Predictive accuracy of wall-modelled large-eddy simulation on unstructured grids

Downloaded from: https://research.chalmers.se, 2025-12-04 10:15 UTC

Citation for the original published paper (version of record):
Mukha, T., Bensow, R., Liefvendahl, M. (2021). Predictive accuracy of wall-modelled large-eddy simulation on unstructured grids. Computers and Fluids, 221.
http://dx.doi.org/10.1016/j.compfluid.2021.104885

N.B. When citing this work, cite the original published paper.

research.chalmers.se offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all kind of research output: articles, dissertations, conference papers, reports etc. since 2004. research.chalmers.se is administrated and maintained by Chalmers Library

(article starts on next page)

Computers and Fluids 221 (2021) 104885

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Predictive accuracy of wall-modelled large-eddy simulation on unstructured grids

T. Mukha[a,*], R.E. Bensow[a,**], M. Liefvendahl[b]

[a] Chalmers University of Technology, Department of Mechanics and Maritime Sciences, Gothenburg 412 96, Sweden
[b] FOI, Totalförsvarets forskningsinstitut, Stockholm 164 90, Sweden

ARTICLE INFO

Article history:
Received 4 August 2020
Revised 2 January 2021
Accepted 12 February 2021
Available online 18 February 2021

Keywords:
Large-eddy simulation
Wall modelling
Unstructured grids
OpenFOAM
Channel flow
Turbulent boundary layer

ABSTRACT

The predictive accuracy of wall-modelled LES is influenced by a combination of the subgrid model, the wall model, the numerical dissipation induced primarily by the convective numerical scheme, and also by the density and topology of the computational grid. The latter factor is of particular importance for industrial flow problems, where unstructured grids are typically employed due to the necessity to handle complex geometries. Here, a systematic simulation-based study is presented, investigating the effect of grid-cell type on the predictive accuracy of wall-modelled LES in the framework of a general-purpose finite-volume solver. Following standard practice for meshing near-wall regions, it is proposed to use prismatic cells. Three candidate shapes for the base of the prisms are considered: a triangle, a quadrilateral, and an arbitrary polygon. The cell-centre distance is proposed as a metric to determine the spatial resolution of grids with different cell types. The simulation campaign covers two test cases with attached boundary layers: fully-developed turbulent channel flow, and a zero-pressure-gradient flat-plate turbulent boundary layer. A grid construction strategy is employed, which adapts the grid metric to the outer length scale of the boundary layer. The results are compared with DNS data concerning mean wall shear stress and profiles of flow statistics. The principle outcome is that unstructured simulations may provide the same accuracy as simulations on structured orthogonal hexahedral grids. The choice of base shape of the near-wall cells has a significant impact on the computational cost, but in terms of accuracy appears to be a factor of secondary importance.

© 2021 The Authors. Published by Elsevier Ltd.
This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

1. Introduction

The majority of the computational fluid dynamics (CFD) solvers used for tackling flow problems occurring in applications are based on finite volume discretization of the governing equations. As with other grid-based methods, the appropriate resolution and quality of the computational grid are crucial for obtaining an accurate solution. In the simplest of cases, the mesh can be fully defined by a single parameter prescribing the cell size. However, more often than not the domain of the simulation involves complex geometric features and discretizing it is a challenging and time-consuming task. The cells of a typical mesh vary in size, shape, degree of anisotropy, etc. Studies analysing and improving the accuracy of the finite volume method on such unstructured meshes can be found in the literature. This includes works on truncation error analysis [1,2], improving accuracy on particular types of grids [3], comparative evaluation of discretization scheme performance [4], and more practical studies looking at the grid effects on simulations of particular flows [5].

When the considered flow is turbulent, the appropriate grid construction strategy depends on the selected turbulence modelling approach. Currently, the workhorse of industrial CFD is Reynolds-averaged Navier Stokes (RANS). Here, when the flow is wall-bounded, it is considered best practice to cover the region occupied by the turbulent boundary layer (TBL) with prismatic layers. This way the cells are aligned with the surface of the wall and the steep velocity gradient can be properly resolved. In the rest of the domain the cell shape is not restricted. Previously, the most common approach was to use triangular (tri-)prisms for the TBL and tetrahedra in the rest of the domain. Recently, however, advances in meshing software made polygonal (poly-)prisms combined with arbitrary polyhedra an attractive option, due to the fact that this often results in a smaller number of higher-quality cells. Finally,

* Principal corresponding author.
** Corresponding author.
E-mail addresses: timofey@chalmers.se, timofey.mukha@protonmail.com (T. Mukha), rickard.bensow@chalmers.se (R.E. Bensow), mattias.liefvendahl@foi.se (M. Liefvendahl).

https://doi.org/10.1016/j.compfluid.2021.104885
0045-7930/© 2021 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

quadrilateral (quad)-dominant prismatic layers are also used, as well as hex-dominant meshes in the bulk flow.

Scale-resolving turbulence modelling approaches, such as direct numerical simulation (DNS) and large-eddy simulation (LES), have until recently mainly been used in academic studies of canonical flows. The domains of such flows are well-suited for discretization with a high-quality structured grid. This is reflected in existing guidelines for grid construction. For example, based on analysis of integral length scales, it is typically recommended that in the TBL the spanwise resolution should be about twice as high as the streamwise. Such a recommendation can be quite difficult to follow for a flow where the direction of the stream is subject to change, and for certain cell shapes may lead to deterioration of quality. The DNS/LES community also appears to hold a degree of scepticism towards attempts at using these simulation techniques in conjunction with second-order numerical methods and unstructured grids. As a result, the effects of the properties of unstructured grids on the predictive accuracy of DNS/LES have not been studied extensively. For example, the only publication to directly compare the accuracy of DNS across several grid types appears to be the study of Komen et al. [6]. Nevertheless, development of solvers for scale-resolving simulations on unstructured grids is an active area of research, see e.g. [7].

It can be argued that with some exceptions both DNS and LES are still rarely used for anything but fundamental studies of turbulent flows, mainly due to the rapidly scaling costs of LES for flows with TBLs. However, approaches that allow to alleviate this issue, such as hybrid LES/RANS and wall-modelled (WM)LES are used increasingly actively in applied research and industrial CFD. Unfortunately, for these methods the dependence of the accuracy on the grid construction strategy is often even less predictable than for standard DNS/LES. Establishing best practice meshing guidelines for these methods is therefore a pressing matter.

In the remainder of the paper we focus on grid construction strategies for WMLES. For a detailed review of this turbulence modelling approach the reader is referred to [8,9]. Several studies using WMLES with an unstructured grid can be found in the literature. In most of them, the idea of covering the TBL region with prismatic cells is adopted. In [10], WMLES is used to model a flow around a car model using tri-prism layers attached to the wall boundaries and tetrahedra in the bulk of the domain. The same grid type is used in [11] to study the flow around a cylinder, and a rod-arifoil configuration in [12]. Flow around a displacement ship hull in model scale is considered in [13]. Poly-prisms are used in the TBL region, and polyhedra elsewhere. The NASA Juncture flow is investigated in [14]. Here, a special grid construction approach referred to as 'Voronoi hexagonal close packed point-seeding method' is used. Based on the figures, the cells appear to be polyhedral. Generally, the flow quantities analysed in these studies agree favourably with the reference data, indicating that it is possible to use an unstructured WMLES grid and achieve accurate results.

