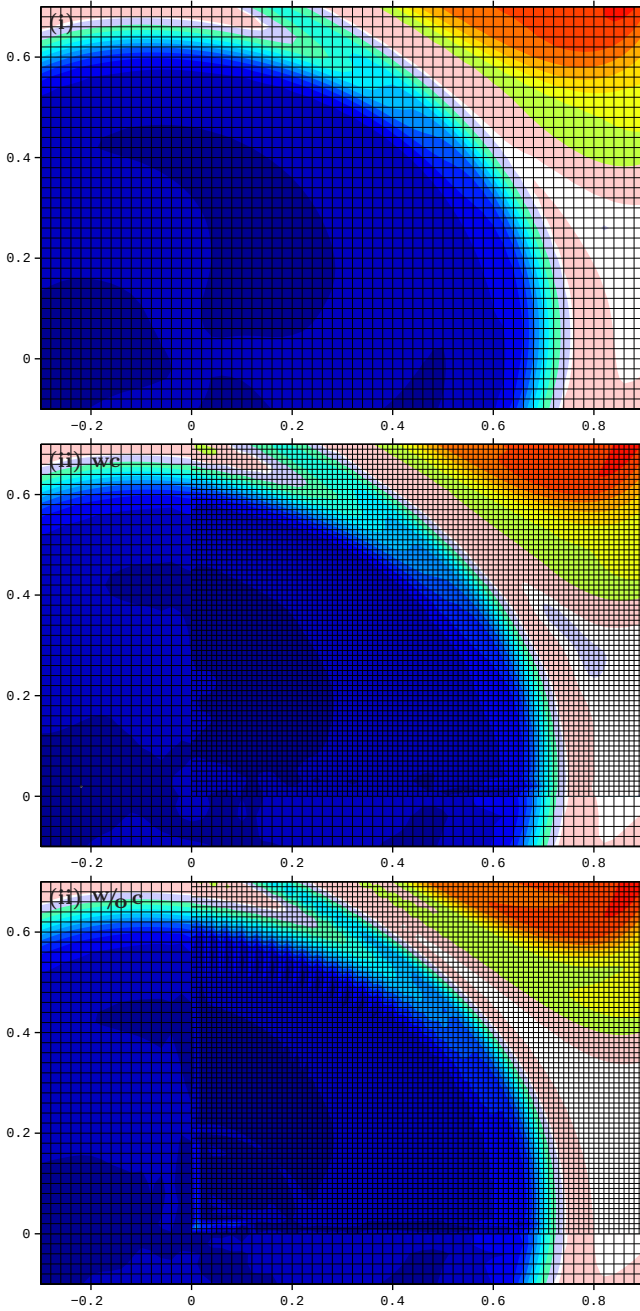
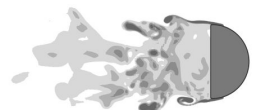


FIGURE 4.11 – Triangular vortex: contours of vorticity at $T = 30$. **Top:** uniform grid; **Middle:** refined grid with adjusted scheme; **Bottom:** refined grid without adjusted scheme.



The choice of this particular hierarchically refined grid is challenging for the method as vorticity structures are periodically convected through the grid interfaces: from the coarse to the fine region and later from the fine to the coarse region again. During the simulation, the vorticity field will undergo a large number of global revolutions, thus passing several times through the refined region. Moreover, the center of the computational domain is a corner of the refined region: it is interesting to observe the behavior of the method in this configuration, particularly in a region where the velocity tends to zero and the same vortex particles are thus continuously involved in mixed grid schemes. The time step for the simulation was set to $\Delta T = 0.1$. Because of the high shear in the annular region, the particle grid is distorted rapidly: the particles are therefore redistributed every time step.

Figure 4.10 shows vorticity contours at different times for the uniform grid and for the hierarchically refined grid (with correction) side by side. As can be seen from those results, the vorticity fields for both grids remain identical even for long simulation times and despite the complex filamentation events. Only a very close look at the results on the hierarchically refined grid reveals very small oscillations in the vorticity field. It also shows that the vortex filaments are slightly better captured in the fine region.

Figure 4.11 takes a closer look at the vorticity field at $T = 30$ for three additional simulations that were performed on grids twice as coarse as the previous cases. For the hierarchically refined grid, the case with correction and the case without correction are both considered. The vorticity field for the corrected case has clearly less oscillations than that for the case without correction. The behavior at the corner of the refined region is also improved. However, despite the larger oscillations when no correction is applied, the “macroscopic” vorticity field remains almost identical to that for the two other cases. For this reason, we did not show the case without correction on the fine grid: at the global level shown in Fig. 4.10, almost no differences could be seen between the cases with and without correction. This result confirms what was already observed in the validation on the convection of a Gaussian vortex: the oscillations that are introduced at grid interfaces when no correction procedure is applied does not introduce a significant error on the velocity field. The vorticity field is convected as it should but with some spurious oscillations.

4.5 Summary

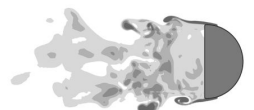
In this chapter, we developed a new conservative redistribution scheme for hierarchically refined grids. Our approach allows to build arbitrarily refined redistribution grids providing a much greater flexibility than classical global mapping approaches. Compared to domain decomposition approaches, it has the advantage of being much simpler: there is no need to modify the global vortex method algorithm.

The redistribution scheme itself is consistent: a general scheme has been defined such that it falls back to the classical M'_4 scheme when applied to a particle being redistributed on a uniform grid of equal size. A correction procedure was successfully developed to eliminate spurious oscillations that appear at the interfaces between different levels of refinement. The scheme is also general in the sense that it can be trivially extended to 2-D and 3-D. Hence, we could implement the hierarchical redistribution procedure in a general way such that the same functions can be used for any dimensions.

The issue of cross level particle interactions was addressed: when it comes to field evaluations, some care has to be taken for the particle interactions near the interfaces between grid levels. It was verified that the vortex method can maintain a second order accuracy as long as the kernel size, σ , used for the close interactions, is set with respect to the local maximum particle size.

From the validation cases, we can conclude that the conservative hierarchical redistribution scheme provides very satisfactory results. The method was first validated on the case of the inviscid convection of a single vortex. Our grid refinement approach leads to a decreased error on the velocity field. However, the error on the vorticity is slightly increased due to some high frequency noise. The amplitude of these oscillations are strongly reduced when the correction procedure is applied. The fact that the correction procedure does not influence the error on the velocity field might lead to conclude that the correction procedure has no effect. It allows nevertheless to obtain smooth vorticity fields which could be critical in 3-D where the spurious oscillations could introduce a vorticity divergence error or non physical instabilities and flow transition.

The second validation case was the simulation of a complex 2-D flow: a circular vortex patch that becomes unstable and produces filamentation and fine scale structures during a complex unsteady phase. The results showed an almost perfect agreement between the uniform grid case and the hierarchically refined grid case. When no correction is applied, spurious oscillations appear



on the vorticity field. They however do not affect the global dynamics of the flow (this could be expected from the first validation cases where it was shown that the velocity field is not affected by those spurious oscillations).

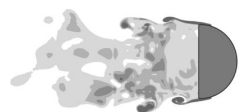
Despite the fact that there is no direct limitation on the hierarchical redistribution scheme, we still think that the refinement level difference at an interface should not be greater than one (i.e., the particle size must be at most divided by two through an interface). Another constraint would be that the interface shape should maintain some degree of regularity: the interface should be straight over sufficient distances. We did however not precisely investigate the effects of those factors.

CHAPTER 5

Flow past a hemisphere using a hierarchically refined grid

5.1 Introduction

The capabilities of the redistribution method on hierarchically refined grids are demonstrated on the flow past a hemisphere at $Re = 3,000$. At this Reynolds number, the boundary layer at separation is still laminar, and the transition to turbulence occurs further downstream in the separated shear layers via Kelvin-Helmholtz instabilities. A sufficiently fine grid is thus required to accurately capture the shear layers emanating from the hemisphere. However, inside the recirculation bubble and further downstream, where the flow has become fully turbulent, the grid requirements are less demanding. It is therefore a typical case where the ability to control the level of grid refinement is essential. As we will see, the use of the hierarchical grid refinement also allows for a reasonable, but marginal, grid size in the boundary layers at the investigated Reynolds number. The limitation comes once again from the fact that the computational elements are isotropic. A factor two grid size refinement in the wall normal



direction requires eight times more grid points in the refined region, while anisotropic refinement would require only two times more elements.

The marginal grid resolution in the boundary layer again oriented our choice towards the flow past the hemisphere. In this case the separation location is determined by the geometry, unlike the sphere, where the separation is closely related to the evolution of the boundary layer. On the other hand, except for some approximate drag coefficient values found in engineering text books, there are very few detailed studies for the flow past the hemisphere (neither experimental, nor numerical), while the flow past the sphere is well documented in the literature. Recently however, the laminar flow past the hemisphere was investigated numerically by Kim and Choi [51] up to $Re = 300$. They showed that there exist significant differences between the unsteady flow past a hemisphere and a sphere: (i) in the case of time-periodic planar-symmetric flow, alternating vortex shedding takes place behind a hemisphere with zero lift force, while nonzero mean lift occurs for the flow over a sphere; (ii) there exists a time-periodic asymmetric flow regime for the hemisphere (not for the sphere); (iii) the onset of axisymmetry breaking and unsteadiness takes place at lower Reynolds numbers than those for the flow past a sphere. The practical use of those results to the present simulation is however limited due to the very different Reynolds numbers (ours being $Re = 3,000$).

Because of the lack of experimental data in the literature, a study of the flow past the hemisphere was performed in our experimental facilities by Willemart [97]. Quantitative results were obtained in a low speed, open-section, wind tunnel. Laser Doppler Velocimetry (LDV) was used to obtain mean velocity profiles and turbulence profiles at $Re = 55,000$. The drag coefficient was measured by Cottin (private communication) using a force balance at $Re = 55,000$ and $Re = 110,000$. Qualitative results were also obtained [97]: the flow was visualized in a towing tank facility at Reynolds numbers ranging from $Re = 1,000$ up to $Re = 20,000$. The experimental results are qualitatively and quantitatively compared to those obtained from our numerical simulation.

5.2 Computational approach

We here use the “classical” vortex and boundary element method, in the sense that the entire vorticity field is discretized using vortex particles (i.e., we are not using attached boundary layer regions, as was done in Section 2.4): the

boundary element method is used to compute the vorticity flux at the wall, such that the no-slip boundary condition is enforced. However, since we are using the hierarchical redistribution scheme, some stages of the method are modified.

In Section 4.3, we already described how the evaluation of fields using discrete integral operators has to be modified when particles from different levels are involved. We will now see how the hierarchical redistribution procedure can be used to facilitate accurate particle redistribution near boundaries. Furthermore, the cartesian particle representation after the redistribution step can be put to profit to correct the vorticity divergence. Finally, we here used a LES subgrid scale model that has the proper near-wall behavior and is essentially inactive in laminar shear-layers.

5.2.1 Redistribution near boundaries

During the redistribution procedure, the distorted particle set is replaced by a new set of particles, which are again located on a regular grid. If boundaries are present, the redistribution of particles close to the wall must be modified, because particles are not allowed inside the body.

The method developed in [77, 78] achieves this by using decentered interpolation kernels. When considering a 1-D problem, those kernels are such that they redistribute the particle strength on a “decentered” set of particles (away from the wall) such that they do not create particles inside the body, while still conserving the desired moments of the vorticity distribution. In 2-D and 3-D, the problem is more complex because, in general, the boundary crosses the redistribution scheme in an arbitrary way. In [77, 78] and in the remainder of the present work, the method consisted in successively redistributing the old particle in each direction, and using centered schemes whenever possible (that is, if it does not create particles inside the body); if not, a decentered scheme is used. The result of this redistribution process is however dependent on the sequence of directions along which the particles were redistributed (i.e., in 2-D one can either redistribute first along the x direction or along the y direction). The sequence of directions to be used is the one that leads to the lowest global penalty which is computed based on the number of decentered schemes used for each sequence. For more details on the method, refer to [77].

The tools developed for the redistribution on hierarchical grids allows for a more elegant approach. Whenever an old particle involves target particles



that are located inside the body, they are simply discarded. The weight of the remaining particles is estimated, as usual, using a centered scheme. Finally, as for the redistribution involving mixed level target particles, the target weights are adjusted such that the required moments are conserved. Since the adjusted weights are computed in order to satisfy the conservation properties while minimizing the difference with the estimated target weights, the global scheme will provide a good continuity near the wall (there is no abrupt switch to a different redistribution scheme for particles located close to the wall). It is also very easy to implement in the general redistribution scheme procedure on hierarchical grids. The modified procedure described at the end of Section 4.2.2 is now as follows (the differences are written in *italic*):

- Determine old particle target level, $l^\bullet$;
- Determine $\mathcal{P}$, the subset of particle affected by the redistribution of the old particle *and located outside the body*;
- Compute β_i the estimated target weights using Eq. (4.5);
- **If** all target particles are at the same level **and** *no target particles are located inside the body*, **then** $\alpha_i = \beta_i$ (the estimated target weights do not need to be adjusted),
else compute the adjusted weights, α_i , using Eqs. (4.12) to (4.14).

This procedure can still be used as is, even if mixed target particle levels are involved near the wall. Finally, when the old particle is located very close to the wall ($d_{\text{wall}} < 0.5h$), the order of the scheme is reduced to $m = 1$ (instead of $m = 2$): if one tries to conserve higher order moments in this configuration, spurious oscillation appear due the decentered location of the old particle with respect to the new target particles.

5.2.2 Vorticity divergence correction

The problem of the vorticity field divergence has already been addressed several times throughout this work; however it is has not yet been solved in a satisfactory way. In Section 2.3, the problem was circumvented through the use of a low order, dissipative, redistribution scheme that prevented the development of small scale structures and thus the divergence problem. In Section 2.4, a “relaxation scheme” was introduced: the particle strengths are set back to the curl of the velocity field computed at particle locations (this field is, by definition, divergence free). This relaxation procedure is, again, a diffusive operation.

When it is applied sufficiently scarcely, the global diffusive effect remains fairly low. However, the control on this procedure is very limited. Finally, besides the lack of control, both approaches have the drawback that they are equally active everywhere in the computational domain: even in locations where the vorticity divergence is not an issue, an equal amount of diffusion will be introduced.

A good way to maintain the divergence free property of the vorticity field consists in regularly re-projecting the vorticity field onto a divergence free basis. This can be done exactly in the same way the incompressibility constraint is enforced on the velocity field in a fractional-step method:

$$\omega^{\text{new}} = \omega - \nabla \Phi, \quad (5.1)$$

where Φ is obtained by solving the following Poisson equation:

$$\begin{cases} \nabla^2 \Phi &= \nabla \cdot \omega, \\ \frac{\partial \Phi}{\partial n} &= -\omega \cdot \mathbf{n}, \quad \text{at the wall} \end{cases} \quad (5.2)$$

where the Neumann boundary condition ensures that the normal component of the vorticity vanishes at the wall.

This method is used by Cottet and Poncet [80, 23, 24] in a Vortex-in-Cell (VIC) approach. They argue that the procedure is possibly dissipative and is therefore only used to enforce a divergence free initial condition. It seems that in their computations, they do not require any further control of the vorticity divergence. This is probably because their computations are relatively well resolved DNS of moderate Reynolds number flows.

Cocle *et al* [15, 14, 99] have successfully applied this procedure to the simulation of high Reynolds number flows using LES (also in a VIC approach). They have shown that it does not introduce any significant additional diffusion: their results compared very well with those obtained for the same configuration using an energy conserving spectral code.

In both approaches, the differential operators in Eqs (5.1) and (5.2) are computed on the VIC grid using finite differences. Equation (5.2) is solved using an efficient grid based Poisson solver. We here transposed the approach to the Lagrangian vortex element method. We take advantage of the fact that, after the hierarchical redistribution procedure, the particles are located on a regular cartesian grid: the divergence operator in Eq. (5.2) can be evaluated using a finite volume approach. The main difference comes from the resolution of the Poisson problem: this has to be done using a Green's function approach,



as for the velocity and streamfunction evaluation from the vorticity field. Furthermore, the boundary condition on Φ is enforced through a boundary element method. The fast multipole solver and the boundary element solver had to be adapted to allow for a scalar source term (instead of a vector valued source term); this is obviously quite straightforward.

The vorticity divergence in a Lagrangian vortex element implementation can thus be summarized as follows:

- After redistribution, evaluate the vorticity divergence at redistributed particle locations through a finite volume operator. In practice, the particle neighbor cells are obtained using the hierarchical grid structure.
- Evaluate ω at the wall: it is obtained from the regularized vorticity field representation.
- Solve the boundary element problem such that $\frac{\partial \Phi}{\partial \mathbf{n}} = -\omega \cdot \mathbf{n}$ at the wall. This is done exactly in the same way as the vortex sheet is computed in order to cancel the slip velocity at the wall (we also use the same body surface discretization).
- $\nabla \Phi$ can now be evaluated at the redistributed particle locations using a Green's function approach by adding the contribution of the volume source terms (i.e., the particles) and the boundary contribution (i.e., the boundary element panels).
- Correct the particle strengths using Eq. (5.1).

Notice that this approach could also be applied when using “classical” mapped redistribution grids, provided that the finite volume operator takes the redistribution grid metrics into account.

It is clear that this approach does not enforce the vorticity field to be exactly divergence free at the discrete level (in the sense of the finite volume divergence operator). This is because the Laplacian operator is not build from the combination of the effective discrete divergence and discrete gradient operators: $\nabla_h \cdot (\nabla_h \Phi)$. The same is true for the divergence correction procedure in the VIC method. However, even if this would be a desirable property, the approximate vorticity divergence correction is sufficient to maintain the divergence of the vorticity field at an acceptable level for long time simulations.

5.2.3 LES model

We here use an effective viscosity based subgrid-scale model to take into account the turbulent dissipation that occurs at small scales. In Section 2.4 the classical Smagorinsky model was used. However, without dynamic procedure to adjust the model coefficient, the Smagorinsky model does not have the proper near wall behavior: it is too dissipative in the laminar sublayer region (and more generally in laminar shear layers).

The Wall-Adapting Local Eddy-viscosity (WALE) model of Nicoud and Ducros [71] represents a good alternative to the Smagorinsky model: it has the correct near wall behavior without the need for additional damping functions or a dynamic procedure. As a consequence of this property, the model is also inactive in laminar shear layers. The model is therefore well suited to capture transitional flows such as the one studied here. Finally, the WALE model is also very easy to implement as it is based on local quantities only. The properties of the WALE model will be further investigated in Chapter 7.

The WALE model eddy-viscosity is given by:

$$\nu_t = (C_w \Delta)^2 \frac{(\mathcal{S}_{ij}^d \mathcal{S}_{ij}^d)^{3/2}}{(\bar{S}_{ij} \bar{S}_{ij})^{5/2} + (\mathcal{S}_{ij}^d \mathcal{S}_{ij}^d)^{5/4}} \quad (5.3)$$

where $\bar{S}_{ij}$ is the deformation rate tensor for the resolved field. $\mathcal{S}_{ij}^d$ is defined as follows:

$$\mathcal{S}_{ij}^d = \frac{1}{2} (\bar{g}_{ij}^2 - \bar{g}_{ji}^2) - \frac{1}{3} \delta_{ij} \bar{g}_{kk}^2, \quad (5.4)$$

with $\bar{g}_{ij}^2 = \frac{\partial \bar{u}_i}{\partial x_k} \frac{\partial \bar{u}_k}{\partial x_j}$ and δ_{ij} the Kronecker symbol. The model constant C_w is here set to 0.5; this value is given by Nicoud in [71] and was calibrated numerically on isotropic decaying turbulence. In our computations the subgrid characteristic length scale Δ is set to h to local particle size.

5.3 Numerical setup

The flow past the hemisphere is investigated at $Re = U_\infty D/\nu = 3,000$. An outflow boundary condition is imposed at $x/D = 12.5$, using the same approach as in Section 2.4. As we investigate a relatively high Reynolds number case, this distance is considered to be sufficient since the wake becomes rapidly turbulent. At lower Reynolds numbers (e.g., $Re = 300$) larger domains are required due to the fact that the flow is dictated by large coherent structures.



The hierarchical redistribution grid is built to enable a resolution as good as possible of the boundary layers and the separated shear layers. The size of the redistribution is decreased inside the separation bubble. It is further decreased downstream of the hemisphere. Five levels of refinement are used. Slices through the 3-D hierarchical redistribution grid are shown in Figs 5.2 and 5.3. At the coarsest level for $x/D > 6.0$, the grid size $h/D = 0.1$. The grid is then gradually refined. The finest level is defined as the annular region where the shear layers initially separate from the hemisphere, and the hemispherical “shell” to capture the boundary layers. The grid size at the finest level is $h/D = 6.25 \cdot 10^{-3}$. This is a decent grid size for the separated shear layers (especially if one wants to perform LES). For the boundary layers however, the wall normal grid size is quite poor. The number of grid points for the “3-D equivalent” region to the one shown in Fig. 5.2 is about $8 \cdot 10^6$. The total number of particles during the turbulent regime is about $4 \cdot 10^6$. It should be noticed that roughly 70% of the particles are present at the finest level, inside the shear layers and the boundary layer regions. This number is a good indicator of the efficiency of the method: we are able to locate a significant fraction of the computational elements in a small region, exactly where they are required. If one would use a constant resolution in the near wake region the number of particles would be dramatically increased: remember that a factor two in the grid size (i.e., a one level refinement) requires eight times more particles.

The sharp edge of the hemisphere is regularized such that it can be sufficiently well captured (Fig. 5.2). It is replaced by a round edge of radius $r_{\text{edge}}/D = 1.75 \cdot 10^{-2}$. To do this, the hemisphere itself is not modified but the back face of the body is moved downstream by r_{edge} and connected to the hemisphere by a quarter torus section. The regularized hemisphere is discretized using 64,000 panels. The panel size is similar to the particle size in the boundary layer. At redistribution, particles at distance smaller than $d_{\text{halo}} = 1.5 \cdot 10^{-3}$ to the body surface are considered as being inside: this reduces the probability for particles to cross the boundary in between two redistribution steps.

The time step for the simulation is $\Delta t U_{\infty}/D = 5 \cdot 10^{-3}$. The time step is essentially determined by the quality constraint based on the maximum vorticity, $|\boldsymbol{\omega}|_{\text{max}} \Delta t = \mathcal{O}(1)$: this ensures that the particles do not rotate too much with respect to each other during one time step. Obviously, this constraint is the most restrictive in the boundary layer region. Notice that Constantinescu and Squires, [16, 17] and Georges *et al.* [40] both used $\Delta t U_{\infty}/D = 4 \cdot 10^{-2}$ for the LES of the flow past a sphere at $Re = 10,000$: both authors use codes that

were initially developed for RANS and URANS computations, and that were adapted for LES by introducing non-dissipative, energy conserving, schemes. Those codes have the advantage of being fully implicit (for both the diffusion and convection time integration): an essential ingredient when it comes to capturing efficiently high Reynolds number attached boundary layer regions.

To avoid a too high growth rate for the number of particles, new particles having a strength $|\alpha| < 10^{-4}|\alpha|_{\max}$ and $Re_h = |\omega|h^2/\nu < Re_{h,\text{trsh}}$ (where $|\omega|h^2 = |\alpha|/h$) are deleted. Here, a value of $Re_{h,\text{trsh}} = 10^{-3}$ is used. The latter criterion is based on the fact that particles for which the “mesh” Reynolds number is very low can be deleted.

The computational time required to reach the turbulent regime is reduced by initializing the simulation using a well developed coarse grid solution. This is another advantage of the hierarchical redistribution procedure: fine grid (or coarse grid) solutions can easily and consistently be obtained from a given solution on a different grid. Once the turbulent regime was reached on the fine grid, the simulation was run for $tU_\infty/D = 25$ (5,000 time steps) in order to obtain statistically converged quantities such as the mean velocity, turbulent fluctuations, etc. The simulation was run for a total of 7,500 time steps on 40 CPU’s (AMD Opteron at 2.6GHz) and required about 280 hours (wall clock time).

In order to visualize the flow, a passive scalar tracer is injected along a circle at the surface of the hemisphere (at $x/D = -0.15$). The passive scalar is transported by the particles and is redistributed in the same way the vorticity is. The molecular diffusion of the passive tracer is set to zero in order to simulate a high Schmidt number configuration (this is the case for fluorescein in water, used for the experimental water tank visualizations).

5.4 Results

5.4.1 Time evolution

Figure 5.4 shows the instantaneous vorticity field in the $x - y$ plane ($z = 0$) at three consecutive times during the simulation. The time, $T = tU_\infty/D$, is defined with respect to an arbitrary time when the flow has reached its turbulent regime. One can see the intense boundary layer that separates along the edge of the hemisphere. The separated shear layer become rapidly unstable through



Kelvin-Helmoltz (KH) instabilities. At $T = 0.5$, the instabilities are clearly visible in the upper shear layer. During the course of the simulation however, those instabilities are not always present. When looking at the animated time frames, one can see that the instabilities rather appear periodically in a burst like manner. In fact, it appears that the shear layer is also strongly perturbed in the azimuthal direction. There seems to be a competition between the azimuthal and longitudinal KH instabilities: when the azimuthal perturbations are bigger, the shear layer becomes fully three-dimensional before the KH instabilities have had enough time to develop. Once the shear layers have become unstable, they rapidly become fully turbulent and their intensity decreases rapidly. The vorticity is therefore much lower in the recirculation region and the turbulent wake.

The KH instabilities are even better identified when examining at the corresponding instantaneous tracer fields, shown in Fig. 5.5. At $T = 0$, two KH vortices can already be clearly identified, while they were barely visible in the vorticity plot. At $T = 0.5$, those two vortices have been convected downstream: they are identified by the two high tracer concentration spots, while a third vortex is forming on the separated shear-layer. It is interesting to see that the passive tracer field gives a quite different image of the flow. It is due to the Schmidt number: a high Schmidt number will result in very small structures for the transported scalar field while the underlying fluid dynamics are much smoother. In our computation, the Schmidt number is infinite. However, the ability to capture small structures in the tracer field is still limited by the local grid resolution. Those very fine structures can be seen in the experimental visualization (Fig. 5.6). But, if one would measure the corresponding experimental vorticity field (using Particle Image Velocimetry, for example) one would probably obtain a picture similar to Fig. 5.4.

The colormap for the tracer field plots was chosen to match as closely as possible the images obtained in a towing tank using Laser Induced Fluorescence (LIF) visualization, as shown in Fig. 5.6. From a qualitative point of view, if one compares the experimental visualization with the passive tracer visualization, the resemblance is striking: especially when comparing it to the numerical tracer field at $T = 0.5$. The distance between two successive KH vortices is very similar (about $0.6D$ in both cases). The distance at which they form behind the hemisphere is also in good qualitative agreement.

In order to get a more quantitative insight into the unsteady dynamics, the time signal of the velocity was gathered during the course of the simulation at

different locations in the wake. Figure 5.7, shows the velocity components as a function of time in three of those locations (see also Fig. 5.4 for their relative position with respect to the hemisphere and the wake). The spectrum of the radial velocity component (i.e., u_y in this configuration) was also computed for each location and is shown in Fig. 5.8. The velocity spectrum is defined as follows:

$$\hat{\mathbf{u}}(f) = \frac{1}{(t_1 - t_0)} \int_{t_0}^{t_1} \mathbf{u}(t) \exp(-i 2\pi f t) dt \quad (5.5)$$

The first probing location is situated in the early separated shear layer, where the KH instability has not saturated yet. The radial and azimuthal velocity fluctuations are still low there. The radial velocity spectrum has a clear maximum at the frequency $St = f D/U_\infty = 1.2$; this frequency thus corresponds to the KH instability. In the second position, located further downstream, the velocity spectrum is broadened, corresponding to the transition to turbulence. At this location, the high frequency KH peak is still present but lower frequencies appear. The third location is located along the axis of the hemisphere in the fully turbulent wake. The velocity spectrum there has a very well defined spike at $St = 0.2$.

The observed low frequency value, $St = 0.2$, is in very good agreement with the so-called “main shedding frequency” measured experimentally for the flow past a sphere in the subcritical regime [1, 4, 83], as well as with numerical results obtained by several authors [16, 17, 40]. The high frequency mode, associated with the KH instability, is also observed experimentally and numerically for the flow past the sphere. At $Re = 10,000$, the high-frequency mode is reported to be in the range $St \approx 2.0 - 2.5$ [86, 16, 17]. The corresponding Strouhal number however increases with the Reynolds number because the shear layers become thinner and become unstable at smaller wavelengths. One recalls that, for a time-developing shear layer with velocity profile $u = U \tanh(y/\delta)$, the unstable modes are those for which $0 \leq k\delta \leq 1$ and the most unstable one is $k\delta = 0.44$ (i.e., $\lambda = 14.2\delta$); λ is thus directly proportional to δ .

5.4.2 Averaged fields

In this Section, the average velocity fields and velocity fluctuation fields are investigated. They are obtained through time averaging of the instantaneous velocity fields. In addition, taking advantage of the axisymmetry of flow, spatial averaging is also performed. This allows to obtain well converged averages. The



present numerical results are compared to mean velocity and fluctuation profiles obtained using Laser Doppler Anemometry for the flow past the hemisphere at $Re = 55,000$.

We first consider the streamlines of the mean flow (Fig. 5.9). The recirculation bubble extends up to $x/D = 1.47$ in the streamwise direction and $r/D = 0.57$ in the radial direction. Figure 5.21 also shows a comparisons between the experimental and the numerical streamlines. The experimental results have however to be taken with some care as the streamlines were reconstructed from seven mean velocity profiles only (at $x/D = [0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 4.0]$). The streamfunction in between those profiles was obtained using cubic spline interpolation, while upstream of $x/D = 0.5$, the streamfunction was even rebuilt using extrapolation. The overall shape of the recirculation bubble should nevertheless be correct. The length of the recirculation bubble in the experiment is $x/D = 1.6$ and the radial extent is $r/D = 0.6$ which is in good agreement with the simulation (considering that the Reynolds number is different): it is also known that the length of the recirculation zone increases with Reynolds number; the present results are consistent with that. Figure 5.9 further shows the norm of the averaged velocity vector.

Figure 5.10 shows the average velocity fields for the streamwise and radial velocity components. The minimum negative axial velocity is $U_x = u_x/U_\infty = -0.44$ at $x/D = 0.89$. The average velocity profiles are compared to the experimental profiles in Figs 5.16 and Figs 5.17. In those figures, the thin lines represent the local grid size for the numerical results; the bullets represent the effective sampling points for the experiment. The qualitative agreement is excellent everywhere. Outside of the recirculation bubble, the numerical results also match perfectly with the experimental results. Inside the recirculation bubble, the experiment exhibits a slightly stronger recirculation: the axial velocity deficit is about 10% larger compared to the numerical results. The radial velocity has again an excellent qualitative agreement. Notice the small region along the profile at $x/D = 0.5$ where the radial velocity drops to zero (this occurs at the centerline of the separated shear layer): the numerical and experimental results exhibit the exact same behavior. The quantitative agreement is also very good. The largest difference is observed along the profile at $x/D = 1.5$, as it is located outside of the recirculation bubble for the numerical simulation, while it is still inside for the experiment.

We now turn to the turbulent fluctuation fields. The mean streamwise, radial and azimuthal velocity fluctuations are shown in Figs 5.11 and 5.12.

It is interesting to notice that the axial velocity fluctuations are significant even for small distances to the hemisphere ($x/D = 0.25$), inside the separated shear layer region. The radial and azimuthal fluctuations however remain fairly low up to $x/D = 0.5$. This shows that the separated shear layers are becoming rapidly unsteady but are still laminar. They only become fully three-dimensional further downstream. This is confirmed by the contours of the primary shear stress $\langle u'_x u'_r \rangle$, shown in Fig. 5.12. From a numerical point of view, the transition between grid levels can clearly be identified on the fluctuation plots. This is inherent to the method: as the grid size is modified by a factor of two, which translates into a large variation of the subgrid scale effective viscosity (see Fig. 5.15). This finally leads to a discontinuity of the resolved turbulence along the interfaces between grid levels. Figure 5.13 is provided to show that the statistics are sufficiently converged: the mean-cross correlation fluctuations, $\langle u'_x u'_\theta \rangle$ and $\langle u'_\theta u'_r \rangle$, should be zero when the turbulence statistics are converged; this is relatively well verified here.

The average streamwise and radial RMS velocity fluctuation profiles are compared to the experimental profiles in Figs 5.18 and 5.19. Again, the qualitative agreement is excellent. The computation captures very well the narrow fluctuation spike along the $x/D = 0.5$ profile, which is related to the KH instabilities. From a quantitative point of view, the numerical results also agree very well (especially in the near-wake region, where the grid is well resolved). The largest differences are observed in the region where the grid is coarsened.

The turbulent shear stress profiles are also shown using line plots in Fig. 5.20. However, they could not be compared with experimental results as the LDV system has only one component left: the cross correlation fluctuations could thus not be obtained.

Figure 5.14 shows the resolved turbulent kinetic energy field and the mean vorticity field. The turbulent kinetic energy is relatively low in the early separated shear layers as they are still laminar. Inside the recirculation bubble close to the back face the turbulence level is also very low: it is essentially a dead zone. The average vorticity field clearly shows the thin boundary layers and separated shear layers which are then rapidly diffused through turbulence.

Finally, the averaged SGS eddy-viscosity and an instantaneous SGS eddy-viscosity field are shown in Fig. 5.15. As already mentioned, large variations of the SGS viscosity are observed over the different grid levels (remember that the SGS viscosity scale with Δ^2 , where $\Delta = h$ is the local grid size). The average eddy-viscosity in the near wake region is very low: between 0.02ν and 0.2ν .



The instantaneous field shows however that the SGS model is locally active ($\nu_t/\nu = \mathcal{O}(1)$) even in the near wake regions. Further downstream, as the grid size increases, the LES model reaches high values locally. The SGS model is thus essential there to dissipate small-scale structures that cannot be resolved by the grid.

5.4.3 Force evaluation

Since there is no direct access to the pressure field, evaluating forces in vortex methods is not straightforward. The classical technique to evaluate the force, $\mathbf{F}$, acting on a body, is to compute the time derivative of the linear impulse:

$$\frac{\mathbf{F}}{\rho} = -\frac{d\mathbf{I}}{dt}, \quad (5.6)$$

where ρ is the density, and $\mathbf{I}$ is given by:

$$\mathbf{I} = \frac{1}{\mathcal{N}-1} \int_V \mathbf{x} \times \boldsymbol{\omega} dV, \quad (5.7)$$

with $\mathcal{N}$ the dimension of the space ($\mathcal{N} = 3$ in 3-D, $\mathcal{N} = 2$ in 2-D) and V the volume occupied by the fluid. This method is very robust, but requires to resolve the entire wake at an acceptable resolution. In long time simulations, this requirement cannot be met: the far wake is not sufficiently resolved and/or an outflow boundary condition is used.

An alternative is to perform a momentum balance on a control volume that encloses the body. Such an approach was developed by Noca *et al.* [73, 74] and was successfully applied to the Lagrangian vortex element method by Ploumhans *et al.* [78]. Consider a control volume V_{CV} , with outer surface S_{CV} , and inner surface S , the body surface. The force acting on the body can be expressed as [73, 74]:

$$\begin{aligned} \frac{\mathbf{F}}{\rho} = & -\frac{1}{\mathcal{N}-1} \frac{d}{dt} \int_{V_{CV}} \mathbf{x} \times \boldsymbol{\omega} dV + \int_{S_{CV}} \left(\frac{1}{2} (\mathbf{u} \cdot \mathbf{u}) \mathbf{n} - (\mathbf{n} \cdot \mathbf{u}) \mathbf{u} \right) dS \\ & - \frac{1}{\mathcal{N}-1} \int_{S_{CV}} (\mathbf{n} \cdot \mathbf{u}) (\mathbf{x} \times \boldsymbol{\omega}) dS + \frac{1}{\mathcal{N}-1} \int_{S_{CV}} (\mathbf{n} \cdot \boldsymbol{\omega}) (\mathbf{x} \times \mathbf{u}) dS \\ & + \frac{1}{\mathcal{N}-1} \int_{S_{CV}} \mathbf{x} \times (\mathbf{n} \times \nabla \cdot \mathbf{T}) dS + \int_{S_{CV}} \mathbf{n} \cdot \mathbf{T} dS \\ & + \frac{1}{\mathcal{N}-1} \frac{d}{dt} \int_S \mathbf{x} \times (\mathbf{n} \times \mathbf{u}) dS - \int_S \mathbf{n} \cdot \mathbf{u} dS, \end{aligned} \quad (5.8)$$

with $\mathbf{T} = (\nu + \nu_t) (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$. If the velocity vanishes at the wall, the integrals on S cancel. In that case, Eq. (5.8) does not require information at the boundary, and the CV can be chosen so as to enclose only a well-resolved region of the flow. Ploumhans [78] further adapted Eq. (5.8), to transform the volume integral into a surface integral, that can then be evaluated using the regularized fields:

$$\int_{V_{CV}} \mathbf{x} \times \boldsymbol{\omega} dV = \int_{S_{CV}+S} ((\mathbf{u} \cdot \mathbf{x}) \mathbf{n} - \mathbf{u} (\mathbf{x} \cdot \mathbf{n}) + (\mathcal{N} - 1) \mathbf{x} (\mathbf{u} \cdot \mathbf{n})) dS \quad (5.9)$$

Note that the surface integral on S vanishes when $\mathbf{u} = \mathbf{0}$ at the wall (as for the flows considered in this paper).

In the present simulation, the control volume approach was used. Two control volumes were defined: (i) a spherical volume centered at $\mathbf{x}_c/D = [0, 0, 0]$ with radius $r/D = 1.5$ and (ii) a cubic control volume, centered at $\mathbf{x}_{\min}/D = [0, 0, 0]$ with length $c/D = 2$.

The drag coefficient is defined as:

$$C_D = \frac{F_x}{(\frac{1}{2}\rho U_\infty^2)(\pi D^2/4)}, \quad (5.10)$$

It was obtained for both control volumes: (i) $C_D = 0.492$ and (ii) $C_D = 0.496$. Both results are in good agreement. Recall that the drag coefficient for the sphere is $C_D = 0.4$ at $Re = 10,000$ [1, 16, 17, 40].

In order to further validate this result, we have computed the pressure distribution at the surface of the hemisphere using an approximate approach. The idea is to obtain the potential flow outside of the boundary layer and the separated shear layer. The pressure at these locations is then simply given by the Bernoulli equation. Finally, based on the fact that the pressure does not vary across a boundary layer and the separated shear layer, the surface pressure can be obtained (the recirculation zone is considered as a “dead zone” and is thus supposed to be at constant pressure).

Figure 5.22a shows a zoom on the mean velocity field near the hemisphere. A curvilinear region is defined along the upstream part of the body and it extends to cover the separated shear layer. The curvilinear system is defined



TABLE 5.1 – Drag coefficients for the flow past the hemisphere. CVC and CVS refer to the results obtained from the spherical and cubic control volume respectively. BER are the results obtained using the Bernoulli approach.

		Re	C_D	C_{D_f}/C_D	Estimated Error
Present	CVC	$3.0 \cdot 10^3$	0.492	9.3%	
	CVS	$3.0 \cdot 10^3$	0.496		
	BER	$3.0 \cdot 10^3$	0.504		
Cottin		$5.5 \cdot 10^4$	0.434		8%
		$1.1 \cdot 10^5$	0.469		6%
Hoerner		10^4 – 10^6	0.42		

as follows:

$$\phi \leq \pi/2 \quad \begin{cases} x &= D \left(\frac{1}{2} + s \right) \cos(\pi - \phi) \\ y &= D \left(\frac{1}{2} + s \right) \sin(\pi - \phi) \end{cases} \quad (5.11)$$

$$\phi > \pi/2 \quad \begin{cases} x &= D \left(\frac{\phi}{2} - \frac{\pi}{4} \right) \\ y &= D \left(\frac{1}{2} + s \right) \end{cases} \quad (5.12)$$

where, $\phi = [0, \frac{3}{4}\pi]$ and $s = [0, 0.15]$. Figures 5.22b and 5.22c show the mean velocity and vorticity fields in the curvilinear coordinate system. The boundary layer vorticity thickness, as function of ϕ , is defined as the wall normal distance $\delta_\omega^{.95} = D s_\delta$, where s_δ is such that,

$$\omega(\phi, s_\delta) = 0.05 \omega_{\max, \phi} \quad , \quad (5.13)$$

and $\omega_{\max, \phi}$ is defined as the maximum of vorticity along the s coordinate: $\omega_{\max, \phi} = \max(\omega(\phi, s))$ for s in $[0, 0.15]$. The boundary layer vorticity thickness is plotted in the absolute and curvilinear coordinate systems in Figs 5.22a, b and c.

The external potential velocity, $u_p(\phi)$, is then obtained along the edge of the boundary layer. The corresponding pressure coefficient, $c_p = (1 - u_p^2/U_\infty^2)$, is shown in Fig. 5.22d. The pressure coefficient on the back of the hemisphere

is obtained from its value along the separated shear layer (i.e., for $\phi > \pi/2$ in 5.22d): $c_{p,\text{base}} = -0.295$.

The skin friction can be obtained directly by evaluating the vorticity at the body surface: $c_f(\phi) = 2\nu\omega(\phi, 0)/U_\infty^2$. It is shown in Fig. 5.22e.

The pressure and friction components of the total drag coefficient can now be evaluated by integrating the pressure and skin friction coefficients over the surface of the hemisphere:

$$C_{D_p} = 2 \int_0^{\pi/2} c_p(\phi) \sin(\phi) \cos(\phi) d\phi - c_{p,\text{base}} = 0.457 \quad (5.14)$$

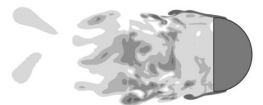
$$C_{D_f} = 2 \int_0^{\pi/2} c_f(\phi) \sin^2(\phi) d\phi = 0.047 \quad (5.15)$$

The total drag coefficient obtained by the approximate approach is thus: $C_D = 0.504$, which is in very good agreement with the value obtained through the control volume approach. Notice that the skin friction represents about 10% of the total drag coefficient. For the flow past a sphere at $Re = 10,000$, Achenbach [1] obtains $C_{D_f}/C_D = 5.5\%$ experimentally while Constantinescu *et al.* [16, 17] and Georges *et al.* [40] both report a fraction of about 8.5%, from their numerical simulations. At lower Reynolds numbers, the skin friction is obviously increased: it should scale as $1/\sqrt{Re}$.

The drag coefficient of the hemisphere was measured experimentally by Cottin (private communication) using a force balance at $Re = 55,000$ and $Re = 110,000$ in our low speed, open-section, wind tunnel. Due to the small size of the hemisphere ($D = 55$ mm), the measured force is small (of the order of one Newton). The incertitude on the measured force is thus relatively large. The drag coefficient at $Re = 55,000$ and $Re = 110,000$ was found to be $C_D = 0.434 \pm 8\%$ and $C_D = 0.469 \pm 6\%$ respectively. Those results are in good agreement with our numerical results. However, it is clear that they are not an absolute reference. Finally, Hoerner [44] provides the approximate drag coefficient for several bluff bodies. The hemisphere is reported at $C_D = 0.42$ for $10^4 < Re < 10^6$. The drag coefficients for the present simulation and the experimental references are given in Table 5.1.

5.5 Summary

In this chapter, the hierarchical redistribution procedure was used to perform the simulation of the flow past a hemisphere at $Re = 3,000$ in an efficient way:



the grid refinement procedure allowed to precisely control the distribution of computational elements. Our simulation was able to correctly capture the initially laminar detached shear layers and their transition to turbulence through Kelvin-Helmoltz instabilities. The results were compared, from a qualitative and quantitative point of view, with experimental results obtained for the flow past a hemisphere at $Re = 3,000$ (LIF visualizations) and $Re = 55,000$ (LDV measurements). The mean velocity and turbulent fluctuation profiles are in excellent qualitative agreement, and in good quantitative agreement, with those obtained experimentally. The main-shedding frequency (low frequency) and the shear layer frequency (high frequency) were both detected and compare well the values reported for the flow past a sphere in the subcritical regime.

The turbulent flow and the grid configuration used here are really challenging for the hierarchical grid refinement approach (from the point of view of information transfer between grid levels). The vorticity generated at the body surface is initially transported on the fine grid. Downstream, it is transferred to a coarser level grid. Then, slightly further downstream (before the end of the recirculation region), the grid is further coarsened. At that location, vorticity is thus also permanently convected from a coarse grid to a fine grid region. Furthermore, there is a continuous interaction between the vorticity inside the recirculation region and the refined detached shear layers. These results show that the hierarchical redistribution procedure is able to handle three-dimensional, complex, turbulent flows using relatively involved grid refinement regions.

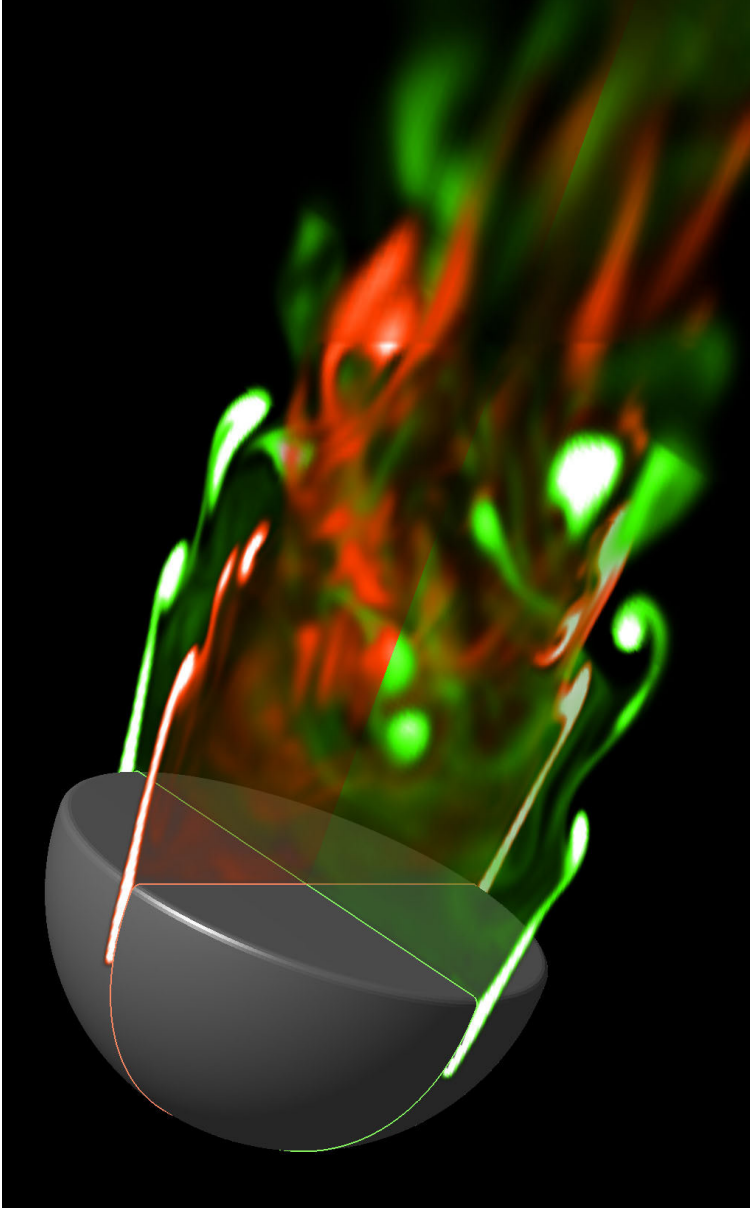
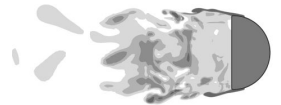


FIGURE 5.1 – Perspective view of two slices through the tracer field for the flow past the hemisphere at $Re = 3,000$: x - y and y - z midplanes are illuminated using respectively a green “laser sheet” and a red “laser sheet”.



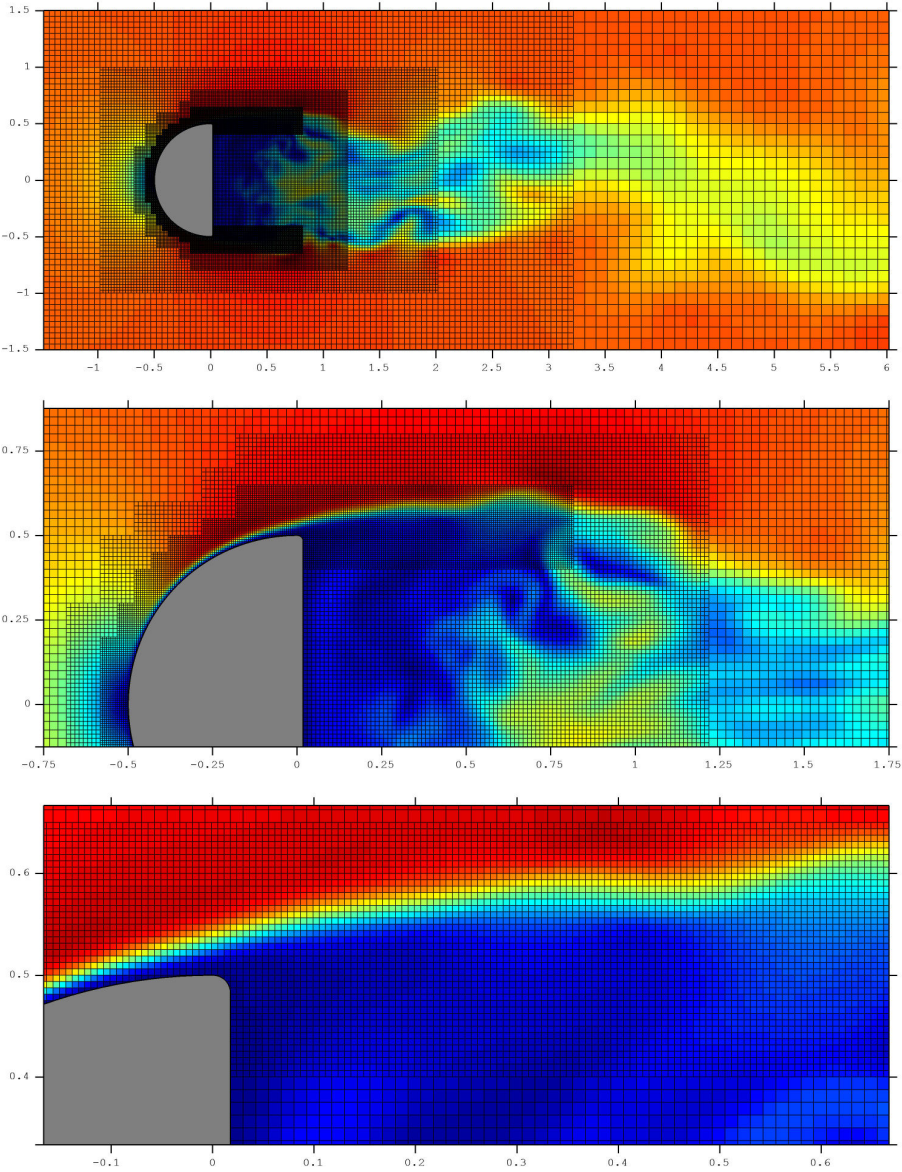


FIGURE 5.2 – Hierarchical grid for the flow past the hemisphere at $Re = 3,000$. Three views of the x - y midplane ($z/D = 0$) through the 3-D grid are shown. The field is a snapshot of the velocity norm at a time when the flow is in the turbulent regime.

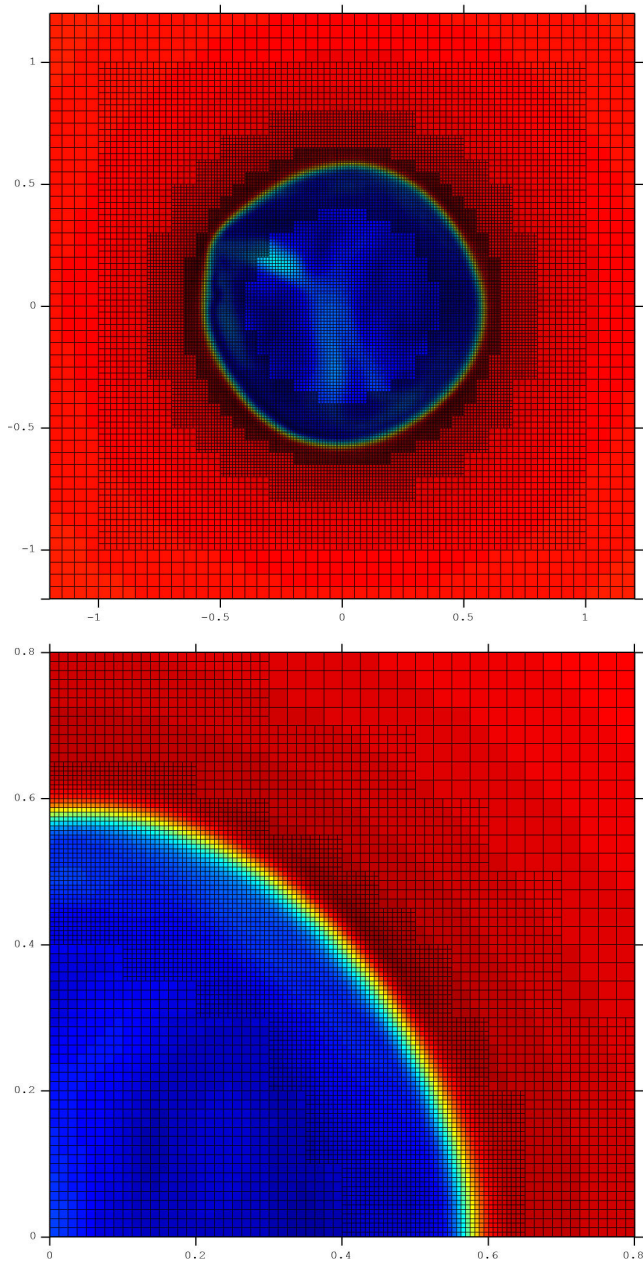


FIGURE 5.3 – Hierarchical grid for the flow past the hemisphere $Re = 3,000$. Two views of the y - z midplane ($x/D = 0.25$) through the 3-D grid are shown. The field is a snapshot of the velocity norm at a time when the flow is in the turbulent regime.



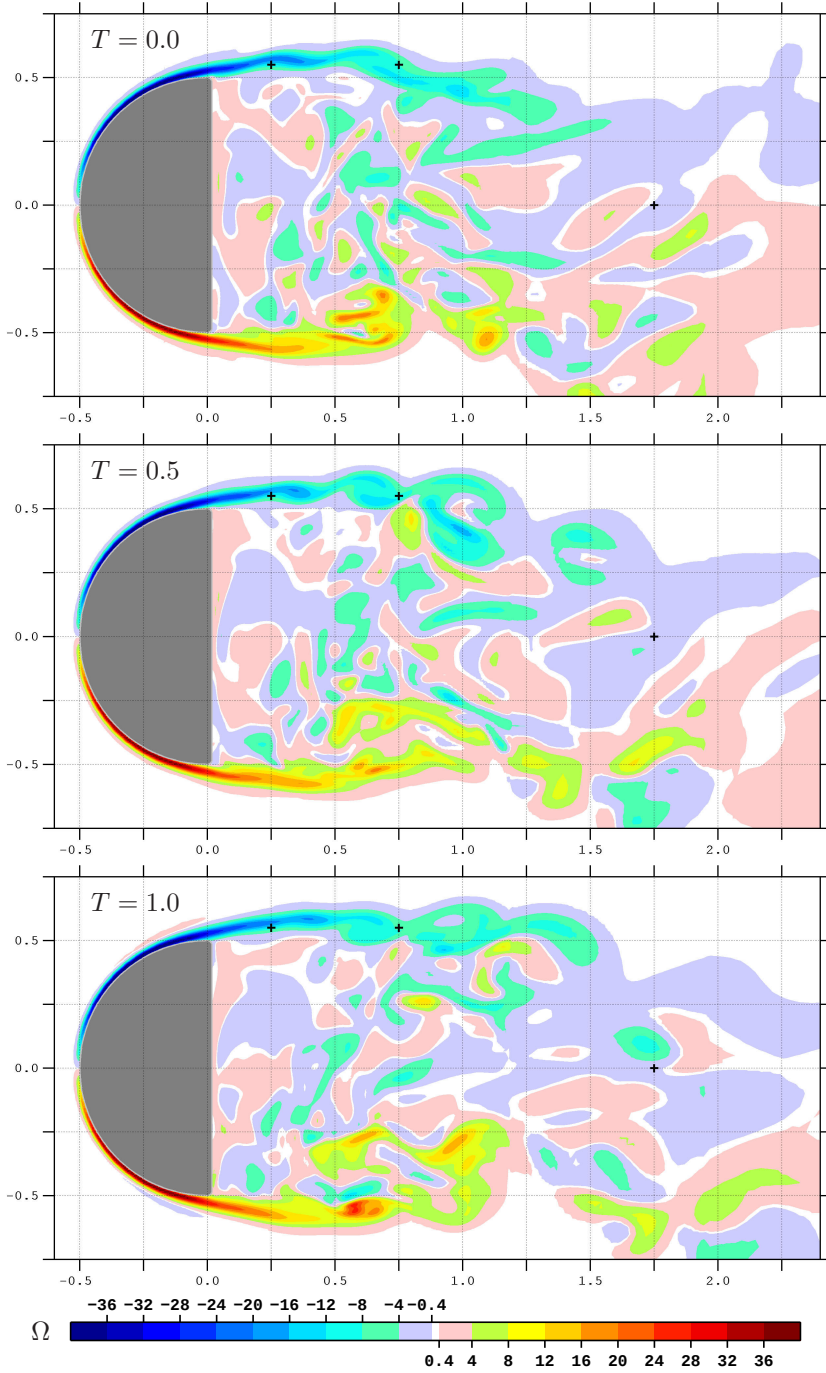


FIGURE 5.4 – Instantaneous vorticity field, $\Omega_y = \frac{\omega_y D}{U_\infty}$, in x - y plane ($z = 0$) at three distinctive times, $T = \frac{t U_\infty}{D}$. $T = 0$ is an arbitrary time when the flow has reached its turbulent regime. The locations where the velocity time signals are also shown (+).

