Large-Eddy Simulation of Turbulent Channel Flow

Timofey Mukha, Mattias Liefvendahl

Department of Information Technology
Uppsala University
Box 337, SE-751 05 Uppsala, Sweden

Technical report 2015-014
May 2015
ISSN 1404-3203

Contents

Chapter 1

Introduction

Results and analysis are reported from a simulation campaign, using large-eddy simulation (LES), of fully developed turbulent channel flow. Channel flow is a classical model problem for the investigation of wall-bounded turbulence, and it has been extensively studied using both experimental and computational methods, [11, 7]. The flow takes place between two infinite parallel planes, and it is driven by a constant pressure gradient. Due to the particularly simple geometry, specially designed spectral methods can be applied to the problem, [9, 7, 10, 3, 5], which provides extremely good accuracy and allows for direct numerical simulation (DNS), i.e. without turbulence modelling, up to moderate Reynolds numbers (at least $\mathrm{Re}_\tau > 2000$).

The purpose of the present study is to evaluate the predictive capability of the numerical methods and turbulence modelling (a particular LES-subgrid model is tested) implemented in the general-purpose CFD-software `OpenFOAM`[1]. The subgrid model is based on the ideas first proposed in [14], and relies on a subgrid viscosity which is computed using an additional transport equation for the subgrid kinetic energy, see section 2.3 for a detailed description. One Reynolds number is investigated, $\mathrm{Re}_b = 13\,350$, which roughly corresponds to $\mathrm{Re}_\tau = 395$, see the discussion in section 3.1, for the parameters of the problem and the definition of these numbers. Corresponding simulations are carried out on three grid refinement levels in order to investigate grid convergence, and how the subgrid model behaves with varying grid size. The coarsest grid contains 135 000 cells and the finest grid $8.65{\cdot}10^6$ cells. The results obtained are naturally not as accurate as those obtained by spectral methods, at the same computational cost. But, since the methods employed here can be applied to general configurations/geometries, it is essential to have a good understanding of their predictive capability for wall-bounded turbulence, and channel flow is an ideal case for this investigation.

A wide range of turbulence statistics are computed, and are included in the mesh refinement study. Due to the geometry/symmetry of the problem, all statistics only depend on the wall-normal coordinate. The resulting profiles are presented for the following statistical quantities. The mean (streamwise) velocity and all components of the Reynolds stress tensor, hence including the level of turbulent fluctuations of the velocity components. Higher order statistical moments, skewness and flatness, are also computed for all three velocity components. Profiles of the vorticity fluctuations are included as well. For all these quantities, the statistics are computed using time series at a given spatial location. In addition to this, the two-point correlations of velocity fluctuations have been

[1] Disclaimer: this study is not approved or endorsed by OpenCFD Limited, the producer of the `OpenFOAM` software and owner of the OPENFOAM® and OpenCFD® trade marks.

computed, at a number of wall-normal distances. The results are compared with DNS-data, [10], and the analysis provides detailed information on the accuracy obtained for the computed turbulence statistics and how this depends on the mesh resolution.

This study should be useful for CFD-users of `OpenFOAM`, and similar software, and provides detailed information concerning grid requirements, and other settings, for problems where turbulent boundary layers play an important role. All of the data presented in the report is made available online[2] along with complete `OpenFOAM` simulation case set-up files.

[2] `https://bitbucket.org/lesituu/channel_flow_data`

Chapter 2

Turbulence modelling and numerical methods

The mathematical model and numerical methods employed in the study are described in this chapter. This includes a description of the underlying governing equations, the approach used towards turbulence modelling, the numerical methods employed for discretizing the equations, and the algorithm employed by the solver program to take the pressure-velocity coupling into account.

2.1 Governing equations

The momentum equation for an incompressible viscous fluid is the (incompressible) Navier-Stokes equation which, in the absence of external forces, is given by,

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) = -\frac{1}{\rho} \nabla \tilde{p} + \nu \Delta \mathbf{u}. \tag{2.1}$$

Here $\mathbf{u}$ is the velocity of the fluid, ρ is the density, $\tilde{p}$ is the pressure, and ν is the kinematic viscosity. The incompressibility of the fluid is expressed by,

$$\nabla \cdot \mathbf{u} = 0. \tag{2.2}$$

Equations (2.1) and (2.2) constitute the governing equations which fully describe the fluid flow. Note that ρ and ν are assumed constant. Below, $p = \tilde{p}/\rho$ will be referred to as the "pressure" for brevity. Both the notation, (u, v, w), and the notation, (u_1, u_2, u_3) will be used for the Cartesian components of the velocity vector.

2.2 Large-eddy simulation

The need for a different approach in turbulent fluid flow simulation rather then attempting to solve (2.1)-(2.2) directly stems from the fact that resolving all the relevant scales of turbulent motion requires both spatial and temporal resolutions that are incompatible with the computational resources currently available.

The key idea of LES is to directly calculate the evolution of only those eddies that are associated with length scales larger than a certain given length scale Δ. This makes using reasonably coarse computational meshes possible as well as increasing the time-step. Formally, the separation of scales is achieved via a filtering operation which is mathematically expressed as a convolution of the relevant flow field with a chosen filter kernel,

$$\overline{\phi}(\mathbf{x},t) = \iiint_{-\infty}^{\infty} \phi(\mathbf{x},t) G\left(\mathbf{x}-\boldsymbol{\xi},\Delta\right) \mathrm{d}^3\boldsymbol{\xi}, \tag{2.3}$$

where G is the kernel and Δ is the filter's cut-off width, which is a parameter that defines the size of the scales that are filtered out. Throughout the report we will use the over-bar to denote filtered variables. The unresolved part of the flow field, which is left out after the filtering is defined as,

$$\phi''(\mathbf{x},t) = \phi(\mathbf{x},t) - \overline{\phi}(\mathbf{x},t). \tag{2.4}$$

The length scales associated with the unresolved part are commonly referred to as subgrid scales (SGS).

A large variety of filters with different properties is described in the literature. For an in-depth discussion of the most important such filters, we refer to, [11, 13]. In spite of the large number of filters available, most of them are difficult to apply in a general-purpose CFD-code. The filter most commonly used in conjunction with finite volume discretization is the top-hat filter, see page 99 of [16],

$$G\left(\mathbf{x}-\boldsymbol{\xi},\Delta\right) = \begin{cases} 1/\Delta^3, & |\mathbf{x}-\boldsymbol{\xi}| \leq \Delta/2 \\ 0, & \text{otherwise.} \end{cases} \tag{2.5}$$

It is evident from (2.5) that the filtering gives a value which is an average over a rectangular volume Δ^3. A common choice for Δ is the cubic root of the volume of the (local) computational cell,

$$\Delta = \sqrt[3]{\Delta x \Delta y \Delta z}, \tag{2.6}$$

where Δx, Δy and Δz are the cell-sizes along the corresponding coordinate axes. This choice of Δ conveniently makes $\overline{\phi}$ equal to the average value of ϕ in the computational cell. This implies that no explicit filtering needs to be performed during the computational procedure, instead the filtering is built into the discretization method itself. Due to this attractive property the combination of the top-hat filter (2.5) and the Δ defined as (2.6) is used in the channel flow simulations described in this report.

By formally applying the filtering operation to the continuity equation (2.2) and Navier-Stokes equations (2.1) it is possible to derive conservation laws for the filtered flow variables. When doing this, it is convenient to use indicial tensor notation, which leads to more compact expressions.

Due to the linearity of the continuity equation, applying the filtering is straightforward. The form of the equation remains unchanged,

$$\frac{\partial \overline{u}_i}{\partial x_i} = 0. \tag{2.7}$$

This result implies that the SGS velocity field $\mathbf{u}''$ is also solenoidal. Filtering the Navier-Stokes equations results in the following.

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial}{\partial x_j}(\overline{u_i u_j}) = -\frac{\partial \overline{p}}{\partial x_i} + \nu \frac{\partial^2 \overline{u}_i}{\partial x_j \partial x_j} \tag{2.8}$$

The main complication here is that the advection term in (2.8) cannot be expressed in terms of $\overline{u}_i$. The common way to address this issue is to introduce the so called SGS stress tensor $\mathbf{B}$, the components of which are defined by,

$$B_{ij} = \overline{u_i u_j} - \overline{u}_i \overline{u}_j. \tag{2.9}$$

Inserting this into equation (2.8) leads to,

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial}{\partial x_j}(\overline{u}_i \overline{u}_j) = -\frac{\partial \overline{p}}{\partial x_i} - \frac{\partial B_{ij}}{\partial x_j} + \nu \frac{\partial^2 \overline{u}_i}{\partial x_j \partial x_j}. \tag{2.10}$$

In order to close this system of equations, $\mathbf{B}$ has to be modelled.

2.3 Subgrid stress modelling

A rich variety of approaches to modelling $\mathbf{B}$ has been developed. A thorough description of many of them can be found in [13]. Only a small amount of the models proposed in the literature has been implemented in general-purpose CFD packages. The reason is that the implementation of some of the models is often difficult, or impossible, to combine with the general framework of the code or the discretization practices the code is based on.

A common approach to SGS-modelling it to employ the Boussinesq assumption, which is the hypothesis that the SGS-stress can be modelled in a structurally similar way to the viscous stress, [13]. The analogous idea is also used in most RANS turbulence models. Mathematically it can be expressed as,

$$\mathbf{B} = \frac{1}{3}\mathrm{Tr}(\mathbf{B})\mathbf{I} + \nu_{sgs}\left(\nabla \mathbf{u} + \nabla^T \mathbf{u}\right), \tag{2.11}$$

where Tr($\mathbf{B}$) stands for the trace of the tensor $\mathbf{B}$, $\mathbf{I}$ is the identity matrix, and ν_{sgs} is the so called SGS viscosity which, in turn, is to be computed from the filtered velocity field.

Assuming (2.11), the task is then to obtain a way of calculating ν_{sgs}. In order to be able to do that, one has to adopt the hypothesis, that a characteristic length scale l_{sgs} and time scale t_{sgs} are sufficient to describe the subgrid scales, [13]. Then, based on dimensional grounds, the SGS viscosity can be calculated as

$$\nu_{sgs} \sim \frac{l_{sgs}^2}{t_{sgs}} = u_{sgs} l_{sgs}, \tag{2.12}$$

where u_{sgs} is the corresponding velocity scale. A natural choice for l_{sgs} is the filter cut-off width Δ. The choice of u_{sgs} is less obvious and different models use different approaches.

Here we use a model based on solving a transport equation for the subgrid turbulent kinetic energy k_{sgs}, which was proposed independently by several researchers, [2, 6, 14, 18, 19, 15]. The natural choice for the characteristic velocity scale is then $u_{sgs} = \sqrt{k_{sgs}}$.

The transport equation for k_{sgs} is,

$$\frac{\partial k_{sgs}}{\partial t} + \frac{\partial \overline{u}_i k_{sgs}}{\partial x_i} = 2\nu_{sgs}|\overline{D}_{ij}|^2 - C_e \frac{k_{sgs}^{3/2}}{\Delta} + \frac{\partial}{\partial x_i}\left(\nu_{sgs}\frac{\partial k_{sgs}}{\partial x_i}\right) + \nu \frac{\partial^2 k_{sgs}}{\partial x_i \partial x_i}, \tag{2.13}$$

where $\overline{D}_{ij}$ is the filtered rate of strain tensor, and $C_e = 1.048$ is a constant. The expression for ν_{sgs} is then taken to be,

$$\nu_{sgs} = C_k \Delta \sqrt{k_{sgs}}, \tag{2.14}$$

where, $C_k = 0.094$, is another model constant.

Physically the four terms on the right-hand side of (2.13) represent, respectively, the production of turbulence by the resolved scales, turbulent dissipation, turbulent diffusion, and viscous dissipation. More details on the derivation of (2.13) and the employed modelling assumptions can be found on page 128 in [13]. Note that, $k_{sgs} = \mathrm{Tr}(\mathbf{B})/2$, and therefore can be used for calculating the isotropic subgrid stresses.

When defined as (2.4), ν_{sgs} does not exhibit correct behaviour in the limit $y \to 0$, where y is the distance from a wall. To rectify this problem a damping function can be employed. In the simulations presented here the van Driest damping function is used. It has the following form.

$$f = \frac{\kappa}{C_\Delta}\left(1 - e^{-y^+/A^+} y\right) \tag{2.15}$$

Here, $\kappa = 0.41$, is the von Karman constant, $C_\Delta = 0.158$, and, $A^+ = 26$.