Initial efforts on carefully assessing how the properties of the unstructured grid affect the predictive accuracy of WMLES have been reported in [15], which includes Mukha and Liefvendahl as co-authors. The work presented here can be seen as a significant extension of this study. The goal is to contribute towards finding the answers to the following questions. What is a suitable unstructured grid construction strategy for WMLES? What cell size metric is appropriate for comparing the resolution of grids consisting of cells of different shapes? How much, if at all, does the accuracy of WMLES deteriorate when an unstructured grid is used instead of structured orthogonal hexahedra? Excluding hexahedra, what cell shapes (or their combination) lead to the best accuracy?

Since these questions are very broad, here we limit the scope to attached TBLs, with fully-developed turbulent channel flow and a zero-pressure-gradient flat-plate TBL flow considered as test cases. A grid construction strategy suitable for producing a high-quality mesh for WMLES of such flows is presented in Section 2.2. Following the established practice, the boundary layer region is discretized with prismatic cells, with three candidate shapes for the base of the prisms considered: a quadrilateral, a triangle, and an arbitrary convex polygon. The corresponding grids are constructed to be equivalent with respect to an introduced grid resolution metric based on the average distance between the centre of a given cell and its neighbours, see Section 2.1. The effects of having part of the TBL discretized with arbitrary non-primsatic cells are also evaluated. For the case of channel flow, the effect of the resolution of the grid is also studied with 4 grid refinement levels considered.

To construct the grids we use Gmesh, Pointwise® and Simcenter Star-CCM+®. The former is open-source, and the latter two are widely used in industry. The open-source finite volume-based CFD software OpenFOAM® 4.1 is used to run the simulations, coupled with an open-source library for wall-modelling [16]. To ensure the reproducibility of the study and also to broaden its impact, the article is supplemented by a rich dataset[1], containing all the obtained results as well as ready-to-run OpenFOAM® cases, including the associated meshes.

A brief summary of the contents of the remainder of the paper follows. In Section 2, a metric for the resolution of the WMLES grid is proposed, and the employed grid construction strategy is presented. A short overview of surface meshing algorithms is given. Section 3 introduces the CFD methods that are used in the study. The results of the conducted simulations are presented and discussed in Section 4, and concluding remarks are given in Section 5.

2. Unstructured grid generation

This section discusses how an unstructured grid suitable for WMLES of an attached TBL flow can be constructed. In the first subsection, a metric for the grid resolution is introduced and related to properties of geometrical shapes that are commonly used in surface meshing. Next, a strategy for constructing a boundary layer mesh is presented. The final subsection is dedicated to the choice of the surface meshing algorithm.

2.1. Grid resolution metrics

A fundamental difficulty in comparing results from simulations using different grid cell shapes is defining a metric of grid resolution with respect to which the different grids are constructed. In this work, the average distance between the centre of a given cell and the centres of its neighbours, d, is employed to that end. Two grids with the same distribution of d are thus considered equivalent in terms of resolution. This approach is justified by the fact that d directly enters the discrete formulation of the governing equations and, therefore, the measured grid resolution is explicitly connected to the discretization error. Two alternative metrics that could be considered are the cell volume (or its cubic root) and the average length of the cell edges. The former is often used for the LES filter width, whereas the latter is a common input parameter to grid generation algorithms.

It is instructive to compare these three metrics for three fundamental two-dimensional cell shapes: an equilateral triangle, a square, and a regular hexagon. The relationships between the area

[1] DOI: 10.6084/m9.figshare.12482438.

Table 1
The relationship between the area S, the edge-size a, and the diameter of the inscribed circle d for an equilateral triangle, a square, and a regular hexagon.

	S(a)	S(d)	d(a)
Triangle	$\frac{\sqrt{3}}{4}a^2 \approx 0.43a^2$	$3\frac{\sqrt{3}}{4}d^2 \approx 1.30d^2$	$\frac{\sqrt{3}}{3}a \approx 0.57a$
Square	a^2	d^2	a
Hexagon	$3\frac{\sqrt{3}}{2}a^2 \approx 2.60a^2$	$\frac{\sqrt{3}}{2}d^2 \approx 0.87d^2$	$\sqrt{3}a \approx 1.73a$

S, the edge-size a, and d (which for these elements coincides with the diameter of the inscribed circle) are given in Table 1. It is evident that the choice of the resolution metric has a very significant effect on the comparative properties of the grids constructed using different elements. For example, for a given a, covering a particular surface will require ≈ 6 times more triangles than hexagons. However, if d is fixed, one would need approximately 1.5 more polygons than triangles instead!

In WMLES, the aim is to resolve the structures in the outer layer of the TBL, whereas the inner layer is accounted for by the wall model. The resolution of the grid should thus be computed with respect to the local turbulent boundary layer thickness δ. When d is used to characterize the cell size, the corresponding metric is thus δ/d. In the literature [8,17], the resolution is often measured as the number of cells within a δ^3-cube. A similar metric based on d is $(\delta/d)^3$, but, as discussed above, the number of cells this corresponds to heavily depends on the shapes of the cells constituting the grid.

2.2. Boundary layer meshing algorithm

Using unstructured grids makes it possible to accurately control d/δ even when δ varies in space. A reasonable choice is to keep it constant, so that the same level of turbulence resolution is maintained throughout the whole boundary layer. A high-quality mesh for the TBL region can be constructed by extruding the surface mesh to form prismatic layers. Based on these considerations, an unstructured grid generation algorithm suitable for WMLES has been introduced in [15]. It consists of the following steps.

1. Estimate the distribution of δ by running a RANS precursor or using approximative analytical relations, when such are available.
2. Construct a surface mesh with the chosen distribution of d/δ on the wall boundary.
3. Extrude prismatic layers between the wall and δ, with the number of layers equal to δ/d. The height of each prism is thus equal to d.
4. Apply suitable discretization to the rest of the computational domain.

Most of the grids used in this work are generated according to this procedure. In some channel flow simulations, however, only part of the boundary layer is covered with prismatic cells with the goal of analysing the effect of such a change on the predictive accuracy of the simulation. This is discussed in more detail in Section 4.1.

2.3. Selection of surface meshing algorithms

A crucial step in the mesh generation procedure defined above is generating a high-quality surface mesh that serves as basis for the extruded prismatic cells. In this work, three types of cells are considered for constructing the surface mesh: triangles (tris), quadrilaterals (quads), and arbitrary polygons (polys). For the latter two, indirect meshing methods [18] are used, meaning that the mesh generation consists of two steps: constructing a suitable triangular grid and then converting it into a mesh consisting of either quads or polys, respectively. In the case of a quad grid, recombination is applied to the triangles, whereas a polygonal grid is obtained as a dual of the triangulation.

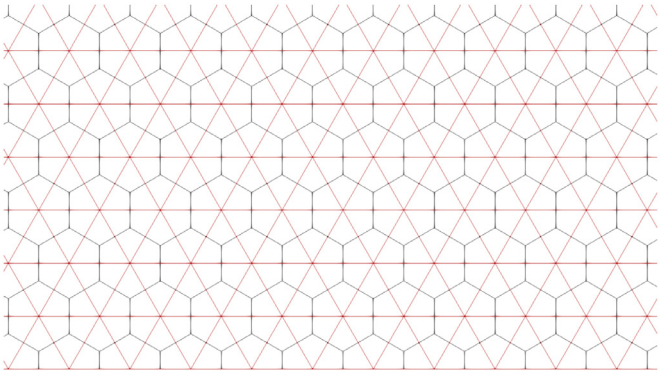

Fig. 1. A surface triangulation and its dual. The domain boundary is shown at the bottom.

Recombination refers to taking several grid cells and combining them into a single cell. Here, two triangles are combined into a single quad. This implies that a higher-quality grid is generated if the triangulation consists of right-angled triangles with catheti of equal length. In the ideal case, two such triangles sharing the hypotenuse can be recombined into a square. Then, the edge length of the quads, and consequently d, is equal to the length of the catheti in the underlying triangulation. An excellent algorithm [18] for generating triangulations suitable for recombination is implemented in the free mesh generation software Gmsh [19]. Furthermore, the recombination algorithm guarantees that the mesh consists only of quads [20].

