

Turbulent channel flow tests

July 19, 2018

List of Figures

1	Cuboid representing a channel	9
2	3D rectangular duct	13
3	Velocity profile in the fully developed channel flow [1]	15
4	Nondimensional velocity u^+ vs. the nondimensional distance y^+ (plotted in logarithmic scale) [2]	17
5	Flow between the infinite parallel plates	19
6	Parabolic flow profile with $U_b = 0.1335 \text{ m/s}$	22
7	Mean stream-wise velocity profile normalized by u_τ	26
8	Root-mean-square velocity fluctuations normalized by u_τ in global coordinates	27
9	Rms profiles for the streamwise velocity fluctuation normalised by u_τ	28
10	Rms profiles for the wall-normal velocity fluctuation normalised by u_τ	28
11	Rms profiles for the spanwise velocity fluctuation normalised by u_τ	29
12	Turbulent shear stress normalised by u_τ^2	29
13	Mean velocity profile normalised with u_τ	30
14	Root-mean-square velocity fluctuations normalized by u_τ in global coordinates	30
15	Rms profiles for the streamwise velocity fluctuation normalised by u_τ	31
16	Rms profiles for the wall-normal velocity fluctuation normalised by u_τ	31
17	Rms profiles for the spanwise velocity fluctuation normalised by u_τ	32
18	Turbulent shear stress profiles normalised by u_τ	32
19	Mean velocity profiles normalised by u_τ	33
20	Mean velocity profiles normalised by u_τ	33
21	Mean velocity profiles	34
22	Rms profiles for the streamwise velocity fluctuation normalised by u_τ	34
23	Rms profiles for the wall-normal velocity fluctuation normalised by u_τ	35
24	Rms profiles for the spanwise velocity fluctuation normalised by u_τ	35
25	Turbulent shear stress profiles normalised by u_τ^2	36
26	Mean velocity profiles, $Re_\tau = 550$	37

List of Tables

1	Mesh resolution used in TGV simulations	11
2	Solver settings for TGV	12
3	The size of the domain in x, y, z directions	19
4	Physical quantities	20
5	Mesh statistics	21
6	Mach number	24
7	Solver settings	24
8	Comparison of the resulting physical quantities	25
9	Comparison of the computed physical quantities using WALE model	29
10	Physical quantities, $Re_\tau = 550$	37
11	Solver settings, $Re_\tau = 550$	37
12	Comparison of the computed physical quantities using WALE model & UDNS $Re_\tau = 550$	37
13	Parameters in lattice units	40

Nomenclature

Latin symbols

\bar{u}	filtered stream-wise velocity
\bar{v}	filtered wall-normal velocity
\bar{w}	filtered span-wise velocity
U	Mean stream-wise velocity
u	stream-wise velocity fluctuation
V	Mean wall-normal velocity
v	wall-normal velocity fluctuation
W	Mean span-wise velocity
w	span-wise velocity fluctuation
L_x	Length of the domain, stream-wise direction
L_y	Height of the domain, wall-normal direction
L_z	Width of the domain, span-wise direction

Greek symbols

\mathbb{H}	Quaternions
\mathbb{C}	Complex Numbers
\mathbb{R}	Real Numbers
δ	Half-channel height, wall-normal direction

Brackets, Superscripts, Sub-scripts

ρ	Friction Index
V	Constant Volume

Contents

List of Figures	i
List of Tables	ii
Nomenclature	iii
1 Introduction	2
1.1 Literature survey	4
1.2 Objectives and Motivation	4
2 Lattice Boltzmann theories	5
2.1 BGK	5
2.2 MRT	5
2.3 Cumulant LBM	5
3 Turbulence theories	6
3.1 RANS	6
3.2 LES	7
4 Implementation	8
4.1 Discretisation	8
4.2 Implementation of eddy viscosity contribution	8
4.2.1 Conditioning of the WALE model equation	9
4.3 Taylor-Green vortex simulations (TGV)	10
4.3.1 Simulation settings	10
4.3.2 Analytical settings	11
4.3.3 Results & comparison	11
5 Model set-up and theories	12
5.1 Turbulent channel flow	12
5.1.1 Governing equations	14
5.1.2 Scaling units	15
5.2 Reynolds number	18
5.3 Computational Domain and Meshing	18
5.3.1 Computational Domain	18
5.3.2 Physical parameters	19
5.3.3 Meshing	20
5.4 Initial conditions	21
5.5 Modelling of pressure gradient	22
5.6 Solver settings	23
5.7 Computational details & Post-processing	24
6 Results and discussion	25
6.1 UDNS results	25
6.1.1 Mean velocity profile	26
6.1.2 Turbulence intensities	26
6.1.3 Turbulent shear stress	27
6.2 WALE model results	28
6.2.1 Mean velocity profiles	28

6.2.2	Turbulence intensities	30
6.2.3	Turbulent shear stress	31
6.3	Comparison UDNS and WALE, $Re_\tau = 395$	32
6.3.1	Mean velocity profile	32
6.3.2	Turbulence intensities	34
6.3.3	Turbulent shear stress	36
6.4	Comparison UDNS and WALE, $Re_\tau = 550$	36
6.5	Conclusion and Outlook	38
A	Appendices	39
A.1	Initial conditions	39
A.2	Lattice units	39
A.3	One WALE	40
	References	41

1 Introduction

Navier-stokes equations (NS) are the governing equation for the fluid flow. The flow of fluid can be termed as laminar or turbulent depending on the dimensionless Reynolds (Re) number. Reynolds number of a flow gives the measure of the relative importance of the inertia force over the viscous force. When the viscous forces are dominating i.e. the Re no is small the flow is laminar. Any disturbance present or originating in the laminar flow would be damped out by the viscosity. Above a certain value of Re number the flow becomes turbulent i.e. inertia forces are quite significant and any disturbance or irregularity would be amplified resulting in the flow becoming random and chaotic with the velocity and the pressure changing continuously in time and with in the substantial region of the flow i.e. turbulent regime.

Turbulent flows comprise of the rotating flow structures called turbulent eddies. These structures possess a wide range of length scales. The largest eddies interact with the mean flow and derive the energy from it. These large eddies are anisotropic and are affected by the inertia forces. They are insensitive to the viscosity. These eddies then transfer the energy further to the smaller and the smaller eddies via a mechanism called energy cascade. The energy present in the smaller eddies is dissipated by working against the viscous forces.

Analytical solution of the flow governing equations are only available for certain basic cases viz. Couette flow or Poiseuille flow. The solution for the flow involving turbulence or with complex boundary conditions is not available. Thus the equations have to be solved numerically i.e. numerical simulations. The most straightforward approach for the solution of the turbulent flows is to perform Direct numerical simulation (DNS), where the flow governing equations are discretized and solved numerically. This approach resolves all the existing scales in the flow which eventually results in a computationally expensive simulation. With the latest advancement in the computational resources this approach is applicable only to flows with smaller to moderate Reynolds number (see section). The remedy to this problem is not to resolve all scales of motion present in the flow. This alternative approach is turbulence modeling. In this approach the entire scales of motion present in the flow are modeled, Reynolds averaged Navier stokes equation (RANS), or a certain level of scales are chosen to be resolved and the rest are modeled, Large eddy simulation (LES).

A sophisticated approach to obtain the solution of the turbulent flows is to use the Reynolds decomposition, where the flow properties are decomposed in to their respective mean and fluctuating parts, in combination with the flow governing equations to obtain the RANS equations. RANS equations describe the evolution of the mean flow properties. An additional term is obtained in RANS equations which represents the effects of the turbulent fluctuations i.e. fluctuating velocity components. This additional term has to be modeled to close the set of equations.

In LES large scales are resolved and the smaller scales are modeled. The separation of the scales is achieved by a filtering process (see section). Using the filtering operation, where the flow properties are decomposed in to resolved/filtered and unresolved/residual part, in combination with the NS equations we get the LES equations. These equations describe the evolution of the filtered quantities. Additional term is obtained in the LES equations which represents the effect of the residual scales on to the filtered scales. This additional

term has to be modeled to close the set of equations.

In terms of the computational expense LES lies in between the solution of turbulent flows by DNS and RANS. This method is motivated by the limitations of the RANS and the DNS approach [3]. LES involves modeling of the smaller scales of motion and does not resolve the entire range of scales as done in DNS. This allows this method to be applicable to flows with high Reynolds number. In comparison to RANS, where the entire range of scales is modeled, LES models only the smaller scales and thus it is computationally expensive. This extra computational overhead is justified when an accurate and reliable prediction of the flow is obtained. Specifically in the flows where the unsteadiness due to the larger scales are important i.e. flow over the bluff bodies LES is the appropriate choice. With the recent developments in the computational resources a lot of research is going on to make LES applicable to complex geometries and industrial flows.

Why LBM?

A wide variety of methods are available to find the solution to the fluid flow equations. These methods differentiate from each other on the basis of the level of description of the fluid i.e microscopic, mesoscopic or macroscopic. The methods are grouped under two terminologies: Conventional methods and the particle based methods [4]. The former methods assumes the macroscopic description or continuum approach of the fluid viz. finite difference method, finite volume method. The latter assumes the microscopic or mesoscopic description viz. Molecular dynamics, LBM etc. In this work, LBM has been chosen to perform the LES of the wall-bounded turbulent flow. This method differentiates from the conventional methods in the sense that it does not solve the flow equations directly, but it solves the boltzmann equation. Well known methods exist that can relate the dynamics of this equation to the macroscopic conservation equations [4].

Because of its kinetic nature the LBM enjoys several advantages over other numerical methods and that makes it a lucrative option for the simulation of the fluid flow [?]. *S. Chen et. al* [?] points out some major advantages of LBM over other methods:

- The convection operator (streaming process) of LBM is linear. This behavior is in contrast to the non-linear behavior of the convection term in the conventional methods that use the macroscopic description. Streaming process in combination with the relaxation process (collision operator) is able to recover the non-linear convection behavior present in the macroscopic equations.
- The pressure of the LBM is computed from the equation of state. On the other hand, a poisson equation has to be solved for the pressure in the conventional methods which is quite tedious to discretize and solve numerically.
- The LBM discretises the Boltzmann equation in velocity space and only retains a minimal set of particle velocities that are needed to recover the proper macroscopic behaviour. This makes the LBM considerably more computationally efficient than other particle-based methods [?].
- In a grid, the collision process is local to each node and the interaction between the nodes is linear. It is this property of LBM which makes it

very amenable to high performance computing on parallel architectures including GPU's [4].

All methods have their advantages and disadvantages and there is no one method that is superior to other methods. LBM has also disadvantages: it memory intensive [4], not efficient for steady flow simulations [?], it is limited to incompressible problems [?] etc. It is a relatively young method and is evolving at a fast pace, which means that the applicability of the method to variety of problem is still increasing [4].

1.1 Literature survey

1.2 Objectives and Motivation

2 Lattice Boltzmann theories

2.1 BGK

2.2 MRT

2.3 Cumulant LBM

3 Turbulence theories

i DNS.

Direct numerical simulation (DNS) method uses a modelling-free approach for approximation of the exact solution, as the navier-stokes equation comprises of all turbulence mechanisms. Since the turbulence is three dimensional and inherently instationary it has to take in to account all the components present in the navier stokes equation. Turbulence involves the interaction in between the scales of various different sizes and thus to represent the exact solution or the characteristics of the turbulent flow considered correctly, the DNS method emphasises on the fine spatial and temporal resolution of the meshes. Apart from that this method requires the discretization methods to have low level of numerical dissipation and dispersion. In order to correctly determine the effects of the turbulence intensity, the fluctuating flow variables which satisfy the navier stokes equation are provided as inputs.

ii DES.

Detached eddy simulation, DES, combines the approach of RANS and the LES method; it therefore belongs to the class of the hybrid RANS-LES simulations. The basic idea of this method is that the near wall boundary layer would be resolved using the RANS approach, while the usage of LES to (correctly) resolve such regions requires a very fine spatial and temporal resolution, as the turbulent scales to be resolved in the boundary layers are very small. On the other hand, the detached regions, wakes and the free shear layers are the regions, where the RANS model frequently fails, are handled by the LES.

DES was first introduced by *Spalart et al.*. The goal was to compute the aerodynamic flows involving massive separation with a moderate increase in the grid resolution compared to the RANS method. The idea behind the DES is to segregate the flow, using suitable detectors, in to the domains where the entire energy spectrum will be modelled using the RANS (near wall boundary layer) or the large energy containing structures are resolved using the LES method (detached flow region). Therefore in this approach, a suitable method is chosen, for each domain, in terms of efficiency and accuracy.

In comparison with LES, for the computation of the flows involving the boundary layer, DES reduces the computational effort significantly. Since the flows involved are unsteady and three dimensional turbulent flows, the computational effort will be relatively higher when compared with RANS.