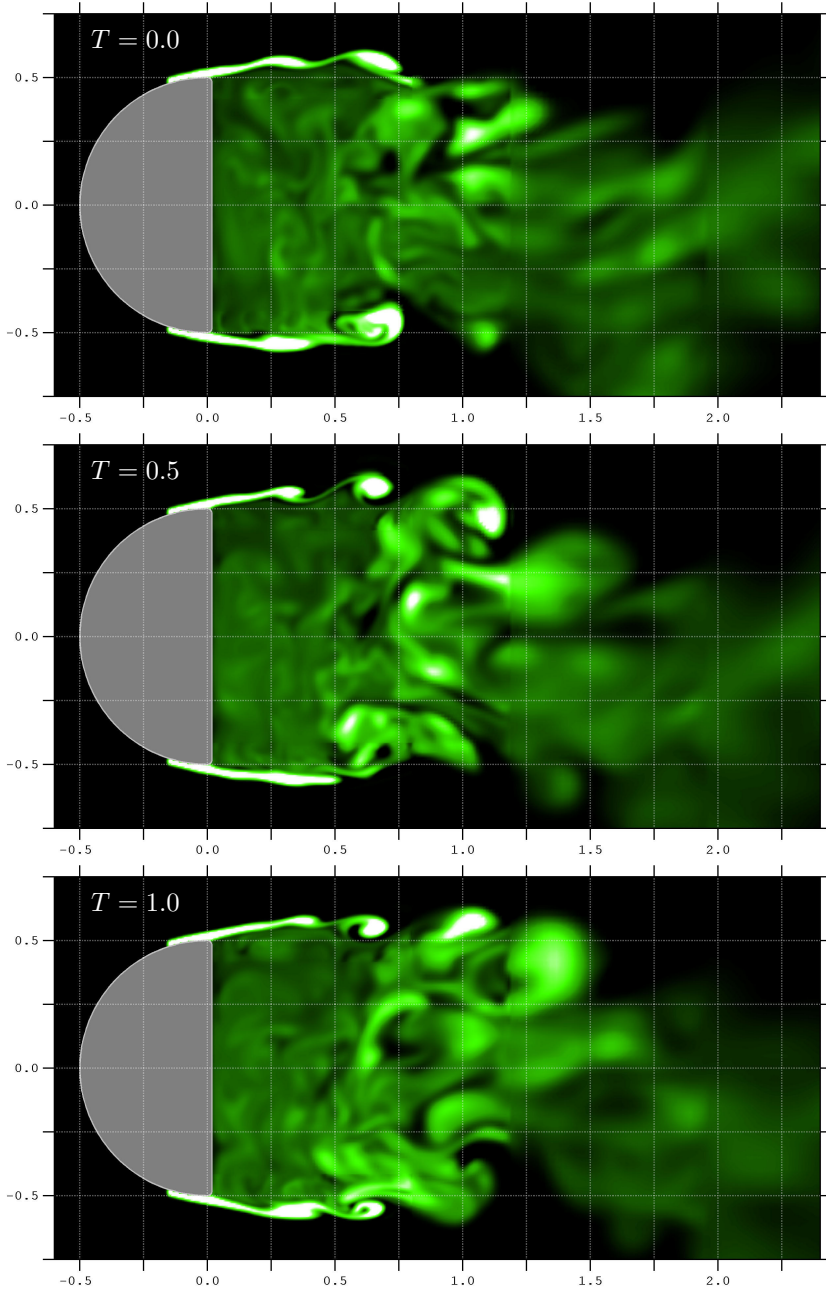
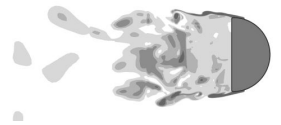


FIGURE 5.5 – Instantaneous tracer field in x - y plane ($z = 0$) at three distinctive times, $T = \frac{tU_\infty}{D}$. $T = 0$ is an arbitrary time when the flow has reached its turbulent regime (same as Fig. 5.4).



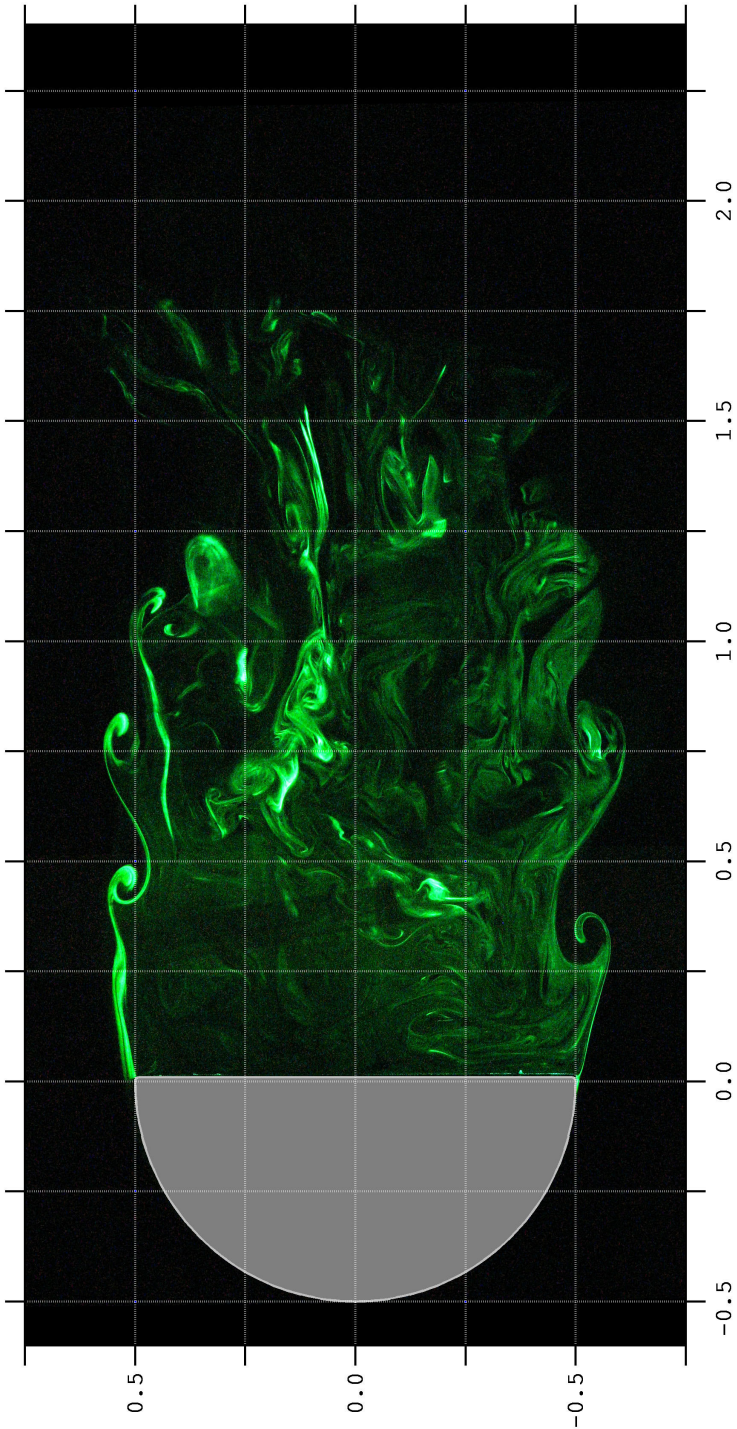


FIGURE 5.6 – Experimental visualization of the flow past the hemisphere at $Re = 3,000$ in a towing tank. Fluorescein is injected in the boundary layer at the front of the hemisphere. It is visualized using a laser sheet through the midplane.

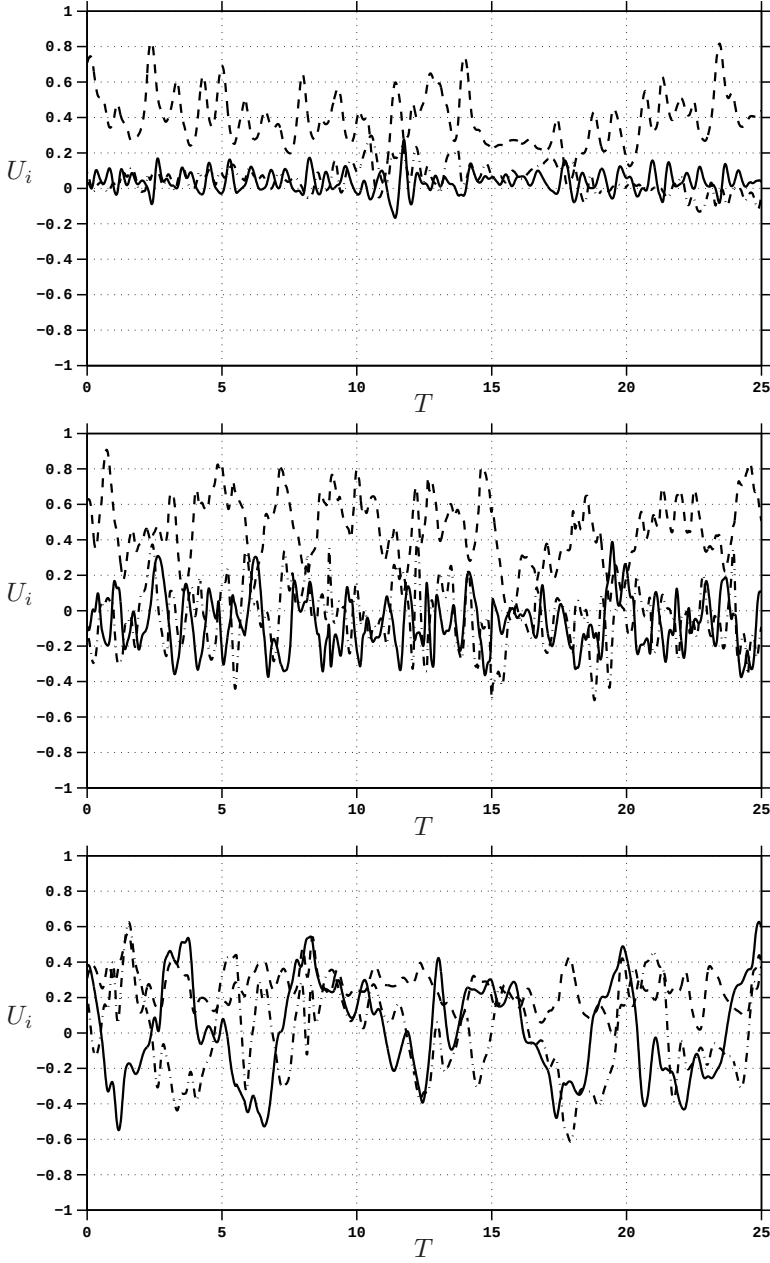
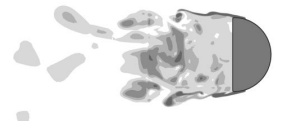


FIGURE 5.7 – Time signal, $T = tU_\infty/D$, for the velocity components, U_i , at three locations in the wake: $\mathbf{x}/D = \{0.25, 0.6, 0.0\}$ (top), $\mathbf{x}/D = \{0.75, 0.5, 0.0\}$ (middle) and $\mathbf{x}/D = \{1.75, 0.0, 0.0\}$ (bottom). The three locations are shown in Fig. 5.4. $U_x = u_x/U_\infty$ (dash), $U_y = u_y/U_\infty$ (plain) and $U_z = u_z/U_\infty$ (dash-dot).



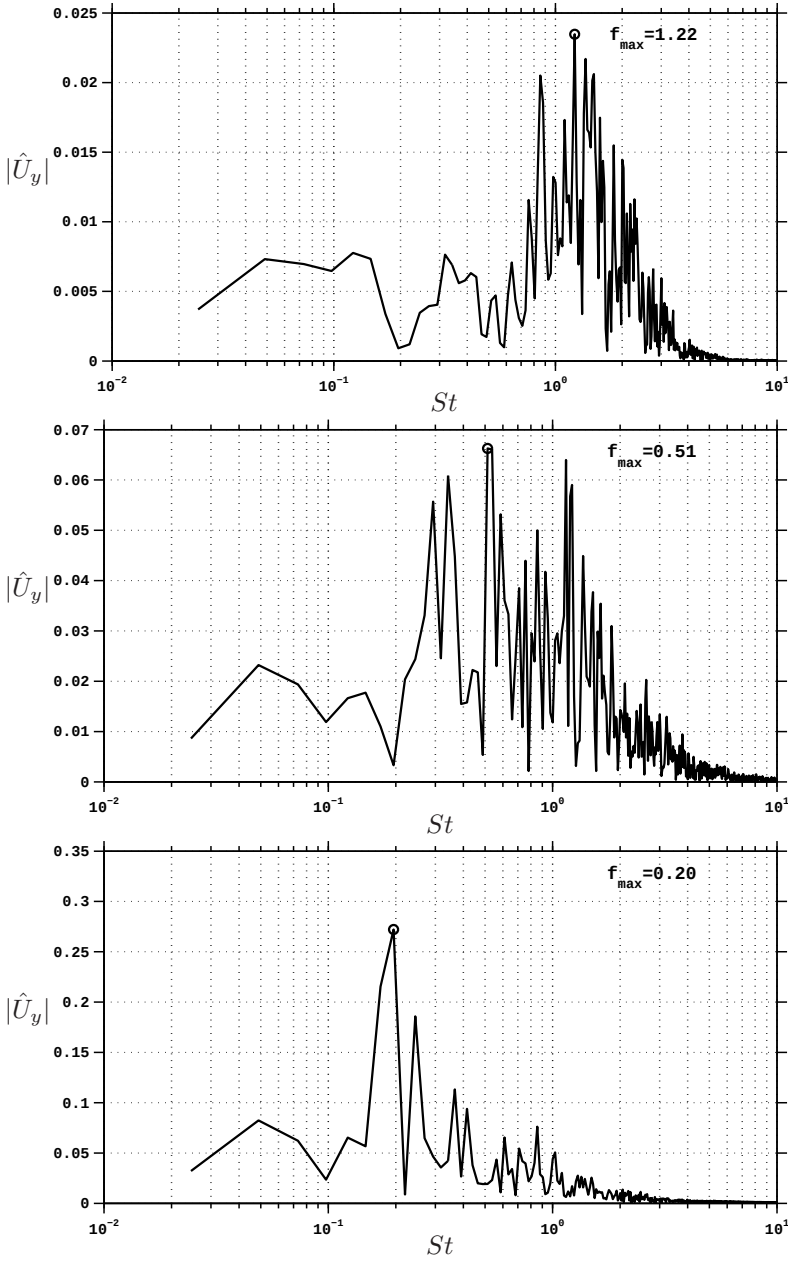


FIGURE 5.8 – Temporal spectra for the radial velocity component, $\hat{U}_y = \hat{u}_y/U_\infty$, at three locations in the wake: $\mathbf{x}/D = \{0.25, 0.6, 0.0\}$ (top), $\mathbf{x}/D = \{0.75, 0.5, 0.0\}$ (middle) and $\mathbf{x}/D = \{1.75, 0.0, 0.0\}$ (bottom). The three locations are shown in Fig. 5.4. The dimensionless frequency is defined as $St = f D/U_\infty$.

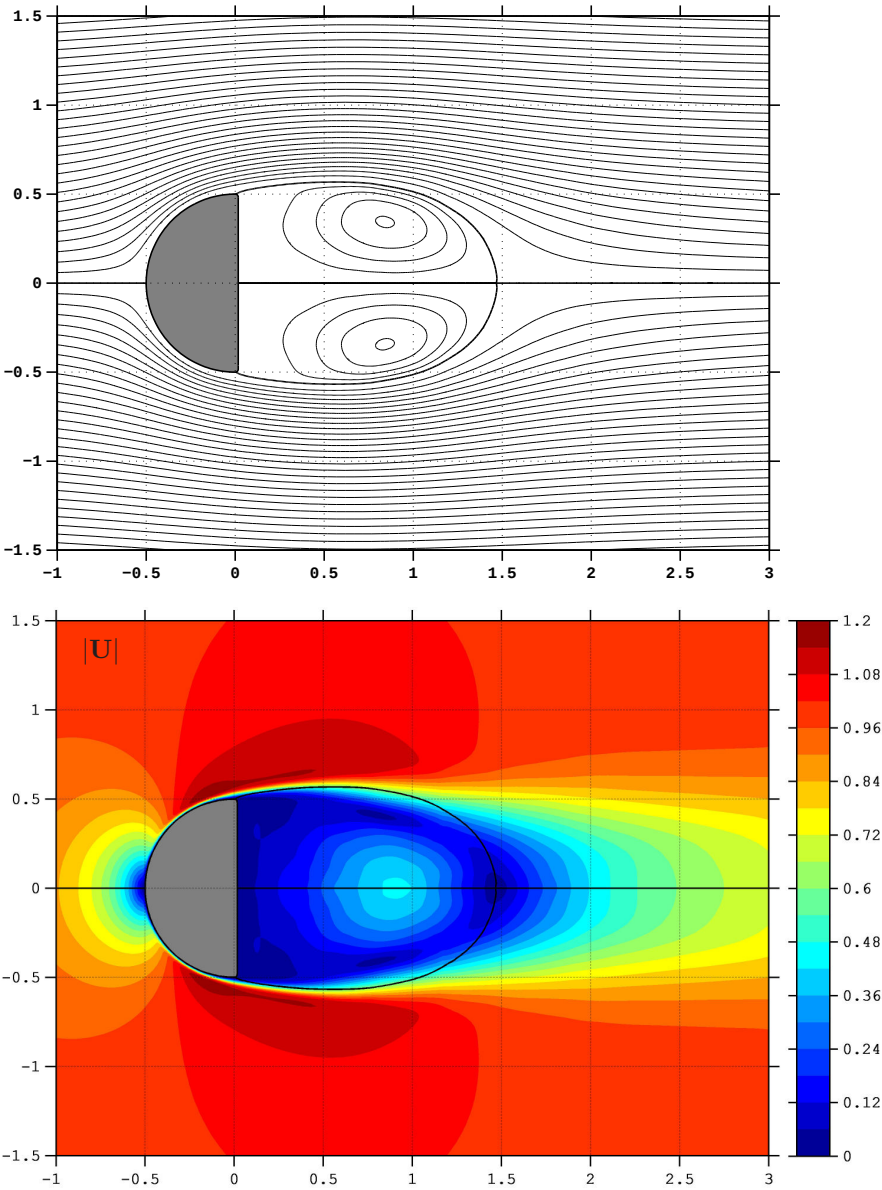
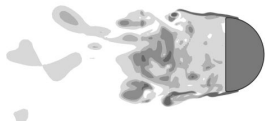


FIGURE 5.9 – Top: Mean streamlines. Bottom: mean velocity field (norm).



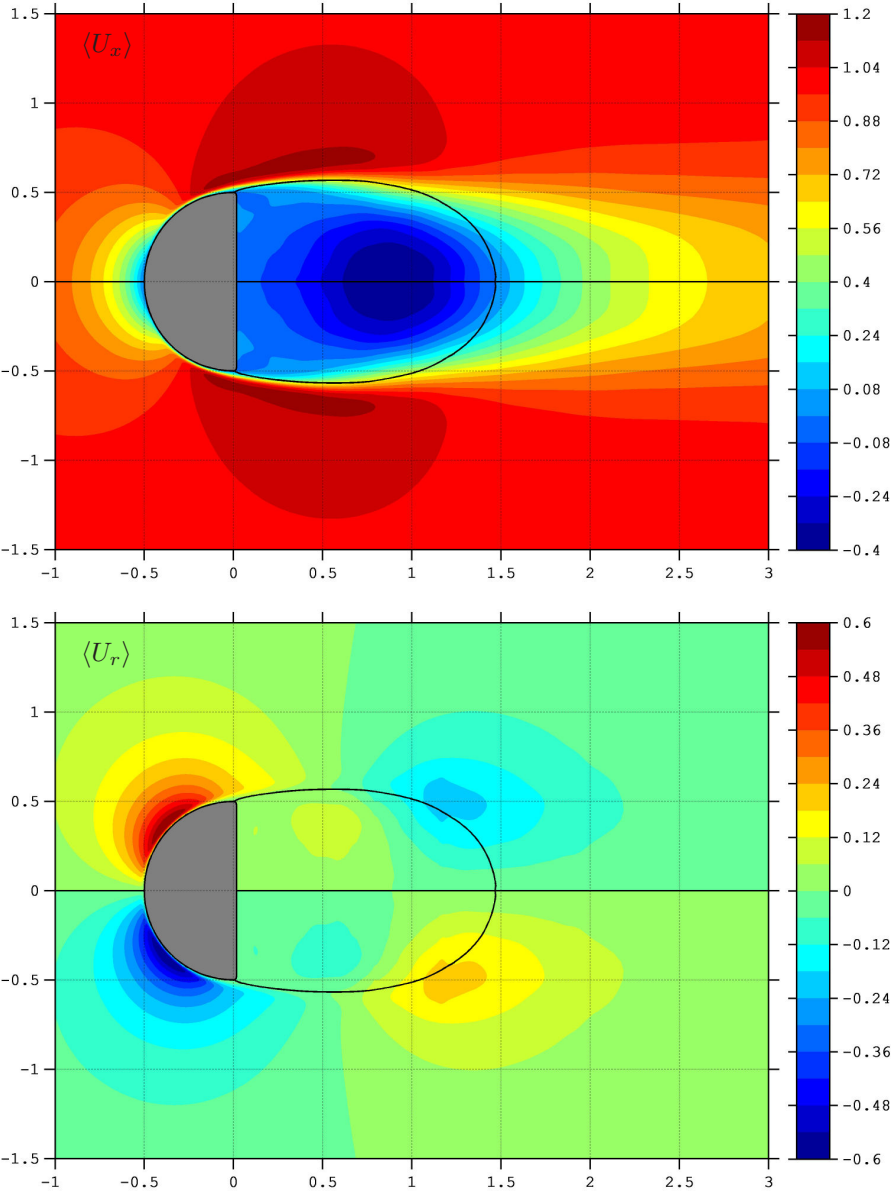


FIGURE 5.10 – Top: Mean streamwise velocity, $\langle U_x \rangle = \langle u_x \rangle / U_\infty$. Bottom: Mean radial velocity, $\langle U_r \rangle = \langle u_r \rangle / U_\infty$.

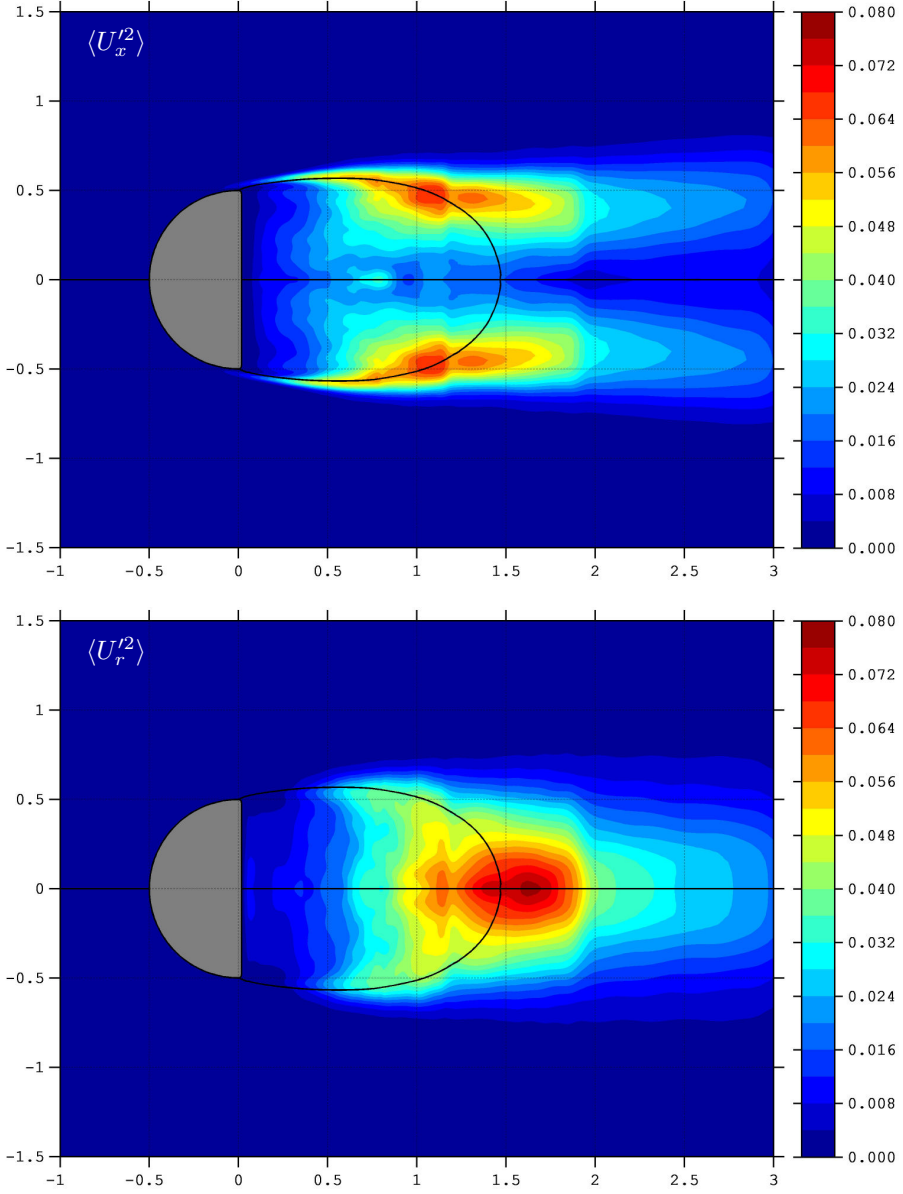
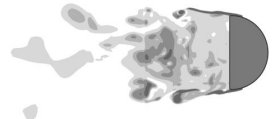


FIGURE 5.11 – Top: Mean streamwise velocity fluctuations, $\langle U_x'^2 \rangle = \langle u_x'^2 \rangle / U_\infty^2$. Bottom: Mean radial velocity fluctuations, $\langle U_r'^2 \rangle = \langle u_r'^2 \rangle / U_\infty^2$.



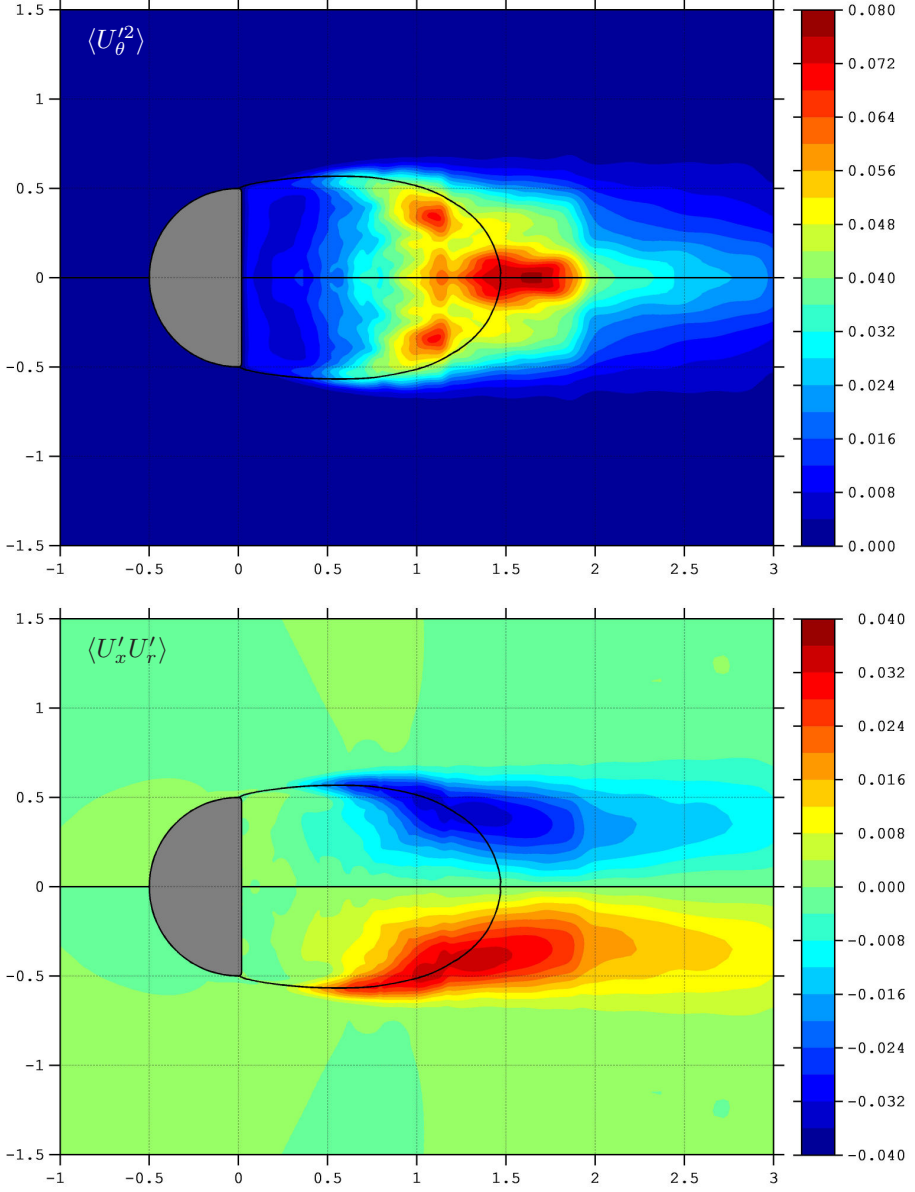


FIGURE 5.12 – Top: Mean azimuthal velocity fluctuations, $\langle U_{\theta}'^2 \rangle = \langle u_{\theta}'^2 \rangle / U_{\infty}^2$. Bottom: Mean cross-correlation fluctuations, $\langle U_x' U_r' \rangle = \langle u_x' u_r' \rangle / U_{\infty}^2$.

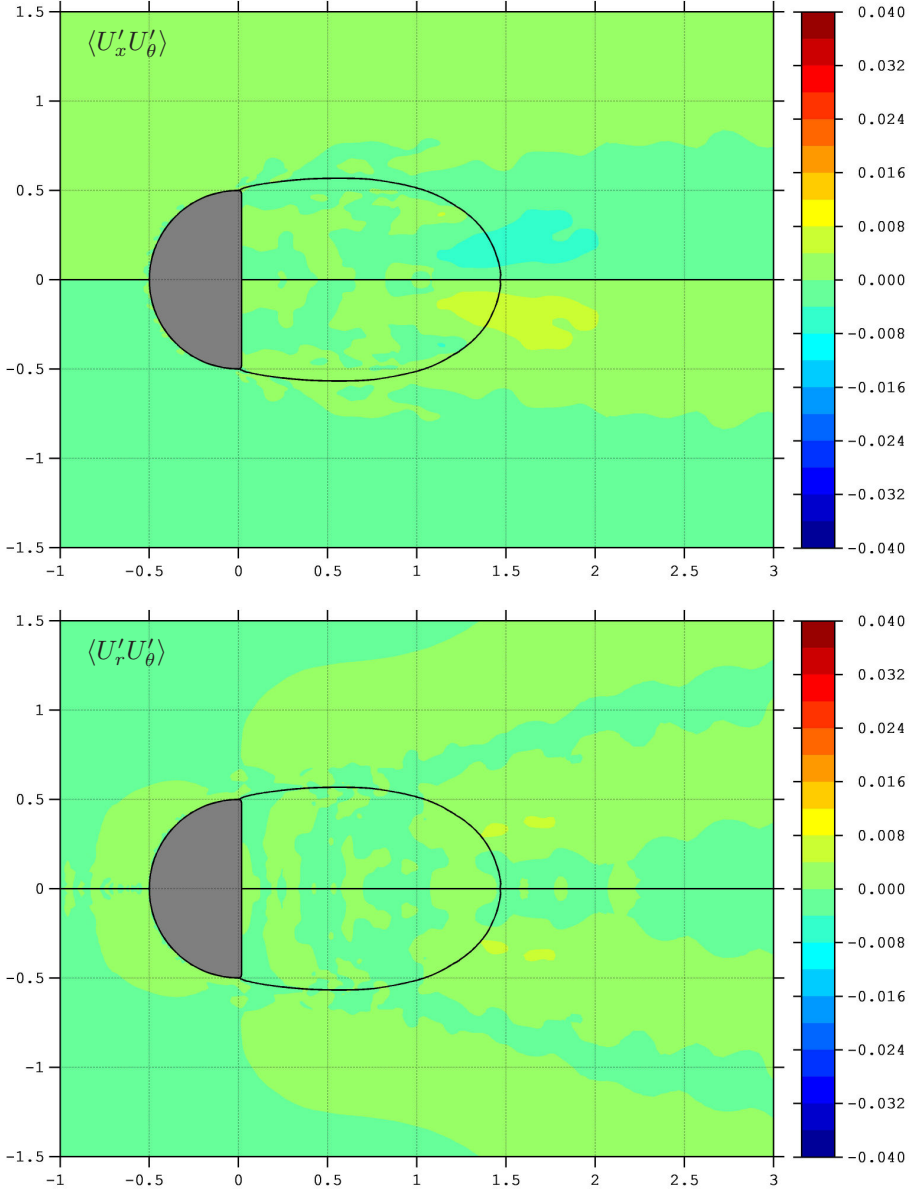
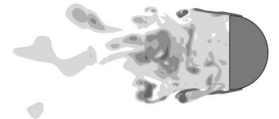


FIGURE 5.13 – Top: Mean cross-correlation fluctuations, $\langle U'_x U'_\theta \rangle = \langle u'_x u'_\theta \rangle / U_\infty^2$.
 Bottom: Mean cross-correlation fluctuations, $\langle U'_r U'_\theta \rangle = \langle u'_\theta u'_r \rangle / U_\infty^2$.



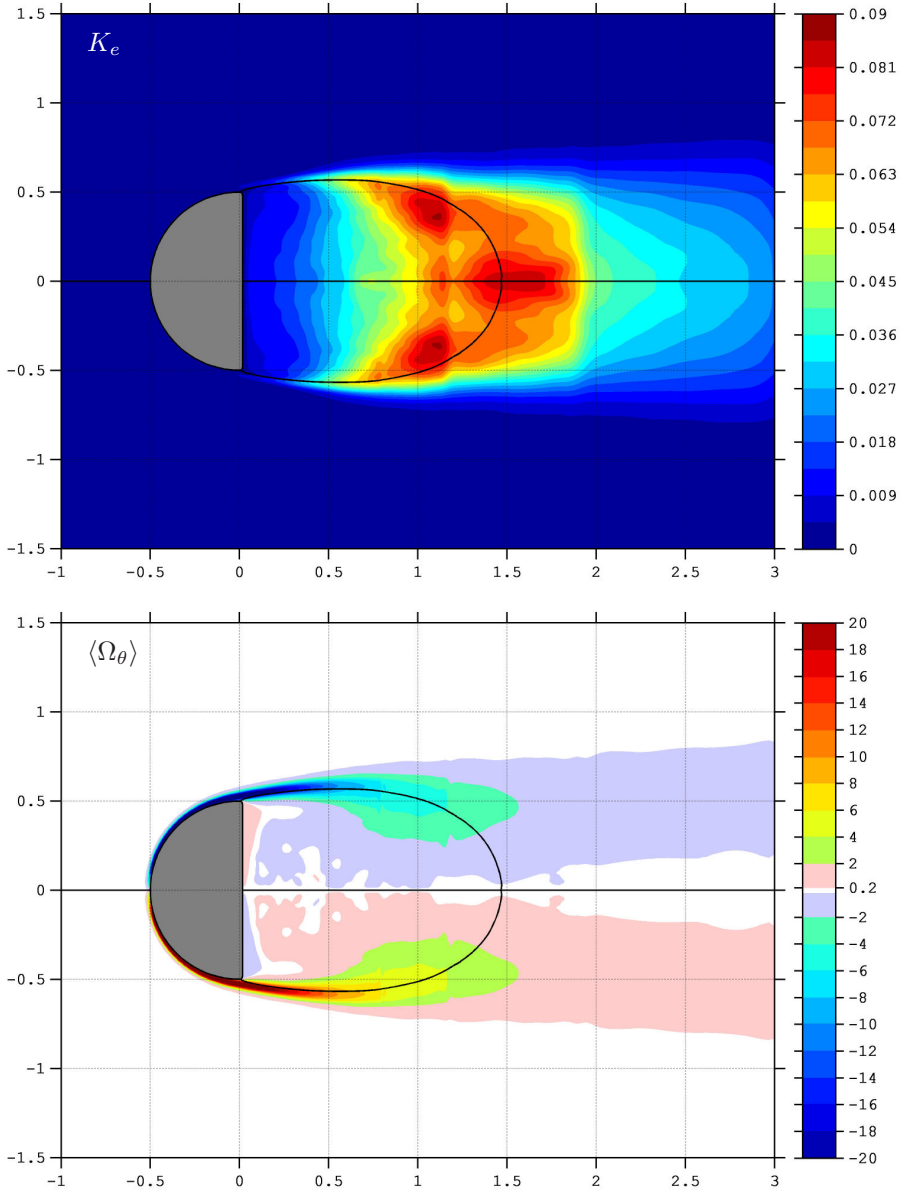


FIGURE 5.14 – Top: Mean kinetic energy, $K_e = \frac{1}{2} \frac{\langle u_x'^2 \rangle + \langle u_y'^2 \rangle + \langle u_\theta'^2 \rangle}{U_\infty^2}$. Bottom: Mean vorticity field, $\langle \Omega_\theta \rangle = \frac{\langle \omega_\theta \rangle D}{U_\infty}$.

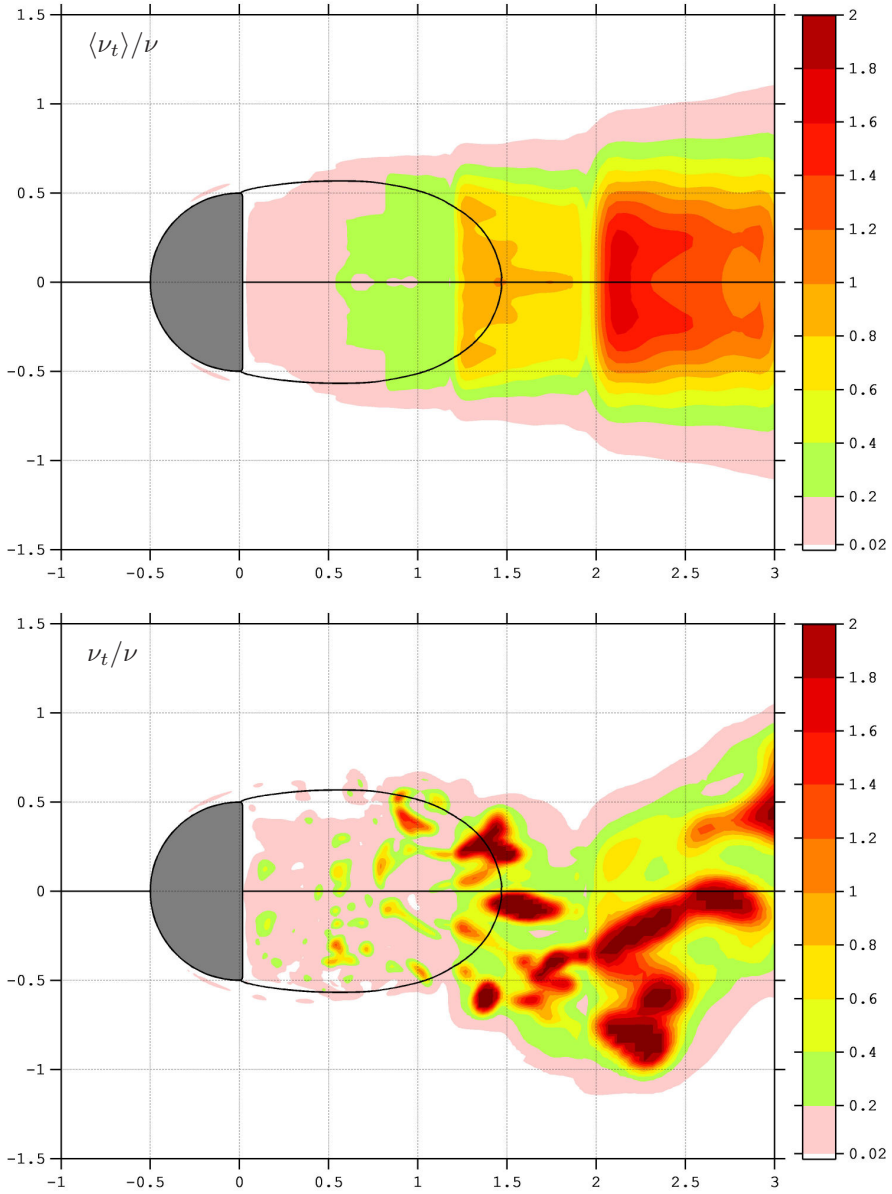


FIGURE 5.15 – Top: Mean subgrid-scale eddy viscosity, $\langle \nu_t \rangle$. Bottom: Instantaneous subgrid-scale eddy viscosity at $T = 0.5$, where $T = 0$ is an arbitrary time when the flow has reached its turbulent regime (same as Fig. 5.4).



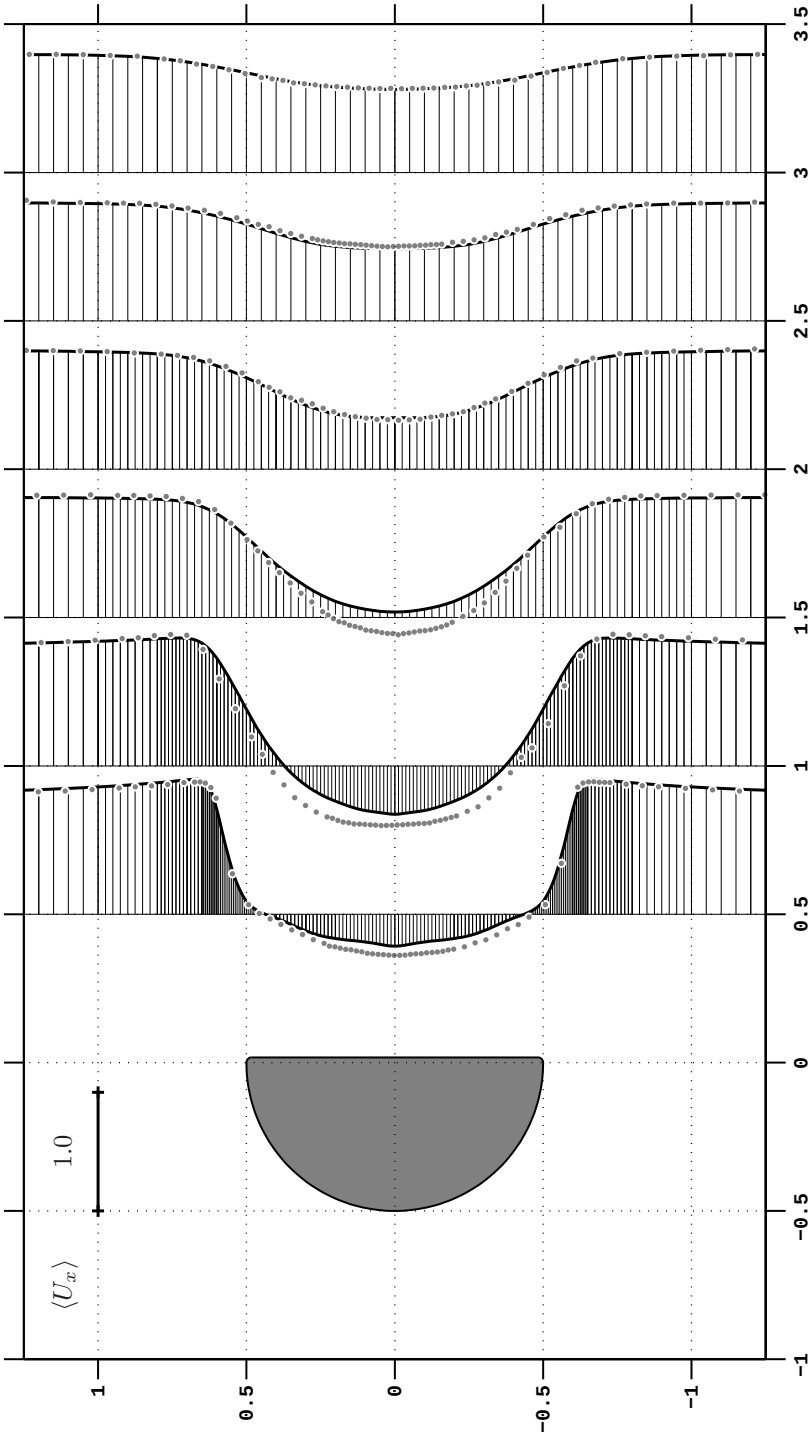


FIGURE 5.16 – Mean streamwise velocity profiles, U_x : numerical results at $Re = 3,000$ (plain) versus experimental wind tunnel LDV measurements at $Re = 55,000$ (bullets).

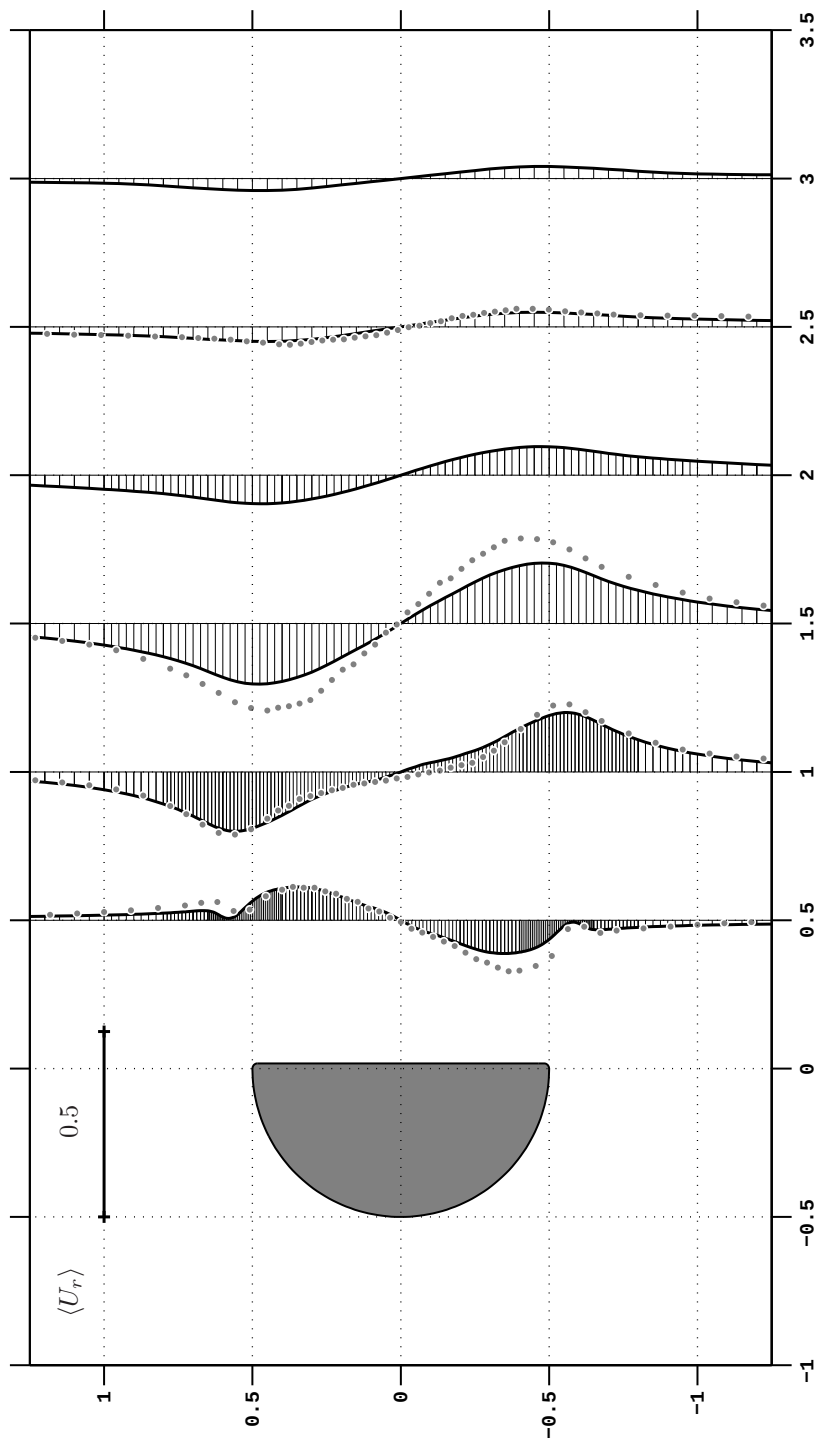


FIGURE 5.17 – Mean radial velocity profiles, U_r : numerical results at $Re = 3,000$ (plain) versus experimental wind tunnel LDV measurements at $Re = 55,000$ (bullets).



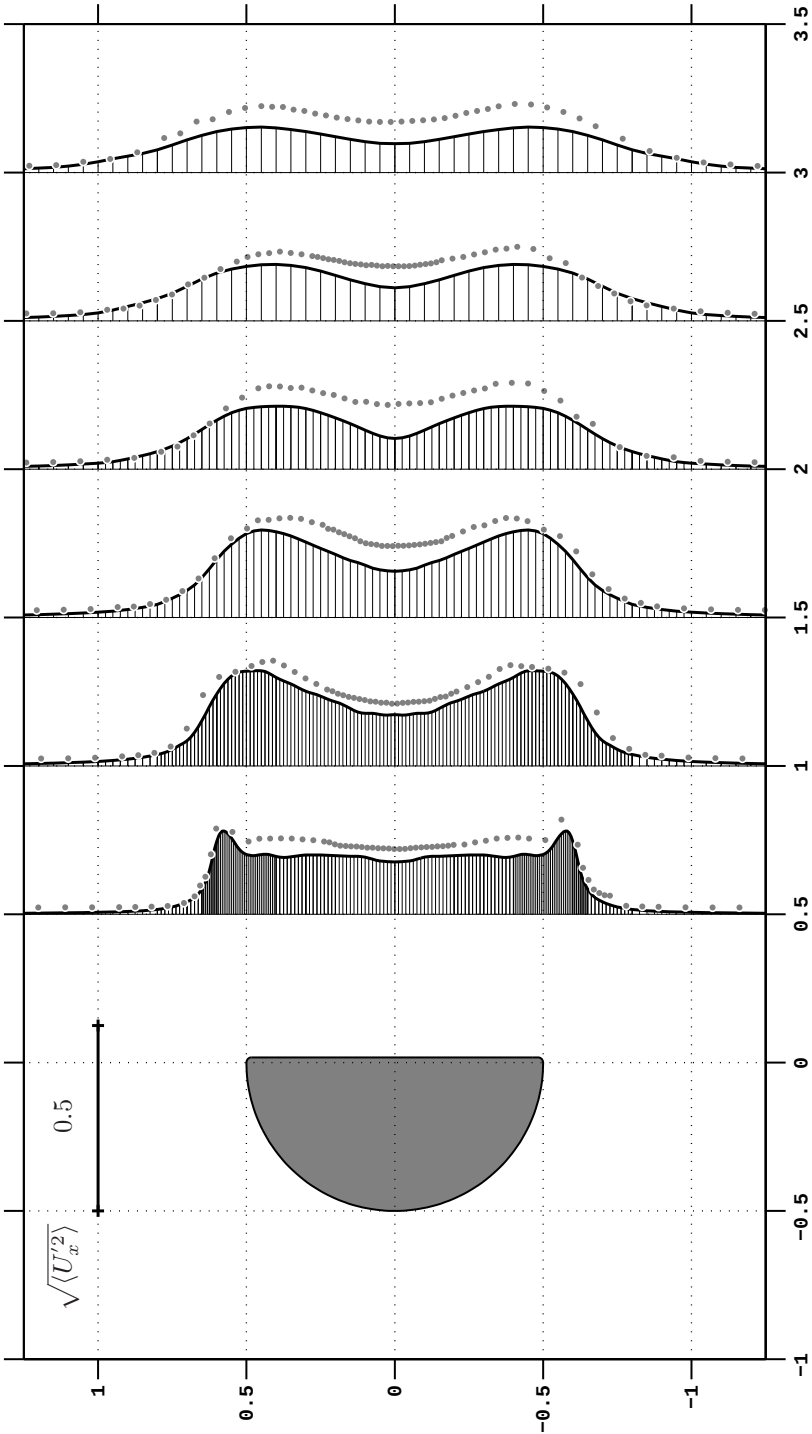


FIGURE 5.18 – Mean streamwise RMS velocity fluctuation profiles, U'_x : numerical results at $Re = 3,000$ (plain) versus experimental wind tunnel LDV measurements at $Re = 55,000$ (bullets).

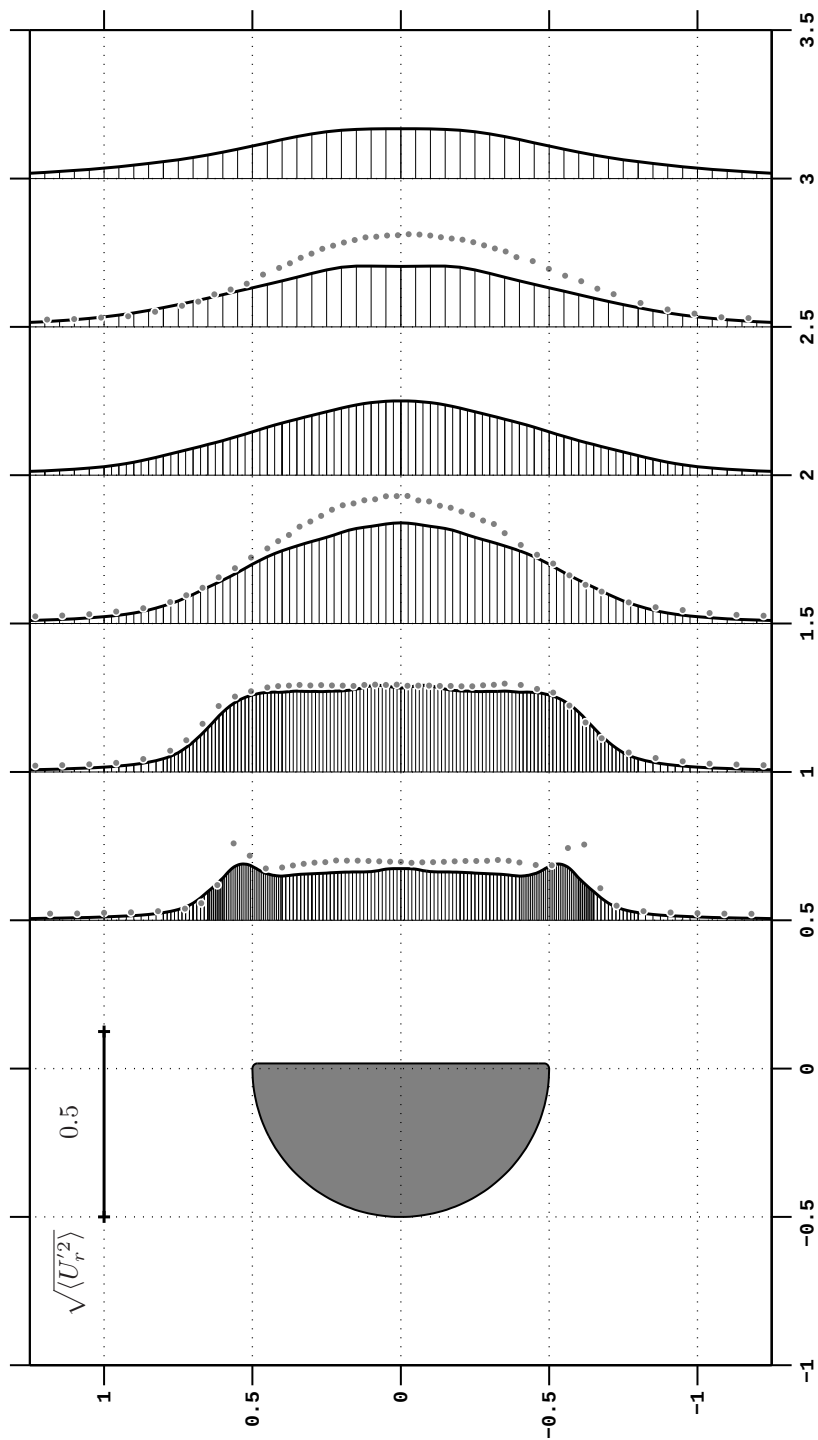


FIGURE 5.19 – Mean radial RMS velocity fluctuation profiles, $\sqrt{\langle U_r'^2 \rangle}$: numerical results at $Re = 3,000$ (plain) versus experimental wind tunnel LDV measurements at $Re = 55,000$ (bullets).