2.4 Discretization and interpolation methods

As many other contemporary CFD codes, `OpenFOAM` is based on the finite volume method for disretizing partial differential equations. The method itself, and its application to the Navier-Stokes equations, are thoroughly described in a large number of monographs dedicated to numerical methods and fluid flow modelling. In particular the books, [16, 1], provide a good introduction to the method in the context of CFD, and a detailed `OpenFOAM`-oriented discussion can be found in, [4, 12, 17]. In this section only a short overview of the method will be presented.

The finite volume method is based upon dividing the computational domain into many small non-intersecting polyhedra called control volumes (CVs). Different approaches exist, but in `OpenFOAM` all variables are stored at the centroids of the control volumes. The value at the centroid represents the whole CV. This can be easily shown to be a second order accurate approximation, see [1].

Derivation of the descretized form of the equations begins with integrating the original equations over a control volume and a time interval Δt. Then, where it is applicable, the Gauss theorem is used to convert volume integrals into surface integrals.

The procedure can be illustrated using a convection-diffusion equation for some quantity ϕ,

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = \nabla \cdot (\Gamma \nabla \phi), \tag{2.16}$$

which is integrated to,

$$\int_t^{t+\Delta t} \left(\frac{\partial}{\partial t} \int_{\mathrm{CV}} \phi \, \mathrm{d}V + \oint_{S_{\mathrm{CV}}} \phi \mathbf{u} \cdot \mathbf{n} \, \mathrm{d}S - \oint_{S_{\mathrm{CV}}} \Gamma \nabla \phi \cdot \mathbf{n} \, \mathrm{d}S \right) dt = 0. \tag{2.17}$$

The equation above is an integral form expressing the conservation of the quantity ϕ. If this is summed over all the control volumes a conservation law for the whole domain is obtained. As a consequence, the finite volume method is intrinsically conservative, which is one of its most attractive properties.

In order to get algebraic equations, the volume integral in (2.17) is approximated as, $\phi_P V_{\text{CV}}$, while surface integration is approximated as a sum over the faces of the control volume:

$$\oint_{S_{\text{CV}}} \phi \mathbf{u} \cdot \mathbf{n} \, \mathrm{d}S \approx \sum_f \left(\mathbf{u}_f \cdot \mathbf{n}\right) S_f \phi_f, \tag{2.18}$$

$$\oint_{S_{\text{CV}}} \Gamma \nabla \phi \cdot \mathbf{n} \, \mathrm{d}S \approx \sum_f \left(\nabla_f \phi \cdot \mathbf{n}\right) S_f \Gamma_f. \tag{2.19}$$

The index f in the above equations stands for the value in the centroid of the face with the corresponding index. These values are unknown and have to be interpolated using the values in the centroids of the cells.

Choosing spatial interpolation and time-marching schemes has to be done with care in order to have a good balance between accuracy and stability. In the simulations presented here a second order backward differencing scheme was used for time marching. To calculate the time derivative in (2.17), the scheme uses the unknown value of ϕ from the current time-step, ϕ^n, and the values from the two preceding time-steps, ϕ^{n-1} and ϕ^{n-2},

$$\frac{\partial \phi_i}{\partial t} \approx \frac{\frac{3}{2}\phi^n - 2\phi^{n-1} + \frac{1}{2}\phi^{n-2}}{\Delta t}, \tag{2.20}$$

where Δt is the time-step. The time integrals of the convective and diffusive terms in (2.17) are approximated by the (to be determined) values at the current time-step, which makes the time-marching scheme fully implicit. Additional information on the second order backward differencing scheme can be found in [4]. The `OpenFOAM`-keyword for this scheme is `backward`.

The time-step Δt was chosen to be small enough to keep the Courant number below one. For meshes M1 and M2 the value of, $\Delta t = 0.2$s, was sufficient, but for the finest mesh M3 it had to be lowered down to 0.08s.

The face-fluxes of momentum were calculated using a linear interpolation scheme (referred to as `linear` in `OpenFOAM`). The same scheme was also used to evaluate the values of the gradients in the centroids of the faces.

The face-fluxes of the subgrid scale turbulent kinetic energy were calculated using a TVD interpolation scheme based on upwind and central differencing, (`linearLimited 1` in `OpenFOAM`). The scheme is based on a flux limiter of the form, $\max(\min(2r, 1), 0)$, where r is the ratio of successive gradients.

2.5 Solver algorithm

The solver `pimpleFoam`, provided as part of `OpenFOAM`, was used to solve the equations derived in sections 2.2 and 2.3. The solver is capable of treating a variety of different flow problems and employ different types of turbulence modelling approaches. Here we give a short overview of the principal components of the underlying algorithm. More information can also be found in [4].

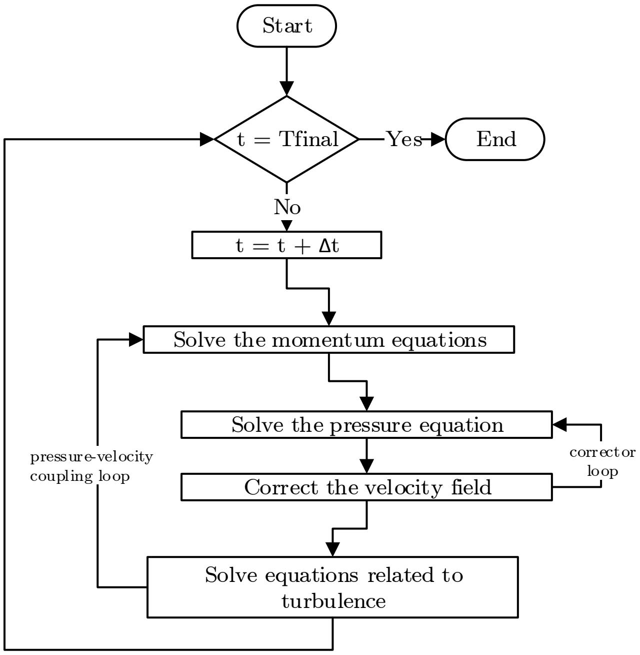
Start
t = Tfinal
Yes
End
No
t = t + Δt
Solve the momentum equations
Solve the pressure equation
corrector loop
pressure-velocity coupling loop
Correct the velocity field
Solve equations related to turbulence

Figure 2.1: Essential steps of the algorithm implemented in the `pimpleFoam`-solver.

The algorithm implemented in the solver is based on a blend of the transient SIMPLE and PISO algorithms, a thorough description of which can be found in [16, 1]. The most important steps of the algorithm solver are shown in figure 2.1.

At the beginning of each time-step, the algorithm increases the current simulation time by the value of the time-step. Then the pressure-velocity coupling loop is executed. Inside the loop, the momentum equation is solved first, after which the corrector loop is entered. Inside the corrector loop, the pressure equation is solved and the velocity field is corrected ensuring that it is divergence-free. Finally, all equations related to turbulence modelling are solved.

It is possible to regulate how many times the pressure-velocity coupling loop is executed. In case only a single iteration is performed, `pimpleFoam`'s algorithm is identical to the PISO-algorithm. Analogously, it is possible to change the number of times the corrector loop is executed. If the loop is executed only once, `pimpleFoam` implements the transient SIMPLE-algorithm.

Increasing the number of iterations inside the pressure-velocity coupling loop, in combination with heavy under-relaxation, makes using larger time-steps possible, which can decrease the total computation time. As indicated above, in the simulations presented in this report, the time-step was set to be small enough to keep the Courant number below one in the whole domain. Therefore, the number of the coupling loop's iterations was set to one. The number of corrector loop iterations was set to two.

Chapter 3

Simulation case set-up

The formulation of the continuous problem and the computational set-up, including physical and numerical parameters, are described in this chapter. The mesh generation and the simulation campaign are also summarized.

3.1 The continuous problem and the physical parameters

Fully developed channel flow is a theoretical construct consisting of a flow between two infinite parallel planes, driven by a constant pressure gradient. It is convenient to describe the flow using the cartesian coordinate system shown in figure 3.1. Let the x-axis be in the direction of the negative pressure gradient. This direction coincides with that of the mean flow and will therefore be referred to as the streamwise direction. The y-axis is taken to be orthogonal to the walls of the channel, pointing from the lower wall to the upper. This direction is referred to as wall-normal. Finally, the z-axis is chosen so that (x, y, z) forms an orthonormal coordinate system. The z-direction is referred to as spanwise.

Since the walls are of infinite size, the geometry of channel flow is fully characterized by one parameter, h, the channel width. However, in a computational experiment, the domain has to be bounded in the streamwise and spanwise directions as well (more on this in section 3.3). The introduction of this artificial truncation introduces two more geometrical parameters, the streamwise truncation length, l_x, and the spanwise truncation length, l_z. The values of l_x and l_z should be large enough to fit the largest existing turbulent structures inside the domain. The values used in the simulations presented here are adopted from [9, 7]. Their adequacy is also confirmed by an a posteriori analysis of two-point autocorrelations of the velocity field presented below, in section 4.7.

The physical parameters of the flow are the driving pressure gradient and the kinematic viscosity of the fluid, ν. When these parameters are specified, the problem is well-defined. As an alternative to the pressure gradient, the (mean) bulk velocity,

$$U_b = \frac{1}{h} \int_0^h \langle u \rangle \, \mathrm{d}y, \tag{3.1}$$

can be prescribed instead. The benefit of using U_b would be that the bulk Reynolds number, $\mathrm{Re}_b = U_b h/\nu$, then would be defined by the input parameters.

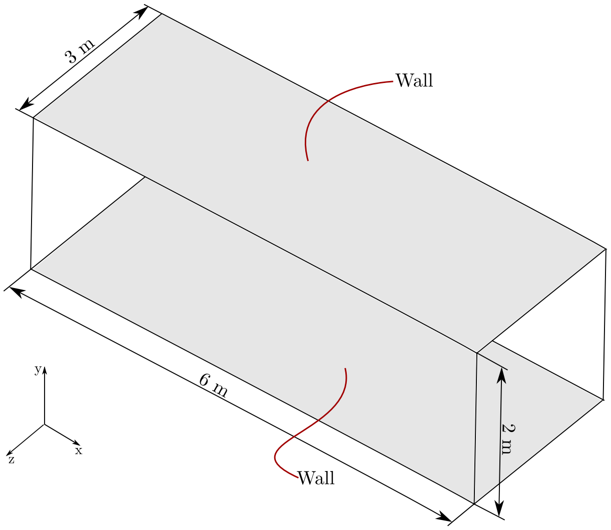

Figure 3.1: Illustration of the channel configuration, the computational domain and the coordinate system.

In addition to the bulk velocity, another characteristic velocity scale is commonly introduced in connection with channel flow. The friction velocity, u_τ, is defined in terms of the wall shear stress, τ_w, and the fluid density, ρ, according to,

$$u_\tau = \sqrt{\tau_w/\rho}.$$

The friction Reynolds number is then defined as,

$$\mathrm{Re}_\tau = \frac{u_\tau \delta}{\nu},$$

where, $\delta = h/2$, is the channel half-width. It is easy to show, see page 267 of [11], that the gradient of the pressure and the wall shear stress are related,

$$-\frac{d\tilde{p}}{dx} = \frac{\tau_w}{\delta}. \tag{3.2}$$

Consequently, prescribing the pressure gradient strictly defines Re_τ. The conclusion is that the choice between the pressure gradient and the bulk velocity as the defining parameter should, at least theoretically, be based on whether it is desirable to have Re_τ or Re_b defined as input. The quantity that is undefined will then have to be computed.

As discussed in section 4.1, the benchmark for evaluating the accuracy of the simulations is DNS data from [10], computed at $\mathrm{Re}_\tau = 395$. This suggests using the same Re_τ as an input parameter. However, important practical considerations suggested doing the opposite. Namely, the computational procedures associated with choosing U_b as input (see section 3.4) were already implemented in `OpenFOAM` and tested by the community. Therefore in the simulations reported here, Re_b is defined and Re_τ is computed.

As a consequence, the value of Re_b had to be chosen to be corresponding to the target value of, $Re_\tau = 395$. This was not straightforward because, in [10], the authors do not provide information regarding U_b or Re_b. However, in [17], the author also compares results from channel flow simulations to a DNS database computed for, $\mathrm{Re}_\tau = 395$, and provides the used U_b and ν. Here the same values are used. The values of Re_τ that were computed in the simulations are presented and discussed in chapter 4, the conclusion is that the chosen value of Re_b is adequate.