A dual of a triangulation is a mesh in which the cell centres of the triangles are used as vertices, whereas the vertices of the triangles in turn become cell centres. The edges of the dual mesh are constructed by connecting the cell centres of triangles that share an edge. Special treatment is necessary at the boundary, where an edge may be built by connecting the cell centre of the triangle to the middle of the boundary edge. An illustration of a triangulation and the corresponding dual is given in Fig. 1. The figure also represents the case in which the triangulation is composed of equilateral triangles, which leads to a high-quality dual grid composed of regular hexagons.

Several algorithms for generating triangulations suitable for conversion to polygonal meshes were tested. In Gmsh, a Delaunay triangulation-based algorithm inspired by Frey and George [21] is implemented. The bottom plot in Fig. 2 shows a part of a flat surface mesh generated by this algorithm. The quality of the grid is quite uniform throughout the surface, but it is easily seen that the resulting dual grid will contain polygons that are far from regular and having a varying number of edges. Another algorithm that was considered is referred to as Frontal Delaunay in Gmsh, and based on the work of Rebay [22]. A grid generalized with Frontal Delaunay is shown in the top plot of Fig. 2. It mainly consists of equilateral triangles, however, elongated streaks of less regular triangles, highlighted with red lines in the figure, are also present. Nevertheless, the quality of the triangles forming the streaks is acceptable, and, consequently, Frontal Delaunay is considered to be more suitable when generating a dual mesh. Two algorithms included in the grid generation software Pointwise® have also been tested. The first one, referred to as Delaunay, produced meshes similar to the Delaunay algorithm in Gmsh. The second one, called Advancing Front, generates the mesh sequentially starting from the bound-

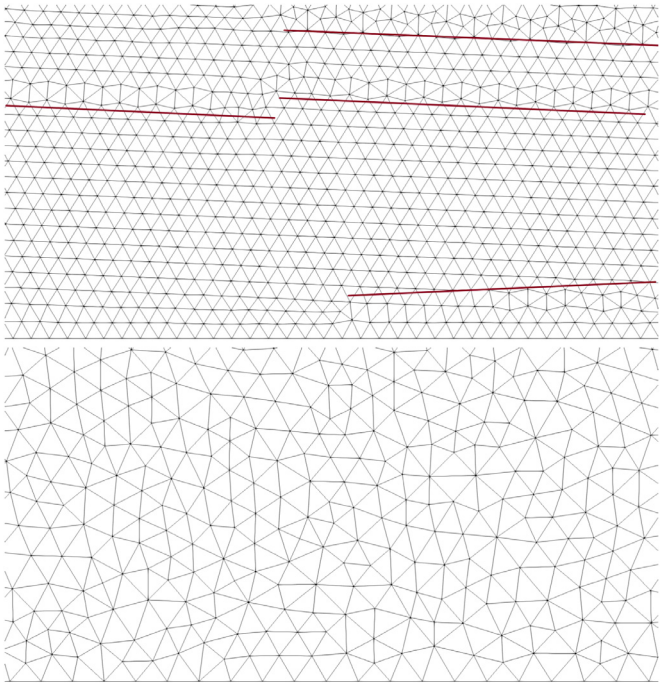

Fig. 2. *Top:* A triangulation generated with the Frontal Delaunay algorithm in Gmsh. Red lines highlight streaks of non-equilateral triangles. *Bottom:* A triangulation generated with the Delaunay algorithm in Gmsh. (For interpretation of the references to color in the text, the reader is referred to the web version of this article.)

aries. More information on these type of algorithms can be found in [23] and the references therein. Equilateral triangles dominate the generated grid, however, in places where two 'fronts' propagating from separate boundaries meet, triangles of slightly lesser quality are present. The result is thus quite similar to the grid produced by the Frontal Delaunay algorithm in Gmsh.

Since both Fontal Delaunay and Advancing Front generate a surface mesh dominated by equilateral triangles, these algorithms are well-suited not only for constructing a dual poly-mesh, but also for the case when the triangles themselves are used as the bases of the prisms.

It is noted that besides for the quality of the mesh, the algorithms also differ in terms of execution speed and robustness. In this work, planar wall surfaces are considered, discretized using a relatively low number of cells ($\lessapprox 5 \cdot 10^5$), which makes these factors unimportant. However, in more practical cases they can become critical.

3. Computational fluid dynamics methods

In this work, the open source CFD software OpenFOAM® version 4.1 is used to solve the LES equations. OpenFOAM® is based on cell-centred finite-volume discretization, and, crucial for this study, grids consisting of arbitrary convex polyhedral cells are supported. The flow is considered incompressible and no explicit filtering is applied to the unknowns, i.e. the velocity, and pressure fields. Instead, the equations are considered to be implicitly filtered by the discretization procedure. The WALE model [24] is selected for modelling the subgrid scales based on its favourable performance in WMLES on structured meshes [25,26].

For wall-modelling, a separate publicly-available library is employed [16]. An algebraic wall model based on Spalding's law-of-the-wall [27] is used. This model is accurate for the flow cases considered here, since they are free of strong pressure gradients, and the TBLs always remain attached. For each wall-face and at each time-step the wall model samples the value of velocity from the LES solution at some distance h from the wall, and uses the law-of-the-wall to compute an estimate of the wall shear stress, τ_w (or equivalently the friction velocity, $u_\tau = \sqrt{\tau_w}$)[2] The value of h has a significant effect on the accuracy of the wall shear stress prediction [28], in particular, it is necessary to sample the velocity further out than the wall-adjacent cell. Here, h is always set to the distance to the centre of the second consecutive off-the-wall cell.

An important issue is the choice of the interpolation scheme for the convective flux. Here, a weighted linear blending of linear interpolation and second-order upwind interpolation is used. The weights of the two schemes are, respectively, 0.75 and 0.25. Both of them are second-order accurate and unbounded, however the numerical diffusion introduced by the upwind scheme is enough to restrict the effect of any introduced non-physical oscillation. This treatment of the convective terms has been shown to provide good results in previous WMLES studies on hexahedral meshes [16,25]. Furthermore, it has proven to be stable enough for simulations on unstructured meshes presented in this work.

The other components of the employed numerical approach are as follows. The diffusive fluxes are interpolated linearly, with an explicit correction applied to reduce errors caused by mesh non-orthogonality, see [29, p. 83]. A second-order backward-differencing scheme is used for integration in time, see e.g. [29, p. 91] for the definition. The PISO algorithm [30] is used for pressure-velocity coupling, with 3 coupling iterations per time-step.

In the simulations of a flat-plate TBL flow presented in Section 4.2, it is necessary to trigger the transition to turbulence. Random volumetric forcing is used to that end. The magnitude of the introduced source term, S, is set according to the following in each cell

$$S = \frac{U_{ref} I_{ref}}{\Delta t} \alpha_r. \tag{1}$$

Here, U_{ref} and I_{ref} are reference values of velocity and turbulent intensity, Δt is the time-step, α_r is a random number sampled from a uniform distribution covering the interval $[-0.5, 0.5]$. The source term is to be applied similarly to a trip-wire in a physical experiment, i.e. in a thin strip of cells close to the wall.