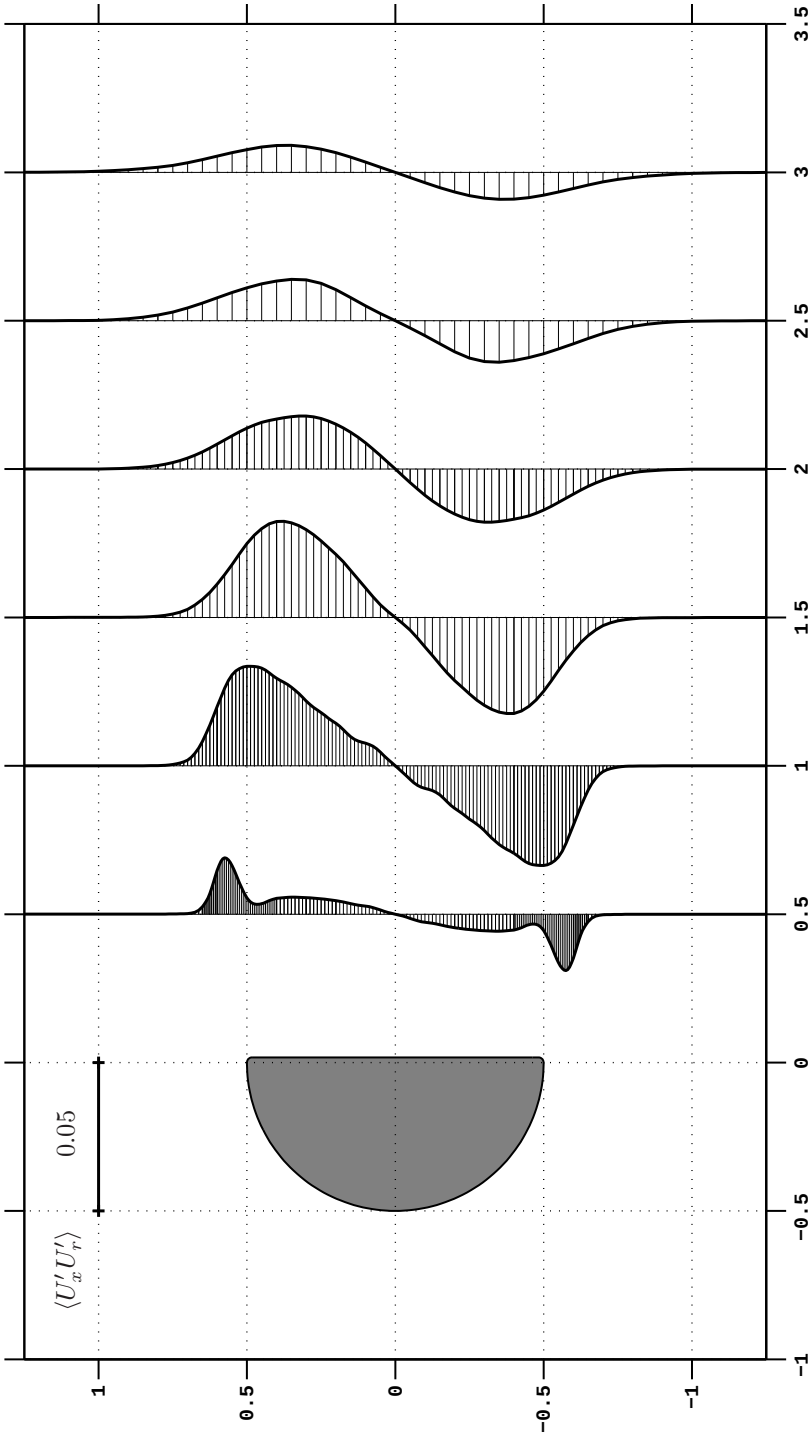


FIGURE 5.20 – Mean cross-correlation velocity fluctuation profiles, $\langle U'_x U'_r \rangle$: numerical results at $Re = 3,000$ (plain). Experimental wind tunnel LDV measurements could not be obtained.

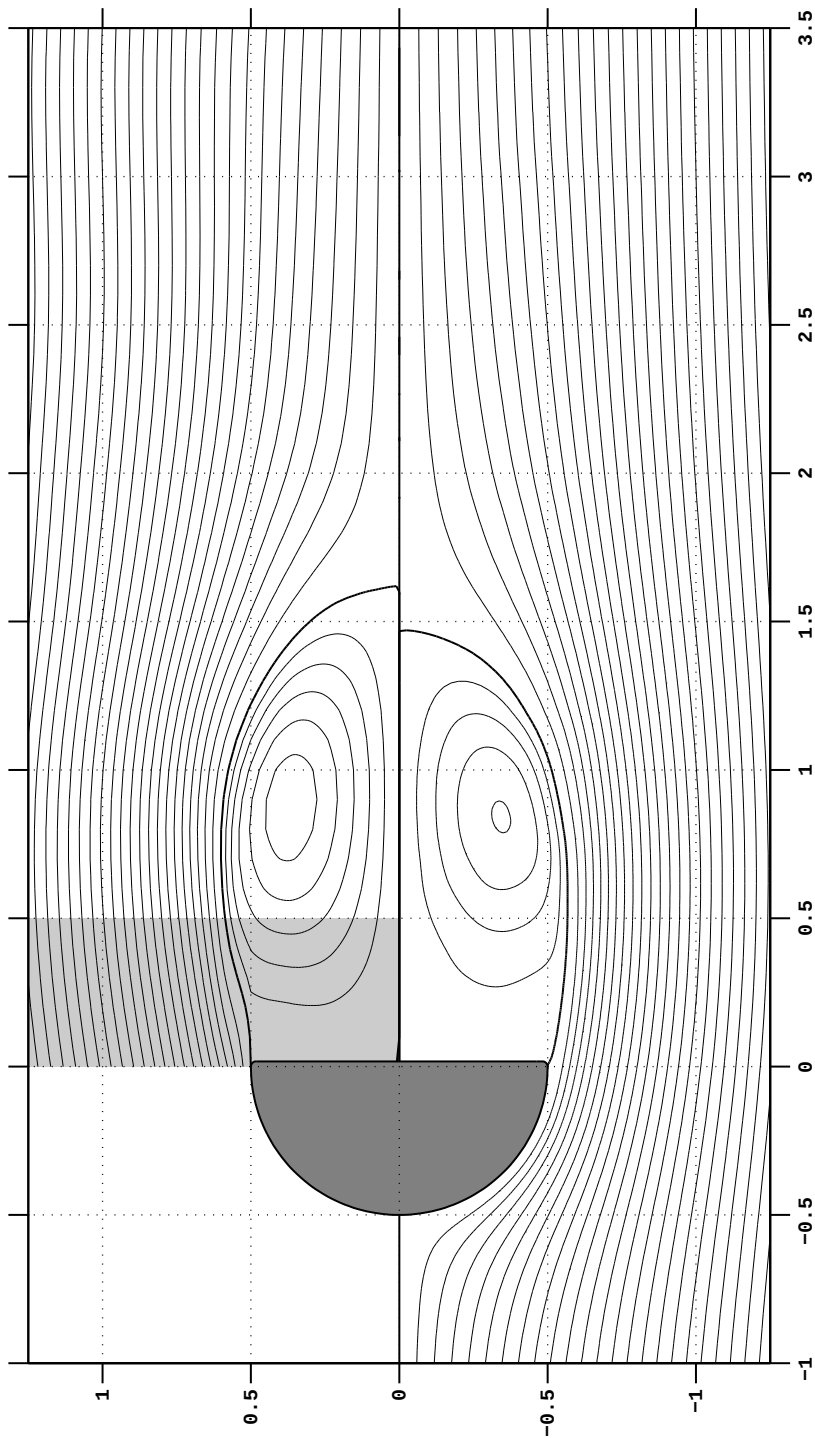
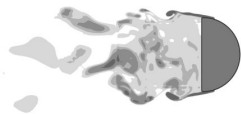


FIGURE 5.21 – Mean streamlines: numerical results at $Re = 3,000$ (bottom) versus experimental results at $Re = 55,000$ (top). The experimental streamlines were reconstructed from measured velocity profiles shown in 5.16. The streamlines in the gray zone are obtained by “extrapolation”.



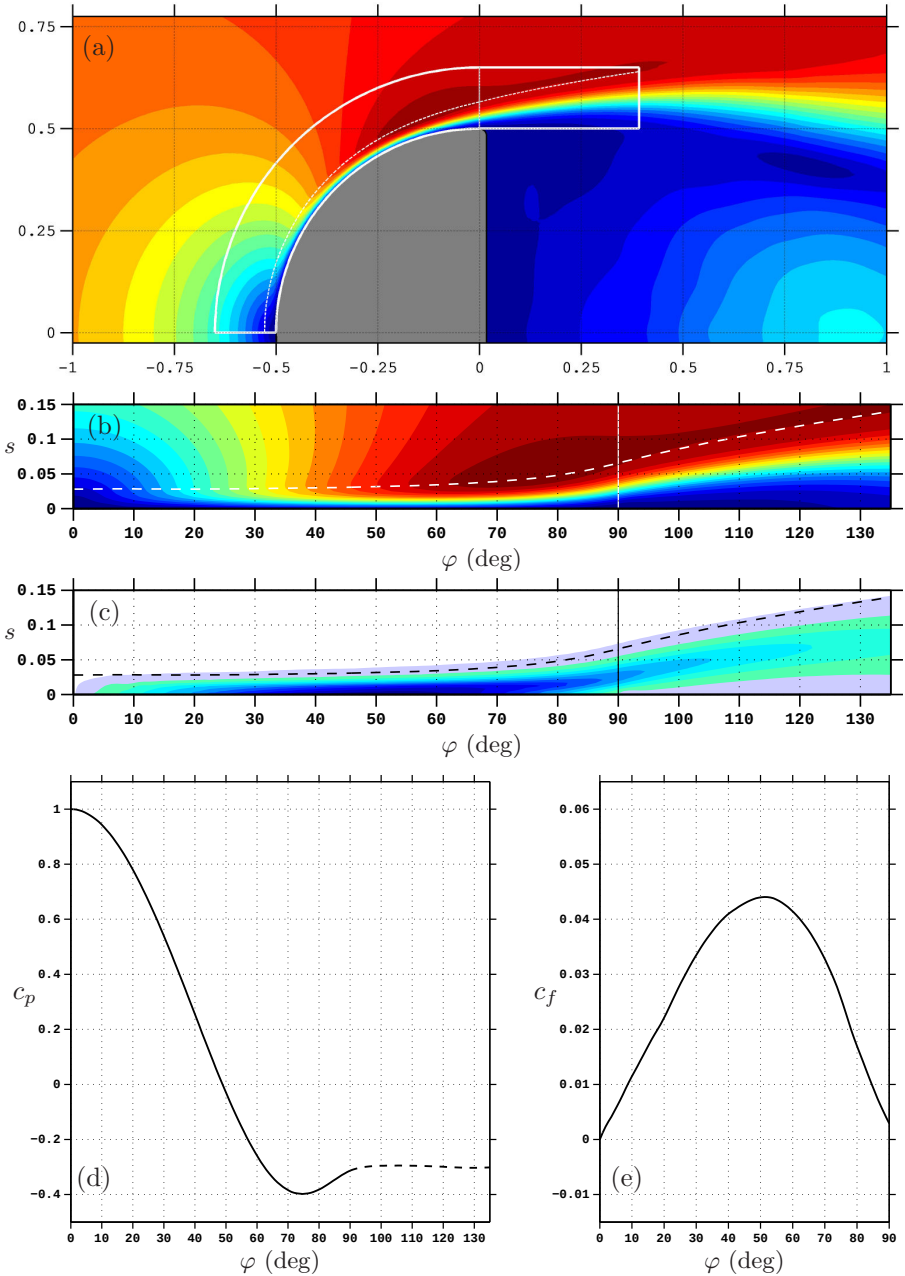


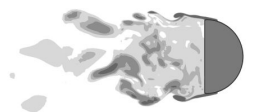
FIGURE 5.22 – (a) Mean velocity field and curvilinear region (plain). (b) Mean velocity field in curvilinear coordinate system. (c) Mean vorticity field in curvilinear coordinate system. The boundary layer thickness, $\delta_{\omega}^{.95}$ (dashed line) is also shown in (a, b, c). (d) Mean pressure coefficient. (e) Mean friction coefficient.

CHAPTER 6

LES calculations of spatially evolving wakes in ground effect

The content of this chapter was written within the FAR-Wake project as a technical report entitled “TR3.1.2-3 LES calculations of spatially evolving wakes in ground effect” [28]. Authors are G. Daeninck, R. Cocle and G. Winckelmans. This technical report is reproduced here as it was delivered to the project.

FAR-Wake, which stands for “Fundamental Research on Aircraft Wake Phenomena,” is a Specific Target Research Project commissioned by the European Union in the 6th Framework Programme (Priority 4: Aeronautics and Space). The project is the continuation of recent efforts, at the European level, to characterize, and understand aircraft wake turbulence. Aircraft in flight leave behind large-scale swirling flows (vortices) which can represent a significant hazard to following aircraft. They are therefore of great importance for practical applications concerning safety and capacity of air transport. In particular, the FAR-Wake Work-Package 3, “Wake vortex evolution near the ground,” is dedicated to the physics and modeling of wake vortex (WV) transport and decay in the region close to the ground. Due to the complex interaction with



the ground, the prediction of wake vortex transport and decay is even more challenging.

The results presented in this chapter are also a contribution to the present thesis: they show an application of vortex methods to a complex, turbulent, “real-life” problem. The 3-D numerical method used here is however different from the one used in the remainder of the thesis: it is based on the Vortex-In-Cell (VIC) approach, that combines Lagrangian and finite difference methods. It also solves the vorticity form of the incompressible Navier-Stokes equations and handles the convection term in a Lagrangian way. The time variation of the vortex particle strength, that includes both the vortex stretching and the dissipation terms, is solved on a regular grid, using 2nd order finite differences. Most importantly, the velocity field, needed for convection and stretching, is obtained by evaluating $\mathbf{u} = \nabla \times \psi$ using 2nd order finite differences: this allows to use very efficient Poisson solvers to solve $\nabla^2 \psi = -\omega$.

Efficient grid based Poisson solvers have an $\mathcal{O}(M \log M)$ computational cost (M being the number of grid points), but the constant in front is significantly smaller than that associated with the $\mathcal{O}(N \log N)$ multipole methods (N being the number of vortex particles). As the Poisson equation for the streamfunction is then solved on a grid, this requires that appropriate boundary conditions be specified on the grid surface. In usual VIC applied to open problems, the grid is taken “sufficiently” large that approximate analytical boundary conditions can be used. The better the required quality, the larger the required grid.

We here use an original approach [15, 99, 98] where the exact boundary condition is obtained dynamically using the fast multipole method. This combination is quite optimal: a compact VIC grid (enclosing tightly and dynamically the non-zero vorticity field) can be used while the proper boundary condition is enforced efficiently. Moreover, as our Parallel Fast Multipole (PFM) solver has a global view of the Poisson problem to be solved, it can be used as a domain decomposition tool: (i) the PFM solver is used to obtain the exact boundary condition on each subdomain, (ii) the stream function is then obtained independently inside each subdomain using a local and efficient Poisson solver. This VIC-PFM (Vortex-in-CellParallel Fast Multipole) approach is between 15 and 20 times faster compared to the “pure” Parallel Fast Multipole method. Since it is parallel and requires only compact grids, it can be used to handle very large problems: problems with up to 100 million grid points can be solved “easily” on 64 CPU’s.

Abstract

The case of a longitudinally developing wake generated by an elliptic wing is investigated at a moderate (yet significant) Reynolds number ($\Gamma_0^*/\nu = 10,000$), and using LES. The exact same vortex wake is also investigated in a longitudinally uniform configuration (i.e., a temporal simulation with periodic boundary conditions) using the same code and LES subgrid scale model. Finally, the 2-D (thus necessarily time developing) DNS of the same case is also performed, as a reference. For the two LES, the simulations are performed up to fully developed turbulence.

For each configuration, the averaged vorticity field, vorticity fluctuation field, and axial velocity field are presented and discussed. For the 3-D configurations, the flow topology evolution is also analyzed. Finally, the time evolution of diagnostics characterizing the wake dynamics are presented: trajectories, circulation, energy, etc.

The results show that, in the initial phase, the flow is essentially two-dimensional. During that phase, the 3-D results match very well with the 2-D reference configuration. One important feature, that is not present in the longitudinally uniform configuration, is the fact that the longitudinally developing wake induces an axial velocity defect in the primary vortices. The longitudinally developing wake therefore exhibits an earlier and more violent transition to turbulence than the longitudinally uniform configuration. The longitudinally developing configuration further differs from the uniform case by the fact that the vortex tubes describe a spiralling motion (i.e., they are not parallel). This feature is suspected to trigger some kind of vortex meandering, only observed in the longitudinally developing configuration.

6.1 Introduction

The objective of this contribution to the FAR-Wake project is to investigate the effects related to the longitudinally developing character of an aircraft wake generated in ground effect. For practical reasons (mainly computational cost and numerical complexity), one usually makes the hypothesis of a longitudinally uniform wake, thus allowing for simple periodic boundary conditions in the longitudinal direction: starting from a given (longitudinally uniform) initial vortex wake, its temporal development is investigated. In reality, however, the vortex wake generated by a wing is not longitudinally uniform: the vortex sheet



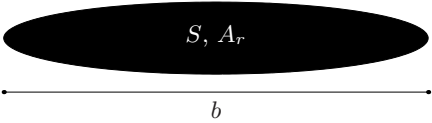
rolls up in a spiralling manner, leading to non parallel vortex lines. Furthermore longitudinally non-uniform effects arise from the presence of the wing at the origin of the vortex wake: the vortex lines are not infinite, but closed at one end, by the wing's circulation.

In order to investigate these effects, the case of a longitudinally developing wake generated by an elliptic wing is here investigated at a moderate Reynolds number and using LES. The numerical code is a combination of the Vortex-In-Cell method and the Parallel Fast Multipole method (VIC-PFM). The exact same vortex wake is investigated in a longitudinally uniform configuration (i.e., a temporal simulation with periodic boundary conditions) using the same code and LES subgrid scale model. Finally, the 2-D (thus necessarily time developing) DNS of the same case is also performed as a reference.

This document is organized as follows. In Section 6.2, we define the parameters characterizing a spatially developing wake. The 2-D and 3-D numerical codes are briefly presented in Section 6.3. The simulation parameters for the three cases (two-dimensional, longitudinally uniform and longitudinally developing) that were investigated are then presented in Section 6.4. Finally, in Section 6.5, we present and compare the simulations results.

6.2 Definition of the longitudinally developing wake

We consider the wake generated by an elliptic wing of wing span b . For a given aspect ratio A_r and wing surface S , it is characterized as follows:

$$\begin{aligned} b &= \sqrt{S A_r} \\ \bar{c} &= \sqrt{S/A_r} \\ c(y) &= \frac{4}{\pi} \bar{c} \sqrt{1 - \left(\frac{y}{b/2}\right)^2} \end{aligned}$$


where, b is the wing span, $\bar{c}$ is the mean chord and $c(y)$ is the chord distribution.

We here use $A_r = 7.5$. This value is typical of a commercial aircraft. Is is also similar to the wing configurations used in the experimental campaigns by UCL and DLR in FAR-Wake Task 3.1.2. The span loading of the elliptic wing is evaluated taking the ground effect into account using a modified “lifting line” theory, as was done in FAR-Wake deliverable D3.1.1-1 (Span loading variations

and wake roll-up in ground effect [29]). It should be noted that, unless otherwise mentioned, we define the reference quantities (circulation, vortex spacing, characteristic time, etc) based on those of the same elliptic wing out of ground effect; as was also done in [29]. Those quantities are written using a star as superscript.

The configuration is further defined by the height of the wing above the ground, h/b , and by the wing lift coefficient, $C_L = L/(\frac{1}{2}\rho U_\infty^2 S)$. The choice of these parameters are essential, as they have a direct effect on the spatial development of the wake but also on the computational requirements of the longitudinally developing simulation.

As was shown in [29], the global dynamics of a vortex sheet roll-up IGE are very similar for different initial heights above the ground: the vortex sheet itself rolls up very rapidly forming the primary vortex and, on the ground, the boundary layer (BL) separates and constitutes a secondary vortex (of opposite strength) that orbits around the primary vortex. The essential trend when the height above the ground decreases is that the dynamics are globally accelerated. More precisely, the time needed for the secondary vortex to separate and to perform one complete orbit around the primary vortex is almost directly proportional to the initial height. Thus, in a longitudinally developing configuration, the distance to perform one orbit is directly related to the initial height: the higher the location of the wing is above the ground, the longer the longitudinal distance required to perform one orbit.

We will now establish the relation between the lift coefficient C_L and the vortex wake characteristic time, $T_0 = b_0/V_0$ (with $V_0 = \frac{\Gamma_0}{2\pi b_0}$, and where Γ_0 is the total circulation of the half vortex wake and b_0 is the spacing between centroids of vorticity). We can express the lift in terms of the circulation along the wing:

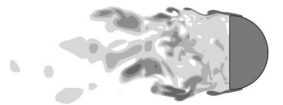
$$L = \rho U_\infty \int_{-b/2}^{b/2} \Gamma(y) dy, \quad (6.1)$$

The wing circulation, $\Gamma(y)$, is related to the wake vortex sheet by $\gamma(y) = -\frac{d\Gamma}{dy}(y)$. It can thus be related to the wake vorticity characteristics Γ_0 and b_0 :

$$\int_{-b/2}^{b/2} \Gamma(y) dy = - \int_{-b/2}^{b/2} y \frac{d\Gamma}{dy} dy = \Gamma_0 b_0. \quad (6.2)$$

Finally, we obtain the lift coefficient as:

$$C_L = 2 \frac{\Gamma_0 b_0}{U_\infty S} \quad (6.3)$$



h/b	t_{rot}/T_0	$C_L = 1.5$	$C_L = 6.0$
		d_{rot}/b	d_{rot}/b
0.125	0.6	17.1	4.26
0.25	1.4	39.8	9.94
0.5	2.9	82.4	20.6
1.0	6.0	170	42.6

TABLE 6.1 – Characteristic distance, d_{rot} , for the secondary vortex to perform one full orbit around the primary vortex, for two lift coefficients, and four initial heights.

This relation can be used to express T_0 as a function of the lift coefficient:

$$T_0 = \frac{4\pi b_0^3}{C_L U_\infty S} \simeq \left(\frac{\pi}{2}\right)^4 \frac{A_r}{C_L} \frac{b}{U_\infty} \quad (6.4)$$

The approximate equality is obtained by assuming that $b_0 \simeq \frac{\pi}{4} b$ (which is exactly true for an elliptical wing out of ground effect) and by remembering that $A_r = b^2/S$. If we now define $d_0 = U_\infty T_0$, the characteristic distance travelled by the wing for the vortex wake to descend by one vortex spacing b_0 , it appears that the wake development distance is directly related to the lift coefficient:

$$d_0 = \frac{4\pi b_0^3}{C_L S} \simeq \left(\frac{\pi}{2}\right)^4 \frac{A_r}{C_L} b \quad (6.5)$$

Based on this relation and on the results obtained in [29] for the roll-up of vortex sheets at different initial heights above the ground, Table 6.1 presents the characteristic distance for the secondary vortex to perform one full orbit around the primary vortex. Results are presented for four initial heights and for two lift coefficients. From these values, it clearly appears that, if one wants to capture a significant part of the wake dynamics (i.e., at least one full orbit), the domain length in the longitudinal direction has to be very large. In order to limit the domain size one can chose a low initial height above the ground and an artificially high lift coefficient.

Following these considerations, we now define the case that is investigated in the longitudinally developing configuration. As already mentioned and as done in [29], the wing configuration is defined with respect to the configuration out of ground effect (OGE). Using the methodology developed in [29], the span loading of a wing OGE can be obtained for the same same wing configuration IGE: due

$\frac{h}{b}$	∞	1.000	0.750	0.500	0.250	0.125
$\frac{\Gamma_0}{\Gamma_0^*}$	1.0000	1.0062	1.0107	1.0219	1.0610	1.1215
$\frac{b_0}{b_0^*}$	1.0000	0.9998	0.9995	0.9982	0.9911	0.9793
$\frac{\Gamma_0 b_0}{\Gamma_0^* b_0^*}$	1.0000	1.0061	1.0102	1.0201	1.0515	1.0983

TABLE 6.2 – Span loading characteristics for an elliptic wing IGE ($A_r = 7.5$). The span loading is characterized by Γ_0 and b_0 and is compared to Γ_0^* and b_0^* for the reference configuration: the elliptic wing (same aspect ratio and span) OGE.

to the ground, the wake characteristics are modified. Table 6.2 presents the span loading characteristics for different heights above the ground with respect to the reference configuration OGE. We chose to investigate the following case:

$$\begin{aligned}
 A_r &= 7.5 \\
 C_L &= 6.0 \\
 h/b &= 0.25 \\
 L_x/b &= 12.5
 \end{aligned}$$

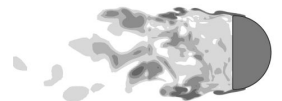
The lift coefficient, C_L is defined as the actual lift coefficient of the wing IGE. Thus, using Eq. (6.3) and remembering that $A_r = b^2/S$,

$$C_L = 2 A_r \frac{\Gamma_0 b_0}{U_\infty b^2}, \quad (6.6)$$

remembering also that for the elliptic wing OGE, $b_0^*/b = \pi/4$, we can evaluate the wing velocity, U_∞ , for this particular configuration:

$$U_\infty = \left(\frac{\pi}{2}\right)^4 \frac{A_r}{C_L} \frac{\Gamma_0 b_0}{\Gamma_0^* b_0^*} \frac{b}{T_0^*} = 8.00 \frac{b}{T_0^*} \quad (6.7)$$

The extent of the computational domain, $L_x/b = 12.5$, for the longitudinally developing configuration was chosen to be somewhat larger than $d_{\text{rot}} \simeq 10$ corresponding to this case. We can already mention that the longitudinally developing simulation of this case requires about 55 million grid points. A more realistic case, with $C_L = 1.5$ for example, would require a four times longer domain (if one wants to reach the same characteristic time in the wake) and thus a four times larger computational grid!



The Reynolds number of the flow, $Re = \Gamma_0^*/\nu$, was here set to 10,000 as it is the case that was most thoroughly investigated in [29] and is sufficiently high to exhibit a turbulent behavior.

The initial vortex sheet produced by the elliptical wing IGE is discretized as follows. Its circulation per unit length is $\gamma(y) = -\frac{d\Gamma}{dy}(y)$ with $\Gamma(y)$ the span loading as provided by the computations in [29]. This singular vortex sheet is first regularized in order to produce a regular vorticity field. For this, we use convolution with a gaussian:

$$\omega_\sigma(y, z) = \frac{1}{2\pi\sigma^2} \int_{-b/2}^{b/2} \exp\left(-\frac{(y-y')^2 - (z-h)^2}{2\sigma^2}\right) \gamma(y') dy' \quad (6.8)$$

where b is the wing span, h is the distance to the ground and σ is the regularization parameter. We here used $\sigma/b = 2 \cdot 10^{-2}$. This is “thicker” than the regularization used in the vortex sheet roll-up simulations in [29], but we are here limited by the minimum vortex sheet thickness that we are able to capture in the 3-D simulations where the grid resolution is limited. The regularized vorticity field is then discretized using particles put on multiple layers across the sheet.

6.3 Numerical codes

Vortex methods are lagrangian methods and are therefore efficient to simulate incompressible unsteady flows (we refer the interested reader to the reviews in [19, 99, 98]): they have negligible dispersion error and good energy conservation. In 2-D we use a “purely” Lagrangian approach, while, in 3-D, we use a hybrid approach, known as “Vortex-in-Cell”.

6.3.1 Two-dimensional vortex particle method

The vortex particle method solves the vorticity formulation of the incompressible Navier-Stokes equations. In 2-D, the evolution of the vorticity is given by:

$$\frac{D\omega}{Dt} = \nu \nabla^2 \omega, \quad (6.9)$$

where $\omega \mathbf{e}_z = (\nabla \times \mathbf{u})$. The vorticity field is discretized using a set of n vortex particles characterized by an area, S_p , and a circulation, $\Gamma_p = \int_{S_p} \omega dS$. The

method then follows the position of the particles and the evolution of their circulation as follows:

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}_p, \quad \frac{d\Gamma_p}{dt} = (\nu \nabla^2 \omega)_p S_p. \quad (6.10)$$

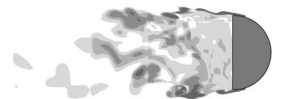
The induced velocity field is determined using a regularized Biot-Savart integral (we here use Gaussian regularization functions). The velocity is computed using a fast-multipole method which yields an $\mathcal{O}(n \log n)$ computational cost instead of $\mathcal{O}(n^2)$ if the Biot-Savart interactions were computed directly. The viscous diffusion is handled using a particle strength exchange scheme (PSE): the PSE requires an interaction of each particle with their respective neighboring particles (these are easily determined “on the fly”, together with the fast multipole velocity evaluation).

In order to be accurate, Lagrangian methods require a good overlap of the particle regularization function. Even if the particles are initially uniformly distributed, in time the particles are convected by the flow and the required particle overlap deteriorates. Therefore every few time steps, the old particle set is replaced by a new set located on a regular lattice: the particles are “re-distributed”. This is done using high order interpolation schemes.

For unbounded flow simulations, the vortex particle method can be applied “as-is”. In order to simulate the flow above a viscous ground, one obviously needs to enforce a zero velocity at the wall. By symmetrizing the vorticity field with respect to the ground (as in the previous sections), we are able to enforce exactly a no-through flow over the whole (i.e. infinite) ground plane. This however does not enforce the no-slip condition at the wall, as required. The no-slip is enforced by emitting a vorticity flux at the wall:

$$\nu \frac{\partial \omega}{\partial \mathbf{n}} = \frac{\Delta \gamma}{\Delta t}, \quad \Delta \gamma = u_{\text{slip}}, \quad (6.11)$$

where u_{slip} is the slip velocity resulting from the symmetrized vorticity field. This flux must be emitted during a time Δt . In effect, the vortex sheet $\Delta \gamma$ is distributed to neighbor vortex particles. The no-slip boundary condition is only applied to a limited region where the viscous ground interaction actually occurs.



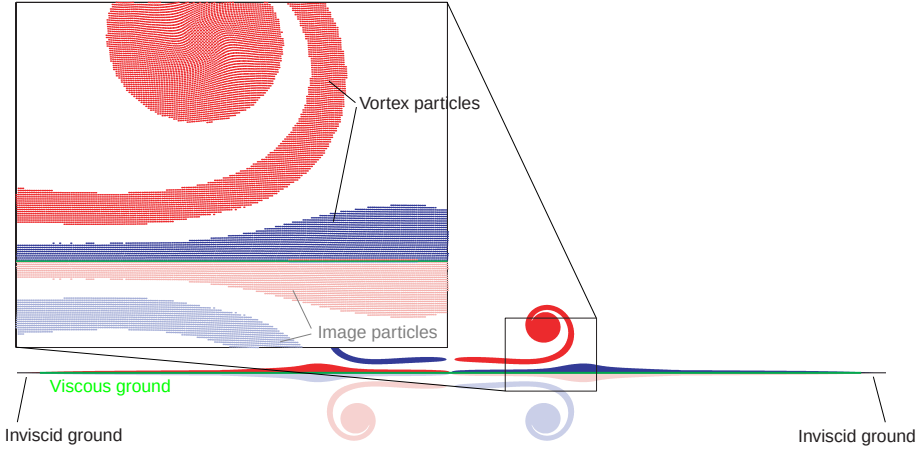


FIGURE 6.1 – **Vortex particle method IGE.** This figure shows the symmetrized vorticity field (in order to enforce no-through flow on the ground) and the extend of the viscous ground (where no-slip is enforced by imposing a vorticity flux at the wall). The zoom shows the vortex particle locations.

6.3.2 Three-dimensional combination of Vortex-in-Cell and Parallel Fast Multipole methods

The three-dimensional code (called VIC-PFM for short) is based on a different approach that combines Lagrangian and finite difference methods but also solves the vorticity form of the incompressible Navier-Stokes equations:

$$\frac{D\boldsymbol{\omega}}{Dt} = \nabla \cdot (\mathbf{u}\boldsymbol{\omega}) + \nabla^2 \boldsymbol{\omega} + \nabla \cdot (\nu_{\text{sgs}} (\nabla \boldsymbol{\omega}^s + (\nabla \boldsymbol{\omega}^s)^T)), \quad (6.12)$$

where ν_{sgs} is the effective subgrid-scale viscosity and $\boldsymbol{\omega}^s$ the “small-scale” part of the LES field. The numerical solution of equation (6.12) is sought in a two-fold manner. The convective part is evaluated using a Lagrangian approach, thus with negligible dispersion errors. The time variation of the vortex particle strength, that includes both the vortex stretching and the dissipation terms, is solved on a regular grid, using 2nd order finite differences. Interpolation from the Lagrangian particles to the Eulerian grid is done using the M'_4 scheme [19, 98]. Once on the Eulerian grid, the stream-vector $\boldsymbol{\psi}$ is evaluated by solving the Poisson equation

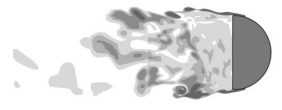
$$\nabla^2 \boldsymbol{\psi} = -\boldsymbol{\omega}. \quad (6.13)$$

The velocity field, needed for convection and stretching, is then obtained by evaluating $\mathbf{u} = \nabla \times \psi$ using again 2nd order finite differences. The global time marching procedure is carried out using a 2nd order leap Frog scheme for the convection and the Adams-Bashford scheme for the diffusion. Finally, the intrinsic divergence-free character of the vorticity vector field is imposed by a proper reprojection of the discrete vorticity field (which also requires solving a Poisson equation).

Particular to the present implementation of this procedure is the treatment of the boundary conditions on the grid when solving equation (6.13). At a given time step, the boundary conditions for ψ are determined using a Green's function approach via a Parallel Fast Multipole method (PFM). With this approach, the unbounded domain condition can be accurately approximated on a relatively small grid: only as large as the vorticity field itself. This allows for a significant reduction of the computational cost for solving equation (6.13) when compared with more classical methodologies [99, 15, 98]. Furthermore, the method can be parallelized using a domain decomposition method: the PFM code, which has a global view of the whole field, is used to obtain proper boundary conditions on each subdomain [15, 99, 98].

In order to enforce the no-through flow on the ground, the symmetrizing procedure (as described in the previous section) can be trivially extended to the 3-D case: the boundary conditions on the grid (for solving the Poisson equation) are obtained using the Green's function approach on a properly symmetrized vorticity field. The no-slip condition at the wall is enforced through a vorticity flux at the wall, as in 2-D.

The LES modeling is done using a multiscale subgrid-scale model: the model acts on the small scale part of LES field, ω^s , only. It is obtained from the complete LES field, ω , using a compact discrete filter (applied iteratively). We here use the Regularized Variational Multiscale (RVM) model [14]. The advantage of such model is that it preserves the inertial range while providing dissipation at the high wave numbers: it is only active during the complex phases of the flow, while remaining inactive during the initial phase, when the vortices are essentially laminar.



6.4 Simulation configurations

We now discuss the particular setup and computational details for the three simulations that were performed. From now on, we refer to the longitudinally developing case as the “Space Developing” case (SD) and to the longitudinally uniform case as the “Time Developing” case (TD). The two-dimensional case is noted (2D).

6.4.1 Two-dimensional setup

For the two-dimensional simulation, the Lagrangian Vortex Element Method is used, as in [29]. The spatial discretization is here sufficiently fine to perform a well resolved simulation, i.e. a DNS. The 2-D simulation can thus be used as a reference for the initial stage of the flow which is essentially two-dimensional (at least for the time developing case).

The size of the redistribution lattice, Δ , is set to $\Delta/b = 2.5 \cdot 10^{-3}$ leading to an initial number of about 6,000 particles. The time step for the simulation was set to $\Delta t/T_0^* = 5 \cdot 10^{-4}$ (where $T_0^* = b_0^*/V_0^*$, with $V_0^* = \Gamma_0^*/(2\pi b_0^*)$). The particles were redistributed every 4 time steps. By the end of the simulation, the number of particles was several hundreds of thousands.

6.4.2 Three-dimensional time developing setup

For the time developing configuration, periodic boundary conditions are used in the longitudinal direction. The domain length is $L_x/b = 4$. In the span-wise and wall-normal directions, open domain conditions are used. The initial condition is the same as the two-dimensional setup, extruded in the longitudinal direction. In order to trigger the development of three-dimensional instabilities, the initial condition was perturbed: a random displacement was imposed on the discrete particle locations. The magnitude, ϵ , of the random displacement was set to $\epsilon/b = 10^{-4}$.

The resolution of the Vortex-in-Cell grid is set to $\Delta/b = 10^{-2}$ and is uniform in all directions. The initial number of particle is about one million. The time step for the simulation was set to $\Delta t/T_0^* = 5 \cdot 10^{-4}$. The particles were redistributed every 4 time steps. The flow was computed up to $t/T_0^* = 2.0$ (4000 time steps). By the end of the simulation, the number of particles was $8 \cdot 10^6$ and the size of the Vortex-in-Cell grid was $L_x/b = 4$, $L_y/b = 4$ and

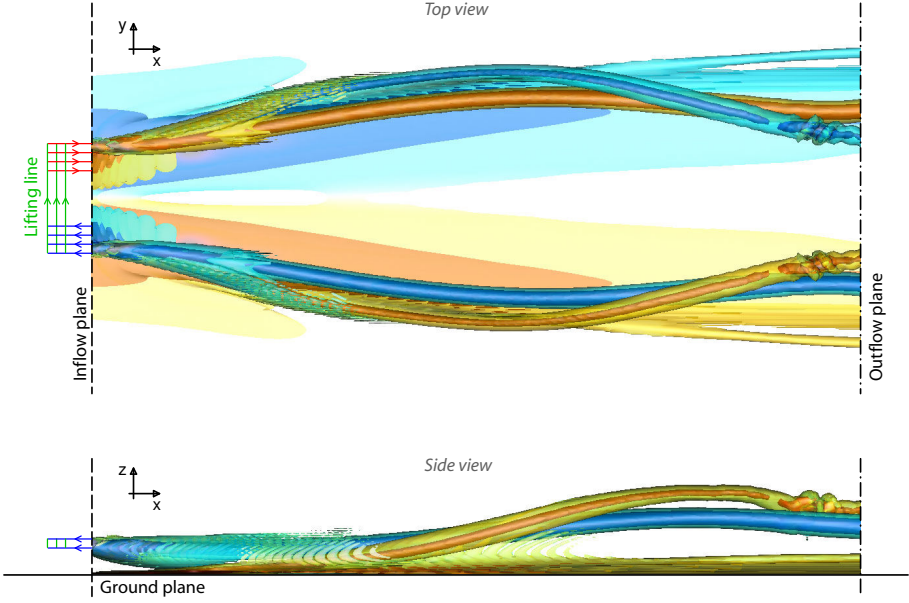


FIGURE 6.2 – Schematic of the spatially developing domain. The lifting line (green) and the initial vortex sheet (red and blue) are located next to the inflow plane at $h/b = 0.25$ above the ground. The distance between the inflow plane and the outflow plane is $L_x/b = 12.5$ (not to scale).

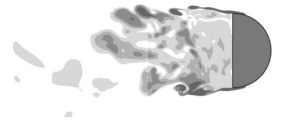
$L_z/b = 1.25$ (thus $N_x = 400$, $N_y = 400$ and $N_z = 125$ for a total of 20 million grid points).

6.4.3 Three-dimensional space developing setup

The setup of the space developing configuration is less “conventional”. It essentially consists of a lifting line (modeling the elliptical wing) shedding a vortex sheet. The wake vortex sheet then dynamically evolves in space and time, due to its own induced velocity and the velocity induced by the lifting line itself. In order to “close” the problem, an outflow boundary condition is imposed: this is done by enforcing symmetries on the flow. It leads to the following constraints on the vorticity field (here written for a y - z outflow plane located at $x = 0$):

$$\omega_n(-x, y, z) = \omega_n(x, y, z), \quad (6.14)$$

$$\omega_t(-x, y, z) = -\omega_t(x, y, z), \quad (6.15)$$



where ω_n and ω_t are respectively the normal and the tangential components of the vorticity field. This ensures that the normal component of vorticity is conserved through the outflow plane and thus allows for a natural outflow condition for wake vortices.

Figure 6.2 shows a schematic of the spatially developing domain. The effective computational domain for the Vortex-in-Cell method extends between the inflow and outflow plane ($L_x/b = 12.5$). In the transverse directions the grid size is adapted automatically to the region where the vorticity is non zero. When computing the boundary conditions on the VIC grid however, the lifting line vorticity, the inflow vortex sheet and the complete symmetrized vorticity fields (ground and outflow symmetries) are taken into account. The resulting velocity field thus corresponds to a true open domain calculation of a lifting line and its wake above the ground.

The free flow velocity, U_∞ , was set such as to obtain the lift coefficient, $C_L = 6$: $U_\infty T_0^*/b = 8$. The same resolution as the time developing calculation is used: $\Delta/b = 10^{-2}$ and it is also uniform in all directions. The time step for the simulation was set to $\Delta t/T_0^* = 5 \cdot 10^{-4}$. The particles were redistributed every 4 time steps. The number of particles was about $14 \cdot 10^6$ and the size of the Vortex-in-Cell grid was $L_x/b = 12.5$, $L_y/b = 4$ and $L_z/b = 1$ (thus $N_x = 1250$, $N_y = 400$ and $N_z = 100$ for a total of 50 million grid points).

Compared to the time developing simulations (both 2-D and 3-D), the computation has here to be carried on for sufficiently long times for the flow to reach its regime state (i.e., its statistical equilibrium). Indeed, the flow is initialized with a spatially uniform wake. A complete flow through time, $t = L_x/U_\infty$, (i.e., the time for the flow to completely traverse the computational domain) is required to “evacuate” this initial condition. A visualization of the temporal development of the space developing simulation showed that the flow had reached a well established regime at $t/T_0^* = 2.5$ (thus after about 1.6 flow through times). Once this regime state was attained, the flow had to be computed further, to attain statistical convergence of the time averaged fields of interest. The averaged fields were considered sufficiently converged at $t/T_0^* = 5$ after initialization (i.e., statistics were gathered over $2.5 T_0^*$). The complete simulation thus required 10,000 time steps.

6.5 Results

We now discuss the results obtained for the 2-D simulation (2D), the 3-D time developing simulation (TD) and the 3-D space developing simulation (SD). Instead of describing each case separately, we compare and analyze the differences between the three simulations from different point of views.

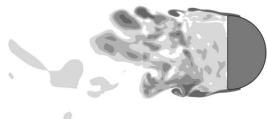
6.5.1 Global wake dynamics

The global wake dynamics for the three simulations are presented from Figures 6.3 to 6.5: vorticity contours are shown at distinctive times, $T^* = t/T_0^*$, in the right half plane. The 2D results correspond to “snap shots” of the two-dimensional vorticity field. For the TD simulations, the longitudinal component of the space averaged vorticity field is shown. In the SD simulation, the vorticity field corresponds to the longitudinal component of the time averaged vorticity field in distinctive planes at fixed distances behind the lifting line. The location of the averaging planes are shown in Fig. 6.10. In this case, T^* is defined based on the distance in the wake: $T^* = (x/U_\infty)/T_0^*$.

As can be seen from those results, the dynamics are essentially identical for all three simulations up to $T^* = 0.75$. A closer look at $T^* = 0.25$ shows that the separation occurs slightly earlier (in space and time) in the TD and SD simulations when compared to the 2D case. It should be noted that both 3-D results have a coarser grid than the 2D case: the boundary layer is thus not as well captured. This could explain the small observed difference between the 2D and TD results. The separation in the SD case occurs even slightly earlier and is related to the fact that lifting line induces an additional velocity gradient on the boundary layer, and thus enhances the boundary layer separation.

The TD case remains very close to the 2D results up to $T^* = 1.25$, indicating that the flow is essentially two-dimensional up to that time. After that time, when the secondary vortex interacts with the ground, the TD flow becomes turbulent. At $T = 2.0$, the primary wake vortex is strongly diffused and the secondary vorticity from the boundary layer has lost much of its coherence.

The SD case transitions much earlier to turbulence: at $T^* = 1.0$, it is clear that some 3-D instabilities are already present in the wake as the averaged vorticity field appears to be much more diffused than the 2D and TD cases. At $T^* = 1.5$, the maximum vorticity in the SD case does not exceed $\Omega = \omega T_0^* = 50$



while the TD case still has values above $\Omega = 200$. Note that the $T^* = 2.0$ plane is not available as the SD domain extends up to $T^* = 1.56$.

6.5.2 RMS vorticity field fluctuations

In order to further illustrate the onset of instabilities and the transition to turbulence in both 3-D cases, Figures 6.6 and 6.7 present RMS vorticity fluctuations contours (longitudinal and transverse components) for the TD case and the SD case.

Let's first mention that, for the TD case, we only show fluctuations for $T^* \geq 1.0$: before that time, the fluctuations are negligible: they are below the threshold of the chosen contours. The TD results obviously show that the transverse vorticity fluctuations are much larger than the longitudinal fluctuations. As we will see from the three-dimensional vorticity isosurfaces (Fig. 6.9), this can be explained by the fact that the instabilities generate “ Ω -loop”-like structures. The longitudinal vortex tubes are strongly bent and the resulting loops have transverse vorticity of opposite direction on each side of the loop; hence, the large transverse vorticity fluctuations. At $T^* = 2.0$, the transverse fluctuations in the outer vortex region are still at least twice larger than the longitudinal fluctuations. At that time, the largest fluctuations are observed in the primary vortex core. We note, however, that the fluctuations are high on an annulus around the vortex core; the core itself remains essentially laminar.

We now turn to the SD case. The first observation is that the fluctuation level is initially much higher than the TD case. Remember that, in the TD case, the initial perturbation level was set explicitly to a very low level. No explicit perturbation is added in the SD case; however, the discrete procedure used to shed the wake vorticity at the inflow plane introduces a perturbation (at a still fairly low level). At $T^* = 1$, the fluctuation level in the SD case is about 10 times higher than in the TD case. As will also be seen from the three-dimensional wake topology (Figs. 6.10 to 6.11), the wake also exhibits “ Ω -loop” structures and the related vorticity fluctuations are observed; they appear however earlier than in the TD case. At $T^* = 1.5$, the region where large longitudinal fluctuations are observed appear to be much larger than in the TD case, even when compared with TD at $T^* = 2.0$. In comparison to the TD case, the SD case exhibits a more “homogeneous” turbulence: the order of magnitude of the fluctuations in the longitudinal and transverse directions are similar in the SD case. The large high longitudinal fluctuation region is further

explained by the observed vortex “meandering” in the SD simulation: this can be seen in the three-dimensional wake visualizations (Figs. 6.10 to 6.15). Thus, in the averaged view, the primary vortex appears to be overly diffused. An instantaneous snapshot would reveal a turbulent, but still intense, vortex.

6.5.3 Axial flow component

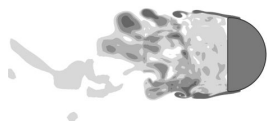
One would be tempted to conclude from the vorticity fluctuation results that the essential differences between the time and the space developing flow configurations are related to the initial level of perturbation. While it is clear that a higher level of perturbation in the TD case would trigger the transition to turbulence earlier, there are other differences in the type of instabilities that cannot be explained solely by this effect.

The essential feature of the spatial developing flow, that was not discussed so far, is the presence of an axial flow component in the wake vortices. The fact that the roll-up of the vortex sheet occurs in a spiraling manner (instead of parallel vortex lines) introduces a significant tangential vorticity component in the rolled-up vortices. The axial flow introduces necessarily a velocity deficit in the rolled-up wake vortices. The result can be seen as the combination of a vortex and an axial jet. The time averaged axial velocity for the SD case is shown in Fig. 6.8. It can be seen that the axial velocity reaches values as low as $-4b/T_0^*$.

The parameter that determines the type of instabilities that occur in a vortex-jet configuration is related to the swirl number, the ratio between the maximum axial velocity and the maximum tangential velocity: $S = u_{\theta, \max}/u_{x, \max}$. Based on the tangential velocity profiles from Fig. 6.21, $S = 1.8$ at $T^* = 0.5$ and $S = 0.87$ at $T^* = 1.0$. The low values of the swirl number indicate that the vortices have a strong jet component and will thus be more unstable than in the TD configuration (that has no axial jet).

The presence of the ground and the separation of a secondary vortex there further complicates the picture. At $T = 0.5$, it can be seen that the secondary vortex was formed together with a (less intense) positive axial jet. The swirl number for the secondary vortex is: $S = 2.3$ at $T^* = 0.5$ (based on the tangential velocity profile from Fig. 6.22).

We also note that, the strength of the axial velocity component is directly related to the lift coefficient: a “real” configuration (i.e., $C_L = 1.5$) will exhibit



an axial velocity component about 4 times lower than the current configuration ($C_L = 6$). The swirl number will be increased in the same proportion, thus reducing the gap between the temporal and spatial simulations.

6.5.4 Three-dimensional wake topology

Figure 6.9 shows the three-dimensional evolution of the vorticity field for the TD simulation: each frame shows two isosurfaces of vorticity norm, colored by the longitudinal vorticity component. At $T^* = 1$, short wave instabilities can be seen in the core region of the secondary vortex. As it interacts with the ground at $T^* = 1.25$, the instabilities strongly increase and, at $T^* = 1.5$, “ Ω -loop”-like structures have formed and wrap themselves around the primary vortex. While vorticity continues to emanate from the boundary layer, it is continuously tilted and forms complicated loops that also wrap around the primary vortex. This corresponds to the phenomenon described in Section 6.5.2 about the transverse fluctuations in the TD case. By $T^* = 2.0$, the turbulence is fully developed (see also Fig. 6.25).

Figure 6.10 shows a snapshot, from three different viewpoints (top, side and perspective), of two isosurfaces of vorticity norm (same as for the TD case) when the SD case has reached its regime: one can see the global spiraling motion of the wake vortices. The three-dimensional dynamics are clearly quite different compared to the TD case. The secondary vortices interact with the primary vortex before having performed a complete revolution around it. The interaction is initiated by a medium wave instability that propagates on the secondary vortex. This is better illustrated by the time sequence shown in Fig. 6.11. One would even be tempted to describe this instability on the secondary vortex as some kind of vortex “meandering”. Figures 6.12 and 6.13 give a closer look (top and side view) at the instability on the secondary vortex and its early interaction with the primary vortex (zoom on region $x/b = [4, 8]$ thus, $T^* = [0.5, 1]$). One can see how the medium wave instability periodically interacts with the primary vortex. It also shows that short wave instabilities with a double helical structure are present. Figures 6.14 and 6.15 consider the more distant wake (region $x/b = [8, 12]$ thus, $T^* = [1, 1.5]$). It shows how the secondary vortex partially reconnects with the primary through a very complex spiraling motion. Due to this interaction, the primary vortex appears to “burst” into two or even three helical vortices which then further “break down” into small-scale turbulence.

6.5.5 Time evolution of diagnostics

Let us now look into the time evolution of some diagnostics characterizing the wake dynamics up IGE for the three cases investigated.

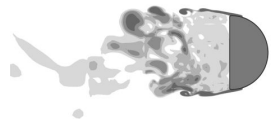
We first define the center of a vortex. It is obtained by fitting a vorticity profile model (here a low order algebraic function) on the averaged vorticity field in a least mean square sense. The vortex center is then defined as the center of the fitted function. This allows to obtain a good definition which is insensitive to local fluctuations in the averaged field when the flow has become turbulent. When the flow is still laminar this essentially comes down to defining to vorticity center based on the local extremum of vorticity (as is usually done).

Figure 6.16 shows the trajectories of the primary vortex center in the right-half plane for the three investigated cases. The initial trajectories of the 2D and TD cases are identical in the early roll-up phase. The SD trajectory is slightly lower, probably due to the downwash velocity induced by the lifting line itself in the near wake region. During the rebound, the TD trajectory is somewhat shifted inwards (compared to the 2D case): this is explained by the earlier separation of the boundary layer (as already mentioned in Section 6.5.1). The rebound loop in the SD case is completed earlier compared to the 2D and TD cases. However, as the SD case becomes turbulent immediately afterwards, the motion of the primary vortex is “decelerated” and, at $T^* = 1.5$, the primary vortex center locations are very close for all cases. The 2D and TD trajectories are shown up to $T^* = 2.0$, by that time the 2D case has clearly separated from the TD trajectory due to the absence of three-dimensional effects.

Based on the vorticity center, we can define the vortex circulation profile. It is determined using

$$\Gamma(r) = \int_0^{2\pi} \int_0^r \omega_x(r', \theta) r' dr' d\theta \quad (6.16)$$

where the integration radius is taken from the vortex center, and the vorticity is respectively the instantaneous, the space averaged and the time averaged vorticity field for the 2D, TD and SD cases. Based on the circulation profile, one usually defines the Γ_{5-15} : the average circulation contained between 5 and 15 meter for a wake generated by a 60 meter wing spang (thus between $1/12b$ and $1/4b$). In our case however, this quantity does not have much sense as the initial height of the wake above the ground is $b/4$ (the Γ_{5-15} quantity immediately includes opposite sign vorticity from the boundary layer



and the secondary vortex). Therefore, we here define $\Gamma_{\max}$: the maximum of the circulation profile. This quantity is a good indicator of the total circulation contained in the primary vortex. Figure 6.17 shows the “maximum” circulation, $\Gamma_{\max}$, for the primary vortex as a function of time. The evolution of $\Gamma_{\max}$ for the 2D and TD cases are essentially identical up to $T^* = 1.2$, as the TD case remains two-dimensional until then. Then, when the TD flow becomes three-dimensional at $T^* = 1.4$, the circulation decreases rapidly. The SD case is only slightly different up to $T^* = 0.8$; past that time, the rapid circulation decrease occurs. In terms of the spatial location in the wake, this corresponds to $x/b = 6.4$: referring to Figs. 6.12 and 6.13, one can see that this corresponds to the first location where the secondary vortex periodically interacts with the primary vortex.

The circulation profile, $\Gamma(r)$, can also be used to evaluate the vortex induced tangential velocity profile,

$$u_\theta(r) = \Gamma(r)/(2\pi r). \quad (6.17)$$

The maximum tangential velocity, $u_{\theta,\max}$, is determined from this profile, along with the “effective vortex core radius”, r_c : the radius where the maximum occurs. Figures 6.18 and 6.19 show the temporal evolution of those quantities. Interestingly, $u_{\theta,\max}$ and r_c for the TD case remain close to the 2D case for a longer time than it was the case for $\Gamma_{\max}$. This might indicate that, while the total circulation of the primary vortex decreases due to three-dimensional effects and turbulence, the inner vortex core remains unaffected (at least for some time) indicating a laminar vortex core. The picture for the SD case is dramatically different: at $T^* = 0.8$ (here perfectly in sync with the decrease of $\Gamma_{\max}$), $u_{\theta,\max}$ decreases rapidly while r_c increases accordingly. At $T^* = 1.5$, $u_{\theta,\max}$ is about 2.5 times lower than 2D and TD cases and r_c is about twice larger. This indicates that the primary vortex is very rapidly destroyed in the SD case: at $T^* = 1.0$, the vortex core is already strongly “diffused” according to this picture. While, the additional instabilities due to the presence of the velocity deficit certainly play an important role here, we think that this effect is combined with another phenomenon: some kind of vortex meandering that induces oscillatory motions on the vortex center. The average vortex core therefore looks more diffused than it really is.

Figures 6.20 and 6.21 further illustrate the previous comments by showing respectively the circulation profiles, $\Gamma(r)$, and the tangential velocity profiles, $u_\theta(r)$, at different times for all cases. Figure 6.22 shows $\Gamma(r)$ and $u_\theta(r)$ at

$T^* = 0.5$ for the secondary vortex in the SD case: these results (with the axial velocity results in Fig. 6.8) can be used to determine the swirl number for the secondary vortex at that time.

Another interesting view of the vortex system is obtained when considering the energy of the averaged velocity fields (either longitudinally space averaged in the TD case or time averaged in the SD case). The time evolution of the cross-plane kinetic energy (i.e., only the transverse components of the velocity) is shown in Fig. 6.23. The results are very much in line with the previous observation. It is important to notice that the slope of the rapid energy decay region for the SD case is much steeper than the TD case. Again, these results show that the SD case transitions to turbulence earlier but also that the energy decay mechanism is significantly enhanced.