The geometrical and physical parameters are summarized in table 3.1.

Quantity	**Notation**	**Value**	**Unit**	**Expression**
Channel width	h	2.0	m	2δ
Streamwise length	l_x	6.0	m	- - -
Spanwise length	l_z	3.0	m	- - -
Kinematic viscosity	ν	$2 \cdot 10^{-5}$	m^2/s	
Bulk velocity	U_b	0.1335	m/s	- - -
Bulk Reynolds number	Re_b	13 350	- - -	$U_b h/\nu$

Table 3.1: Physical and geometrical parameters.

3.2 Mesh generation

The geometrical simplicity of channel flow makes it easy to construct structured hexahedral computational meshes. One of the major goals of this study is to investigate the effects of the grid size on the results of the simulation. Three different computational meshes are used. These meshes, as well as the simulations that use them, will be referred to as M1, M2 and M3. Using the same name for the meshes and the simulations does not introduce confusion, since the mesh is the only parameter that distinguishes the simulations from one another.

Mesh M1 is the coarsest of the three. Mesh M2 contains eight times more cells, the grid spacing along each axis being reduced by a factor of two, as compared to M1. Mesh M2 is refined in the same way, leading to the finest grid M3, with eight times more cells than M2. The mesh information is summarized in table 3.2. Note that Δx^+, Δz^+ and y^+ are calculated using a theoretical value of u_τ, based on the value of $\mathrm{Re}_\tau = 395$.

Name	**Cells along axes**	**Total size**	$\boldsymbol{\Delta x^+}$	$\boldsymbol{\Delta z^+}$	$\boldsymbol{y^+}$**, first cell**
M1	60×50×45	135 000	39.50	26.30	1.90
M2	120×100×90	1 080 000	19.75	13.16	0.96
M3	240×200×180	8 640 000	9.88	6.58	0.47

Table 3.2: The computational meshes used in the simulations.

In order to increase the resolution of the turbulent structures, and the large gradients occurring in the near-wall region, a bias was applied to the y-grading of the meshes, see figure 3.2. The bias was defined through the ratio between the largest and the smallest cell-size along y, which was set to 10.7028.

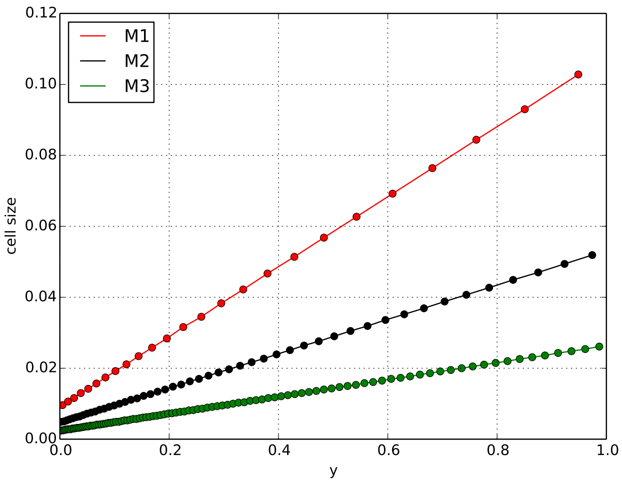

Figure 3.2: Cell size as function of y, for the computational meshes M1, M2 and M3 respectively. The points show the locations of the cell centers.

3.3 Boundary conditions

In order to simulate a domain of infinite size in the streamwise and spanwise directions, two pairs of periodic boundary conditions are introduced. The first pair connects the boundaries at $x = 0$ and $x = l_x$, and the second the boundaries at $z = 0$ and $z = l_z$. The remaining two boundaries represent walls. For the resolved velocity field, $\overline{\boldsymbol{u}}$, the wall induces a no-slip boundary condition,

$$\overline{\boldsymbol{u}} = 0.$$

This implies a von Neumann condition for the pressure. The no-slip condition also applies to the unresolved fluctuations of velocity, $\boldsymbol{u}_i''$, which entails that $k_{sgs} = 0$ everywhere on the wall.

Field	**Type**	**Value**
$\overline{u}_i$	Dirichlet	0
k_{sgs}	Dirichlet	0
p	von Neumann	0

Table 3.3: Boundary conditions at the walls.

3.4 Modelling the pressure gradient

As explained above, the pressure gradient is not used as a defining parameter for the simulations. Instead, the bulk velocity U_b is. However, the pressure gradient has to somehow be computed, and its effect taken into account. The problem is resolved by introducing an additional external force term into the momentum equation. This artificial force drives the flow, and the magnitude of the force is determined by the prescribed bulk velocity. At each time step, the actual U_b is re-calculated, and an adjustment to the magnitude of the external force is made, to correct the value.

3.5 The simulation campaign

Before time averaging could be started, all the transient processes related to initial conditions had to pass away. The simulation for each mesh size therefore consisted of two phases. During the first, preliminary phase, the sampling of the statistical quantities of interest was not performed. During the second, averaging phase, the time averaging was started, and the flow was simulated until the sampling time interval was large enough to make statistical errors insignificant.

Table 3.4 summarizes the durations of both phases of all three simulations. Time is given in three different units. Seconds, non-dimensional units, tu_τ/δ, and the number of flow-through times based on bulk velocity, $\#T_{f-t}$. The theoretical value of u_τ, based on the target friction Reynolds number, $Re_\tau = 395$, was used for calculating the non-dimensional time.

Mesh	**Preliminary phase duration**			**Averaging phase duration**		
	t[s]	tu_τ/δ	$\#T_{f-t}$	t[s]	tu_τ/δ	$\#T_{f-t}$
M1	10 000	79	222.5	90 000	711	2003
M2	20 000	158	445	90 000	711	2003
M3	10 000	79	222.5	31 000	245	690

Table 3.4: Meshes and simulation times used in the simulation campaign.

The reason for the preliminary phase to be double as long for simulation M2 compared to the other simulations is simply because the initial data for time, 20 000s, was already available due to some prior experimentations with the code. The duration of the averaging phase for M3 is shorter than that of M1 and M2 due to the computational expenses associated with transient simulations on a mesh that large. However, it was sufficient for all the relevant statistical quantities to converge. The simulation times of M1 and M2 can thus be considered excessive.

Chapter 4

Results

In this chapter the results obtained from the conducted simulations are presented and discussed. Before proceeding, a number of preliminary remarks are made.

- The definitions of the statistical quantities investigated here are collected in the appendix A. In the appendix, the computational procedure for the statistics, using temporal and spatial averaging, is also described.
- In order to assess the accuracy of the results, data from a DNS, reported in [10], is used. The data corresponds to $\mathrm{Re}_\tau = 395$, and is available online.[1]
- All of the quantities were calculated using only the filtered part of the velocity field, $\overline{\boldsymbol{u}}$. The contribution of the subgrid scales was thus not explicitly taken into account. To simplify the notation, the over-bar used to denote the filtered quantities will be dropped in this chapter, but it will always be implied. Angular brackets $\langle\cdot\rangle$ will be used to denote the average value, and a single prime will be used for the fluctuations.
- Where appropriate, the obtained results will be shown both as a function of y/δ and as a function of y^+. In the literature, this is sometimes referred to as presenting the data in, respectively, global and wall coordinates, [7]. As the names suggest, the former is more convenient for accessing overall behaviour, and the latter for investigating the behaviour near the wall. Due to the fact that all of the profiles are either symmetric or anti-symmetric around the plane $y = \delta$, it suffices to plot the data over the interval $[0, 1]$ in global coordinates.

4.1 Global flow quantities

By global flow quantities, we mean computed scalar variables characterizing the flow, such as friction velocity and mean center-line velocity. This is the starting point of our analysis, and these results are summarized in table 4.1. The most important of them is the computed average friction velocity u_τ or, equivalently, the Reynolds number Re_τ based on that velocity. Note that we will skip the averaging brackets, $\langle\cdot\rangle$, and always denote the averaged friction velocity simply by u_τ.

[1] http://turbulence.ices.utexas.edu/MKM_1999.html

Recall that the target value of Re_τ was 395, because that corresponds to the data from the DNS simulations used for benchmarking the results.

As evident from the table, Re_τ is under-predicted in all the simulations. However, the results improve significantly with the refinement of the mesh, so there is a reason to believe that upon further refinement the computed value would eventually converge to 395.

Parameter	**M1**	**M2**	**M3**	**Comment**
Average friction velocity u_τ, m/s	0.00724	0.00746	0.00766	Target value is 0.0079
Average centreline velocity U_c, m/s	0.15174	0.15210	0.15381	—
Re_τ	362	373	383	Target value is 395
Re_c	7587	7605	7690	—
U_b/u_τ	18.44	17.90	17.43	Target value is 16.90
U_c/u_τ	20.96	20.39	20.08	—
U_c/U_b	1.14	1.14	1.15	—
Viscous length scale $\delta_\nu = \nu/u_\tau$	0.00276	0.00268	0.00261	Target value is 0.00253

Table 4.1: Computed global flow parameters.

In the following sections, most of the results are presented as scaled with u_τ. The results of each simulation are scaled with the value of u_τ obtained from that simulation. That is, a different scaling factor is applied to the results from each of the simulations. The non-dimensional wall distance y^+ is proportional to u_τ. The values of y^+ used for presenting the results are computed for each simulation individually, using the obtained values of u_τ. This means that data points located at an equal value of y^+, for all three simulations, are actually located at different values of y.

4.2 Mean velocity profiles

This section will be devoted to the inspection of the obtained profiles of the average streamwise velocity. Other components are not considered, since their mean value is zero in the entire domain. The normalized profiles of $\langle u \rangle$ are presented in figure 4.1.

In the graph on the left side of the figure, the profiles are presented in global coordinates, scaled with the bulk velocity U_b. It is somewhat hard to distinguish between the results computed on the three different meshes, but the curve corresponding to M1 does stand out from the other two. For $y/\delta < 0.2$ it lies beneath the two other profiles, but then intersects them and lies above instead. Finally, at $y/\delta \approx 0.7$ it crosses the M3-curve again and nearly coincides with the profile from M2.

To further address the issue of accuracy, it is necessary to view the same results, but in wall coordinates and scaled with the friction velocity. This representation of the data is given in the right plot in figure 4.1.

We start the analysis in the viscous sub-layer. The coarsest mesh M1 has only two points located there and thus cannot resolve it adequately. But the values predicted at the two existing points are nevertheless in excellent agreement with DNS. Simulations M2 and M3 improve on the results obtained for M1, due to the increase in mesh resolution. In M3 the viscous sub-layer is resolved well enough for linear interpolation between the computed points to accurately represent the DNS data across the whole sub-layer.

In the buffer region the results from all three simulations begin to diverge from the DNS data, specifically, the values are under-predicted. The profiles from M2 and M3 are indistinguishable in

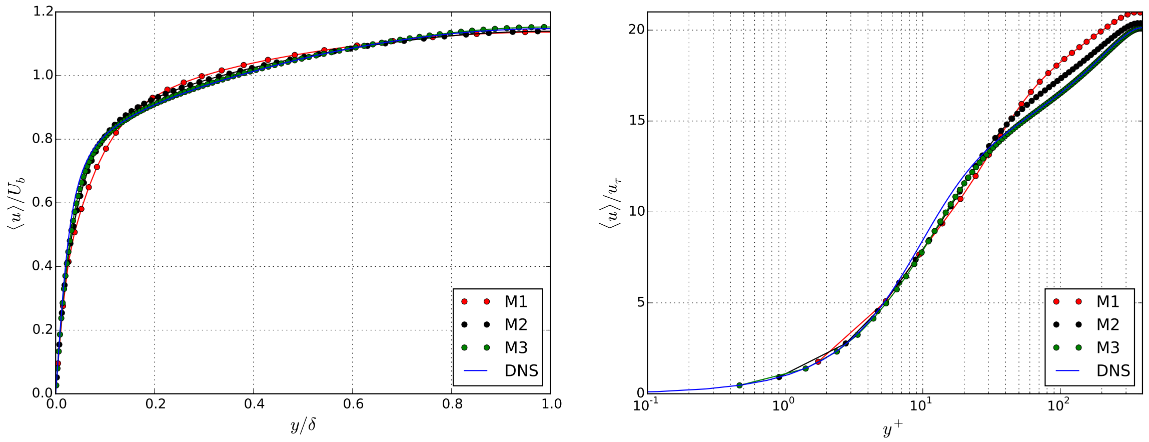

Figure 4.1: Profiles of the mean of the normalized streamwise component of velocity.

this region, and M1 performs slightly worse, leading to a larger under-prediction.