A very good compilation of the hybrid RANS-LES approach is found in *Fröhlich and von Terzi (2008)*

3.1 RANS

Resolving the entire range of scales present in the flow is not necessary for most engineering purposes and just the information of the mean properties is sufficient. This makes RANS a widely used approach for the turbulence modeling [?]. The fundamental step in RANS is the decomposition of the instantaneous flow

quantities into their corresponding mean and the fluctuating part:

$$\begin{aligned} u_i(t) &= \bar{u}_i + u_i', \\ p(t) &= \bar{p} + p' \end{aligned} \quad (1)$$

This is known as the Reynolds decomposition. The incompressible continuity and the Navier-Stokes equation are :

$$\begin{aligned} \frac{\partial u_i}{\partial x_i} &= 0, \\ \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} &= -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2} \end{aligned} \quad (2)$$

Using the eq.(1) in the eq.(2) and time averaging the resulting equations we get the following equations:

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \quad (3)$$

$$\frac{\partial \bar{u}_i}{\partial t} + \bar{u}_j \frac{\partial \bar{u}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial \bar{p}}{\partial x_i} + \nu \frac{\partial^2 \bar{u}_i}{\partial x_j^2} - \frac{\partial \overline{u_i u_j}}{\partial x_j} \quad (4)$$

Eq.(3) is the time-averaged continuity equation and the eq.(4) is the RANS equation. The time-averaged Navier-Stokes equation has the same form as the instantaneous navier-Stokes equation. An additional term, $\frac{\partial \overline{u_i u_j}}{\partial x_j}$, is present in the right hand side of the eq.(4). This additional term is the result of the non-linearity in the convection term of the NS equation and it represents the effect of the fluctuating velocities on to the mean flow. This term also posses a diffusive character as the viscous stress term.

3.2 LES

$$S_{ij}^d = \frac{1}{2} (\overline{g_{ij}^2} + \overline{g_{ji}^2}) - \frac{1}{3} \delta_{ij} \overline{g_{kk}^2} \quad (5)$$

where $\overline{g_{ij}^2} = \overline{g_{ik}} \overline{g_{kj}}$ and δ_{ij} is the Kronecker delta.

$$S_{ij}^d = \overline{S_{ik}} \overline{S_{kj}} + \overline{\Omega_{ik}} \overline{\Omega_{kj}} - \frac{1}{3} \delta_{ij} [\overline{S_{mn}} \overline{S_{mn}} - \overline{\Omega_{mn}} \overline{\Omega_{mn}}] \quad (6)$$

where

$$\overline{S_{ij}} = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial \bar{x}_j} + \frac{\partial \bar{u}_j}{\partial \bar{x}_i} \right)$$

is the strain rate tensor and the rotation rate tensor is :

$$\overline{\Omega_{ij}} = \frac{1}{2} \left(\frac{\partial \bar{u}_i}{\partial \bar{x}_j} - \frac{\partial \bar{u}_j}{\partial \bar{x}_i} \right)$$

$$S_{ij}^d S_{ij}^d = \frac{1}{6} \left(\overline{S^2} \overline{S^2} + \overline{\Omega^2} \overline{\Omega^2} \right) + \frac{2}{3} S^2 \Omega^2 + 2IV_{S\Omega} \quad (7)$$

with the notations:

$$S^2 = \overline{S_{ij}} \overline{S_{ij}}, \quad \Omega^2 = \overline{\Omega_{ij}} \overline{\Omega_{ij}} \quad \text{and} \quad IV_{S\Omega} = \overline{S_{ik}} \overline{S_{kj}} \overline{\Omega_{jl}} \overline{\Omega_{li}}$$

$$\nu_t = (C_w \Delta)^2 \overline{OP}$$

$$\overline{OP} = \frac{(S_{ij}^d S_{ij}^d)^{3/2}}{(\overline{S_{ij}} \overline{S_{ij}})^{5/2} + (S_{ij}^d S_{ij}^d)^{5/4}} \quad (8)$$

4 Implementation

The WALE model has been implemented initially in the D3Q27 Matrix LBM solver [18] developed in the Institute of Computational modelling in Civil Engineering (iRMB). This code uses two set of arrays to store the distribution functions. The WALE model implementation requires the computation of the strain rate tensor and the vorticity rate tensor, which requires the information of the velocity gradients. The velocity gradients have been computed using the finite differences. The macroscopic velocities computed from the distribution functions are used to compute the velocity gradients in every time-step. Apart from the WALE model implementation the topic on the conditioning of the WALE model equation will be discussed here. Towards the end of the section the numerical results of the Taylor green vortex simulations will be compared to the analytical solution to confirm the validity of the order of of the eddy-viscosity obtained from the WALE model.

4.1 Discretisation

Several velocity gradients i.e. $\partial u_i / \partial x_i$ and $(\partial u_i / \partial x_j + \partial u_j / \partial x_i)$ are directly available from the Cumulant LB solver [5]. The computation of the strain rate tensor can be easily carried out with those available gradients. But the available information of the velocity gradients was not enough to compute the rotation rate tensor i.e. $(\partial u_i / \partial x_j - \partial u_j / \partial x_i)$. Since the vorticity calculations had to be performed using the finite difference method, it was decided to compute the velocity gradients using the finite difference method. *Weickert et.al* [6] implemented the WALE model in the MRT LB framework and they have computed the velocity gradients using the finite differences.

The purpose of the SGS models is to dissipate the resolved turbulent fluctuations [7]. When finite differences are used in combination with the LES method great care has to be taken to choose the non-dissipative finite difference scheme. This is done to avoid the masking effect of the numerical dissipation on to the dissipation introduced by the SGS models. Taking that into consideration a central difference scheme was chosen to compute the velocity gradients.

Treatment of velocity gradients near the boundaries

In order to avoid the fluid nodes, near the boundaries, accessing the non-fluid nodes due to the central differencing, forward differences and backward differences were used at the respective boundaries to compute the velocity gradients. Fig.(4.1) shows a cuboid marked with the red dots at the inlet, front and the bottom face. The first set of fluid nodes near these faces use forward difference to compute the velocity gradients and the first set of fluid nodes at the exit, top and back faces use backward differences to compute the gradients along the respective directions. Rest fluid nodes used central differences.

4.2 Implementation of eddy viscosity contribution

The WALE model has been implemented in the LB kernel itself "LB_Kernel.2SOD". Few other variations were tried but they required thread synchronisation which is not optimal to do in terms of the efficiency. Hence, it was implemented in the kernel itself. An additional viscosity called eddy viscosity (ν_t) has to be

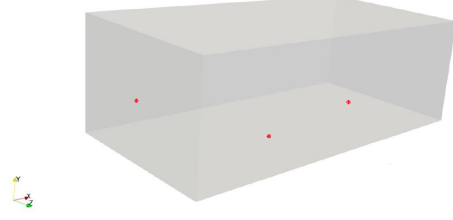


Figure 1: Cuboid representing a channel

introduced in to the solver to model the turbulence. This is given by:

$$\nu_t = (C_W \Delta_x)^2 \overline{OP} \quad (9)$$

where C_W is the model constant and $C_W = 0.5$ as mentioned in the *Nicoud et.al* [8]. In this LBM-LES investigation we will be using the implicit filtering, where the filter size is proportional to the grid size. The filter size $\Delta_x = 1$ is taken since we have a uniform lattice and a uniform mesh.

The total viscosity is the summation of the molecular viscosity and the eddy viscosity:

$$\nu_{total} = \nu + \nu_t \quad (10)$$

The total viscosity and the total relaxation time are related by [6] :

$$\tau_{total} = 3\nu_{total} + \frac{1}{2} \quad (11)$$

The value of ν is provided as the input to the solver and the ν_t is computed in the solver. To compute ν_t the eq.(6) & eq.(8) are used. Now the relaxation parameter responsible to include the transport properties in the solver is $s9$ and it is related to the total relaxation time as follows:

$$s9 = \frac{1}{\tau_{total}} \quad (12)$$

4.2.1 Conditioning of the WALE model equation

In order to perform the initial tests of the implementation an in-built case in the D3Q27 Matrix solver was used. It is a 3D case with a cube placed in the channel. These tests were to get a feeling whether the implementation is working properly or not. No comparison or any such investigation was performed with the test case. Thus, no results of the tests will be presented. It is has been mentioned here for the sake of completion.

When performing such tests it was found that the solver generated "NaN" (Not a number) in the output. After examining the individual resulting quantities viz. all terms in the eq.(8), τ_{total} etc. It was observed that for certain initial time-steps the numerator and the denominator of the eq.(8) were becoming zero and eventually it resulted in the mathematical error of the form $\frac{0}{0}$.

This issue was resolved by adding a small number in the denominator of the

eq.(8). Till now the implementation and the tests were performed using the single precision on the local machine equipped with GeForce GTX 760. The small parameter (ϵ) of the order 10^{-10} was added in the numerator and this resolved the issue of the ill-conditioned equation. The resulting eddy viscosity looks like:

$$\nu_t = (C_w \Delta)^2 \frac{(S_{ij}^d S_{ij}^d)^{3/2}}{(\bar{S}_{ij} \bar{S}_{ij})^{5/2} + (S_{ij}^d S_{ij}^d)^{5/4} + \epsilon} \quad (13)$$

Once the model started to work it was then fitted into more efficient implementation of Cumulant LB solver.

4.3 Taylor-Green vortex simulations (TGV)

Once the implementation was fitted in the new code it was tested for the case of turbulent channel flow (see section 5.1). It was observed that the resulting ν_t was very small i.e. of order 10^{-11} for single precision. Several options were tried, but it was difficult to narrow down the cause for it.

Hence it was decided to perform the Taylor-Green vortex [9] simulation. The TGV is an unsteady flow of a decaying vortex, which has an exact closed form solution of the incompressible Navier–Stokes equations in Cartesian coordinates [10]. Since we know the analytical solution of this simple flow, in comparison to the three dimensional turbulent channel flow, this case was chosen to verify the WALE model implementation for the ν_t

4.3.1 Simulation settings

A rectangular computational domain has been chosen to perform the TGV simulation. The aspect ratio of the domain, h/l , is chosen to be: 1.2. The domain is of Quasi 3D type as we want to perform two-dimensional TGV simulation. This is done as any three dimensional LB solver cannot simulate 2D flows. The input physical quantities and the output are in Lattice Boltzmann units. The width of the domain, in y-direction, is $4\Delta x$.

A hierarchy of three meshes have been generated to see the effect of resolution on to the ν_t . The resolution for length i.e. along the x-coordinate direction varies from $32\Delta x$ to $128\Delta x$ and the resolution for the height i.e. along the z-coordinate direction varies from $48\Delta x$ to $192\Delta x$. All faces of the Quasi 3D domain are assigned with periodic boundaries. The equations used for the initialisation have been specified as follows:

$$\rho(t=0) = \frac{3}{4} U^2 \left(\cos(x) \frac{4\pi}{L_x} + \cos(z) \frac{4\pi}{L_z} \right) \frac{L_z}{L_x} \quad (14)$$

$$u(t=0) = U \sin(x) \frac{2\pi}{L_x} \cos(z) \frac{2\pi}{L_z} \quad (15)$$

$$w(t=0) = U \cos(x) \frac{2\pi}{L_x} \sin(z) \frac{2\pi}{L_z} \quad (16)$$

$$v(t=0) = 0 \quad (17)$$

where $U\Delta x/\Delta t$ is the initial velocity scale and L_x , L_z are respectively the length and the height of the domain. The table (1) shows the different mesh resolutions along with the U for different mesh resolutions. The value of the kinematic viscosity was taken as $\nu = 10^{-4}m^2/s$. The viscosity is kept fixed and the resolution along with the U are varied to have the same Reynolds number for all mesh resolutions.

	$N_x \times N_y \times N_z$	U
Mesh 1	32 x 4 x 48	0.1
Mesh 2	64 x 4 x 96	0.05
Mesh 3	128 x 4 x 192	0.025

Table 1: Mesh resolution used in TGV simulations

4.3.2 Analytical settings

Using the analytical solution equations of the incompressible Navier-Stokes equation for the TGV and the WALE model the analytical ν_t can be obtained. The WALE model was implemented in MATLAB for computing the analytical ν_t . The parameters shown in table (1) are also applicable here. The two dimensional analytical solution equations [4] are as follows:

$$\begin{aligned} u &= U \sqrt{\frac{k_x}{k_z}} \sin(k_x x) \cos(k_z z) e^{-\frac{t}{t_d}} \\ w &= U \sqrt{\frac{k_z}{k_x}} \cos(k_x x) \sin(k_z z) e^{-\frac{t}{t_d}} \end{aligned} \quad (18)$$

where $k_x = 2\pi/L_x$ and $k_z = 2\pi/L_z$. The term t is the total simulation time and the $t_d = 1/\nu(k_x^2 + k_z^2)$ is the decay time of the vortex.

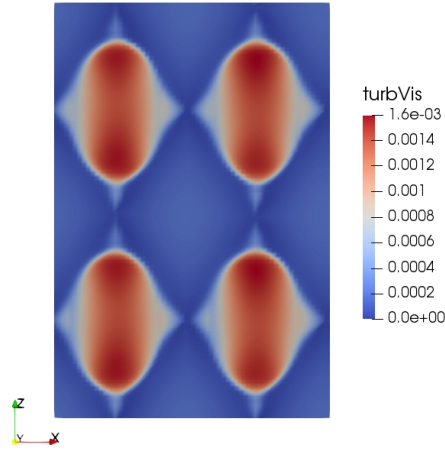
While computing the analytical value of ν_t it was observed that the terms in the numerator and denominator of eq.(13) were very small, order 10^{-15} to 10^{-25} compared to the value of ϵ added. This was the reason for not getting the correct values of ν_t . Thus the value of ϵ was set to 10^{-30} for single precision.

4.3.3 Results & comparison

In this subsection the numerically and analytically computed TGV results obtained with the changed value of ϵ is presented. The table(2) shows the number of time-steps, N_t , for which the simulation of each resolution was run and the time interval, Δt for writing out the results. With the increase in the resolution, the simulations were also run for a longer time. The factor with which the resolution changes is same as the factor of increase in the N_t .