Finally, Figs. 6.24 and 6.25 presents a detailed view of the energy evolution for the TD case. In particular, Fig. 6.25 shows the modal energy spectrum at different times. At $T^* = 1$, the spectrum shows that there is a competition between medium wave instabilities and short wave instabilities. Later, the flow becomes fully turbulent and the spectrum exhibits the typical large- to small-scale energy cascade.

6.6 Conclusion

We have presented all the simulation results for the three investigated configurations of “one case”: the roll-up and evolution of the wake generated by an elliptical wing IGE. The three configurations differ by the number of physical features they capture. The simplest configuration is that of a purely 2-D simulation IGE: its use is clearly limited, but it allowed to have an accurate DNS reference. The three-dimensional longitudinally uniform configuration is able to account for the three-dimensional instabilities and the turbulent evolution of the wake, but it makes the hypothesis of a globally parallel vortex wake. In order to take into account the fact that the vortex sheet actually rolls up in a spiralling manner, it is necessary to model the wake as it is produced in reality: in a spatially developing (SD) configuration. This was done in the third configuration. The objective of this study was to investigate the effects related to the longitudinally developing character of the wake by comparing the results from the different configurations.

The following observations were made:



- During the initial phase, when the flow is essentially two-dimensional, the three-dimensional results (TD and SD cases) match very well with the 2D reference case. The SD case shows greater differences than the TD case but this is logical as additional three-dimensional effects are present, even before three-dimensional instabilities appear.
- All the simulation results show that the SD case transitions earlier to turbulence than the TD case. This can be partly related to the fact that the initial perturbation level in the TD case is lower than in the SD case.
- The growth rate of the instabilities in the SD case also appears to be much larger than for the TD case. One important feature, that is not present in the TD simulation, is the fact that the spatially developing wake induces an axial velocity defect in the primary vortices. The resulting wake vortices can thus be seen as a combined vortex-jet configuration, which exhibits additional instabilities. The strength of the velocity defect is furthermore directly related to the wing lift coefficient, which was set to a large value ($C_L = 6$) due to computational resources limitations. In a more realistic configuration, the axial velocity component would thus be lower, which in turn would reduce the strong instabilities, bringing the SD results closer to the TD results.

One way to investigate if the differences between the TD and SD configurations are related to the axial velocity component would be to investigate a longitudinally uniform configuration with a comparable axial “jet” superposed on the initial vortex. This configuration would certainly be closer to the space developing configuration. It would however not take into account features such as the positive axial velocity that appears in the secondary vortices and, most importantly, the fact that the vortex lines are not parallel in reality. From the SD simulation, we suspect that the observed meandering motion of the primary and secondary vortices might be related to the fact that the vortices describe a spiralling motion.

To conclude, we think the present results show that longitudinally uniform wake simulations (TD) are certainly a valid tool to investigate wake vortices. While our results showed sometimes large differences between the space and time developing configurations, one must remember that the investigated space developing case is somewhat extreme (due to its close distance to the ground and its very high lift coefficient). Furthermore, one could improve the TD simulations by introducing some axial flow. However, it also appears that

some features that can only be taken into account in a spatially developing configuration have an important effect.

The next step would therefore be to perform a space developing simulation in a more realistic configuration (i.e., a lower lift coefficient). This would greatly increase the computational demands. The VIC-PFM code, in its present state, would be able to run such a simulation: it is a matter of running a four times larger domain using four times more processors. It is also a matter of handling four times larger files and of analyzing the results: something quite demanding! ... yet doable.

Finally, it should be noted that the space developing configuration is still very different from real aircraft wakes as we here consider a “traced wing” rather than a self-propulsive configuration. In reality, beside the velocity defects observed here, velocity excess regions exist (from the jets) as the total jet impulse has to compensate the induced and profile drag. In this case, where jets and wakes interact with vortices, there are many more possibilities for interaction between streamwise and transverse vorticity and associated instabilities.



2-D Time developing

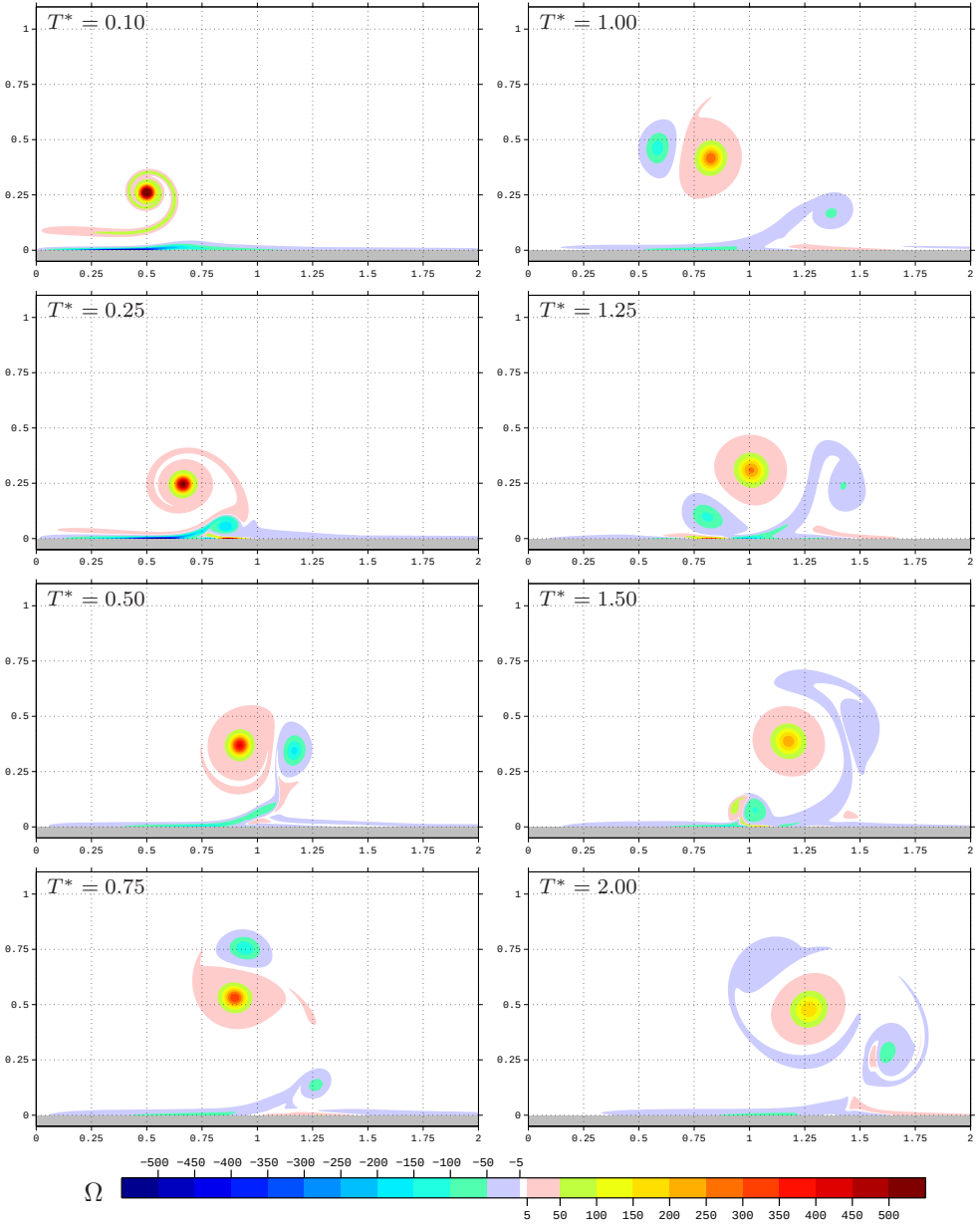


FIGURE 6.3 – 2-D Time developing DNS. Vorticity contours are shown for the right half plane at distinctive times: $T^* = \frac{t}{T_0^*}$ and $\Omega = \omega T_0^*$.

3-D Time developing

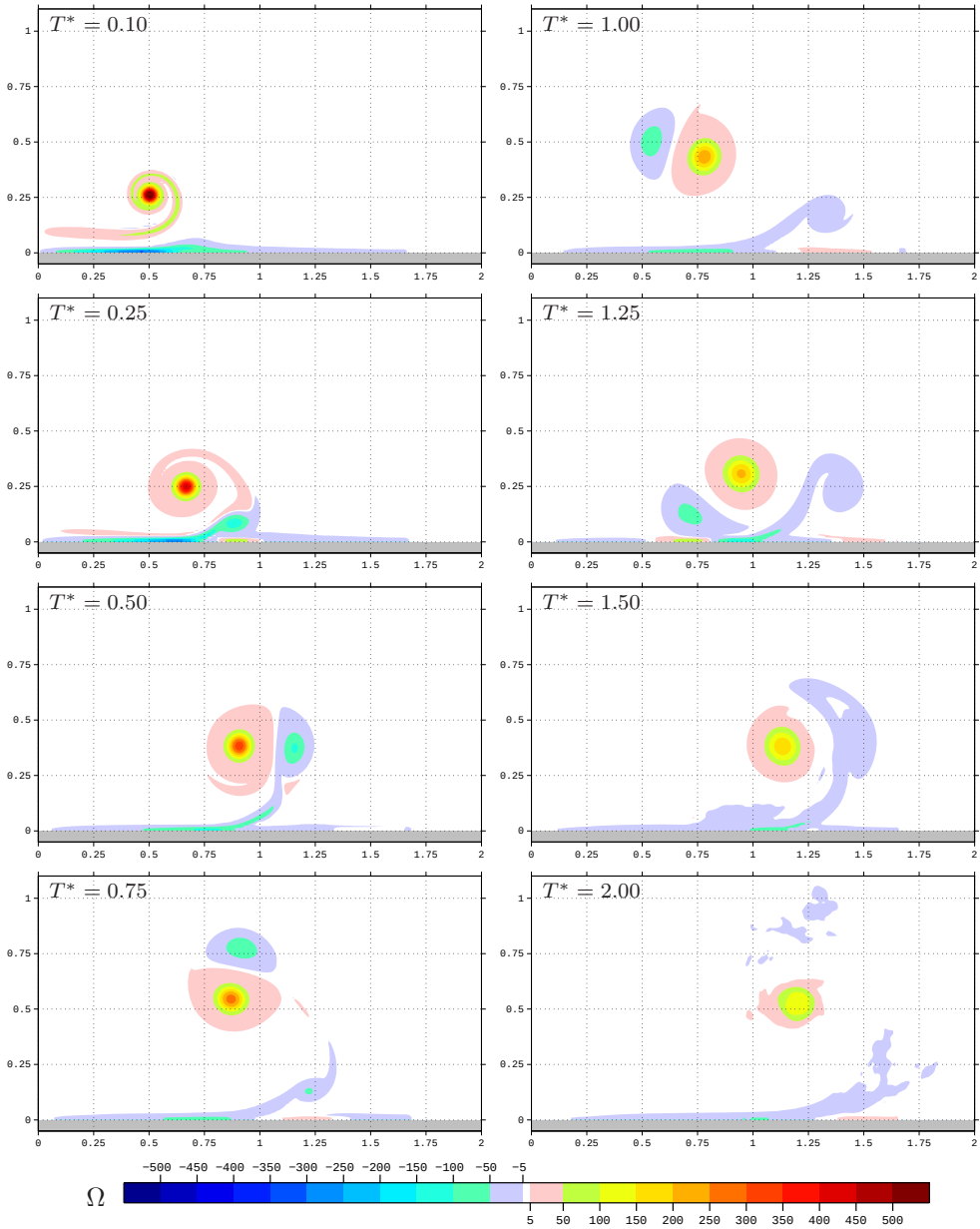
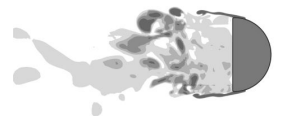


FIGURE 6.4 – 3-D Time developing LES. Space averaged vorticity contours are shown for the right half plane at distinctive times: $T^* = \frac{t}{T_0^*}$ and $\Omega = \omega T_0^*$.



3-D Space Developing

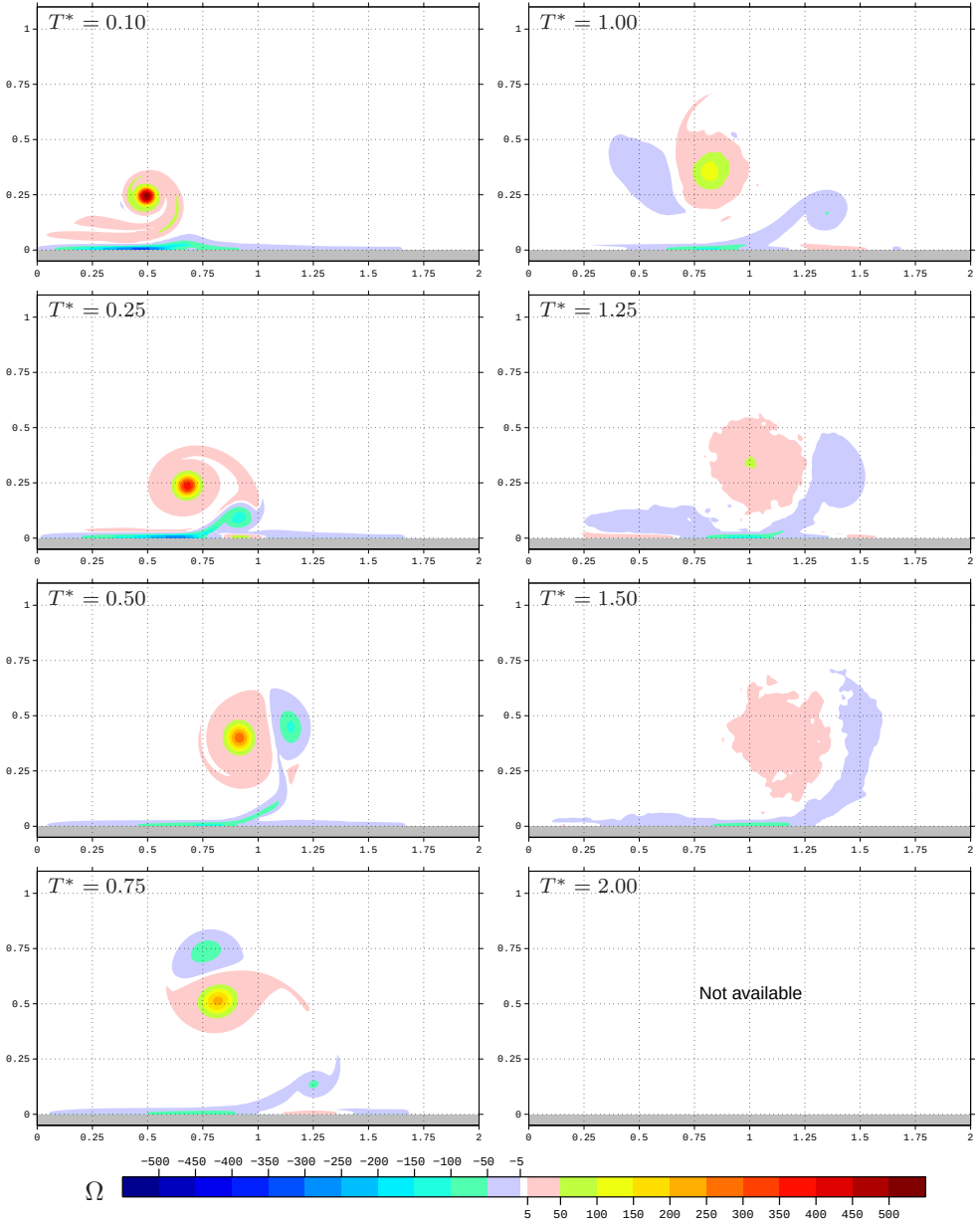


FIGURE 6.5 – 3-D Space developing LES. Time averaged vorticity contours are shown for the right half plane in distinctive slices: $T^* = \frac{x}{U_\infty} \frac{1}{T_0^*}$ and $\Omega = \omega T_0^*$.

3-D Time developing

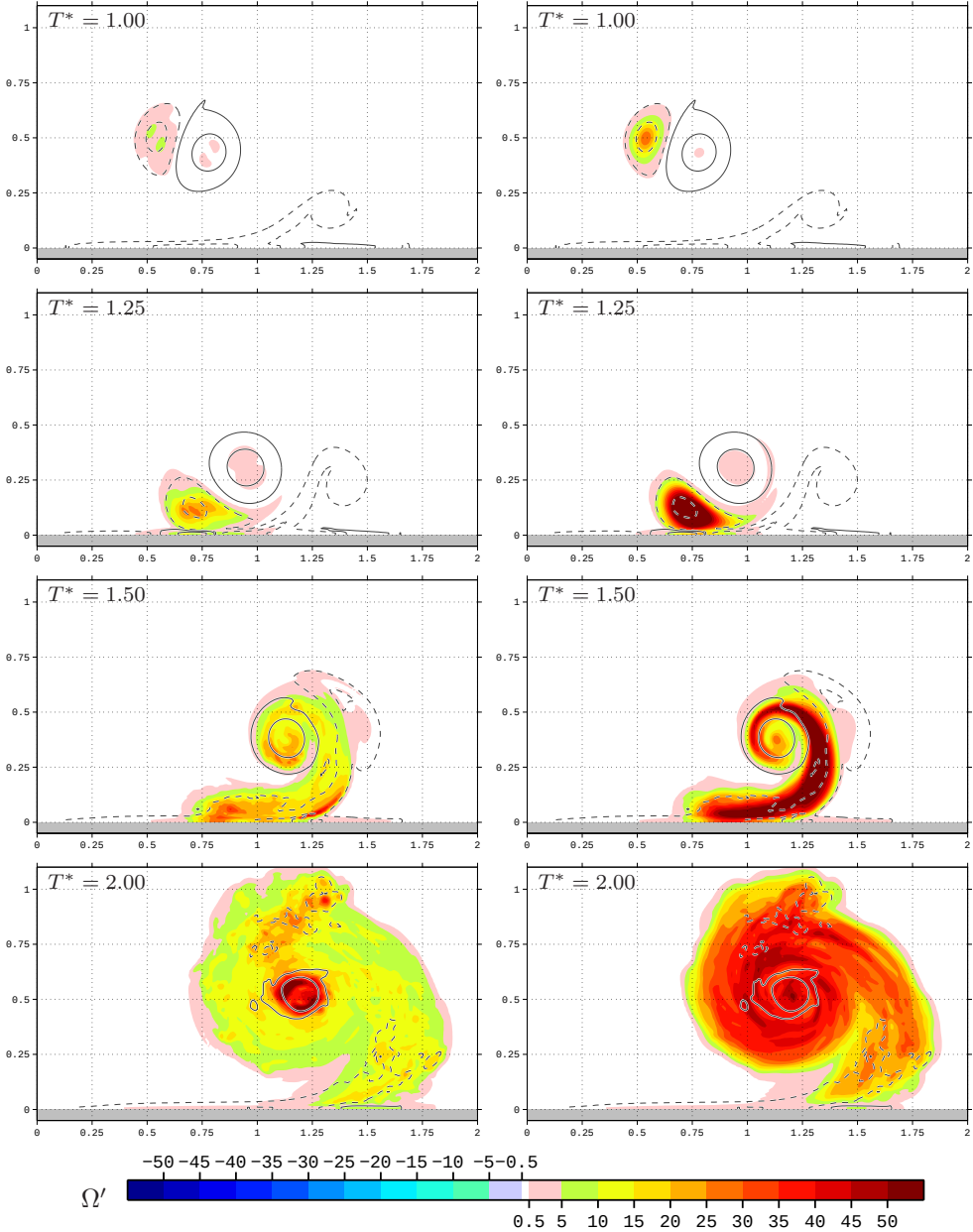
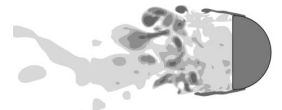


FIGURE 6.6 – 3-D Time developing LES. Space averaged fluctuations of longitudinal vorticity, $\sqrt{\omega_x'^2}$, (left) and of transverse vorticity, $\sqrt{\omega_y'^2 + \omega_z'^2}$, (right) at distinctive times: $T^* = \frac{t}{T_0^*}$ and $\Omega' = \omega' T_0^*$. Two positive (plain) and negative (dashed) contours of averaged longitudinal vorticity are also shown.



3-D Space developing

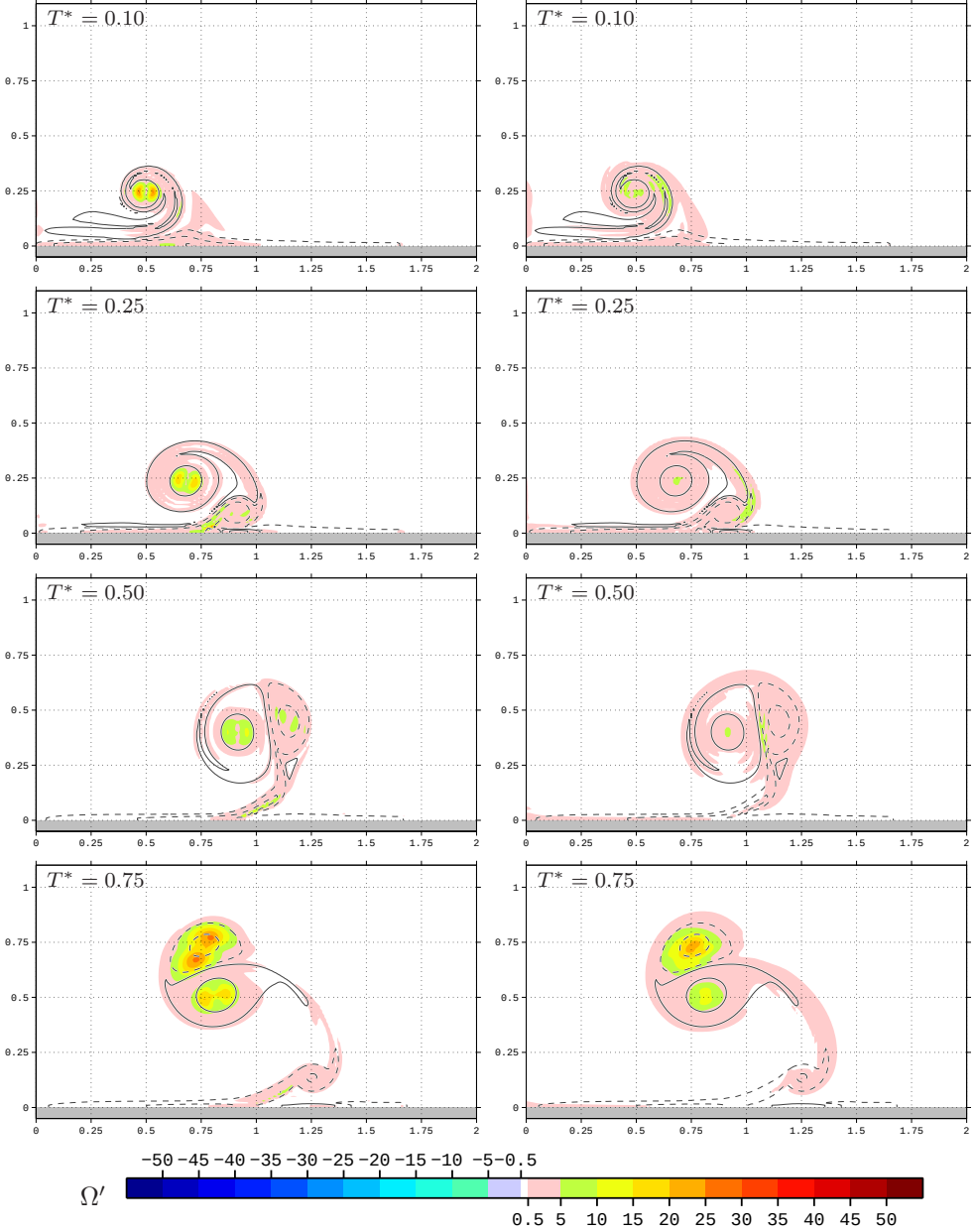


FIGURE 6.7 – 3-D Space developing LES. Time averaged fluctuations of longitudinal vorticity, $\sqrt{\omega_x'^2}$, (left) and of transverse vorticity, $\sqrt{\omega_y'^2 + \omega_z'^2}$, (right) in distinctive slices: $T^* = \frac{x}{U_\infty} \frac{1}{T_0^*}$ and $\Omega' = \omega' T_0^*$. Two positive (plain) and negative (dashed) contours of averaged longitudinal vorticity are also shown.

3-D Space developing

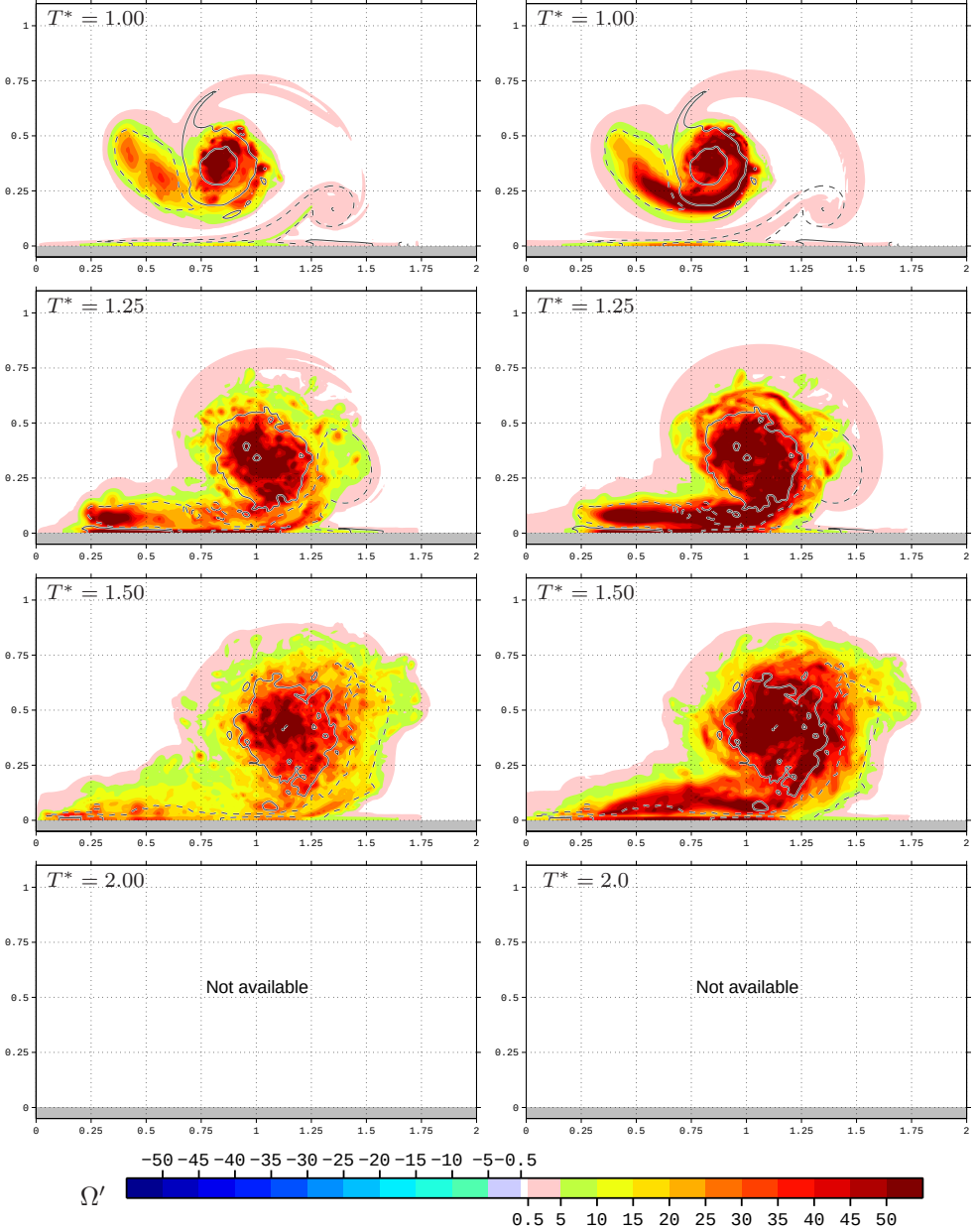
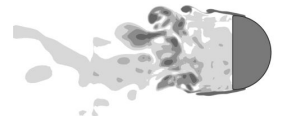


FIGURE 6.7 – (cont.) 3-D Space developing LES. Time averaged fluctuations of longitudinal vorticity, $\sqrt{\omega_x'^2}$, (left) and of transverse vorticity, $\sqrt{\omega_y'^2 + \omega_z'^2}$, (right) in distinctive slices: $T^* = \frac{x}{U_\infty} \frac{1}{T_0^*}$ and $\Omega' = \omega' T_0^*$. Two positive (plain) and negative (dashed) contours of averaged longitudinal vorticity are also shown.



3-D Space Developing

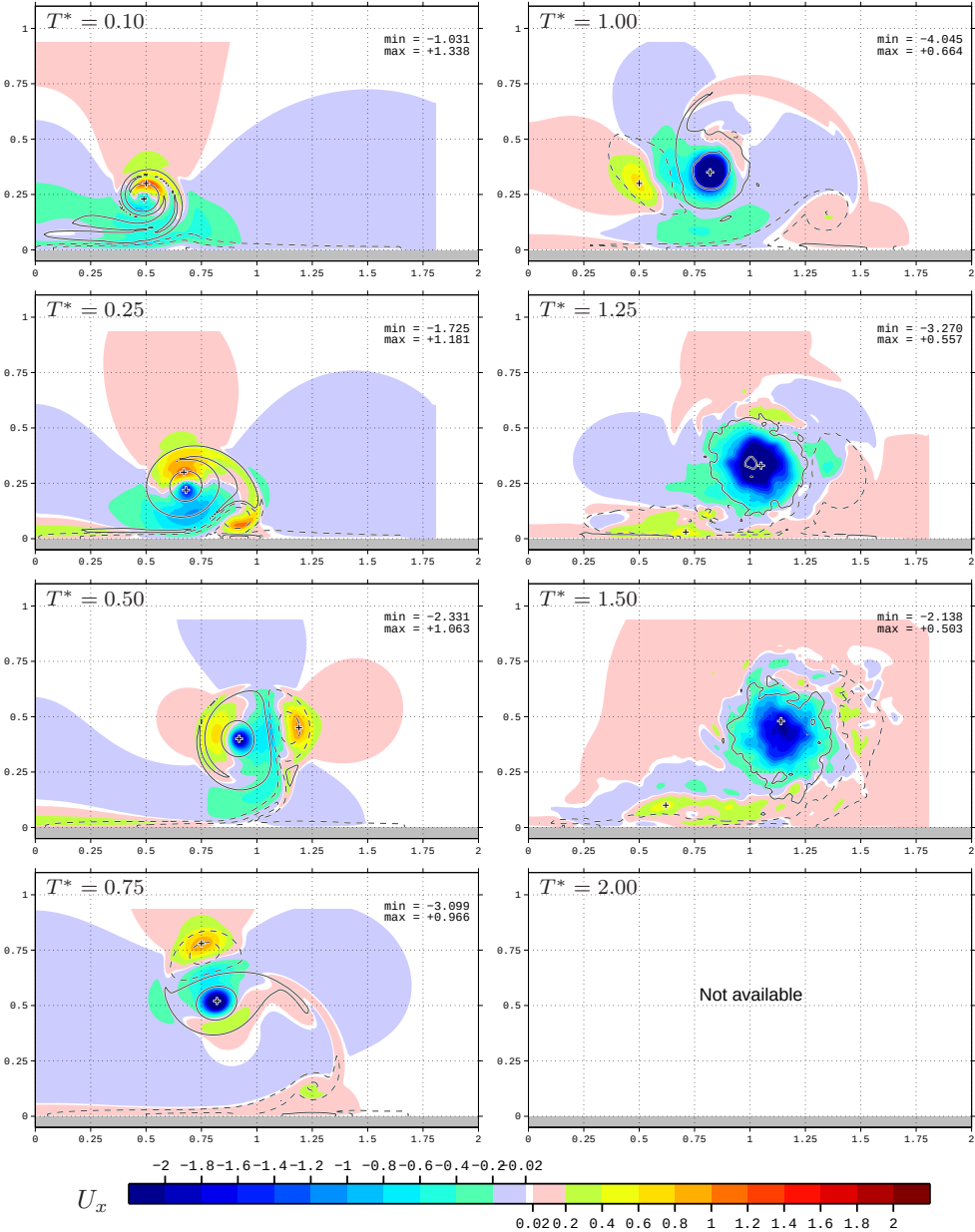


FIGURE 6.8 – 3-D Space developing LES. Time averaged axial velocity contours are shown for the right half plane in distinctive slices: $T^* = \frac{x}{U_\infty} \frac{1}{T_0^*}$ and $U_x = u_x \frac{T_0^*}{b}$.

3-D Time developing

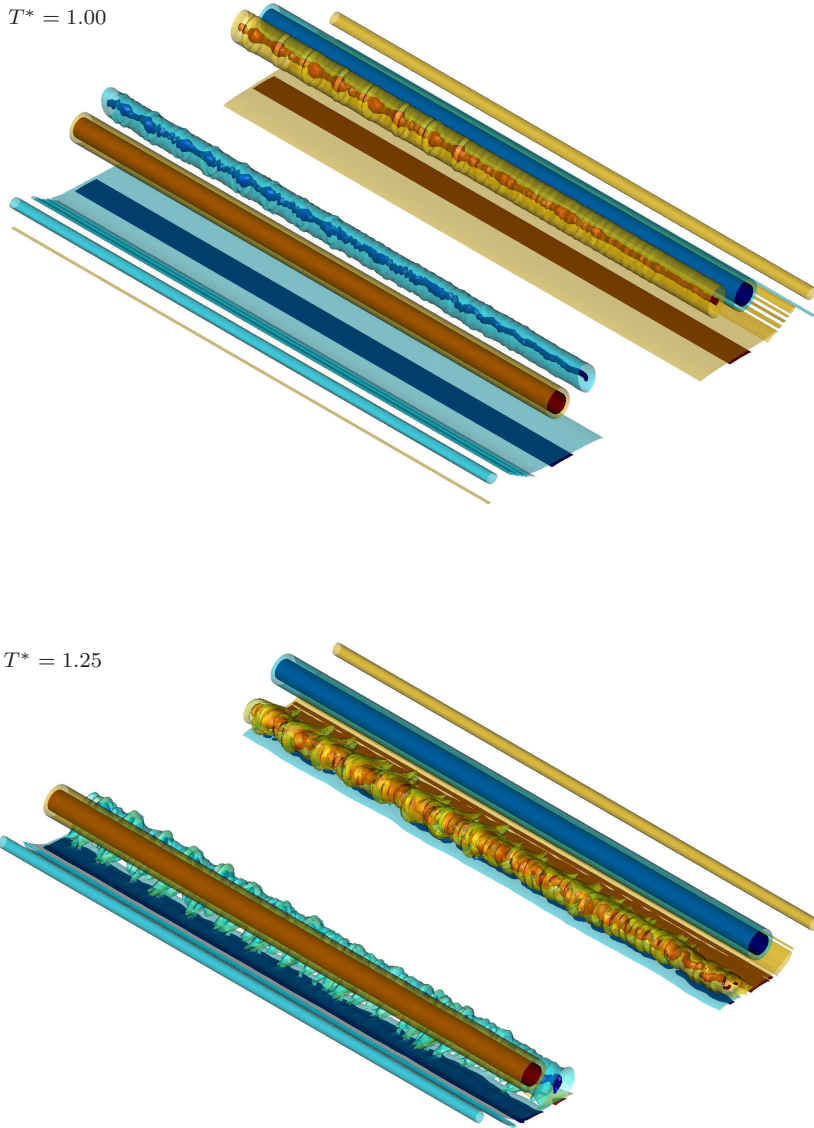
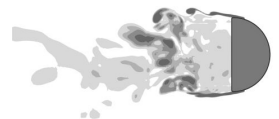


FIGURE 6.9 – 3-D Time developing LES. Isosurfaces of vorticity norm, $||\Omega|| = 80$ (opaque) and 25 (translucent), colored by the longitudinal component of vorticity at distinctive times.



3-D Time developing

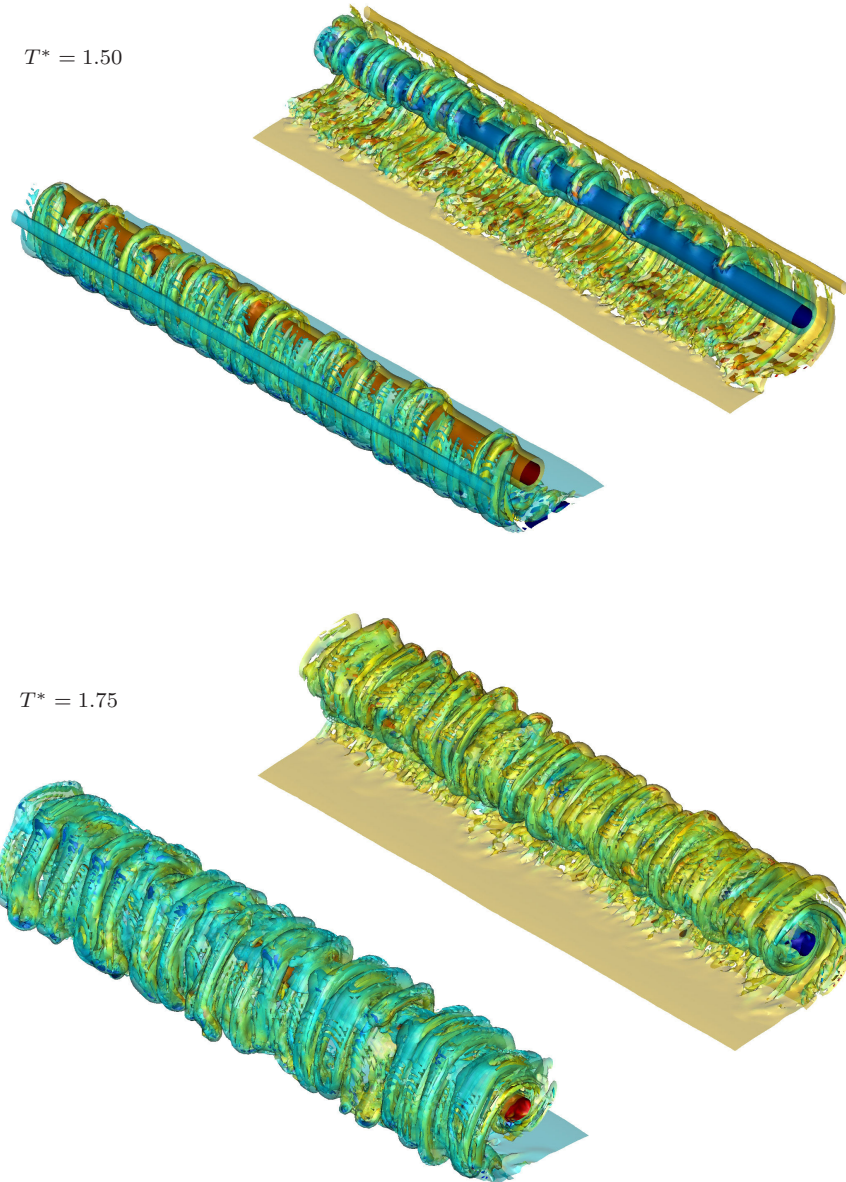


FIGURE 6.9 – (cont.) 3-D Time developing LES. Isosurfaces of vorticity norm, $||\Omega|| = 80$ (opaque) and 25 (translucent), colored by the longitudinal component of vorticity at distinctive times.

3-D Time developing

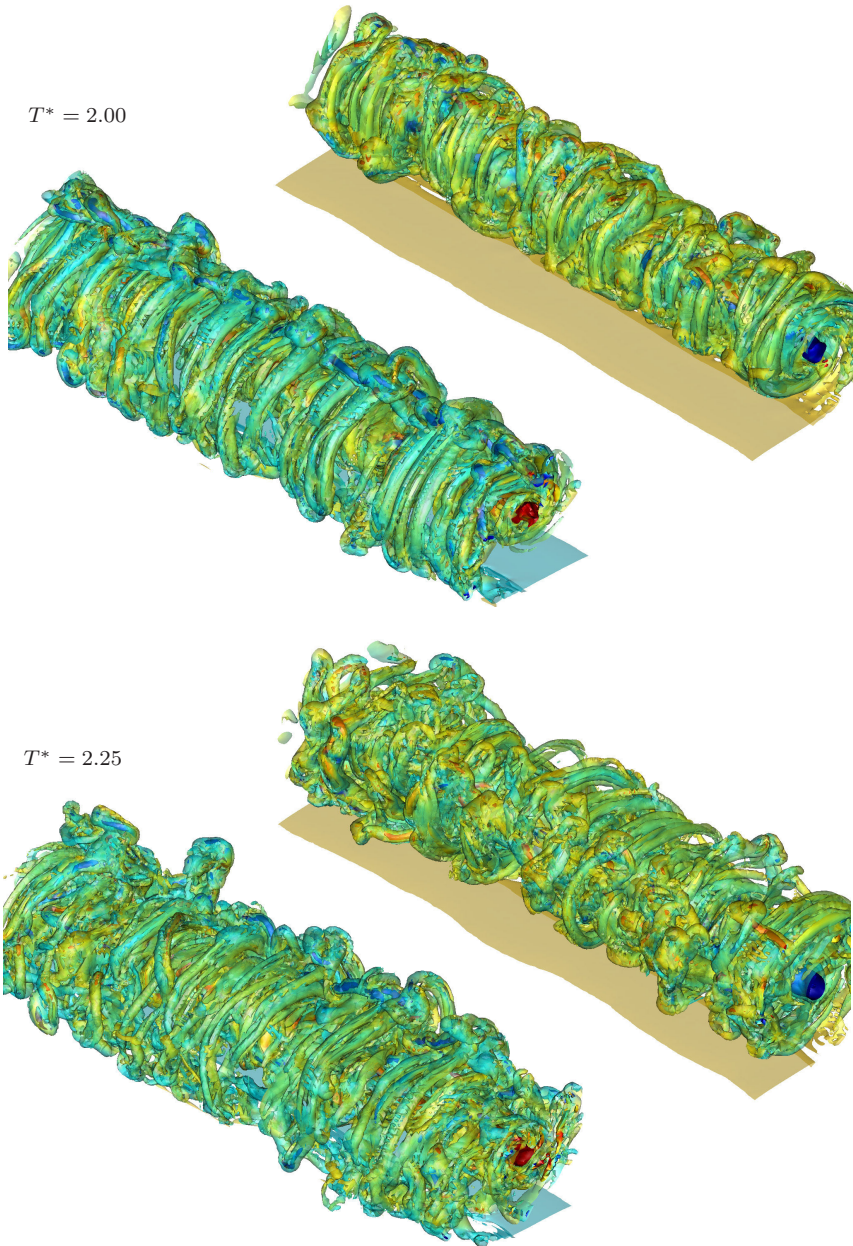
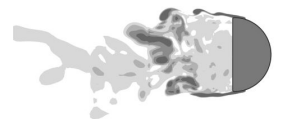


FIGURE 6.9 – (cont.) 3-D Time developing LES. Isosurfaces of vorticity norm, $||\Omega|| = 80$ (opaque) and 25 (translucent), colored by the longitudinal component of vorticity at distinctive times.



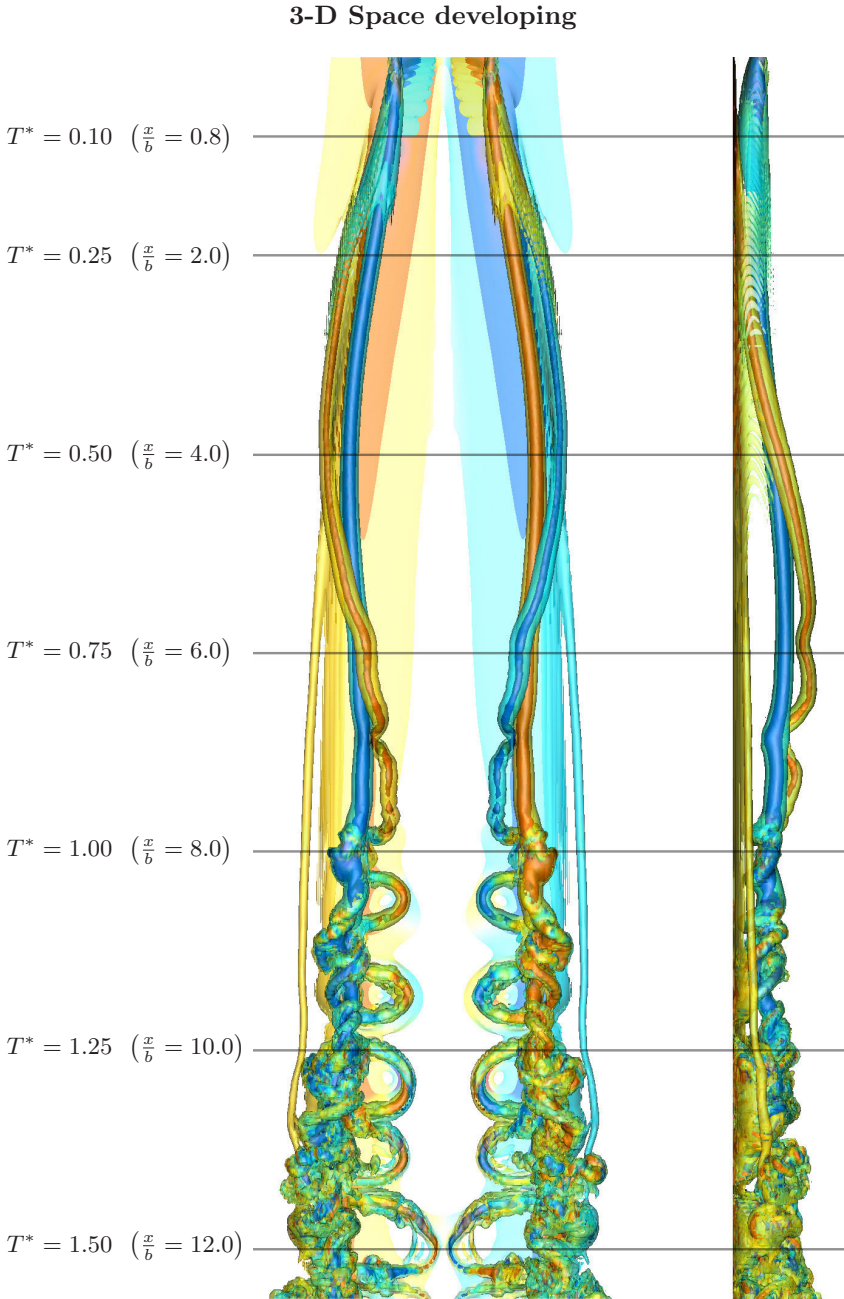


FIGURE 6.10 – 3-D Space developing LES. Isosurfaces of vorticity norm, $||\Omega|| = 80$ (opaque) and 25 (translucent), colored by the longitudinal component of vorticity when flow has reached its regime state. Top and side views. The locations of the time averaging planes used in Figs. 6.5, 6.7 and 6.8 are also shown.

3-D Space developing

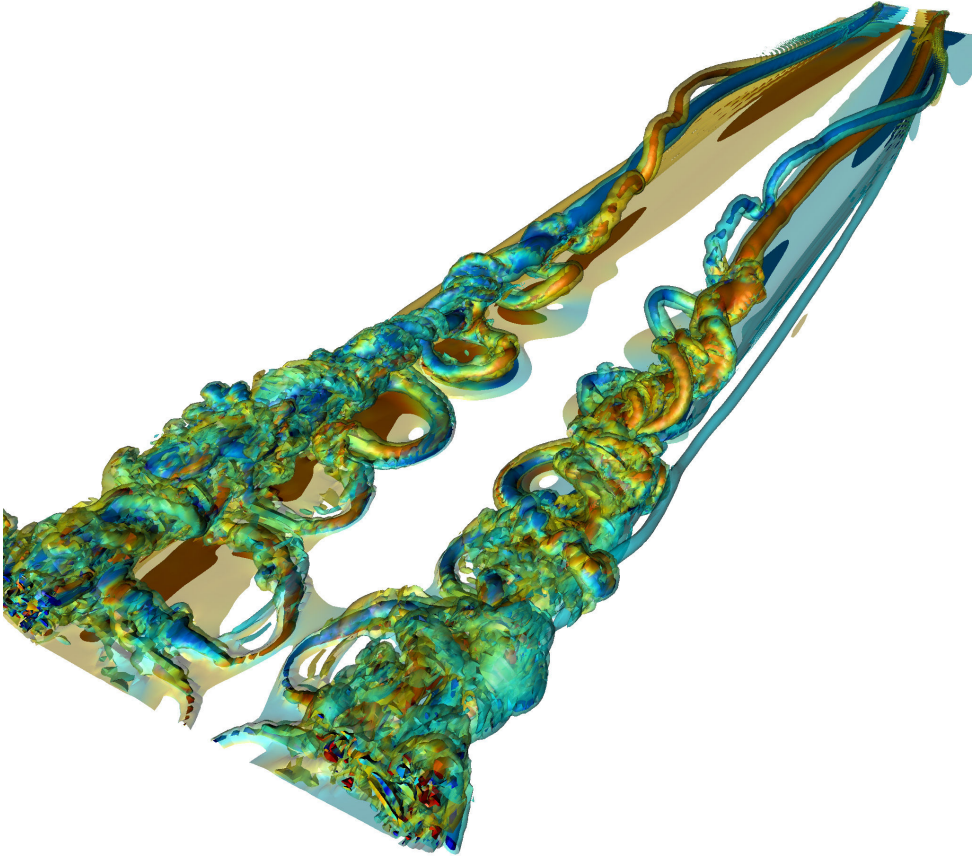


FIGURE 6.10 – (cont.) 3-D Space developing LES. Isosurfaces of vorticity norm, $||\Omega|| = 80$ (opaque) and 25 (translucent), colored by the longitudinal component of vorticity when flow has reached its regime state. Perspective view.



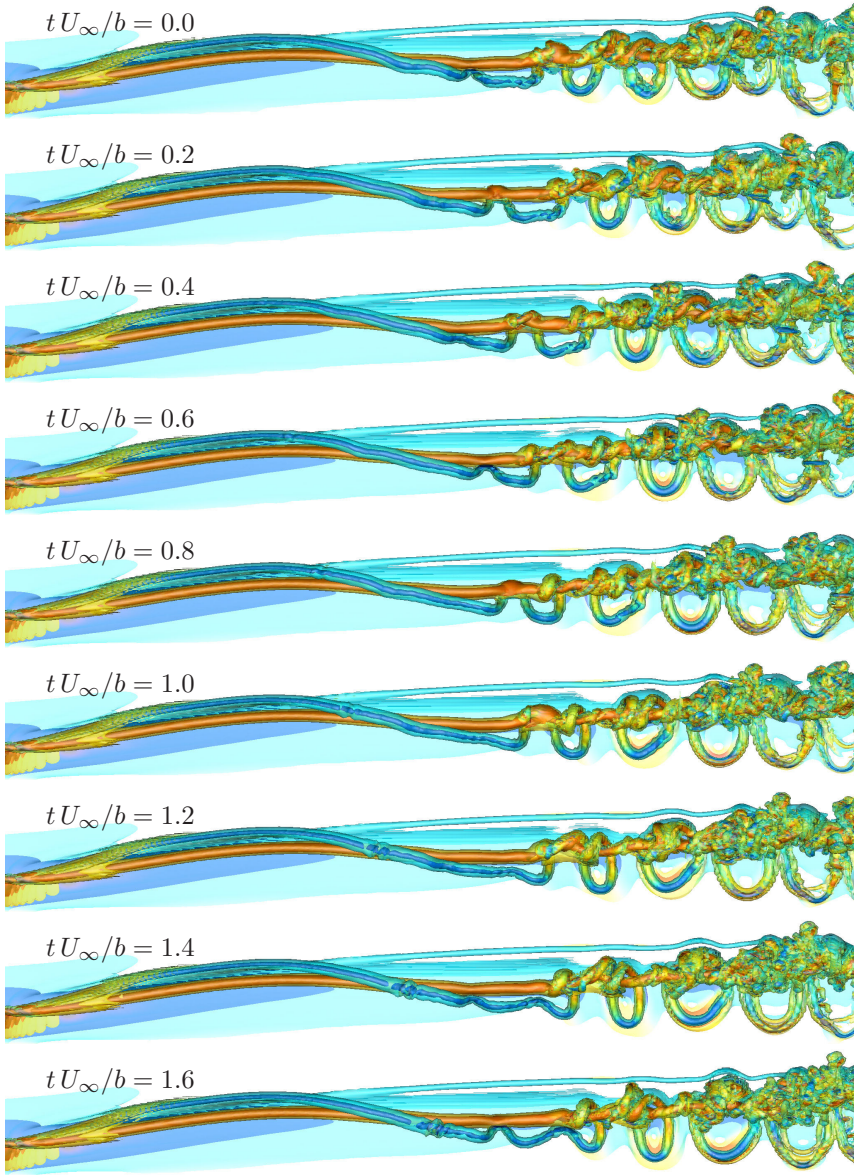


FIGURE 6.11 – 3-D Space developing LES. Time sequence of isosurfaces of vorticity norm, $||\Omega|| = 80$ (opaque) and 25 (translucent), colored by the longitudinal component of vorticity. Top view of right half-domain for each time frame are shown. $t = 0$ here corresponds to an arbitrary time (the same as in Fig. 6.10), once the system has reached its regime.

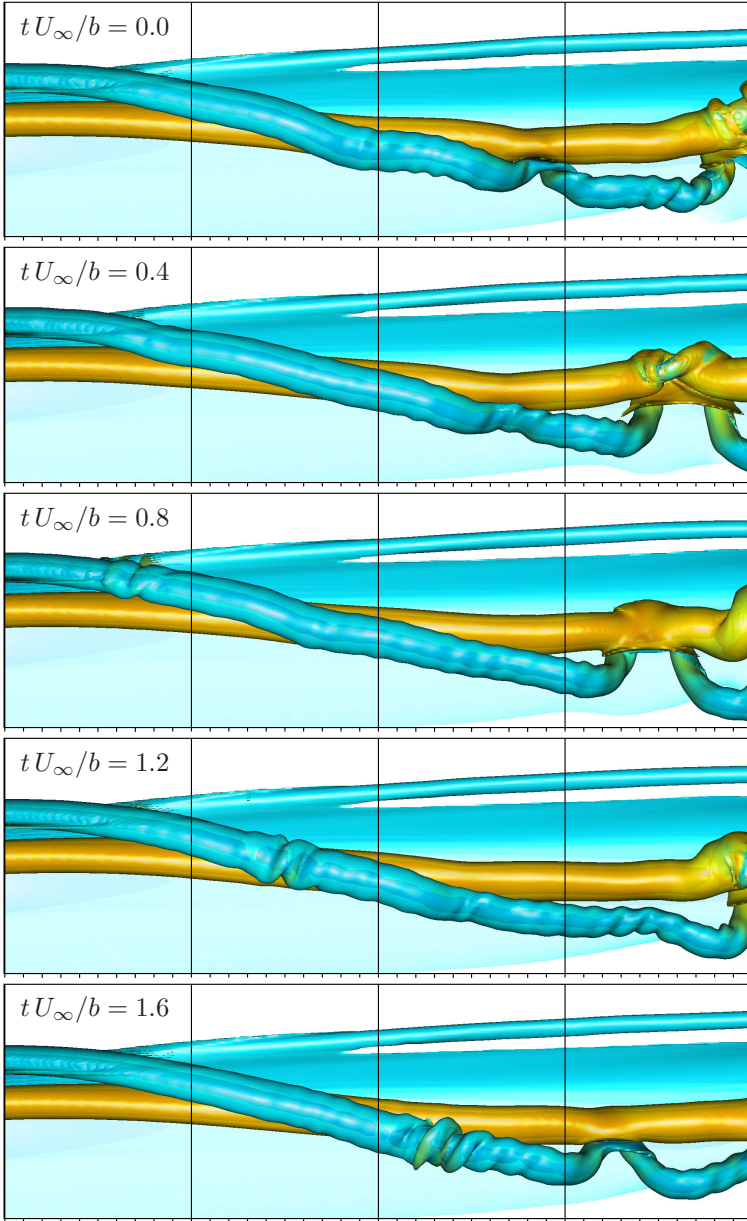


FIGURE 6.12 – 3-D Space developing LES. Time sequence of isosurfaces of vorticity norm, $||\Omega|| = 80$ (opaque) and 25 (translucent), colored by the longitudinal component of vorticity. Top view of right half-domain (zoom on region $x/b=[4, 8]$) for each time frame are shown. $t = 0$ here corresponds to an arbitrary time (the same as in Fig. 6.10), once the system has reached its regime.