In the log-law region the simulation using finest mesh M3 gives results that are in very good agreement with DNS across the whole region. Both M2 and M1 produce results that over-predict the velocity values, but the results from M2 are more accurate.

We can conclude that the positive effect of grid refinement is evident from the gradual increase in the accuracy of the obtained results. However all three meshes, even M1, are fine enough to produce a mean velocity profile which would be accurate enough for most engineering applications.

4.3 Velocity fluctuations

The components of the Reynolds stress tensor are the primary quantities describing the turbulent fluctuations. In this and the following section, the predictions for each component of the tensor will be analysed in detail.

We begin the inspection with the diagonal components. Statistically, these components, $\langle u_i'^2\rangle$, are the variances of the components of velocity. However, it is customary to consider the standard (root-mean-square) deviations, $u_i^{rms} = \sqrt{\langle u_i'^2\rangle}$, of u_i instead. Figures 4.2-4.4 show the distribution of the standard deviation of the three components of velocity in the wall-normal direction. A common result for all three components is that the profiles converge towards the DNS data as the mesh gets refined.

The simulation on the finest mesh M3 accurately estimates the location of the maxima of the root-mean-square values. The peak-values, however, are under-estimated. Using coarser meshes leads to under-prediction of the maxima even more, and the locations of the maxima get shifted away from the wall. The peaks are also more diffused, the coarsening of the mesh prevents the sharp gradients from getting resolved. This behaviour is more pronounced for the the wall-normal and spanwise components of velocity, v and w respectively.

Closer to the core region of the channel, for $y/\delta > 0.5$, the difference in the results obtained on the different meshes is less definite. For all the components, the profiles lie slightly under the DNS data, with the exception of the profile of u^{rms}, obtained from M1, which lies above the DNS.

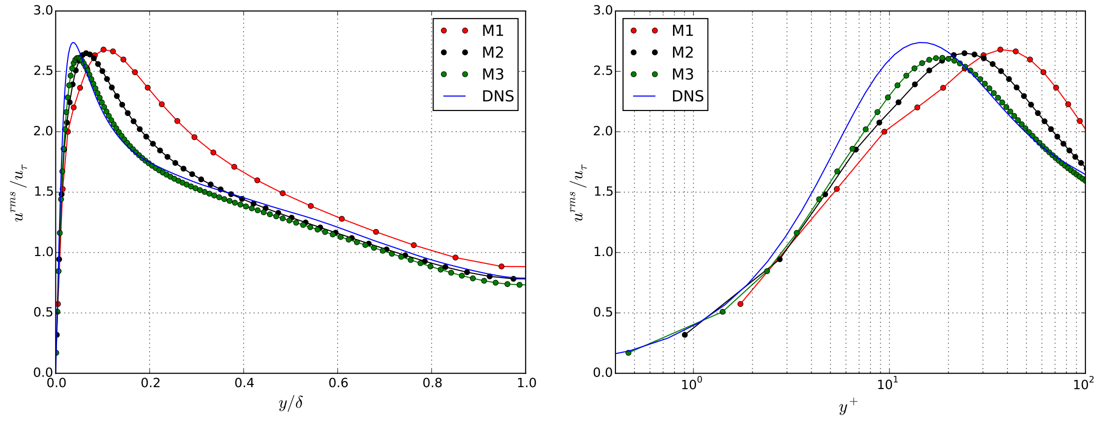

Figure 4.2: Profiles of the normalized standard deviation of the streamwise component of velocity, u^{rms}/u_τ.

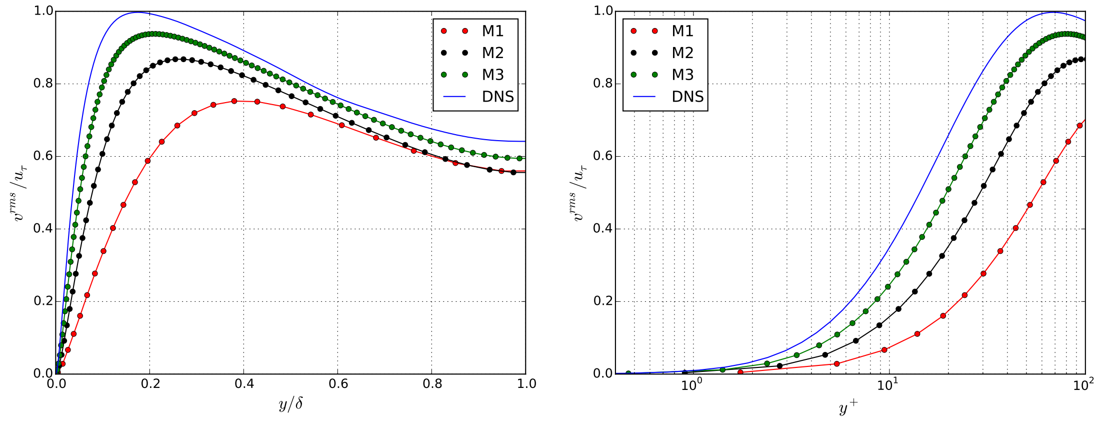

Figure 4.3: Profiles of the normalized standard deviation of the wall-normal component of velocity, v^{rms}/u_τ.

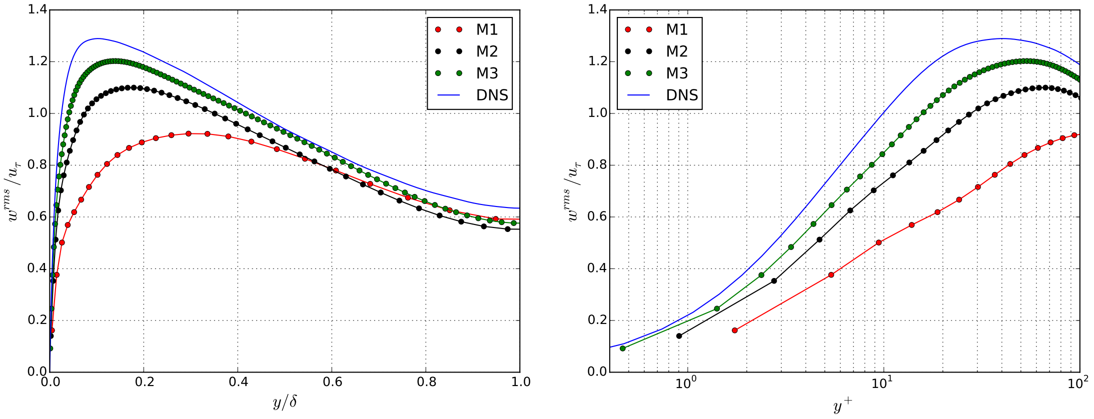

Figure 4.4: Profiles of the normalized standard deviation of the spanwise component of velocity, w^{rms}/u_τ.

4.4 Turbulent shear stress

We continue the analysis with the off-diagonal components of the Reynolds stress tensor. The symmetry of channel flow implies that the xz and yz components of the Reynolds stress tensor are equal to zero. Therefore the only component left to analyse is the xy component, the negative of which is referred to as the turbulent shear stress.

It can be easily shown analytically that for channel flow the profile of the total shear stress, i.e. the sum of the viscous and turbulent shear stresses varies linearly across the channel, see page 267 of [11]. However, viscous stresses play a significant role only in the viscous wall region, $y^+ < 50$, and consequently, outside that region, the profile of the turbulent shear stress alone can be expected to be of linear form.

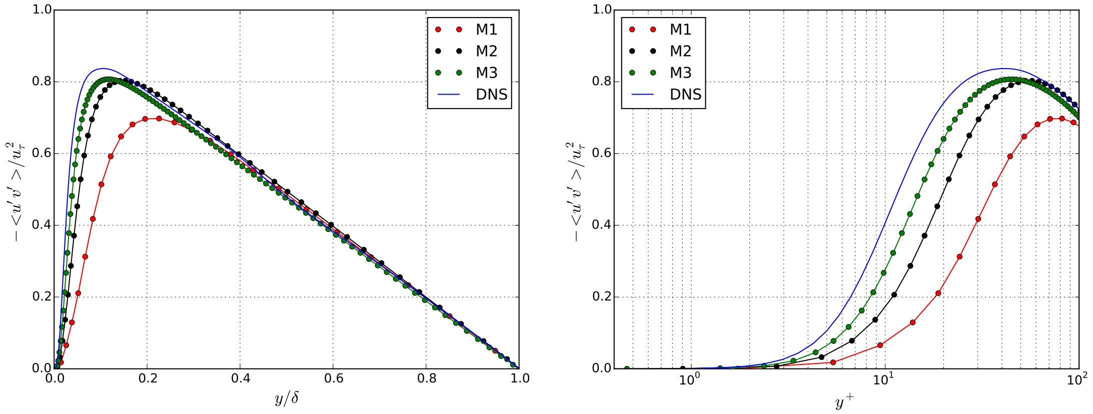

Figure 4.5: Profiles of the normalized turbulent shear stress, $-\langle u'v'\rangle/u_\tau^2$.

The profiles of the computed turbulent shear stress are presented in figure 4.5. The results are in good agreement with the theoretical analysis presented above. The profile from the DNS database has a peak at $y^+ = 30$, and assumes a linear profile for higher y^+-values. The results obtained with the finest mesh M3 predict the location of the peak with very good accuracy, but under-estimate the values of the shear stress in the viscous wall region. The results obtained with M1 and M2 are less consistent with the DNS data. Note also that the difference in accuracy between M1 and M2 is significantly larger than between M2 and M3. For $y^+ > 100$, all three meshes produces a linear profile that is in very good agreement with DNS data.

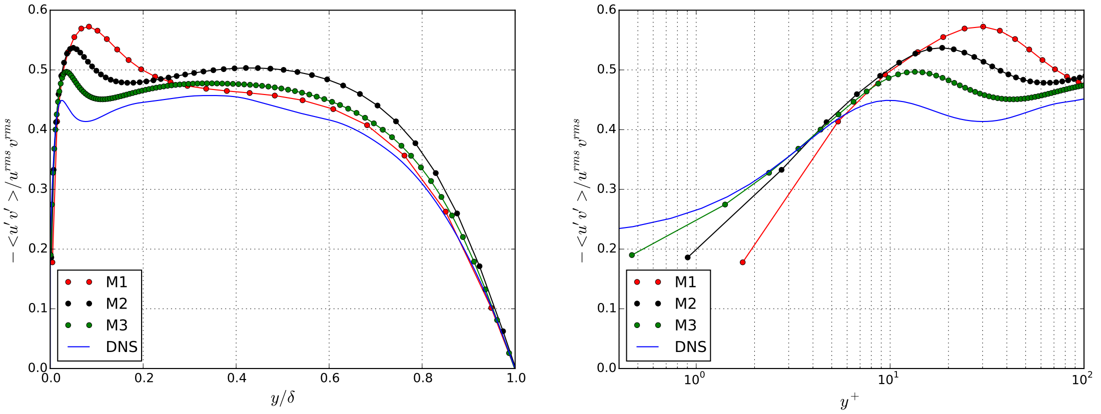

Figure 4.6: Profiles of the negative of the correlation coefficient of u and v, $-\langle u'v'\rangle/(u^{rms}v^{rms})$.

It is possible to look at the turbulent shear stress from a different angle by scaling it differently. The stress -$\langle u'v'\rangle$ is the negative of the covariance of u' and v'. Thus normalizing it with $u^{rms}v^{rms}$ will give the negative of the correlation of u' and v'. The term correlation coefficient is also used to describe the same quantity, see page 57 in [11].

Figure 4.6 shows the graphs of the obtained profiles of the negative of the correlation coefficient. It is interesting, that there is a local peak near the wall, followed by a local minimum. In [7], the presence of the peak is explained by the presence of certain “organized motion” in that region. The peak can be observed in the graphs of the profiles obtained from all three simulations. Moreover, the obtained peaks are exaggerated, and are global maxima, whereas in the DNS data the peak is only a local maximum. The simulations M3 and M2 give more accurate results in general. Specifically, they also produce the local minimum following the peak, whereas M1 fails to reproduce that behaviour.

Interestingly, M1 does perform better than the simulations on the finer meshes further away from the wall. The gain in accuracy, compared to M3, is marginal, but compared to M2, the improvement is quite significant. However, the analysis of the quantities that form the correlation coefficient, namely $\langle u'v'\rangle$, u^{rms} and v^{rms}, that has been presented in the previous sections of this chapter, showed that for each component individually, the results from M1 are less accurate than those obtained from M2 and M3. Therefore the apparent superiority of the results from M1 for the correlation coefficient should be regarded as accidental.

