OpenFOAM®
Journal

Volume 6 [Educational Papers], Pages 1–24
ISSN: 2753-8168

Large-Eddy Simulation of Turbulent Channel Flow with OpenFOAM

Timofey Mukha[1,*] and Matteo Parsani[1,2]

[1]Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia

[2]Physical Science and Engineering Division, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
Email address: `timofey.mukha@kaust.edu.sa, matteo.parsani@kaust.edu.sa`

DOI: https://doi.org/10.51560/ofj.v6.159
Results with version(s): OpenFOAM v2306
Repository: https://github.com/timofeymukha/openfoam_channel_les

Abstract. This article presents the results of large-eddy simulations (LES) of fully developed turbulent channel flow at $Re_\tau \approx 1000$ using OpenFOAM solvers. The purpose of the work is two-fold. One is to provide publicly available validation results for an important LES benchmark, together with the case setup and codes used for post-processing. The second is to serve as an entry point for those starting out with scale-resolving simulations of turbulence with OpenFOAM. To that end, the exposition of the material is somewhat lengthier and more pedagogical than in a typical scientific paper. The simulations cover a range of grid resolutions to demonstrate the sensitivity of the predictive accuracy to the cell size. Furthermore, the results from the implicit LES are compared with those obtained using the Sigma subgrid-scale model. The post-processing aims to cover most quantities typically considered in the analysis of a turbulent flow, including velocity statistics up to fourth order, fluctuations of vorticity, spatial spectra and two-point correlations, linear spectral coherence, and the budget of the turbulent kinetic energy. To facilitate the analysis, we draw on both built-in OpenFOAM capabilities and community contributions. The results show that upon sufficient resolution of the inner scales, OpenFOAM accurately simulates wall-bounded turbulence, showing excellent agreement with reference data from direct numerical simulations. However, explicit subgrid scale modelling fails to improve the simulations' accuracy.

1. Introduction

A fully developed turbulent channel flow is defined as a flow between two infinite walls, driven by a constant pressure gradient strong enough to transition to the turbulent regime. Since pioneering studies [1–4] this flow has been used for studying wall-bounded turbulence, validating the implementation of scale-resolving simulation approaches, such as large-eddy simulation (LES) and direct numerical simulation (DNS), and benchmarking new turbulence models. The reason why turbulent channel flow is such an attractive case is that while exhibiting the complexity of turbulent boundary layers, it is geometrically very simple and does not require inflow or far-field boundary conditions. To illustrate the rich physics of this flow, a snapshot of a spanwise slice of the velocity magnitude from one of the simulations presented below is given in Fig. 1. The multiscale nature of turbulence is clearly visible, with eddies becoming smaller towards the two walls.

Without aiming for a comprehensive review, we highlight several more recent studies on channel flow, many of which originate from the group of Jiménez. These include investigations into the scaling of velocity spectra [5] and velocity fluctuations [6], the influence of the Reynolds number [7], and the effect of domain size [8], among others [9, 10]. A major milestone was reached in the DNS study by Lee and Moser [11], which achieved a friction Reynolds number of $\mathrm{Re}_\tau \approx 5\,200$ (defined below). We, like many others, use this dataset as a reference in our work. It has also been employed in subsequent studies on wall-bounded turbulence, such as the work of Lee and Moser [12]. More recently, two DNS studies have extended the simulated Reynolds number further, reaching $\mathrm{Re}_\tau \approx 8\,000$ [13] and $\mathrm{Re}_\tau \approx 10\,000$ [14], respectively. This body of works has provided significant insights into the physics of wall-bounded turbulence.

[*] Corresponding author

Received: 18 February 2025, Accepted: 2 January 2026, Published: 15 February 2026

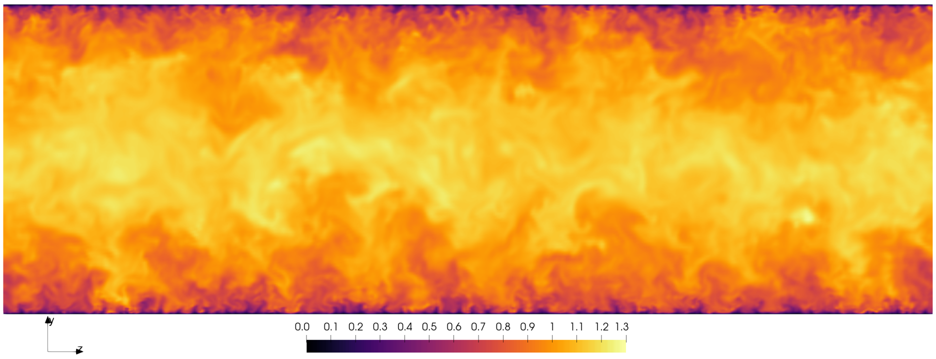

Figure 1. A spanwise slice of the velocity magnitude field, scaled with the bulk velocity (see Eqn. (3)), from the simulation on the M3 mesh (details in Section 3).

Given how important channel flow is as a benchmark, it is not surprising that several previous OpenFOAM-related works presented results for it, for example Refs. [15–18]. This paper also presents a validation study, mainly for OpenFOAM's transient incompressible solver `pimpleFoam`. The main novelty is that we consider a high Reynolds number, $\mathrm{Re}_\tau \approx 1000$, and a rich set of simulation outcomes: single-point statistics of velocity up to fourth order, vorticity fluctuations, spatial two-point correlations and spectra, linear spectral coherence, the Reynolds stress anisotropy map, and the budget of the turbulent kinetic energy. Furthermore, several grid resolutions are considered to provide an intuition for the sensitivity of the results to this important simulation parameter. Additionally, we evaluate the performance of subgrid scale modelling and its connection to numerical dissipation.

Accompanying the paper is a data and code repository containing ready-to-run OpenFOAM cases, the obtained flow statistics for each case, and Python codes used for post-processing. The repository's goal is to facilitate reproducibility and serve as a point of entry for new users of OpenFOAM who would like to use it for scale-resolving simulations. In the same spirit, the text of this article is adjusted to be more accessible to newcomers in the field. An attempt is made to occasionally convey not only "hard facts" but also some subjective experiences of the authors. That being said, familiarity with the basic theory of wall-bounded turbulence is still assumed. The material of Chapter 7 in Pope's textbook [19] would be more than sufficient.

The remainder of the paper is structured as follows. Section 2 presents the computational fluid dynamics (CFD) methods used in the study and also touches on possible modifications for more applied cases. The setup of the channel flow simulations is discussed in Section 3. In Section 4, the results of the simulations are analyzed. Finally, Section 5 gives concluding remarks.

2. Computational fluid dynamics methods

The governing equations for LES of incompressible flow are formulated as

$$\begin{aligned}
&\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j}, \quad (i = 1, 2, 3), \\
&\frac{\partial u_j}{\partial x_j} = 0.
\end{aligned} \tag{1}$$

Here, summation over repeated indices is implied, and i indexes the three spatial coordinates, x_i. The velocity components are denoted with u_i; the density is ρ and the pressure p. For convenience, we also use the triples (x, y, z) and (u, v, w), indicating the streamwise, wall-normal, and spanwise coordinate directions, and the respective components of the velocity vector. The symbol τ_{ij} indicates the total deviatoric stress tensor. The latter is composed of the resolved and subgrid contributions,

$$\tau_{ij} = 2\nu S_{ij} + \tau_{ij}^{\mathrm{sgs}}, \tag{2}$$

where

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

is the strain rate tensor, ν is the kinematic viscosity of the fluid, and τ_{ij}^{sgs} the subgrid stress (SGS) tensor.

The unknowns, u_i and p, are considered to be implicitly spatially filtered by the computational grid. Since this is valid throughout the paper, we do not introduce any additional notation to denote the filtering. In the context of the cell-centered finite volume method used in OpenFOAM, implicit filtering is equivalent to a filter kernel defined as the reciprocal of the local cell volume. Note that while this approach is very common, the consequence is that the definition of the unknowns is coupled to the grid. In other words, u_i and p formally represent a different quantity whenever the grid is changed. So, by definition, grid convergence is impossible for this type of LES. However, as the grid size reduces, one will effectively transition to a direct numerical simulation (DNS).

The subgrid stress tensor τ_{ij}^{sgs} requires modelling. A plethora of models have been proposed through the years, most of them based on the Boussinesq hypothesis, which states that τ_{ij}^{sgs} can be modelled as $2\nu_{\text{sgs}}S_{ij}$, where ν_{sgs} is the subgrid viscosity. Under this assumption, it is possible to lump the subgrid and molecular viscosity stresses into one term. The classical model of this kind is the Smagorinsky model [20], which requires near-wall modification to avoid excessive dissipation. Other algebraic models have recently gained popularity in the engineering community due to their simplicity and low computational overhead. These include the Vreman model [21], the WALE model [22], and the Sigma model [23]. The latter two are available in OpenFOAM, whereas the Vreman model is not. Here, we select the Sigma model for the studies on the influence of SGS modelling. The eddy viscosity field produced by this model satisfies four important principles. It is always positive; it exhibits cubic behaviour towards the wall; it is zero for two-component or two-dimensional flows; it is zero for axisymmetric flows and isotropic expansion or contraction.

As an alternative to modelling the subgrid stress tensor, one can simply ignore it. The key consideration here is the relationship between the subgrid scale viscosity and the numerical one. One can argue that introducing ν_{sgs} is pointless when its contribution is anyway smaller than the numerical diffusion. Komen *et al.* [24] show that for OpenFOAM, this is a common scenario. Not having an SGS model is sometimes referred to as implicit LES. However, originally that term implied a specific selection of numerical schemes such that the truncation error would fit for the job of modelling the subgrid stresses, see the article of Fureby and Grinstein [25] and references therein. In the current work, most simulations are performed with no explicit SGS modelling. However, in Section 4.6, we compare the results from implicit LES with a simulation with the Sigma model.

We now consider the various choices for discretizing Eqn. (1) in OpenFOAM and solving the resulting system of equations. Generally, the choice of numerical schemes is dictated by the balance between the desire to achieve the best accuracy and the necessity to introduce a certain amount of numerical diffusion to keep the simulation stable. Here, the channel flow simulations are performed on fine orthogonal hexahedral grids and a time-step that holds the maximum Courant–Friedrichs–Lewy (CFL) number [26] below 0.5 (see Appendix B for an analysis of the effect of the timestep size). Consequently, it is possible to aim for best accuracy without compromising the stability. In particular, we use linear interpolation, `Gauss linear`, for convection, diffusion, and gradient terms, which is second-order accurate. No limiters are applied. For the diffusion term, additional corrections are typically necessary to account for grid non-orthogonality [27], but in our case we can omit them (use `orthogonal` surface-normal gradient schemes). The second-order backward-differencing scheme, `backward`, is used for time-integration, which is generally a very stable implicit scheme and a good first choice in OpenFOAM when second order is desired.

We note that for convective fluxes, an alternative to the linear scheme is to use midpoint interpolation. This is no longer second-order accurate on stretched grids, but possesses the attractive property of preserving the kinetic energy of the flow in the inviscid limit [18].

We use a segregated method to solve the governing equations, employing the PISO algorithm [28], as implemented in `pimpleFoam`, when the so-called outer correction loop is not in use. At each time-step, 3 iterations of the pressure correction loop are performed. This choice is, perhaps, conservative as PISO is often defined as an algorithm with 2 correction steps, following the original article by Issa [28]. However, we are not aware of any studies that carefully measure the effect of this parameter on the accuracy of LES.

Considering the choice of the algorithms indicated above, two linear solvers should be selected: one for the velocity components and one for the pressure. Of the two, the pressure solver plays a much bigger role in the runtime of the simulation. The bi-conjugate gradient solver, equipped with a diagonal-based incomplete LU (DILU) preconditioner, can be used for the velocity. Another valid choice would be OpenFOAM's `smoothSolver`. For pressure, we consider two options: the multigrid solver called `GAMG` and a preconditioned conjugate gradient solver, `PCG`, coupled with a diagonal-based incomplete Cholesky preconditioner (DIC). The latter is equivalent to DILU for the case of symmetric matrices. It is necessary to consider both the theoretical properties of the solver's algorithms and its parallel efficiency. Some recent benchmark data comparing different OpenFOAM solvers can be found in the work of Bnà *et al.* [29]. Essentially, `GAMG` outperforms `PCG-DIC` at lower core counts, with parity achieved somewhere between 1000 and 2000 cores. This data is in line with the authors' experience. However, the multigrid solvers offer a vast number of tunable parameters that could, in principle, improve its performance. In particular, the importance of the processor agglomeration algorithm has been highlighted in a talk by T. Holzmann at the 18th OpenFOAM Workshop. Unfortunately, those details were not published. Finally, we also point out the availability of bindings to the PETSc library, which offers a vast set of linear solvers. The multigrid solver, in particular, has demonstrated excellent scaling and performance [29]. The bindings are distributed as a submodule inside the OpenFOAM code base, and we encourage the reader to consider them for their simulations. In general, we recommend the `PCG-DIC` pair as the initial choice for any simulation, as it requires little tuning, scales well, and is robust enough to tackle poorer grids, initial conditions, etc. Once the simulation is running, the alternatives discussed above can be explored.

3. Fully developed turbulent channel flow

A fully developed incompressible turbulent channel flow is completely defined by the value of the Reynolds number, which should be constructed based on a length and velocity reference scale. For the former, the channel half-height, δ, is commonly used, although the full height is sometimes used instead. For the velocity, one can consider the bulk velocity, U_b, or the friction velocity, u_τ. Other choices, like the centreline velocity, are also possible but are seldom used in practice.

At this point, we introduce the overline operator, denoting a combination of temporal and spatial averaging in the *wall-parallel* directions (x and z). The latter is always used for channel flow since the streamwise and spanwise directions are statistically homogeneous, which means we can enhance the accuracy of the temporal statistics by averaging them across the x-z plane.