	Time-steps, N_t	Δt
Mesh 1	4000	100
Mesh 2	8000	200
Mesh 3	16000	400

Table 2: Solver settings for TGV



(a) global coordinates

5 Model set-up and theories

Apart from the free shear flows, most turbulent flows are bounded by one or more solid surfaces depending on whether they belong to class of the internal or external flow of fluids i.e. flow through pipes and flow over a car respectively. In this investigation one of the simplest internal flow has been chosen for the validation & testing of the implementation: fully developed channel flow. This flow is considered to be of prime importance as it has played a prominent role in the development of the study of the wall bounded turbulent flows [3]. [3] has described the fundamental theories of the full developed channel flow in great detail. [11] also has a good compilation on the fully developed channel flow.

In this section a brief description of the turbulent channel flow with some basic theories is presented. Later on it is followed by description of the computational domain, meshing, initial conditions, boundary conditions, pressure gradient modelling, solver settings and in the end by post-processing.

5.1 Turbulent channel flow

Fully developed turbulent channel flow is a classic benchmark case and has been studied extensively to investigate and understand the mechanics of wall-bounded

turbulent flows. It is the geometrical simplicity of this case that has lured to a large number of, computational (DNS) and experimental, investigations being carried out for the turbulent channel flow over the years and as a result, an adequate database (DNS data) for certain Re-numbers is already available [12]. Historical progression in the turbulent channel flow simulations have been very well documented in the literature by *Kim et al.* [13]. Also, Kim has pointed out several difficulties related to the experimental measurements and how numerical simulations serve as an essential tool, alongside the experimental results, for the study of the wall-bounded turbulence. The recent development along with the comparison of the DNS results of the turbulent channel flow simulations for higher Re-numbers is documented in [14].

As shown in figure 2, a 3D rectangular duct has been chosen as the compu-

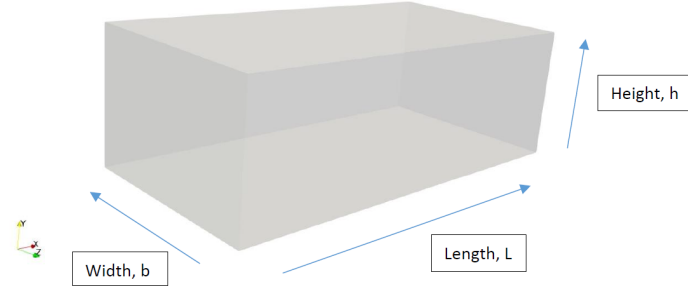


Figure 2: 3D rectangular duct

tational domain. The positive x-axis is the stream-wise direction or the flow direction with the flow varying in the y-direction, the height $h = 2\delta$, the wall-normal direction. The z-axis is the span-wise direction, the width b . The domain is long and has a large aspect ratio ($b/\delta \gg 1$). The bottom ($y = 0$) and top ($y = 2\delta$) surfaces of the domain are walls. The mid-plane ($y = \delta$), which is frequently termed as the half-channel width in the literature, is ambiguous with the assigned Cartesian coordinate directions in this case and hence term half-channel height will be used ($h = \delta$).

The mean velocities in the three directions are U , V , W , the stream-wise, wall-normal and span-wise velocities, and the corresponding fluctuating velocities are u, v and w . The mean span-wise or cross-stream velocity is zero, $W = 0$. In the turbulent channel flow test case, fully developed flow region in the channel is of our interest i.e. it is the region where the velocity statistics of the flow remain invariant to spatial changes, x-direction in our case. Also, the domain is large enough in the span-wise direction, z , compared to the half-channel height, δ , so that the flow remains statistically independent of the z -direction. In short, we have two homogeneous flow directions along x & z -axis. Thus, the flow is said to be statistically one-dimensional as the flow statistics vary only in one-direction i.e. y -direction.

5.1.1 Governing equations

Turbulent channel flow is fully developed, where the velocity statistics along the stream-wise direction remain constant i.e. $\partial U / \partial x = 0$, and statistically one dimensional, where the flow statistics vary only in the wall-normal(y) direction. There is no mean flow in the span-wise direction i.e $W = 0$ and thus the gradient of the span-wise velocity in the span-wise direction is zero i.e. $\partial W / \partial z = 0$. The mean continuity equation for the incompressible flow results in the mean wall normal velocity gradient to be zero:

$$\partial V / \partial y = 0 \quad (19)$$

The eq. 19 when integrated, within the wall limits $[0, 2\delta]$, gives $V = \text{const}(y)$ and using the no-slip boundary condition, $U_i = 0$, applied at the walls results in $V = 0$. Now to satisfy the resulting continuity equation eq. 19, V has to be zero everywhere in the domain. Using the above made considerations and applying it to the mean stream-wise momentum equation we will have the following force balance:

$$\frac{\partial \tau}{\partial y} = \frac{\partial P}{\partial x} \quad (20)$$

where $\tau = \left(\mu \frac{\partial U}{\partial y} - \rho \overline{u_i u_j} \right)$ is the total shear stress.

The eq. 20 shows that the total shear stress gradient and the pressure gradient balance each other in the full developed channel flow. It shows an equality of the form $f(x) = g(y)$ and for this equality to hold it is important that both $f(x)$ and $g(y)$ are individually constants. In eq. 20 both the gradients are dependent on different variables and thus they must be individually constants for this equality to hold.

$$\frac{\partial P}{\partial x} = \text{const} \ \& \ \frac{\partial \tau}{\partial y} = \text{const}$$

Total shear stress is the summation of the viscous stress, $\mu \frac{\partial U}{\partial y}$, and the Reynolds stress, $-\rho \overline{u_i u_j}$. At the wall with the no-slip boundary condition applied, the Reynolds stresses are reduced to zero and thus the shear stress near the wall or to be specific at the wall would entirely be dominated by the viscous stresses. At the walls the Reynolds stresses drop to zero and the stress at the walls can be defined as the *wall shear stress*, the region where the viscous stresses are dominant:

wall shear stress

$$\begin{aligned} \tau_w &= \mu \frac{\partial U}{\partial y} \\ \tau(0) &= \tau_w = -\tau(2\delta) \end{aligned} \quad (21)$$

Now we can integrate the eq. 20, from 0 to 2δ , for the pressure gradient and express the solution in terms of the τ_w :

$$-\frac{\partial P}{\partial x} = \frac{\tau_w}{\delta} \ , \quad (22)$$

From the fully developed channel flow it is known that the mean velocity profile

(time-averaged) remains constant. It is clear from eq. 21 that wall shear stress is linearly proportional to the velocity gradient near the wall. A constant velocity profile implies that the wall shear stress will also be constant in a fully developed channel flow. Thus from eq. 22 we can say that the pressure gradient is constant. Integrating the eq. 20, from 0 to some y , with the no-slip boundary condition at the wall results in the following solution:

$$\tau(y) = \tau_w \left(1 - \frac{y}{\delta}\right) \quad (23)$$

It is this pressure gradient that drives the flow through the infinite parallel plates. The pressure gradient in eq. 22 is negative as the viscous effects decrease the pressure along the stream-wise direction. For a given pressure gradient and the half-channel height we get a linear shear stress profile from eq. 22 and 23 independent of the fluid properties and the state of the fluid motion (laminar or turbulent) [3]. Experimental and numerical results [13] show that the total shear stress profile is anti-symmetric about the channel mid height $y = \delta$ and thus $\tau(\delta) = 0$.

5.1.2 Scaling units

The Reynolds stresses decrease towards the wall, remain zero at the wall and so does the eddy viscosity (*Boussinesq hypothesis*). The value of the eddy viscosity ranges from zero, at the wall, to several orders of magnitude higher than the molecular viscosity, in the core region of the turbulent boundary layer [1]. From the figure. 3 it can be seen that the velocity profile changes more slowly (more flatter compared to the parabolic flow profile) in the core of the turbulent boundary layer, where the eddy viscosity is higher than the molecular viscosity, and there is an abrupt change in the velocity profile near the wall, where the molecular viscosity is dominant. The turbulent flow profile has the largest gradient near the wall.

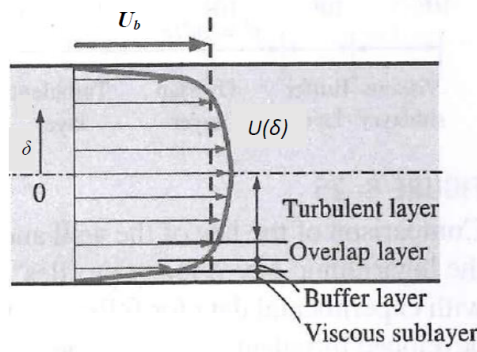


Figure 3: Velocity profile in the fully developed channel flow [1]

Turbulent flow along the wall can be considered to consist of four different layers, characterised by their distance from the wall. All four layers have been approximately shown in figure 3. The layer closest to the wall is the viscous sub-layer, where the viscous effects are dominant. The velocity profile changes

linearly with the distance from the wall and the flow is streamlined. Next to the viscous sub-layer is the buffer layer, where the turbulent stresses start to develop, but still the viscous stresses are dominant. Above the buffer layer is the overlap layer, where the turbulent stresses are significant, but still not dominant enough as the viscous stresses. The layer above the overlap layer for the rest of the flow is the outer or turbulent layer, where the turbulence stresses dominate over the viscous stresses.

A single analytical relation for the entire turbulent velocity profile is not suitable, as different regions or layers execute different flow characteristics. The best possible option is to find the key parameters or the functional forms using the dimensional analysis and later on experimental data can be used to obtain the numerical values of the any constants.

i. Viscous sub-layer : With the no-slip boundary condition applied at the wall, $u_i(0, t) = 0$, the entire contribution to the shear stress is purely via viscous stresses i.e. τ_w . From eq. 21 it is clear that shear stress is proportional to the molecular viscosity. The thickness of this layer is very small, but it plays an important role on the flow characteristic because of the presence of large velocity gradients in it. The velocity profile remains linear in this layer and also it has been shown in certain experiments [1]. Since the velocity profile is linear the gradient in the equation 21 remains constant i.e. $\partial U / \partial y = U / y$. Thus the wall shear stress can be written as:

$$\frac{\tau_w}{\rho} = \frac{\nu U}{y} \quad (24)$$

Near the wall the important parameters are : τ_w , ρ , ν . The square root of the term on the left hand side has the dimensions of velocity and thus we define a new parameter as the **friction velocity**, u_τ :

$$u_\tau = \sqrt{\frac{\tau_w}{\rho}} \quad (25)$$

Replacing u_τ in the eq. 24 we have **the law of wall**:

$$\frac{U}{u_\tau} = \frac{u_\tau y}{\nu} \quad (26)$$

The quantity ν / u_τ has the dimensions of length and it is called **viscous length**, δ_ν , and it is used to nondimensionalize the distance from the wall, y . In the analysis of the boundary layer flow, it is convenient to work with the nondimensionalised distance and the velocity. The nondimensionalised quantities are represented with a +.

Nondimensionalised variables :

$$y^+ = \frac{y u_\tau}{\nu} \text{ and } u^+ = \frac{U}{u_\tau} \quad (27)$$

Normalised law of the wall :

$$u^+ = y^+ \quad (28)$$

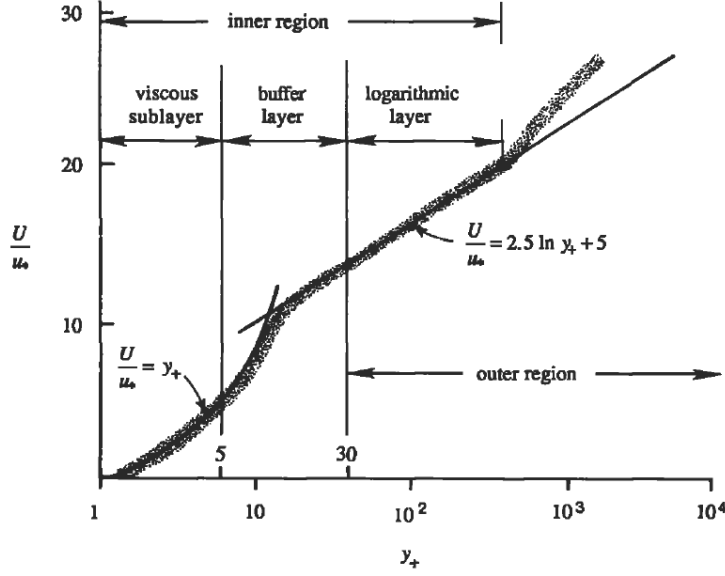


Figure 4: Nondimensional velocity u^+ vs. the nondimensional distance y^+ (plotted in logarithmic scale) [2]

This law shows a very good correlation with the experimental data for smooth surfaces for $0 \leq u_\tau y / \rho \leq 5$. Thus for the $y^+ = 5$ we will have the linear velocity profile in the viscous sub-layer.

ii. Overlap layer: From the dimensional analysis and the experimental results it proved that the velocity in the overlap region is proportional to the logarithm of distance and it can be expressed as:

$$u^+ = \frac{1}{\kappa} \ln y^+ + B \text{ or } u^+ = 2.5 \ln y^+ + 5.0 \quad (29)$$

where κ and B are constants and equal to 0.40 and 5.0 respectively. The eq. 29 is called the *logarithmic law* and the overlap layer is also called as the *logarithmic layer* or *log-law layer*. It is clear from the figure 4 that the eq. 29 represents the experimental data quite satisfactorily from $y^+ \geq 30$ till outer region. Neither eq. 29 nor the eq. 26 represents the experimental data satisfactorily in the buffer layer. From Figure. 4 it is seen that :

viscous sub-layer : $y^+ \leq 5$

buffer layer : $y^+ \geq 5$ till $y^+ \leq 30$

logarithmic layer : $y^+ \geq 30$ till $y^+ \leq 10^3$

iii. Outer layer: The velocity law for this region is:

$$\frac{(U_{max} - U)}{u_\tau} = \frac{1}{\kappa} \ln \left(\frac{y}{\delta} \right) \quad (30)$$

The difference between the mean center line velocity, U_{max} , and the mean velocity, U is called the **velocity defect** and thus this law is called the **velocity defect-law**. In this region the turbulent stresses are dominant compared to the viscous stresses and this is also seen from the eq. 30: the normalised velocity profile in the core region of the turbulent channel flow is dependent on the half-channel height and is not dependent on the viscosity.