4.5 Skewness and flatness

We now turn to statistics of higher order, namely the skewness and flatness of the three components of velocity. Skewness measures the asymmetry of the probability density function (PDF) or, in other words, it allows to tell which tail of the PDF, the left or the right, is “fatter” or longer. Physically, this allows to conclude whether it is the high or the low values of velocity that give a larger contribution to the deviation from the mean value. Flatness can be seen as a measure of the “heaviness” of the tails. Thus, high flatness indicates that more variance is a result of strong infrequent deviations. Physically, this can be interpreted as high intermittency of the flow.

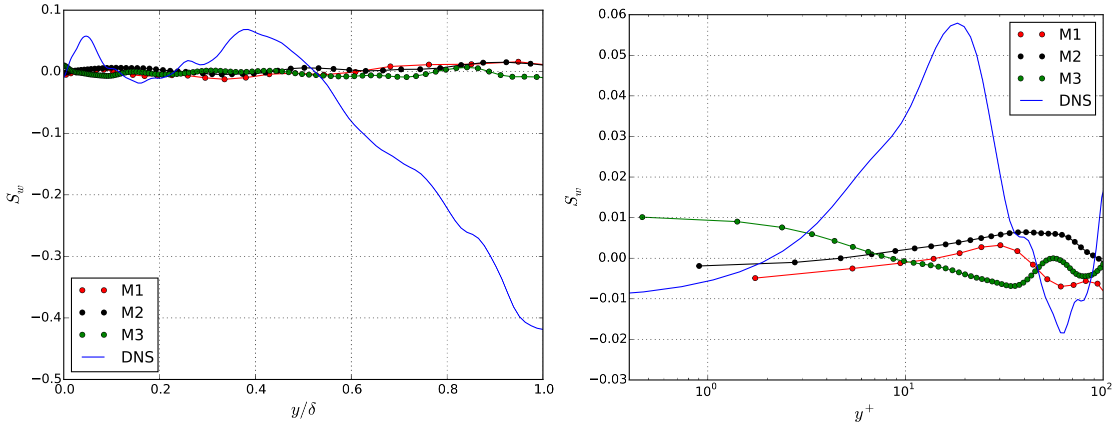

Figure 4.7: Profiles of the skewness of the spanwise component of velocity, S_w.

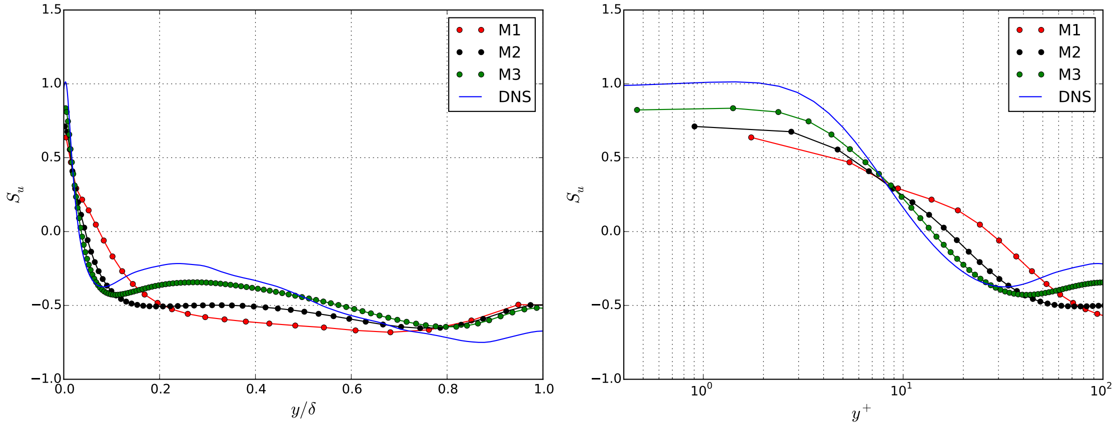

Figure 4.8: Profiles of the skewness of the streamwise component of velocity, S_u.

We first consider the skewness of the spanwise component of velocity. Due to symmetry, we should have, $S_w = 0$, throughout the channel, therefore this is a good measure of statistical convergence. The obtained profiles of S_w are shown in figure 4.7. Evidently, all three simulations produced

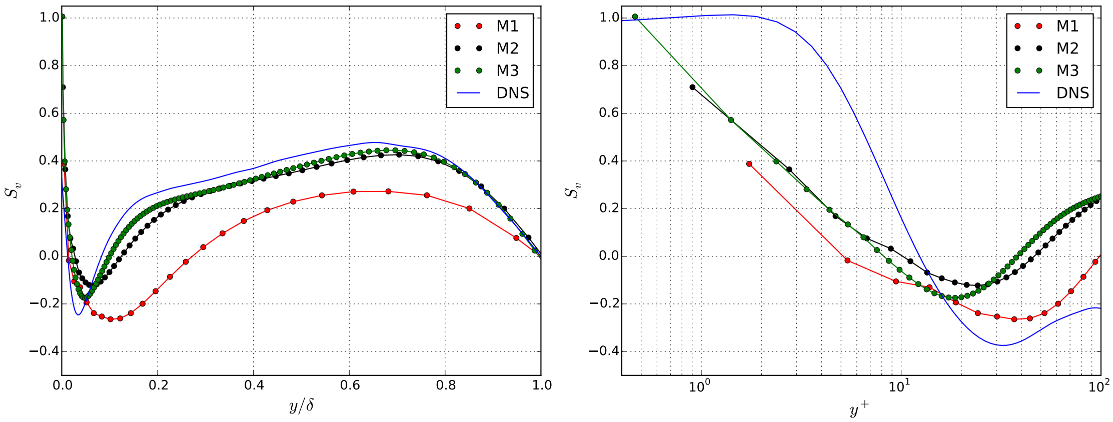

Figure 4.9: Profiles of the skewness of the wall-normal component of velocity, , S_v.

a satisfactory result. However, the DNS data is far from accurate. Especially for $y/\delta > 0.5$ the deviation from zero is large. There is no information on the sampling time in the DNS database. This result, for S_w, however indicates how far the DNS results are from being converged, and that the time interval used there is shorter than what we have employed here. It should be noted, that higher order moments converge slower, and the results in previous sections leave no doubt that the DNS data is accurate for averages and second order statistical moments.

Figure 4.8 shows the profiles of the skewness of the streamwise component of velocity. The profiles from all the simulations are in acceptable agreement with the DNS data, but only M3 reproduces all the features of the profile such as the local maximum at $y/\delta \approx 0.25$. The values of S_u are positive near the wall, but become negative at $y^+ \approx 10$. The following interpretation of this fact is found in [9]. Positive skewness indicates that the right tail of the PDF is long or fat, therefore the deviation of u from the mean is primarily due to the arrival of high-speed fluid from the core of the channel. Away from the wall the opposite occurs, and it is the low-speed fluid from the near-wall region entering the core of the channel that is responsible for most of the occurring fluctuations.

The obtained profiles for S_v are also in good agreement with the DNS data across the whole channel, aside from the viscous sub-layer. But, since the accuracy of the DNS results are subject to some doubt for higher statistical moments, it is hard to draw definite conclusions on whether the simulations have actually failed to reproduce the correct behaviour.

The computed flatness factors have their peak values in the vicinity of the wall, see figures 4.10-4.12. Especially F_v assumes high values there. This indicates that the flow near the wall is highly intermittent.

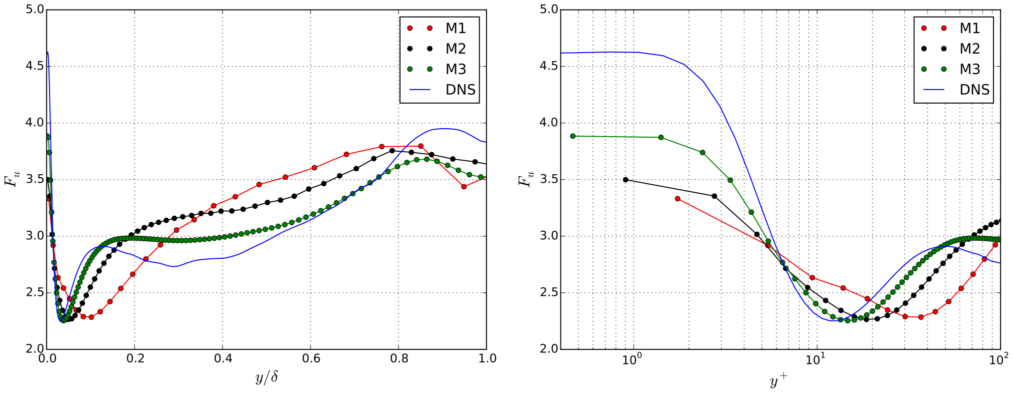

Figure 4.10: Profiles of the flatness of the streamwise component of velocity, F_u.

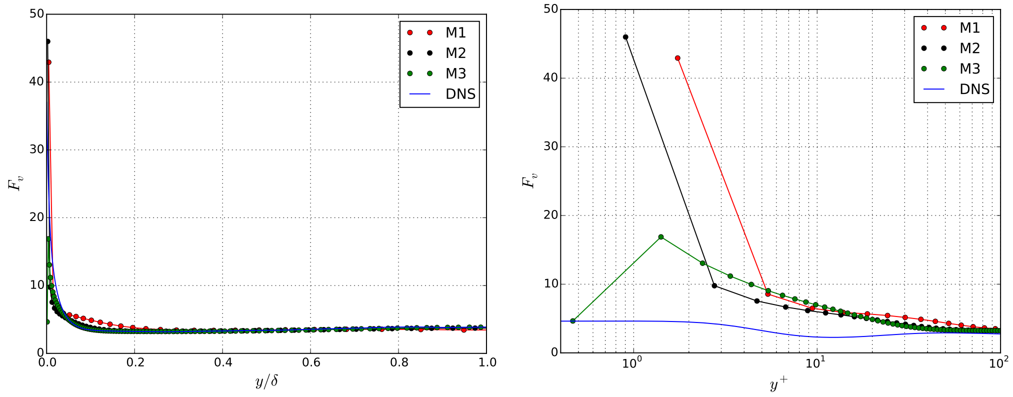

Figure 4.11: Profiles of the flatness of the wall-normal component of velocity, F_v.

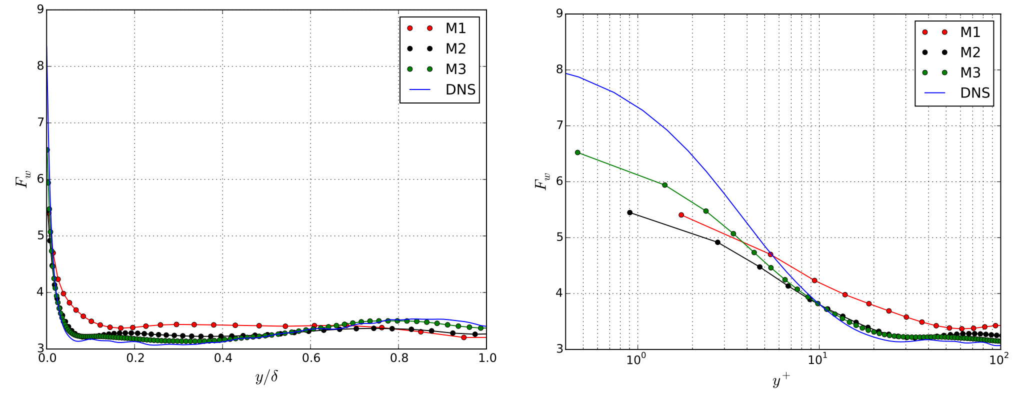

Figure 4.12: Profiles of the flatness of the spanwise component of velocity, F_w.

As with the other quantities in this section, it is difficult to evaluate the accuracy of the flatness predictions since the DNS data, which serves as benchmark, is not necessarily accurate itself for this statistical quantitiy. The agreement between the DNS and the LES profiles is quite good, which indicates that, in general, the behaviour of the flatness is represented correctly.

4.6 Vorticity fluctuations

Another quantity of interest is the standard deviation of the components of the vorticity vector, $\boldsymbol{\omega} = \nabla \times \mathbf{u}$. Analysis of vorticity profiles can lead to additional insights regarding the nature and behaviour of the vortical structures present in the flow in question.