4. Numerical experiments

In this section, results from WMLES of two flows are presented: fully developed turbulent channel flow and flat-plate zero-pressure-gradient TBL flow. Both flows are simulated using several types of unstructured grids, and in the case of channel flow different resolutions of the grids are also considered. The focus of the analysis is kept on the comparison between the results using different grids rather than the accuracy of WMLES in general and the associated trends in the error profiles. A detailed discussion of the latter the reader is referred to [25]. In that work, the employed methodology is tested with respect to a large variety of model parameters, including grid resolution, distance to the sampling point, grid anisotropy, and subgrid scale modelling. The simulation settings chosen here follow the best-practice guidelines developed in that study.

The triple u, v, w will be used to denote, respectively, the streamwise (x), wall-normal (y) and spanwise (z) components of (the implicitly-filtered) velocity. Angular brackets are used to denote the average value in time and across statistically homoge-

[2] In the governing equations the pressure gradient and viscous stresses (including τ_w) are normalized by the fluid density. Following standard practice, we will nevertheless use the terms 'pressure' and 'stress' when referring to the corresponding mass-specific quantities in the remainder of the text.

neous spatial directions, and the prime symbol is used to denote the fluctuations with respect to the average.

4.1. Fully developed turbulent channel flow

4.1.1. Simulation set-up

As the first test case, fully developed turbulent channel flow at $\mathrm{Re}_b = 125\,000$ is considered. Here, $\mathrm{Re}_b = U_b\delta/\nu$ is the bulk Reynolds number of the flow, with $U_b = 1$ m/s denoting the bulk velocity, $\delta = 1$ m the channel half-height, and $\nu = 8 \cdot 10^{-6}$ m^2/s the kinematic viscosity. The corresponding friction velocity-based Reynolds number, $\mathrm{Re}_\tau = \langle u_\tau \rangle \delta/\nu$, is ≈ 5200 according to the DNS data [31], which is used here as reference for computing errors in the obtained profiles of the flow quantities. To set Re_b, a time-dependant source term is added to the streamwise momentum equation, which adjusts the velocity field to match the prescribed value of U_b.

The computational domain is of size $8\delta \times 2\delta \times 6\delta$. The use of unstructured grids makes periodic boundary conditions more complicated to implement since there is no one-to-one mapping between the faces of the two patches coupled by periodicity. In OpenFOAM®, the so-called Arbitrary Mesh Interface (AMI) can be used to rectify this issue. The AMI computes the face connectivity between the two patches and ensures that the numerical fluxes are mapped conservatively.

Solution fields for the same flow obtained in previous studies were mapped onto the constructed grids to serve as initial conditions. To remove any associated artefacts, all simulations were first run for $300T$, where $T = \delta/U_b$ is a characteristic time scale for the outer layer. Afterwards, the gathering of statistical data was started, and the simulations were allowed to run for another $1000T$. In all the simulations, the time-step Δt was set to $5 \cdot 10^{-3}T$. The temporal resolution of turbulent motions is thus grid-independent. All the parameters of the simulations are summarized in Table 2.

Table 2
Parameters of the channel flow simulations.

Bulk velocity, U_b	1 m/s
Kinematic viscosity, ν	$8 \cdot 10^{-6}$ m^2/s
Channel half-height, δ	1 m
Bulk Reynolds number, Re_b	125 000
Target friction Reynolds number, Re_τ	5 200
Domain size, $L_x \times L_y \times L_z$	$8\delta \times 2\delta \times 8\delta$
Characteristic time scale, T	1 s
Time-step, Δt	$5 \cdot 10^{-3}T = 5 \cdot 10^{-3}$ s
Averaging time	$1000T$, 1000 s

4.1.2. Grid generation

In terms of the grid generation procedure defined in Section 2.2, channel flow can be seen as a limiting case of two TBLs that together occupy the whole domain and have a constant thickness δ equal to the channel half-height. As consequence, the prismatic layers should occupy the whole domain. Thus, given the surface mesh for the bottom wall, a three-dimensional grid can be obtained simply by building the appropriate number of layers between 0 and 2δ in the wall-normal direction. The utility `extrudeMesh` included in OpenFOAM® was used to that end.

As mentioned in Section 2.3, tri-, quad- and poly-prism layers are considered in the study. Note that in the case of channel flow, the quad-prism grid reduces to a structured hexahedral (hex) one. The results from simulations on this mesh type are used to define a reference point for the accuracy that can be expected from the WMLES. Additionally, Simcenter Star-CCM+® was used to generate a grid in which poly-prisms extend only up to 0.2δ, with the rest of the domain meshed using arbitrary polyhedra. This grid is referred to as poly-hybrid. It is interesting to see if the simulation using the poly-hybrid grid will exhibit significant deterioration of accuracy, because in practical situations δ is known only approximately, and consequently a part of the TBL can be expected

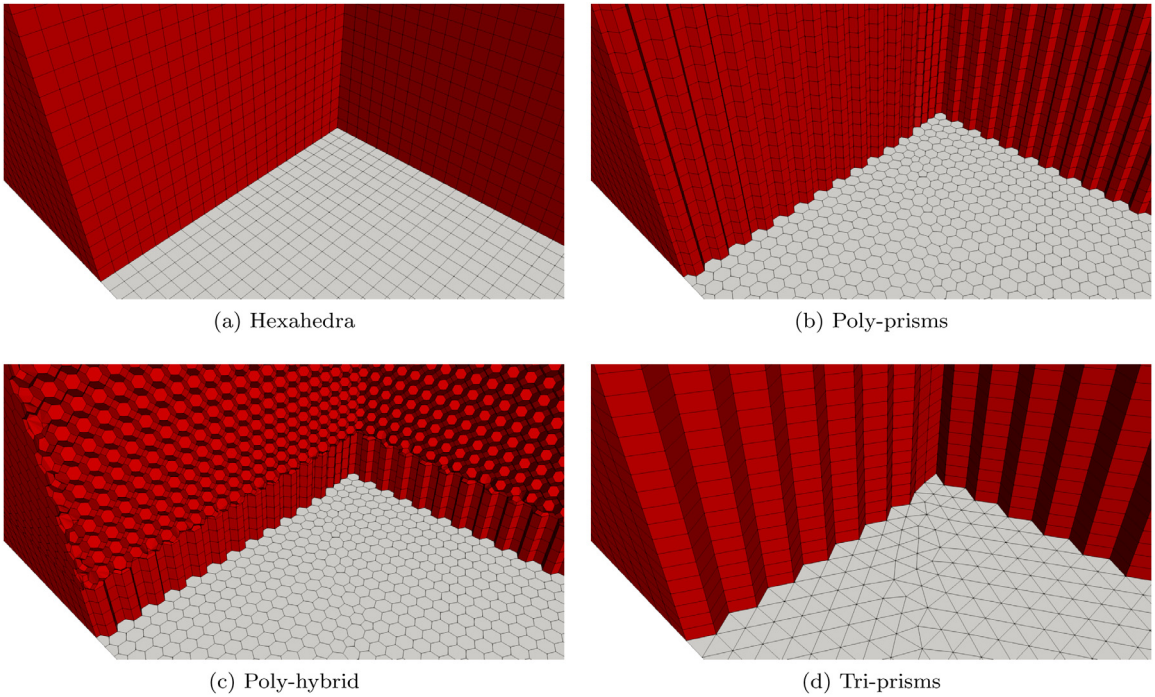

Fig. 3. Grid types used in channel flow simulations. In each plot, the grey elements indicate the wall-surface grid, and the volume grid is cut for visualization. The grids shown in the figure correspond to $d = 1/20$.