The bulk velocity is defined as the temporal and spatial average of the velocity field across the entire channel. In other words, it is the integral of $\overline{u}$ across the channel height,

$$U_b = \frac{1}{\delta}\int_0^\delta \bar{u}\mathrm{d}y. \tag{3}$$

The friction velocity, u_τ is constructed from the mean magnitude of the wall shear stress, $\overline{\tau}_w$, as

$$u_\tau = \sqrt{\overline{\tau}_w/\rho} = \sqrt{\nu\frac{\mathrm{d}\overline{u}}{\mathrm{d}y}}. \tag{4}$$

The corresponding Reynolds numbers are defined as $\mathrm{Re}_b = U_b\delta/\nu$ and $\mathrm{Re}_\tau = u_\tau\delta/\nu$. It is important to recognize that Re_τ can be written as the ratio of the outer and inner length scales of the flow: δ/δ_ν, where $\delta_\nu = \nu/u_\tau$ is the inner (viscous) length-scale. Another useful observation is that Re_τ is equal to the value of $y^+ = y/\delta_\nu$ at the center of the channel. The plus superscript will be used to denote inner scaling of all quantities in the remainder of the paper.

Both Re_b and Re_τ can be used to define the same flow, but the exact functional relationship between these two Reynolds numbers is unknown. To set Re_b, one needs to enforce U_b during the simulation. This task is typically done using a simple controller, which adjusts the strength of the source term, driving the flow at run time such that the desired U_b is maintained. In OpenFOAM, this functionality is readily available as an `fvOption` called `meanVelocityForce`.

Physically, the source term is interpreted as the mean pressure gradient driving the channel flow. Its strength can be computed directly [30] as

$$-\frac{\mathrm{d}\bar{p}}{\mathrm{d}x} = \frac{\overline{\tau}_w}{\delta}. \tag{5}$$

This equation represents the mean momentum balance in the channel. Note that Eqn. (4) and Eqn. (5) provide an explicit formula for the magnitude of the mean pressure gradient necessary to obtain a chosen u_τ. In OpenFOAM, the source can be set using the `fvOption` called `vectorSemiImplicitSource`.

When one of the two Reynolds numbers is set to be exact, the other will be an outcome of the simulation. So, if Re_b is set, there will generally be some error in Re_τ and thus the velocity gradient at the walls. Conversely, for a set Re_τ that gradient will be exact, but some error in U_b can be expected.

An important practical consideration is that setting Re_b usually leads to much shorter transient times necessary to achieve fully developed turbulence. For this reason, this approach is employed here. In particular, we consider channel flow at $\mathrm{Re}_b = 20\,000$, for which high-quality DNS data is provided by Lee and Moser [11]. This latter reference shows that this corresponds to $\mathrm{Re}_\tau \approx 1000$.

Having defined the physical conditions to be simulated, it is necessary to choose the size of the computational domain. Since the streamwise and spanwise directions are statistically homogeneous, the domain dimensions L_x and L_z should be chosen according to the size of the turbulent structures present in the flow. The largest such structures are $\sim \delta$, which means L_x and L_z should be some multiple of that.

Statistically, the aim is to have the flow completely decorrelated across the dimensions of the domain. Otherwise, a spurious correlation would be introduced by the computational setup, thus contaminating the result. It should be noted that since spatial averaging is used to improve the convergence of the statistics, a larger domain, in principle, allows smaller physical simulation times. This introduces a correction to the extra computational effort required to run using a larger domain.

An in-depth analysis of the influence of the size of the domain on channel flow simulation outcomes has been performed in the work of Lozano-Durán *et al.* [8]. The outcome is that $L_x = 2\pi\delta$ and $L_z = \pi\delta$ are sufficient to reproduce single-point statistics faithfully. However, in our experience, this necessitates very long averaging times to obtain convergence. Moreover, we aim to capture two-point statistics, such as energy spectra. Therefore, here we set $L_x = 8\delta$ and $L_z = 6\delta$, exceeding the bare minimum requirements.

We now turn to boundary conditions. Periodic boundary conditions (the type `cyclic` in OpenFOAM) are applied in the homogeneous directions. At the walls, the velocity is set to zero per the no-slip condition, whereas for pressure, the value of the gradient is set to zero.

For the flow in the channel to transition to turbulence, it is necessary to introduce some perturbation in the initial conditions that would trigger the transition. When using the linear scheme for convective fluxes, accompanying numerical oscillations are sometimes sufficient to do the job, particularly at higher Re-numbers where the flow is very unstable. Nevertheless, it is better not to start from a uniform velocity profile. Here, we set the initial velocity field as

$$\begin{aligned}
u &= \frac{u_\tau}{\kappa}\log(y^+ + 1) + u_\tau B + \varepsilon k_z \sin(k_x x)\cos(k_z z),\\
v &= \varepsilon \sin(k_x x)\sin(k_z z),\\
w &= \varepsilon k_x \cos(k_x x)\sin(k_z z).
\end{aligned} \tag{6}$$

Here, $\varepsilon = 0.03$, $k_x = 8$, $k_z = 9$, $\kappa = 0.41$, $B = 5.5$. The first three parameters and the form of the harmonics in Eqn. (6) are not carefully selected, and the expression, in general, is not tuned in any way (the resulting field is not divergence-free, for example). It was simply observed that this harmonic perturbation is sufficient to trigger the transition. The expression for the streamwise component is augmented with the classical log-law of the wall to mimic somewhat the mean velocity profile. The value of u_τ used here is based on the expected Re_τ.

Note that once one has conducted a channel flow simulation and needs to run a new one (on a different grid, for example), an excellent idea is to map the instantaneous velocity from the old simulation to use as initial conditions. In OpenFOAM, the `mapFields` utility can be used to that end. In this work, for consistency, we initialize all the simulations using Eqn. (6).

An important task is to identify the time to start collecting the temporal statistics, which should be based on monitoring the development of the transient associated with the initial conditions. Here, we use the spatially-averaged value of u_τ on one of the channel walls. This is easily set up using a combination of the `yPlus` function object, which computes the y^+ value at the first off-wall cell center, and the `surfaceFieldValue` function object, which will compute the spatial average on the wall. Knowing ν and the distance to the off-wall cell center, we can easily recover the signal for u_τ during post-processing. An alternative could be to use the volume-averaged kinetic energy of the flow.

As usual, one should also consider the relevant characteristic scales. In particular, time may be non-dimensionalized using the eddy turnover time (ETT), δ/u_τ. We may expect that at least a few ETTs will be necessary to eliminate the transient. Similarly, we should relate the time span used for averaging to the ETT. The necessary span depends on the order of the collected statistical moments—as higher order moments converge slower—and also on the domain size, as discussed above. We average across 25 ETT, which should be sufficient for at least first- and second-order moments, based on the literature (see, for example, Table 1 in Lozano-Durán *et al.* [8]).

A typical evolution of the spatial average of the instantaneous u_τ is shown in Fig. 2. (Unlike in Eqn. (4), we only average the velocity gradient across the x-z plane to get the time-local friction velocity and see how that develops in time.) The first 12.5 ETTs are treated as the transient, whereas the rest of the signal, right of the vertical dashed line, indicates the averaging window. It is clear that towards the end of the simulation, the value of average u_τ is stabilized. To perform the temporal averaging in OpenFOAM, the `fieldAverage` function object is used. To further average spatially over x and z, the `postChannelFlow`[1] utility is employed, which is an improved version of the built-in `postChannel`.

It should be noted that the method employed here to figure out the transient's length relies, to some extent, on knowing the ETT *a priori*. In such cases, a reasonable guess is readily available, and one can run the simulation for that duration and verify the estimate by examining a diagnostic quantity, like

[1] Available at https://github.com/timofeymukha/postChannelFlow

u_τ. Afterward, the collection of statistics is commenced. A more general approach involves determining the length of the transient *a posteriori*. For that, the averaging is started immediately, but is restarted periodically after some fixed time interval. These window-averaged quantities are saved to disk at the end of each interval, forming a series of outputs for the statistics—commonly referred to as batches. After the simulation is finished, one can evaluate the transient in the same manner as before, and simply remove the batches that are considered contaminated by the transient. Afterward, one can average across the remaining batches to get the final statistics. OpenFOAM does support window-averaging in the `fieldAverage` function object. The only downsides of this approach are the increased storage requirements and the additional post-processing step.

We also note that using exactly 12.5 ETTs as the end of the transient is essentially an ad hoc choice based on a visual inspection of the u_τ trend (Fig. 2). Similarly, using 25 ETTs for collecting the statistics is selected based on prior studies, which is not possible if a new flow is considered. A better alternative is to rely on mathematical estimators for the errors in the sampled statistics. We refer the reader to thesis of Xavier [31] for an overview of relevant works, e.g. Refs. [32–34]. In the recent article by Rezaeiravesh *et al.* [35] the authors demonstrate how the error estimation can be done *in situ*, that is, during the simulation runtime.

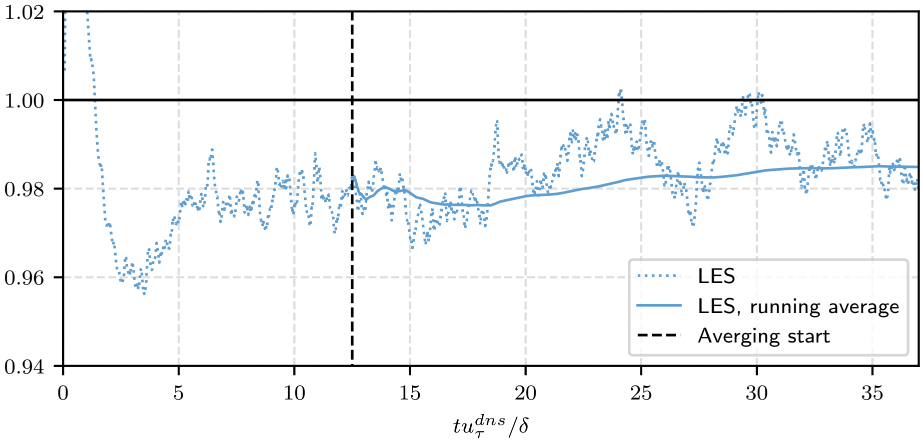

Figure 2. Dotted blue line: The time-evolution of the instantaneous friction velocity averaged across the bottom wall of the channel, normalized by the reference DNS value, u_τ^{dns}. Data from the simulation on the M3 grid. Solid blue line: the running temporal average of the data shown by the dotted line. The dashed vertical line shows the start time of the averaging. The value of u_τ^{dns} is also used to scale the abscissa.

While `fieldAverage` can be used for single-point statistics, there is no convenient way to collect two-point statistics on the fly, so this has to be done *a posteriori*. The `probes` function object is used to sample the velocity and pressure from the entire half-plane in z-y, spanning from 0 to δ in y. For the largest grid (see below), this amounts to 1000×116 probes. The sampling is set to occur every fifth time-step to reduce the file sizes. Despite that, the probe file is around 650 GB for the largest case. To reduce the file size and make it possible to lazy-load the data into RAM memory, the ASCII files produced by OpenFOAM are converted to the HDF5 format, using the functionality in the Python library `ofreaders`[2]. This reduces the size by approximately a factor 5, i.e. 130 GB. This is, unfortunately, still too large to share in the Git repository accompanying this article. We should also note that the half-plane-based statistics are, by definition, not as well converged as the ones where the averaging is applied in both x and z. The difference is examined in Appendix A and is shown to be negligible for second-order statistics.

The remaining aspect of the computational setup to be discussed is meshing. The best practices for LES meshes of boundary layer flows are generally well established. The guidelines are based on the knowledge of the characteristic sizes of turbulent structures populating the near-wall region (see Section 4.4). The smallest energetic eddies are present in the inner layer; therefore, their resolution drives the total mesh size. Consequently, it is common to reason about the mesh size in terms of viscous units, i.e., Δx^+, Δy^+, and Δz^+. For the wall-parallel resolution in a DNS, one would typically aim for $\Delta x^+ = 10$, $\Delta z^+ = 5$ or less. For example, for the reference DNS used here [11] the values are $\Delta x^+ = 10.9$, $\Delta z^+ = 4.6$. For

[2] Available at https://gitlab.com/chalmers-marine-technology/ofreaders

LES, $\Delta x^+ = 25$, $\Delta z^+ = 10$, and even somewhat larger grids should be acceptable, although it will be shown below that the accuracy of second-order statistics deteriorates quite fast. The cost of LES of wall-bounded flows is thus not that significantly smaller than DNS. Results from OpenFOAM channel flow simulations on a range of grid sizes can be found in, e.g., Refs. [17, 24, 36, 37]. If forced to choose whether to compromise on the spanwise or the streamwise resolution, it appears more profitable to refine in z; see, for example, results in the thesis of Mukha [37]. But overall, the convergence pattern of flow quantities with refinement is difficult to predict [38, 39].

For the wall-normal direction, the grid size distribution is a function of the distance from the wall. In the bulk of the channel, the eddies are of sizes $\sim \delta$, and the mean velocity profile is quite flat. On the other hand, in the direct proximity of the wall, strong velocity gradients are present, requiring $\Delta y^+ \leq 1$ to resolve them correctly. Coupling these regions is the log-layer, where the size of the eddies grows as $\sim y$. This is a very basic description, but it is sufficient to justify the commonly used meshing approaches, which all bias the resolution towards the wall.

Here, 3 grids are used, referred to as M1, M2, and M3. A summary of the mesh sizes can be found in Tab. 1. For M1, $\Delta x^+ = 40$, $\Delta z^+ = 24$, and for M2 and M3, these sizes are reduced by a factor of, respectively, 2 and 4. Thus, M3 is close to DNS resolution in the wall-parallel direction. For the wall-normal resolution, we use the approach developed by Rezaeiravesh and Liefvendahl [40]. The resulting distribution for M1 is shown in Fig. 3. A region of constant resolution is observed near the wall, with $\Delta y^+ \approx 0.8$. In the bulk of the channel, the resolution is also constant, measuring $\Delta y^+ \approx 36$, which corresponds to $\delta/25$. Connecting these two regions is a block of cells for which a geometric expansion factor is applied. For grids M2 and M3, the wall resolution of grid M1 is refined by a factor of 2. As a result, M3 lacks an additional wall-normal refinement to bring the grid to the level of a DNS in all three spatial directions. Unfortunately, this would put the simulation cost outside the available core-hour budget.