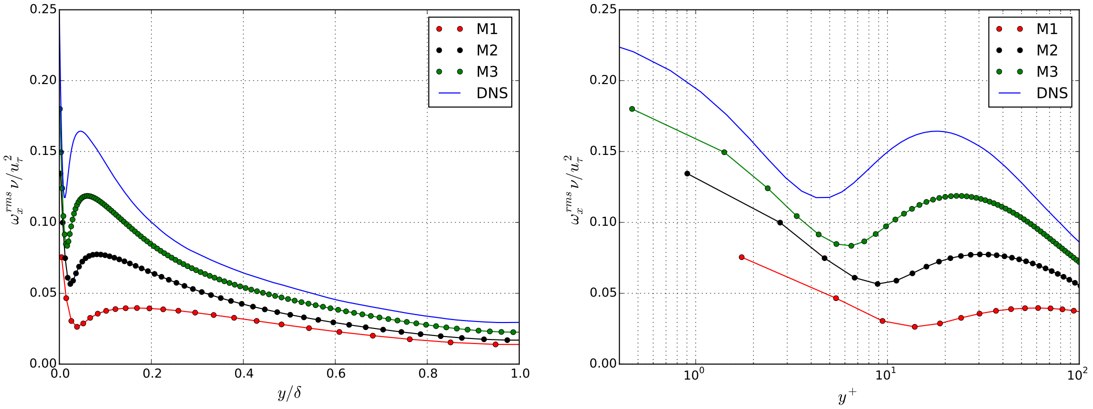

Figure 4.13: Profiles of the normalized standard deviation of the streamwise component of vorticity, $\omega_x^{rms}\nu/u_\tau^2$.

The profiles of the standard deviation of the three components of vorticity are shown in figures 4.13-4.15. As with the previously analysed quantities, all three simulations adequately reproduce the main features found in the DNS data. However, the accuracy of the predicted values is not equally good. Even the results from M3 quite heavily under-predict the values for all three components of the fluctuations, and the results from M1 differ from the DNS data as much as by a factor of 5 at certain points.

A possible explanation for the loss of accuracy in the results, as compared to the previously discussed accuracy of the predictions of the velocity fluctuations, is discussed in [9]. The authors note that the relative contribution of the small scales to vorticity fluctuations is significantly higher than to the velocity fluctuations. Applied to LES, this means that the fact that we do not resolve the subgrid scales introduces a large error to the calculation of vorticity fluctuations as opposed to velocity fluctuations, where the introduced error is less significant. In the same paper the authors reason that all three components of $\boldsymbol{\omega}^{rms}$ have similar magnitude away from the channel walls due to the fact that the small scales tend to be isotropic in that region. This can also be used as an explanation for why the errors in the obtained results are relatively small in the core of the channel, as the subgrid scales are easier to model there.

In [9, 7], the authors also address the existence of the local minimum in ω_x^{rms} near the wall, which is immediately followed by a local maximum. The given explanation for this behaviour is that there exists a near-wall, streamwise vortical structure, that has its center (in average) located at the local maximum of ω_x^{rms}, and its edge at the local minimum.

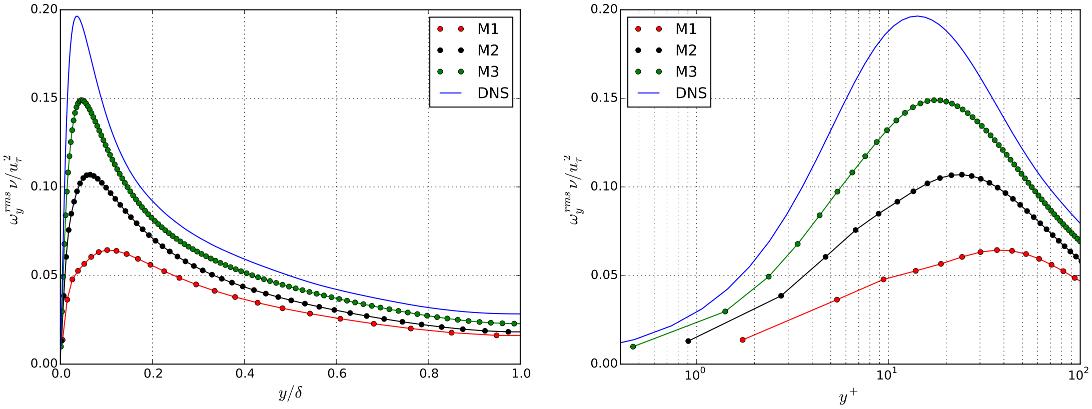

Figure 4.14: Profiles of the normalized standard deviation of the wall-normal component of vorticity, $\omega_y^{rms}\nu/u_\tau^2$.

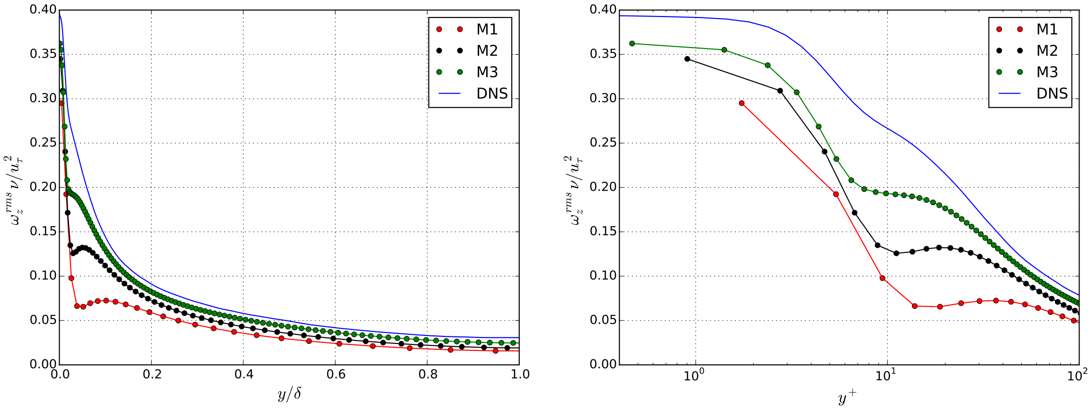

Figure 4.15: Profiles of the normalized standard deviation of the spanwise component of vorticity, $\omega_z^{rms}\nu/u_\tau^2$.

4.7 Two-point velocity correlations

We conclude the analysis by considering the spatial two-point autocorrelations of the velocity components R_{u_i}. The significance of these quantities lie in the fact that they can be used to compute integral length scales of the turbulent structures present in the flow, and their connection to the velocity spectrum. Additionally, R_{u_i} can be used to assert the adequacy of the chosen size of the computational domain. The domain is large enough, if the two-point correlations become negligibly small on the length scale of the domain, in the respective directions.

In figures 4.16-4.21, autocorrelations in both the streamwise and the spanwise direction are shown. Calculation of the autocorrelations was done at the following y^+-values: 10, 40, 150 and 392. Therefore, each figure contains four graphs, corresponding to the different y^+-values. Note that here the theoretical value of u_τ corresponding to, $\mathrm{Re}_\tau = 395$, was used to compute the values of y^+.

Table 4.2 contains the values of the integral length scales, L_x and L_z, calculated from the profiles displayed in the figures.

		$\boldsymbol{L_x}$				$\boldsymbol{L_z}$			
	$\boldsymbol{y^+}$	**M1**	**M2**	**M3**	**DNS**	**M1**	**M2**	**M3**	**DNS**
$\boldsymbol{uu}$	10	1.91	0.97	0.67	0.57	0.08	0.03	0.01	0.02
	40	1.75	0.81	0.71	0.66	0.07	0.04	0.02	0.04
	150	1.11	0.63	0.74	0.80	0.09	0.06	0.05	0.05
	392	0.84	0.62	0.66	0.49	0.19	0.16	0.14	0.14
$\boldsymbol{vv}$	10	0.84	0.34	0.19	0.16	0.03	0.01	0.01	0.01
	40	0.88	0.34	0.22	0.18	0.04	0.01	0.01	0.02
	150	0.48	0.18	0.17	0.15	0.07	0.07	0.08	0.07
	392	0.24	0.13	0.08	0.09	0.19	0.19	0.20	0.19
$\boldsymbol{ww}$	10	0.75	0.36	0.27	0.22	0.06	0.04	0.04	0.04
	40	0.53	0.24	0.14	0.13	0.12	0.10	0.08	0.09
	150	0.27	0.11	0.08	0.08	0.24	0.15	0.16	0.13
	392	0.26	0.18	0.10	0.14	0.33	0.25	0.22	0.24

Table 4.2: Integral length scales L_x and L_z.

The figures indicate that the chosen size of the domain is large enough to fit all the relevant turbulent structures. The curves from M3 all decay towards values very close to zero, with the exception of $R_u(x)$ at y^+ equal to 40, 150 and 392. The latter results are in relatively good agreement with the DNS-curves however.

All of the plots in figures 4.16-4.21 indicate that M1 over-predicts the two-point correlations and therefore the integral length scales. This is especially clear in the plots of $R_u(x)$, at y^+ equal to 10 and 40, where the value of $R_u(x)$ is close to 0.4 at the domain's boundary. The M2-results are, in general, in very good agreement with DNS, and are often only marginally less accurate than those obtained from M3. The only exception is $R_u(x)$, at $y^+ = 10$, where the value at the boundary is over-predicted quite significantly.

From table 4.2, it is evident that, near the wall, the turbulent eddies are elongated along the streamwise direction. This is indicated by the fact that the integral length scales L_z are significantly smaller than the scales L_x. This is a well-documented result, and a thorough discussion can be found in [9]. Conversely, in the core of the channel L_x and L_z are of the same order of magnitude.

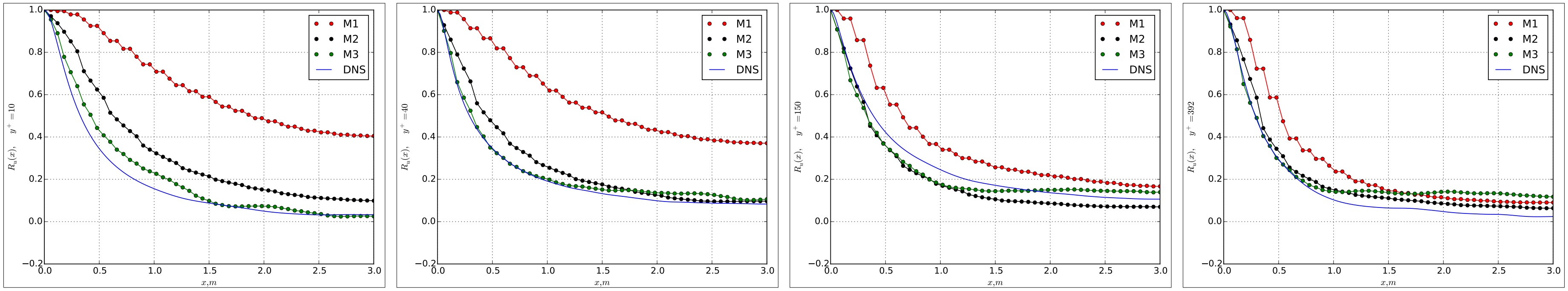

Figure 4.16: Profiles of the spatial autocorrelations of the x component of velocity along x. From left to right: $y^+ = 10$, $y^+ = 40$, $y^+ = 150$, $y^+ = 392$.

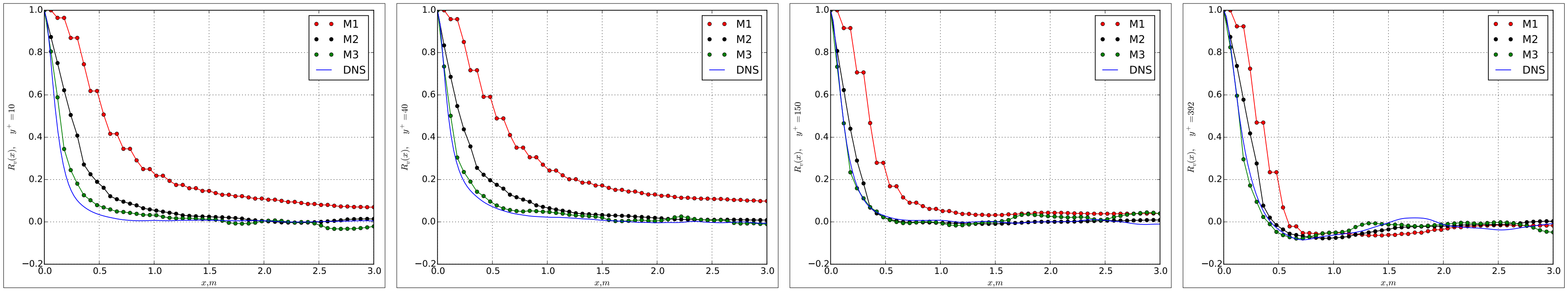

Figure 4.17: Profiles of the spatial autocorrelations of the y component of velocity along x. From left to right: $y^+ = 10$, $y^+ = 40$, $y^+ = 150$, $y^+ = 392$.