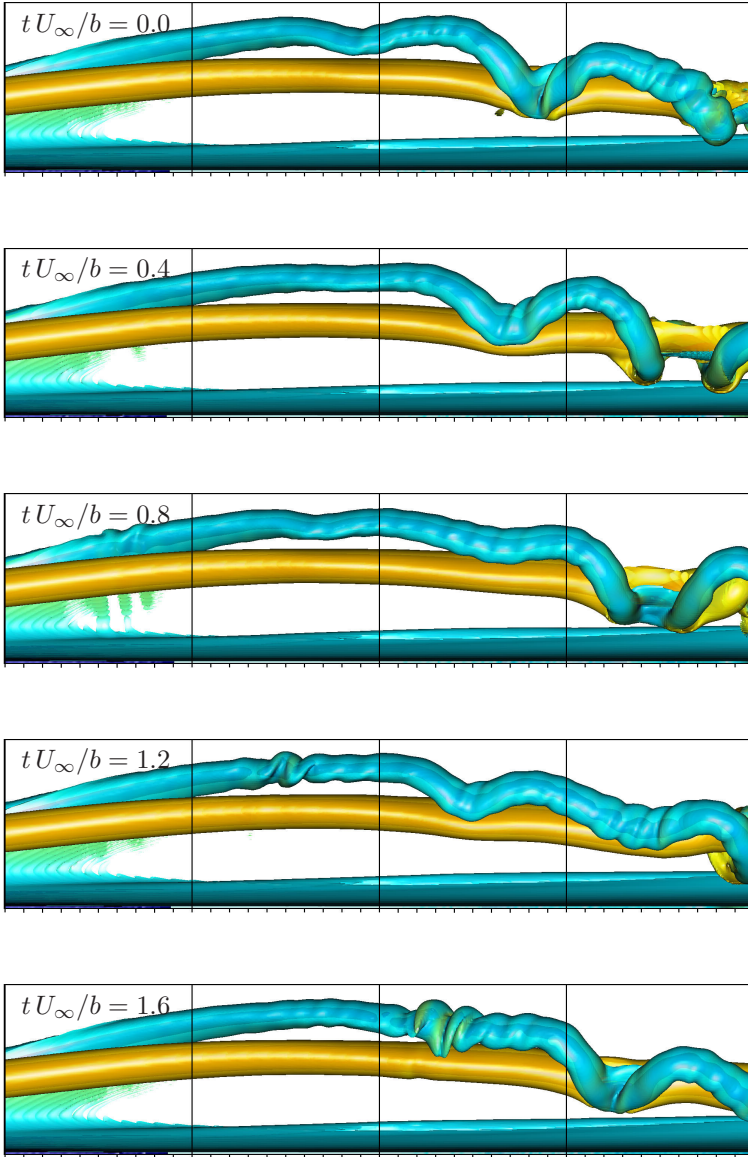


FIGURE 6.13 – 3-D Space developing LES. Time sequence of isosurfaces of vorticity norm, $\|\Omega\| = 80$ (opaque) and 25 (translucent), colored by the longitudinal component of vorticity. Side view of right half-domain (zoom on region $x/b=[4, 8]$) for each time frame are shown. $t = 0$ here corresponds to an arbitrary time (the same as in Fig. 6.10), once the system has reached its regime.

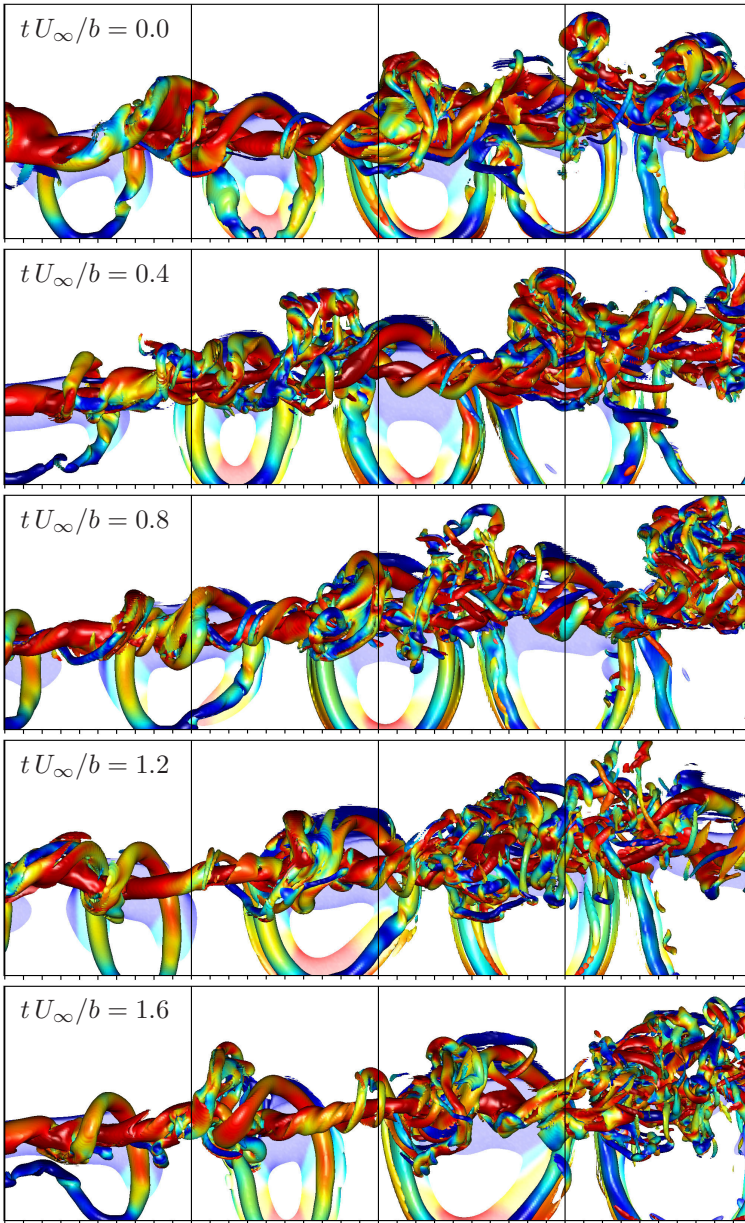


FIGURE 6.14 – 3-D Space developing LES. Time sequence of isosurfaces of vorticity norm, $||\Omega|| = 80$, colored by the longitudinal component of vorticity. Top view of right half-domain (zoom on region $x/b=[8, 12]$) for each time frame are shown. $t = 0$ here corresponds to an arbitrary time (the same as in Fig. 6.10), once the system has reached its regime.



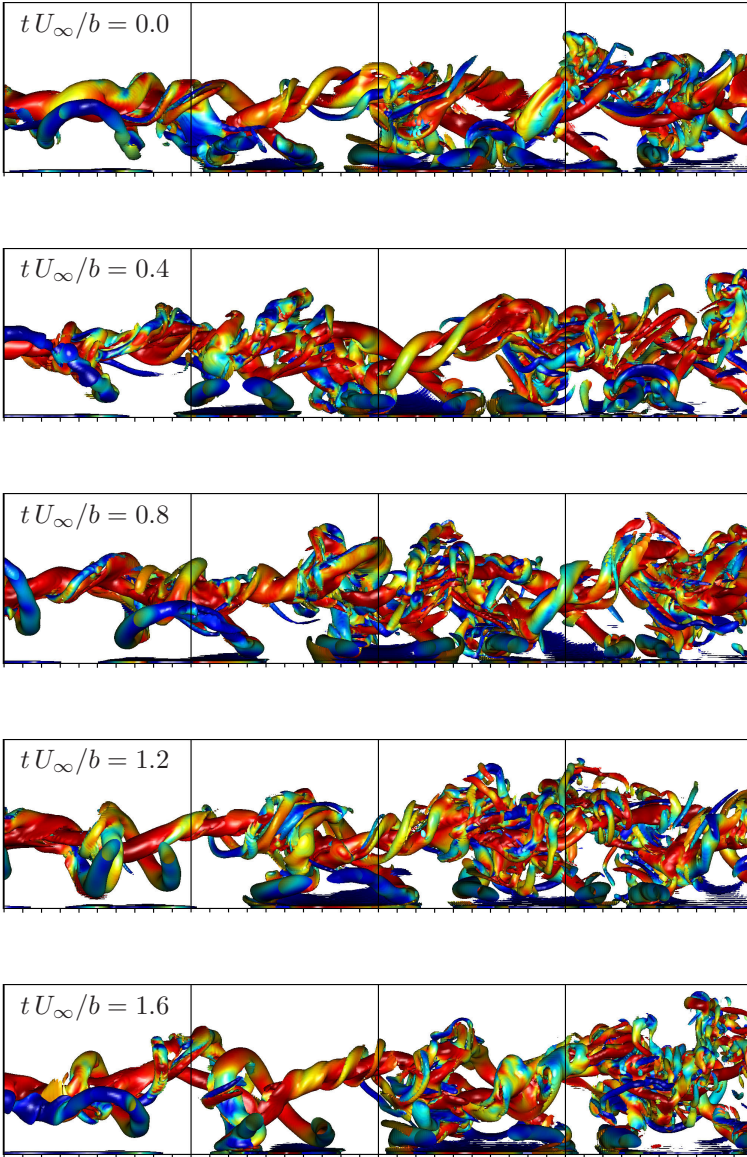


FIGURE 6.15 – 3-D Space developing LES. Time sequence of isosurfaces of vorticity norm, $||\Omega|| = 80$, colored by the longitudinal component of vorticity. Side view of right half-domain (zoom on region $x/b=[8, 12]$) for each time frame are shown. $t = 0$ here corresponds to an arbitrary time (the same as in Fig. 6.10), once the system has reached its regime.

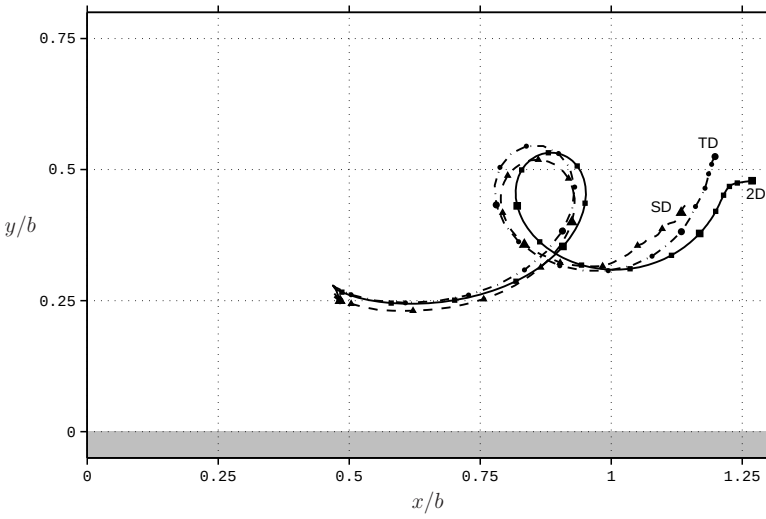


FIGURE 6.16 – Trajectories of primary vortex locations for the 2-D DNS (2D), the 3-D time developing LES (TD) and the 3-D space developing LES (SD). Symbols are shown at constant time intervals: small symbols for $\Delta T^* = 0.1$, large symbols for $\Delta T^* = 0.5$.

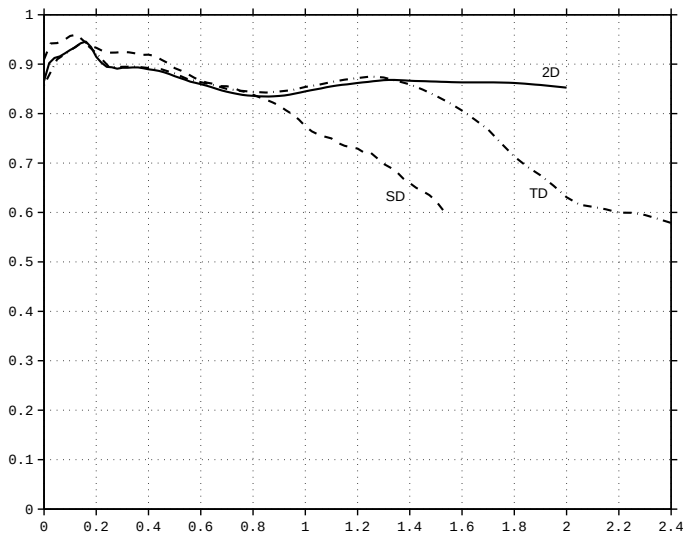


FIGURE 6.17 – Time evolution of maximum circulation, $\Gamma_{\max}$, for the 2-D DNS (2D), the 3-D time developing LES (TD) and the 3-D space developing LES (SD).



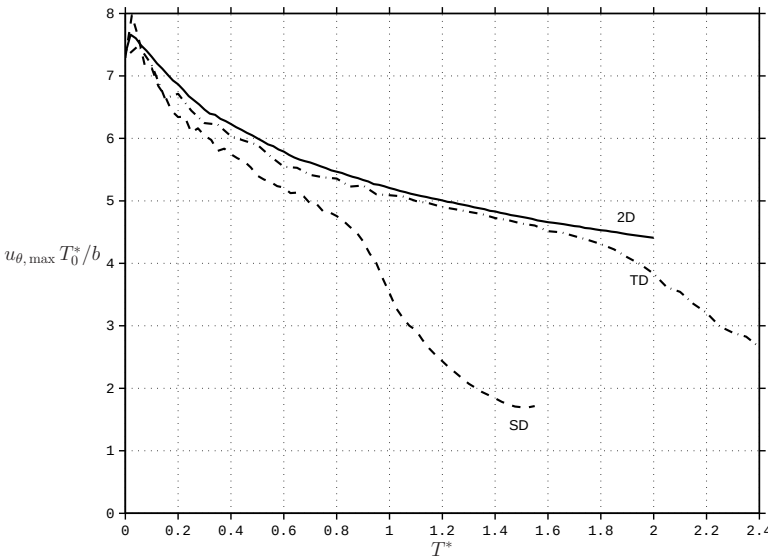


FIGURE 6.18 – Time evolution of $u_{\theta, \max}$ for the 2-D DNS (2D), the 3-D time developing LES (TD) and the 3-D space developing LES (SD).

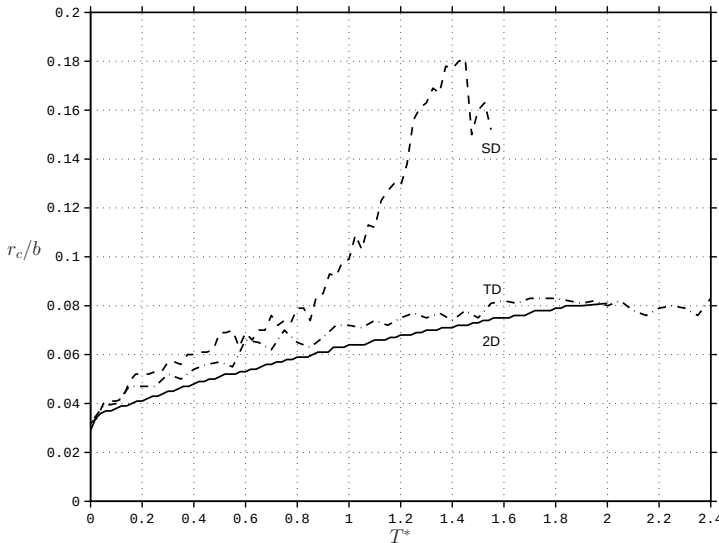


FIGURE 6.19 – Time evolution of r_c for the 2-D DNS (2D), the 3-D time developing LES (TD) and the 3-D space developing LES (SD).

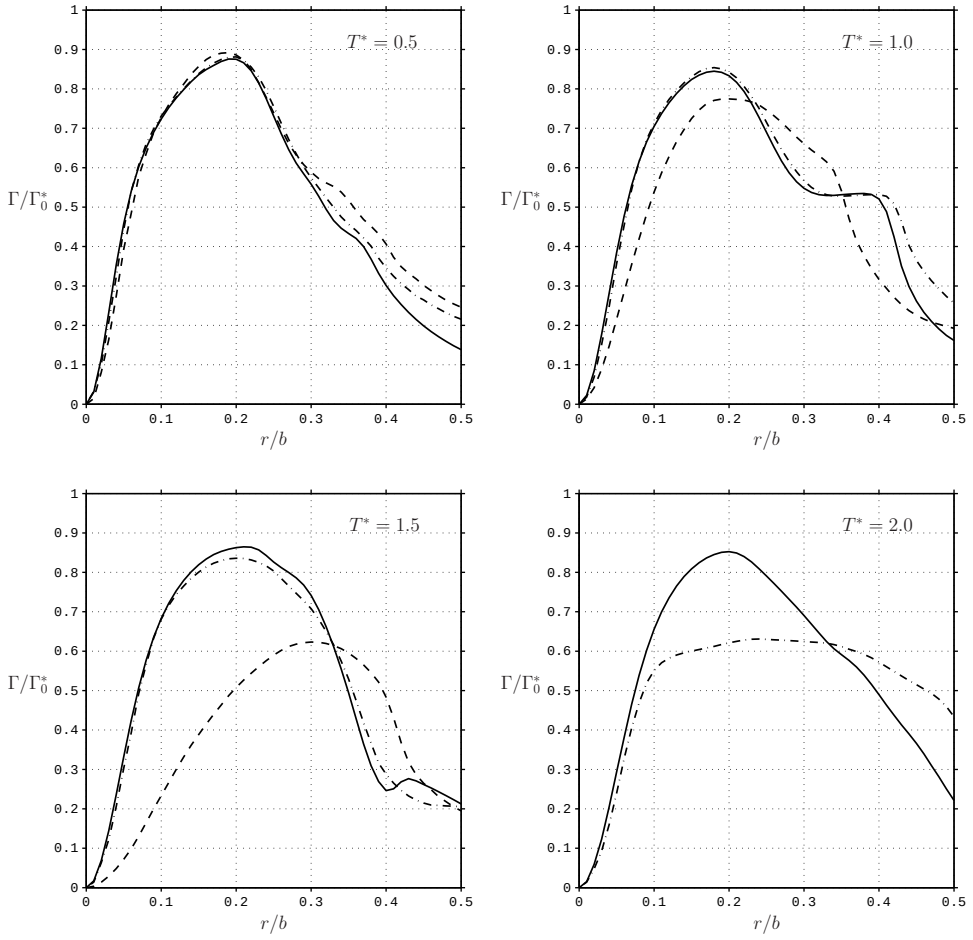
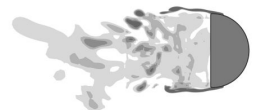


FIGURE 6.20 – Circulation profiles, $\Gamma(r)$, for the 2-D DNS (plain), the 3-D time developing LES (dash-dot) and the 3-D space developing LES (dash).



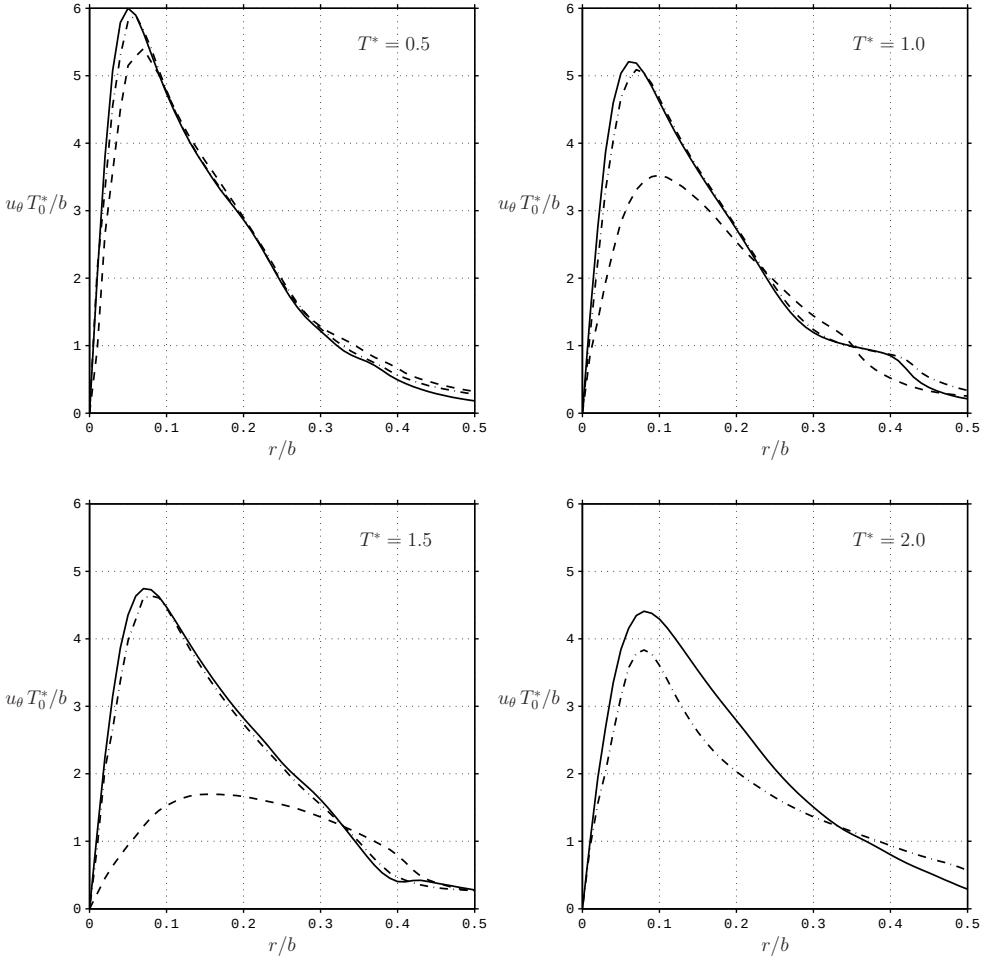


FIGURE 6.21 – Tangential velocity profiles, $u_\theta(r)$, for the 2-D DNS (plain), the 3-D time developing LES (dash-dot) and the 3-D space developing LES (dash).

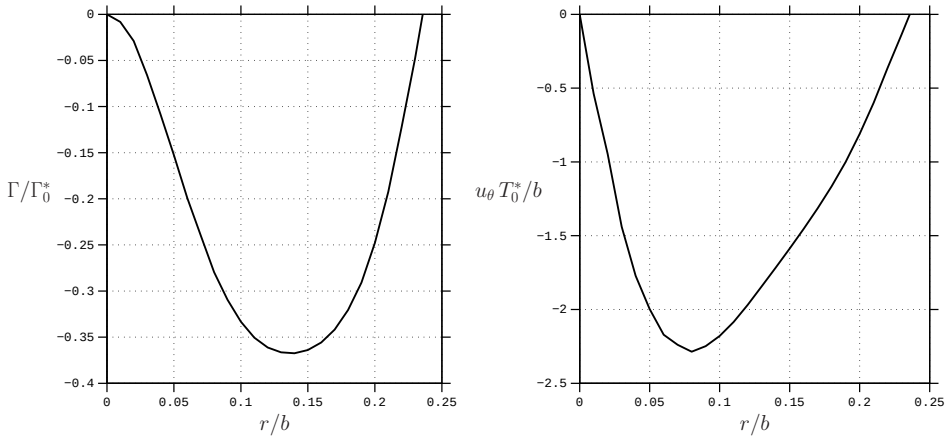


FIGURE 6.22 – 3-D Space developing LES. Circulation profile, $\Gamma(r)$ (left), and tangential velocity profile, $u_\theta(r)$ (right), for the secondary vortex at $T^* = \frac{x}{U_\infty} \frac{1}{T_0^*} = 0.5$.

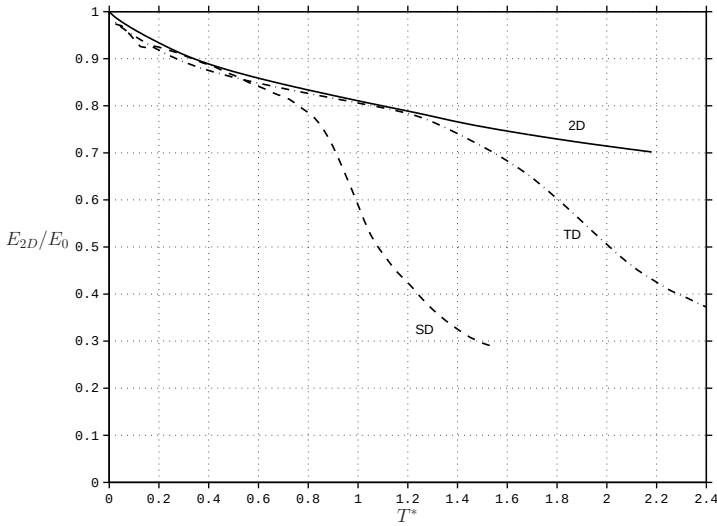
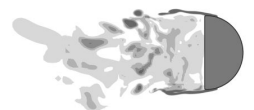


FIGURE 6.23 – Time evolution of cross-plane kinetic energy for the 2-D DNS (2D), the space averaged 3-D time developing LES (TD) and the time averaged 3-D space developing LES (SD).



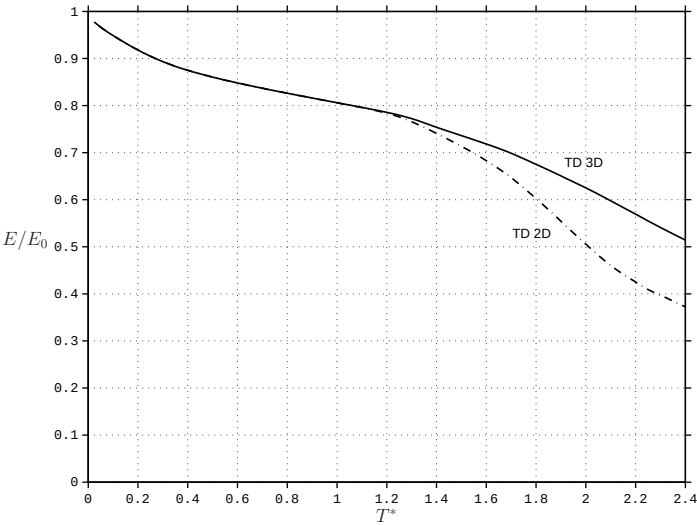


FIGURE 6.24 – 3-D Time developing LES. Time evolution of kinetic energy for the space averaged field (TD 2D) and for the complete 3-D field (TD 3D).

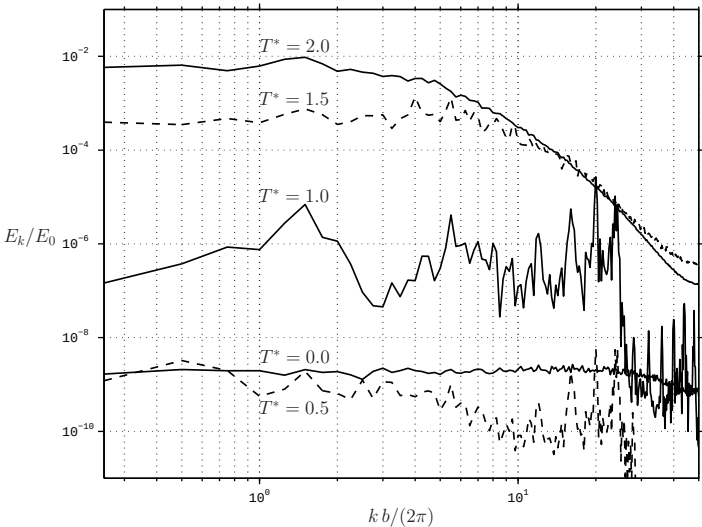


FIGURE 6.25 – 3-D Time developing LES. Modal energy spectra for distinctive times.

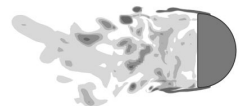
CHAPTER 7

Investigation of a hybrid RANS-LES approach

7.1 Introduction

In the previous chapters of the present work, we essentially focused our attention on developing an efficient numerical method for the simulation of turbulent flows in external aerodynamics. However, no matter how efficient the numerical method, there will always be the issue of turbulence modeling. Even with the currently available computational resources, most engineering type applications are still out of reach for wall-resolved large-eddy simulation (LES). Indeed, a very large number of grid points is required in the near-wall region of attached turbulent boundary layers to be able to apply LES. This is because LES requires that most of the Reynolds stresses are actually resolved by the unsteady resolved motions. Bagget *et al.* [3] calculated that the number of grid points required for accurate LES of a turbulent boundary layer scales as $N \sim Re_\tau^2$. Therefore, in the recent years, there has been a lot of effort put into combining the Reynolds-Averaged Navier Stokes (RANS) approach with LES, in order to reduce its computational cost. Several approaches have been proposed to achieve this; they can essentially be subdivided into two categories.

The first approach is to avoid resolving the near-wall layer altogether by



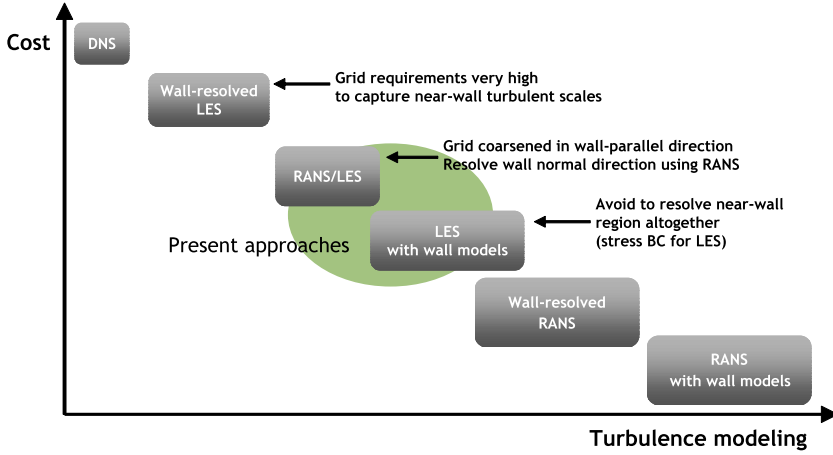


FIGURE 7.1 – From DNS to RANS: computational cost versus modeling.

applying wall models which provide wall stress boundary conditions. Such methods have been used with some success, among others, by Balaras *et al.* [5], Cabot and Moin [8], Wang and Moin [95].

The second approach, which we will investigate in this chapter, involves the use of grids coarsened in the wall-parallel directions, while leaving the wall-normal resolution essentially unchanged. As not all turbulent scales can be resolved on such grids, additional modeling is required. This is done by replacing the LES by the RANS model equations in the near-wall region. RANS is well suited for these type of grids as only the wall-normal gradients must be resolved.

A well-known approach in this category is the detached-eddy simulation (DES) [90] which was designed to simulate massively separated flows: it is obtained by modifying the one equation Spalart-Allmaras eddy-viscosity RANS model such that it behaves as a LES subgrid-scale (SGS) model outside attached boundary layer regions. In this approach, the switching from RANS to LES is closely related to the local grid size. In order to work properly, DES requires that the wall parallel grid spacing in attached boundary layers is larger than the boundary layer thickness (this ensures that the RANS model is active). If this is not the case, the model will switch to LES within the attached boundary layer, resulting in the failure of the method. The pressure driven channel flow is a typical flow configuration where DES fails to accurately

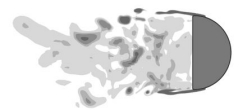
predict the flow, as discussed in [72].

In fact, the RANS-LES transition is the major challenge for all RANS-LES approaches. Similar shortcomings to those of DES were also observed by other authors using different RANS-LES approaches. De Langhe *et al.* applied their RANS-LES model, based on the Renormalization Group (RG) theory, [34] to the turbulent channel flow and obtained similar results to the DES solution [35]. Yet another approach is the Scale-Adaptive Simulation (SAS) model by Menter and Egorov [65]. As for DES, it is based on a RANS turbulence model (Menter's SST model [64, 66]) by modifying a turbulence length-scale to allow the SAS model to dynamically adjust to resolved structures and act as an LES-like model.

In this chapter, we will further investigate a RANS-LES approach that is based on a novel concept developed by Templeton *et al.* [91] for the near-wall treatment of LES. The approach is tested on the turbulent channel flow, which is a simple but challenging case for RANS-LES modeling. The case of the "serpentine passage" was also investigated. Figure 7.1 gives a simple overview of existing turbulence modeling approaches with respect to the level of modeling and computational cost. It also indicates where the current approach is situated with respect to the existing modeling approaches. The exact location of each approach with respect to the others is of course disputable. In particular, one could argue that the grid requirements of a wall-resolved RANS computation are higher (certainly in the boundary layer regions) than those for a LES with wall models. On the other hand the LES requires not only to converge in time to a regime solution, it further requires to be run for a sufficiently long time in order to obtain converged statistics. Their relative location in terms of turbulence modeling is also disputable and will essentially depend on the investigated flow.

From a numerical point of view, we here used the Eulerian velocity-pressure formulation of the Navier-Stokes equations. The idea being, in the end, to use a RANS-LES type approach in the near-wall region which would be treated using an Eulerian near-wall solver in the hybrid Eulerian-Lagrangian approach (as developed in Chapter 3).

The numerical code is briefly described, in Section 7.2. As the code was originally a purely DNS code, we first implemented and validated various LES and RANS models (Section 7.3). Finally, the RANS-LES coupling approach is presented and investigated in Section 7.4.



7.2 Numerical method and computational details

We here solve the incompressible Navier-Stokes equations in their classical velocity-pressure formulation:

$$\nabla \cdot \mathbf{u} = 0 \quad (7.1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\frac{1}{\rho} \nabla p + \nabla \cdot (\nu_e (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)), \quad (7.2)$$

where, ν_e is an effective viscosity. Depending on the type of simulation that is performed (i.e., DNS, LES, RANS) the variables, $\mathbf{u}$, p and ν_e are defined differently:

- **DNS:** $\mathbf{u}$ and p represent the time and space dependent velocity and pressure fields. The effective viscosity reduces to the laminar viscosity in this case: $\nu_e = \nu$.
- **LES:** $\mathbf{u}$ and p represent the grid truncated time and space dependent velocity and pressure fields. The subgrid-scale velocity fluctuations are not resolved, and their effect on the resolved scales has to be modeled. We here consider subgrid viscosity models: thus, $\nu_e = (\nu + \nu_{\text{sgs}})$, where ν is the laminar viscosity and ν_{sgs} is the modeled subgrid-scale viscosity.
- **RANS:** $\mathbf{u}$ and p represent the mean velocity and (modified) mean pressure fields. The Reynolds stress tensor is here modeled using the turbulent viscosity hypothesis: thus, $\nu_e = (\nu + \nu_t)$, where ν is the laminar viscosity and ν_t is the modeled turbulent viscosity.

In order to solve Eqs (7.1) and (7.2), in either DNS, LES or RANS mode, a finite volume approach is used. The approach is based on the scheme developed by Rosenfeld *et al.* [84, 10] where volume flux variables are used with the traditional time-splitting fractional step method. The computational domain is transformed from the Cartesian (x, y, z) system to a body-fitted curvilinear non-orthogonal system (ξ, η, ζ) . The present implementation of the approach is however limited to geometries which are complex in two directions while the third direction must be treated using Cartesian coordinates (i.e., $z = \zeta$) and periodic boundary conditions. This permits to reduce the three-dimensional Poisson equation to a series of wave-number dependent two-dimensional problems (which are solved using an efficient multigrid iterative solver).

The time dependent variables are thus the volume fluxes across each face of the computational cells. Spatial derivatives are approximated using second-order accurate central differences in the curvilinear domain. The temporal discretization uses a combination of explicit and implicit schemes. The diffusive term in generalized coordinates is split into double derivative terms (e.g., $\frac{\partial^2}{\partial \xi^2} + \frac{\partial^2}{\partial \eta^2} + \frac{\partial^2}{\partial \zeta^2}$) and mixed derivative terms. The double derivative terms are discretized using an implicit time discretization: the second order Crank-Nicolson scheme. The mixed derivative terms are treated explicitly, together with the convection term, using the second order Adams-Bashforth scheme.

The temporal discretization of the momentum equation can thus be written as:

$$\begin{aligned} \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} &= \frac{1}{2} \left(3H(\mathbf{u}^n) - H(\mathbf{u}^{n-1}) \right) + \frac{1}{2} \left(3D_e(\mathbf{u}^n) - D_e(\mathbf{u}^{n-1}) \right) \\ &+ \frac{1}{2} \left(D_i(\mathbf{u}^*) + D_i(\mathbf{u}^n) \right) + F^n \end{aligned} \quad (7.3)$$

where $H(\cdot)$, $D_e(\cdot)$, $D_i(\cdot)$ and F are respectively: the nonlinear convection operator, the explicit diffusion operator, the implicit diffusion operator and a volume forcing term. The resulting implicit system is solved using an ADI approach to enable the use of one-dimensional tridiagonal solvers in each direction.

The intermediate velocity, $\mathbf{u}^*$ is then corrected to enforce mass conservation:

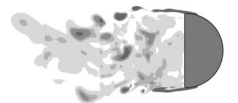
$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla_h(p^{n+1}) \quad (7.4)$$

where p^{n+1} is obtained by applying the divergence free constraint on $\mathbf{u}^{n+1}$, which results in the pressure Poisson equation:

$$\nabla_h \cdot \mathbf{u}^* = \frac{1}{\Delta t} \nabla_h^2 p^{n+1} \quad (7.5)$$

which is solved so as to enforce the mass conservation at the discrete level.

For more details on the method, please refer to [84]. The DNS version of the code used here was originally developed by Wu *et al.* [102]. We adapted the code to enable the use of an effective space and time dependent effective eddy viscosity (which is also treated implicitly). The code was modified to also allow for periodic boundary conditions in the streamwise direction. A source term was added to the momentum equations to enforce a pressure gradient which drives the flow. The pressure gradient can be adjusted dynamically to maintain a constant mass flux through the domain. This is an important feature as it has been shown that the flow converges more rapidly to a statistically stationary regime when the mass flow is imposed.



7.3 Code validation: from DNS to RANS and DES

7.3.1 DNS of turbulent channel at $Re_\tau = 180$

A list of commonly used quantities for the turbulent channel flow and the symbols adopted in this chapter are given in Appendix D.

The turbulent channel flow at $Re_\tau=180$ was computed using direct numerical simulation (DNS). The results are compared to the DNS by Moser *et al.* [69].

The computational domain size is $4\pi H \times 2H \times \frac{4}{3}\pi H$ in respectively stream-wise, normal and spanwise directions. A 128^3 grid is used with a hyperbolic tangent mapping in the wall-normal direction:

$$y(j) = \frac{1}{\tanh(\alpha)} \tanh\left(\alpha \left(1 - 2\frac{j}{N_y}\right)\right) \quad (7.6)$$

The stretching parameter $\alpha = 1.6$ was chosen to ensure that the first grid point is located at $y^+ < 1$. The resulting grid resolution is $\Delta x^+ = 17.6$, $\Delta y_w^+ = 0.8$ (at the wall), $\Delta y_c^+ = 4.9$ (at the channel center) and $\Delta z^+ = 5.9$. The time step for the simulation was $\Delta t^+ = 10^{-3}$.

Constant pressure gradient

We first simulated the channel flow with a constant pressure gradient. The initial condition consisted of a perturbed Poiseuille flow with a mass flow set equal to that of the fully turbulent channel at $Re_\tau=180$. Because a constant pressure gradient is imposed the flow first accelerates before it transitions to turbulence. Once the flow becomes fully turbulent, it still has to recover from this momentum excess before converging to the statistically steady solution.

Statistics were gathered from $t^+ = 30$ up to $t^+ = 35$. At that time, the mass flow was almost stabilized but one could still notice a slightly decreasing mean slope. The mean velocity profile, velocity fluctuation profiles and shear stress profile are shown in Fig. 7.2. The results are in good agreement with those obtained by [69] considering the fact that we here use a second order method (while the reference DNS uses a spectral/Chebychev-tau discretization). The total shear stress (i.e., the sum of the laminar and turbulent stresses) is a straight line going from one at the wall to zero at the center of the channel, indicating the good convergence of the results.

Constant mass flow

We performed a second simulation with a constant imposed mass flow rate, such that $Re_b = \frac{H U_b}{\nu} = 2822$ as in [69] (where U_b is the bulk velocity). The simulation was initialized using the solution at $t^+ = 35$ from the constant pressure simulation (scaled to match the imposed mass flow). The resulting “effective” Re_τ from this simulation is 186.73, to be compared with 178.12 obtained by [69]. The results are also shown in Fig. 7.2. The slight difference with the constant pressure gradient simulation is probably due to the fact that the second simulation is better converged. Notice that the results from this simulation were scaled according to “effective” mean slip velocity, U_τ , and the effective viscous length scale, $\delta_\nu = \nu/U_\tau$, as it should. If one uses the “aimed” U_τ and δ_ν (the ones that correspond to $Re_\tau=180$) one would see an almost perfect agreement with the reference computation in the outer region of the flow.

7.3.2 LES of turbulent channel at $Re_\tau = 395$

Dynamic Smagorinsky model

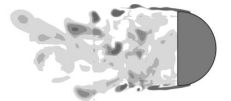
The Smagorinsky model (SM) is a widely used eddy-viscosity SGS model [88]. The eddy viscosity is given by:

$$\nu_{\text{sgs}} = (C_s \Delta)^2 \sqrt{2 S_{ij} S_{ij}} \quad (7.7)$$

where, Δ is the effective grid spacing, C_s is the model constant and $\bar{S}_{ij}$ is the deformation rate tensor for the resolved field. The Smagorinsky model yields good results when it is applied to homogeneous turbulent flows. However, in transitional flows, the Smagorinsky model behaves much less satisfactorily. One of the major flaws of the model is that the eddy-viscosity has the incorrect near-wall scaling.

In order to adapt the model to the local structure of the flow, Germano *et al.* [41] proposed an algorithm for adapting the Smagorinsky model by automatically adjusting the model constant, C_s , at each point in space and time. This adaptive procedure is known as the Dynamic Smagorinsky Model (DSM). The dynamic procedure is based on the application of a test filter, to the already discretized equations.

We here implemented the Germano-Lilly version of the dynamic procedure [60]. A good description of this procedure can be found in Sagaut [85]. In



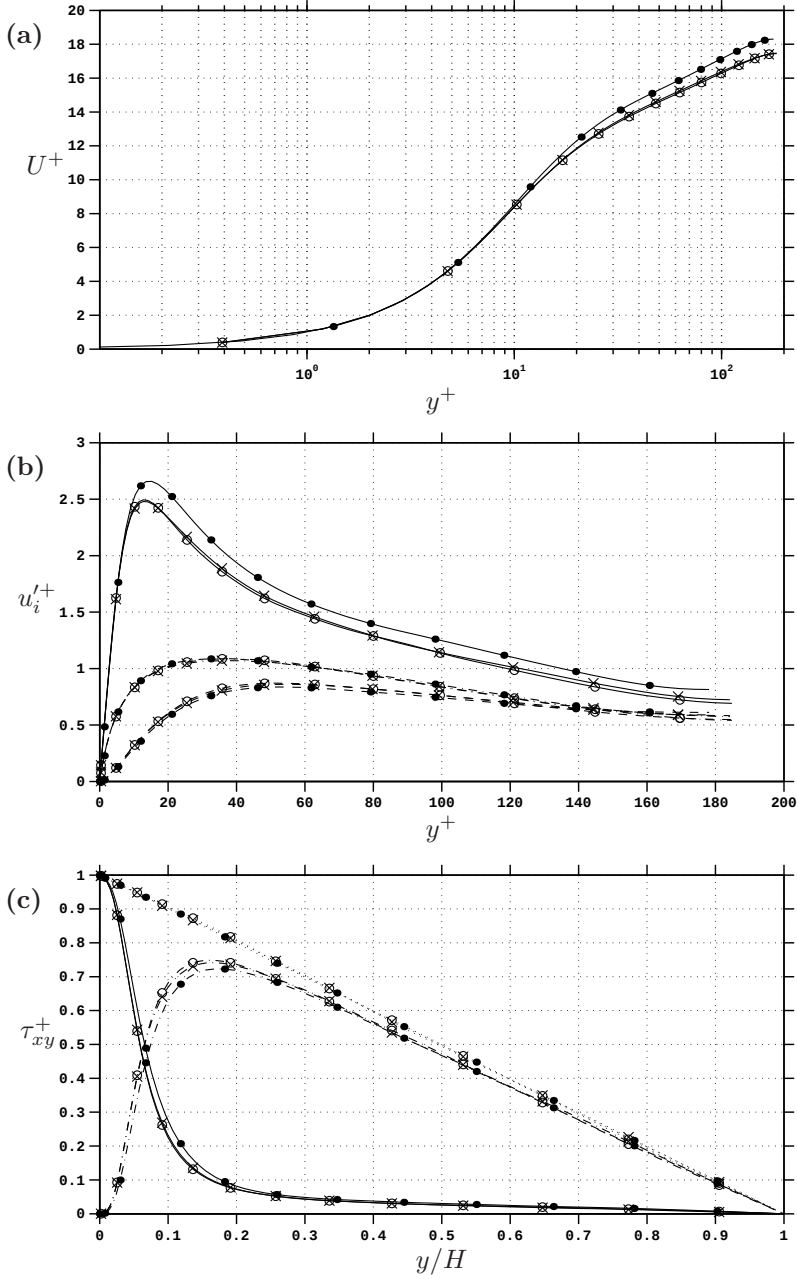


FIGURE 7.2 – DNS at $Re_\tau = 180$: constant mass flow ($\circ$), constant pressure gradient ($\times$) and DNS [69] ($\bullet$). (a) Mean velocity profile. (b) RMS velocity profile: u' (plain), v' (dash), w' (dash-dot). (c) Mean shear stress profile: τ_{xy}^{visc} (plain), $-\langle u'v' \rangle$ (dash-dot), τ_{xy}^{tot} (dot).

practice, the test filter function used here is a discrete approximation of a tophat filter of twice the celle size. In one dimension the filter is:

$$\hat{\phi}_i = \frac{1}{6}\phi_{i-1} + \frac{2}{3}\phi_i + \frac{1}{6}\phi_{i+1} \quad (7.8)$$

The test filter is applied in the homogeneous directions of the flow only.

Wall Adapting Local Eddy-viscosity model

The Wall Adapting Local Eddy-viscosity (WALE) SGS model developed by Nicoud *et al.* [70, 71] was also implemented. The model is based on the square of the velocity gradient tensor and accounts for the effects of both the strain and the rotation rate to obtain the local eddy-viscosity. Its greatest advantage is that it recovers the proper y^3 near-wall scaling for the eddy-viscosity without requiring a dynamic procedure.

The WALE model eddy-viscosity is given by:

$$\nu_{\text{sgs}} = (C_w \Delta)^2 \frac{(\mathcal{S}_{ij}^d \mathcal{S}_{ij}^d)^{3/2}}{(S_{ij} S_{ij})^{5/2} + (\mathcal{S}_{ij}^d \mathcal{S}_{ij}^d)^{5/4}} \quad (7.9)$$

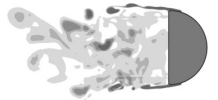
where $\overline{S}_{ij}$ is the deformation rate tensor for the resolved field. $\mathcal{S}_{ij}^d$ is defined as follows:

$$\mathcal{S}_{ij}^d = \frac{1}{2} (g_{ij}^2 - g_{ji}^2) - \frac{1}{3} \delta_{ij} g_{kk}^2, \quad (7.10)$$

with $g_{ij}^2 = \frac{\partial u_i}{\partial x_k} \frac{\partial u_k}{\partial x_j}$ and δ_{ij} the Kronecker symbol. The model constant C_w is here set to 0.5; this value is given by Nicoud in [71] and was calibrated numerically on isotropic decaying turbulence. In our computations the subgrid characteristic length scale Δ is set to the cubic root of the local cell volume.

Results

Both the DSM and WALE models were validated on the turbulent channel flow at $\text{Re}_\tau=395$. The results are compared to the DNS by Moser *et al.* [69]. The computational domain size is $2\pi H \times 2H \times \pi H$ in respectively streamwise, normal and spanwise directions. The computational grid is $80 \times 65 \times 64$, the mapping stretching parameter is $\alpha = 2.75$ (the grid is the same as the one used by Gullbrand in [43]). The resulting grid resolution is $\Delta x^+ = 31$, $\Delta y_w^+ = 0.6$ (at the wall, the first cell center is at $y^+ = 0.3$), $\Delta y_c^+ = 34$ (in the center of the channel) and $\Delta z^+ = 19$. The time step for the simulation was $\Delta t^+ = 10^{-3}$. We



here imposed a constant mass flow rate for both cases, such that $Re_b = 6930$ as in [69].

The flow was initialized from the Poiseuille solution with a small additional random perturbation. The flow transitioned naturally to turbulence. This can be seen on Fig. 7.4b and c: it shows the time evolution of the spatially averaged skin friction for the WALE case. One can see that the skin friction is initially low and corresponds to that of the laminar solution. It remains roughly constant until the flow transitions to turbulence. After transition the flow reaches a statistically steady state.

For both simulations, statistics were gathered from $t^+ = 20$ up to $t^+ = 100$. These statistics are thus very well converged. The mean velocity profile, the RMS velocity fluctuation profiles and the mean shear stress profiles are shown in Fig. 7.3. The average velocity profiles for both the DSM and WALE simulations are in almost perfect agreement with the reference DNS. The RMS velocities are essentially identical for the DSM and WALE but there are some difference with the DNS. The streamwise velocity fluctuations are slightly over-predicted, while the spanwise and wall-normal components are under-predicted. The same behavior was observed by other authors using the DSM [48, 43].

We now turn to the shear stress profiles. Obviously, the laminar shear stress profile for the DSM, WALE and reference DNS all collapse on the same curve, since the mean velocity profiles are (almost) identical for all cases. We again observe that the total shear stress is a straight line, indicating the good convergence of the results. The DSM and the WALE results differ from each other and from the DNS when considering the resolved turbulent stresses. Both the DSM and WALE are slightly lower than the DNS, as a fraction of the turbulent stress is captured by the subgrid-scale model. It is interesting to notice that the DSM modeled SGS stress is significantly larger than the WALE modeled one (about twice as large in the near wall region).

The difference for modeled turbulent stresses is better understood when considering the time and space averaged SGS eddy-viscosity (Fig. 7.4a). The subgrid scale viscosity for the WALE and DSM models both exhibit the correct near-wall behavior (i.e. $\nu_{sgs}(y) \sim y^3$). They are however significantly different: in the near wall region ($y^+ < 10$): the DSM SGS viscosity is about 10 times larger than the WALE one; it is the opposite in the center of the channel. Near the wall, the SGS viscosity is however very low compared to the laminar viscosity. The differences between WALE and DSM should therefore be negligible in that region. In the outer region, the differences are significant: the WALE SGS

viscosity has the same order of magnitude as the laminar viscosity there, while the DSM viscosity is up to 10 times lower. Considering these large differences, it is interesting to observe that the mean quantities such as the velocity profiles and velocity fluctuations are very similar.

Fig. 7.4a also shows the SGS viscosity for the classical Smagorinsky model (DS: without dynamic procedure): it is here obtained “a priori”, by applying the SM to the velocity field from the DSM simulation. The resulting SGS viscosity field is then averaged in space in the periodic directions. As expected the near-wall behavior of the SM is completely incorrect. In the center of the channel, the SGS viscosity is similar to the one obtained with WALE.

Finally, the spanwise (time and plane averaged) energy spectrum is shown in Fig. 7.5 for the WALE and DSM computations and for the reference DNS. Notice that the DNS has modes up to $k_z H = 254$ while the LES computations are truncated at $k_z H = 62$. The modal energy is plotted for each case at three different wall normal locations: at $y^+ = 20$, which is approximately the location of maximum streamwise fluctuations, at $y^+ = 100$, which is inside the logarithmic region, and in the center of the channel at $y^+ = 395$. At $y^+ = 20$, the DSM and WALE modal energies are very similar; they agree well with the DNS for the low wavenumbers; the high number fluctuations are slightly above the DNS (this effect is even more visible at $y^+ = 5$, not shown). At larger wall normal distances, the DSM has lower energy in the low wave numbers than WALE; at high wave numbers the trend is inverted. In the center of the channel ($y^+ = 395$), the WALE computation has significantly less energy in the high wave numbers than both the DNS and the DSM computation. In general the agreement with the reference DNS, in terms of modal energy, is better for the DSM computation. These observations seem to indicate that the WALE SGS viscosity is too high in the center of the channel.

7.3.3 RANS and DES of turbulent channel at $Re_\tau = 2,000$

The one-equation Spalart-Allmaras (SA) [89] and the two-equation $k - \omega$ [96] RANS turbulence models were implemented. The models and their corresponding constants are given in Appendix E.

In Fig. 7.6a, the mean velocity profile for the turbulent channel at $Re_\tau = 2,000$ using SA is compared to that obtained using the commercial code Fluent. The Fluent computation has $N_y = 128$ grid points with $\Delta y_w^+ = 0.8$. Our grid has $N_y = 64$ grid points in the wall normal direction and the first grid point



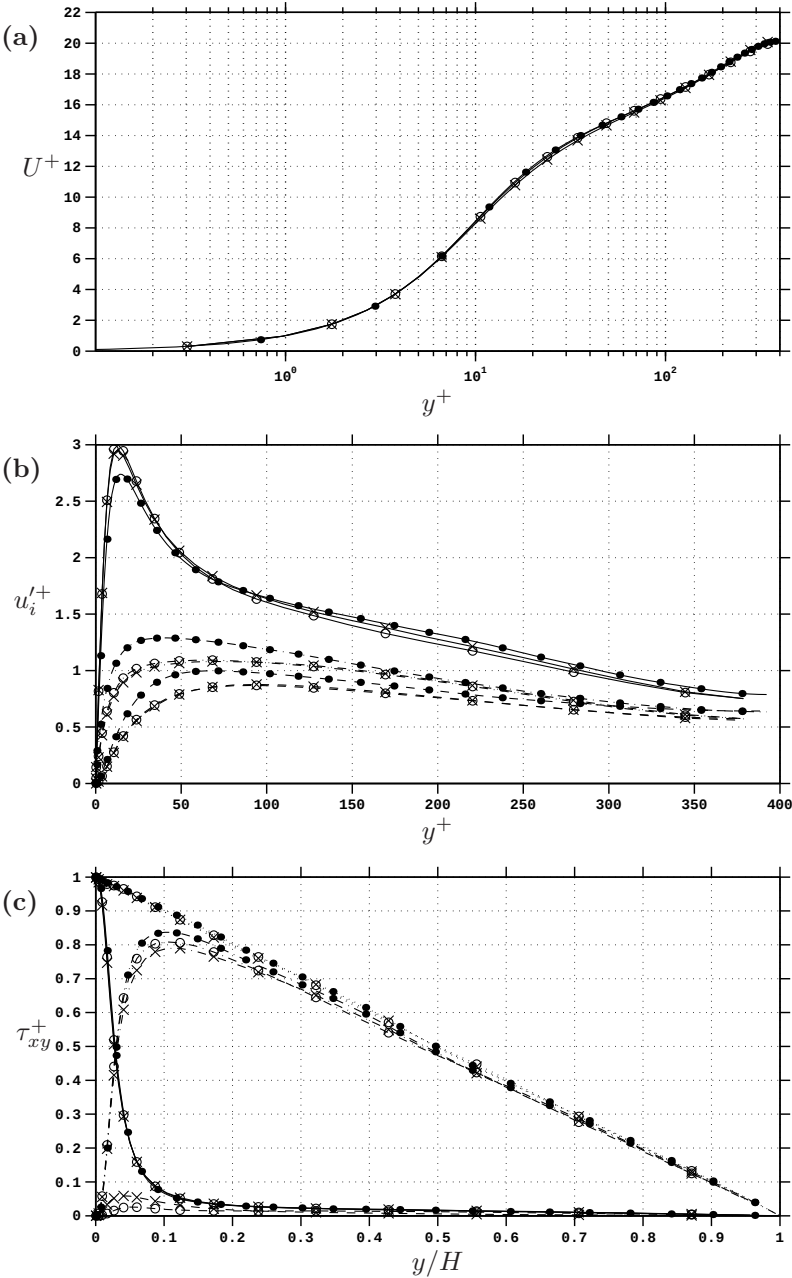


FIGURE 7.3 – LES at $Re_\tau = 395$: WALE ($\circ$), DSM ($\times$) and DNS [69] ($\bullet$). (a) Mean velocity profile. (b) RMS velocity profile: u' (plain), v' (dash), w' (dash-dot). (c) Mean shear stress profile: τ_{xy}^{visc} (plain), τ_{xy}^{sgs} (dash), $-\langle u'v' \rangle$ (dash-dot), τ_{xy}^{tot} (dot).