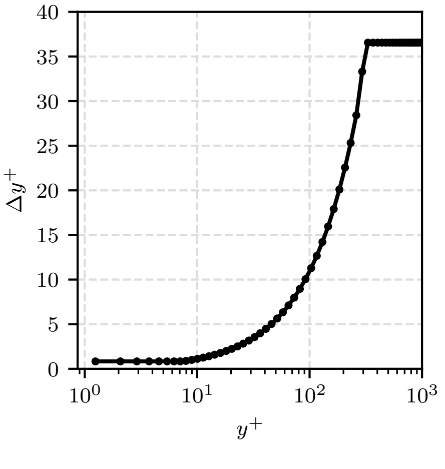

Figure 3. The wall-normal mesh size distribution for mesh M1.

Tab. 1. Computational meshes. Size in inner units computed using the DNS [11] value of δ_ν, for consistency.

Mesh	N_x	N_y	N_z	$N_{tot} \cdot 10^{-6}$	Δx^+	Δz^+	$\max(\Delta y^+)$
M1	200	115	250	5.8	40	24	36
M2	400	232	500	46.4	20	12	18
M3	800	232	1000	185.6	10	6	18

4. **Results**

4.1. **Effect of grid resolution.** We begin the analysis by considering single-point statistics of velocity and vorticity from simulations M1, M2, and M3. Here and below, we compare our results with two DNS datasets. The main reference is the data of Lee and Moser [11]. However, it does not provide the values of skewness and flatness. Therefore, we use an alternative dataset [41] produced using the pseudo-spectral code SIMSON [42].

The mean streamwise velocity is shown in Fig. 4. In outer scaling, the results are difficult to distinguish, and all three curves overlap quite well with the DNS data. This is typical when Re_b, and thus the correct volumetric flow rate, is enforced. Even on very coarse grids, the mean velocity may look accurate in outer scaling in this setup. The plot in inner scaling reveals more differences among the results on different meshes. One should be mindful that errors in u_τ are propagated into the profiles in this scaling. For example, here we see that the M1 curve appreciably over-predicts the DNS. Looking at the computed errors in u_τ listed in the first 3 rows of Tab. 2, we see that for M1, the value is under-predicted, which explains the behavior of $\bar{u}^+$. For the other two simulations, the error in the friction velocity is much smaller, so the inner-scaled velocity profile becomes more accurate. The change from M2 to M3 is quite small. For both profiles, the errors are mainly visible close to the centreline of the channel.

Tab. 2. Relative error in the friction velocity.

Simulation	**Relative error**
M1, `pimpleFoam`, Implicit	-6.72 %
M2, `pimpleFoam`, Implicit	-1.68 %
M3, `pimpleFoam`, Implicit	-1.39 %
M1, `RKSymFoam`, Implicit	-3.91 %
M2, `RKSymFoam`, Implicit	0.2 %
M1, `pimpleFoam`, Sigma model	-7.58 %
M2, `pimpleFoam`, Sigma model	-2.34 %
M1, `RKSymFoam`, Sigma model	-4.86 %

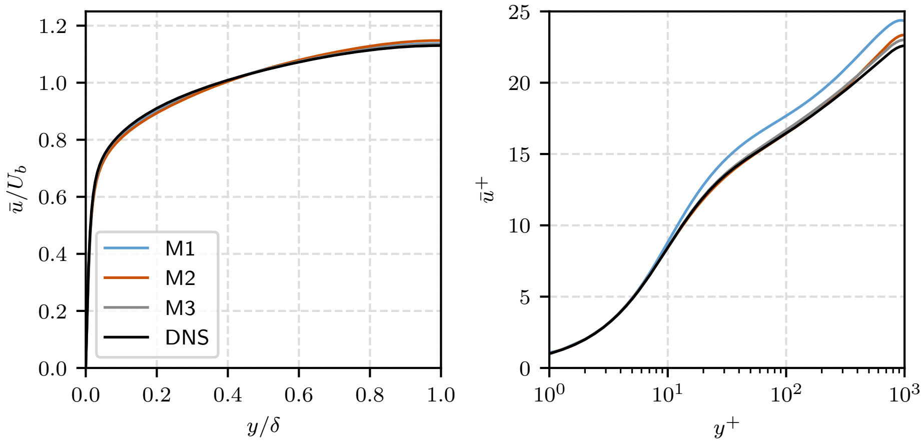

Figure 4. Mean streamwise velocity profiles in outer (left plot) and inner (right plot) scaling.

Next, the Reynolds stresses are considered, see the top row of Fig. 5. The profiles are shown in inner scaling since it is more sensitive, as discussed above. As expected from the analysis of mean velocity, the results on grid M1 are not accurate, except the shear stress $\overline{u'v'}^+$. The error structure for M1 is very typical of under-resolved LES; see, for example, the conclusions in Komen *et al.* [24]. The peak in $\overline{u'u'}^+$ is over-predicted, whereas the values in the center of the channel are under-predicted. The stress values are too small for $\overline{v'v'}^+$ and $\overline{w'w'}^+$. The underlying reason is the under-prediction of the pressure strain rate term in the evolution equations of the normal stresses, which is responsible for redistributing the energy between them [24].

What differs from previous observations regarding $\bar{u}^+$ is the clear increase in accuracy from M2 to M3. Indeed, for M3, the agreement with DNS is excellent. But there is some degree of error cancellation at play. Recall that the error in u_τ pushes the profiles up, and all the M3 curves still slightly under-predict the DNS. The largest error is for the $\overline{v'v'}^+$ component, which is perhaps to be expected since the mesh in this direction is not refined with respect to M2. For completeness, we also show the profiles of the turbulent kinetic energy, $k = 0.5 \sum_i \overline{u_i' u_i'}$ (Fig. 5, bottom row).

The mean velocity and Reynolds stresses constitute the basic quantities to be analyzed when studying a turbulent flow with LES. As the next step, we consider the second-order statistics of vorticity. This quantity relies on the accurate computation of the velocity gradients. This makes its prediction a more

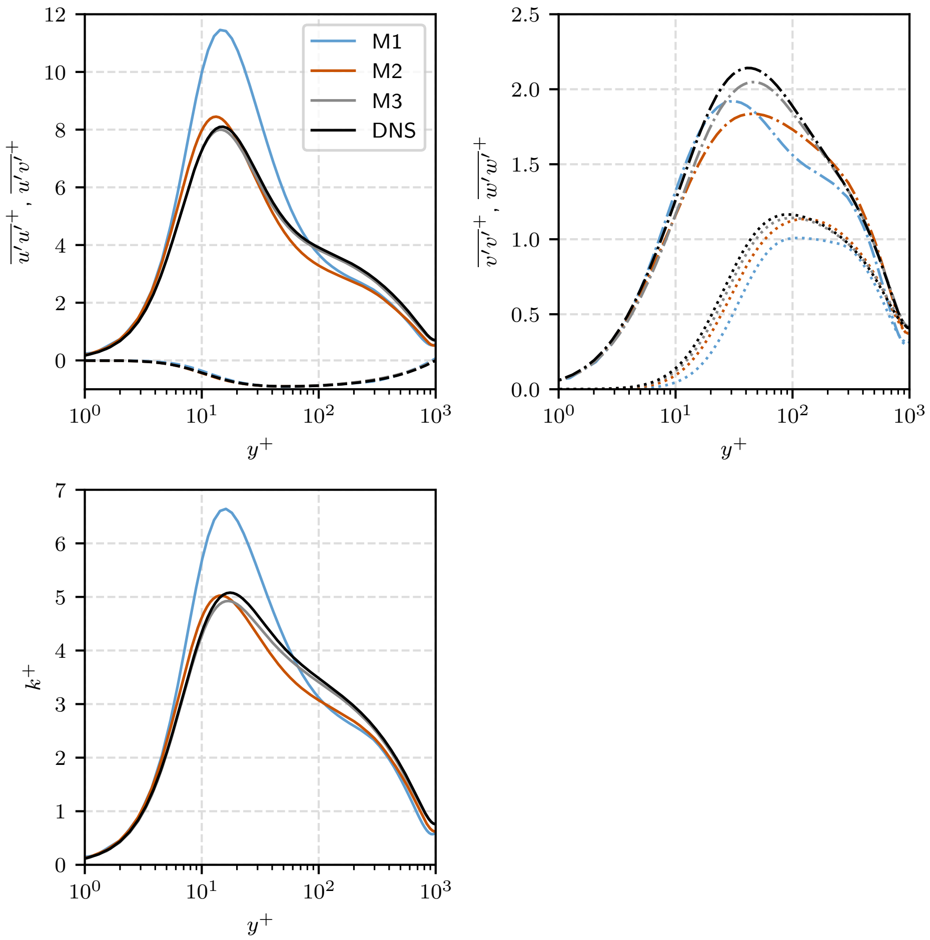

Figure 5. *Top row:* Reynolds stress profiles in inner scaling. Solid lines: $\overline{u'u'}^+$; dashed lines: $\overline{u'v'}^+$; dotted lines: $\overline{v'v'}^+$, dash-dotted lines: $\overline{w'w'}^+$. *Bottom row:* The turbulent kinetic energy profiles in inner scaling.

difficult task, and typically, the results for vorticity are somewhat less accurate than for velocity. This holds for the curves presented in Fig. 6. However, for the M3 mesh the results are still very good for all components but the $\overline{\omega'_x\omega'_x}^+$, for which the values are under-predicted in the buffer region.

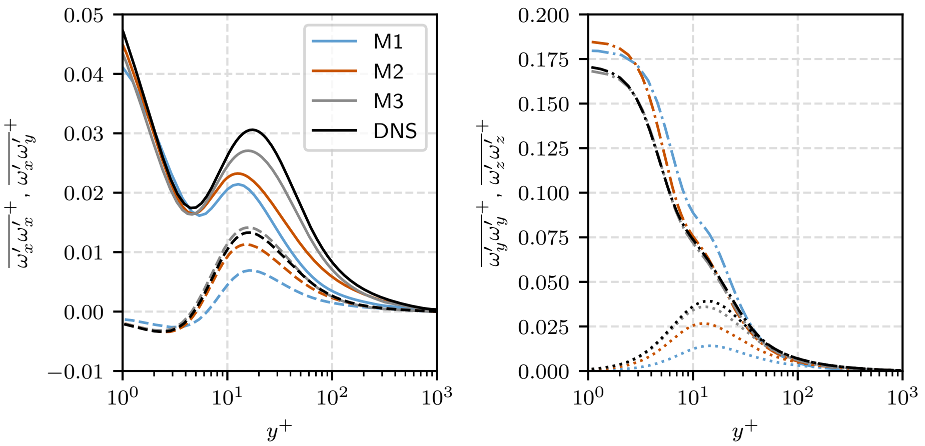

Figure 6. (Co-)variances of vorticity components. Solid lines: $\overline{\omega'_x\omega'_x}^+$; dashed lines: $\overline{\omega'_x\omega'_y}^+$; dotted lines: $\overline{\omega'_y\omega'_y}^+$; dash-dotted lines: $\overline{\omega'_z\omega'_z}^+$.

We continue the analysis with the consideration of higher-order statistics of velocity, namely skewness and flatness (kurtosis), i.e. the third and fourth standardized moments defined as

$$S_\phi = \frac{\overline{\phi'^3}}{\overline{\phi'^2}^{3/2}}, \quad F_\phi = \frac{\overline{\phi'^4}}{\overline{\phi'^2}^{2}}, \tag{7}$$

where ϕ is some quantity. Skewness measures the asymmetry of the probability density function (PDF), with positive skewness indicating a longer right tail and the opposite for negative skewness. Flatness measures the heaviness of the tails, meaning that high values of F_ϕ indicate a larger contribution of outliers.

Recall that we use the in-plane sampled probe data to compute these statistics, meaning that they are less converged than the profiles of the other quantities in this section. In combination with the fact that higher-order statistics take a longer time to converge in general, the averaging errors cannot be considered negligible. For channel flow, the spanwise component of the skewness, S_w, should be zero-valued, so its profile indicates how well the third-order moments are converged. The corresponding curves are in the right plot of Fig. 7, and the spread around zero is appreciable. Nevertheless, we see that for mesh M3, the agreement for S_u is excellent, and for S_v, the deviation is mainly restricted to the log-law region. The effect of mesh resolution is very strong, with a clear bump in accuracy with every refinement.

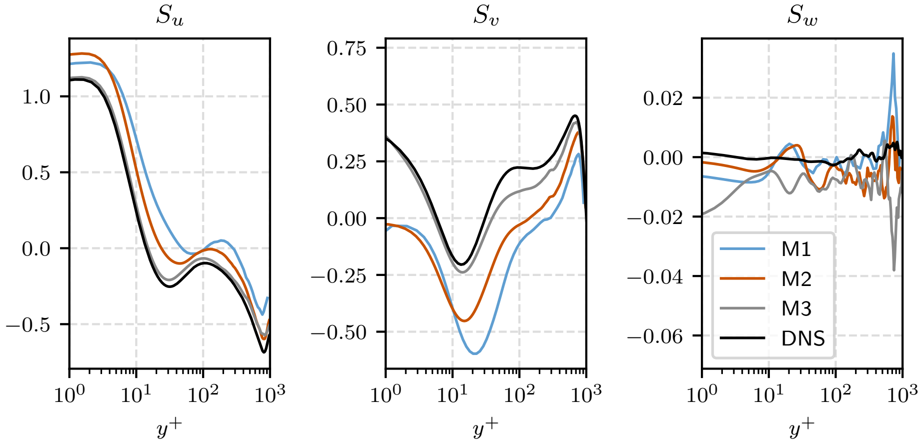

Figure 7. The profiles of skewness for the three velocity components.