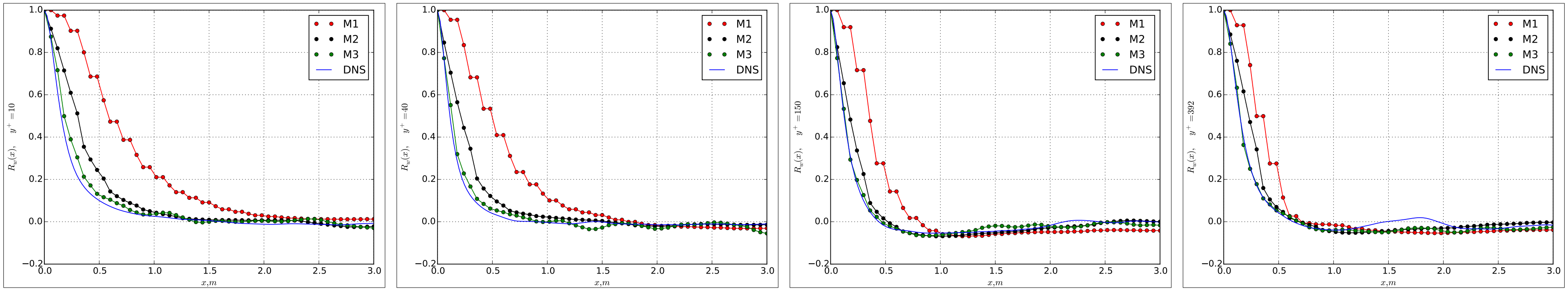

Figure 4.18: Profiles of the spatial autocorrelations of the z component of velocity along x. From left to right: $y^+ = 10$, $y^+ = 40$, $y^+ = 150$, $y^+ = 392$.

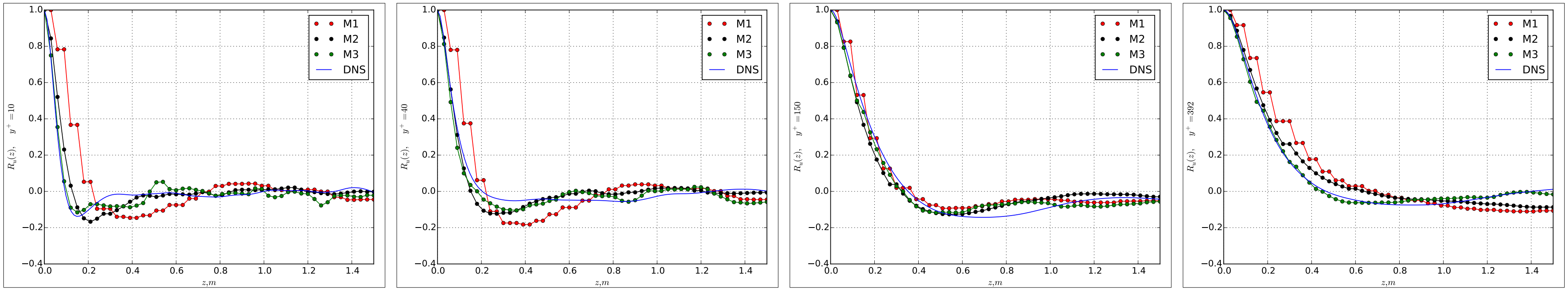

Figure 4.19: Profiles of the spatial autocorrelations of the x component of velocity along z. From left to right: $y^+ = 10$, $y^+ = 40$, $y^+ = 150$, $y^+ = 392$.

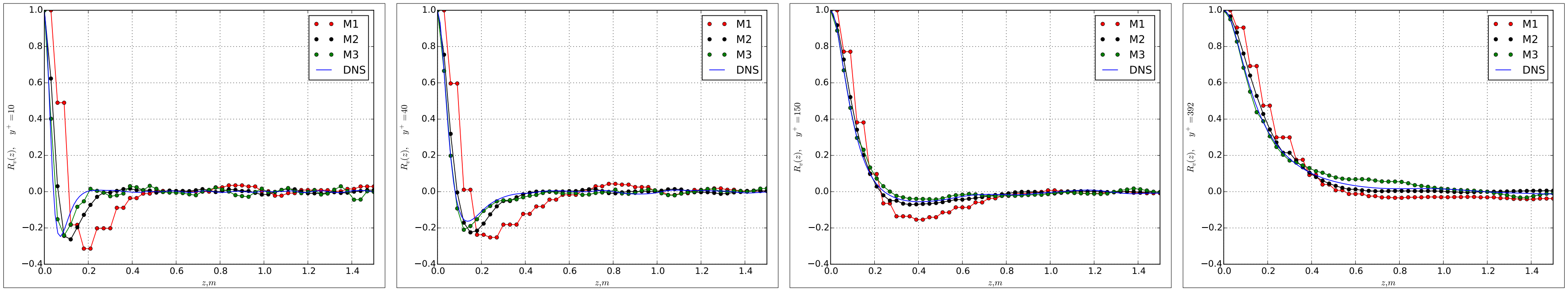

Figure 4.20: Profiles of the spatial autocorrelations of the y component of velocity along z. From left to right: $y^+ = 10$, $y^+ = 40$, $y^+ = 150$, $y^+ = 392$.

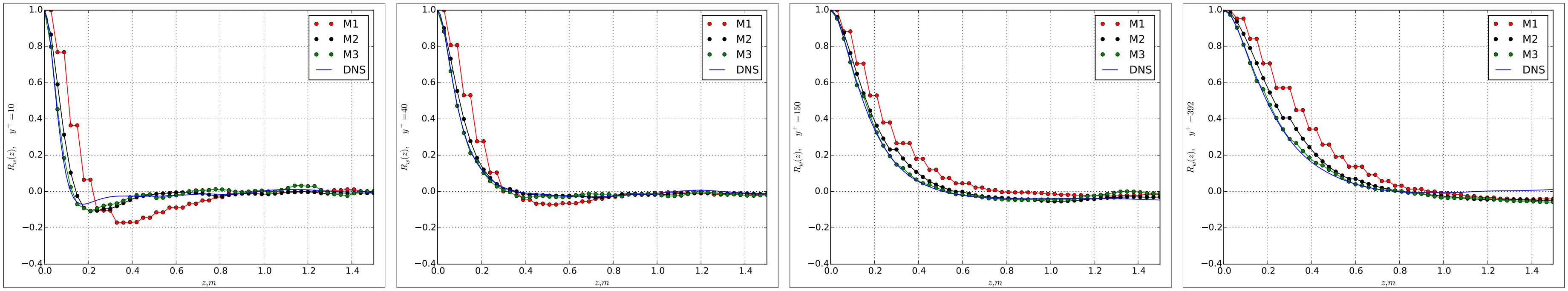

Figure 4.21: Profiles of the spatial autocorrelations of the y component of velocity along z. From left to right: $y^+ = 10$, $y^+ = 40$, $y^+ = 150$, $y^+ = 392$.

Chapter 5

Conclusions

This report presents results and analysis from a simulation campaign, using LES and `OpenFOAM`, of channel flow at $\mathrm{Re}_b = 13\,350$ (which corresponds to, $\mathrm{Re}_\tau \approx 395$). One particular subgrid model has been used, see section 2.3, and simulations have been carried out on a sequence of three successively refined computational grids. The coarsest grid contains 135 000 cells and the finest grid $8.65{\cdot}10^6$ cells. The primary purpose of the study is to evaluate the predictive capability of the numerical methods and turbulence modelling employed. Channel flow has been chosen since it is a very well documented test case with wall-bounded turbulence, and we use DNS-data, [10], to validate our LES results.

The results include profiles for an extensive list of statistical quantities, up to fourth order moments of the velocity components, see chapter 4. For all profiles, we include LES-results obtained on all three mesh refinement levels, as well as DNS-results. The analysis is focused on assessing the predictive accuracy for the different statistical quantities, and how it improves with mesh refinement. A general conclusion, clearly supported by the results, is that the coarsest grid provides a reasonably good prediction of the mean velocity profile, whereas for higher order statistical moments, the gain of using finer grids is evident. Furthermore, the convergence of the results, with grid refinement, is essentially regular. In connection with each set of results, these statements are quantified, and the specifics of the discrepancies are discussed in detail. Two important observations are high-lighted here. (i) For most quantities, the extrema is located in the vicinity of the wall. Using a coarse grid shifts the location of the extrema further away from the wall. (ii) For vorticity, which depends more markedly on the small scales, the prediction of the level of fluctuations (second order moments of components) is less accurate than the corresponding statistical measure for the velocity.

Two-point autocorrelations of the velocity components are also included, in addition to the profiles of statistical moments just discussed. The autocorrelations have been computed at different wall-normal locations, in the streamwise and spanwise directions, for all velocity components and for all grid refinement levels. The integral length scales based on the autocorrelations have also been computed, and are compared with DNS-results. Grid refinement level has a significant effect on the autocorrelations, the integral length scales being much over-predicted on the coarsest grid. The autocorrelations are also used to assess the choice of spatial extent of the computational domain in the streamwise and spanwise directions (in which periodic boundary conditions are employed). The conclusion is that these lengths are sufficiently large, not to negatively affect the results. At least for the finest grid, for which the prediction of the autocorrelations is acceptable. Although in the streamwise direction, the margin is not very large. The largest streamwise integral length scale is, $L_x \approx 0.8$ m, at $y^+ = 150$, and the domain streamwise length is $l_x = 6.0$ m.

The report also includes an overview of the turbulence modelling and the particular subgrid model used. The finite volume method is shortly reviewed and an outline is given of the overall algorithm implemented in the `OpenFOAM` solver program. All of the data presented in the report is made available online[1] along with complete `OpenFOAM` simulation case set-up files. In conclusion, we would like to emphasize two features of this study. (i) It presents thoroughly validated results for an important canonical flow with wall-bounded turbulence, obtained with a software which can be used for general purpose CFD, with complicated geometries. (ii) The simulation cases are made publicly available and the results can be easily reproduced with this open-source software. For these reasons, in particular, the authors hope that the study will be useful for researchers and CFD-users of `OpenFOAM`, and similar software.

[1] `https://bitbucket.org/lesituu/channel_flow_data`

Bibliography

[1] Ferziger, J.H., Peric M. *Computational Methods for Fluid Dynamics*, 3rd edition, Springer, 2002.

[2] Horiuti, K. *Large Eddy Simulation of Turbulent Channel Flow by One Equation Modeling*, J. Phys. Soc. Japan, 54(8), 2855-2865, 1985.

[3] Iwamoto, K., Kasagi, N., Suzuki, Y. *Direct Numerical Simulation of Turbulent Channel Flow at Re_τ=2320*, Proc. 6th Symp. Smart Control of Turbulence, Tokyo, March 6-9, 2005.

[4] Jasak, H. *Error Analysis Estimation for the Finite Volume Method with Applications to Fluid Flows*, Ph.D. thesis, The Imperial College of Science, Technology and Medicine, 1996.

[5] Jimenez, J., Hoyas, S., Simens, M.P., Mizuno, Y. *Turbulent Boundary Layers and Channels at Moderate Reynolds Numbers*, J. Fluid Mech., 657, 335-360, 2010.

[6] Kim, W.W., Menon, S. *An Unsteady Incompressible Navier-Stokes Solver for Large-Eddy Simulation of Turbulent Flows*, Int. J. Numer. Meth. Fluids, 31, 983-1017, 1999.

[7] Kim, J., Moin, P., Moser, R. *Turbulence Statistics in Fully Developed Channel Flow at Low Reynolds Number*, J. Fluid Mech., 50, 133-160, 1987.

[8] Kundu, P.K., Cohen, I.M., Ayyaswamy, P.S. *Fluid Mechanics*, 4th edition, Academic press, 2007.

[9] Moin, P. Kim, J. *Numerical Investigation of Turbulent Channel Flow*, J. Fluid Mech., 118, 341-377, 1982.