5.2 Reynolds number

Since the Reynolds number is used to characterize the flow, great care was taken to simulate the same flows i.e. same Reynolds number as that of the DNS data was used. The dimensionless Reynolds number requires three parameters for its definition: characteristics length (δ), characteristic velocity (V_{ch}) and the kinematic viscosity (ν) of the fluid.

$$Re = \frac{V_{ch}\delta}{\nu}$$

Several choices of V_{ch} are possible, but in this investigation two different V_{ch} have been chosen and the respective Reynolds number are as follows :

i. Friction Reynolds number (Re_τ):

$$Re_\tau = \frac{u_\tau\delta}{\nu} = \frac{\delta}{\delta_\nu} \quad (31)$$

ii. Bulk Reynolds number (Re_b):

$$Re_b = \frac{U_b}{\nu} 2\delta \quad (32)$$

The bulk mean velocity is defined as:

$$U_b = \frac{1}{h} \int_0^h U dy$$

5.3 Computational Domain and Meshing

5.3.1 Computational Domain

We are interested to investigate the fully developed turbulent channel and that implies a sufficient length and width of the domain for the flow in the channel to be fully developed. Since the stream-wise and the span-wise directions are homogeneous, periodic boundary conditions are applied on either side of the two homogeneous directions. All flow properties along with the state of the flow are set to be equal for the equivalent points on the periodic boundary pairs. Use of periodic boundaries is equivalent to having an infinitely long domain in that direction. This implies that we have a flow between two infinite parallel planes, Figure 5. Since the artificial periodic boundaries are enforced great care has to be taken that the domain is long enough to avoid this artificial boundary conditions from influencing the results. *Moin et.al* [15] has specified certain aspects to be considered in the process of choosing the dimensions of the flow domain for the numerical simulation of the turbulent channel flow. According

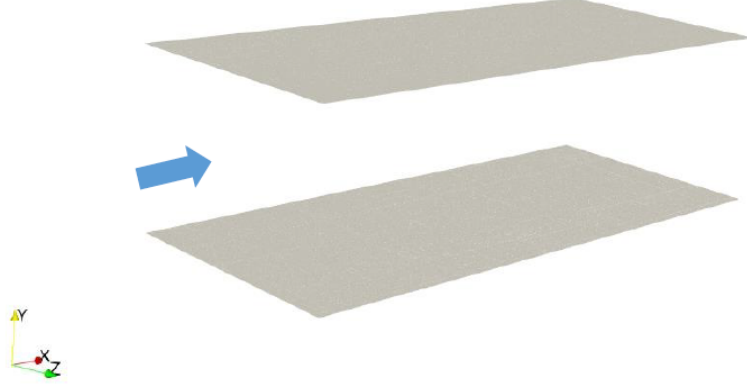


Figure 5: Flow between the infinite parallel plates

to [15] the experimental two-point correlation functions have to be considered for choosing the stream-wise (L_x) and the span-wise (L_z) lengths. The usage of periodic boundary conditions is justified if the distance between the pair of periodic boundaries is at least twice the distance, at which the correlation function becomes zero. This means that the domain size is enough to accommodate the largest eddy without the pair of periodic boundaries influencing each other.

To validate the LES results of the fully developed turbulent channel flow the DNS data from the database of [16] for the $Re_\tau = 395$ is used. The authors have shown the adequacy of the domain by performing the two-point spatial correlations. Same domain size used in [16] has been chosen for performing the LES of channel flow, Table 3. Thus, no separate study for two-point correlation has been performed in this investigation.

$Nominal Re_\tau$	$L_x \times L_y \times L_z$
395	$2\pi\delta \times 2\delta \times \pi\delta$

Table 3: The size of the domain in x, y, z directions

Several authors [17, 18] have performed the turbulent channel flow simulations with smaller domain dimensions compared to the dimensions used in the reference data, [16], in order to either save the computational time or to satiate the upper limit of the total grid points on a Graphic card [18]. But *Bespalko* [19] suggests that for performing the validation studies the dimensions of the domain should be similar to that used in the reference data. Else, the usage of smaller domain dimensions might add some kind of error and this might be than attributed to the error originating from the implementation under test.

5.3.2 Physical parameters

The physical parameters that are necessary for the model set-up and later on for the simulations are pressure gradient(to drive the flow), discussed in the later

section, u_τ , ν and U_b In [16] the authors have not specified u_τ , ν and U_b used in the simulations. [20] has computed and specified these parameters obtained from the DNS results of [16] for $Re_\tau = 395$, Table 4 :

<i>Physical quantity</i>	<i>Value in S.I Units</i>
Half channel height, δ	1.0 m
Friction velocity, u_τ	0.0079 m/s
Bulk velocity, U_b	0.1335 m/s
Kinematic Viscosity, ν	0.00002 m ² /s

Table 4: Physical quantities

In order to compute the bulk mean velocity from the DNS data (Mean stream-wise velocity), trapezoidal rule can be used to solve the equation of U_b numerically. i.e. *trapz* command in Matlab.

5.3.3 Meshing

In order to see the effects of mesh resolution a hierarchy of meshes was created. Three uniformly spaced meshes, in all directions, were generated namely: Mesh 1 (coarse), Mesh 1.5 (medium) and Mesh 2 (fine). The resolution of Mesh 1 was refined by a factor of 1.5 in all directions for Mesh 1.5 and similarly by a factor of 2 for Mesh 2. To compute the mesh spacing the dimensionless distances in wall normal and wall parallel directions (stream-wise and span-wise) were used. The u_τ used for spacing computation is taken from the DNS data. Initially the plan was to go for Mesh 3, refinement by factor 4 in all directions, but the total number of mesh points were ~ 88 million, so the idea was dropped and an intermediate mesh resolution was chosen and hence the name Mesh 1.5.

$$\begin{aligned}
 x^+ &= \frac{\Delta x u_\tau}{\nu} \\
 y^+ &= \frac{\Delta y u_\tau}{\nu} \\
 z^+ &= \frac{\Delta z u_\tau}{\nu}
 \end{aligned} \tag{33}$$

Because of the uniform meshes the spacing is same in all three directions and from now onward it will be referred as Δ^+ . The bottom and top faces of the domain were defined as the wall, no-slip boundary condition. The no-slip boundary condition was realized by using the half-way bounce back boundary condition. Because of the half-way bounce back boundary condition the distance of the nearest cell to the wall, y^+ , is halved [4]. The y^+ will always be a smaller than Δ^+ by a factor of 2. The following table 5 shows the mesh statistics used in the investigation, where N_x , N_y , N_z denote the number of nodes in the stream-wise, wall-normal and span-wise directions. In order to facilitate the grid generator the size of domain was reduced to $6\delta \times 2\delta \times 3\delta$.

The domain size has been slightly altered in this investigation and this leads to the necessity for performing the two-point correlations to prove the adequacy of the domain. Premnath *et.al* [21] suggests doing it otherwise.

	$N_x \times N_y \times N_z$	Total nodes	$\sim \Delta^+$	$\sim y^+$
Mesh 1	192 x 64 x 96	1179648	12	6
Mesh 1.5	288 x 96 x 144	3981312	8	4
Mesh 2	384 x 128 x 192	9437184	6	3

Table 5: Mesh statistics

5.4 Initial conditions

It is known from the theory of the internal fluid flow that there exists a hydrodynamic entry length (L_h), starting from the inlet ($x = 0$), that the fluid has to travel, in the channel, for the flow to become fully developed. This means when we integrate the flow governing equations, in time, the fluid has to travel the distance L_h ($\sim x = 10H$) in every time-step, to generate a fully developed flow, which is unnecessary considering the computational efforts and also the storage requirements. The idea is to consider only that portion of the channel which will be fully developed, turbulent and neglecting the initial part (L_h) of the channel. This implies that specific initial conditions for the flow to become fully developed and turbulent has to be provided.

Different approaches have been used by different authors to generate the initial condition for the incompressible, three-dimensional, turbulent channel flow simulations. *Moin et.al* [15] has specified few consistency criteria for initial velocity field. *Sagaut* [22] & *Fröhlich* [11] have a concise description of most commonly used approaches with their respective advantage, disadvantages and their applications to suitable flows. Two of the several approaches mentioned in [22] & [11] will be discussed here.

i. First approach: In this approach a simulation is performed for some Reynolds number smaller than the one to be simulated. The resulting fields of this simulations are then used as the starting point for the required Reynolds number and is simulated further in time until a steady state, where the statistics of the flow do not change with time, is achieved. This approach is termed as *Precursor simulation* in several literature.

ii. Second approach: In this approach a laminar flow is chosen as the base flow and certain set of disturbances are applied on the base flow. The base flow super-imposed with disturbances is then integrated in time until the flow transitions to a turbulent flow. This approach is computationally intensive, as the flow takes time to become turbulent. There are several options available to impose the disturbances on to the base flow. Several authors have used this approach for the generating appropriate initial conditions. In this investigation this approach has been used.

For the initialisation of our simulations the approach of *Eugene de Villiers* [20] was followed. *Eugene de Villiers* introduced some sort of near wall streak cycle, where he uses the fundamentals of the turbulent structures near the walls. A parabolic flow profile is chosen as the base flow. To generate the parabolic flow profile the bulk mean velocity, $U_b = 0.1335 \text{ m/s}$, which was computed by [20], from the mean velocity DNS data of [16] for $Re_\tau = 395$, is used.

Parabolic flow profile has been applied in the stream-wise direction, $u(y, 0)$, on all the fluid nodes in the domain. Figure 6 shows the parabolic profile used as

the base flow. A combination of sine and cosine waves have been used to break the flow symmetry and to develop the turbulence in the flow. Initially the the

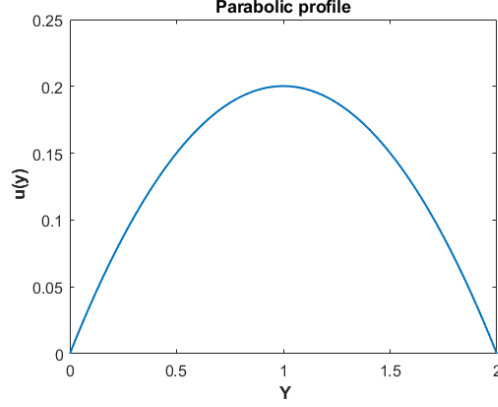


Figure 6: Parabolic flow profile with $U_b = 0.1335 \text{ m/s}$

disturbances were applied in the stream-wise direction, but that approach was quite time consuming. It took approximately 4.5 million time-steps for the flow to become turbulent. In the next approach the disturbance was applied in the wall-normal direction, v . This approach reduced the time it took for the flow to become turbulent, 100,000 time-steps. The exact equations for the initialization have been specified in the the eq. 36 in Appendix.

5.5 Modelling of pressure gradient

In the theoretical section it was shown that the pressure gradient drives the flow through a channel. With the periodic boundary conditions applied in the both streamwise and spanwise direction no information on the pressure gradient is available and thus some kind of external forcing is required to drive the flow through the channel. This requires an additional step, to model the pressure gradient, in the numerical simulation. An additional force term will be added on the right hand side of the Navier-stokes equation. Forcing is only applied in the streamwise direction, F_x . Two most commonly used modelling strategies will be discussed here.

i. Constant Pressure gradient :

In this approach we set F_x to some constant value, thus explicitly setting the driving force, which becomes constant and uniform. Since the forcing is constant the mass flow rate will be varying in time, which is attributed to the wall shear stress, until the flow is fully developed (see eq. 22).

$$F_x = -\frac{\partial P}{\partial x} = \frac{\tau_w}{\delta} = \frac{\rho u_\tau^2}{\delta} \quad (34)$$

Thus the simulation a posteriori yields the mean flow rate, U_b , required to generate the prescribed frictional losses [23]. [24, 17] have used this forcing strategy.

ii. Constant Flow rate : In this approach the forcing is not constant, but it is time dependent. The forcing is adjusted in every time step to keep the

mass flow rate in the channel equal to the specified flow rate. Flow rate per channel cross-sectional area is U_b . Thus U_b is provided as the input and we get the mean friction velocity, u_τ as the output which we compare with that of the DNS result. This is the most widely used approach in the turbulent channel simulations [20, 18, 25] and it has been utilised in this investigation. Detailed description of this approach is specified rarely and the authors [23, 26] describe the approach in detail. [23] has shown a comparison between different forcing strategies applied on the turbulent channel flow.

5.6 Solver settings

Once the flow in the channel becomes turbulent, the flow is allowed to further develop in time until it reaches a statistically steady state i.e. the statistics of the flow variables are invariant to temporal changes. Once the velocity field reaches the statistically steady state, the equations are further integrated in time to obtain the time average of the various statistical correlations and these time averaged quantities are further averaged in space along the two homogeneous directions i.e spatial averaging over the x-z planes.