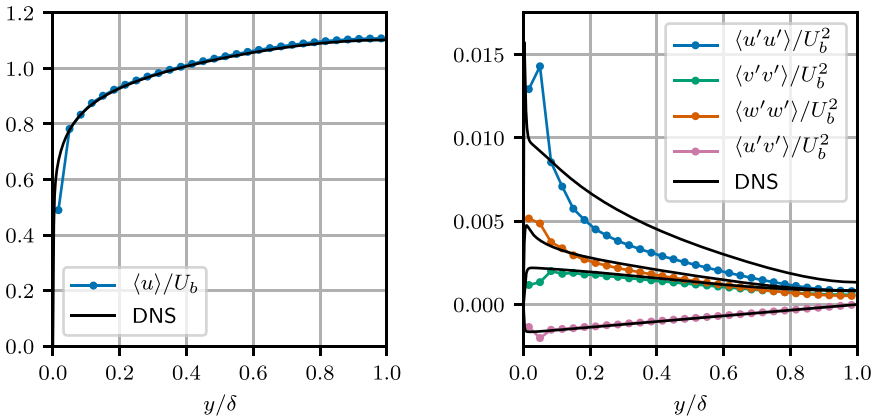

Fig. 4. Results from the simulation on the $d = \delta/30$ hex mesh and the reference DNS data.

Table 3
The number of cells, N_{cells}, in millions, in the grids used for channel flow simulations, and the number of hours, N_{hours}, it would take to advance each simulation by one T on a single processor.

d	h/δ	Hex		Poly-prisms		Poly-hybrid		Tri-prisms	
		N_{cells}	N_{hours}	N_{cells}	N_{hours}	N_{cells}	N_{hours}	N_{cells}	N_{hours}
$\delta/15$	0.1	0.365	0.25	0.424	0.36	0.554	0.84	0.287	0.15
$\delta/20$	0.075	0.864	0.65	1.000	1.69	1.283	2.26	0.652	0.39
$\delta/25$	0.065	1.688	1.55	1.940	2.03	2.781	4.44	1.282	0.91
$\delta/30$	0.05	2.916	2.33	3.356	2.53	4.489	7.88	2.218	1.59

to end up outside the prismatic layers. The same situation could also arise when the geometry is so complex that extruding 15–30 layers is impossible, but 3–5 layers could be constructed. An illustration of the four different grid types considered is provided in Fig. 3.

Besides for different cell shapes, different densities of the grids are considered: $d = \delta/15$, $\delta/20$, $\delta/25$, and $\delta/30$. Thus, simulations were performed using a total of 16 different grids. The number of cells, N_{cells}, in each grid is given in Table 3. It is easy to verify that for the hex, tri- and poly-prism grids the sizes follow the analytical relationships presented in Section 2, meaning that the triangles are close to equilateral, and the polys are predominantly regular hexagons.

A measure of the cost of the simulations, N_{hours}, is also reported in Table 3. To obtain N_{hours}, log files were first used to compute the wall-clock time, in hours, it took to advance each simulation by one T. These numbers were than multiplied by the number of processors each respective simulation was run on. Thus, N_{hours} is an estimate of how many hours it would take to advance each simulation by one T if the simulations were to be run in serial. Note that the metric is exact only under the assumption of linear scaling of computational effort with the number of cores. The scaling tests previously performed on the hardware used to run the simulations showed that within the range of core numbers employed in the study this assumption holds very well. The data in Table 3 shows clearly that the cell topology has a large effect on N_{hours}. This is not only due to the difference in N_{cells}, but also due to the fact that the number of non-zeros in the system matrix scales with the average number of faces of each cell. This is particularly vivid when looking at the values for the poly-hybrid grid. For instance, on the coarsest level of resolution, the number of cells for this grid is comparable to that of the poly-prism one, but the cost is more than twice as large, $N_{\text{hours}} = 6.66$ versus 2.93. The simulations on the tri-prism grid are, on the other hand, by far the cheapest. For example, the simulation on the $\delta/30$ tri-prism mesh is less expensive than on the $\delta/20$ poly-prism mesh.

Table 4
Relative error in Re_τ in simulations of turbulent channel flow.

d/δ	Hex	Poly-prisms	Poly-hybrid	Tri-prisms
1/15	0.24	−0.22	0.37	−3.24
1/20	−0.49	−0.77	−0.65	−3.12
1/25	−1.14	−1.40	−1.05	−4.31
1/30	−1.66	−1.85	−1.32	−4.70

4.1.3. Results

The results of the simulations are now presented and discussed. For the majority of the quantities of interest, the obtained profiles will be presented in terms of the relative error, in percent, with respect to the reference DNS. For example, for the mean streamwise velocity the profiles of

$$\epsilon[\langle u \rangle / U_b] = \frac{[\langle u(y) \rangle / U_b]_{WMLES} - [\langle u(y) \rangle / U_b]_{DNS}}{[\langle u(y) \rangle / U_b]_{DNS}} \cdot 100$$

will be shown. Since the size of WMLES cells are quite large at low δ/d, the reference value at each cell centre is obtained by averaging the DNS profile across the wall-normal extent of the corresponding cell. The primary reason for not showing the profiles of the quantities themselves instead is that data corresponding to different simulations would be very difficult to distinguish in the plots. However, to equip the reader with better intuition regarding the obtained level of predictive accuracy, scaled profiles of the first- and second-order velocity statistics are shown in Fig. 4 for the simulation conducted on the densest hex grid.

The performance of the wall model is now evaluated by considering the relative error in the obtained values of Re_τ, see Table 4. Comparing the values corresponding to different meshes, the simulation on the tri-prism grid stands out as the least accurate, whereas using the other three grids results in a similar level of agreement with DNS, with the relative error not exceeding 2%. No monotonous improvement of the predictions of Re_τ with the density of the grid is observed. However, this is to be expected. As extensively discussed in previous works [16,25], for channel flow

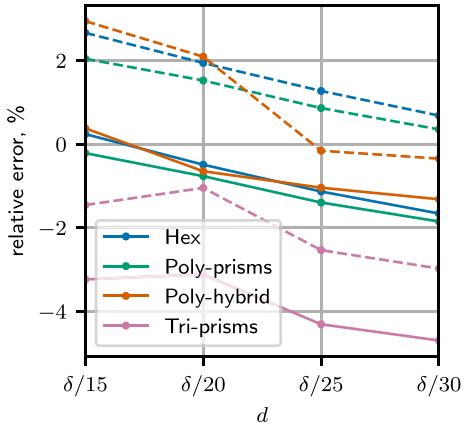

Fig. 5. *Solid lines:* Relative error in Re_τ. *Dashed lines:* relative error in the mean streamwise velocity at the sampling point.

the error in the wall model's predictions is heavily correlated with the error in the velocity at the sampling point. In Fig. 5, both errors are plotted and indeed a clear connection between the two quantities can be observed. Thus, for the error in Re_τ to improve with mesh refinement, convergence in $\langle u \rangle$ at the sampling point is necessary. Since in the current simulations the sampling point is fixed to the second off-the-wall cell, its location shifts linearly towards the wall as the mesh gets refined. The characteristic size of the eddies in the log-law region, in turn, also scales linearly with the distance from the wall, meaning that their resolution at the sampling point remains effectively constant with mesh refinement. For a more detailed discussion see [28].

Fig. 6 shows the relative error profiles of the outer-scaled mean velocity, $\epsilon[\langle u \rangle / U_b]$. Several trends previously reported in studies on hexahedral grids are observed. The least accurate solution, heavily under-predicting DNS data, is always obtained in the wall-adjacent cell (the data point lies outside the ordinate range in the figure). This is the fundamental reason behind not using this cell for sampling input into the wall model. Excluding this point, the errors for all considered mesh resolutions and grids do not exceed 3%. For all four grids, some degree of convergence of the profiles with mesh refinement is present. On the finest grids errors above the wall-adjacent cell do no exceed 2%. Overall, the solution on the hexahedral grid is more accurate than on the unstructured grids, but only slightly. The least accurate profiles are obtained using the tri-prism grid, in particular for low values of δ/d.