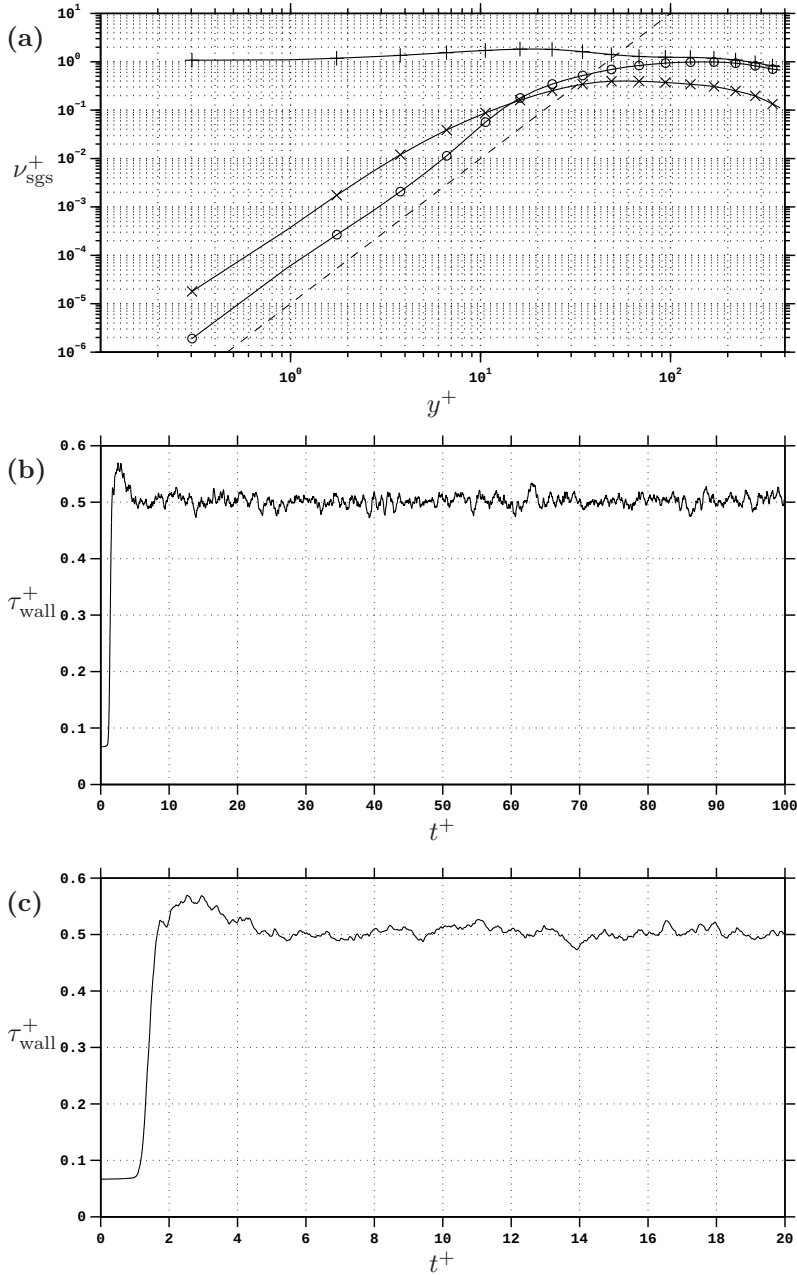
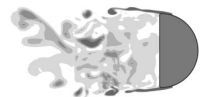


FIGURE 7.4 – LES at $Re_\tau = 395$. (a) SGS viscosity profile: WALE ($\circ$), DSM ($\times$), SM ($+$), theoretical near wall behavior (dash). (b) Time history of wall shear stress (WALE), τ_{wall} . (c) Time history of wall shear stress (WALE), τ_{wall} (zoom on startup phase).



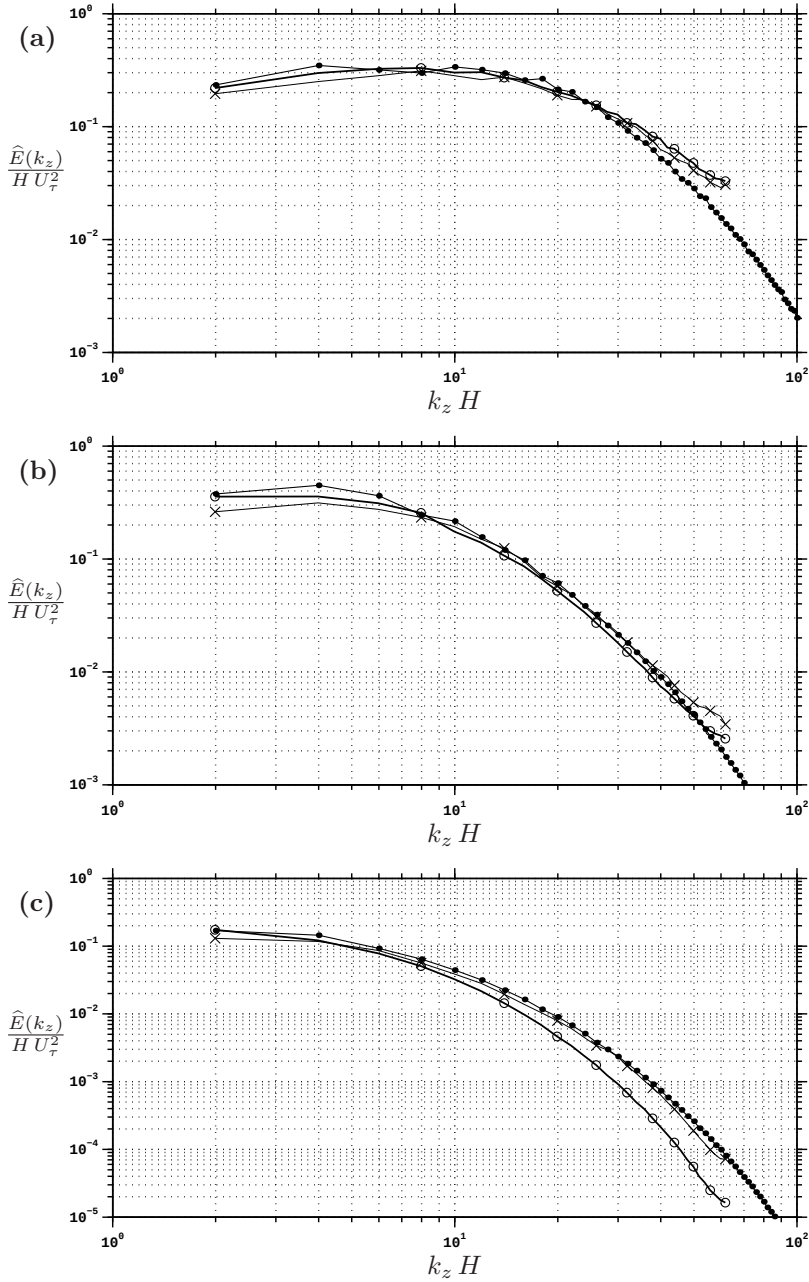


FIGURE 7.5 – LES at $Re_\tau = 395$, energy spectra in spanwise direction at different y^+ locations: WALE ($\circ$), DSM ($\times$) and DNS [69] ($\bullet$). (a) $y^+ \simeq 20$. (b) $y^+ \simeq 100$. (c) $y^+ \simeq 395$.

is located at $\Delta y_w^+ = 0.8$ (the $N_y = 128$ case was also performed and is slightly closer to the Fluent results). The results are in very good agreement with each other, as it should (the only difference in this case being the numerical method).

The reference DNS computation by Hoyas and Jimenez [45] is also shown and lies slightly higher than SA in the log-law region. In between $y^+ = 70$ and $y^+ = 400$ ($y/H = 0.2$), the DNS computation matches very well the log law:

$$U^+ = \frac{1}{\kappa} \log(y^+) + C \quad (7.11)$$

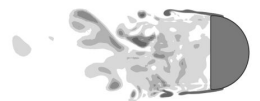
with $\kappa = 0.41$ and $C = 5.2$. The SA solution is slightly lower because the model constants we use here are those that were calibrated for the turbulent boundary layer, not for the turbulent channel.

The Detached-Eddy Simulation (DES) model [90] derived from the SA RANS model was also implemented. The DES modification concerns the destruction term in the eddy-viscosity transport equation by modifying the characteristic length scale, d , in equation (E.5). In the RANS SA model, d is the distance to the nearest wall and expresses the confinement of the eddies by the wall. In the DES model, one simply replaces d with $\tilde{d}$, which is defined as:

$$\tilde{d} = \min(d, C_{\text{DES}}\Delta), \quad (7.12)$$

where $\Delta = \max(\Delta x, \Delta y, \Delta z)$. The role of Δ is to allow the energy cascade down to the grid size: away from the wall the DES model is thus intended to act as an LES model while, near the wall, the model reverts to the classical RANS SA model.

DES was conceived for external aerodynamics, where thin boundary layers and massive separation regions coexist. Very encouraging results have been obtained for such flows [92, 16, 17]. Such good results are however dependent on the design of the computational grid: the user has to direct the model to operate in the appropriate RANS or LES mode. In its intended use, the boundary layers are treated entirely with RANS equations by creating a grid with a large spacing parallel to the wall, compared to the boundary-layer thickness. Nikitin *et al.* [72] investigated the behavior of DES in a channel flow, thus in a configuration where the boundary layer thickness (basically the half channel width, H) is much larger than the wall parallel grid spacing. They essentially observed the presence of a double log region: one close to the wall, which corresponds to the RANS region, and a “resolved” log region corresponding to the LES region.



In between, there is some kind of buffer layer. As a result, the predicted skin friction is significantly too low.

Figure 7.6a also shows the results obtained using the DES model with our code. They can be compared to the DNS and RANS solutions. The computational domain size is $2\pi H \times 2H \times \pi H$ in respectively streamwise, normal and spanwise directions. The number of grid points is $N_x = 64$, $N_y = 64$ and $N_z = 32$: the associated grid size is $\Delta x^+ = 196$, $\Delta y_w^+ = 0.8$ (at the wall), $\Delta y_c^+ = 294$ (in the center of the channel) and $\Delta z^+ = 196$ (thus clearly too coarse to perform wall resolved LES). The switching location from the RANS to the LES mode thus occurs at $y^+ \simeq 200$ (at that location $d = \Delta$). The unphysical buffer region is precisely located around the switching location. Those results compare very well with those obtained in [72].

Figure 7.6b shows the mean turbulent viscosity profiles for the RANS and DES computations: the turbulent viscosity is essentially identical for both cases near the wall (in the RANS region) while, towards the center of the channel, the DES viscosity is much lower and exhibits a typical LES behavior.

In Fig. 7.6c the mean turbulent shear stress is shown for the RANS, DES and DNS [45] solutions. Obviously, the turbulent stress is entirely modeled for the RANS case, while it is entirely resolved for the DNS. The agreement between RANS and DNS is excellent. For DES, the modeled turbulent shear stress is dominant up to $y/H = 0.1$ (i.e., the switching location, $y^+ = 200$), while towards center of the channel the turbulent stress is almost entirely resolved. The total turbulent shear stress for DES (not shown) is slightly below the RANS and DNS solutions in the RANS region, hence the underpredicted skin friction.

7.3.4 Discussion

The results presented in this section were used to both validate the code in its different modes (DNS, LES, RANS and DES) and, at the same time, to illustrate the capabilities and the limitations of those different levels of modeling. In fact, we covered the whole spectrum of possible turbulence modeling approaches as presented in Fig. 7.1.

The approximate computational cost (to obtain statistically converged results) corresponding to those simulations are given in Table 7.1. The trends are quite obvious:

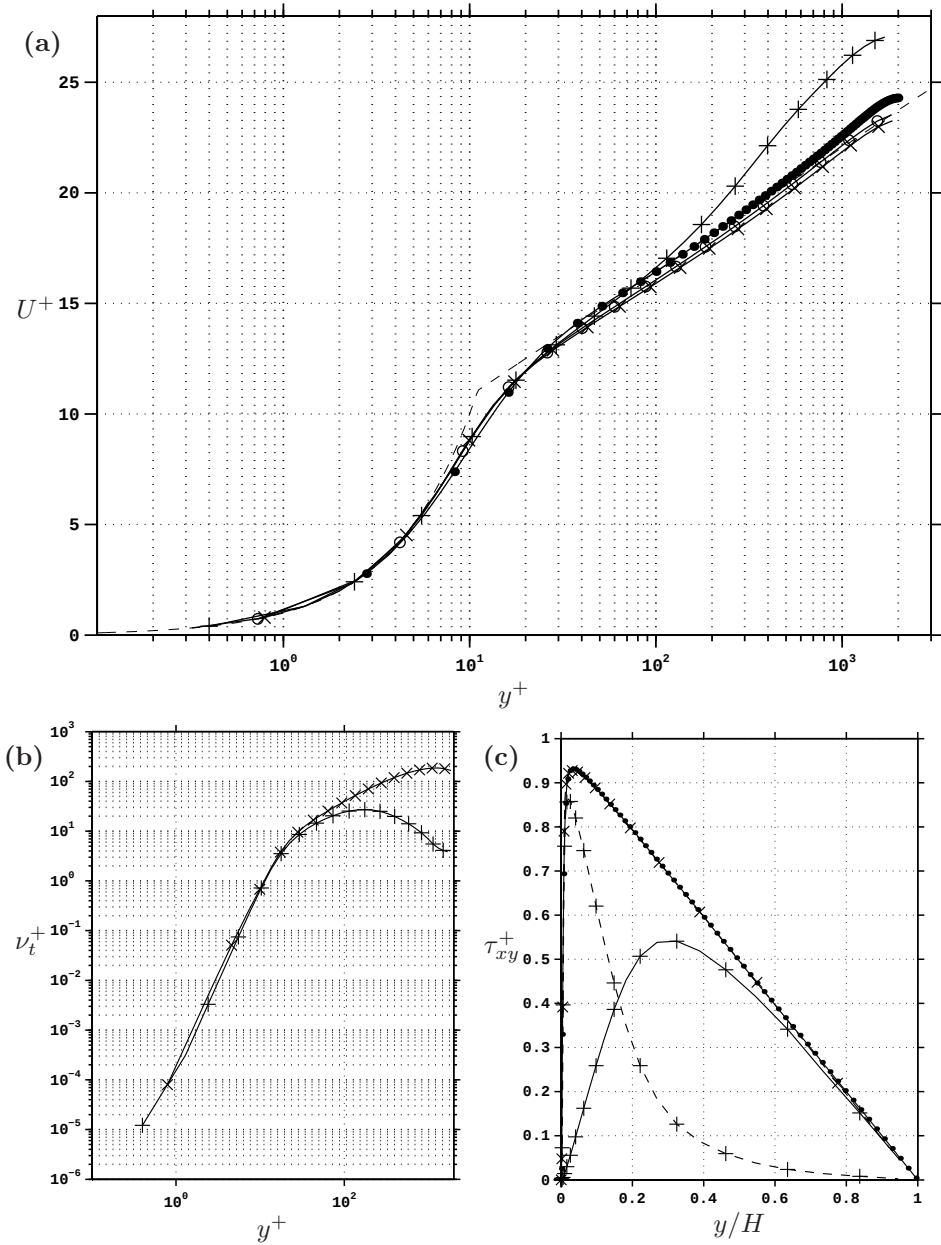


FIGURE 7.6 – RANS and DES at $Re_\tau = 2,000$: RANS-SA ($\times$), DES-SA ($+$), RANS-SA using Fluent ($\circ$) and DNS [45] ($\bullet$). (a) Mean velocity profile. (b) Mean turbulent viscosity. (c) Mean shear stress profile: τ_{xy}^{mod} (dash), $-\langle u'v' \rangle$ (plain).

TABLE 7.1 – Overview of computational cost for simulations performed in Section 7.3. The Reynolds number, grid size, number of processors, and wall time are given for each case. All simulations were performed on an SGI Altix 3700 SMP, using Itanium 2 processors at 1.3GHz.

	Re_τ	Grid size	Procs	Time (hours)
DNS	180	$128 \times 129 \times 128$	8	20
LES	395	$80 \times 65 \times 64$	4	15
DES	2000	$64 \times 33 \times 32$	4	5
RANS	2000	$(32) \times 65 \times 1$	1	0.25

- The DNS simulation is the most expensive while at the same time it is also at the lowest friction Reynolds number. The advantage being that it gives direct access to all turbulent quantities.
- The cost of the LES is certainly lower than DNS for a moderate Re_τ increase. However, in terms of modeling and quality of the results, they are almost on par with DNS.
- A significant reduction of the computational cost and increase of Re_τ is achieved by DES. The results for the turbulent channel flow are disappointing due to the “buffer layer” between the region in which the turbulent stress is modeled and the region where it is resolved. One should however remember that DES was not designed for this type of grid/flow configuration.
- RANS has a very low computational cost, especially in this trivial geometry; it could even be lower: we here used $N_x = 32$, while we could have used $N_x = 1$. In this case the results are in excellent agreement with DNS. But this is essentially because the model was designed to do so! In more complex configurations, such as cases with massive separations, RANS results can miss essential flow features. Finally, RANS provides limited access to turbulent quantities.

This illustration shows that the requirements for DES (or any type of hybrid RANS/LES) for engineering type applications (i.e., high Reynolds number

and complex geometries) are well in reach of currently available computational resources. Furthermore, the success of DES in particular for applications it was precisely designed for (i.e., flows with attached boundary layers and massive separation regions) are very encouraging [92, 16, 17]. However, as was illustrated here, there is still a lot of room for improvement. In the next section, we further investigate a new hybrid RANS/LES coupling that was recently developed by Templeton *et al.* [91].

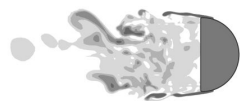
7.4 Investigation of near-wall dynamic coupling of LES with RANS turbulence models

This investigation is the result of work done in close collaboration with G. Kalitzin, G. Medic and J. Templeton when visiting the Flow, Physics and Computation division at Stanford University. It is essentially a compilation of two articles that were originally published in [62, 31].

A novel concept for the near-wall treatment of LES has been presented in [50, 63, 91]. In this approach a RANS-like eddy viscosity is imposed in the near-wall region and is corrected with the resolved turbulent stress. Combined with a wall stress model, it has been applied successfully to channel flow at high Reynolds numbers. The near-wall treatment has also been applied with wall-parallel coarsening, which is further investigated here. In [50, 63, 91], the RANS eddy-viscosity was precomputed from the RANS equation for channel flow using the averaged velocity profile from the LES and stored in a look-up table.

In the present investigation, the RANS-LES coupling formulation proposed in [50, 63, 91] is adapted for various RANS turbulence models. The RANS-LES coupling is presented in Section 7.4.1.

In Section 7.4.2, the results computed with an LES-based table for the RANS eddy-viscosity are compared to computations with tables precomputed using $k-\omega$ [96] and Spalart-Allmaras [89] models. Then it is shown that the RANS eddy-viscosity can also be obtained by solving the RANS turbulence model equations simultaneously with the LES (Section 7.4.3). The analysis presented in these sections is carried out for the channel flow at $Re_\tau = 395$. The coupled RANS-LES computations and the reference LES computation all use the same wall-resolved grid. This is obviously not the final purpose of



RANS-LES coupling (as there are no computational savings in this case), but rather a proof of concept.

In Section 5, the proposed formulation is applied on grids coarsened in the wall-parallel directions, to investigate the possibility of computational savings. The results are presented for the channel flow at $Re_\tau = 950$ and are compared with DNS results of [37].

Finally, the RANS-LES coupling approach is applied to a serpentine passage and compared with the DNS by Laskowski [58] at $Re_\tau = 180$. This is again a “proof of concept” case as the Reynolds number is moderate and the grid sufficiently fine to perform full LES. It is however an interesting case due to the increased geometrical complexity, which leads to more complex flow features.

7.4.1 Near-wall RANS-LES coupling

The basic idea of the current approach goes as follows. In the near-wall region, the grid is not sufficiently resolved to capture enough turbulent fluctuations and thus to resolve a significant portion of the Reynolds stresses there to enable wall-resolved LES. Thus, instead, one solves the RANS equations in the near wall region by simply replacing the SGS viscosity by the RANS eddy-viscosity (up to a user defined wall-normal distance). This is the basic idea of RANS-LES coupling approaches in general. It is also essentially what is done in DES, without however directly acting on the eddy-viscosity but rather by modifying the eddy-viscosity transport equation. If this approach is applied “as is”, problems occur in the vicinity of the switching region (as observed for the DES of the channel flow in Section 7.3.3). In the LES region, velocity fluctuations are missing: essentially, because of delays in their generation by instabilities. In other words, the resolved Reynolds stresses deriving from velocity fluctuations (“LES content”) have not replaced the modeled Reynolds stresses yet. The problem can also be seen from the RANS side: due to the neighbor LES region, velocity fluctuations penetrate the RANS region. Thus, in the RANS region, the Reynolds stresses are accounted for twice: once through the RANS eddy-viscosity and once through the “resolved” turbulent velocity fluctuations. The idea is here to correct the RANS eddy-viscosity to account for those resolved stresses.

The Navier-Stokes equations given in Eqs. (7.1) and (7.2) are solved throughout the entire computational domain with an effective viscosity, $\nu_e = \nu + \nu_t^{\text{sgs}}$. In the near-wall region however, the instantaneous SGS eddy-viscosity, ν_t^{sgs} ,

is replaced by a RANS eddy-viscosity, ν_t^{rans} , corrected in order to take the resolved turbulent stress into account.

We here briefly describe how the corrected RANS eddy-viscosity, ν_t^{corr} , is obtained; additional discussion can be found in [50, 63, 91]. Let's compare the averaged LES and RANS equations for streamwise velocity:

$$\frac{d}{dy} \left(-\hat{u}\hat{v} + (\nu + \nu_t^{\text{sgs}}) \frac{d\hat{u}}{dy} \right) = \frac{d\bar{p}}{dx}, \quad (\text{LES}) \quad (7.13)$$

$$\frac{d}{dy} \left((\nu + \nu_t^{\text{rans}}) \frac{d\bar{u}}{dy} \right) = \frac{d\bar{p}}{dx}, \quad (\text{RANS}) \quad (7.14)$$

where $\hat{u}$ and $\hat{v}$ are the streamwise and wall-normal velocity components of the LES, and $\overline{(\cdot)}$ is the RANS averaging operator. Thus by identification, we obtain:

$$-\overline{\hat{u}\hat{v}} + (\nu + \nu_t^{\text{sgs}}) \frac{d\hat{u}}{dy} = (\nu + \nu_t^{\text{rans}}) \frac{d\bar{u}}{dy} \quad (7.15)$$

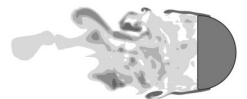
Assuming that ν_t^{sgs} and the wall-normal velocity gradient are independent, the mean subgrid stress can be written as:

$$\overline{\nu_t^{\text{sgs}} \frac{d\hat{u}}{dy}} \simeq \overline{\nu_t^{\text{sgs}}} \frac{d\hat{u}}{dy}. \quad (7.16)$$

For an ideal LES and RANS, the velocity gradients in Eq. (7.15) are equal. With these assumptions, Eq. (7.15) provides a relation between the RANS and LES eddy-viscosities and the resolved turbulent stress: $\overline{\nu_t^{\text{sgs}}} = \nu_t^{\text{rans}} + \hat{u}\hat{v} / \frac{d\hat{u}}{dy}$. Finally, in the RANS region, we will use the corrected RANS eddy viscosity, ν_t^{corr} , given by:

$$\nu_t^{\text{corr}} = \nu_t^{\text{rans}} + \overline{\hat{u}\hat{v}} / \frac{d\hat{u}}{dy}. \quad (7.17)$$

Since $\overline{\hat{u}\hat{v}}$ and $d\hat{u}/dy$ have opposite signs, the second term on the right hand side of equation (7.17) is negative. Thus in the presence of turbulent fluctuations, the near-wall LES viscosity, ν_t^{corr} , is always less than the RANS viscosity, ν_t^{rans} , the difference being a dynamic correction for the resolved fluctuations. The RANS eddy-viscosity can be obtained from either precomputed look-up tables or from a simultaneous solution of RANS turbulence models. The velocity gradient, $d\hat{u}/dy$, and the turbulent stress, $-\overline{\hat{u}\hat{v}}$, come from the LES. The averaging operator can either be plane- or time-averaging. When using this approach, it is necessary to clip the eddy-viscosity (as is standard practice when using an SGS model, where the eddy-viscosity is clipped whenever its value drops below zero). Here, the eddy-viscosity is clipped at the level of the SGS model.



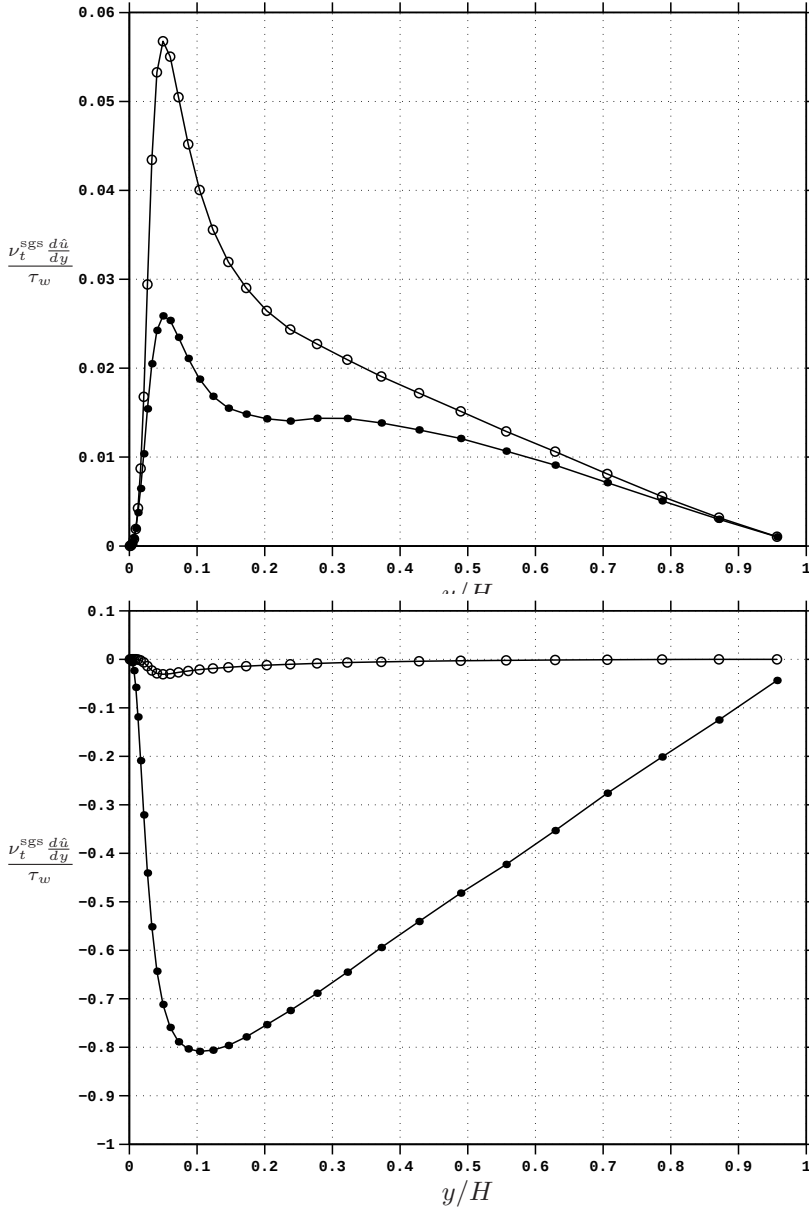


FIGURE 7.7 – LES at $Re_\tau = 395$ using WALE. **Top:** Mean modeled shear stress, $\frac{\nu_t^{\text{sgs}} \frac{d\hat{u}}{dy}}{\tau_w}$ (●), and approximate mean modeled shear stress, $\frac{\bar{\nu}_t^{\text{sgs}} \frac{d\hat{u}}{dy}}{\tau_w}$ (○). **Bottom:** Mean cross-correlation velocity fluctuations, $\frac{\hat{u}\hat{v}}{\tau_w}$ (●), and mean cross-correlation of the velocity gradient fluctuations and the subgrid-scale viscosity fluctuations, $\frac{(\nu_t^{\text{sgs}})' \frac{d\hat{u}'}{dy}}{\tau_w}$ (○).

In order to assess the actual validity of the assumption made in Eq. 7.16, we evaluated both sides of the approximate identity. Figure 7.7 shows the results obtained from the wall resolved LES at $Re_\tau = 395$ (WALE) presented in Section 7.3.2. It can be seen that the difference between the actual modeled shear stress, $\overline{\nu_t^{sgs} \frac{d\tilde{u}}{dy}}$, and the approximate one, $\overline{\nu_t^{sgs} \frac{d\tilde{u}}{dy}}$, is quite large. The difference between both curves is the cross-correlation function of the velocity gradient fluctuations and the subgrid-scale viscosity fluctuations, $\overline{(\nu_t^{sgs})' \frac{d\tilde{u}'}{dy}}$. The exact relation between the LES subgrid scale viscosity and the RANS turbulent viscosity is thus given by:

$$\overline{\nu_t^{sgs}} = \nu_t^{rans} + \frac{\overline{\hat{u}\hat{v}} - \overline{(\nu_t^{sgs})' \frac{d\tilde{u}'}{dy}}}{\frac{d\tilde{u}}{dy}} \quad (7.18)$$

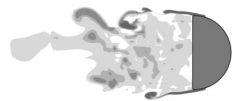
Thus in Eq. 7.17, one would have to take into account an additional correction. However, the additional term is relatively small compared to the turbulent shear stress, $\overline{\hat{u}\hat{v}}$, as can be seen from Figure 7.7. Here and in the original method developed in [50, 63, 91], this additional correction is neglected. Furthermore, it's practical implementation is not straightforward. Nevertheless, it might be worth further investigation.

7.4.2 Precomputed tables for ν_t^{rans}

The RANS/LES coupling is investigated for plane channel flow at $Re_\tau = 395$. The channel dimensions are $2\pi h \times 2h \times \pi h$. A wall-resolved grid used in the computations consist of $80 \times 64 \times 64$ cells in the streamwise, wall-normal and spanwise directions, respectively, with the first cell centers at $y^+ = 0.3$.

The RANS eddy-viscosity, ν_t^{rans} , used in Eq. (7.17) to compute the near-wall eddy-viscosity, ν_t^{corr} , can be stored in precomputed look-up tables. These look-up tables are structured using a method similar to [49] for RANS wall functions.

The tables are constructed using several approaches. The first one uses an averaged velocity profile obtained from the wall-resolved LES of channel flow at $Re_\tau = 395$. The look-up table for the eddy-viscosity, $\nu_t^+(y^+)$, is obtained from the non-dimensional RANS equation for channel flow $(1 + \nu_t^+) du^+/dy^+ = 1 - y^+/Re_\tau$, where du^+/dy^+ is the gradient of the averaged velocity taken from the LES. This assures that, when this eddy-viscosity is used everywhere, the averaged velocity profile from LES is recovered. Two additional tables were constructed by storing the eddy-viscosity from the results of channel flow



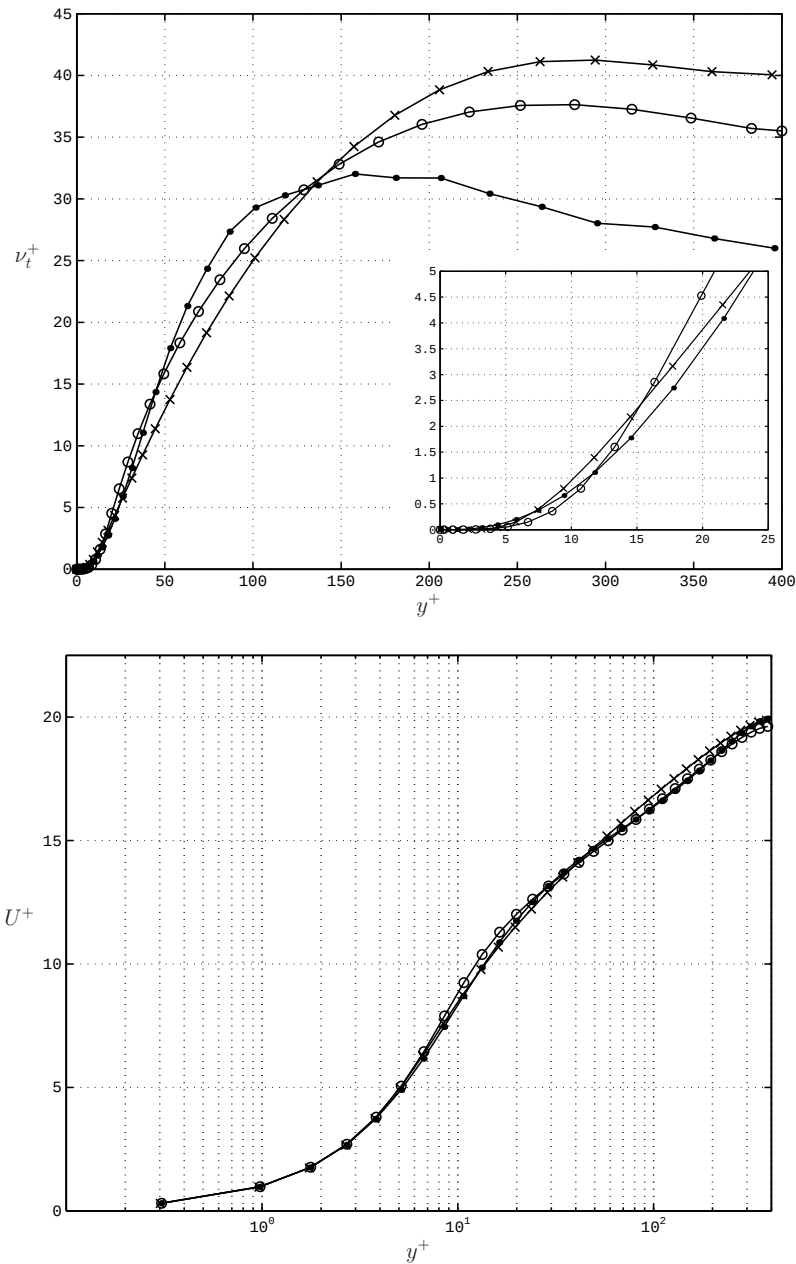


FIGURE 7.8 – Channel flow at $Re_\tau = 395$. **Top:** Precomputed ν_t^{rans} tables obtained from mean WALE LES velocity profile ($\bullet$), SA RANS ($\circ$) and $k-\omega$ RANS turbulence model ($\times$). **Bottom:** Corresponding RANS velocity profiles.

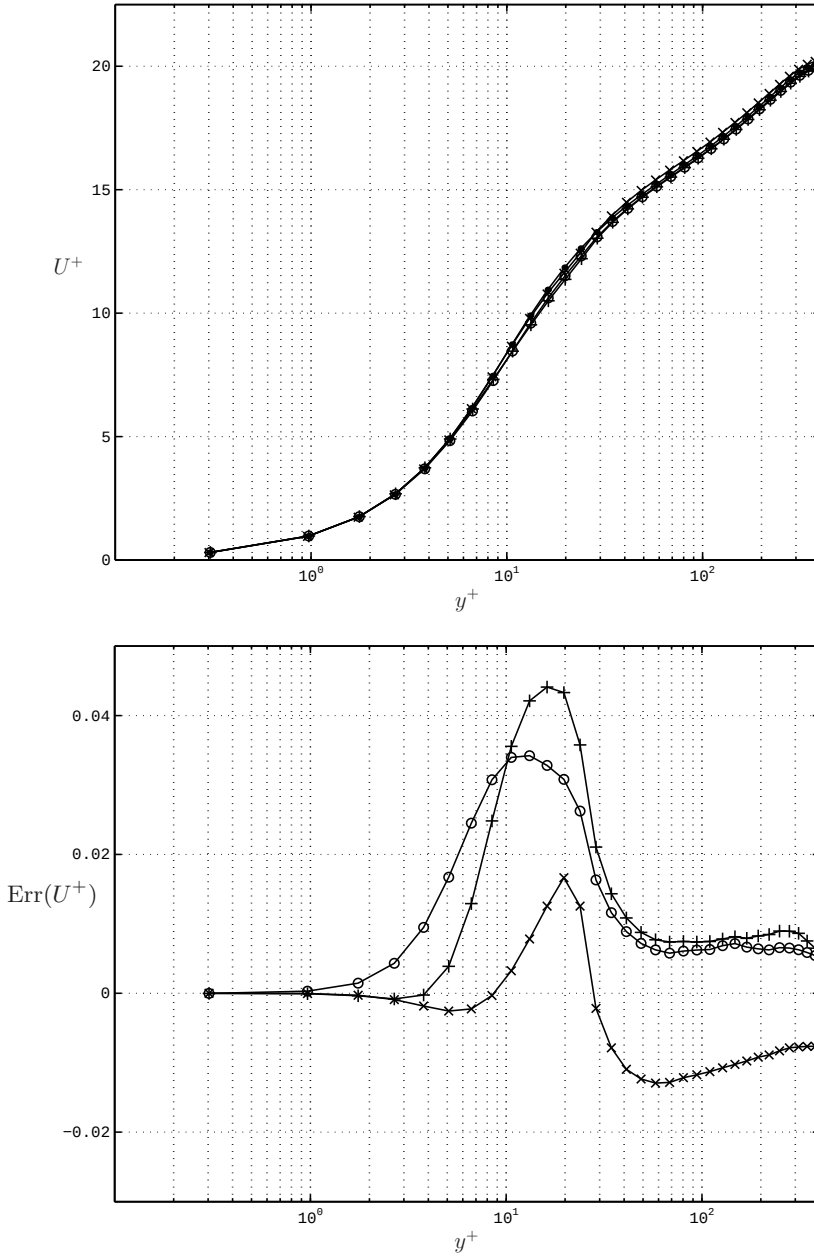
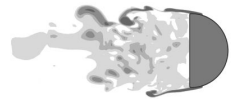


FIGURE 7.9 – Channel flow at $Re_\tau = 395$. RANS-LES using precomputed tables up to $y^+ = 20$ and WALE in LES region: LES WALE table (o), SA table ($\times$), $k-\omega$ table (+), full WALE LES reference ($\bullet$). **Top:** Mean velocity, U^+ . **Bottom:** Relative velocity error, $\text{Err}(U^+) = (U_{\text{LES}}^+ - U^+)/U_{\text{LES}}^+$.



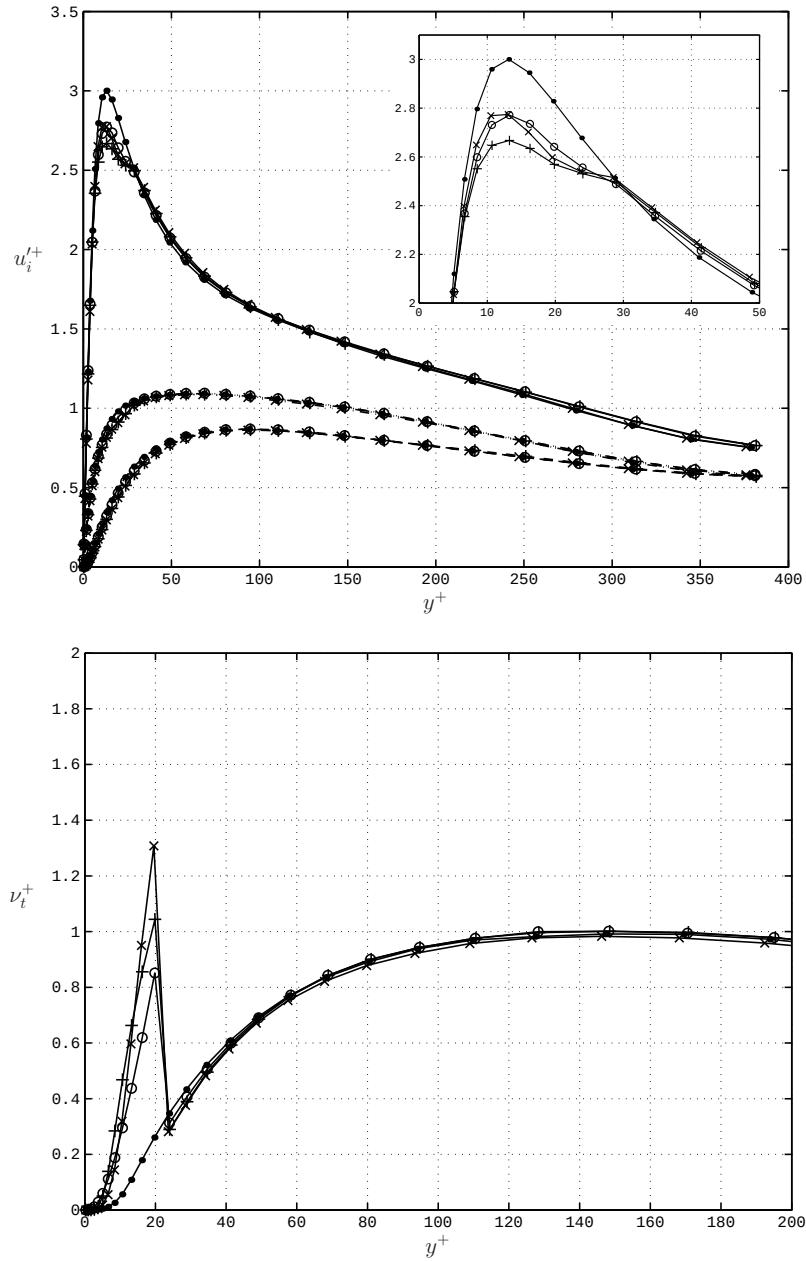


FIGURE 7.9 – (cont.) Channel flow at $Re_\tau = 395$. RANS-LES using precomputed tables up to $y^+ = 20$ and WALE in LES region: LES WALE table ($\circ$), SA table ($\times$), $k-\omega$ table ($+$), full WALE LES reference ($\bullet$). **Top:** RMS velocity fluctuations. **Bottom:** Mean turbulent eddy-viscosity, ν_t .

RANS computations using Spalart-Allmaras and k - ω models (described in Appendix E). All three RANS eddy-viscosity profiles are shown in Fig. 7.8. Note that the eddy-viscosities are quite different both in the center of the channel and near the wall (see also zoom in Fig. 7.8).

These tables can be used throughout the channel to compute a RANS solution. The corresponding velocity profiles are shown on the bottom of Fig. 7.8 with some differences observed in the buffer and the logarithmic layer.

Next, these RANS eddy-viscosity tables are used to compute ν_t^{corr} in the near-wall region of the RANS-LES simulation. This near-wall region extends up to $y^+ = 20$ (first 12 cells above the wall). Results for all three tables and a traditional LES are presented in Fig. 7.9. The SGS model for all cases is the WALE model. The velocity profiles computed with all three tables compare well with the full LES. Interestingly, the spread in the velocity profiles for all four computations is smaller than for the three RANS computations presented in Fig. 7.8. The corrected eddy-viscosity ν_t^{corr} dynamically adjusts independently of the near-wall RANS table.

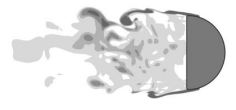
A closer look at the relative error $(U_{\text{LES}}^+ - U^+)/U_{\text{LES}}^+$ (i.e., the difference between the RANS-LES solutions and the full LES solution) reveals that the LES-based table has the smallest error in the logarithmic region. However, the error for all computations is relatively small (less than 1% in the logarithmic layer), which suggests that the use of eddy-viscosity coming from RANS turbulence models is appropriate to be used with this approach. Note that the error is largest where the switch from the near-wall treatment to full LES occurs, as shown in the plots for the eddy-viscosity in the same figure. This also affects the peak in the streamwise velocity fluctuations, see Fig. 7.9.

7.4.3 Dynamic coupling with RANS model equations

This section describes a dynamic coupling procedure for LES with RANS turbulence model equations. A RANS turbulence model is solved simultaneously with the LES simulation. The LES provides an averaged velocity $\bar{u}$ that is used to compute the convection and production terms in the RANS turbulence model equations. For example, for the k - ω model, the equations are:

$$\frac{\partial k}{\partial t} + \bar{u}_j \frac{\partial k}{\partial x_j} = P_k - C_\mu \omega k + \frac{\partial}{\partial x_j} \left((\nu + \sigma_k \nu_t) \frac{\partial k}{\partial x_j} \right) \quad (7.19)$$

$$\frac{\partial \omega}{\partial t} + \bar{u}_j \frac{\partial \omega}{\partial x_j} = \gamma \frac{\omega}{k} P_k - \beta \omega^2 + \frac{\partial}{\partial x_j} \left((\nu + \sigma_\omega \nu_t) \frac{\partial \omega}{\partial x_j} \right) \quad (7.20)$$



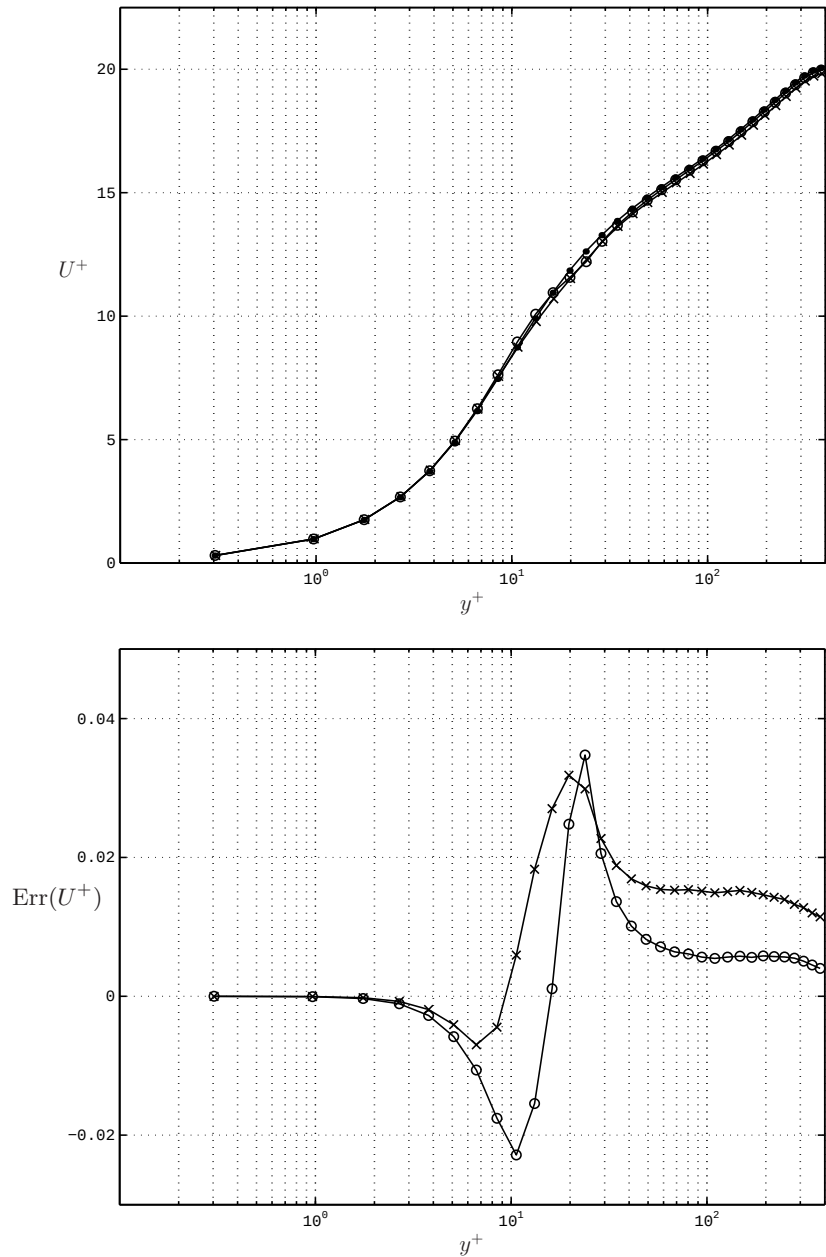


FIGURE 7.10 – Channel flow at $Re_\tau = 395$. RANS-LES using dynamically coupled RANS model equations up to $y^+ = 20$ and WALE in LES region: SA model ($\circ$), $k-\omega$ model ($\times$), full WALE LES reference ($\bullet$). **Top:** Mean velocity, U^+ . **Bottom:** Relative velocity error, $\text{Err}(U^+) = (U_{\text{LES}}^+ - U^+)/U_{\text{LES}}^+$.

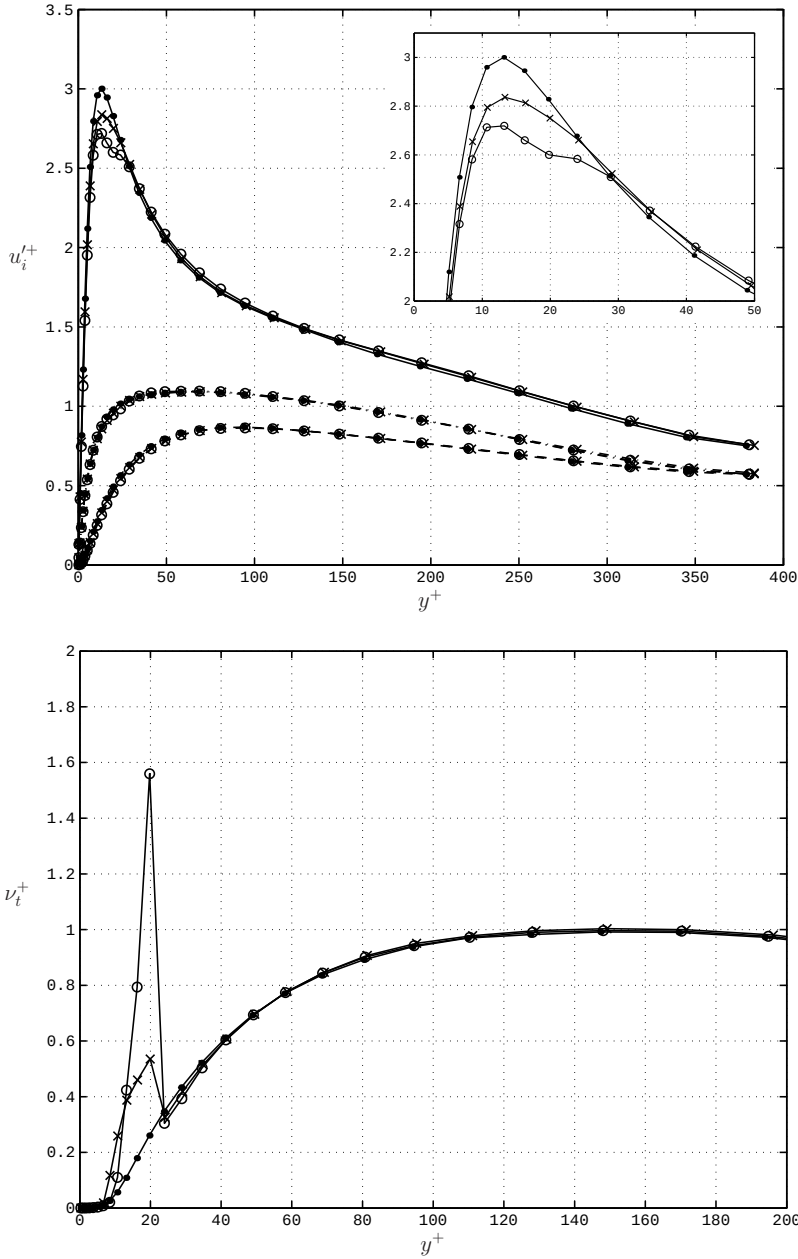
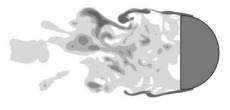


FIGURE 7.10 – (cont.) Channel flow at $Re_\tau = 395$. RANS-LES using dynamically coupled RANS model equations up to $y^+ = 20$ and WALE in LES region: SA model ($\circ$), $k-\omega$ model ($\times$), full WALE LES reference ($\bullet$). **Top:** RMS velocity fluctuations. **Bottom:** Mean turbulent eddy-viscosity, ν_t .



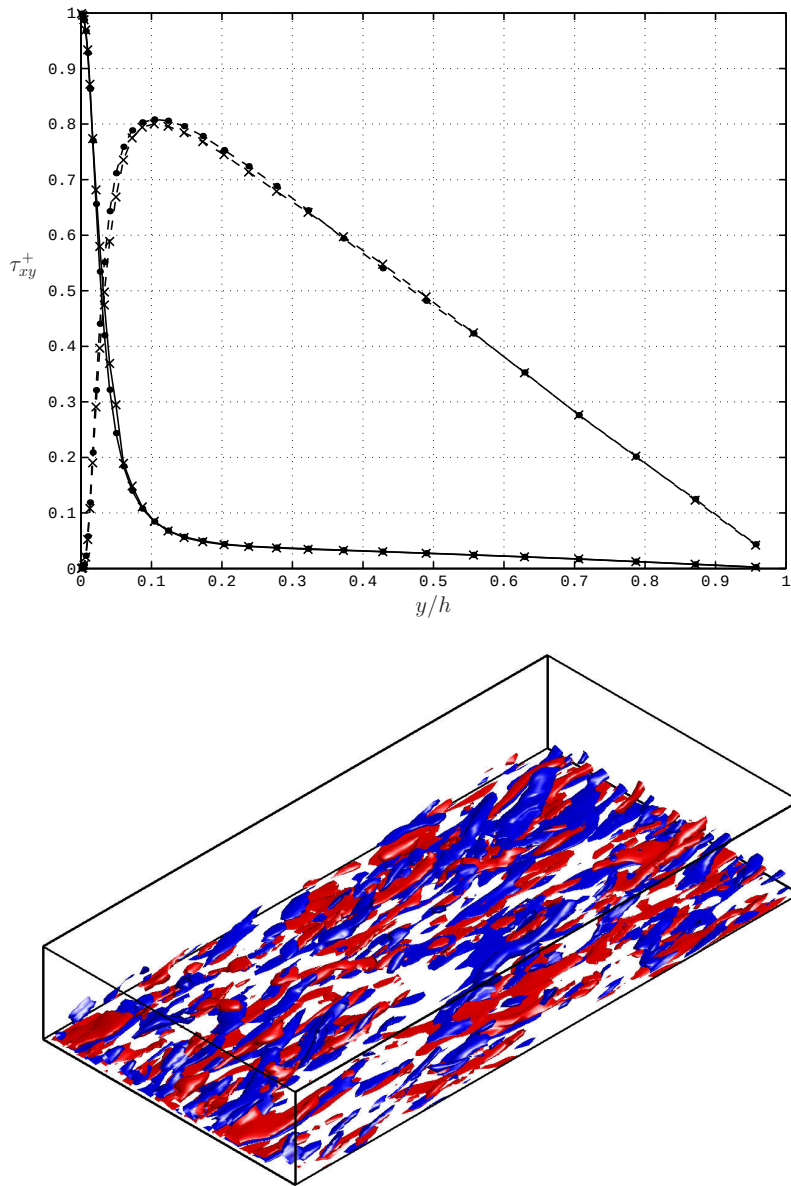


FIGURE 7.11 – Channel flow at $Re_\tau = 395$. RANS-LES using dynamically coupled RANS model equations up to $y^+ = 20$ and WALE in LES region. $k-\omega$ model ($\times$), full WALE LES reference ($\bullet$). **Top:** Mean resolved turbulent stress (dash) and mean viscous and modeled turbulent stress (plain). **Bottom:** Isosurfaces of instantaneous streamwise vorticity in half channel.

$$P_k = 2 \nu_t S_{ij} S_{ij}, \quad S_{ij} = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right) \quad (7.21)$$

In the case of channel flow, we use plane-averaging to compute $\bar{\bar{u}}$.

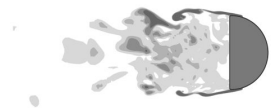
The RANS turbulence model provides ν_t^{rans} which is then corrected with the resolved turbulent stress using equation (7.17) and used in the near-wall region in the LES.

Results for LES dynamically coupled with Spalart-Allmaras and $k-\omega$ turbulence models are compared to LES in Fig. 7.10. These results differ only slightly from the results obtained with the precomputed tables presented in Fig. 7.9. However, as can be seen from the relative error for U^+ , a stronger variation is observed near the switching point with the dynamic coupling. Interestingly, the dynamic coupling with $k-\omega$ produces a smaller jump in the eddy-viscosity at the switching location when compared to results with the precomputed $k-\omega$ table. The streamwise velocity fluctuations are also slightly closer to the traditional LES.

The stress balance presented in Fig. 7.11 reveals that the resolved turbulent stress computed with the RANS-LES dynamic coupling ($k-\omega$ model equations) in the near-wall region practically coincides with the results from the LES. This indicates that the RANS-LES computation actually falls back to a full LES solution of the flow. This is already a nice property of the method: if the grid is fine enough to perform full LES, it will do so. On the other hand, those RANS-LES computations have no advantage over the full LES computation as the grid size is the same. In Section 7.4.5, we will investigate how the model acts in presence of wall coarsened grids.

The isosurfaces of the instantaneous streamwise vorticity computed using the RANS-LES ($k-\omega$ model equations) are also presented in Fig. 7.11. These plots illustrate that the near-wall treatment retains coherent turbulent structures.

Note that the dynamic coupling of LES with RANS turbulence model equations was also tested without the correction from Eq. (7.17) (i.e., with the full RANS eddy-viscosity), ν_t^{rans} , in the near-wall region. In such computations, the abrupt changes of the mean velocity profiles at the switching location (at $y^+ = 20$) were more pronounced, leading to an excess production of turbulence and instabilities. A result for the Spalart-Allmaras model is presented in Fig. 7.12. This indicates that the correction with the resolved stress in equation (7.17) is a key element in the coupling procedure.