To determine the statistically steady state in the channel one can observe the evolution of the several quantities viz. turbulent kinetic energy (volume averaged), τ_w , u_τ etc. over some non-dimensional time. Refer to [27] for the detailed description for steady state determination and spatial averaging. Here two non-dimensional time will be specified *large-eddy turn-over-time (LETOT)*, tu_τ/δ , and the flow through the domain time, tU_b/δ . The u_τ obtained from DNS data is used for computing the LETOT. LETOT is the time required by the large eddy to breakdown and transfer the energy to the small eddies [19]. In this investigation no specific quantity had been monitored to determine the statistically steady state. An estimation that ~ 10 LETOT is sufficient for the flow to be fully developed and statistically steady [28] had been followed.

The lb simulations are generally performed in *lattice units*, where all the physical parameters are represented by dimensionless numbers. This implies that converting the dimensional quantities into dimensionless lattice quantities and vice versa to interpret the simulation results. Exploiting the law of similarity we can ensure that we simulate the same kind of flow in both the unit systems by maintaining certain non-dimensional parameters equal in both unit systems i.e. Reynolds number.

LB is mostly used for simulating the incompressible fluids where the Mach number is smaller ($Ma < 0.3$). The flow in this investigation is also incompressible. Smaller Mach number $Ma \ll 1$ implies that the flow in the channel evolves with smaller time-step, Δt , i.e. flow evolves slowly and thus renders the simulation as computationally expensive. In order to avoid that, Ma has been artificially increased to speed-up the simulation i.e. coarser Δt by keeping the Re number constant. The computation of the parameters required to increase the Ma is specified in the Appendix. The table 6 shows the actual, Ma_{act} , and the artificial, Ma_{art} , Mach number.

There are two stages in the simulation of turbulent channel flow: Flow development stage and the averaging stage. In the flow development stage the flow is allowed to develop till it reaches the statistically steady state and in the averaging stage the statistics of the flow variables are sampled. Table 7 shows

Mach number	Value
Ma_{act}	0.000389213
Ma_{art}	0.180647487

Table 6: Mach number

the time, dimensional and non-dimensional values, required by different mesh resolutions for each stage. As suggested by [28] ~ 10 *LETOT* is sufficient for

	$\sim time$	Mesh1	Mesh1.5	Mesh2
Flow developing stage	$t [s]$	7324	9806	12871
	tu_τ/δ	57	77	101
	tU_b/δ	977	1309	1718
Averaging stage	$t[s]$	65917	76002	69873
	tu_τ/δ	520	600	552
	tU_b/δ	8800	10146	9328

Table 7: Solver settings

the flow development stage, but the table 7 shows some higher values as it also involves the duration over which the flow becomes turbulent. In the averaging stage the averaging was carried out for longer duration to make sure that we have symmetric Reynolds stress component profiles. This decision was made after observing the profiles generated from every temporal and spatial averaged data set. Mean velocity profiles relatively require less time to have a steady symmetric profile.

5.7 Computational details & Post-processing

Simulations have been carried out on the Phoenix cluster [29] of TU Braunschweig. Computations have been performed on the one of the 8 available GPU nodes. Visualisation of the large data sets has been performed using the one of the 6 available visualisation nodes on the cluster. Time averaging of the flow variables along with the other turbulence statistics is performed during the simulations within the solver itself i.e. running time average as referred in the literature [15]. Spatial averaging is performed a posteriori in Matlab. Paraview together with python scripting is used to manipulate and extract the data from the resulting time-averaged data sets. Paraview is also used for the visualisation of the instantaneous data i.e. velocity contour plots, iso-volumes etc. The data extracted by the python scripts is then used by the Matlab scripts to further manipulate and visualise the data in the form of profile plots i.e mean velocity profiles, Reynolds stresses.

6 Results and discussion

In this section the results of the simulations will be presented and discussed. Two different variations of the LB solver were tested. One with the optimised values of the relaxation parameters [5] and other with the relaxation parameters used as specified in [30]. The results presented in this section are performed with the former variation of the LB solver and the results of the latter variation will just be presented in the Appendix (A.3), as the choice of value one for all the parameters is the most stable choice but not accurate enough [5]. Firstly, the results of the LB simulations carried out without any turbulence model will be presented and discussed. The result of the simulations with no-model will be termed as the *under resolved DNS* (UDNS). Under resolved because the resolution is coarser compared to the reference mesh resolution and DNS because no additional modelling is done for turbulence. It will be followed by the discussion of the results from the LB-LES with the WALE model and will be compared with the UDNS and DNS results. Few general comments applicable for all the simulations are:

- The DNS results of [13], for $Re_\tau = 395$, will be used as a reference for the comparison. The quantities obtained from the DNS data will be referred to as the *target* quantities.
- The results shown here are plotted against the global coordinates, y/δ , and local coordinates, y^+ .
- All simulations have been performed using the single precision on GPG-PUs.
- The normalised wall distance y^+ used in the profile plots is computed a posteriori for all the simulations.

6.1 UDNS results

To test the accuracy of the turbulent channel flow results from the LB solver, against the DNS data, these simulations have been performed. The physical quantities resulting from the simulations for all meshes are listed in the table (8). The average friction velocity (u_τ) and the Re_τ resulting from the computed u_τ are the important quantities. From now on u_τ represents the average friction velocity. The computed value of u_τ is under-predicted for all the simulations and so is the resulting Re_τ , but the values approach the target u_τ with the increase in the mesh resolution. We know from the definition of u_τ , that u_τ is proportional to the τ_w . Thus, smaller values of the u_τ implies smaller values of τ_w .

Physical quantity	Mesh1	Mesh1_5	Mesh2	Target
$u_\tau, m/s$	0.0070	0.0074	0.0076	0.0079
Re_τ	352	371	382	395
$U_c, m/s$	0.1565	0.1563	0.1557	0.1591

Table 8: Comparison of the resulting physical quantities

6.1.1 Mean velocity profile

Mean velocity profiles presented here have been averaged in time and space. It is seen from the Fig. (7), the mean velocity profiles for all the meshes over-predict the DNS data. As discussed in the section 5.1.2 the viscous sublayer plays a dominant role on the flow characteristics as higher velocity gradients are present there. Since the viscous sublayer is under-resolved in the simulations, u_τ and eventually the τ_w is under-predicted and as a result we see the different flow characteristics obtained from the simulations i.e. over-predicting velocity profiles in comparison to the DNS data. It is clear that with the better resolution of the viscous sublayer, the correct flow characteristics will be obtained and it is apparent from the Fig. (7) that with the mesh refinement, the velocity profiles approach closer to the DNS data. Velocity profile for mesh3 nearly collapses on to the DNS data. Thus a positive effect of mesh resolution is seen from the mean velocity profile plots.

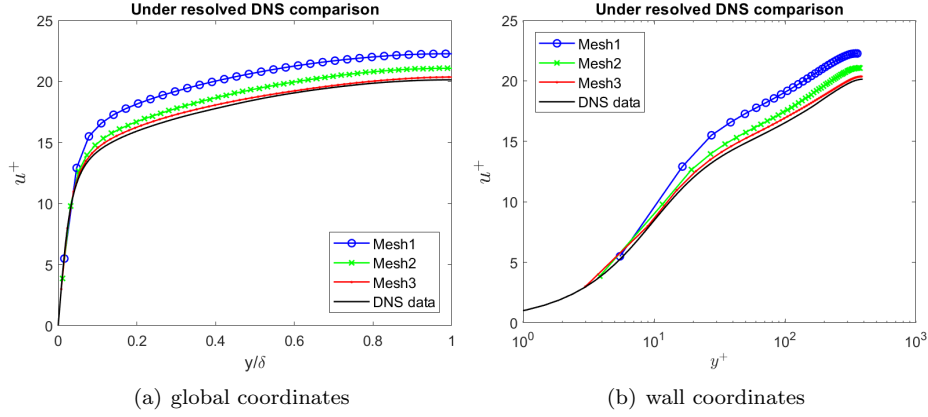


Figure 7: Mean stream-wise velocity profile normalized by u_τ

6.1.2 Turbulence intensities

Turbulence intensities is the term used for the root-mean-square (rms) profiles of the velocity fluctuations. The rms profiles shown in this section have been averaged in the following order: the time-averaged data set written out in every time-step is spatially averaged and these time and space averaged data sets are again time-averaged to have symmetric rms profiles. The Fig. 8 shows the rms velocity fluctuations plotted in the global coordinates. The rms profiles are symmetric about the half-channel height i.e. $y = \delta$ and the symmetry of the profiles about δ indicates the adequacy of the sampling taken for average [13]. Turbulence intensities normalised by u_τ are compared with different mesh resolutions and the DNS data¹.

Fig.(9)-(11) shows the comparison of the components of the rms profiles of the streamwise, wall-normal and spanwise velocity fluctuations, respectively,

¹the DNS data for the velocity fluctuation profiles have been provided as variances ($\overline{u'^2}$) and not the rms ($\sqrt{\overline{u'^2}}$) values

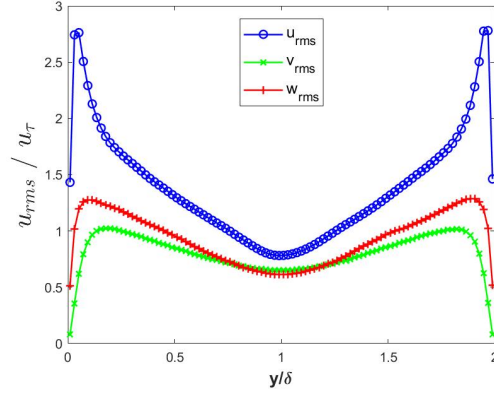


Figure 8: Root-mean-square velocity fluctuations normalized by u_{τ} in global coordinates

with the different mesh resolutions and the DNS data in wall coordinates. The profiles do not start from zero value, as only the fluid nodes are plotted. The general observation from all three velocity fluctuation profiles is that they converge to the DNS data with the increase in the mesh resolution.

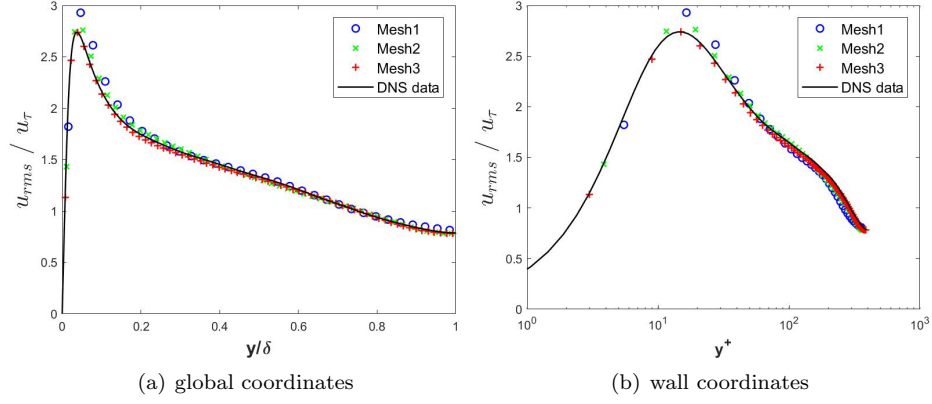
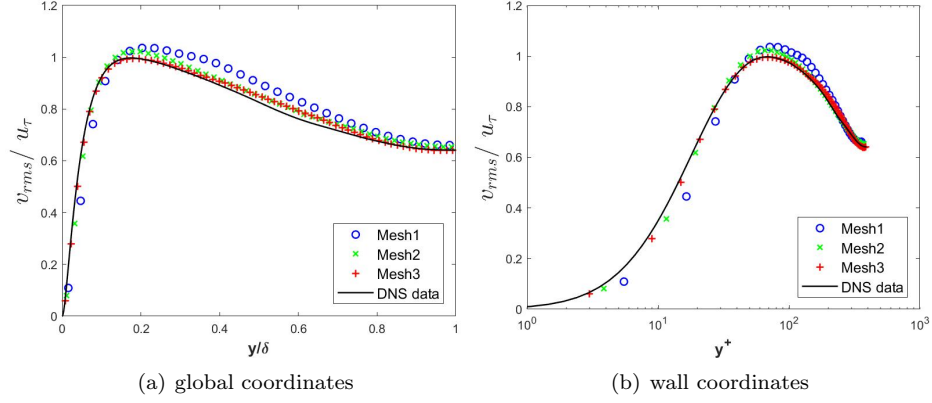
Mesh 2 and Mesh 3 accurately estimates the location of the peak values for all three velocity component profiles, respectively, compared to the DNS data. Mesh 3 for all the velocity component profiles predict the peak value fairly close to that of the DNS data. All three velocity component profiles for Mesh 3 are in fairly good agreement with the DNS data.

The rms value of the streamwise profile increases from zero at the wall to some peak value in either buffer layer, $5 < y^+ < 30$, or in the log-law layer, $y^+ > 30$ till $y/\delta < 0.3$ and then drops gradually towards the centre of the channel. The velocity profile for the wall-normal component develops slowly in comparison to the streamwise and the spanwise component profiles. The approximate location of the peak value for streamwise profile (DNS) is at $y^+ = 14$ [13] i.e. buffer layer. The location of peak value for the spanwise component (DNS) is at $y^+ = 40$ i.e. log-law region and the location for the wall-normal component (DNS) is at $y^+ = 70$ also in the log-law region, but towards the end of it. The resulting profiles of Mesh 3 for all components approximately shows the same locations for the peak values as that of the corresponding DNS data.