An alternative way of looking at the data is to compute the maxima of the error profiles across the outer layer,

$$\epsilon_{\max}[\langle u \rangle / U_b] = \max_{y \in [0.2\delta, \delta]} |\epsilon[\langle u \rangle / U_b]|, \tag{2}$$

and plot them against the simulation cost metric, N_{hours}. This is shown in Fig. 7. Under this perspective, the convergence towards DNS with grid refinement is better elucidated. Additionally, it is easy to see which grid type provides the best solution at a particular cost. It is clear that the hex-grid constitutes the optimal choice. The tri-prism grid comes second. Even though for a given resolution the results on this grid are worse than that on the poly-prism grid, the simulations are significantly cheaper, making it possible to achieve better accuracy for an equivalent cost. The poly-hybrid grid stands out as the least attractive alternative due to the high cost of the simulations.

Attention is now turned to the second-order moments of the velocity field. It is first noted that, in the outer layer, $\langle u'v' \rangle$ is not

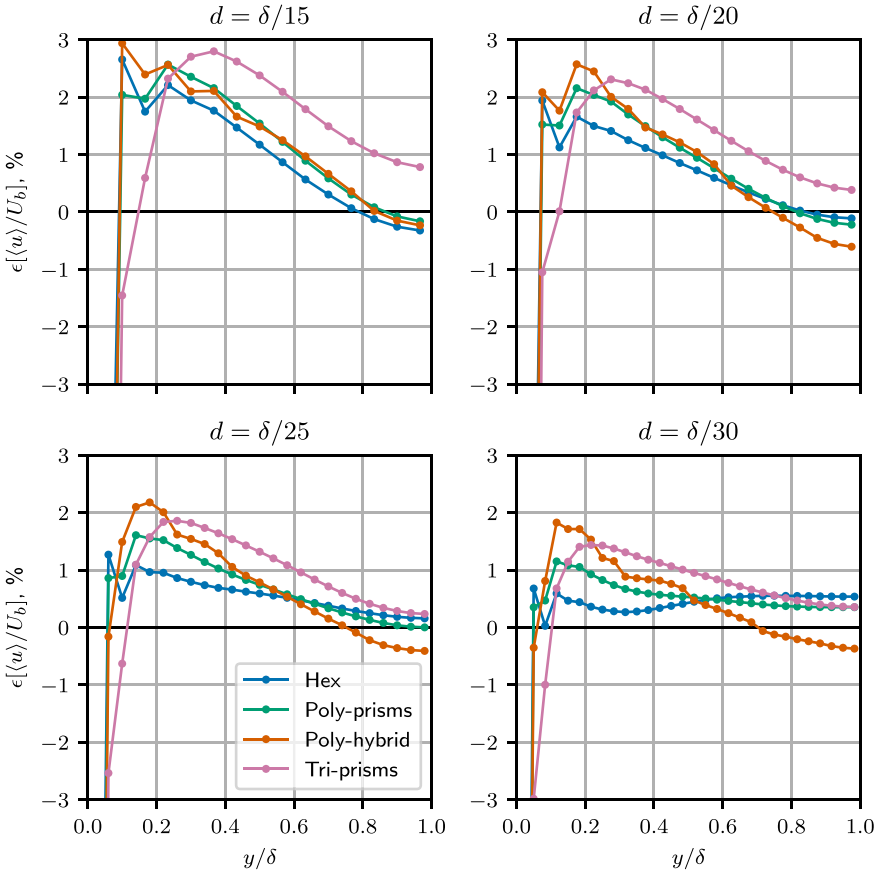

Fig. 6. Relative error in the outer-scaled mean velocity profiles, $\epsilon[\langle u \rangle / U_b]$, in simulations of turbulent channel flow.

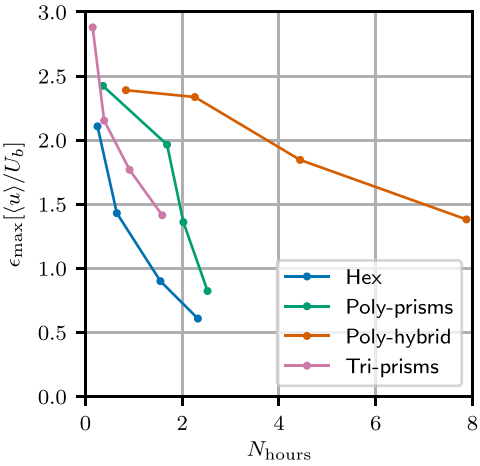

Fig. 7. Maximum error in the outer-scaled mean velocity profiles in simulations of turbulent channel flow, $\epsilon_{\max}[\langle u\rangle/U_b]$, plotted against the simulation cost.

an independent quantity in the case of channel flow. In particular [32], $\langle u'v'\rangle \approx \langle u_\tau\rangle^2(y/\delta - 1)$, so in the prediction of $\langle u_\tau\rangle$ determines the accuracy of $\langle u'v'\rangle$. For this reason, we do not further consider this quantity.

The error profiles in the resolved turbulent kinetic energy are shown in Fig. 8. As expected based on previous works, the overall level of accuracy is significantly lower than what could be obtained for the mean velocity profile. The largest errors occur in the inner region ($y \lessapprox 0.2$), where k is heavily over-predicted. Farther from the wall, under-prediction occurs instead. Interestingly, the results on the structured grid are only better in the inner region, whereas in the core of the channel all the unstructured grids lead to a better result. Presumably, this is due to some fortuitous cancellation of errors.

Lastly, the power spectral density (PSD) of the velocity signal at $y/\delta \approx 0.5$ is considered. To estimate the PSD, Welch's method was employed. The sampled time-series of the velocity signal were divided into four segments with 50% overlap, and the Hanning window applied to each segment. A periodogram was then computed for each segment, and afterwards averaging across the four segments was applied. Taking advantage of the spanwise periodicity of the solution, several time-series were acquired, corresponding to different spanwise locations. To produce the final estimate of the PSD, an average across the estimates for each time-series was computed. In the context of turbulent flows, a similar procedure was used in, for example, [33].

The PSDs of all three velocity components were computed for each of the conducted simulations, however, the results can be conveyed by looking at the streamwise velocity only, since the other two components exhibit very similar qualitative behaviour. In the left plot of Fig. 9, the PSDs of u' from the four simulations on the poly-prism grid are presented. Virtually no difference in the results is observed in the frequency band corresponding to the most energetic eddies. The pronounced peaks that can be seen in this region are due to the periodicity of the solution in the streamwise direction. This is evident from the fact that the locations of the peaks are close to being multiples of the flow-through time (8 s). The effect of increased mesh resolution is visible in the [1, 10] frequency band. A portion of the band is occupied by the inertial range, as indicated by the $-5/3$-slope of the PSD. At higher