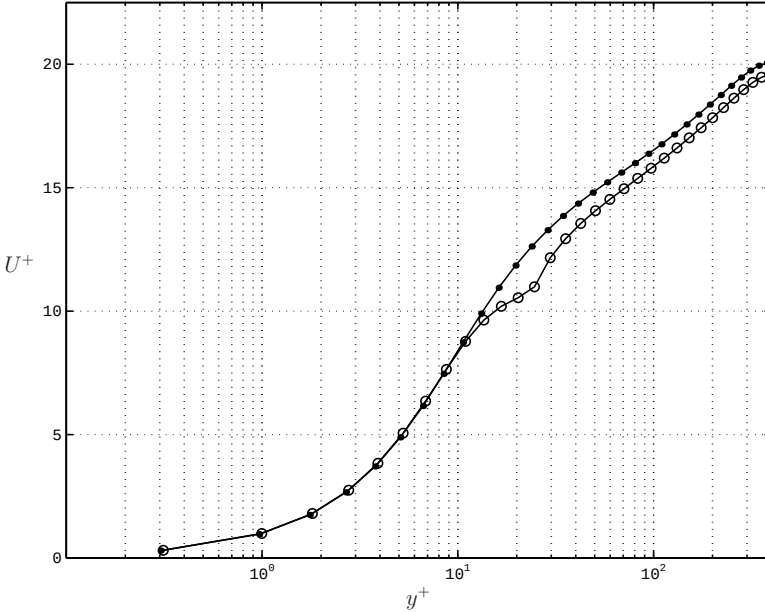


FIGURE 7.12 – Channel flow at $Re_\tau = 395$. RANS-LES using dynamically coupled RANS model equations up to $y^+ = 20$ and WALE in LES region. SA model **without the correction** in equation (7.17) ($\circ$), full WALE LES reference ($\bullet$). Mean velocity, U^+ .

7.4.4 Influence of the subgrid-scale model

The influence of the LES subgrid-scale model on the RANS-LES coupling was also analyzed. Computations were performed using the dynamic (DSM) and “classical” Smagorinsky model (SM) using ν_t^{corr} in the near-wall region (with ν_t^{rans} obtained from precomputed tables). The “classical” Smagorinsky model was intentionally used here with the constant coefficient, $C_s = 0.18$ and with no damping functions near the wall.

The results for the dynamic Smagorinsky model are not presented here as they are very similar to the results obtained with WALE. This was to be expected as both models have a similar behavior when used in full LES.

The “classical” Smagorinsky model is known to fail in the near-wall region, as shown in Fig. 7.13. It significantly underpredicts the mean-velocity profile, as well as the velocity fluctuations. This is due to the incorrect behavior of the eddy-viscosity in the near-wall region. The results significantly improve

if the “classical” Smagorinsky model is used with the RANS-LES near-wall treatment; the eddy-viscosity is now reduced to zero at the wall. Note that the near-wall region was here extended up to $y^+ = 80$: as can be seen in Fig. 7.13, the SM eddy-viscosity is still much too high below that value, and would thus produce erroneous results. These results nevertheless show that we could use the non-dynamic Smagorinsky model for the hybrid RANS-LES as long as the RANS domain is large enough.

7.4.5 Computational savings through wall-parallel coarsening

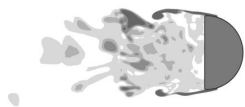
The previous sections offer a proof of concept for the proposed RANS-LES coupling. For this approach to allow computational savings over the traditional wall-resolved LES, it must be applicable to coarser grids. If the no-slip boundary conditions are to be used, only limited coarsening in the wall-normal direction can be applied. Thus, savings must be achieved through wall-parallel coarsening.

Wall-parallel coarsening is analyzed for the channel flow at $Re_\tau = 950$ and compared to DNS results of del Alamo *et al.* [37]. The channel dimensions are $2\pi h \times 2h \times \pi h$. Both a RANS-LES with the $k - \omega$ model computation and a full LES computation are performed on a grid with $64 \times 64 \times 64$ cells and $\Delta y_w^+ = 0.7$. The SGS model in these computations is the WALE model, and the near-wall treatment is applied up to $y^+ = 50$ (approximately $y/H = 0.05$).

The requirements on the grid spacing in the wall-parallel directions can be related to spanwise spacing of streamwise streaks; see for example Kline *et al.* [53]. It has been suggested that the grid spacing in the spanwise direction should not exceed 50 plus units. The grid used here matches that criterion.

The results are presented in Fig. 7.14. When the traditional LES is used on a grid this coarse, the mean velocity is significantly overpredicted. In contrast, the results with the near-wall treatment agree much better with the DNS.

As already observed in previous computations with wall-parallel coarsening at $Re_\tau = 395$ with $32 \times 129 \times 32$ cells, presented in Medic *et al.* [63], the differences in the velocity fluctuations are more significant. The near-wall treatment improves the results when compared to the traditional LES, but the near-wall peak in the streamwise velocity fluctuations is still overpredicted. Obviously, one could not expect that the current method would resolve the



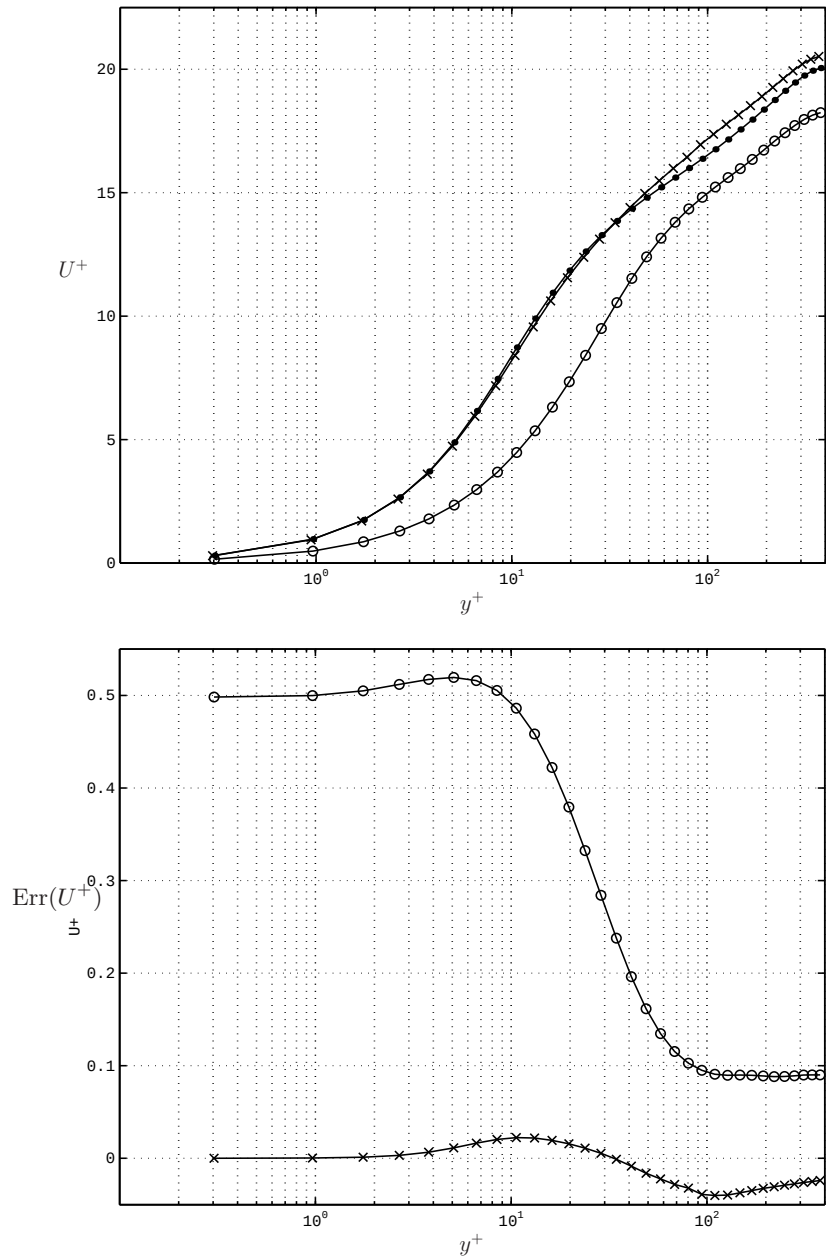


FIGURE 7.13 – Channel flow at $Re_\tau = 395$. RANS-LES using precomputed table up to $y^+ = 80$ and SM in LES region ($\times$), full SM LES ($\circ$), full WALE LES reference ($\bullet$). **Top:** Mean velocity, U^+ . **Bottom:** Relative velocity error, $\text{Err}(U^+) = (U_{\text{LES}}^+ - U^+)/U_{\text{LES}}^+$.

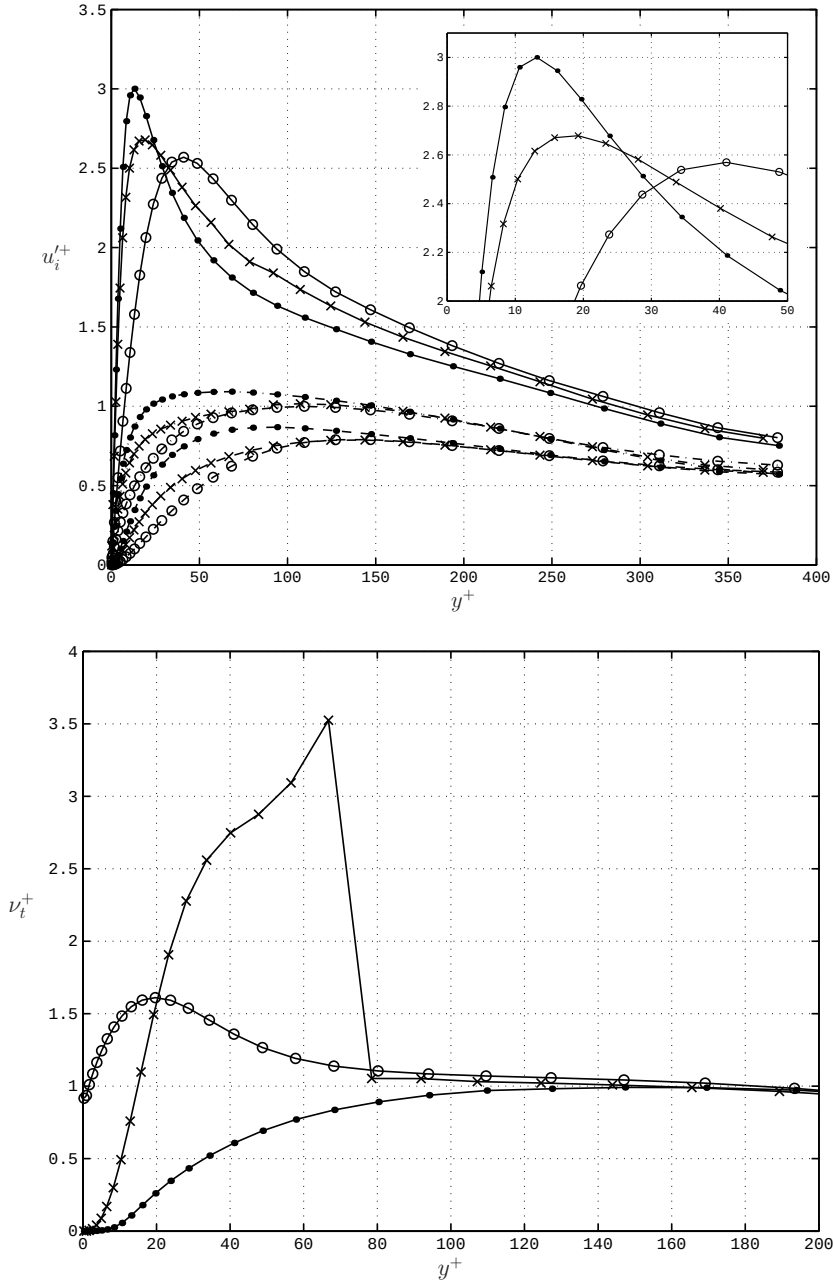


FIGURE 7.13 – (**cont.**) Channel flow at $Re_\tau = 395$. RANS-LES using precomputed table up to $y^+ = 80$ and SM in LES region ($\times$), full SM LES ($\circ$), full WALE LES reference ($\bullet$). **Top:** RMS velocity fluctuations. **Bottom:** Mean turbulent eddy-viscosity, ν_t .

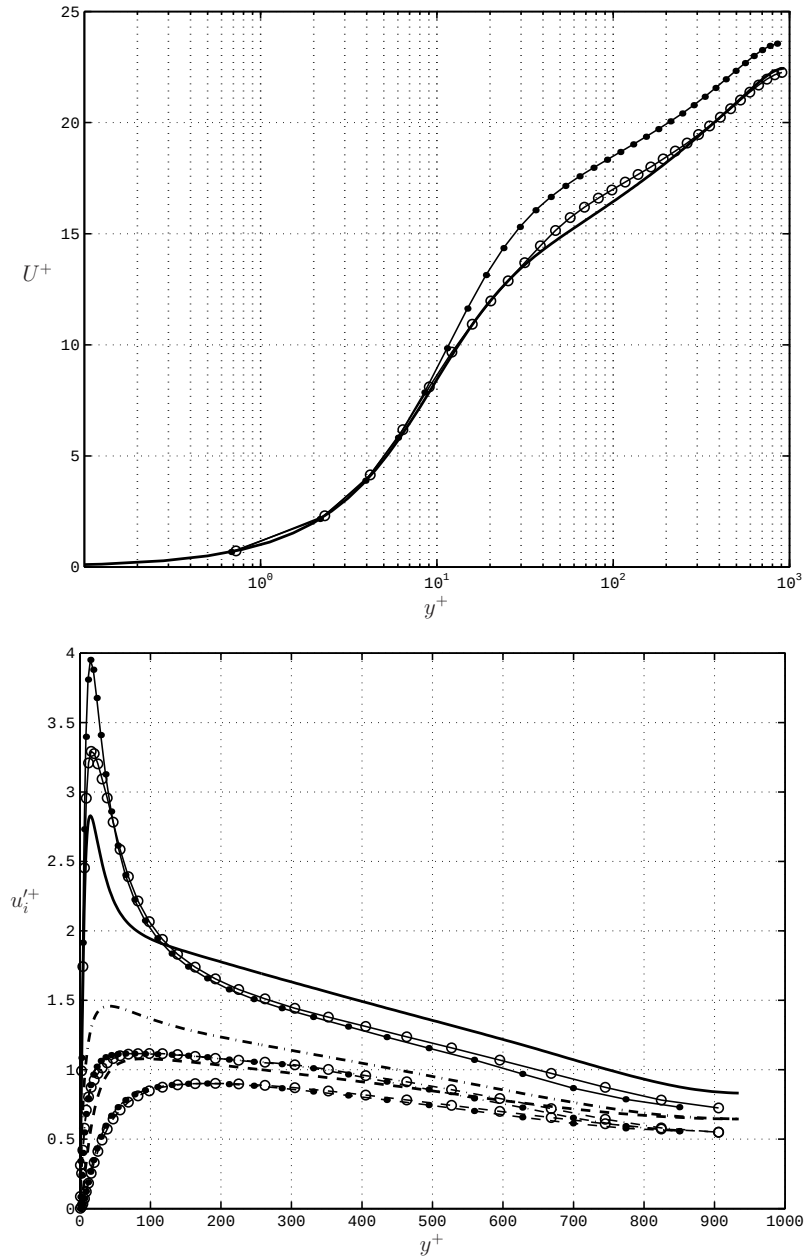


FIGURE 7.14 – Channel flow at $Re_\tau = 950$. RANS-LES using dynamically coupled RANS $k-\omega$ model equations up to $y^+ = 50$ and WALE in LES region ($\circ$), full WALE LES ($\bullet$), DNS reference [37] (plain). **Top:** Mean velocity, U^+ . **Bottom:** RMS velocity fluctuations.

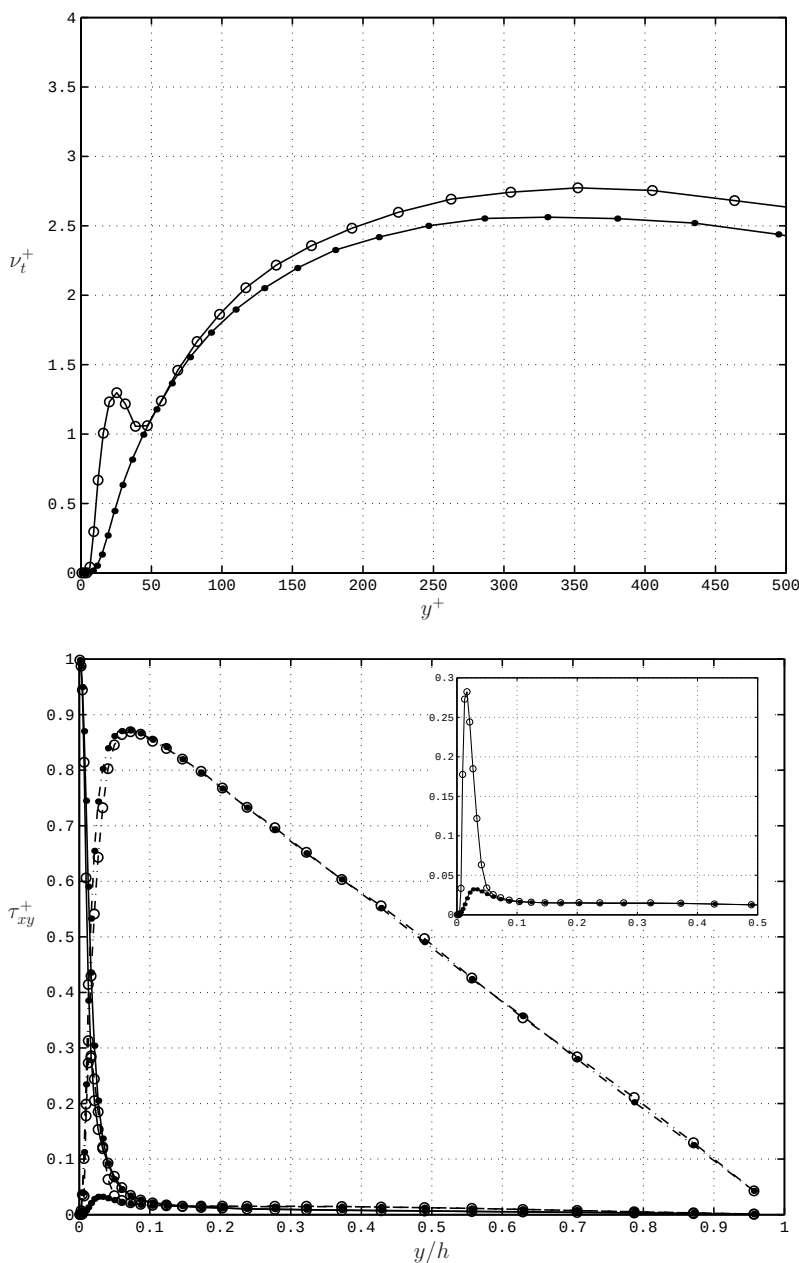


FIGURE 7.14 – **Cont.** Channel flow at $Re_\tau = 950$. RANS-LES using dynamically coupled RANS $k-\omega$ model equations up to $y^+ = 50$ and WALE in LES region ($\circ$), full WALE LES ($\bullet$), DNS reference [37] (plain). **Top:** Mean turbulent eddy-viscosity, ν_t . **Bottom:** Mean shear stress: viscous stress (plain), resolved turbulent stress (dash-dot), modeled turbulent stress (dash). A zoom on the modeled turbulent stress is shown.

correct turbulent features in the near-wall region. However, one might expect that the turbulent fluctuations would be reduced in the near-wall region, due to the RANS treatment. In fact, those fluctuations are related to large scale structures rather than small scale turbulence (this can be concluded from the velocity spectra, not shown). These structures are probably not physical, but are apparently sufficient to obtain sufficient “LES content” such that there is almost no second buffer region as obtained using DES.

It is interesting to observe that the corrected eddy-viscosity, ν_t^{corr} is strongly corrected near the switching location. Nevertheless, the increase of the near-wall eddy-viscosity, compared to the full LES, results in a significant increase in the modeled component of the turbulent stress in the near-wall region (as shown in the detail of the stress balance plot).

At these Reynolds numbers, the savings in CPU time for wall-parallel coarsening are about one order of magnitude, when compared to full LES. This indicates that wall-parallel coarsening used with the proposed RANS-LES approach may be a viable technique for the simulation of high Reynolds number wall bounded flows.

7.4.6 Serpentine passage

The channel flow, analyzed in the previous section, represents a proof of concept for the proposed dynamic coupling of LES with RANS turbulence models in the near-wall region. A more challenging application is the serpentine passage, computed using DNS in [58]. Although periodic in the streamwise and spanwise directions, this problem offers increased complexity over channel flow due to the curved walls, changing pressure gradients and separation. The geometry is typical of internal cooling passages in gas turbine blades. Accurate flow predictions in these passages are necessary to improve their design.

Although this geometry appears relatively simple, it is well-known that RANS turbulence models have difficulties accurately predicting the critical flow features. LES has been shown to significantly improve the results for these flows, but the computational cost is prohibitive for higher Reynolds numbers. The aim of coupling LES with RANS turbulence models in the near-wall region is to allow the use of grids coarsened in wall-parallel directions, while preserving the accuracy of LES.

The geometry and the computational grid are presented in Fig. 7.15. The height of the channel is $2H$, the length of the straight part of the passage

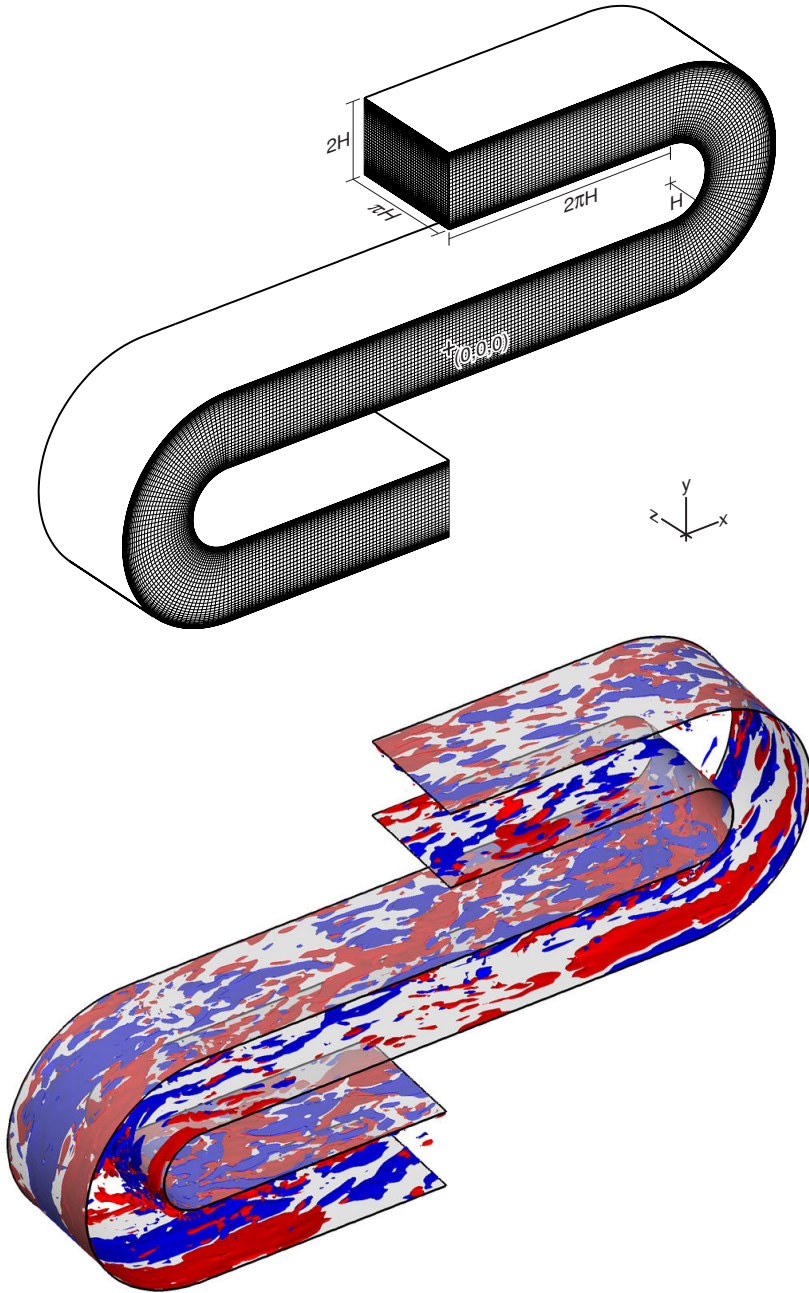


FIGURE 7.15 – Serpentine passage at $Re_\tau = 180$. **Top:** Computational grid for the serpentine passage. **Bottom:** Isosurfaces of the instantaneous streamwise vorticity for full WALE LES (color indicates positive and negative vorticity).

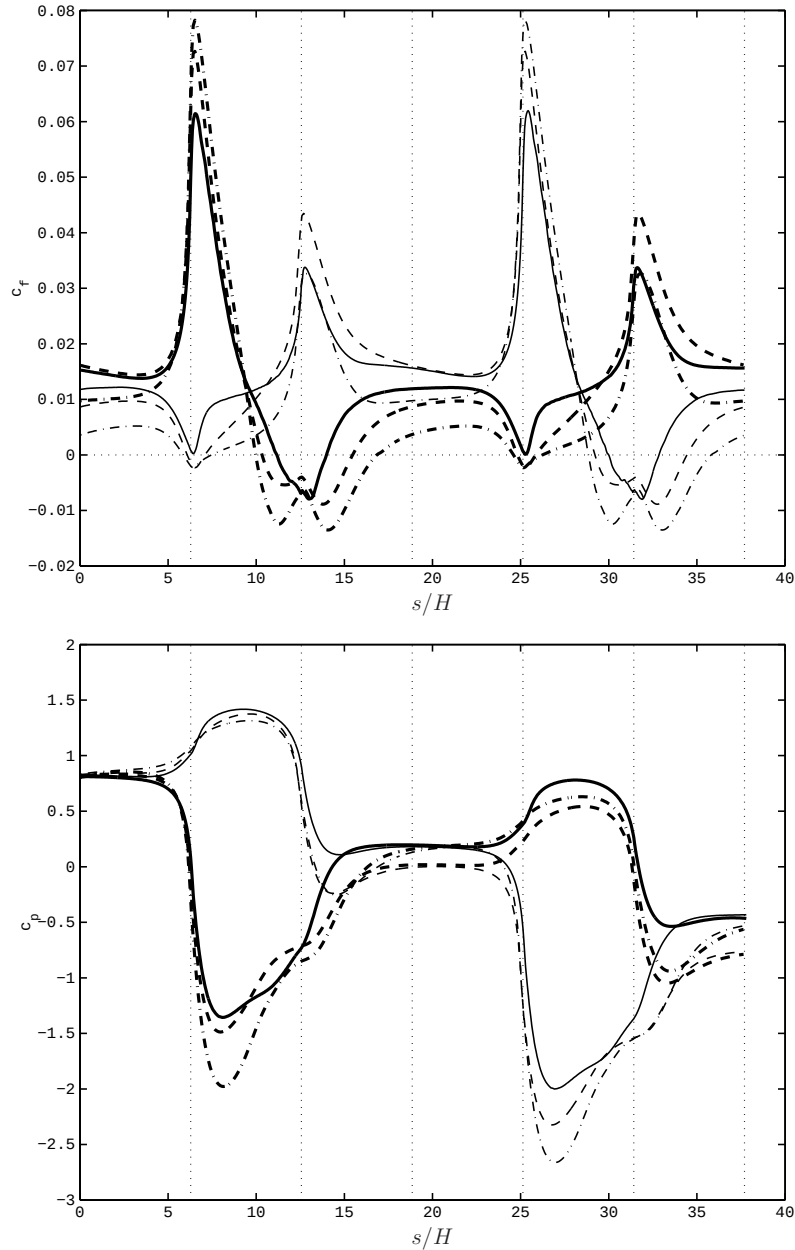


FIGURE 7.16 – Serpentine passage at $Re_\tau = 180$. RANS with SA model (dash-dot), RANS with $k-\omega$ model (dash) and reference DNS [58] (solid) along both inner (thick) and outer walls (thin). **Top:** Skin friction. **Bottom:** Pressure coefficient.

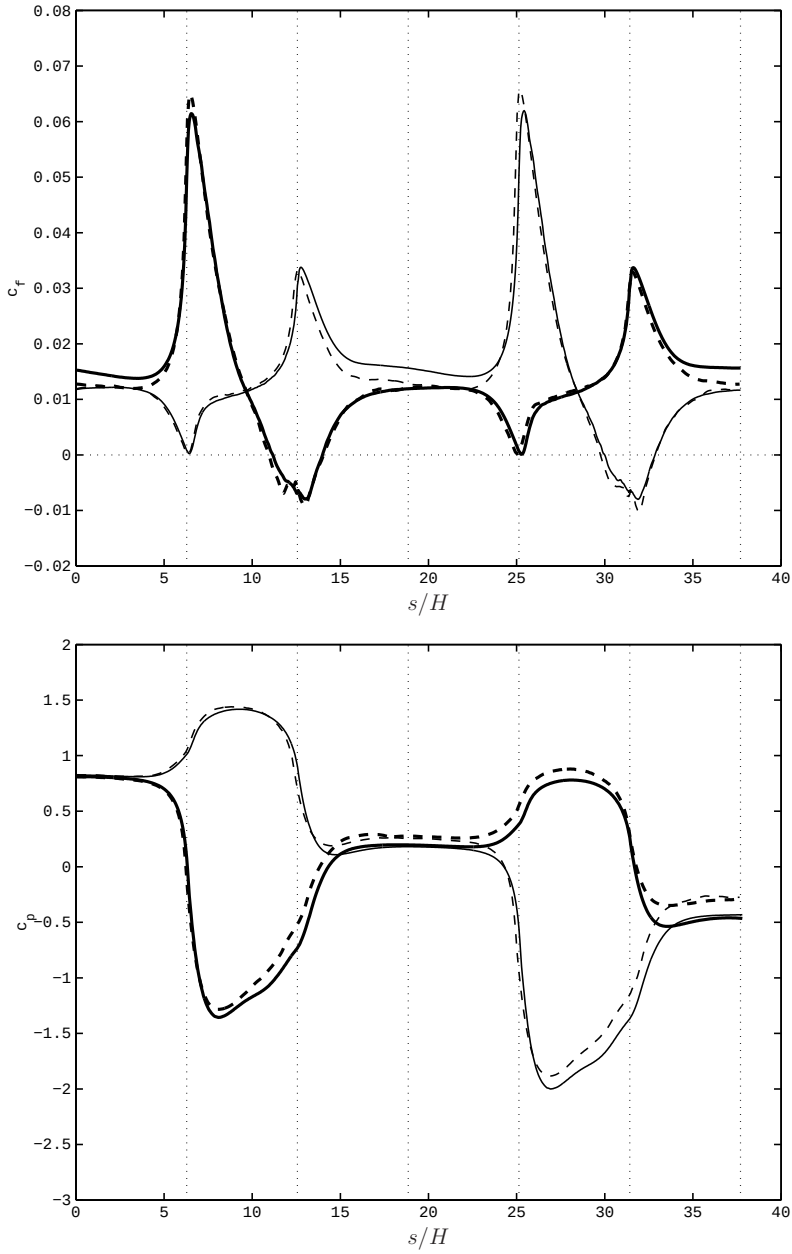
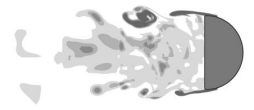


FIGURE 7.17 – Serpentine passage at $Re_\tau = 180$. WALE LES (dash) and reference DNS [58] (solid) along both inner (thick) and outer walls (thin). **Top:** Skin friction. **Bottom:** Pressure coefficient.



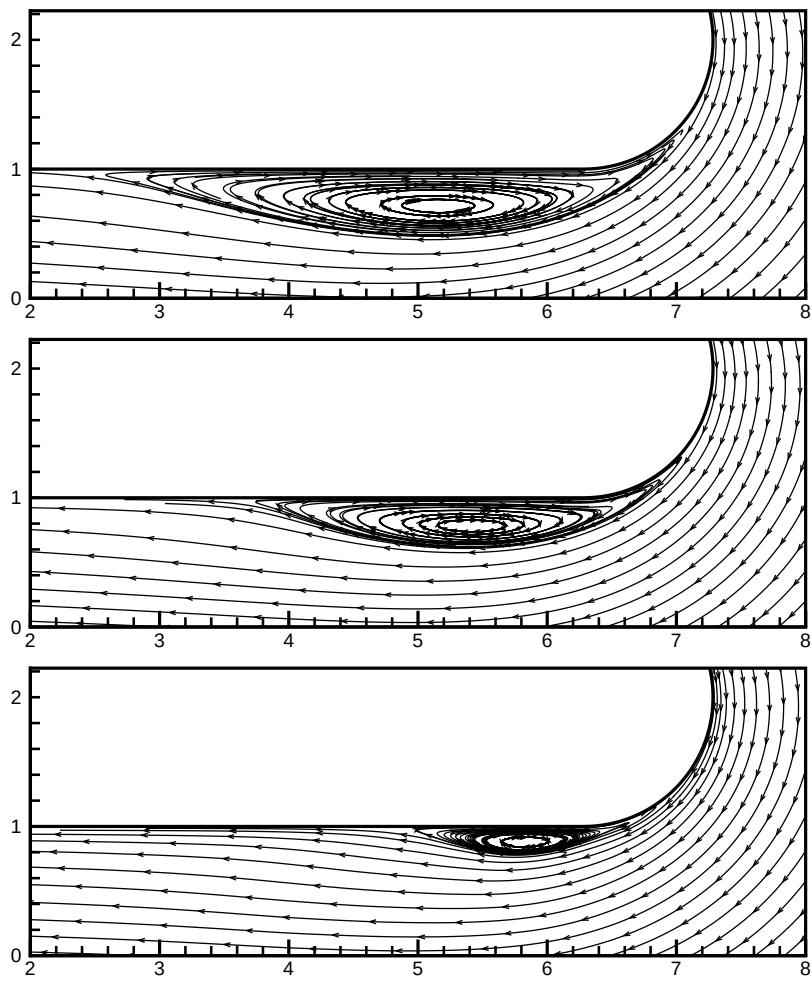


FIGURE 7.18 – Serpentine passage at $Re_\tau = 180$. Mean flow streamlines. **Top:** RANS SA. **Middle:** RANS $k-\omega$. **Bottom:** WALE LES.

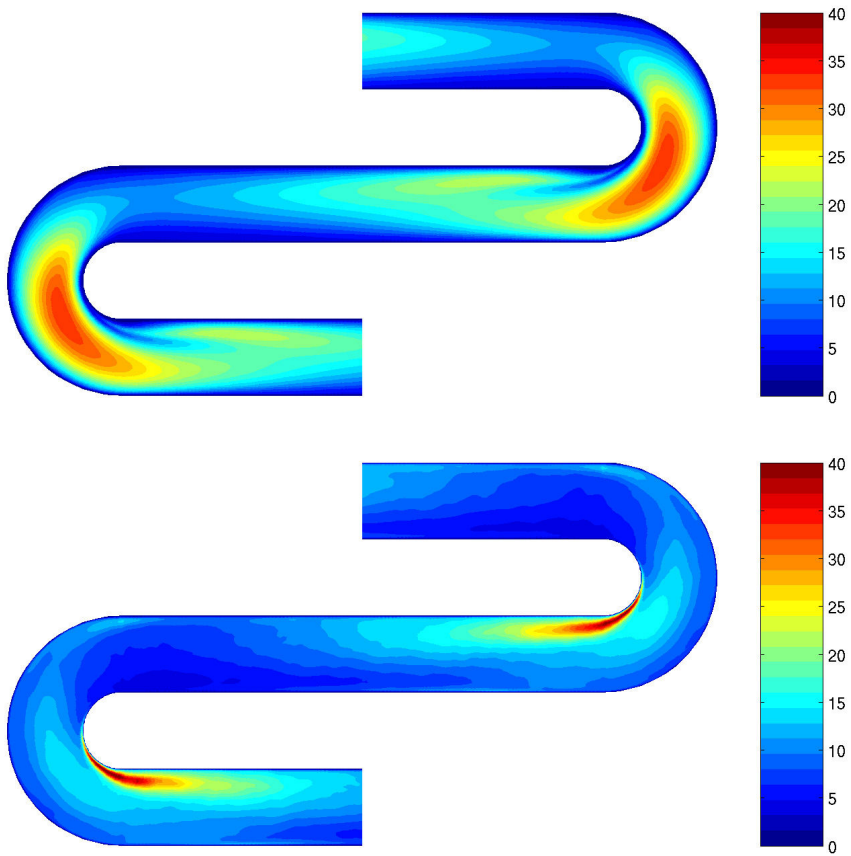


FIGURE 7.19 – Serpentine passage at $Re_\tau = 180$. Turbulent kinetic energy. **Top:** RANS $k-\omega$. **Bottom:** WALE LES.

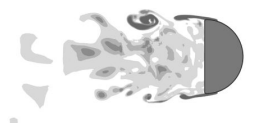


TABLE 7.2 – Mean separation characteristics.

	SA	k - ω	LES	DNS [58]
separation angle	106°	116°	134°	140°
x reattachment	2.00	3.38	4.81	4.86
bubble height	10.4%	7.7%	4.4%	5.0%

upstream of the bend is $2\pi H$ and the inner radius of the bend is H . The width in the spanwise direction was set to π . The Reynolds number based on the bulk velocity, U_b , is $Re_b = U_b H / \nu = 2800$ which corresponds to $Re_\tau = 180$. The computational grid consists of $384 \times 48 \times 48$ cells in the streamwise, wall-normal and spanwise directions, respectively. A two-dimensional version of the same grid is used for RANS computations with 384×48 cells. The average y^+ of the first cell centers is about 0.5 with a maximum value of 1.0.

RANS turbulence models poorly predict this flow, as illustrated by the comparison of skin friction and pressure coefficients computed using the Spalart-Allmaras and k - ω models with the DNS in Fig. 7.16. In contrast, the results computed with LES (WALE SGS model) agree well with the DNS, as shown in Fig. 7.17.

An important feature of this flow is the separation encountered along the inner walls downstream of the bends. The mean separation characteristics are presented in Table 7.2. The corresponding streamlines are shown in Fig. 7.18. Clearly, the RANS models predict the separation too early and are overpredicting the size of the recirculating region.

The discrepancies in the skin friction distribution between RANS and LES can be explained in part by predictions of turbulent kinetic energy. Contours of turbulent kinetic energy computed using the k - ω model and LES are presented in Fig. 7.19.

Next, we consider the near-wall treatment for LES that uses the dynamic coupling with RANS turbulence models (Spalart-Allmaras and k - ω). For the serpentine passage, the equation (7.17) can still be applied as long as the quantities $\overline{\hat{u}'\hat{v}'}$ and $\overline{d\hat{u}/dy}$ are defined in a local coordinate system aligned with the wall. In practice, $\hat{S}_{ij}$ and $\overline{\hat{u}'_i\hat{u}'_j}$ are computed in the general coordinate system and are then expressed in the local coordinate system. In these computations, the RANS eddy-viscosity is obtained from a simultaneous solution of RANS

turbulence models using $\bar{\tilde{u}}$. The averaging for all quantities is performed in the spanwise direction and combined with time-averaging.

The computation using dynamic coupling with the Spalart-Allmaras model was performed on the same wall-resolved grid used for the LES. The near-wall region extends up to $y/H = 0.2$. The results are almost identical to those obtained using full LES. This is again due to the fact that the grid used here is sufficiently fine to perform full LES. The pressure and skin friction coefficients are not shown (as there is visually no difference with the full LES solution). The flow separates at 129° , reattaches at $x = 4.80$ and the bubble height is 5.44%. Thus, the separation is slightly larger than when computed with the LES.

No real conclusions can be drawn from these results, other than the fact that, even in complex geometries and with complex flow features such as separation and reattachments, the RANS-LES coupling is able to fall back to full LES when the grid is sufficiently fine. Which is, again, a very nice property for a RANS-LES approach. However, to investigate the computational advantages of this approach, further investigations should be performed at higher Reynolds numbers and on more challenging grids.

7.5 Summary

The RANS-LES coupling formulation proposed in Templeton *et al.* [91] has been further adapted for use with various RANS and LES models. The formulation consists of imposing a RANS eddy-viscosity dynamically corrected with the resolved turbulent stress near the wall.

The RANS eddy-viscosity can either be precomputed or obtained by solving the RANS turbulence model equations simultaneously. Results obtained for channel flow at $Re_\tau = 395$ indicate that the near-wall treatment with the dynamic coupling with the Spalart-Allmaras and $k - \omega$ model equations is a viable technique. The advantage of the dynamic coupling is in the direct use of the averaged LES velocity field in the solution of the turbulence model equations. Thus, the RANS eddy-viscosity adjusts to the actual LES velocity field.

The influence of the subgrid scale model has also been analyzed for channel flow at $Re_\tau = 395$. In addition to the WALE model, the computations were also performed with both the dynamic and the “classical” Smagorinsky model (with



constant coefficient, C_s). The results obtained with the dynamic Smagorinsky model are very similar to the results obtained with the WALE model. If the near-wall region is extended to $y/h = 0.2$, the computations that employ the “classical” Smagorinsky model also yield good results.

The computational advantage of the proposed RANS-LES approach was demonstrated on wall-resolved grids coarsened in wall-parallel directions. The results for the channel flow at $Re_\tau = 950$ on a grid with $64 \times 64 \times 64$ cells showed a good agreement with the DNS results, while the traditional LES on the same grid performed poorly. These results indicate that the proposed near-wall coupling can be used with wall-parallel coarsening to perform LES at a reduced cost.

A preliminary application of the proposed framework to complex flows was also presented for the flow in a serpentine passage. Given the low Reynolds number and the fact that, in this case, the computational grid is sufficiently resolved to perform full LES, the current application was essentially a proof of concept.

Conclusion

The very general and ambitious objective of this thesis was to further develop the three-dimensional vortex method and make it an efficient tool for simulating high Reynolds number flows in external aerodynamics. As the time has come to conclude this work, we have to accept that the vortex method, as we leave it, is not capable of simulating (for example) the flow past a sphere in the supercritical regime nor is it able to predict the “drag crisis” for this flow... But we should rather say “is not capable *yet*”, because a number of challenges presented by this type of flows have already been addressed. We therefore believe that, sometime in the not so distant future, the vortex method, combined with an Eulerian solver for the near-wall regions (as further developed in this thesis), will be used as a tool to investigate the unsteady aerodynamics of flows with massive separation and for engineering-type applications.

Accomplishments

We first investigated flows that could be modeled using prescribed separation regions. The idea was to study the capabilities of the vortex method to handle high Reynolds number turbulent wake flows without actually resolving the detailed evolution of the boundary layer. The quantitative results obtained using this approach were physically close to reality (e.g., vortical structures in the wake of a hemisphere, drag coefficient) yet still limited. It however allowed



to address a number of interesting issues such as: the use of subgrid-scale models for LES in vortex methods, the vorticity divergence problem, and the treatment of outflow boundary conditions as well as the treatment of realistic viscous ground effect conditions (for the flow past a simplified truck).

One of the major contributions of this work is probably the development of a new hybrid Eulerian-Lagrangian approach. We have presented an approach that combines an Eulerian grid-based solver for the near-wall regions and a Lagrangian particle solver for the wake. This approach suppresses the biggest limitation of vortex methods when it comes to simulating high Reynolds number flows, as it allows to efficiently handle the boundary layer regions using the Eulerian solver. The innovation comes from the fact that the Lagrangian domain covers the whole computational domain, yet is under-resolved in the near-wall region. The Eulerian grid-based solver is used to capture the detailed dynamics of the boundary and separation regions. Due to the fact that the Lagrangian domain covers the whole computational domain, the boundary conditions for the Eulerian domain can be retrieved easily. The approach was first developed in 2-D and using the vorticity-velocity formulation in both domains. It was then successfully extended to the case where the “classical” velocity-pressure formulation is used in the Eulerian domain. The extension to 3-D could not be demonstrated in this work but communication tools were already implemented in order to couple an existing 3-D Eulerian Finite Volume code (used in Chapter 7) with the 3-D Lagrangian code. The next step being essentially to run the code and validate the coupled approach in 3-D.

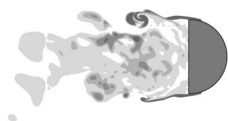
Another important contribution is the development of a new conservative redistribution scheme for hierarchically refined grids. Our approach allows to build arbitrarily refined grids. It addresses another decisive issue for the use of vortex methods for high Reynolds number flows as it provides the required control on the spatial accuracy.

The capabilities of the redistribution method on hierarchically refined grids were successfully demonstrated on the flow past a hemisphere at $Re = 3,000$. The grid refinement procedure allowed to precisely control the distribution of computational elements. Our simulation was able to correctly capture the laminar detached shear layers and their transition to turbulence through Kelvin-Helmoltz instabilities. The results were compared, from a qualitative and quantitative point of view, with experimental results obtained for the flow past a hemisphere at $Re = 3,000$ (LIF visualizations) and $Re = 55,000$ (LDV measurements). In this chapter, the vorticity divergence problem was also success-

fully addressed by regularly re-projecting the vorticity field onto a divergence free basis (by solving a Poisson equation).

The capabilities of vortex methods were further demonstrated through the investigation of aircraft wake vortices in ground effect. These results show an application of vortex methods to a complex, turbulent, “real-life” problem. The 3-D numerical method used in this investigation is however different from the one used in the remainder of the thesis: it is based on the Vortex-In-Cell (VIC) approach, that combines Lagrangian and finite difference methods. We here used an original approach [15, 99, 98] where the exact boundary condition is obtained dynamically using the fast multipole method. This combination is quite optimal: a compact VIC grid (enclosing tightly and dynamically the non-zero vorticity field) can be used while the proper boundary condition is enforced efficiently. The VIC-PFM (Vortex-in-Cell/Parallel Fast Multipole) approach is between 15 and 20 times faster compared to the “pure” Parallel Fast Multipole method used in the remainder of this thesis. While the current VIC-PFM approach still has clear limitations, these results show the path for future developments: one should transfer the developments made in the frame of “pure” Lagrangian methods into the VIC approach.

Finally, we took a closer look at turbulence modeling in the frame of the Eulerian velocity-pressure numerical approach. By introducing turbulence models in an existing DNS code, we had the opportunity to investigate the full spectrum of turbulence modeling: DNS, LES and RANS was investigated on the case of the turbulent channel flow. The Detached-Eddy Simulation (DES) approach, which is a hybrid RANS-LES approach, was also investigated. It was confirmed to provide poor results for the turbulent channel flow. We then investigated and further adapted a RANS-LES coupling formulation proposed in Templeton *et al.* [91]. The formulation consists of imposing a RANS eddy-viscosity dynamically corrected with the LES resolved turbulent stress near the wall. Results obtained using this approach for the turbulent channel flow are encouraging. The extension of the method to more complicated geometries is however not trivial, and its use in practical computations is therefore questionable. However, the idea of correcting the RANS eddy viscosity for the resolved stresses is certainly interesting and should be kept in mind as one might try to adapt it to more general RANS-LES approaches such as DES.



Bibliography

- [1] E. Achenbach. Vortex shedding from spheres. *J. Fluid Mech.*, 62:209, 1974.
- [2] J. Adams. MUDPACK: Multigrid fortran software for the efficient solution of linear elliptic partial differential equations. *Applied Math. and Comput.*, 34:113–146, 1989.
- [3] J. S. Baggett. Some modeling requirements for wall models in Large-Eddy simulation. *CTR Annual Research Briefs*, page 123, 1997.
- [4] V. Bakic. *Experimental investigation of turbulent flows around a sphere*. PhD thesis, TUHH Hamburg, Germany, 2002.
- [5] E. Balaras, C. Benocci, and U. Piomelli. Two-layer approximate boundary conditions for large-eddy simulations. *AIAA Journal*, 34:1111–1119, 1996.
- [6] J. Barnes and P. Hut. A hierarchical $O(n \log n)$ force calculations algorithm. *Nature*, pages 324–446, 1986.
- [7] M. Bergdorf, G-H. Cottet, and P. Koumoutsakos. Multilevel adaptive particle methods for convection-diffusion equations. *SIAM Multiscale Modeling and Simulation*, 4:328–357, 2005.



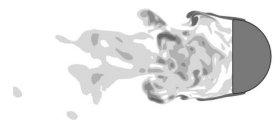
- [8] W. Cabot and P. Moin. Approximate wall boundary conditions in the large-eddy simulation of high Reynolds number flow. *Flow, Turbulence and Combustion*, 63:269–291, 2000.
- [9] G. F. Carnavale and R. C. Kloosterziel. Emergence and evolution of triangular vortices. *J. Fluid Mech.*, 259:305–331, 1994.
- [10] H. Choi, P. Moin, and J. Kim. Direct numerical simulation of turbulent flow over riblets. *J. Fluid Mech.*, 255:503–539, 1993.
- [11] J-P. Choquin and S. Huberson. Particle simulation of viscous flow. *Computers and Fluids*, 17(2):397–410, 1989.
- [12] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Math. Comput.*, 22:745, 1968.
- [13] K. Chua. *Vortex Simulation of Separated Flows in Two and Three Dimensions*. PhD thesis, California Institute of Technology, 1990.
- [14] R. Cocle, L. Dufresne, and G. Winckelmans. Investigation of multiscale subgrid models for LES of instabilities and turbulence in wake vortex systems. In *Lecture Notes in Computational Science and Engineering series*, Symposium on Complex Effects in Large-Eddy Simulation, Limassol (Cyprus), 2005. Springer. To appear 2006.
- [15] R. Cocle, G. Winckelmans, G. Daeninck, and F. Thirifay. An efficient combination of vortex-in-cell and fast multipole methods. *Submitted to J. Comp. Phys. in Dec. 2005*, 2005.
- [16] J. Constantinescu and G. Squires. LES and DES investigations of turbulent flow over a sphere at $Re=10,000$. *Flow, Turb. and Comb.*, 70:267–298, 2003.
- [17] J. Constantinescu and G. Squires. Numerical investigations of flow over a sphere in the subcritical and supercritical regimes. *Phys. of Fluids*, 16(5):1449–1466, 2004.
- [18] G-H. Cottet. Particle-grid domain decomposition methods for the Navier-Stokes equations in exterior domains. In C. Anderson and C. Greengard, editors, *Vortex dynamics and vortex methods*, volume 28 of *Lectures in Applied Mathematics*, pages 100–118, 1991.

- [19] G.-H. Cottet and P. Koumoutsakos. *Vortex Methods: Theory and Practice*. Cambridge Univ. Press, Cambridge (UK), 2000.
- [20] G.-H. Cottet, P. Koumoutsakos, and M. Ould Salihi. Vortex methods with spatially varying cores. *J. Comp. Phys.*, 162(11):164, 2000.
- [21] G.-H. Cottet, M. Lemine, and M. El Hamraoui. Recent developments in vortex methods for the simulation of unsteady incompressible flows. In *Electronic Proc. Third International Workshop on Vortex Flows and Related Numerical Methods*, Toulouse, France, 1999. ESAIM (European Series in Applied and Industrial Mathematics).
- [22] G.-H. Cottet and S. Mas-Gallic. A particle method to solve the Navier-Stokes system. *Numerische Mathematik*, 57:805–827, 1990.
- [23] G.-H. Cottet and P. Poncet. Particle methods for direct numerical simulations of three-dimensional wakes. *Journal of Turbulence*, 3 (028), 2002.
- [24] G.-H. Cottet and P. Poncet. Advances in direct numerical simulations of 3D wall bounded flows by vortex-in-cell methods. *J. Comp. Phys.*, 193:136, 2004.
- [25] C. Cottin. Experimental investigation of the dynamics of counter-rotating vortex pairs in ground effect. Master thesis, CNRS-IRPHE, Marseille, 2005.
- [26] R. Croll, W. Gutierrez, B. Hassan, J. Suazo, and A. Riggins. Experimental investigation of the ground transportation systems (GTS), project for heavy vehicle drag reduction. *SAE Technical Paper*, SP 1145(960907), 1996.
- [27] G. Daeninck. Local refinement in lagrangian vortex particles methods. Travail de fin d'étude, Université catholique de Louvain, Louvain-la-Neuve, 2000.
- [28] G. Daeninck, R. Cocle, and G. Winckelmans. LES calculations of spatially evolving wakes in ground effect. FAR-Wake research project report D 3. 1. 2-4, Université catholique de Louvain, 2006.
- [29] G. Daeninck, O. Desenfans, and G. Winckelmans. Span loading variations and wake roll-up in ground effect. FAR-Wake research project report D 3. 1. 1-1, Université catholique de Louvain, 2006.



- [30] G. Daeninck, P. Chatelain, M. Rubel, G. Winckelmans, and A. Leonard. *Simulation of vehicle aerodynamics using a vortex element method*, volume 19 R of *Lecture Notes in Applied and Computational Mechanics*. 2004.
- [31] G. Daeninck, G. Medic, J. Templeton, and G. Kalitzin. On near-wall dynamic coupling of LES with RANS turbulence models. In *Proceedings of the ASME Turbo Expo 2006: Power for Land, Sea and Air*, Barcelona, Spain, May 2006.
- [32] G. Daeninck, P. Ploumhans, and G. Winckelmans. Simulation of 3-D bluff body flows using vortex methods: from direct numerical simulation towards large-eddy simulation modelling. *Journal of Turbulence*, 3 (043), 2002.
- [33] G. Daeninck and G. Winckelmans. A new hybrid Eulerian-Lagrangian vortex method for flows with massive separation. In *Proceedings of the 6th National Congress on Theoretical and Applied Mechanics*, Ghent, May 2003.
- [34] C. de Langhe, B. Merci, and E. Dick. Hybrid RANS/LES modelling with an approximate renormalization group. I: Model development. *Journal of Turbulence*, 6 (013), 2005.
- [35] C. de Langhe, B. Merci, K. Lodefier, and E. Dick. Hybrid RANS/LES modelling with an approximate renormalization group. II: Applications. *Journal of Turbulence*, 6 (014), 2005.
- [36] P. Degond and S. Mas-Gallic. The weighted particle method for convection-diffusion equations. I: The case of an isotropic viscosity; II: The anisotropic case. *Math Comput*, 53:485, 1989.
- [37] J. del Alamo, J. Jimenez, P. Zandonade, and R. Moser. Scaling of the energy spectra of turbulent channels. *J. Fluid Mech.*, 500:135–144, 2004.
- [38] L. Dufresne. *A Spectral/B-Spline Method for the Navier-Stokes Equations in Unbounded Cylindrical Coordinates*. PhD thesis, Université Laval, Québec (Canada), 2001.
- [39] J. Eldredge, A. Leonard, and T. Colonius. A general deterministic treatment of derivatives in particle methods. *J. Comp. Phys.*, 180:686, 2002.

- [40] L. Georges, G. Winckelmans, and P. Geuzaine. LES of compressible shock-free flows on hybrid unstructured grids. In *Proceedings of the 7th National Congress on Theoretical and Applied Mechanics*, Mons, May 2006.
- [41] M. Germano, U. Piomelli, P. Moin, and P. Cabot. A dynamic subgrid-scale eddy viscosity model. *Phys. of Fluids*, 3(7):1760–1765, 1991.
- [42] L. Greengard and V. Rohklin. A fast algorithm for particle simulations. *J. Comp. Phys.*, 73:325, 1987.
- [43] J. Gullbrand and F. Chow. The effect of numerical errors and turbulence models in large-eddy simulation of channel flow, with and without explicit filtering. *J. Fluid Mech.*, 495:323–341, 2003.
- [44] S. F. Hoerner. *Fluid dynamic drag*. Published by author, 1958.
- [45] S. Hoyas and J. Jimenez. Scaling of velocity fluctuations in turbulent channels up to $re_\tau = 2003$. *Phys. of Fluids*, 18:011702, 2006.
- [46] W. Janna. *Introduction to Fluid Mechanics*. PWS Publishing Company, 1993.
- [47] W. Janna. *Turbulent flows*. Cambridge Univ. Press, 2000.
- [48] H. Jeanmart and G. Winckelmans. Comparison of recent dynamic subgrid-scale models in the case of the turbulent channel flow. In *Proceeding of the Summer Program*, pages 331–342, Center for Turbulence Research, NASA Ames/Stanford University, USA, 2002.
- [49] G. Kalitzin, G. Medic, G. Iaccarino, and P. A. Durbin. Near-wall behavior of RANS turbulence models and implications for wall functions. *J. Comp. Phys.*, 204(1):265–291, 2005.
- [50] G. Kalitzin, J. A. Templeton, and G. Medic. A near-wall eddy-viscosity formulation for LES. In *Lecture Notes in Computational Science and Engineering series*, Symposium on Complex Effects in Large-Eddy Simulation, Limassol (Cyprus), 2005. Springer. To appear 2006.
- [51] D. Kim and H. Choi. Laminar flow past a hemisphere. *Phys. of Fluids*, 15(8):2457–2460, 2003.
- [52] J. Kim and P. Moin. Application of a fractional-step method to incompressible Navier-Stokes equations. *J. Comp. Phys.*, 59:308, 1985.