6.1.3 Turbulent shear stress

Turbulent shear stress is also referred to as the co-variance of the Reynolds stress tensor i.e. $-\overline{u'v'}$. This component is responsible for the turbulent diffusion of the momentum. It is also sometimes referred to as the resolved shear stress.

Fig. (12) shows the turbulent shear stress profile normalised with the u_{τ}^2 plotted in wall coordinates and compared against the DNS data and with the mesh resolutions. As seen from the Fig. (12) the increase in the mesh resolution results in the better agreement to the DNS data. Mesh 3 here shows a very good agreement to the DNS data and also it predicts the location of the peak value, $y^+ = 35$ fairly well.

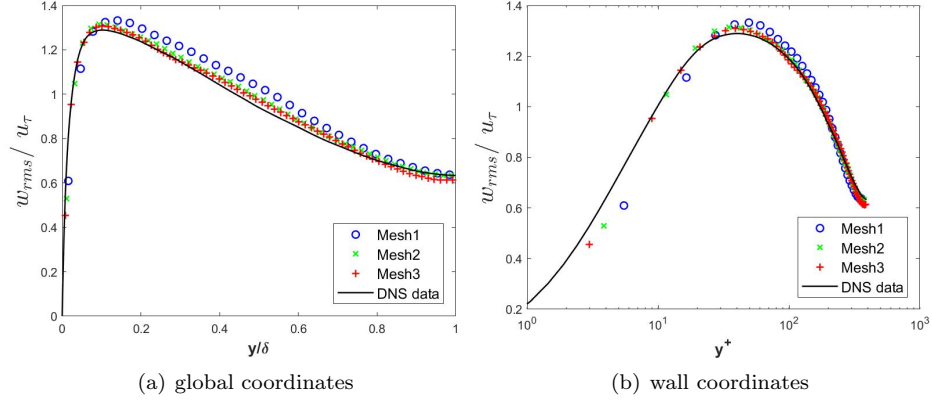
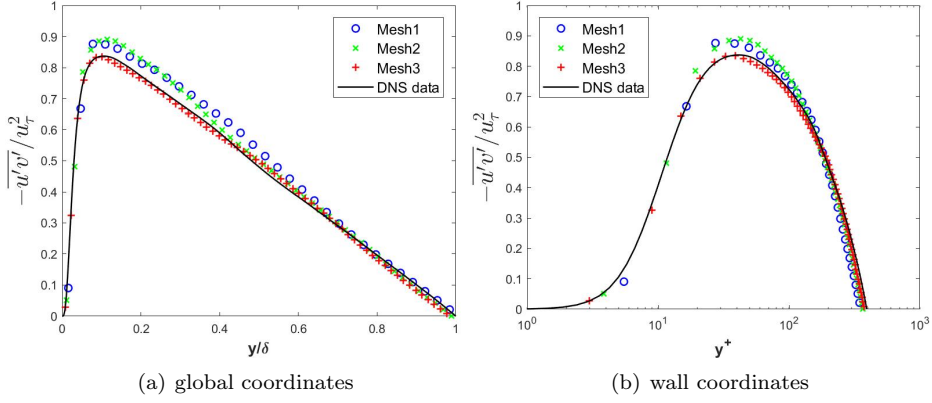
Figure 9: Rms profiles for the streamwise velocity fluctuation normalised by u_τ Figure 10: Rms profiles for the wall-normal velocity fluctuation normalised by u_τ

6.2 WALE model results

In this section the results obtained from the combination of the LB flow solver with the WALE model (LES) performed on the relatively coarser mesh, in comparison to the resolution of the DNS data, will be presented. The main purpose is to access the accuracy with which the aforementioned combination reproduces the mean flow profiles and other turbulence statistics when compared to the DNS data. The computed physical quantities have been listed in the table 9 for all mesh resolutions. The same trend is seen for the physical quantities as mentioned in the section 6.1. The physical quantities approach closer to the respective target values with the increase in the mesh resolution.

6.2.1 Mean velocity profiles

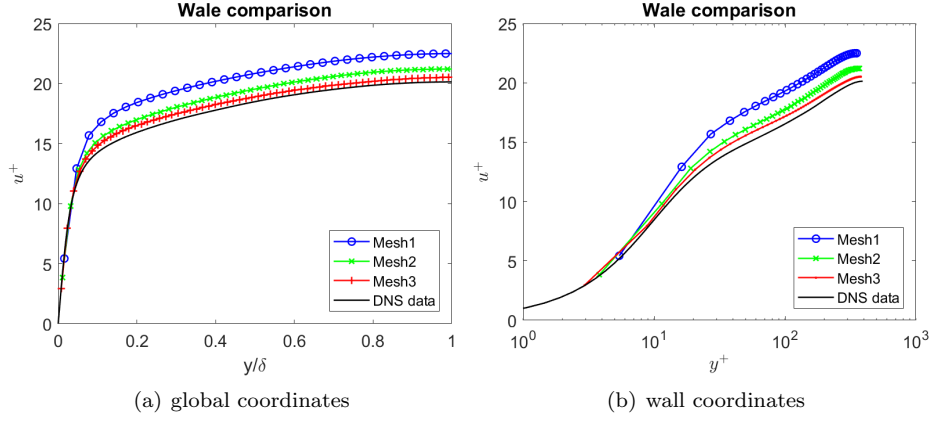
The mean velocity profiles, normalised with the u_τ , presented here are averaged in time and space. Fig. (13(a-b)) shows the mean velocity profiles plotted in

Figure 11: Rms profiles for the spanwise velocity fluctuation normalised by u_τ Figure 12: Turbulent shear stress normalised by u_τ^2

Physical quantity	Mesh1	Mesh2	Mesh3	Target
$u_\tau, m/s$	0.0069	0.0073	0.0075	0.0079
Re_τ	347	367	377	395
$U_c, m/s$	0.1562	0.1557	0.1547	0.1591

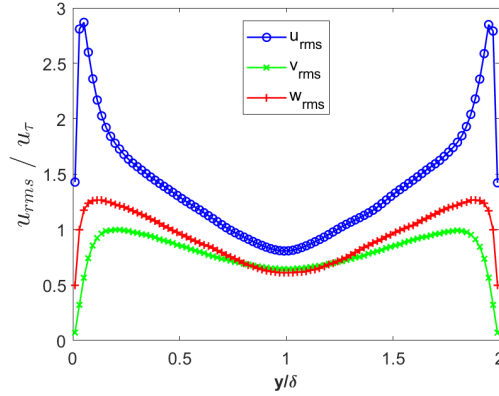
Table 9: Comparison of the computed physical quantities using WALE model

the global and local coordinates. It is clear from the both the figures that with increase in the mesh resolution the velocity profiles approach closer to the DNS data. All profiles over-predict the DNS data, but the velocity profile for the mesh 3 is relatively in good agreement with the DNS data. The reason for the over-prediction is similar to that explained in section 6.1.1.

Figure 13: Mean velocity profile normalised with u_τ

6.2.2 Turbulence intensities

The rms profiles presented here are averaged in the same fashion as that of the section 6.1.2. To show the adequacy of sampling taken for averaging the symmetric rms profiles, about the half-channel height $y = \delta$, of all three velocity fluctuations is shown in Fig. (14). The comments made in the section 6.1.2 for the velocity profiles are also applicable here. The rms velocity profiles for all three components show a good agreement towards the DNS data as the mesh resolution increases.

Figure 14: Root-mean-square velocity fluctuations normalized by u_τ in global coordinates

From Fig. (15) it is clear that the location of the peak value is predicted nearly correctly by rms profile of mesh 3. Rms profiles of all meshes overpredict the reference(DNS data) peak value. Overall mesh 3 shows a better agreement to the DNS data, although with a slight overprediction of the peak value.

From Fig. (16) it appears that profile for mesh 1 is shifted towards the right and so is the location of the peak value. With mesh refinement the profiles shift towards the left and the peak value is under-predicted by mesh 1 and mesh 3.

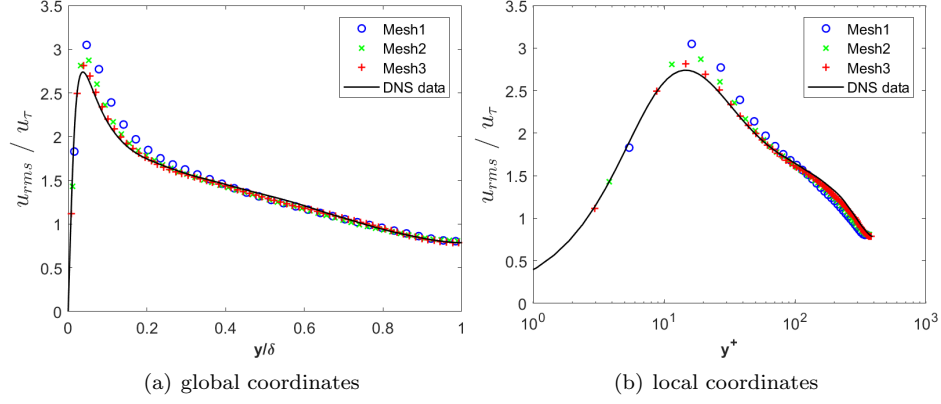


Figure 15: Rms profiles for the streamwise velocity fluctuation normalised by u_τ

Mesh 2 somehow overpredicts the peak value. All profiles underpredict in the viscous sublayer (mesh 2 and mesh 3) and the buffer layer. Resolution for mesh 3 approximates the DNS data quite well, although, with some underprediction in the log-law layer i.e. peak value.

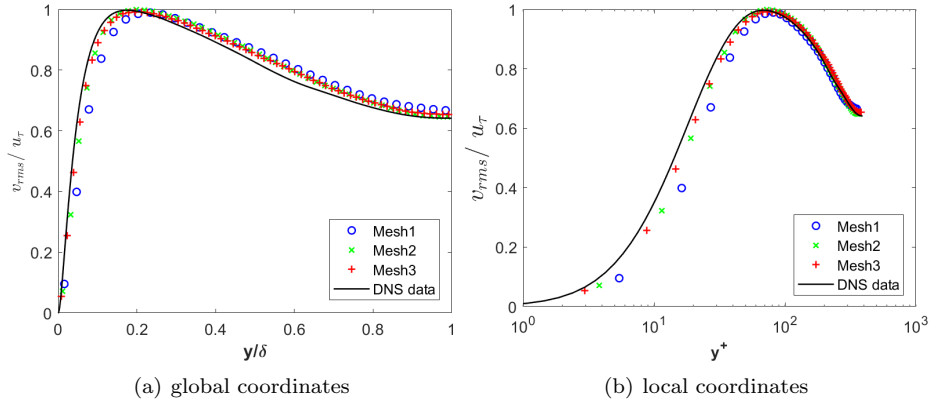
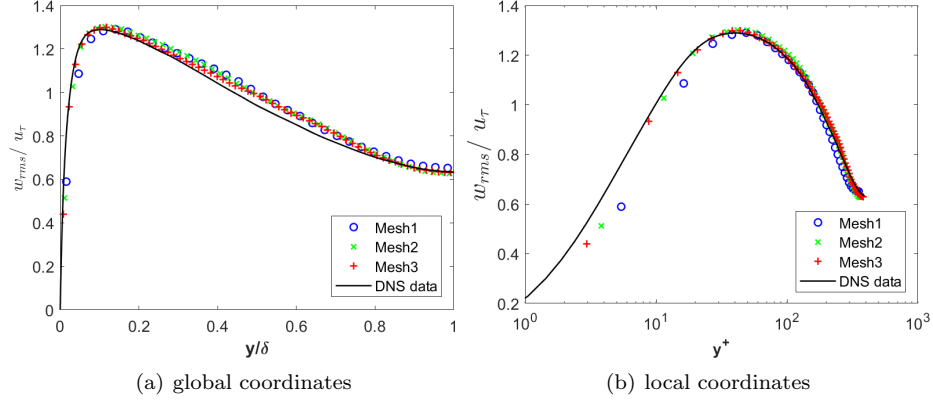
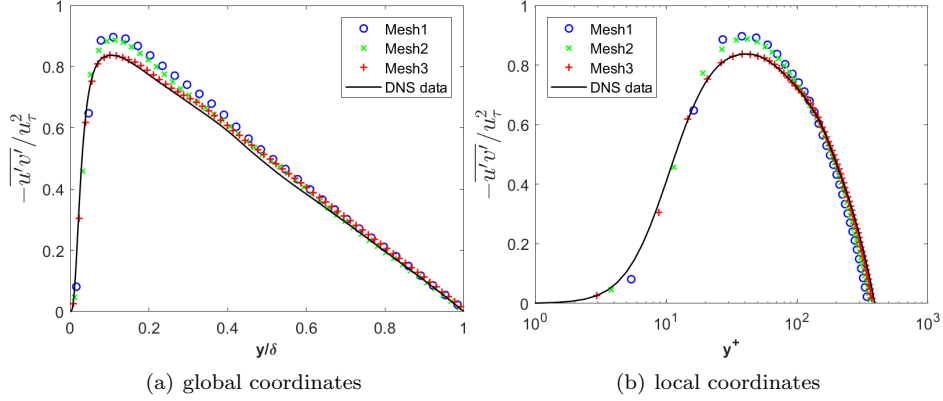


Figure 16: Rms profiles for the wall-normal velocity fluctuation normalised by u_τ

Fig. (17) shows that all the profiles overpredict the location of the peak values and also the peak values. Mesh 3 shows a better agreement with the DNS data, with some overprediction in the log-law layer.

6.2.3 Turbulent shear stress

Fig. (18) shows the turbulent shear stress normalised by u_τ^2 plotted in global and wall coordinates. Mesh 1 and mesh 2 overpredicts the DNS data in the log-law layer and eventually the peak value. Profile for mesh 3 completely collapses on to the DNS data.