The flatness values are shown in Fig. 8. The profiles for F_v and F_w demonstrate the difficulty of predicting extreme near-wall events, a study of which can be found in Lenaers *et al.* [43]. For F_w, on mesh M3, the LES comes close to DNS data, but for F_v, the resolution remains insufficient, and the rare high-v' events are not fully captured. Away from the wall, both quantities are well-predicted on all meshes. Note that the value of 3, which they approach, corresponds to the Gaussian distribution. The situation is somewhat reversed for the streamwise component: the agreement with DNS is better close to the wall.

To summarize the grid refinement study, it is clear that the M1 grid resolution is not sufficient to capture the flow physics faithfully. The wall friction, a critical parameter for many applications, is not captured well, with ≈ 7% error in the friction velocity (and thus nearly 13% for the skin friction coefficient). Using the M2 grid, the results improve significantly, with the mean flow well captured and the Reynolds stresses arguably well-represented. Recall that M2 has wall-parallel resolution $\Delta x^+ = 20$, $\Delta z^+ = 12$, which, according to our results, can be considered the upper bound for getting good predictive accuracy. The results on the M3 grid match the DNS very well for an LES. Certain deviations are observed for the vorticity fluctuations and the high-order statistics (the latter possibly due to lack of convergence). However, the results provide strong evidence that using a DNS grid (which would be M3 refined by a factor of ≈ 2 in the wall-normal direction) and a larger domain size, the agreement for the quantities analyzed in this section would be on the same level of discrepancy as among the existing DNS databases.

4.2. **Turbulent kinetic energy budget.** The analysis proceeds with considering the term budget of the turbulent kinetic energy, k, concentrating on the results obtained with mesh M3. For the derivation of the transport equation for k, the reader is referred to, for example, Chapter 7 in Pope's textbook [19]. The terms can be split into three categories: the production of k by the mean velocity gradients, its dissipation to internal energy, and transport terms that redistribute the energy in space. The latter

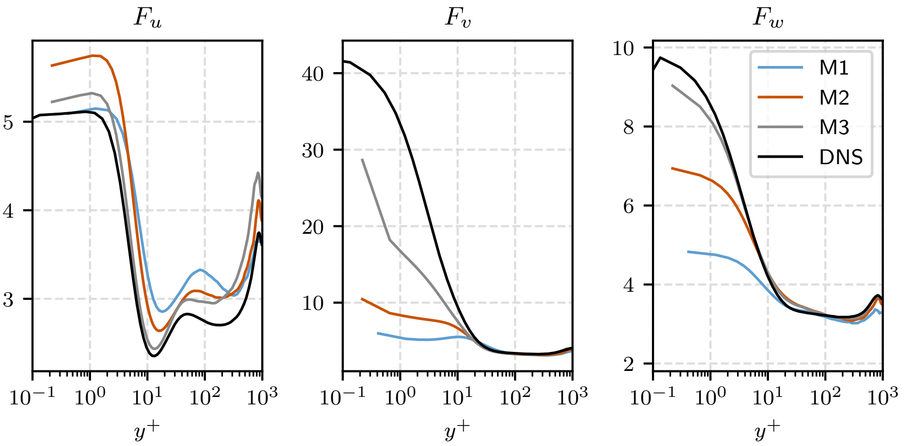

Figure 8. The profiles of flatness for the three velocity components.

category consists of three terms: viscous transport, turbulent transport, and pressure transport. To compute these terms, we use the program `runTimeChannelBudgets` developed by J. Hopman [44]. This software provides a function object to compute the budget terms in the course of the simulation, and an executable utility to post-process them into one-dimensional profiles across the channel.

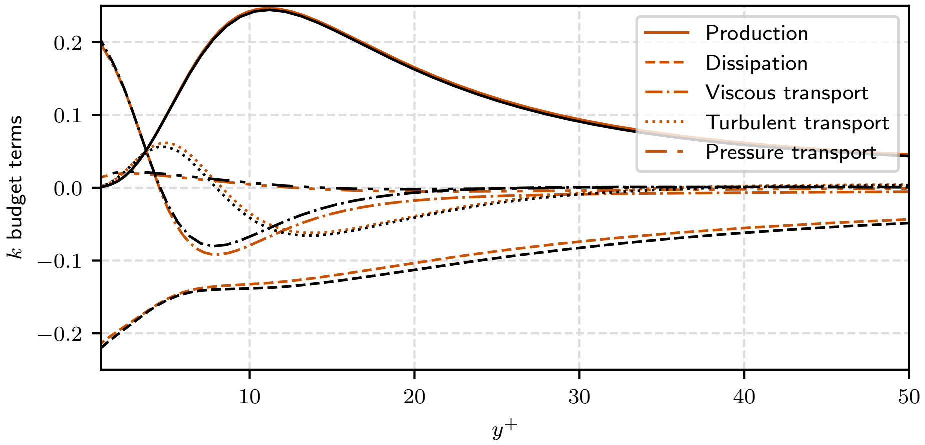

Figure 9. Inner-scaled budget terms in the transport equation for turbulent kinetic energy, k. Orange lines: the simulation on the M3 grid; black lines: reference DNS [11].

The results are presented in Fig. 9, focusing on the inner layer and intentionally matching the plot style of Pope [19] for the reader's benefit. The agreement with the DNS is generally very good. While not aiming for a complete analysis, the following key observations can be made about the observed profiles. The transport terms above the buffer layer are small. This means that k is produced and distributed locally, and the boundary layer is in so-called "equilibrium" (see Section 4.2.3 in Spalart's article [45] for a critical discussion of this concept). The mentioned locality is crucial for some turbulence modeling approaches where the grid does not resolve the inner layer well, e.g. wall-modelled LES. The production peak is located in the buffer layer and exceeds the local dissipation. This excess of energy is transported away, and viscous transport, in particular, is active all the way to the wall, where it compensates for dissipation.

4.3. **Reynolds stress anisotropy.** This section analyzes the Reynolds stress tensor using a so-called anisotropy-invariant map. One motivation for looking at the Reynolds stresses in more detail is that providing means of computing them using mean flow characteristics constitutes Reynolds-average Navier-Stokes modeling.

A brief discussion of the technique can be found in Chapter 11 in Pope's textbook [19], and a more detailed one in Banerjee *et al.* [46]. Reviewing the properties of second-order tensors (see, e.g., Appendix B in Ref. [19]) may also be useful. Recall that since the Reynolds stress tensor is symmetric, one can find a set of so-called principal axes that form a coordinate system in which the tensor is diagonal. The values on the diagonal are the eigenvalues of the tensor, which we denote as λ_i^{Re}. Note that all three must be non-negative because they still represent the variance of the velocity components in the new coordinate system.

Now, consider the normalized deviatoric part of $\overline{u_i'u_j'}$ defined as

$$b_{ij} = \frac{\overline{u_i'u_j'}}{2k} - \frac{1}{3}\delta_{ij}, \tag{8}$$

where δ_{ij} is the Kronecker delta. Since b_{ij} is by construction traceless, the sum of its eigenvalues, λ_i^b, is zero. Therefore, given some λ_1^b and λ_2^b, the last eigenvalue is found as $\lambda_3^b = -\lambda_1^b - \lambda_2^b$. This gives rise to the idea of characterizing b_{ij} by two invariant scalar measures. Furthermore, it is easy to show that λ_i^{Re} and λ_i^b are connected as

$$\lambda_i^b = \frac{\lambda_i^{Re}}{\sum_{j=1}^3 \lambda_j^{\mathrm{Re}}} - \frac{1}{3}. \tag{9}$$

Due to the aforementioned non-negativity criterion for λ_i^{Re}, this puts constraints on what λ_i^b are physically *realizable*. Producing only realizable stress tensors is desirable for any turbulence model.

It is possible to map out b_{ij} using its eigenvalues directly, but the more standard approach, introduced by Lumley and Newman [47], is to consider the invariants of this tensor. Recall that a second-order tensor has three principal invariants. The first one is the trace, but since b_{ij} is deviatoric, it is not useful. The other two are

$$\mathrm{II}_b = \lambda_1^b\lambda_2^b + \lambda_1^b\lambda_3^b + \lambda_2^b\lambda_3^b, \tag{10}$$

$$\mathrm{III}_b = \lambda_1^b\lambda_2^b\lambda_3^b. \tag{11}$$

In later works, for example, Choi and Lumley [48] and in the textbook of Pope [19], yet another set of scalars is used, defined as

$$\xi^3 = \mathrm{III}/2, \quad \eta^2 = -\mathrm{II}/3. \tag{12}$$

The selection of the two scalars changes the shape of the realizability region and the trajectory of the state of b_{ij} (as it changes in time or in space). Generally, the former has a form akin to a triangle and is referred to as the Lumley triangle in the literature. As seen in Fig. 10, in ξ-η, two sides of the triangle are actual lines.

Different parts on the boundary of the triangle correspond to specific limiting states of the anisotropy. In particular, the lower corner corresponds to a fully isotropic Reynolds stress tensor. From Eqn. (9), it is easy to see that if some λ_i^b is set to $-1/3$, then the corresponding λ_i^{Re} becomes zero. Accordingly, the state when exactly one λ_i^b is $-1/3$ is referred to as two-component turbulence and corresponds to the triangle's upper boundary. Using the geometrical analogy between eigenvalues λ_i^{Re} and the semiaxes of an ellipsoid, the two-component case corresponds to the ellipsoid degenerating into a disk. In the top-right corner, the turbulence becomes one-component, for which two λ_i^b are $-1/3$, and the ellipsoid collapses to a line. The two linear sides of the triangle correspond to "axisymmetric" cases for which two of λ_i^b are equal. Here, the term axisymmetric again refers to the two ellipsoid axes being equal.

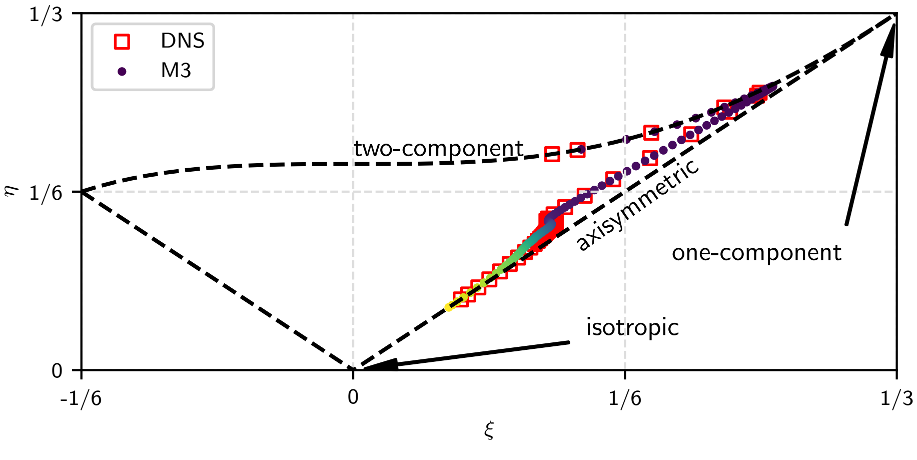

Figure 10. Anisotropy map of the Reynolds stress tensor. For the M3 results, color encodes increasing y^+ from black to yellow. Every fifth point is shown for the DNS data to avoid cluttering.

The markers in Fig. 10 show the anisotropy development in the channel as a function of y^+. The latter values are encoded using a colormap for the LES results. The red markers show the DNS data; the agreement is generally excellent. We see that at low y^+ the turbulence is two-component. It is the wall-normal stress, which is dominated by the two wall-parallel ones [19]. As y^+ grows, the turbulence moves towards isotropy, with the trajectory being almost parallel to the side of the triangle, where turbulence is axisymmetric. Note that the majority of the points clustered near the middle of the axisymmetric axis are in the log-law region.

4.4. **Two-point statistics.** It is well known that coherent structures populate boundary layers, and much research has been done to analyze their behavior. Recall that the mesh size for the simulations was selected based on existing knowledge of the size of the structures in the inner layer. Near-wall velocity streaks may be the best known characteristic of boundary layer flows. Fig. 11, inspired by Pirozzoli and Bernandini [49], illustrates them in the log-layer. The figure is a volume rendering of u'/U_b, with the opacity tuned to zero outside narrow bands close to -0.1 and 0.1. The result is close to two isosurfaces but with an additional "thickness" enhancing depth perception without using shadows. The streaks represent alternating low and high momentum regions that are elongated in the streamwise direction and occupy a fraction of δ in z. These observations explain why Δz^+ has to be smaller than Δx^+.

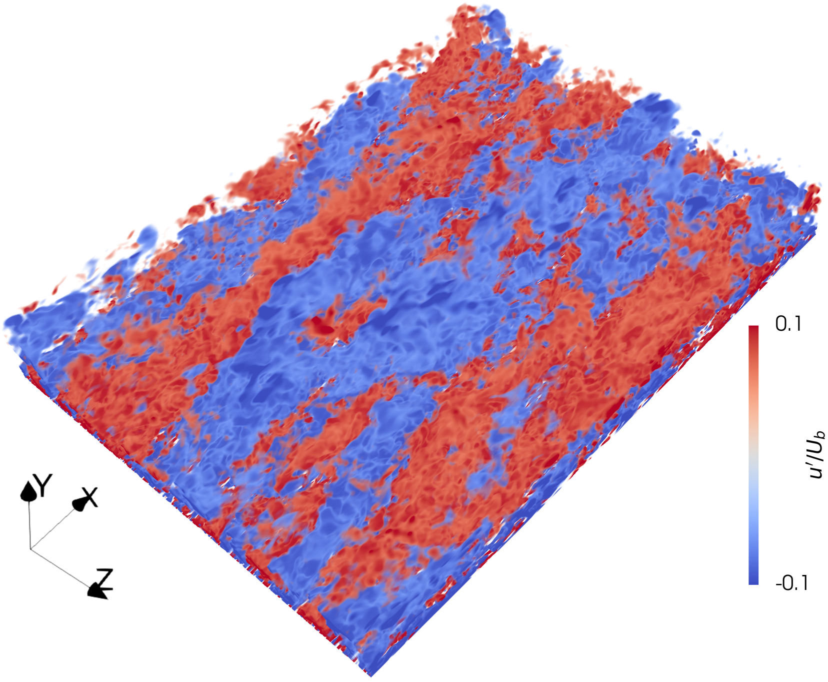

Figure 11. Volume rendering of u'/U_b in the lower half of the channel with opacity concentrated around the -0.1 and 0.1 values.