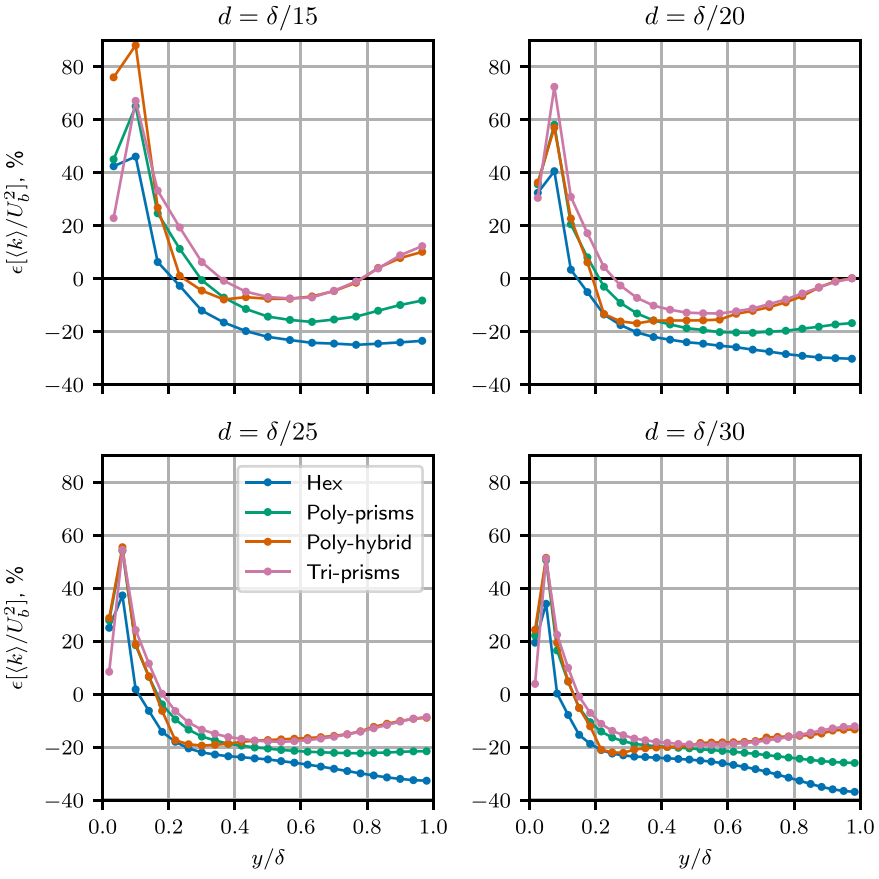

Fig. 8. Relative error in the outer-scaled turbulent kinetic energy profiles, $\epsilon[\langle k\rangle/U_b^2]$, in simulations of turbulent channel flow.

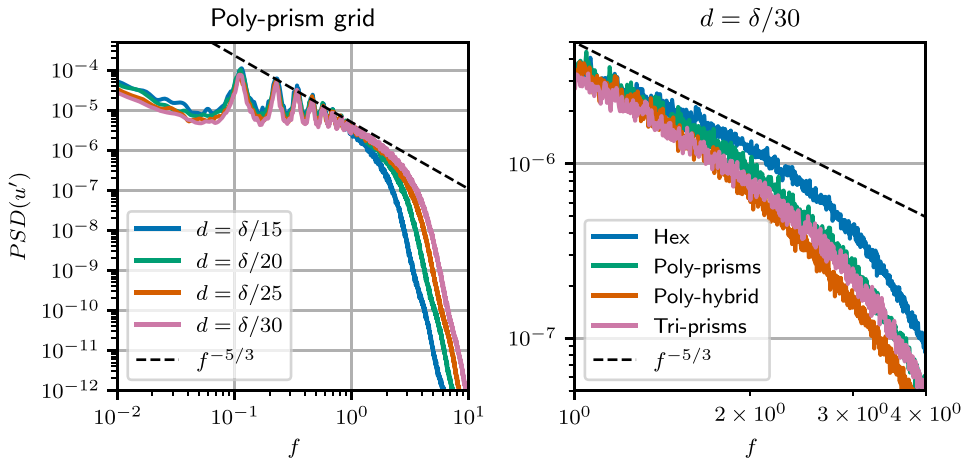

Fig. 9. Power spectral densities of the streamwise velocity fluctuations.

resolutions, the width of the inertial range increases due to the reduced amount of numerical dissipation. However, even on the finest grid, it is very narrow compared to what can be expected at this Reynolds number.

In the right plot of Fig. 9, the PSDs of u obtained in simulations on all the considered grid types are shown. For all four, the finest grid is considered, $d = \delta/30$. To be able to show the difference in the results, the plot focuses on a narrow frequency band containing the inertial range. One can see that, as expected, using the hexahedral grid leads to the least amount of damping. Among the unstructured grids the poly-hybrid grid is marginally more diffusive than the other two. However, the overall difference in the exhibited level of numerical diffusion among the considered grids is clearly very small.

4.1.4. Summary

A concise summary of the findings of the channel flow study is now given. The results on the four considered grid types are generally similar in their level of accuracy. This allows to conclude that the employed grid density metric, d, is well suited for this type of study. Further support for this is given by the PSD plots in Fig. 9, which show that the level of numerical diffusion is very similar on all the grid types.

Among the unstructured grids, the best results for a given d/δ were obtained using poly-prisms. The results on the tri-prism mesh stand out in their relatively poor prediction of the mean friction velocity (and thus Re_τ), which, in turn, transitions into an even larger error in the turbulent shear stress. Generally, the error in $\langle u \rangle$ is also somewhat larger when the tri-prism mesh is used. However, when the errors are compared with respect to the cost of the simulation, it turns out that the tri-prisms constitute the best choice, owing to the fact that the simulations on these grids are very cheap. Recall that the poly-hybrid grid was considered to evaluate whether covering a part of the TBL with non-prismatic cells would lead to a significant deterioration of accuracy. This turned out to not be the case, and the obtained accuracy is acceptable, even if subpar to that of the simulations on the poly-prism grid.

4.2. Flat-plate turbulent boundary layer

4.2.1. Simulation set-up

The set-up of the TBL simulations is identical to that used in [15]. A box of size $L_x = 2$ m, $L_y = 1$ m, $L_z = 0.2$ m constitutes the computational domain. At the inlet ($x = 0$), a uniform velocity profile $U_0 = 20.4$ m/s imposed, and transition is forced at $x = 0.1$ m by applying random volumetric forcing (1), as discussed in Section 3. The reference velocity and turbulence intensity parameters in (1) are set to U_0 and $0.025\,\mathrm{m}^2/\mathrm{s}^2$, respectively. The strip of cells where the source term is active is 2 cm long, 3 mm high and stretches across the whole domain in the spanwise direction. For the purpose of mesh generation, the flow is considered fully turbulent at $x_0 = 0.4$ m. This estimate is inserted into the following power law [34] to obtain the distribution of δ in the domain:

$$\mathrm{Re}_\delta = 0.1222\mathrm{Re}_x^{0.8628}. \tag{3}$$

Here $\mathrm{Re}_\delta = \delta U_0/\nu$ and $\mathrm{Re}_x = xU_0/\nu$, with $\nu = 1.65 \cdot 10^{-5}$ m^2/s. For $x < x_0$ the thickness is considered constant for simplicity, $\delta = \delta(x_0) \approx 8.1$ mm.

The maximum thickness of the TBL is predicted to be $\delta_{\max} \approx 3.2$ cm, which leads to $L_y/\delta_{\max} \approx 31.25$ and $L_z/\delta_{\max} \approx 6.25$. These ratios are sufficiently high to avoid negative effects due to employed boundary conditions, which are periodic in the spanwise direction and symmetric at the top boundary. At the outlet, a homogeneous Neumann condition is applied to velocity and the pressure is set to 0.

The time-step used in each simulation is $\Delta t = 2$ μs. For the initial conditions, the velocity and pressure fields from the WMLES of the same flow reported in [15] are used. Statistics were gathered over a period of 0.5 s, corresponding to $\approx 5T_{ft}$, where T_{ft} is the flow-through time. All the simulation parameters are summarised in Table 5.