Figure 17: Rms profiles for the spanwise velocity fluctuation normalised by u_τ Figure 18: Turbulent shear stress profiles normalised by u_τ

6.3 Comparison UDNS and WALE, $Re_\tau = 395$

In this section a comparison will be made between the UDNS and WALE model results. The purpose of this comparison is to see how accurately does the combination of LB solver and WALE model predict the results in comparison to that of the UDNS results. Simultaneously they will also be compared with the DNS data. Only the comparison for the mesh 3 will be shown in here and the comparison for other resolution will be shown in the Appendix. From the table (8) and (9) it can be seen that u_τ and the resulting Re_τ for the WALE model are smaller in comparison with that of the UDNS.

6.3.1 Mean velocity profile

Fig. (19) and Fig. (20) show the comparison of the mean velocity profiles, normalised by u_τ , with the results of UDNS, WALE and the DNS data in global and wall coordinates. Both the profiles overpredict the DNS data. The profile of WALE shows a slight more overprediction in comparison to the UDNS

profile.

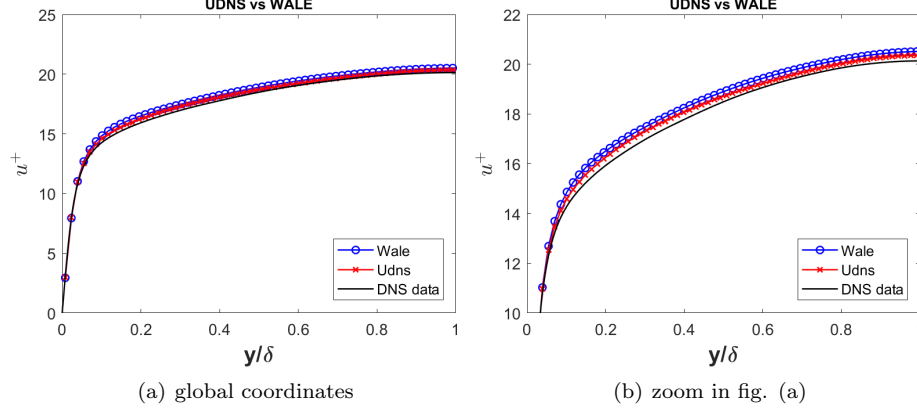


Figure 19: Mean velocity profiles normalised by u_τ

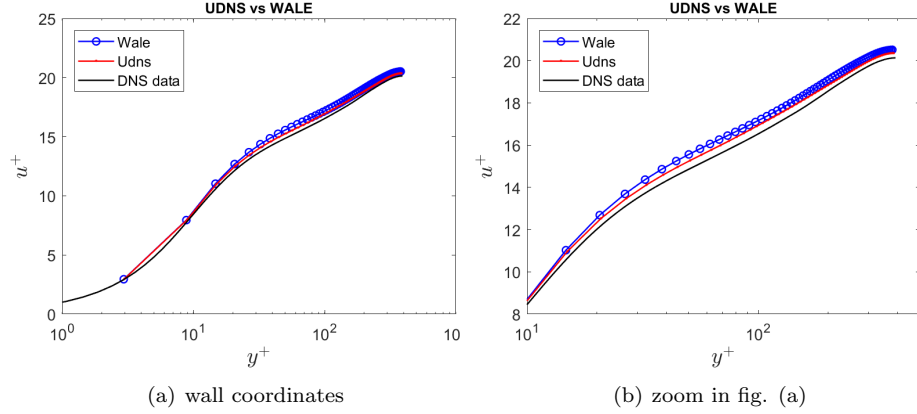


Figure 20: Mean velocity profiles normalised by u_τ

The general explanation for the overprediction is the resulting u_τ being smaller, for WALE & UDNS, in comparison to the DNS data. The same applies to the overpredicting profile of the WALE in comparison to the UDNS profile. Ideally, both profiles, WALE & UDNS, should underpredict the reference profile as we perform the simulations using under resolved mesh. This behaviour can be seen when we plot just the mean velocity and not the normalised mean velocity, u^+ . Fig. 21(a) shows the mean velocity profile in the global coordinates and the Fig. 21(b) shows the zoomed in view of the mean velocity profile. It can be clearly seen that both UDNS & WALE model underpredicts the reference mean centreline velocity, U_c . Profile for the WALE has the smallest centreline velocity. Thus, the comparison between the profiles of UDNS and WALE with each other as well as with the DNS data does not show any drastic difference. A marginal difference is seen between the profiles of the UDNS and WALE.

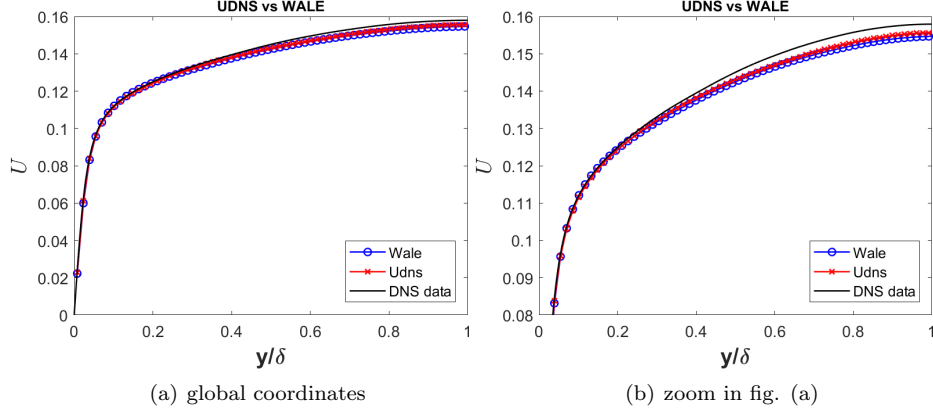


Figure 21: Mean velocity profiles

6.3.2 Turbulence intensities

Fig. (22) shows the comparison of the u_{rms} profiles of UDNS and WALE with each other and also with the profile of the DNS data. Profiles of UDNS and WALE show a good agreement with the DNS data with the profile of WALE slightly overpredicting the peak value in the buffer layer. Apart from the slight overprediction in the buffer layer, the profiles for UDNS and WALE are also in good agreement with each other.

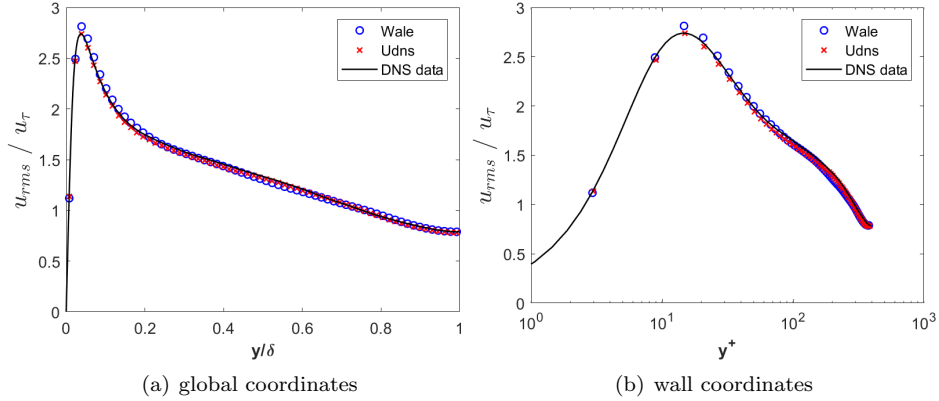
Figure 22: Rms profiles for the streamwise velocity fluctuation normalised by u_τ

Fig. (23) shows the v_{rms} profile of UDNS and WALE with each other and also with the profile of the DNS data. The comparison of the WALE and the UDNS profile shows that the WALE model dampens out the wall normal velocity fluctuations. This damping effect is present till $y^+ = 100$ and after that the profiles are in good agreement with each other. The peak value is underpredicted and the location of the peak value is overpredicted by both the UDNS and WALE profiles in comparison to the DNS data.

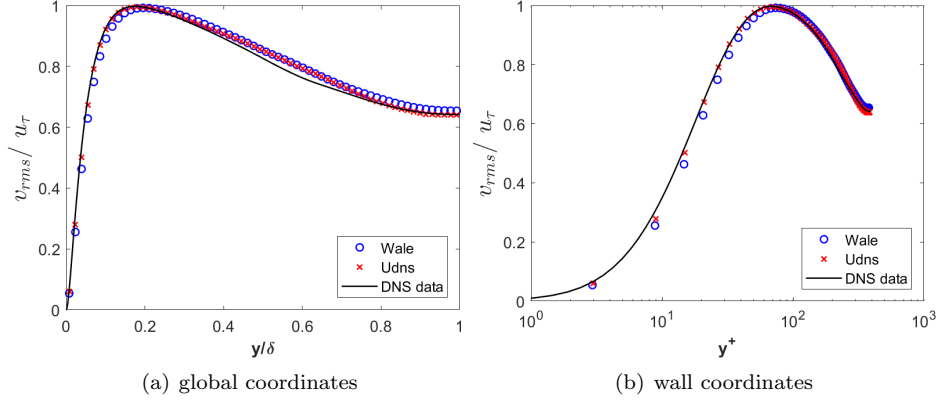


Figure 23: Rms profiles for the wall-normal velocity fluctuation normalised by u_τ

Fig. (24) shows the w_{rms} profile of UDNS and WALE with each other and also with the profile of the DNS data. The WALE model dampens out the spanwise velocity fluctuations in comparison to the UDNS profile. This effect is present till $y^+ = 100$ and after that the profiles are in good agreement. Both profiles overpredict the peak value and the location of the peak value in comparison to the DNS data. This can be seen by the slightly leftwards tilted velocity profiles.

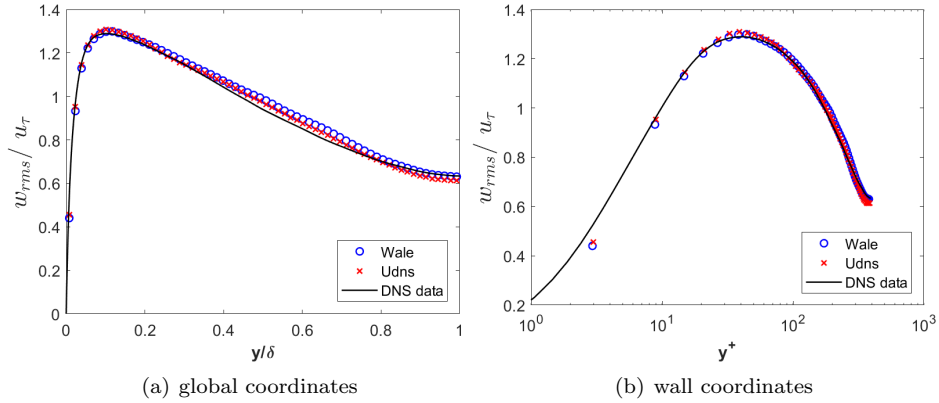


Figure 24: Rms profiles for the spanwise velocity fluctuation normalised by u_τ

Thus, from the comparison of the rms velocity profiles for all components with the UDNS and WALE it is observed that the mean velocity profile for the WALE model overpredicts the UDNS profile $y^+ = 100$ and then they are in good agreement. Also does the WALE profile overpredicts the peak value in comparison to that of the DNS data. With the other two fluctuating velocity components the WALE model dampens out the respective fluctuations till around $y^+ = 100$ in comparison to the UDNS profile. For wall-normal velocity fluctuation both profiles underpredict the peak value and for the spanwise ve-

locity fluctuation both profiles overpredict the peak value. From the comparison of the WALE and the UDNS profiles it seems that the ν_t has a minimal effect.

6.3.3 Turbulent shear stress

Fig. (25) shows the comparison of the turbulent shear stress profiles of WALE and UDNS with each other as well as with the DNS data. WALE and UDNS profiles are in good agreement with each other except for the region between $y^+ = 30$ to $y^+ = 170$ where the UDNS profile underpredicts the WALE and the DNS data profiles. After $y^+ = 170$ the profile fluctuates about the DNS data like a mean. Thus this underprediction can be attributed to the insufficient averaging for that case. Overall profile of WALE is in good agreement with the DNS data.

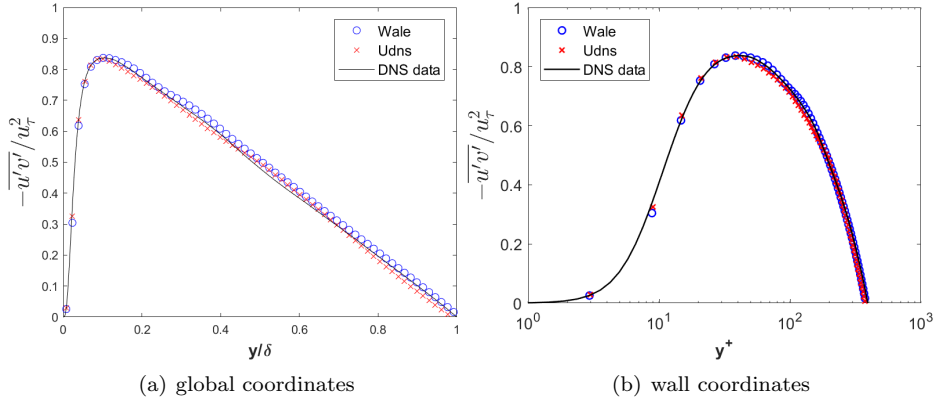


Figure 25: Turbulent shear stress profiles normalised by u_τ^2

6.4 Comparison UDNS and WALE, $Re_\tau = 550$

The comparison of the UDNS and the WALE model profiles for $Re_\tau = 395$ did not show any noteworthy difference. A slight difference, less than 1%, is observed in the mean velocity profile and the Reynolds stress profiles. The initial assumption was that ν_t is not sufficient enough to create a noticeable difference. Thus it was decided to perform a simulation with the higher Reynolds number, $Re_\tau = 550$. The DNS results of [14], for $Re_\tau = 550$, will be used as a reference for the comparison. The DNS data for the comparison can be obtained from the database of turbulent channel flow [12].