Quantitative analysis of the structures is often based on two-point statistics, that is, statistical quantities derived from observations of the flow variables at two distinct points in space or time. Both physical and frequency (Fourier) spaces can be used, and here we begin by considering the one-dimensional spanwise energy spectra, denoted as E_{uu}, E_{vv}, E_{ww} for the three velocity components, respectively. The energy spectrum is a statistic in the Fourier space, showing how the energy of a particular velocity component is distributed across frequencies (in our case, spatial, although time spectra can also be considered). Integrating the spectrum across the entire frequency axis recovers the variance of the signal, i.e., the corresponding component of the Reynolds stress.

Popular scientific computing packages for dealing with Fourier analysis (like SciPy [50]) base the names of the routines on the conventions of the signal processing community. This can cause significant confusion, therefore, a short note is made on the difference between some of these definitions compared to what is used for turbulence. In signal processing, the energy of, for example, $u'(z)$ is defined as an integral over the entire z axis,

$$\int_{-\infty}^{\infty} u'(z)u'(z)dz. \tag{13}$$

Since $u'(z)$ is periodic, it has infinite energy under this definition, and the energy spectrum is not defined. The same holds for any stationary signal. Instead, $\overline{u'(z)u'(z)}$ is referred to as power[3], and its distribution over frequencies as the power spectral density (PSD). By contrast, in turbulence, $\overline{u'(z)u'(z)}$ comes in as a component of k, the energy, and the energy spectrum is the corresponding term for the PSD.

Like any other statistic, the PSD computed from a sampled signal will often contain noise errors due to the finite amount of samples and under-resolution. Several methods, such as Welch's, allow for reducing the noise level by sacrificing some frequency resolution. Here, we compute the discrete PSD as is, that is, compute the fast Fourier transform of the signal, square the obtained coefficients, and divide the result by the number of samples squared. This process is called computing the periodogram and can be accomplished using a function provided in, for example, SciPy. Since the spectrum is spatial, its computation is performed at each time value, and afterward, a temporal average is computed to provide the final result.

Figure 12 shows the spectra for each velocity component, taken at 3 different y^+ values: 12, 200, and 850. Overall, the agreement with DNS is excellent, to the extent that it is somewhat deceiving. For example, E^+_{vv} at $y^+ = 12$ appears to overlap perfectly with the DNS. Still, we have seen that the integral of this spectrum, i.e., the corresponding $\overline{v'v'}^+$ value at the same wall-normal location, does have an appreciable discrepancy with the DNS, see Fig. 12. This serves as a reminder that log-log scaling can obscure certain errors.

Nevertheless, some error patterns can be seen in the LES results. One is the aliasing of low-frequency modes for the streamwise component, which manifests in a more distinct peak than in the DNS data. This is caused by the insufficient size of the simulation domain to represent the largest scales of the flow properly. The other discrepancy is the pileup of energy in E^+_{uu} and E^+_{ww} at high frequencies and $y^+ = 12$. This is a typical error caused by insufficient resolution. Essentially, energy from unrepresented frequencies gets transferred to those that are resolved in the simulation. Extra energy may also come simply from numerical noise. Since the turbulent structures are smallest near the wall, it is clear why these effects will manifest themselves most prominently in that region. Closer to the core of the channel the smallest scales are instead over-dissipated for E^+_{uu} and E^+_{vv}.

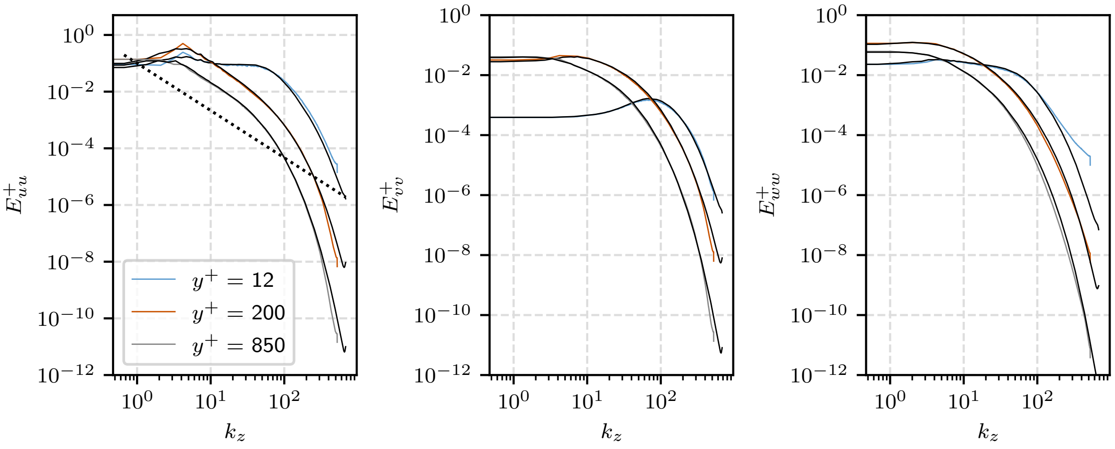

Figure 12. One-dimensional spanwise spectra at different y^+ values. Black solid lines show the reference DNS. The dashed line shows $k_z^{-5/3}$.

For the streamwise component, the plot in Fig. 12 also features a reference line showing $k_z^{-5/3}$, which is Kolmogorov's prediction for the inertial range, under the assumption of homogeneity and isotropy. The assumptions hold better further away from the wall, which explains why a -5/3 region is not observed at $y^+ = 12$. Generally, the inertial range is still short at this Reynolds number.

Instead of looking at the spectrum itself, the so-called premultiplied spectrum is often considered, where the values are multiplied by the frequency (or wavenumber). The idea is that if we look at, for example, $f_z E_{uu}$ in logarithmic scaling for the f_z axis, then the total area under the profile will still correspond to the total energy of u' (just like it does for E_{uu} in linear scaling). Figure 13 compares the obtained premultiplied spectrum of u' with the DNS reference. The data is shown across the entire $[0, \delta]$ half-plane. Instead of the frequency, the ordinate in the figure is the inner-scaled wavelength λ^+_z. As expected, the most prominent peak in the values is found in the buffer layer. However, the considered

[3] Sometimes, *average* power, as opposed to *instantaneous power* defined as $u'(z)u'(z)$.

Reynolds number is sufficiently high for the secondary peak in the outer layer ($y^+ \approx 200$) to begin developing. In the region of this peak, some discrepancy between the LES and the DNS can be observed, likely due to the smaller domain size for the former.

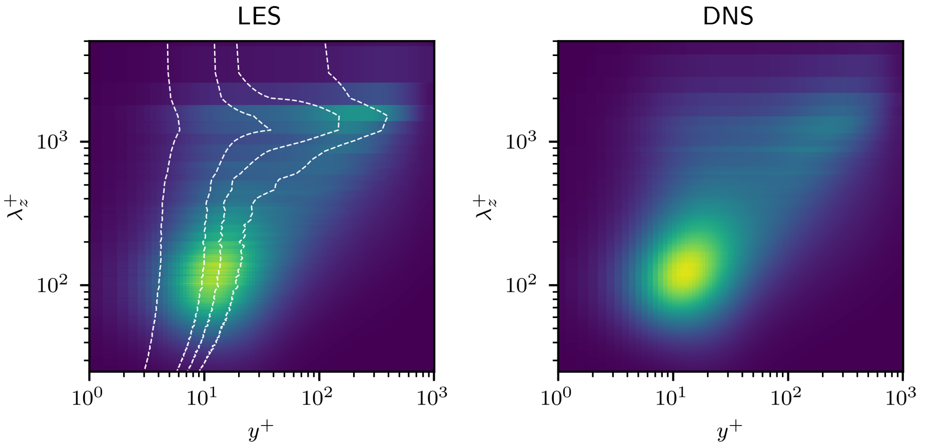

Figure 13. The distribution of $f_z E_u$ across the $[0, \delta]$ half-plane of the channel. Dashed lines for the LES plot additionally show, from left to right, the isolines of $\gamma^2 = (0.95, 0.6, 0.4, 0.2)$ for $y^+_{ref} = 2$.

For the LES, Fig. 13 also shows the isolines of the linear coherence spectrum, which is defined as

$$\gamma^2\left(y, y_{ref}; \lambda_z\right) \equiv \frac{|\overline{\hat{u}(y; \lambda_z)\hat{u}^*(y_{ref}; \lambda_z)}|^2}{\overline{|\hat{u}(y; \lambda_z)|^2} \cdot \overline{|\hat{u}\left(y_{ref}; \lambda_z\right)|^2}}. \tag{14}$$

Here, y_{ref} is a reference wall-normal distance, with respect to which γ^2 is computed, the hat represents the coefficients of the fast Fourier transform in z, and the asterisk superscript the complex conjugate. For a given wavelength λ_z and wall-normal distance y, the numerator of γ^2 measures the squared modulus of the covariance of a given Fourier mode with its counterpart at the reference y_{ref}. The covariance distribution across λ_z is the co-spectrum of the two signals, just like the ordinary spectrum shows the distribution of a single signal's variance. The denominator of Eqn. (14) provides normalization, such that at $y = y_{ref}$, the value of γ^2 is one.

The isolines of γ^2 in Fig. 13 are computed for $y^+_{ref} = 2$. The values decay with the distance to the reference, as expected. The interesting observation is that at small wavelengths, the signals decorrelate quickly, but at the location of the outer peak, non-zero values persist well into the log-layer. This shows that the inner- and outer-layer velocities are coupled at large wavelengths. A similar result is shown for boundary layer simulations by Tenarro *et al.* [51] and experimental results of Baars *et al.* [52]. The coupling between the outer and inner layers is an active research area. The pioneering paper of Marusic *et al.* [53] is a good entry point to the subject.

We now consider two-point statistics in physical space, in particular, the one-dimensional two-point auto-correlations in z. The definition of the quantity is

$$R_{\phi\phi}(z) = \frac{\overline{\phi'(z_{ref})\phi'(z_{ref} + \Delta z)}}{\overline{\phi'(z_{ref})\phi'(z_{ref})}}, \tag{15}$$

where ϕ is again some flow variable and z_{ref} is a reference value (which, unlike y_{ref} for the coherence, is arbitrary because the flow is homogeneous in z), and Δz is the distance from the reference. At sufficiently large Δz, the correlation is expected to go down to zero. This serves as a way to test the adequacy of the chosen domain size: if the size is too small, R does not decay to zero within its limits. The LES results are shown in Fig. 14 at the same y^+ as the one-dimensional spectra above. We see that R_{vv} decays to 0 already at $z/\delta \approx 1$, whereas R_{ww} does so slower, but the domain size remains sufficient. For the streamwise component, R_{uu}, the results in the log layer and near the centerline are less clear-cut as the values oscillate around 0. This behavior corresponds well to the observations made for the premultiplied spectrum above.

The "wavy" behavior of R, where it dips from 1 to a low value and then comes up again, indicates the streaky velocity pattern shown in Fig. 11. The distance to the first dip provides a rough estimate of the spacing of the streaks.

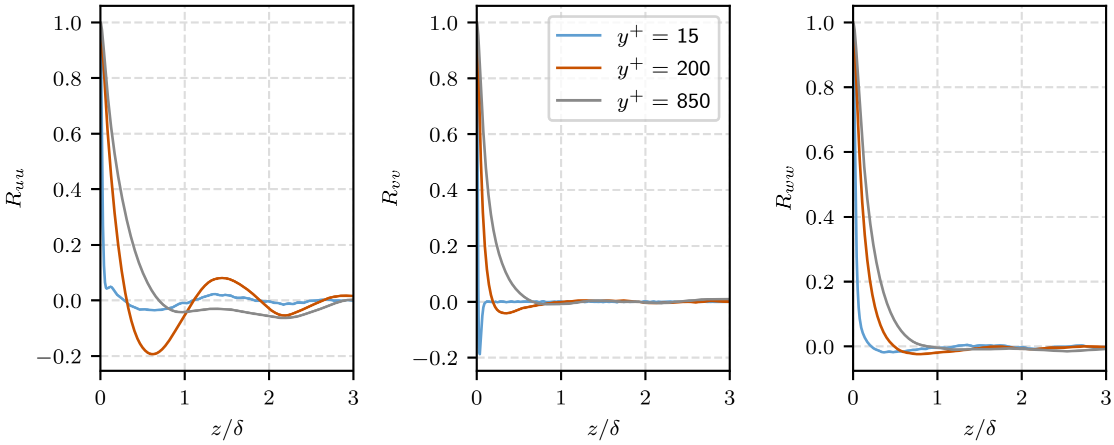

Figure 14. Two-point correlations in z of the velocity components at 3 different y^+ values.

4.5. **Comparison with RKSymFoam.** The solver `RKSymFOAM` is an alternative to `pimpleFoam`, specifically targeting high-fidelity simulations [54]. The solver uses a rich library of Runge-Kutta time-stepping methods and recent developments in symmetry-preserving projection methods. A detailed account of the latter is available in Santos *et al.* [55], but the key property in terms of accuracy is preserving the kinetic energy in the inviscid limit.

It is interesting to quantify and compare the numerical dissipation introduced in the `RKSymFOAM` and `pimpleFoam` solvers. To that end, we use the metric proposed by Komen *et al.* [24]. In that work, the numerical dissipation present in the flow is evaluated as the residual of the k budget (presented in Section 4.2 above for the simulation on the M3 grid). The idea is that since numerical dissipation is acting on k, the budget will have a positive residual, denoted ϵ_{num}.