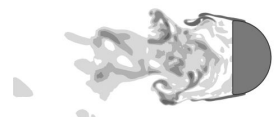
- [53] S. J. Kline, W. C. Reynolds, F. A. Schraub, and P. W. Runstadler. The structure of turbulent boundary layers. *J. Fluid Mech.*, 30:741–773, 1967.
- [54] R. C. Kloosterziel and G. F. Carnevale. On the evolution and saturation of instabilities of two-dimensional isolated circular vortices. *J. Fluid Mech.*, 388:217–257, 1999.
- [55] P. Koumoutsakos and A. Leonard. High resolution simulation of the flow around an impulsively started cylinder using vortex methods. *J. Fluid Mech.*, 296:1, 1995.
- [56] P. Koumoutsakos, A. Leonard, and F. Pépin. Boundary conditions for viscous vortex methods. *J. Comp. Phys.*, 113:52–61, 1994.
- [57] P. Koumoutsakos and D. Shiels. Simulations of the viscous flow normal to an impulsively started and uniformly accelerated flat plate. *J. Fluid Mech.*, 328:177–227, 1996.
- [58] G. M. Laskowski. *Inverse Design of a Turbine Cascade Passage and DNS of a Stationary and Rotating Serpentine Passage*. PhD thesis, Stanford University, 2005.
- [59] M. Lemine. *Couplage de méthodes numériques en simulation directe d'écoulements incompressibles*. PhD thesis, Univ. de Grenoble – Joseph Fourier, 1998.
- [60] D. Lilly. A proposed modification of the germano subgrid-scale closure method. *Phys. of Fluids*, 4:237–248, 1992.
- [61] E. Marchandise, J-F. Remacle, and N. Chevaugeon. A quadrature-free discontinuous galerkin method for the level set equation. *J. Comp. Phys.*, 212(1):338–357, 2006.
- [62] G. Medic, G. Daeninck, J. Templeton, and G. Kalitzin. A framework for near-wall RANS/LES coupling. *CTR Annual Research Briefs*, 2005.
- [63] G. Medic, J. A. Templeton, and G. Kalitzin. A formulation for near-wall RANS/LES coupling. *J. Comp. Phys.*, 2005.
- [64] F. R. Menter. Zonal two-equation k - ω turbulence model predictions. *AIAA Paper*, (93-2906), 1993.

- [65] F. R. Menter and Y. Egorov. A scale-adaptive simulation model using two-equation models. In *Proceedings of the 44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2005.
- [66] F. R. Menter, M. Kuntz, and R. Langtry. Ten years of industrial experience with the SST turbulence model. In *Proceedings of Turbulence, Heat, and Mass Transfer 4*, pages 625–632, 2003.
- [67] J. J. Monaghan. Extrapolating B-splines for interpolation. *J. Comp. Phys.*, 60:253–262, 1985.
- [68] J. J. Monaghan. Particle methods for hydrodynamics. *Comp. Phys. Reports*, 3:71–124, 1985.
- [69] R. Moser, J. Kim, and N. Mansour. DNS of turbulent channel flow up to $Re_\tau=590$. *J. Fluid Mech.*, 11:943–945, 1999.
- [70] F. Nicoud. *Problèmes de frontières en mécanique des fluides : Modélisation et discrétisation*. PhD thesis, Habilitation à Diriger des Recherches - Institut National Polytechnique de Toulouse, 2000.
- [71] F. Nicoud and F. Ducros. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. *Flow, Turb. and Comb.*, 62(3):183–200, 1999.
- [72] N. V. Nikitin, F. Nicoud, B. Wasistho, K. D. Squires, and P. R. Spalart. An approach to wall modeling in large-eddy simulations. *Phys. Fluids. Letters*, 12(7):1629, 2000.
- [73] F. Noca, D. Shiels, and D. Jeon. Measuring instantaneous fluid dynamic forces on bodies using only velocity fields and their derivatives. *J. Fluids and Structures*, 11:345, 1997.
- [74] F. Noca, D. Shiels, and D. Jeon. A comparison of methods for evaluating time-dependent fluid forces on bodies using only velocity fields and their derivatives. *J. Fluids and Structures*, 11:345, 1999.
- [75] M. Ould-Salihi, G-H. Cottet, and M. El Hamraoui. Blending finite-difference and vortex methods for incompressible flow computations. *SIAM Journal Scientific Computation*, 22(5):1655–1674, 2000.
- [76] P. Ploumhans. *Simulation of High Reynolds Number Flows Past Bluff Bodies Using Vortex and Boundary Element Methods*. PhD thesis, Université catholique de Louvain, Louvain-la-Neuve, 2001.



- [77] P. Ploumhans and G. Winckelmans. Vortex methods for high resolution simulations of viscous flow past bluff-bodies of general geometry. *J. Comp. Phys.*, 165:354, 2000.
- [78] P. Ploumhans, G. Winckelmans, J. Salmon, A. Leonard, and M. Warren. Vortex methods for high-resolution simulation of three-dimensional bluff body flows; application to the sphere at $Re=300$, 500 and 1000. *J. Comp. Phys.*, 178(1):427, 2002.
- [79] P. Ploumhans, G. Winckelmans, and J. K. Salmon. Vortex particles and tree codes: I. flows with arbitrary crossing between solid boundaries and particle redistribution lattice; II. vortex ring encountering a plane at an angle. In *Proc. 3rd International Workshop on Vortex Flows and Related Numerical Methods*, page 25, Toulouse, France, August 24–27 1998. ESAIM (European Series in Applied and Industrial Mathematics).
- [80] P. Poncet. *Méthodes particulières pour la simulation des sillages tridimensionnels*. PhD thesis, Univ. de Grenoble – Joseph Fourier, 2001.
- [81] P. Poncet. Vanishing of mode B in the wake behind a rotationally oscillating circular cylinder. *Phys. of Fluids*, 14(6), 2002.
- [82] P. Poncet. Topological aspects of the three-dimensional wake behind rotary oscillating circular cylinder. *J. Fluid Mech.*, 517:27–53, 2004.
- [83] M. Provansal. From the double vortex street behind a cylinder to the wake of a sphere. In *Conference on bluff body wakes and vortex induced vibration (BBVIV3)*, Port Douglas, Queensland, 2002.
- [84] M. Rosenfeld, D. Kwak, and M. Vinokur. A fractional-step solution method for the unsteady incompressible navier-stokes equations in generalized coordinate systems. *J. Comp. Phys.*, 94(1):102–137, 1999.
- [85] P. Sagaut. *Large Eddy Simulation for Incompressible Flows: an Introduction*. Scientific Computation. Springer, Berlin, Heidelberg, 2001.
- [86] H. Sakamoto and H. Haniu. A study of vortex shedding from spheres in a uniform flow. *J. Fluids Eng.*, 112:386, 1990.
- [87] J. Salmon and M. Warren. Skeletons from the treecode closet. *J. Comp. Phys.*, 111(1):136, 1994.

- [88] J. Smagorinsky. General circulation experiments with the primitive equations. *Mon. Weath. Rev.*, 91:99–152, 1963.
- [89] P. R. Spalart and S. R. Allmaras. A one-equation turbulence model for aerodynamic flows. *La Recherche Aéronautique*, 1:1–23, 1994.
- [90] P. R. Spalart, W-H. Jou, M. Strelets, and S. R. Allmaras. *Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach*, pages 4–8. First AFOSR International Conference on DNS/LES, Ruston, LA, 1997.
- [91] J. A. Templeton, G. Medic, and G. Kalitzin. An eddy-viscosity based near-wall treatment for coarse grid LES. *Phys. Fluids*, 17, 2005.
- [92] A. Travin, M. Shur, M. Strelets, and P. Spalart. Detached-eddy simulations past a circular cylinder. *Flow, Turb. and Comb.*, 63:293–313, 1999.
- [93] M. Ubroi and P. Freymuth. Turbulent energy balance and spectra of the axisymmetric wake. *Phys. of Fluids*, 13(9), 1970.
- [94] M. Van Dyke. *An Album of Fluid Motion*. Parabolic Press, Stanford, 1982.
- [95] M. Wang and P. Moin. Dynamic wall modeling for large-eddy simulation of complex turbulent flows. *Phys. Fluids*, 14(7):2043–2051, 2002.
- [96] D. C. Wilcox. *Turbulence modeling for CFD*. DCW Industries, La Canada, California, 1993.
- [97] J-C. Willemart. Investigation expérimentale d'écoulements turbulents autour d'un hémisphère. Travail de fin d'étude, Université catholique de Louvain, Louvain-la-Neuve, 2006.
- [98] G. Winckelmans. *Vortex Methods*, volume 3 (Fluids) of *Encyclopedia of Computational Mechanics*, chapter 5. John Wiley & Sons, 2004.
- [99] G. Winckelmans, R. Cocle, L. Dufresne, and R. Capart. Vortex methods and their application to trailing wake vortex simulations. *Special issue on Aircraft Trailing Vortices, C. R. Phys.*, 6(4–5):467–486 (see <http://france.elsevier.com/direct/COMREN>), 2005.



- [100] G. Winckelmans and A. Leonard. Contributions to vortex particle methods for the computation of three-dimensional incompressible unsteady flows. *J. Comp. Phys.*, 109(2):247–273, 1993.
- [101] G. Winckelmans, J. K. Salmon, A. Leonard, and M. S. Warren. Three-dimensional vortex particle and panels methods: Fast tree-code solvers with active error control for arbitrary distributions/geometries. In *Proc. Forum on Vortex Methods for Engineering Applications*, page 25, Albuquerque, NM, February 22–24 1995.
- [102] X. Wu and P. Durbin. Evidence of longitudinal vortices evolved from distorted wakes in a turbine passage. *J. Fluid Mech.*, 446:199–228, 2001.

APPENDIX A

Flow past lifting surfaces using a lifting line model

Vortex wakes that emanate from lifting surfaces (aircraft wings, wind turbine blades, helicopter blades) correspond to trailing vorticity that is spatially compact (sheet-like). For incompressible flows, they can be efficiently simulated using three-dimensional vortex element methods.

The lifting surfaces can be represented by bound vorticity elements: lifting lines in the simplest formulation (using Prandtl lifting line theory), lifting surfaces (discretized using vortex dipoles, i.e., discrete vortex lattices) and, in the more sophisticated formulation, vortex panels for DNS or LES of the viscous Navier-Stokes equations.

We here use the simplified approach where the airfoil is idealized as a “lifting line” vortex. The detailed airfoil cross section and flow around it is not explicitly discretized: it is modeled. At each moment in time, t , and at every location along the span, y , the proper lifting line circulation, Γ , is determined to satisfy the “Kutta” condition:

$$\Gamma(y, t) = \frac{1}{2} |\mathbf{U}_{\text{eff}}| (a_0 c \sin(\alpha_{\text{eff}}) + c_{l_0}) \quad (\text{A.1})$$

where, a_0 , c and c_{l_0} are the local airfoil characteristics: respectively the local lift coefficient, chord and the zero lift coefficient. α_{eff} is the local effective angle



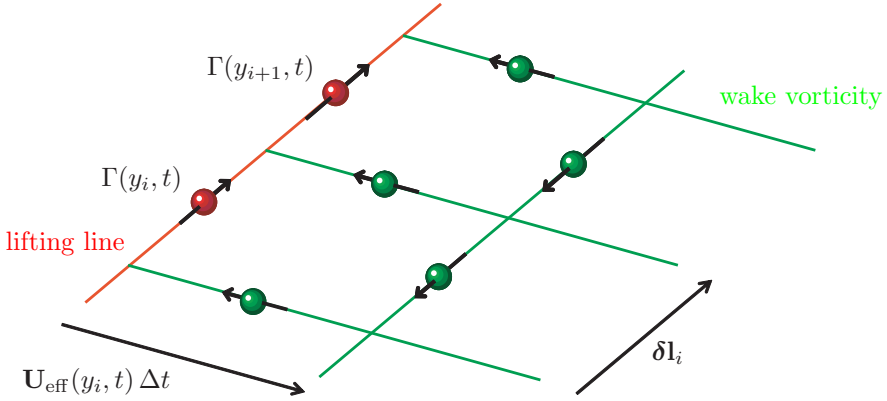


FIGURE A.1 – Vorticity generation along lifting line. Attached particles (red) and new wake particle (green) are shown.

of attack: it takes the geometrical angle of attack into account as well as the induced angle of attack due to the freestream and wake. Finally, U_{eff} is the local effective velocity (due to freestream and wake).

Due to the circulation variation along the lifting line, vorticity has to be injected into the flow in order to satisfy the solenoidal condition on the vorticity field. A similar approach as the one developed in Section 2.2 is used to shed wake vorticity into the flow.

The lifting line is discretized using N shedding edges (Fig. A.1). The attached vorticity strength (i.e., $\alpha_i = \int \omega dV$) associated to each of them and their location is given by

$$\alpha_i^{\text{att}}(t) = \Gamma(y_i, t) \delta \mathbf{l}_i \quad (\text{A.2})$$

$$\mathbf{x}_i^{\text{att}}(t) = \mathbf{x}(y_i, t) \quad (\text{A.3})$$

where $\delta \mathbf{l}_i$ is the curvilinear vector increment for the shedding edge. At each time step, wake vortices are generated and shed into the flow:

- N parallel vortices of strength:

$$\alpha_i^{\text{shed}, \parallel}(t) = (\Gamma(y_i, t) - \Gamma(y_i, t - \Delta t)) \delta \mathbf{l}_i \quad (\text{A.4})$$

$$\mathbf{x}_i^{\text{shed}, \parallel}(t) = \mathbf{x}(y_i, t) + \mathbf{U}_{\text{eff}}(y_i, t) \Delta t \quad (\text{A.5})$$

- $N + 1$ perpendicular vortices of strength:

$$\alpha_i^{\text{shed},\perp}(t) = (\Gamma(y_{i+1}, t) - \Gamma(y_i, t)) \mathbf{U}_{\text{eff}}(y_i, t) \Delta t \quad (\text{A.6})$$

$$\begin{aligned} \mathbf{x}_i^{\text{shed},\parallel}(t) &= \frac{1}{2} (\mathbf{x}(y_i) + \mathbf{x}(y_{i+1})) + \\ &\quad \frac{1}{2} (\mathbf{U}_{\text{eff}}(y_i, t) + \mathbf{U}_{\text{eff}}(y_{i+1}, t)) \Delta t \end{aligned} \quad (\text{A.7})$$

Once the particles have been shed, they are convected and stretched according to the local flow field.

The typical time step as described in Section 1.3 has to be modified as follows:

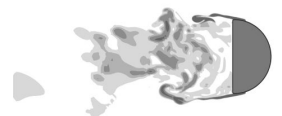
1. Compute the fields $(\mathbf{u}, \nabla \mathbf{u})$ induced by the free and the attached vorticity (i.e., the lifting line vorticity);
2. Convect and stretch the wake vortex particles according to the local velocity and velocity gradient;
3. Compute the lifting line circulation in order to satisfy the Kutta condition using Eq. (A.1) and update the attached particle strength and location using Eqs. (A.2) and (A.3);
4. Shed the N “parallel” and the $N + 1$ “perpendicular” new wake particles into the flow according to Eqs. (A.4) through (A.7).

Note that, in this approach, no redistribution is performed and no viscous diffusion is taken into account.

Flow past a rectangular wing with flaps

Results for the flow past a rectangular wing with flaps are shown in Figs. A.2 and A.3. The wing generates a four-vortex system: two outer vortices are created at the wing tips and two inner vortices are created at the inner flap tips. The inner vortices are counter-rotating with respect to the outer vortices. The simulation includes the impulsive start of the wing and thus the startup vortex which can be seen at the end of the wake in Fig. A.2. The dimensionless time is defined as $T = t U_\infty / b$, where b is the wing span.

An interesting phenomenon captured by this simulation is the development of the so-called “end-effect” waves that propagate along the vortices. End-effect instabilities appear on vortices as soon as there is a variation along a



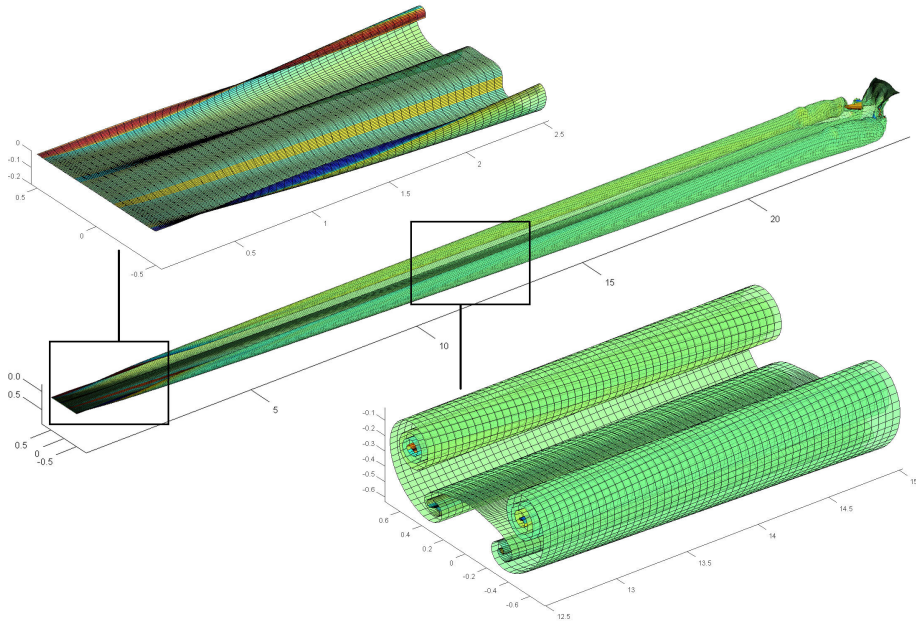


FIGURE A.2 – Flow past a rectangular wing with flaps: visualization of the entire vortex sheet at $T = 25$. The zoom on the near and intermediate wake regions are also shown.

vortex tube of its radial circulation profile. In this case, the variation is related to the bending of the vortex tube at the location where the wake vortices are connected to the startup vortex. The development and propagation of the end-effect wave can be seen in Fig. A.3.

As the computational approach used here is inviscid, the development of the end-effect instabilities also leads to the numerical blowup of the computation.

Flow past a wind turbine

The method was also applied to the flow past a simplified, three-blade, wind turbine. Each blade is modeled using a lifting line.

Figure A.4 visualizes the vortex wake generated by the wind turbine blades. Each blade generates a helical shaped vortex sheet with, at its top and bottom, strong blade-tip vortices. A comparison between an experimental visualization of the flow past a propeller and the current simulation is shown in Fig. A.5.

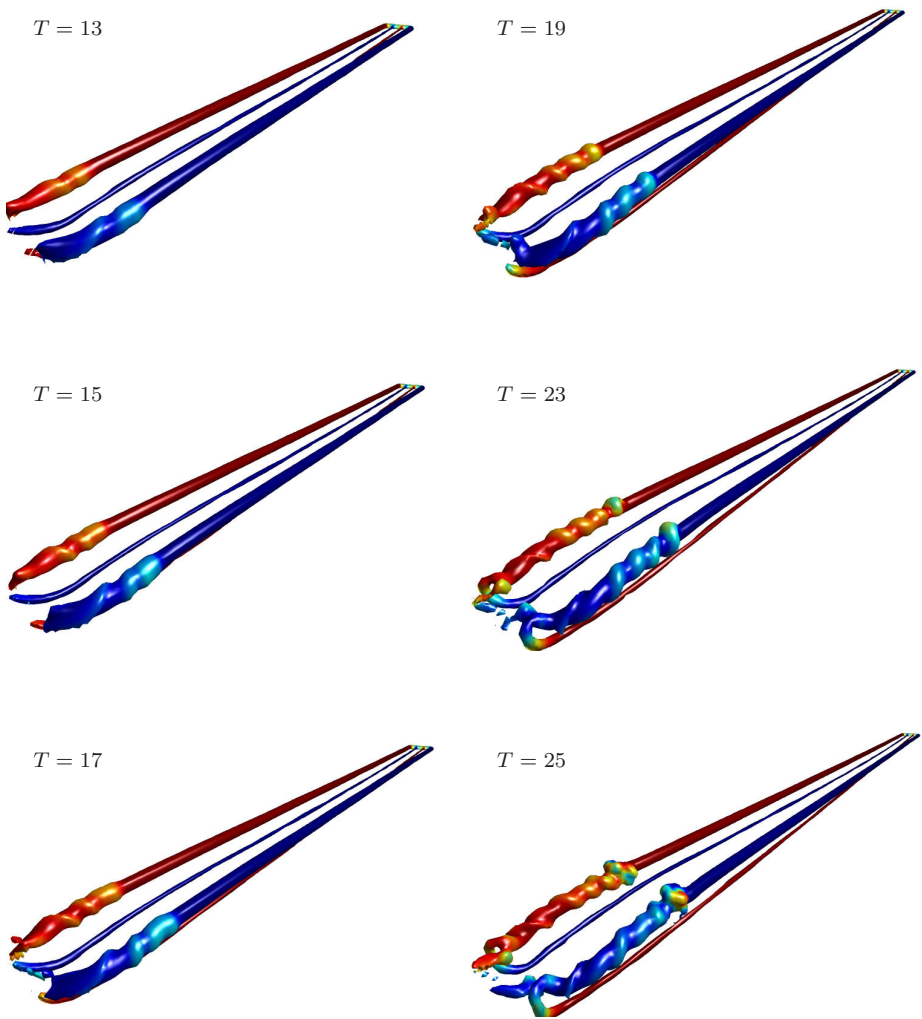


FIGURE A.3 – Flow past a rectangular wing with flaps: isosurface of vorticity norm colored by longitudinal vorticity component.

One can clearly see that the blade tip vortices for the propeller case rotate in the opposite direction compared to the wind turbine case: this is obviously because the propeller accelerates the flow while the wind turbine decelerated the flow. Finally, Fig. A.6 shows the axial and tangential velocity fields in two cross planes.



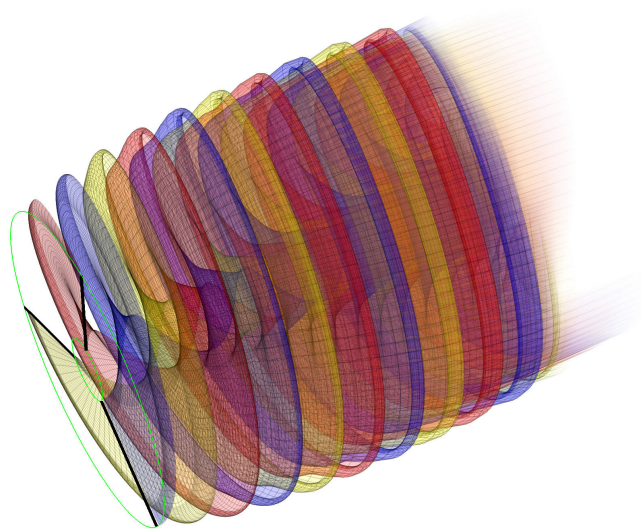


FIGURE A.4 – Flow past a wind turbine. The vortex sheets generated by the three blades are visualized (only part of the wake is shown).

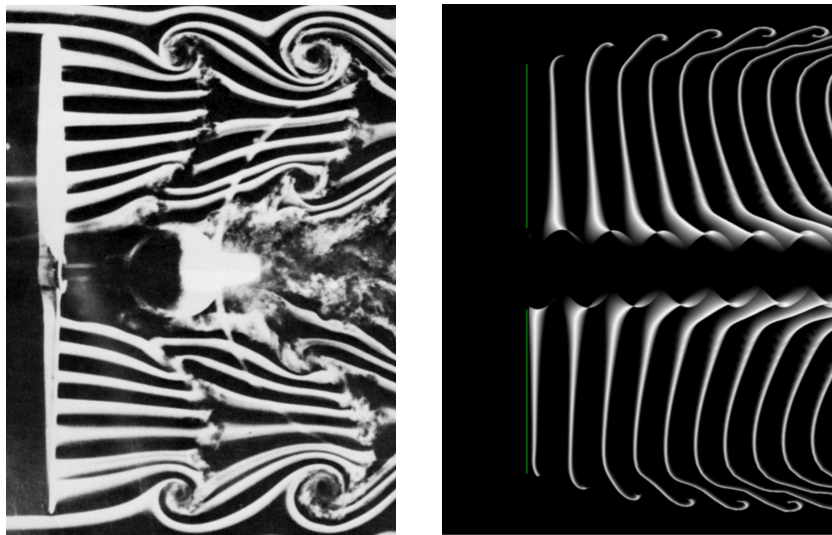


FIGURE A.5 – **Left:** Experimental smoke visualization for the flow past a propeller, taken from Van Dyke [94]; the vortex sheets traces and the blade tip vortices can be seen. **Right:** Visualization of the vortex sheets in a mid-plane for the current simulation of the flow past a wind turbine.

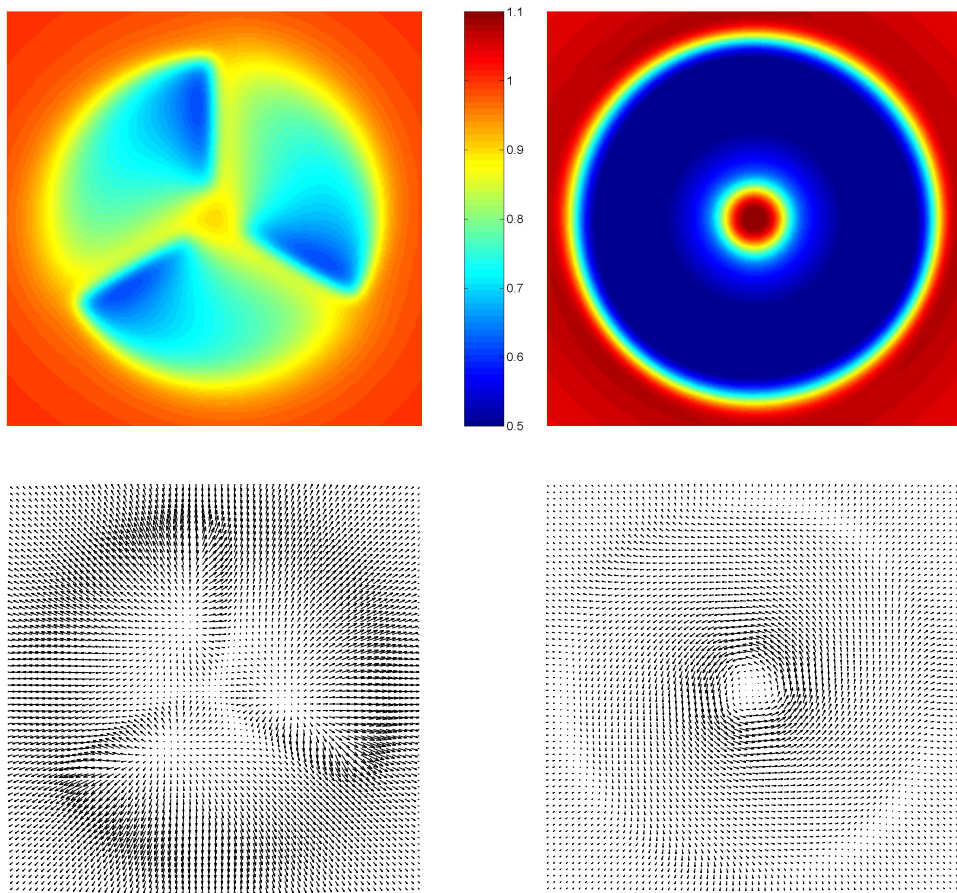


FIGURE A.6 – Flow past a wind turbine: axial velocity component, u_x/U_∞ , (**top**) and tangential velocity vector (**bottom**) in two $y - z$ cross planes. **Left:** slightly upstream of the wind turbine blades, $x/D = -0.05$. **Right:** Downstream of the wind turbine, $x/D = 0.5$.



APPENDIX B

Mapping of the Physical Domain: 3-D Trumpet Mapping

In external flows, the vorticity is located in the boundary layer and in the wake, with the magnitude of the vorticity decreasing as one goes downstream. It is thus more efficient to have high resolution near the body and coarser resolution in the far wake. To achieve this, the physical domain, with spatially varying resolution, is mapped onto a regular indicial lattice (i, j, k) . This approach has been used previously to perform 2-D simulations (Cottet *et al.* [20], Lemine [59], Cottet and Koumoutsakos [19], Ploumhans and Winckelmans [77]) and was extended for 3-D simulations (P. Ploumhans *et al.* [78]). We present here a new mapping which smoothly goes from a constant $h \times h \times h$ mapping to an exponentially growing mapping (see Fig. B.1). The mapping function is given by:

$$\begin{cases} x &= x_0 + R_0 f_x(i) \cos \theta(j) \cos \phi(k) \\ y &= y_0 + R_0 f_y(i) \sin \theta(j) \cos \phi(k) \\ z &= z_0 + R_0 f_z(i) \cos \theta(j) \sin \phi(k) \end{cases} \quad (\text{B.1})$$



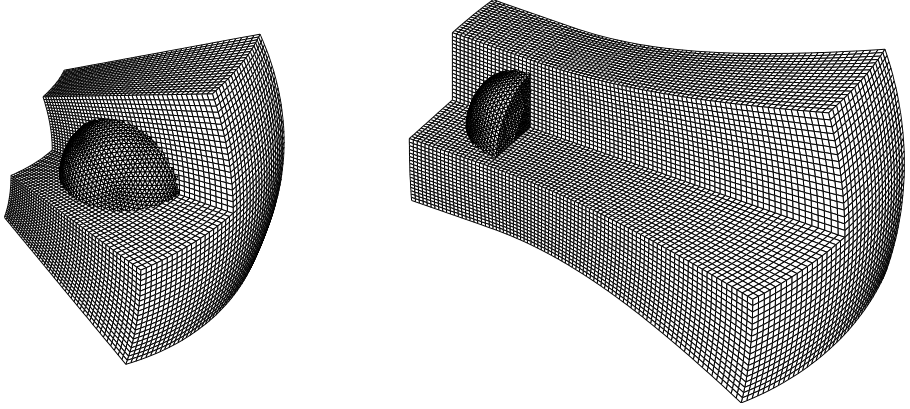


FIGURE B.1 – **Left:** Exponential mapping. **Right:** Trumpet mapping.

where,

$$\begin{cases} \theta(j) &= \frac{2\pi}{m}j, \quad -m/2 \leq j \leq m/2 \\ \phi(k) &= \frac{2\pi}{m}k, \quad -m/2 \leq k \leq m/2 \\ f_x(i) &= 2 \cosh\left(\frac{2\pi}{m}i\right), \\ f_y(i) &= f_z(i) = 2 \sinh\left(\frac{2\pi}{m}i\right). \end{cases} \quad (\text{B.2})$$

One can see that for $\frac{2\pi}{m}i \gg 1$ the mapping reduces to the “classical” exponential mapping where $r_x(i) = r_y(i) = r_z(i) = R_0 \exp\left(\frac{2\pi}{m}i\right)$.

The Jacobian of the mapping, J , is equal to

$$J = \left(\frac{2\pi}{m}\right)^3 f_x(i) \cos \theta_j \cos \phi_k (f_x^2(i) - \sin^2 \theta_j - \sin^2 \phi_k) \quad (\text{B.3})$$

It cancels for $j = \pm m/4$ and $k = \pm m/4$, and when $i = 0$ for $|i| + |j| = m/4$. In practice, this is not a problem, because the parameters of the mapping are chosen such that particles remain “far” from these singularities. There is another reason to use moderate values of j/m and k/m : the desire that the volume associated to each particle (which is a $1 \times 1 \times 1$ cube in the mapped domain) be as close as possible to a cube in the physical domain.

After each redistribution, new particles in the mapped domain have position $(i + 1/2, j + 1/2, k + 1/2)$ and the corresponding positions obtained from Eqs. (B.1) and (B.2) in the physical domain. The volume of fluid associated

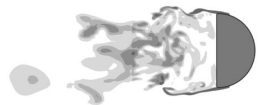
with the particle is

$$\begin{aligned}
 V &= \int \int \int J di dj dk \\
 &= 8 R_0^3 \sin\left(\frac{\pi}{m}\right) \cdot \sinh\left(\frac{\pi}{m}\right) \cdot \cosh\left(\frac{2\pi}{m}i\right) \cdot \cos\left(\frac{2\pi}{m}j\right) \cdot \cos\left(\frac{2\pi}{m}k\right) \\
 &\quad \left[\left(\cosh^2\left(\frac{2\pi}{m}i\right) - \frac{3}{4} \right) \cdot \left(1 + \frac{4}{3} \sinh^2\left(\frac{\pi}{m}\right) \right) + \frac{3}{4} \right. \\
 &\quad \left. - \left(\sin^2\left(\frac{2\pi}{m}j\right) \cdot \cos^2\left(\frac{\pi}{m}\right) + \frac{1}{3} \cos^2\left(\frac{2\pi}{m}j\right) \cdot \sin^2\left(\frac{\pi}{m}\right) \right) \right. \\
 &\quad \left. - \left(\sin^2\left(\frac{2\pi}{m}k\right) \cdot \cos^2\left(\frac{\pi}{m}\right) + \frac{1}{3} \cos^2\left(\frac{2\pi}{m}k\right) \cdot \sin^2\left(\frac{\pi}{m}\right) \right) \right] \quad (\text{B.4})
 \end{aligned}$$

The smoothing parameter, σ , associated to the particle is given by:

$$\sigma = \beta h \quad (\text{B.5})$$

where β is the core overlapping parameter (which is the same for all particles) and $h = V^{1/3}$. The use of this mapping produces a σ that varies smoothly in space.



Multiple moving bodies

The implementation of the 2-D vortex method for multiple moving bodies was initiated through a collaboration with Professor G. Dumas (Université Laval, Québec) while he was visiting Professor at our department.

The vortex element method (VEM) combined with the boundary element method (BEM) was extended for the aerodynamics of arbitrarily moving multiple bodies. This was done by augmenting our previous method for DNS of flows about bodies of general geometry (see Ploumhans and Winckelmans [77]).

We added the displacement (translation and rotation) of the bodies, the Biot-Savart velocity induced by the rotational rate of each body (with same methodology as in “contour dynamics”, reducing the integral over each body area to an integral over its contour), and the vorticity flux correction due to the moving walls. Because of the “grid-free” behavior of the method, cases where the bodies have complex, large amplitude movements (including close relative approaches) are handled conveniently. This approach was originally applied to DNS of multiple oscillating airfoils at moderate Reynolds numbers.

We here present a case where the method was used to numerically reproduce the experimental investigation of a pair of counter rotating vortices in ground-effect by also simulating the precise generation phase of the vortices. This experimental investigation was done as part of the FAR-Wake Project (see Chapter 6) by Cottin [25]. The experimental setup (i.e., the vortex generator)



consists of two flat plates fixed to a common base. Their free edges are sharpened to an angle of 30° . The plates have a high aspect ratio in order to ensure that the flow is essentially two-dimensional. A symmetric inward motion of the plates is generated by a computer-controlled step motor.

To better understand the details of the vortex system generated by this device, we decided to perform a DNS of the exact same setup: using the same geometry, the same law of displacement of the plates and the same Reynolds number as the experiment. Figures C.1 to C.4 show the vorticity contours at distinctive times during the simulation. The major question asked by the experimentalists was to know the amount of opposite sign vorticity entrained during the vortex formation (as can be seen in Fig. C.2). The details results from the simulation still have to be analyzed and further compared with the experimental results. One can already conclude that the global dynamics are in excellent agreement with the experimental results (at least, as long as the flow remains two-dimensional).

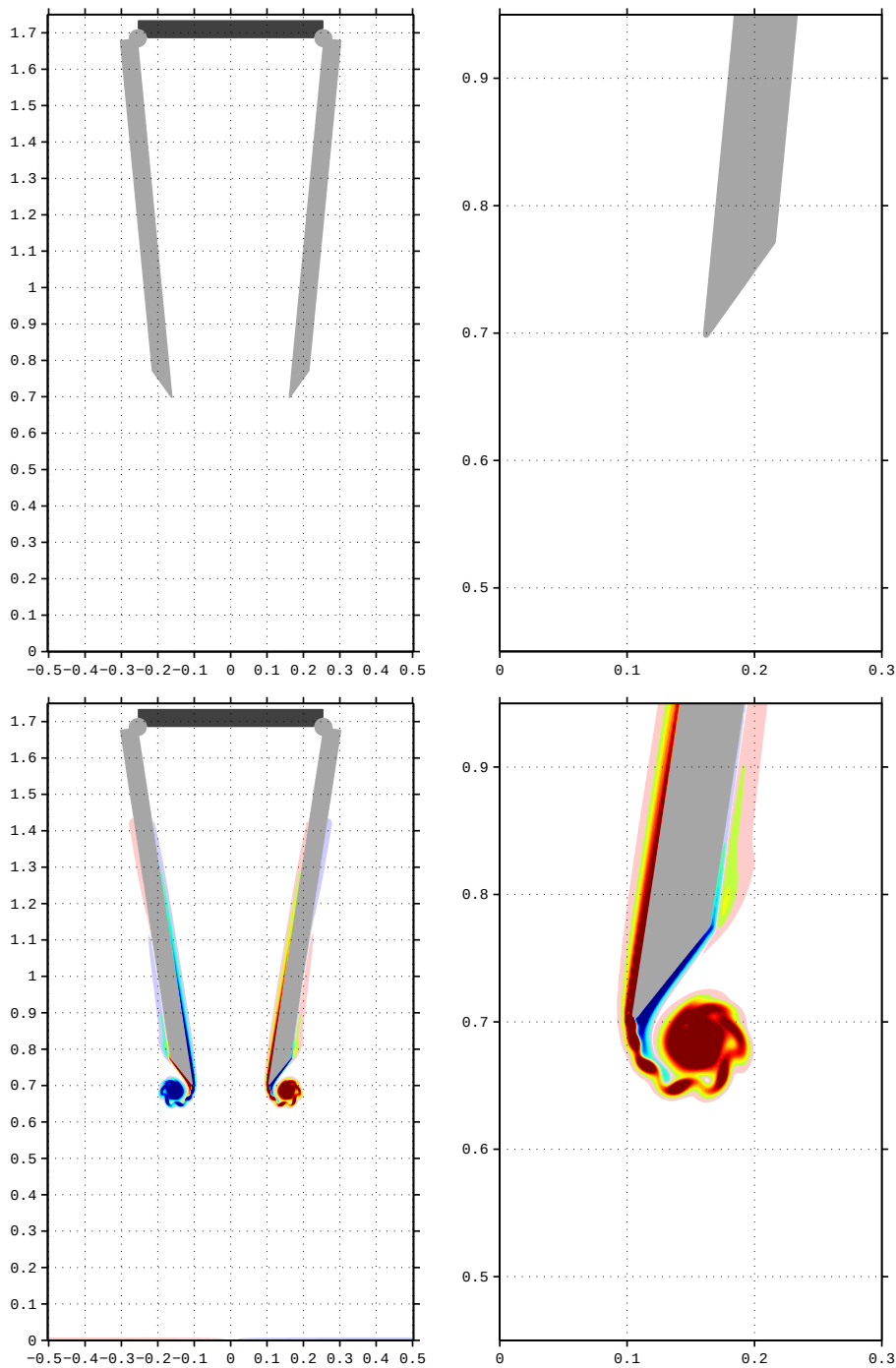
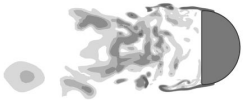


FIGURE C.1 – Vorticity contours for DNS of the vortex generator device at $T = 0.0$ (top) and $T = 0.6$ (bottom).



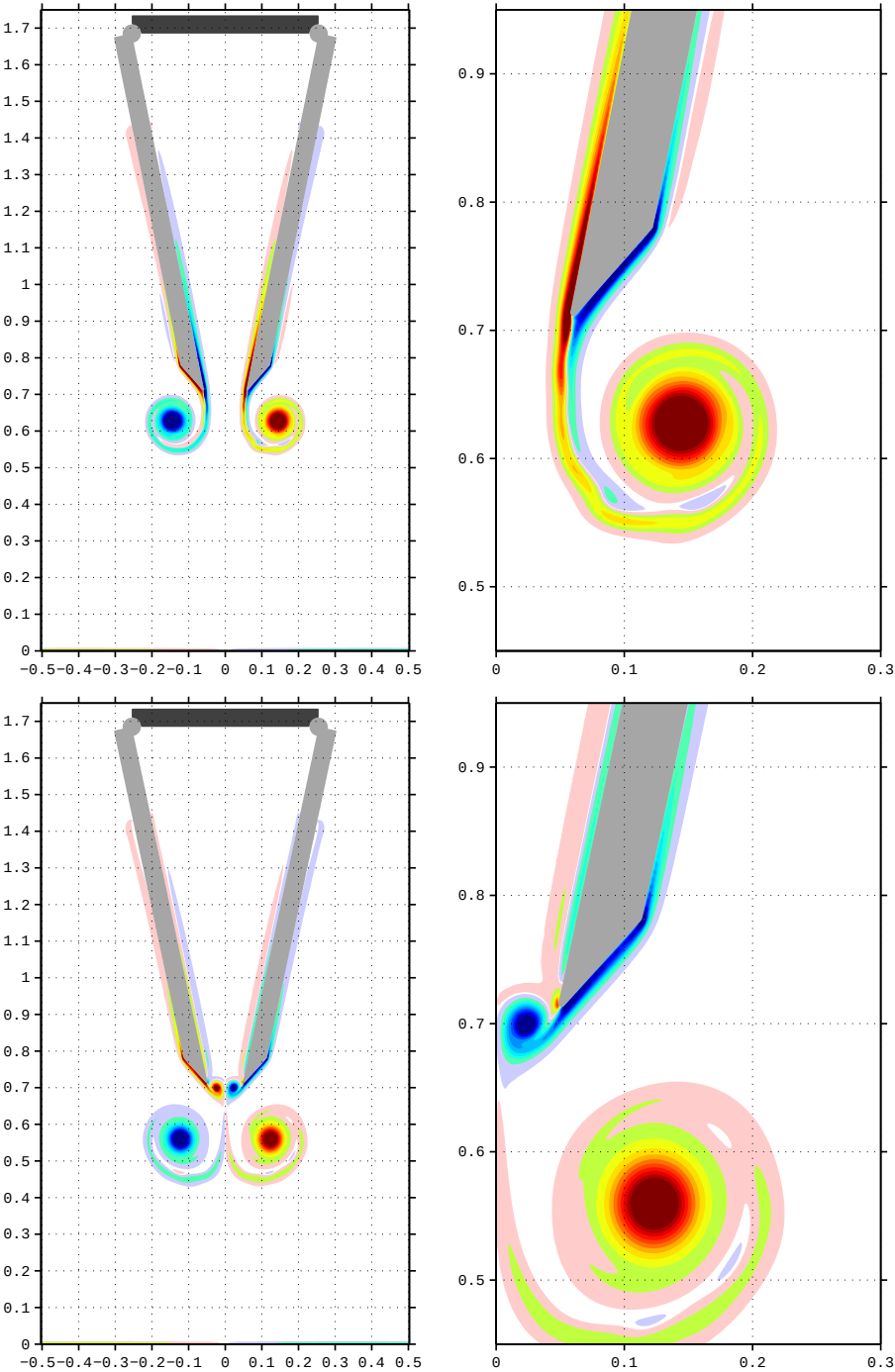


FIGURE C.2 – Vorticity contours for DNS of the vortex generator device $T = 0.9$ (top) and $T = 1.2$ (bottom). The motion of the plates is stopped at $T = 1.0$.

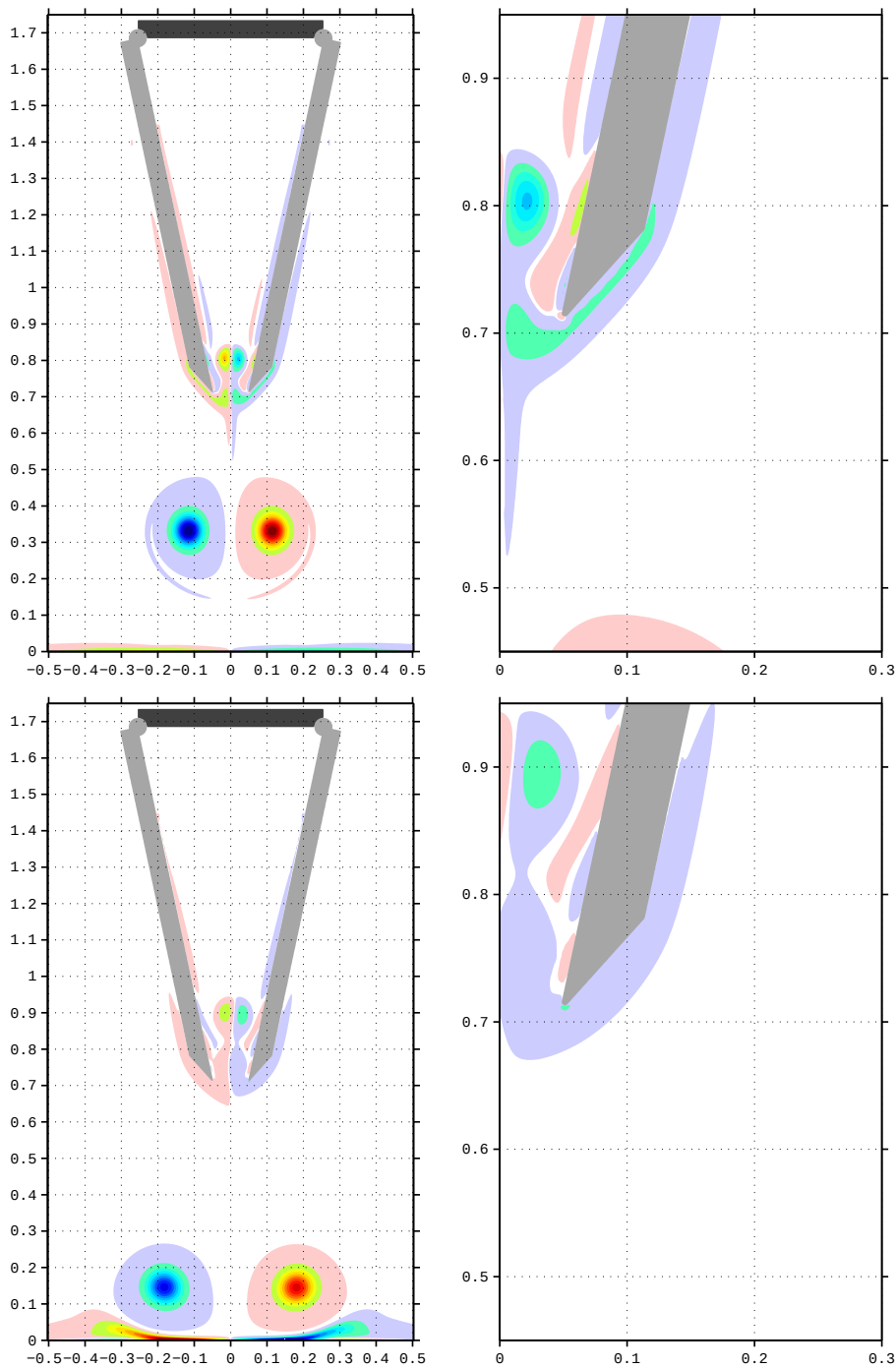
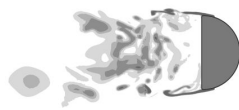


FIGURE C.3 – Vorticity contours for DNS of the vortex generator device $T = 2.0$ (top) and $T = 3.0$ (bottom).



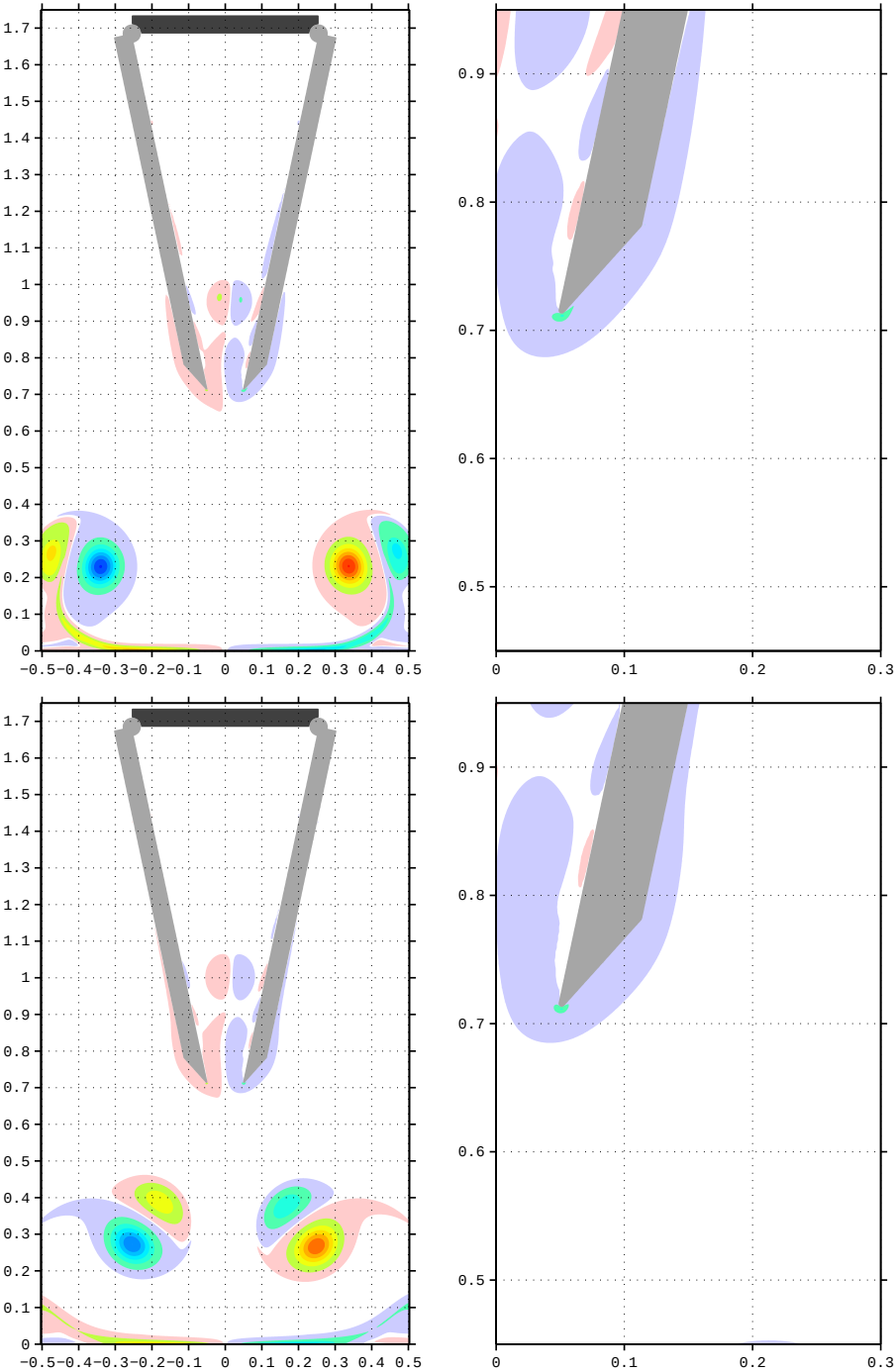


FIGURE C.4 – Vorticity contours for DNS of the vortex generator device $T = 4.0$ (top) and $T = 5.0$ (bottom).

APPENDIX D

List of symbols for channel flow

Dimensional quantities

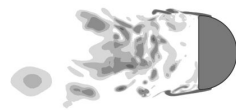
H	Half channel height
ν	Laminar viscosity
ν_t	Turbulent viscosity
$\tau_w = \rho \nu \left. \frac{d\langle u \rangle}{dy} \right _{\text{wall}}$	Mean wall shear stress
$U_\tau = \sqrt{\frac{\tau_w}{\rho}}$	Mean friction velocity
$U_b = \frac{1}{2H} \int_0^{2H} \langle u \rangle dy$	Bulk velocity
$\tau_{xy}^{\text{visc}} = \rho \nu \frac{d\langle u \rangle}{dy}$	Mean viscous shear stress
$\tau_{xy}^{\text{res}} = -\langle u'v' \rangle$	Mean resolved turbulent shear stress
$\tau_{xy}^{\text{mod}} = \rho \langle \nu_t \frac{\partial u}{\partial y} \rangle$	Mean modeled turbulent shear stress

Global dimensionless quantities

$$Re_\tau = \frac{U_\tau H}{\nu} , \quad Re_b = \frac{U_b H}{\nu}$$

Usual wall unit quantities

$$y^+ = \frac{y U_\tau}{\nu} , \quad u^+ = \frac{u}{U_\tau} , \quad U^+ = \frac{\langle u \rangle}{U_\tau} , \quad t^+ = \frac{t U_\tau}{H} , \quad \nu_t^+ = \frac{\nu_t}{\nu} , \quad \tau_{xy}^+ = \frac{\tau_{xy}}{\tau_w}$$



Turbulence models

Spalart-Allmaras turbulence model

The Spalart-Allmaras model [89] consists of one transport equation

$$\partial_t \tilde{\nu} + \bar{u} \cdot \nabla \tilde{\nu} = Q(\tilde{\nu}) + \frac{c_{b2}}{c_{b3}} \nabla \tilde{\nu} \cdot \nabla \tilde{\nu} + \frac{1}{c_{b3}} \nabla \cdot [(\nu + \tilde{\nu}) \nabla \tilde{\nu}] \quad (\text{E.1})$$

where the source term $Q(\tilde{\nu})$ is

$$Q(\tilde{\nu}) = c_{b1}(1 - f_{t2})\tilde{S}\tilde{\nu} + \left(\frac{c_{b1}}{\kappa^2}f_{t2} - c_{w1}f_w\right)\left(\frac{\tilde{\nu}}{d}\right)^2 \quad (\text{E.2})$$

The eddy-viscosity is

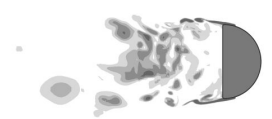
$$\nu_t = \tilde{\nu}f_{v1} \quad (\text{E.3})$$

The model damping functions, auxiliary relations and the trip term are defined as:

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}}, \quad \chi = \frac{\tilde{\nu}}{\nu} \quad (\text{E.4})$$

$$f_w = g \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right)^{\frac{1}{6}}, \quad g = r + c_{w2}(r^6 - r), \quad r = \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2} \quad (\text{E.5})$$

$$\tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 d^2} f_{v2}, \quad S = \sqrt{2S_{ij}S_{ij}}, \quad f_{t2} = c_{t3} \exp(-c_{t4}\chi^2) \quad (\text{E.6})$$



The variable d is the distance to the nearest wall, κ the von Kármán constant and the strain rate tensor is $S_{ij} = \frac{1}{2}(\partial_j u_i + \partial_i u_j)$. Finally, the model closure coefficients are

$$c_{b1} = 0.1355, \quad c_{b2} = 0.622, \quad c_{b3} = 2/3, \quad c_{v1} = 7.1 \quad (\text{E.7})$$

$$c_{w1} = \frac{c_{b1}}{\kappa^2} + \frac{1 + c_{b2}}{c_{b3}}, \quad c_{w2} = 0.3, \quad c_{w3} = 2, \quad c_{t3} = 1.2, \quad c_{t4} = 0.5 \quad (\text{E.8})$$

The wall boundary condition is:

$$\tilde{\nu} = 0 \quad (\text{E.9})$$

k - ω model

In the Wilcox's original k - ω model [96], the eddy-viscosity is defined as:

$$\nu_t = k/\omega \quad (\text{E.10})$$

The equation for turbulent kinetic energy is

$$\partial_t k + u \cdot \nabla k = P_k - C_\mu \omega k + \nabla \cdot [(\nu + \sigma_k \nu_t) \nabla k] \quad (\text{E.11})$$

where

$$P_k = \nu_t S^2, \quad S = \sqrt{2S_{ij}S_{ij}} \quad (\text{E.12})$$

The equation for the specific dissipation rate ω is:

$$\partial_t \omega + u \cdot \nabla \omega = \frac{\gamma \omega}{k} P_k - \beta \omega^2 + \nabla \cdot [(\nu + \sigma_\omega \nu_t) \nabla \omega]. \quad (\text{E.13})$$

The original model constants are

$$\sigma_k = \sigma_\omega = 0.5; \quad \gamma = 5/9; \quad \beta = 0.075; \quad C_\mu = 0.09.$$

The wall boundary condition for k is:

$$k = 0. \quad (\text{E.14})$$

At the wall, the specific dissipation rate ω asymptotically tends to infinity as $\sim 1/y^2$. The boundary condition used here is

$$\omega = \frac{60\nu}{\beta d_1^2} \quad (\text{E.15})$$

where d_1 is the distance from the wall to the cell center of the first cell above the wall.

APPENDIX F

Movie of the flow past a hemisphere at $Re=3,000$

When simulating 3-D unsteady flows, one of the most rewarding outputs is probably the possibility to look at a movie of the actual flow: almost exactly as one would observe in an experimental facility. Since this thesis is essentially focused on unsteady flows, we thought it would be nice to include such a movie in the manuscript. This is what we have done using one of the oldest technologies in the world of animation: the flip book.

Like motion pictures, flip books rely on persistence of vision to create the illusion of continuous motion. By using a series of pictures that vary gradually from one page to the next, the pictures appear to be animated when turning the pages rapidly. Rather than “reading” from left to right, the viewer simply stares at the same location of the pictures in the flip book as the pages turn. The book must also be flipped with enough speed for the illusion to work, so the standard way to “read” a flip book is to hold the book with one hand and flip through its pages with the thumb of the other hand.

We here present the animated evolution of the flow past a hemisphere at $Re = 3,000$ presented in Chapter 5. The frame on this page corresponds to the initial frame: using your thumb, turn the pages rapidly towards the beginning of the document, and enjoy the movie!