The main purpose for performing this simulation is to have some noticeable difference between the WALE and the UDNS results with the increased Reynolds number. The domain size was same as that used for the investigation with $Re_\tau = 395$. The simulation has been performed on the Mesh 1 resolution. The table (10) shows the values of the physical quantities in both S.I and LB units. The Ma number of the flow $Ma_{act} = 0.002915452$ is very small and for the reason mentioned in subsection (5.6) the Ma number is artificially increased to $Ma_{art} = 0.173205081$. The table (Solver settings 550) shows the solver settings.

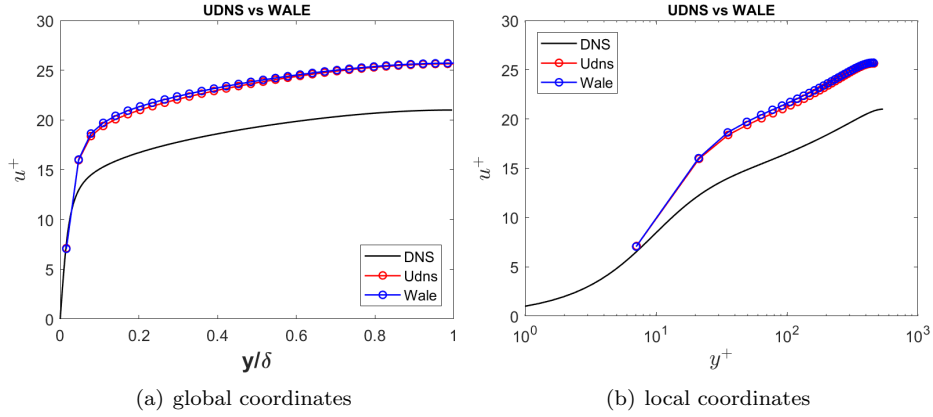
Physical quantity	S.I Units	LB units
Half channel height, δ	1.0 m	$32\Delta x$
Friction velocity, u_τ	0.0543496 m/s	—
Bulk velocity, U_b	1 m/s	$0.1\Delta x/\Delta t$
Kinematic Viscosity, ν	0.0001 m ² /s	$0.00032\Delta x^2/\Delta t$

Table 10: Physical quantities, $Re_\tau = 550$

	$\sim time$	Mesh1
Flow developing stage	$t [s]$ tu_τ/δ tU_b/δ	1562.5 84 1562.5
Averaging stage	$t[s]$ tu_τ/δ tU_b/δ	9062.5 492.54 9062.5

Table 11: Solver settings, $Re_\tau = 550$

Mean velocity profile The resulting quantities for UDNS and WALE have

Figure 26: Mean velocity profiles, $Re_\tau = 550$

listed in the table (12) along with the target values. Fig. (26) shows the mean

Physical quantity	WALE	UDNS	Target
u_τ , m/s	0.0451	0.0455	0.0550
Re_τ	451	455	550

Table 12: Comparison of the computed physical quantities using WALE model & UDNS $Re_\tau = 550$

velocity profiles of UDNS and WALE compared with each other as well as with the DNS results. The profiles for UDNS and the WALE model overpredict the DNS profiles significantly. Noticeable difference is not seen between the UDNS

and WALE profiles. Thus increasing the Reynolds number is also not helpful to see the ν_t influencing the WALE mean velocity profiles. Since no positive outcome is obtained from these profiles, the profiles for the turbulence intensity and the Reynolds stress will not be shown or discussed.

6.5 Conclusion and Outlook

A Appendices

A.1 Initial conditions

The following are the initial conditions, for density and the velocity components resp, used for fully developed channel flow simulations:

$$\rho = 27 U_b^2 \left(\cos(x) \frac{4\pi}{x} + \cos(y) \frac{4\pi}{y} \right) \frac{y}{x} + 27 U_b^2 \left(\cos(x) \frac{4\pi}{x} + \cos(z) \frac{4\pi}{z} \right) \frac{z}{x} \quad (35)$$

$$\begin{aligned} u &= 3 * U_b \left(\frac{y}{h} - 0.5 \left(\frac{y^2}{h^2} \right) \right) \\ v &= U_b * \cos(x) \frac{2\pi}{x} * \sin(y) \frac{2\pi}{y} \\ w &= 0 \end{aligned} \quad (36)$$

A.2 Lattice units

As mentioned earlier the lattice units are the dimensionless lattice values of the corresponding physical parameters. Chapter 7 in [4] provides the detailed explanation on how to non-dimensionalise and several considerations while performing that to the physical units. For the further discussion the lattice units will be marked with an asterisk, *, sign and the unmarked symbols will represent the physical parameters. As mentioned earlier to simulate the same flow, the Reynolds number in both the unit systems has to be equal. Here the Re_b is taken as the reference.

$$\begin{aligned} Re^* &= Re_b \\ \frac{U^* l^*}{\nu^*} &= \frac{U_b l}{\nu} \end{aligned} \quad (37)$$

The total number of grid nodes are same in both unit system, thus :

$$\begin{aligned} N^* &= N \\ \frac{l^*}{\Delta x^*} &= \frac{l}{\Delta x} \\ l^* &= \frac{l}{\Delta x} \Delta x^* \end{aligned} \quad (38)$$

The usage of the uniform grid here implies that $\Delta x^* = 1$. With the same assumption as of Reynolds number the lattice and the physical Mach number must also match and this implies:

$$\begin{aligned} Ma^* &= Ma \\ \frac{U^*}{c_s^*} &= \frac{U_b}{c_s} \\ U^* &= Ma c_s^* \end{aligned} \quad (39)$$

where c_s^* is the lattice speed of sound and equal to $\frac{1}{\sqrt{3}} \approx 0.577$. When simulating the incompressible flow simulations the Mach number is small and hence it is not necessary to equate the Mach number until and unless it is small. Mach number in lattice units is considered small when $Ma^* < 0.3$ [4]. Now l^* and U^* are known the only free parameter ν^* can be computed by using the eq. 37.

As mentioned before the Mach number of the simulation is very small i.e. $Ma \ll 1$, as the flow is incompressible. This means that the U^* will be very small and thus Δt which scales as $\Delta t \propto Ma^*$ will also be very small. The total simulation time, T , is expressed as [4]

$$T \propto \frac{1}{\Delta x^n \Delta t}$$

where n is the spatial dimension. This shows that smaller Δt requires a large number of time-steps for the flow to develop. Hence it is suggested to somehow increase Ma^* and eventually U^* as $U^* \propto Ma^*$. Let us consider an example of Mesh 1 resolution to show how the Mach number was increased.

$$\begin{aligned} U^* &= \frac{Re_b \nu^*}{l^*} \\ Ma^* &= U^* \sqrt{3} \end{aligned} \tag{40}$$

$l^* = 64$ and ν^* is chosen equal to 0.0005 and the resulting $U^* = 0.104296875$. With the increase in the mesh resolution the ν^* also increases to keep the Reynolds number same. Thus the eq. 40 shows how the Ma^* has been increased. The intrinsic restrictions of LB algorithm however puts a limit on the simulation parameters i.e. Ma^* [4]. Thus there is an upper limit till which the Ma^* can be increased. The table 13 shows the parameters in lattice units to increase the Mach number.

	Mesh1	Mesh1.5	Mesh2
U^*	0.104296875	0.104296875	0.104296875
ν^*	0.0005	0.00075	0.001
l^*	64	96	128

Table 13: Parameters in lattice units

A.3 One WALE

The results

References

- [1] Y.A. Cengel and J.M. Cimbala. *Fluid Mechanics: Fundamentals and Applications*. McGraw-Hill, New Delhi, 2 edition, 2011.
- [2] P.K. Kundu and I.M. Cohen. *Fluid Mechanics*. Academic Press, 525 B Streek Suite 1900, San Diego, California 92101-4495, USA, 2002.
- [3] S.B. Pope. *Turbulent Flow*. Cambridge University Press, Cambridge, UK, 2001.
- [4] T. Krüger, H. Kusumaatmaja, A. Kuzmin, O. Shardt, G. Silva, and E.M. Viggen. *The Lattice Boltzmann Method: Principles and Practice*. Springer, Cham, Switzerland, 2016.
- [5] Martin Geier, Andrea Pasquali, and Martin Schönherr. Parametrization of the cumulant lattice boltzmann method for fourth order accurate diffusion part i: derivation and validation. *Journal of Computational Physics*, 348:862–888, 2017.
- [6] M. Weickert, G. Teike, O. Schmidt, and M. Sommerfeld. Investigation of the les wale turbulence model within the lattice boltzmann framework. *Computers and Mathematics with Applications*, 59:2200–2214, 2010.
- [7] Lars Davidson. Fluid mechanics, turbulent flow and turbulence modeling-lecture script.
- [8] F. Nicoud and F. Ducros. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. *Flow, Turbulence and Combustion*, 62:183–200, 1999.
- [9] F.R.S. G. I. Taylor and B.A. A. E. Green. Mechanism of the production of small eddies from large ones. *Proceedings of the Royal Society A Mathematical, Physical and Engineering Sciences*, 158:499–521, 1937.
- [10] Wikipedia. Taylor-green vortex — Wikipedia, the free encyclopedia, 2016. [Online; accessed 17-July-2018].
- [11] Jochen Fröhlich. *Large Eddy Simulationen turbulenter Strömungen*. Teubner/GWV Fachverlage GmbH, Wiesbaden, DE, 2006.
- [12] <http://turbulence.ices.utexas.edu/>.
- [13] J. Kim, P. Moin, and R. Moser. Turbulence statistics in fully developed channel flow at low Reynolds number. *Journal of Fluid Mechanics*, 177:133–166, 1987.
- [14] Myoungkyu Lee and Robert D. Moser. Direct numerical simulation of turbulent channel flow up to $Re_\tau \approx 5200$. *Journal of Fluid Mechanics*, 774:395–415, 2015.
- [15] P.Moin, W.C. Reynolds, and J.H.Ferziger. Large eddy simulation of incompressible turbulent channel flow. Technical report number t-12, Thermosciences Division, Department of Mechanical Engineering, Stanford University, Stanford, California, 1978.

- [16] R.D. Moser, J.D. Kim, and N.N. Mansour. Direct numerical simulation of turbulent channel flow up to $Re_\tau = 590$. *Physics of Fluids*, 11:943–945, 1999.
- [17] R.K. Freitas, A. Henze, M. Meinke, and W. Schröder. Analysis of lattice-boltzmann methods for internal flows. *Computers and Fluids, Elsevier*, 47:115–121, 2011.
- [18] S. Uphoff. *Development and Validation of turbulence models for Lattice Boltzmann schemes*. PhD thesis, TU Braunschweig, Fakultät Architektur, Bauingenieurwesen und Umweltwissenschaften, Braunschweig, Germany, 2012.
- [19] D.J. Besspalko. *Validation of the Lattice Boltzmann Method for Direct Numerical Simulation of Wall-Bounded Turbulent Flows*. PhD thesis, Queen’s University, Department of Mechanical and Materials Engineering, Canada, 2011.
- [20] Eugene de Villiers. *The Potential of Large Eddy Simulation for the Modeling of Wall Bounded Flows*. PhD thesis, Thermofluids Section, Department of Mechanical Engineering, Imperial College of Science, Technology and Medicine, Kensington, London SW7 2AZ, 2006.
- [21] K. Premnath, M.J. Pattinson, and S. Banerjee. Generalized lattice-boltzmann equation with forcing term for computation of wall- bounded turbulent flows. *Computational Physics*, 2009.
- [22] P. Sagaut. *Large Eddy Simulation for Incompressible Flow*. Springer Verlag, 2002.
- [23] Maurizio Quadrio, Bettina Frohnäpfel, and Yosuke Hasegawa. Does the choice of the forcing term affect flow statistics in dns of turbulent channel flow? *Cornell university library*, 47:115–121, 2015.
- [24] K. Premnath, M.J. Pattinson, and S. Banerjee. Dynamic subgrid scale modeling of turbulent flows using lattice-boltzmann method. *Computational Physics*, 2009.
- [25] David Roos Launchbury. *Unsteady Turbulent Flow Modelling and Applications*. Springer Vieweg, Wiesbaden, 2016.
- [26] E. Lenormanda, P. Sagaut, and L. Ta Phuoc. Large eddy simulation of subsonic and supersonic channel flow at moderate reynolds number. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS*, 32:392–406, 2000.
- [27] Javier Jiménez and Parviz Moin. The minimal flow unit in near-wall turbulence. *Journal of Fluid Mechanics*, 225:213–240, 1991.
- [28] Prof. Manfred Krafczyk. personal discussion.
- [29] <https://www.tu-braunschweig.de/it/dienste/21/phoenix>.

- [30] Martin Geier, Martin Schönherr, Andrea Pasquali, and Manfred Krafczyk. The cumulant lattice boltzmann equation in three dimensions: Theory and validation. *Computers and Mathematics with Applications*, 70:507–547, 2015.