[10] Moser, R., Kim, J., Mansour, N. *Direct Numerical Simulation of Turbulent Channel Flow Up to* $Re_\tau = 590$, Physics of Fluids, 11(4), 943-945, 1999.

[11] Pope, S.B. *Turbulent Flows*, Cambridge University Press, 2000.

[12] Rusche, H. *Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions*, Ph.D. thesis, The Imperial College of Science, Technology and Medicine, 2002.

[13] Sagaut, P. *Large Eddy Simulation for Incompressible Flows*, 3rd edition, Springer, 2006.

[14] Schumann, U. *Subgrid Scale Model for Finite Difference Simulations of Turbulent Flows in Plane Channels and Annuli*, J. Comput. Phys., 18, 376-404, 1975.

[15] Stevens, B., Moeng, C.H., Sullivan, P. *Large-Eddy Simulations of Radiatively Driven Convection: Sensitivities to the Representation of Small Scales*, J. Atmos. Sci., 56, 3963-3984, 1999.

[16] Versteeg, H.K., Malalasekera W. *An Introduction to Computational Fluid Dynamics, The Finite Volume Method*, 2nd edition, Prentice Hall, 2007.

[17] De Villiers, E. *The Potential of Large Eddy Simulation for the Modeling of Wall Bounded Flows*, Ph.D. thesis, The Imperial College of Science, Technology and Medicine, 2006.

[18] Yoshizawa, A. *A Statistically-Derived Subgrid Model for the Large-Eddy Simulation of Turbulence*, Phys. Fluids A, 3(8), 2007-2009, 1991.

[19] Yoshizawa, A., Horiuti, K. *A Statistically-Derived Subgrid-Scale Kinetic Energy Model for the Large-Eddy Simulation of Turbulent Flows*, J. Phys. Soc. Japan, 54(8), 2834-2839, 1985.

Appendix A

Definitions and methods of computation for statistical quantities

All definitions of the statistical quantities have been collected in this appendix. The use of time-series, and the symmetry properties of channel flow, in the method of computation is also described.

A.1 Averaging

For any quantity $\phi(\mathbf{x}, t)$, we use the following notation for the decomposition into its mean component, and fluctuations around the mean.

$$\phi(\mathbf{x}, t) = \langle \phi(\mathbf{x}, t) \rangle + \phi'(\mathbf{x}, t) \tag{A.1}$$

Observe that, in this report, ϕ typically is a (space-)filtered flow quantity. Due to the symmetry of channel flow, and the fact that we are interested in the statistically steady state, the mean (and other statistics) only depend on the wall-normal coordinate, $\langle \phi(\mathbf{x}, t) \rangle = \langle \phi(y) \rangle$. Finally, we remark that we discuss the mean both in the sense of an ensemble average, [11], and an average accumulated by time- and space-integration, as described above.

A.1.1 Temporal averaging

In practice, based on time-resolved simulations, the averages for statistically stationary quantities are typically calculated by averaging of time series. For this purpose, we introduce the temporal average, of some quantity $\phi(\mathbf{x}, t)$, over the time interval $[t_1, t_2]$, as,

$$\langle \phi(\mathbf{x}, t) \rangle_{[t_1, t_2]} = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} \phi(\mathbf{x}, \tau) \, \mathrm{d}\tau. \tag{A.2}$$

We have the following limit (for statistically stationary quantities).

$$\lim_{T\to\infty} \langle \phi(\mathbf{x},t)\rangle_{[t,t+T]} = \langle \phi(\mathbf{x},t)\rangle$$

In fact, this does not hold true for all turbulent flows, but it is certainly the case for channel flow. The length of the time interval is crucial for the quality of the computed averages. The time interval used in the simulations presented in this report are given in table 3.4. The (time) convergence of the statistics can be considered to be extremely good for meshes M1 and M2, and as very good for the finer grid M3. The simulations using M3 are naturally associated with a significantly higher computational cost.

In the case with discrete time, as in the simulations, the temporal average is approximated by a sum according to,

$$\langle \phi(\mathbf{x},t)\rangle_{[t_1,t_2]} \approx \langle \phi(\mathbf{x},t)\rangle_N = \frac{1}{N}\sum_{n=0}^{N-1} \phi(\mathbf{x},t_n), \tag{A.3}$$

where, $t_n = t_1 + n\Delta t$, and, $\Delta t = (t_2 - t_1)/(N-1)$.

A direct, "brute force", evaluation of the expression (A.3) would require the storage of 3D data at a large number of time steps. This would lead to excessive requirements on data storage. Instead, the average is accumulated at each time step of the simulation, by updating the current estimate of the mean in the following way.

$$\langle \phi(\mathbf{x},t)\rangle_{n+1} = \frac{\phi(\mathbf{x},t_{n+1}) + n\langle \phi(\mathbf{x},t)\rangle_n}{n+1} \tag{A.4}$$

In this way the storage requirement problem is resolved.

A.1.2 Spatial averaging

Spatial averaging can be used, in addition to temporal averaging, for channel flow, due to its symmetry (statistics only depending on the y-coordinate). For this purpose, we introduce the spatial average, over a cross-section parallel to the walls, as,

$$\langle \phi(\mathbf{x},t)\rangle_{X\times Z} = \frac{1}{l_x l_z}\int_0^{l_x}\int_0^{l_z} \phi(\mathbf{x},t)\,\mathrm{d}x\mathrm{d}z.$$

For the computation of the average, we thus use both temporal and spatial averaging,

$$\langle \phi(\mathbf{x},t)\rangle = \langle \phi(y)\rangle \approx \langle \phi(\mathbf{x},t)\rangle_{[t_1,t_2]\times X\times Z}.$$

In the practical computation, the temporal average is first accumulated during the simulation, as described above, then the spatial averaging is carried out during the post-processing stage. The fact that channel flow allows for the procedure of spatial averaging leads to great savings in simulation time since the convergence of the statistics is very much accelerated.

A.2 Higher order statistical moments

The definition and method of computation of the standard deviation, the flatness and the skewness are described in this section. These are based on the second, third and forth (central) statistical moments respectively which are defined by,

$$\mu_2(\phi) = \langle \phi'^2 \rangle = \langle (\phi - \langle \phi \rangle)^2 \rangle = \langle \phi^2 \rangle - \langle \phi \rangle^2, \tag{A.5}$$

$$\mu_3(\phi) = \langle \phi'^3 \rangle = \langle \phi^3 \rangle - 3\langle \phi \rangle \langle \phi^2 \rangle + 2\langle \phi \rangle^3, \tag{A.6}$$

$$\mu_4(\phi) = \langle \phi'^4 \rangle = \langle \phi^4 \rangle - 4\langle \phi \rangle \langle \phi^3 \rangle + 6\langle \phi \rangle^2 \langle \phi^2 \rangle - 3\langle \phi \rangle^4. \tag{A.7}$$

The above expressions are used in the calculation, which then also must include the averaging of powers of ϕ, which are calculated as described above, using both temporal and spatial averaging. The standard deviation is defined as the square root of $\mu_2(\phi)$ and skewness and flatness are the following normalizations of $\mu_3(\phi)$ and $\mu_4(\phi)$.

$$S_\phi = \frac{\mu_3(\phi)}{\langle \phi'^2 \rangle^{3/2}}, \tag{A.8}$$

$$F_\phi = \frac{\mu_4(\phi)}{\langle \phi'^2 \rangle^2} \tag{A.9}$$

The calculation of covariances of two quantities ϕ and ψ (e.g. non-diagonal components of the Reynolds stress tensor) can be seen as a generalization of (A.5),

$$\text{cov}(\phi, \psi) = \langle \phi' \psi' \rangle = \langle (\phi - \langle \phi \rangle)(\psi - \langle \psi \rangle) \rangle = \langle \phi \psi \rangle - \langle \phi \rangle \langle \psi \rangle. \tag{A.10}$$

A.3 Two-point correlations

The definition and method of computation of the two-point correlation of two quantities, ϕ and ψ, are described in this section. The definition is as follows.

$$R_{\phi\psi}(\mathbf{x}, \mathbf{r}) = \frac{\langle \phi'(\mathbf{x}, t) \psi'(\mathbf{x} + \mathbf{r}, t) \rangle}{\langle \phi'^2 \rangle^{1/2} \langle \psi'^2 \rangle^{1/2}} \tag{A.11}$$

For the special case, $\phi = \psi$, the function, $R_\phi = R_{\phi\phi}$, is referred to as the spatial autocorrelation of ϕ. Due to the symmetry of channel flow, we have,

$$R_{\phi\psi}(\mathbf{x}, \mathbf{r}) = R_{\phi\psi}(y, \mathbf{r}).$$

The calculation of the correlations rely on simultaneous values of the functions in different points. In order to obtain the values, we have introduced a number of points, referred to as probes, in which the complete time history of the flow variables is stored. The correlations are then calculated at the post-processing stage which means that the probe placements limits the possibilities for evaluating the correlations. We have introduced lines of probes parallel to the x- and z-axis respectively, and located at four different y-coordinates, see section 4.7. Each of the eight lines contains 51 probes, evenly distributed through half of the computational domain.

Based on this data, we compute $R_\phi(y_i, x\mathbf{e}_x)$ and $R_\phi(y_i, z\mathbf{e}_z)$. Here y_i is one of the four selected y-vales for probe lines, ϕ is one of the velocity components, $\mathbf{e}_x$ is the unit vector in the x-direction, and $\mathbf{e}_z$ is the unit vector in the z-direction.

The spatial autocorrelation can be used to define an integral length scale in a direction $\mathbf{e}_r$. The definition is as follows.

$$L_\phi(\mathbf{x}) = \frac{1}{R_\phi(0)} \int_0^\infty R_\phi(\mathbf{x}, r\mathbf{e}_r)\, \mathrm{d}r. \tag{A.12}$$

For the evaluation of these integrals, we applied the Simpson's rule.

Recent technical reports from the Department of Information Technology

2015-012 Carlo Garoni, Carla Manni, Stefano Serra-Capizzano, Debora Sesana, and Hendrik Speleers: *Lusin Theorem, GLT Sequences and Matrix Computations: An Application to the Spectral Analysis of PDE Discretization Matrices*

2015-011 Ali Dorostkar, Maya Neytcheva, and Stefano Serra-Capizzano: *Schur Complement Matrix and its (Elementwise) Approximation: A Spectral Analysis Based on GLT Sequences*

2015-010 Nanna Kjellin Lagerqvist: *Resultat och reflektioner kring mailkategorisering av användares mail till Uppsala läns landsting kring åtkomst av journaler via nätet*

2015-009 Emilie Blanc: *Approximation of the Diffusive Representation by Decreasing Exponential Functions*

2015-008 Ali Dorostkar, Maya Neytcheva, and Stefano Serra-Capizzano: *Spectral Analysis of Coupled PDEs and of their Schur Complements via the Notion of Generalized Locally Toeplitz Sequences*

2015-007 Aletta Nylén and Mohamed Shenify: *Proceedings from the 1st Albaha University–Uppsala University Collaborative Symposium on Quality in Computing Education*

2015-006 Farshid Besharati, Mahdad Davari, Christian Danheimer Furedal, Björn Forsberg, Niklas Forsmark, Henrik Grandin, Jimmy Gunnarsson, Engla Ling, Marcus Lofvars, Sven Lundgren, Luis Mauricio, Erik Norgren, Magnus Norgren, Johan Risch, Christos Sakalis, and Stefanos Kaxiras: *The EVI Distributed Shared Memory System*

2015-005 Carlo Garoni, Carla Manni, Stefano Serra-Capizzano, Debora Sesana, and Hendrik Speleers: *Spectral Analysis and Spectral Symbol of Matrices in Isogeometric Galerkin Methods*

2015-004 Sofia Cassel, Falk Howar, Bengt Jonsson, and Bernhard Steffen: *Learning Extended Finite State Machines (extended version)*

2015-003 Ali Dorostkar, Maya Neytcheva, and Björn Lund: *On Some Block-Preconditioners for Saddle Point Systems and their CPU-GPU Performance*

2015-002 Marco Donatelli, Mariarosa Mazza, and Stefano Serra-Capizzano: *Spectral Analysis and Structure Preserving Preconditioners for Fractional Diffusion Equations*

2015-001 Victor Shcherbakov and Elisabeth Larsson: *Radial Basis Function Partition of Unity Methods for Pricing Vanilla Basket Options*

UPPSALA
UNIVERSITET

May 2015
ISSN 1404-3203
http://www.it.uu.se/