Figure 15 shows the estimated numerical dissipation of both codes, on meshes M1 and M2. The profiles from `RKSymFoam` exhibit a much smaller ϵ_{num} peak in the buffer region compared to the `pimpleFoam` simulations (and also the results of Komen *et al.* [24]). Interestingly, for $y^+ > 100$, the residual is very close to that in the DNS data for both simulations on the M2 grid, and for `RKSymFoam` also on the M1 grid. This would indicate that numerical diffusion is essentially inactive in almost the entire outer layer, which seems overly optimistic, particularly on the M1 grid.

Regardless, the reduced ϵ_{num} does not significantly increase the predictive accuracy, but rather causes change in the error pattern, as demonstrated in Fig. 16. The near-wall under-prediction in the streamwise stress is somewhat reduced in the profiles for mesh M1, but the wall-normal stress is, arguably, predicted somewhat worse. On the M2 mesh, the results of the solvers are even more similar. A measurable improvement is, however, obtained in the values of the friction velocity, see Tab. 2.

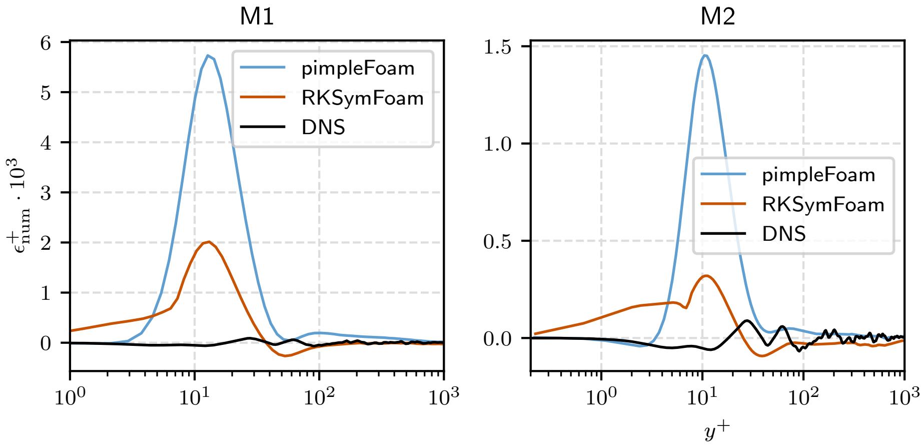

Figure 15. The inner-scaled residual of the k equation budget in the implicit LES, ϵ^+_{num}, using `pimpleFoam` and `RKSymFoam`.

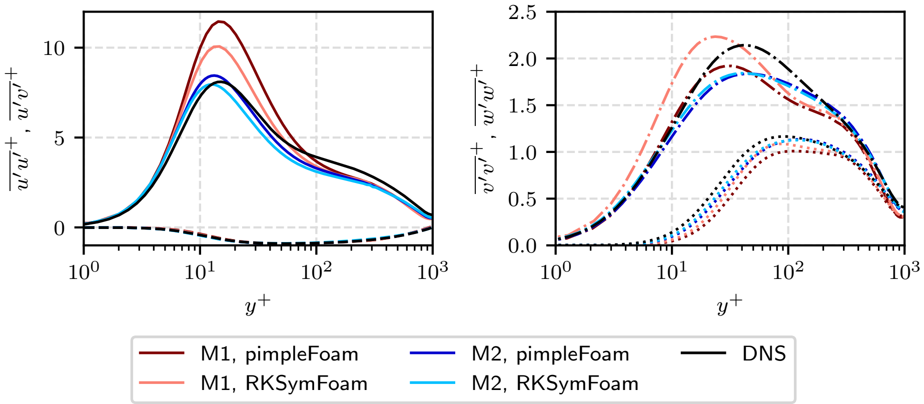

Figure 16. Reynolds stress profiles using `pimpleFoam` and `RKSymFoam`. Solid lines: $\overline{u'u'}^+$; dashed lines: $\overline{u'v'}^+$; dotted lines: $\overline{v'v'}^+$, dash-dotted lines: $\overline{w'w'}^+$.

Unfortunately, the time to solution with `RKSymFoam` on the M2 grid is about 2 times higher than for `pimpleFoam`. Several factors are behind this. Firstly, for the DIRK2 scheme, `RKSymFoam` does 4 pressure-Poisson solves per time step versus 3 for `pimpleFoam`. Furthermore, 2 of these solves are considered "final", for which the solver tries to converge to the set absolute tolerance, thus requiring more iterations. In `pimpleFoam`, this is done only for the last solve. The last factor is that the Poisson solver struggled to converge to the desired tolerance, leading to a much larger amount of iterations. A strong zoom on Fig. 16 reveals small oscillations in `RKSymFoam`'s $\overline{u'u'}^+$ profile at $y^+ \approx 300$. We can speculate that some numerical artifacts have entered the solution, giving the solver a hard time. A detailed investigation of this is beyond the scope of this work.

In conclusion, `RKSymFoam` delivered a lower numerical dissipation, making it better suited for, e.g., DNS and studying subtle flow features, such as the rear near-wall events discussed briefly above, in conjunction with the flatness profiles. It may also be able to push the upper limit of grid coarsening for which the numerical dissipation starts dominating ν_{sgs}, thus making it meaningful to use SGS models. A relevant study in this regard is that of Montecchia *et al.* [36], and a partial investigation is performed in the next section. However, we could not draw a tangible benefit from the improved numerics for resolution typical of implicit wall-resolved LES. Also, further study—or input from experts—is needed on which Runge-Kutta scheme to select to balance accuracy and computational cost.

4.6. **Subgrid-scale modelling.** The results obtained with implicit LES on grids M1 and M2 are here compared to those with the Sigma SGS model [23] on the same grid. Moreover, we also include results from `RKSymFoam` simulations on the M1 grid, with and without explicit SGS modelling. We demonstrate the difference in statistics using the Reynolds stresses only for brevity, see Fig. 17. As indicated by the profiles in the figure, the effect of the SGS model is not very strong, and the results do not consistently improve. In fact, the most profound influence of SGS modelling is on the `RKSymFoam` simulation, but, unfortunately, the results becomes worse when the Sigma model is active. For other quantities, the results are similar.

The failure of SGS models to provide an accuracy boost in finite volume solvers is an issue known to the community. The already mentioned study by Komen *et al.* provides a good quantitative analysis [24], predominately based on introducing $\nu_{\text{num}}/\nu = \epsilon_{\text{num}}/\epsilon$ and comparing it to the ratio ν_{sgs}/ν. Their results demonstrate that for channel flow at $\text{Re}_\tau = 180$ the numerical contribution is larger than the one from SGS model.

In Fig. 18 we compare these viscosity ratios for our simulations. For convenience, the ordinate is presented in logarithmic scale, and the abosolute value of ν_{num} is shown to avoid gaps where it is negative. Interestingly, our results do not agree with Komen *et al.*'s. The SGS viscosities (dashed lines) are consistently higher than the numerical ones (solid lines). The only exception is a small region close to the wall ($y^+ \approx 10$), where in the `pimpleFoam` simulation on the M1 grid the numerical dissipation exceeds the SGS one and in the other simulations the two curves come close.

In Appendix C, we show data from additional simulations at $\text{Re}_\tau = 180$ with the M1 grid, demonstrating that the numerical dissipation on the same level as the DNS, whereas subgrid scale viscosity is, relatively, much higher. Moreover, results using the Smagorinsky model (used in Ref. [24]) are significantly different (less accurate) from the implicit LES simulation. This, again, contradicts the findings in Ref. [24]. However, in Appendix D we show that when a more dissipative scheme is used for the convective fluxes, ν_{num}/ν can greatly exceed the typical ν_{sgs}/ν values, providing some support to the thesis that in more applied simulations SGS modelling may not be meaningful.

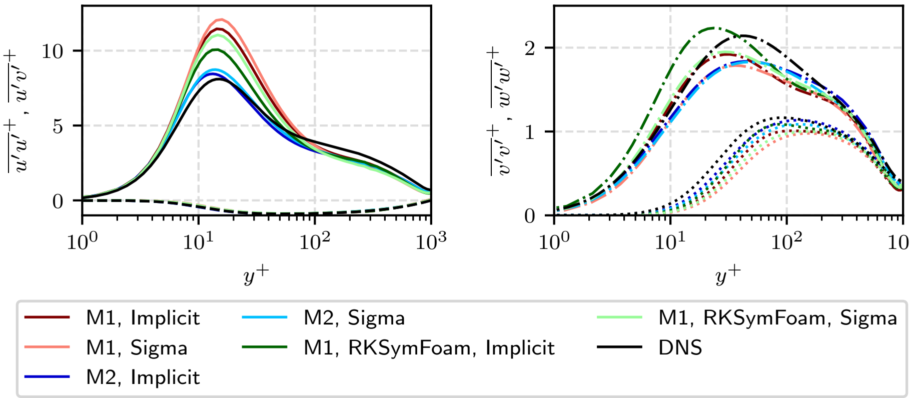

Figure 17. Reynolds stress profiles using implicit LES and the Sigma model. Solid lines: $\overline{u'u'}^+$; dashed lines: $\overline{u'v'}^+$; dotted lines: $\overline{v'v'}^+$, dash-dotted lines: $\overline{w'w'}^+$.

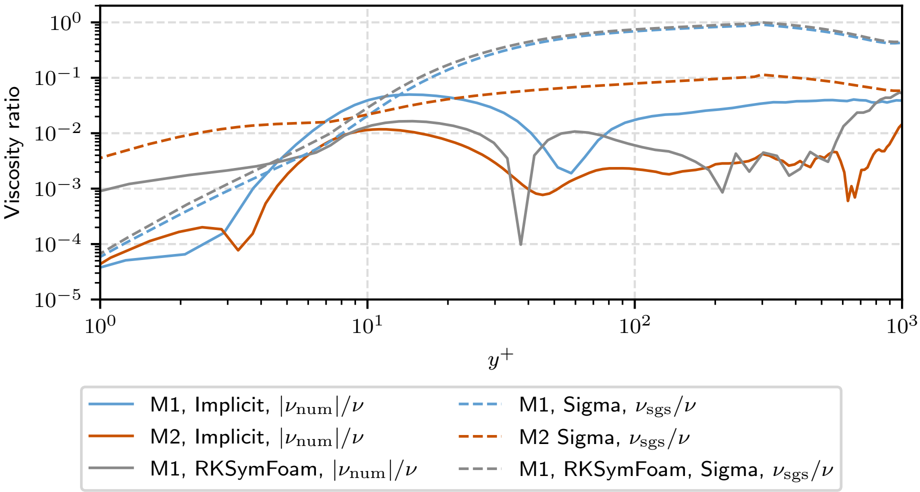

Figure 18. The ratio of the numerical viscosity ν_{num} and ν for the implicit LES and ν_{sgs} and ν for the Sigma LES.

In summary, the results in this section show that SGS models fail to improve the predictive accuracy of LES when used in second-order accurate finite volume solvers. Essentially, they either slightly redistribute the error pattern in y, or make the errors larger. When relatively dissipative schemes are used, this is clearly explained by the SGS viscosity being dominated by numerical dissipation (see Appendix D). On orthogonal meshes and schemes selected to minimize numerical dissipation, the explanation appears to remain elusive. In Komen *et al.* [24], the effect of numerics is still found to be culprit, but we cannot reproduce those findings here. In fact, the most significant effect of SGS modelling was observed in the simulation with `RKSymFoam`, which is also shown to exhibit less dissipation. Further investigations are necessary to shed light on this contradiction.

5. Conclusions

This paper presents detailed validation for OpenFOAM's transient solver `pimpleFoam` for high-fidelity simulation of channel flow at $\mathrm{Re}_\tau \approx 1000$. The results on grid M3 with quasi-DNS wall-parallel resolution ($\Delta x^+ = 10$, $\Delta z^+ = 6$) exhibit excellent agreement with reference data for the mean velocity, the Reynolds stresses, the budget of turbulent kinetic energy, and spanwise spectra. For velocity skewness and flatness, the profiles deviate from the DNS in certain regions, for example, in the viscous sublayer for F_v, but remain accurate overall. A similar agreement is found for the components of the vorticity fluctuation tensor. In summary, the results convincingly demonstrate that `pimpleFoam` can accurately simulate high

Reynolds number wall-bounded turbulence, provided that the turbulent scales are well-resolved with the grid.

The above results are complemented by simulations on coarser grids. As expected, the accuracy deteriorates quickly as the cell size increases. Only the mean velocity remains equally accurate on grid M2 ($\Delta x^+ = 20$, $\Delta z^+ = 12$). Using an SGS model (here, Sigma) does not improve the results. Via the approach in Ref. [24], we show that the SGS viscosity coming from the model is larger than the numerical dissipation. This contradicts the results in Ref. [24] itself, where the opposite relationship has been observed and used to explain the SGS models' inefficacy. Investigation is necessary to shed further light on the interplay between numerical and subgrid dissipation and how the present findings fit with more successful applications of SGS modelling in OpenFOAM, such as the work of Montecchia *et al.* [36]. However, we note that in industrial applications, more dissipative schemes are used than in this study, and we show that this may, indeed, quickly lead to SGS viscosity being dwarfed by numerical effects.

We also used the community-contributed solver `RKSymFoam` [54], which uses Runge-Kutta time-stepping and carefully selected numerical schemes [55]. The produced numerical dissipation on the M1 and M2 grids is lower than that of `pimpleFoam`, with no distinct peak in the k budget imbalance in the buffer region. Unfortunately, this does not lead to a tangible increase in predictive accuracy. However, capitalizing on the reduced dissipation may be possible in a DNS-based study focusing on subtle flow features.

Acknowledgements

For computer time, this research used Shaheen III, managed by the Supercomputing Core Laboratory at King Abdullah University of Science & Technology (KAUST) in Thuwal, Saudi Arabia. The authors thank Johannes Hopman from BarcelonaTech for discussions of the `RKSymFoam` results. The authors thank Siavash Toosi and Philipp Schlatter from FAU Erlangen-Nürnberg for providing the DNS data for high-order moments and also for fruitful discussions of correlations and spectra.

Author Contributions: Conceptualisation, T.M.; methodology, T.M.; software, T.M; validation, T.M; investigation, T.M.; resources, M.P.; data curation, T.M.; writing—original draft preparation, T.M.; writing—review and editing, T.M. and M.P.; visualisation, T.M.; supervision, M.P.; project administration, M.P.; funding acquisition, M.P. All authors have read and agreed to the published version of the manuscript.

Appendix A. Comparison of plane- and volume-averaged data

In this appendix, we compare the statistics based on averaging spatially across the x-z plane of the channel and based on averaging across z only. Both are also temporally averaged over the same time span. We focus the comparison on the Reynolds stresses, which are shown in Fig. 19 for mesh M2. The curves for all four components are very close to each other. A small difference is discernible for $\overline{u'u'}^+$, which is typically the slowest stress to converge in channel flow.

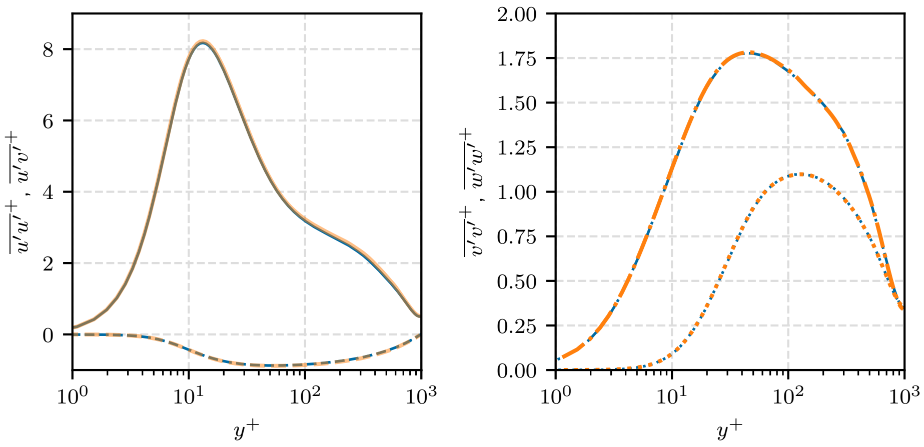

Figure 19. Comparison of plane-averaged Reynolds stresses with volume-averaged. Profiles from mesh M2. Solid lines: $\overline{u'u'}^+$; dashed lines: $\overline{u'v'}^+$; dotted lines: $\overline{v'v'}^+$, dash-dotted lines: $\overline{w'w'}^+$. Blue lines show volume-averaged data, and orange the plane-averaged.

Appendix B. Sensitivity to time step size

As mentioned in Section 3, the simulations are run with a time step that ensures that the maximum CFL number is below 0.5 at each time step of the simulation. That means that the time step changes with mesh resolution, becoming smaller proportional to the decrease in the grid cell size. Here, we evaluate the effect of the time step on the results by computing the implicit LES on grid M1 with adaptive time stepping fulfilling maximum CFL number 0.5, 0.2, and 0.1 respectively. The Reynolds stresses are presented in Fig. 20. A modest effect of the time step is visible on the streamwise stress, when the maximum CFL is reduced from 0.5 to 0.2. For simulations on grids M2 and M3 the effect of the time step can be considered negligible since Δt scales down proportional to the grid resolution.

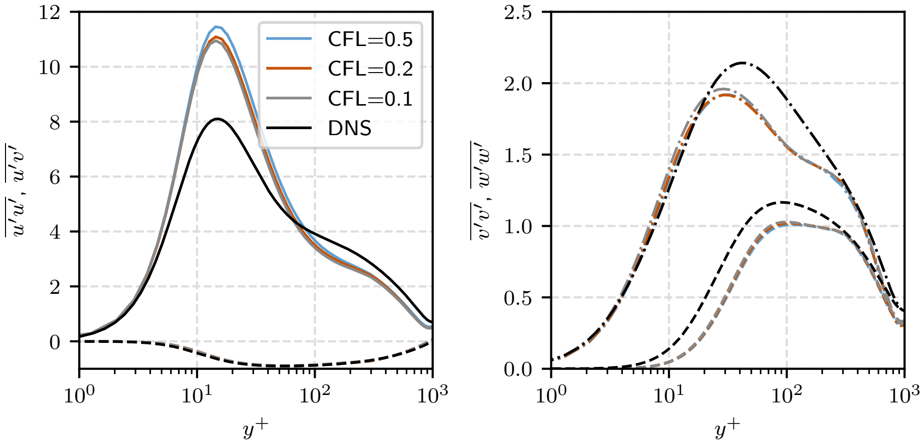

Figure 20. Comparison of simulation results on grid M1, computed with adaptive time-stepping satisfying different maximum CFL number criteria. Solid lines: $\overline{u'u'}^+$; dashed lines: $\overline{u'v'}^+$; dotted lines: $\overline{v'v'}^+$, dash-dotted lines: $\overline{w'w'}^+$.

Appendix C. Simulations at Re$_\tau \approx 180$

In this section, we present results for simulations on the M1 grid performed at Re$_\tau \approx 180$. For reference, we use the same DNS database [11] as for Re$_\tau \approx 1000$. For this case, the M1 grid corresponds to resolution $\Delta x^+ \approx 7.2$, $\Delta z^+ \approx 4.3$, $\max(\Delta y^+) \approx 6.5$. Three simulations are run: with no SGS model, with the Sigma model, and also with the Smagorinsky model [20] with near wall damping. The latter is motivated by the desire to mimic the setup in Komen *et al.* [24], which focuses on quantifying numerical dissipation by looking at the k budget's residual, ϵ_{num}. The principal result of Ref. [24] is that $\nu_{\text{num}}/\nu = \epsilon_{\text{num}}/\epsilon$ is larger than ν_{sgs}/ν, which explained that the lack of any strong effect of SGS modelling on the results.

Tab. 3. Relative error in the friction velocity for simulations at Re$_\tau \approx 180$.

Simulation	Relative error
Implicit	-1.53 %
Sigma	-2.32 %
Smagorinsky	-2.52 %

Our results do not agree with that of Ref. [24]. In Fig. 21, the profiles obtained with the Smagorinsky model diverge significantly from those obtained with the Sigma model and implicit LES. Neither SGS model improves the results (see also Tab. 3), and the Smagorinsky model makes them appreciably less accurate. Furthermore, the analysis of numerical dissipation (see Fig. 22) shows that ϵ_{num} in the implicit LES is on par with DNS data, and the numerical viscosity is significantly smaller than the subgrid one.

The differences in results compared to Komen *et al.* [24] may be due to inconsistencies in the simulation setup. Our grid is slightly finer than the one they use for implicit LES. It is possible that the parameters of the Smagorinsky model (e.g. damping coefficients) are also not matched. However, both studies use linear interpolation for the convective fluxes, which is typically the crucial determinant for the amount of numerical dissipation. Further investigation is necessary to harmonize the findings of both works or better explain the observed discrepancies.

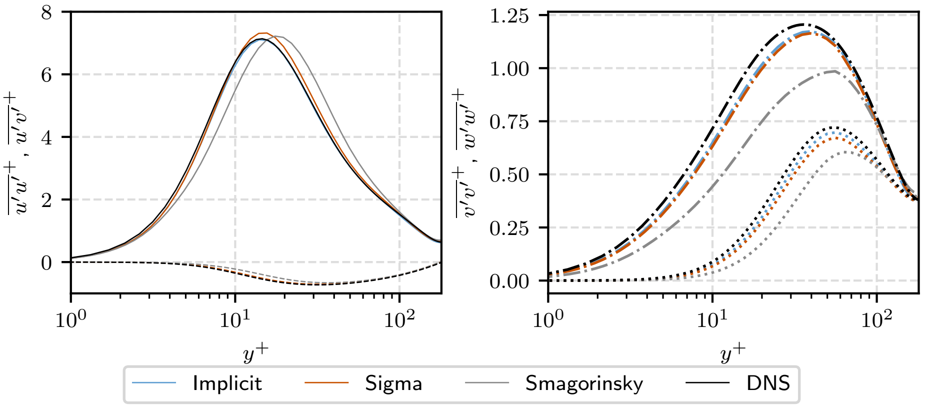

Figure 21. Comparison of simulation results (Reynolds stresses) at $\mathrm{Re}_\tau \approx 180$ on grid M1. Solid lines: $\overline{u'u'}^+$; dashed lines: $\overline{u'v'}^+$; dotted lines: $\overline{v'v'}^+$, dash-dotted lines: $\overline{w'w'}^+$. In the left plot, the blue curves are overlapped by the black ones.

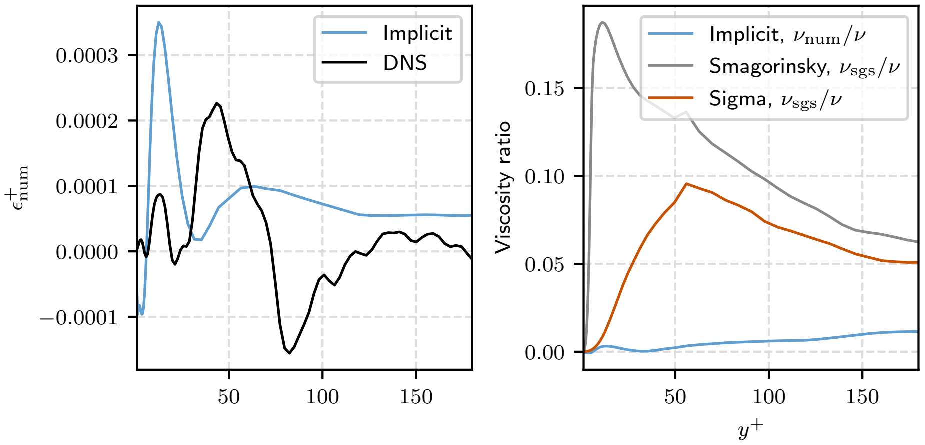

Figure 22. Analysis of numerical and SGS dissipation in simulations at $\mathrm{Re}_\tau \approx 180$. *Left:* The residual of the k-equation budget, $\epsilon^+_{\mathrm{num}}$. *Right:* Ratio of numerical and molecular viscosity, ν_{num}/ν and SGS and molecular viscosity, ν_{sgs}/ν .

Appendix D. Simulations using the LUST scheme

In this section, we demonstrate the effect of using a more diffusive scheme for the convection term. In particular, we use an OpenFOAM scheme called `LUST`, which is a second-order scheme that linearly blends the interpolation weights of the linear scheme and a second-order upwind scheme. The weight of the latter is 25%. `LUST` can be considered for scale-resolving simulations when the grid cannot support using the linear scheme and extra dissipation is required to maintain stability.

The Reynolds stresses are shown in Fig. 23 and reveal a strong deterioration in accuracy due to upwinding. Figure 24 shows that the values of ν_{num}/ν are very high in the `LUST` simulation; in particular, they are much higher than any SGS viscosity introduced on the M1 grid in other simulations.

References

[1] U. Schumann, “Subgrid scale model for finite difference simulations of turbulent flows in plane channels and annuli,” *Journal of Computational Physics*, vol. 18, no. 4, pp. 376–404, 1975. [Online]. Available: http://dx.doi.org/10.1016/0021-9991(75)90093-5

[2] P. Moin and J. Kim, “Numerical investigation of turbulent channel flow,” *Journal of fluid mechanics*, 1982.

[3] J. Kim, P. Moin, and R. D. Moser, “Turbulence statistics in fully developed channel flow at low Reynolds number,” *Journal of Fluid Mechanics*, vol. 177, pp. 133–166, 1987.

[4] R. D. Moser, J. Kim, and N. Mansour, “Direct numerical simulation of turbulent channel flow up to Re[sub τ]=590,” *Physics of Fluids*, vol. 11, no. 4, p. 943, 1999. [Online]. Available: http://dx.doi.org/10.1063/1.869966

[5] J. C. del Álamo, J. Jiménez, P. Zandonade, and R. D. Moser, “Scaling of the energy spectra of turbulent channels,” *Journal of Fluid Mechanics*, vol. 500, no. 2004, pp. 135–144, 2004. [Online]. Available: http://dx.doi.org/10.1017/S002211200300733X

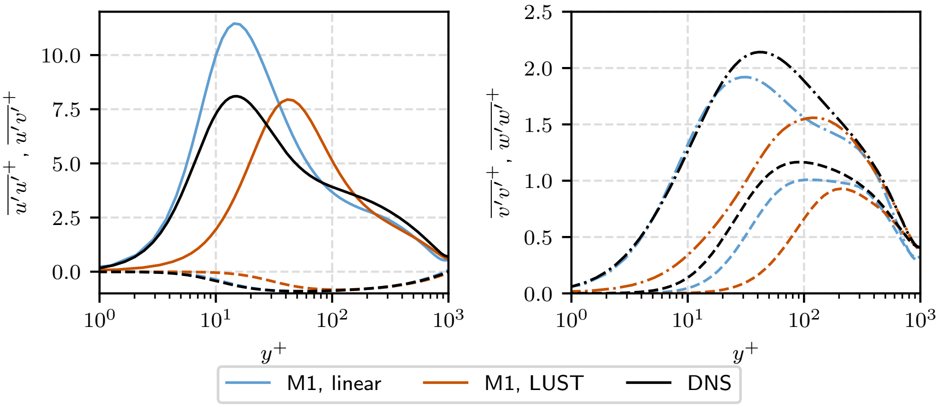

Figure 23. Comparison of simulation results (Reynolds stresses) using the linear and LUST schemes for the convective term. Solid lines: $\overline{u'u'}^+$; dashed lines: $\overline{u'v'}^+$; dotted lines: $\overline{v'v'}^+$, dash-dotted lines: $\overline{w'w'}^+$.

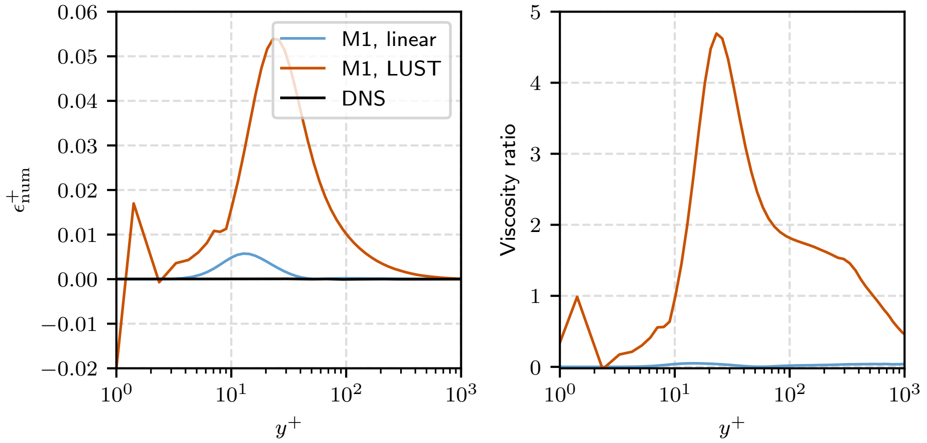

Figure 24. *Left:* The residual of the k-equation budget, $\epsilon^+_{\mathrm{num}}$. *Right:* Ratio of numerical and molecular viscosity, ν_{num}/ν.

[6] S. Hoyas and J. Jiménez, “Scaling of the velocity fluctuations in turbulent channels up to Re[sub τ]=2003,” *Physics of Fluids*, vol. 18, no. 1, p. 011702, 2006. [Online]. Available: http://dx.doi.org/10.1063/1.2162185

[7] ——, “Reynolds number effects on the Reynolds-stress budgets in turbulent channels,” *Physics of Fluids*, vol. 20, no. 10, p. 101511, 2008. [Online]. Available: http://dx.doi.org/10.1063/1.3005862

[8] A. Lozano-Durán and J. Jiménez, “Effect of the computational domain on direct simulations of turbulent channels up to $\mathrm{Re}_\tau = 4200$,” *Physics of Fluids*, vol. 26, no. 1, 2014. [Online]. Available: http://dx.doi.org/10.1063/1.4862918

[9] J. C. del Álamo and J. Jiménez, “Spectra of the very large anisotropic scales in turbulent channels,” *Physics of Fluids*, vol. 15, no. 6, p. L41, 2003. [Online]. Available: http://dx.doi.org/10.1063/1.1570830

[10] A. Lozano-Durán, O. Flores, and J. Jiménez, “The three-dimensional structure of momentum transfer in turbulent channels,” *Journal of Fluid Mechanics*, vol. 694, pp. 100–130, 2012. [Online]. Available: http://dx.doi.org/10.1017/jfm.2011.524

[11] M. Lee and R. D. Moser, “Direct numerical simulation of turbulent channel flow up to $\mathrm{Re}_\tau \approx 5200$,” *Journal of Fluid Mechanics*, vol. 774, pp. 395–415, 2015.

[12] ——, “Spectral analysis of the budget equation in turbulent channel flows at high Reynolds number,” *Journal of Fluid Mechanics*, vol. 860, 2019. [Online]. Available: http://dx.doi.org/10.1017/jfm.2018.903

[13] Y. Yamamoto and Y. Tsuji, “Numerical evidence of logarithmic regions in channel flow at $Re_\tau = 8000$,” *Physical Review Fluids*, vol. 3, no. 1, p. 012602, 2018. [Online]. Available: http://dx.doi.org/10.1103/PhysRevFluids.3.012602

[14] M. Oberlack, S. Hoyas, S. V. Kraheberger, F. Alcántara-Ávila, and J. Laux, “Turbulence statistics of arbitrary moments of wall-bounded shear flows: A symmetry approach,” *Physical Review Letters*, vol. 128, no. 2, p. 024502, January 2022. [Online]. Available: http://dx.doi.org/10.1103/PhysRevLett.128.024502

[15] C. Fureby, A. D. Gosman, G. Tabor, H. G. Weller, N. D. Sandham, and M. Wolfshtein, “Large eddy simulation of turbulent channel flows,” in *Turbulent Shear Flows 11*, 1997.

[16] V. Vuorinen, M. Larmi, P. Schlatter, L. Fuchs, and B. J. Boersma, “A low-dissipative, scale-selective discretization scheme for the Navier-Stokes equations,” *Computers and Fluids*, vol. 70, pp. 195–205, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.compfluid.2012.09.022

[17] T. Mukha and M. Liefvendahl, "Large-eddy simulation of turbulent channel flow," Uppsala University, Department of Information Technology, Tech. Rep., 2015.

[18] E. M. J. Komen, J. A. Hopman, E. M. A. Frederix, F. X. Trias, and R. W. C. P. Verstappen, "A symmetry-preserving second-order time-accurate PISO-based method," *Computers & Fluids*, vol. 225, p. 104979, 2021. [Online]. Available: http://dx.doi.org/10.1016/j.compfluid.2021.104979

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

[20] J. Smagorinsky, "General circulation experiments with the primitive equations," *Monthly Weather Review*, vol. 91, no. 3, pp. 99–163, 1963.

[21] A. W. Vreman, "An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications," *Physics of Fluids*, vol. 16, no. 10, pp. 3670–3681, 2004. [Online]. Available: http://dx.doi.org/10.1063/1.1785131

[22] F. Nicoud and F. Ducros, "Subgrid-scale stress modelling based on the square of the velocity gradient tensor," *Flow, Turbulence and Combustion*, vol. 62, no. 3, pp. 183–200, 1999. [Online]. Available: http://dx.doi.org/10.1023/A:1009995426001

[23] F. Nicoud, H. B. Toda, O. Cabrit, S. Bose, and J. Lee, "Using singular values to build a subgrid-scale model for large eddy simulations," *Physics of Fluids*, vol. 23, no. 8, 2011. [Online]. Available: http://dx.doi.org/10.1063/1.3623274

[24] E. M. J. Komen, L. H. Camilo, A. Shams, B. J. Geurts, and B. Koren, "A quantification method for numerical dissipation in quasi-DNS and under-resolved DNS, and effects of numerical dissipation in quasi-DNS and under-resolved DNS of turbulent channel flows," *Journal of Computational Physics*, vol. 345, pp. 565–595, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.jcp.2017.05.030

[25] C. Fureby and F. F. Grinstein, "Large eddy simulation of high-Reynolds-number free and wall-bounded Flows," *Journal of Computational Physics*, vol. 181, no. 1, pp. 68–97, 2002. [Online]. Available: http://dx.doi.org/10.1006/jcph.2002.7119

[26] R. Courant, K. Friedrichs, and H. Lewy, "Über die partiellen differenzengleichungen der mathematischen physik," *Mathematische annalen*, vol. 100, no. 1, pp. 32–74, 1928.

[27] H. Jasak, "Error analysis and estimation for the finite volume method with applications to fluid flows," Ph.D. dissertation, Imperial College of Science, Technology and Medicine, 1996.

[28] R. I. Issa, "Solution of the implicitly discretised fluid flow equations by operator-splitting," *Journal of Computational Physics*, vol. 62, no. 1, pp. 40–65, 1986.

[29] S. Bnà, I. Spisso, M. Olesen, and G. Rossi, "Partnership for Advanced Computing in Europe PETSc4FOAM : A Library to plug-in PETSc into the OpenFOAM Framework," Partnership for Advanced Computing in Europe, Tech. Rep., 2020.

[30] S. B. Pope, "Ten questions concerning the large-eddy simulation of turbulent flows," *New Journal of Physics*, vol. 6, 2004. [Online]. Available: http://dx.doi.org/10.1088/1367-2630/6/1/035

[31] Xavier, D., "Uncertainty quantification for time varying quantities in turbulent flows," Ph.D. dissertation, KTH Royal Institute of Technology, 2024.

[32] C. Mockett, T. Knacke, and F. Thiele, "Detection of initial transient and estimation of statistical error in time-resolved turbulent flow data," in *8th International ERCOFTAC Symposium on Engineering Turbulence Modelling and Measurements (ETMM8)*, Marsielle, France, 2010.

[33] S. Russo and P. Luchini, "A fast algorithm for the estimation of statistical error in DNS (or experimental) time averages," *Journal of Computational Physics*, vol. 347, pp. 328–340, 2017.

[34] D. Xavier, S. Rezaeiravesh, and P. Schlatter, "Autoregressive models for quantification of time-averaging uncertainties in turbulent flows," *Physics of Fluids*, vol. 36, no. 10, p. 105122, 2024.

[35] S. Rezaeiravesh, C. Gscheidle, A. Peplinski, J. Garcke, and P. Schlatter, "In-situ estimation of time-averaging uncertainties in turbulent flow simulations," *Computer Methods in Applied Mechanics and Engineering*, vol. 433, p. 117511, 2025. [Online]. Available: http://dx.doi.org/10.1016/j.cma.2024.117511

[36] M. Montecchia, G. Brethouwer, S. Wallin, A. V. Johansson, and T. Knacke, "Improving LES with OpenFOAM by minimising numerical dissipation and use of explicit algebraic SGS stress model," *Journal of Turbulence*, vol. 20, no. 11-12, pp. 697–722, 2019. [Online]. Available: http://dx.doi.org/10.1080/14685248.2019.1706740

[37] T. Mukha, "Inflow generation for scale-resolving simulations of turbulent boundary layers," Ph.D. dissertation, Uppsala University, 2016.

[38] J. Meyers and P. Sagaut, "Is plane-channel flow a friendly case for the testing of large-eddy simulation subgrid-scale models?" *Physics of Fluids*, vol. 19, no. 4, p. 048105, Apr. 2007.

[39] S. Rezaeiravesh and M. Liefvendahl, "Effect of grid resolution on large eddy simulation of wall-bounded turbulence," *Physics of Fluids*, vol. 30, p. 055106, 2018. [Online]. Available: http://dx.doi.org/10.1063/1.5025131

[40] ——, "Grid construction strategies for wall-resolving large eddy simulation and estimates of the resulting number of grid points," Uppsala University, Department of Information Technology, Tech. Rep., 2017.

[41] M. Karp, R. Stanly, H. Song, T. Mukha, L. Galimberti, S. Toosi, L. Dalcin, S. Rezaeiravesh, M. Munsch, N. Jansson, S. Markidis, M. Parsani, S. T. Bose, S. K. Lele, and P. Schlatter, "Sensitivity of numerical simulations of turbulence to lower floating-point precision," in *Proceedings of the Center for Turbulence Reseach Summer Program 2024*, 2024, pp. 405–414.

[42] M. Chevalier, P. Schlatter, A. Lundbladh, and D. S. Henningson, "SIMSON : A pseudo-spectral solver for incompressible boundary layer flows," TRITA-MEK 2007:07, Royal Institute of Technology, Stockholm, Sweden, Tech. Rep., 2007.

[43] P. Lenaers, Q. Li, G. Brethouwer, P. Schlatter, and R. Örlü, "Rare backflow and extreme wall-normal velocity fluctuations in near-wall turbulence," *Physics of Fluids*, vol. 24, no. 3, p. 035110, 2012. [Online]. Available: http://dx.doi.org/10.1063/1.3696304

[44] J. A. Hopman, "runTimeChannelBudgets GitHub page," 2023. [Online]. Available: https://github.com/janneshopman/runTimeChannelBudgets

[45] P. R. Spalart, "Philosophies and fallacies in turbulence modeling," *Progress in Aerospace Sciences*, pp. 1–15, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.paerosci.2014.12.004

[46] S. Banerjee, R. Krahl, F. Durst, and Ch. Zenger, “Presentation of anisotropy properties of turbulence, invariants versus eigenvalue approaches,” *Journal of Turbulence*, vol. 8, p. N32, 2007. [Online]. Available: http://dx.doi.org/10.1080/14685240701506896

[47] J. L. Lumley and G. R. Newman, “The return to isotropy of homogeneous turbulence,” *Journal of Fluid Mechanics*, vol. 82, no. 1, pp. 161–178, 1977. [Online]. Available: http://dx.doi.org/10.1017/S0022112077000585

[48] K.-S. Choi and J. L. Lumley, “The return to isotropy of homogeneous turbulence,” *Journal of Fluid Mechanics*, vol. 436, pp. 59–84, 2001. [Online]. Available: http://dx.doi.org/10.1017/S002211200100386X

[49] S. Pirozzoli and M. Bernardini, “Turbulence in supersonic boundary layers at moderate Reynolds number,” *Journal of Fluid Mechanics*, vol. 688, pp. 120–168, 2011. [Online]. Available: http://dx.doi.org/10.1017/jfm.2011.368

[50] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, I. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, “SciPy 1.0: Fundamental algorithms for scientific computing in python,” *Nature Methods*, vol. 17, pp. 261–272, 2020. [Online]. Available: http://dx.doi.org/10.1038/s41592-019-0686-2

[51] Á. Tanarro, R. Vinuesa, and P. Schlatter, “Effect of adverse pressure gradients on turbulent wing boundary layers,” *Journal of Fluid Mechanics*, vol. 883, p. A8, 2020. [Online]. Available: http://dx.doi.org/10.1017/jfm.2019.838

[52] W. J. Baars, N. Hutchins, and I. Marusic, “Self-similarity of wall-attached turbulence in boundary layers,” *Journal of Fluid Mechanics*, vol. 823, p. R2, July 2017. [Online]. Available: http://dx.doi.org/10.1017/jfm.2017.357

[53] I. Marusic, R. Mathis, and N. Hutchins, “Predictive model for wall-bounded turbulent flow,” *Science*, vol. 329, no. 5988, pp. 193–196, July 2010. [Online]. Available: http://dx.doi.org/10.1126/science.1188765

[54] J. A. Hopman and E. M. A. Frederix, “RKSymFoam GitHub page,” 2023. [Online]. Available: https://github.com/janneshopman/RKSymFoam

[55] D. Santos, J. A. Hopman, C. D. Pérez-Segarra, and F. X. Trias, “On a symmetry-preserving unconditionally stable projection method on collocated unstructured grids for incompressible flows,” *Journal of Computational Physics*, vol. 523, p. 113631, 2025. [Online]. Available: http://dx.doi.org/10.1016/j.jcp.2024.113631