4.2.2. Grid generation

The grid generation process follows the algorithm presented in Section 2.2, with quads, polys and tris considered for meshing the wall surface and arbitrary polyhedra used for meshing the freestream region. Downstream of the numerical trip ($x > 0.1$ m), the resolution of the grid is set to $\delta/d \approx 15$, where δ is approximated according to the power law, as discussed above. Unfortunately, higher resolutions could not be afforded due to limitations in the available computational resources. Upstream of $x = 0.1$ m,

Table 5
Turbulent boundary layer simulation parameters.

Estimated maximum TBL thickness, $\delta_{\max}$	3.2 cm
Length of the domain, L_x	2 m $= 62.5\delta_{\max}$
Height of the domain, L_y	1 m $= 31.25\delta_{\max}$
Width of the domain, L_z	0.2 m $= 6.25\delta_{\max}$
Freestream velocity, U_0	20.4 m/s
Kinematic viscosity, ν	$1.65 \cdot 10^{-5}$ m^2/s
Flow-through time, T_{ft}	0.1 s
Time-averaging period	0.5 s $= 5T_{ft}$
Time-step, Δt	$2 \cdot 10^{-6}$ s $= 2.04 \cdot 10^{-5} T_{ft}$

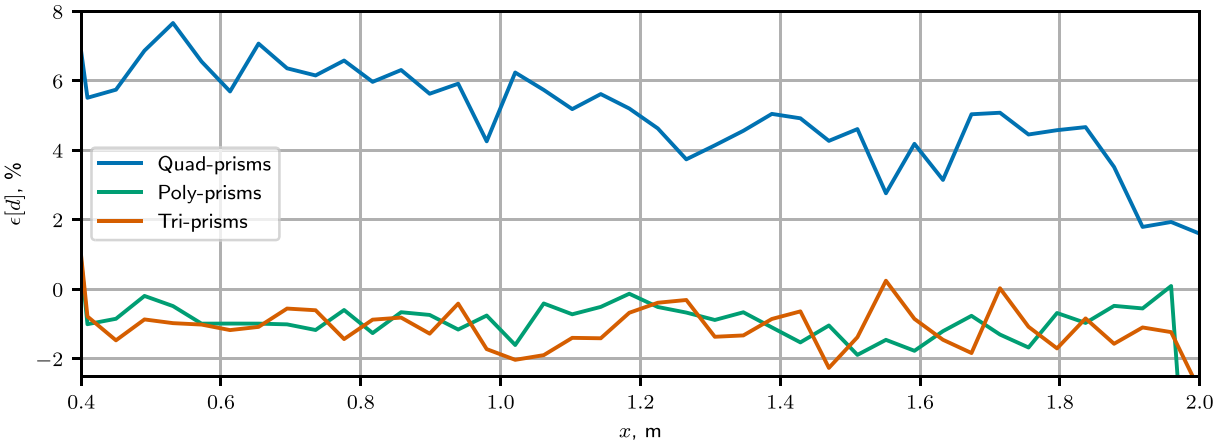

Fig. 10. The relative error in d with respect to the employed analytical estimate.

the mesh is rapidly coarsened, since the flow in that region is laminar and two-dimensional.

All three surface meshes were generated in Gmesh, using the algorithms discussed in Section 2. A particularly useful feature of Gmesh is the possibility to prescribe the distribution of the mesh size as an analytical functional relationship. By that means, a formula based on (3) and the chosen value of d/δ could be used as input to the meshing algorithms, resulting in a very smooth variation of d with x. The relative error in the distributions of d with respect to the analytical estimate is presented in Fig. 10. Unfortunately, for the quad-prism case there is, on average, a 5% over-prediction. This is most likely due to the fact that the input parameter to the surface meshing algorithm does not exactly correspond to the length of the catheti of the generated triangles. However, since the power law (3) and the choice of x_0 are also approximate, this deviation is considered acceptable. In fact, it will be shown below that the power law somewhat under-predicts δ, meaning that for the quad-prism case the ratio of d/δ is reproduced slightly more faithfully.

To generate the volume meshes, Star-CCM+® was used. The result is shown in Fig. 11, featuring a spanwise cut through each of the grids in order to expose their structure. To avoid possible confusion, it is stressed the quad surface mesh is unstructured, although a significant part of the wall surface is covered by regular squares. A major difficulty has been prescribing the variation of the height of the extruded prismatic layers with x. Unfortunately, supplying an analytical relationship as input is not supported, and the only straight-forward option to locally control the height is using 'volumetric controls'. Each control allows to specify the height's value within the boundaries of a particular shape, for example, a box. The approach used was then to define boxes partitioning the whole domain in the streamwise direction and prescribing the layer height in each box. As a result, a piecewise-constant distribution of the height following the analytical estimate could be achieved. To minimize the size of the jump in the height across neighbouring boxes, a large number of boxes, 160, was used. An important reason for why the variation of the layer height should be as smooth as possible is that the jump in the total height leads to a jump in the height of the first few cells, and one of them (here, the second), serves as the sampling point for the wall model. A jump in h, in turn, leads to a slight jump in the predicted wall shear stress. Here, the distribution of h is sufficiently smooth to avoid such kind of artefacts in the solution, see Fig. 13.

The total number of cells in each grid is provided in Table 6. Additionally, its decomposition into the cells covering the TBL and the freestream regions is given. Looking at the former, the expected grid size relationship between quad-, poly- and tri-prisms is observed: the tri-prism grid contains the least amount of cells, and the poly-prism the highest. For the poly- and tri-prism grids the freestream region is covered by ≈ 3 million cells, but for the quad-prism grid the number is higher, ≈ 4.78 million, which makes the total size of the quad- and poly-prism grids almost equal, with the quad-prism grid being slightly larger.

The cells in the freestream region are heavily clustered in a thin slice directly above the prismatic layers, see Fig. 11. In Star-CCM+®, their total number is heavily influenced by the 'volume growth rate' user input parameter, which defines the rate of transition in the cell volume from the prismatic layers towards the top boundary. The same value of the growth rate was prescribed for all three grids, but in spite of that more cells were generated in the case of quad-prisms. It is difficult to say whether this is an artefact of the mesh generation algorithm or if it is fundamentally more difficult to couple a generic polyhedral grid with a quad-dominated surface mesh. The important point is that the cells constituting the transition region from the TBL to the freestream may account for a significant portion of the mesh and care should be taken in defining the properties of the grid in this region. On the one hand, it is attractive to keep the amount of cells in the transition region to a minimum to reduce simulation time. On the other hand, it represents a buffer that could be useful in case the values δ are underestimated.

Table 6 also reports the simulation time metric N_{hours}. As expected, the simulation on the tri-prism grid is the cheapest. The quad- and poly-prism grid simulations N_{hours} is roughly the same. Recall that a good a-priori metric of the computational complexity is the number of faces. The prismatic layers of the quad-prism grid contain fewer faces than that of the poly-prism, but this difference is compensated by the larger number of cells in the freestream region, where the cells have a large number of faces due to their arbitrary polyhedral shapes.

Table 6
The total number of cells, $N_{\text{cells}}^{\text{total}}$, in millions, in the grids used for the TBL simulations. The corresponding number of prismatic cells covering the TBL region, $N_{\text{cells}}^{\text{TBL}}$, and the freestream region, $N_{\text{cells}}^{\text{FS}}$. The number of hours, in thousands, it would take to advance each simulation by one T_f on a single core, N_{hours}.

	$N_{\text{cells}}^{\text{total}}$	$N_{\text{cells}}^{\text{TBL}}$	$N_{\text{cells}}^{\text{FS}}$	N_{hours}
Quads	11.23	6.45	4.78	4.3
Polys	11.19	8.03	3.15	4.3
Tris	8.59	5.59	3.00	3.6

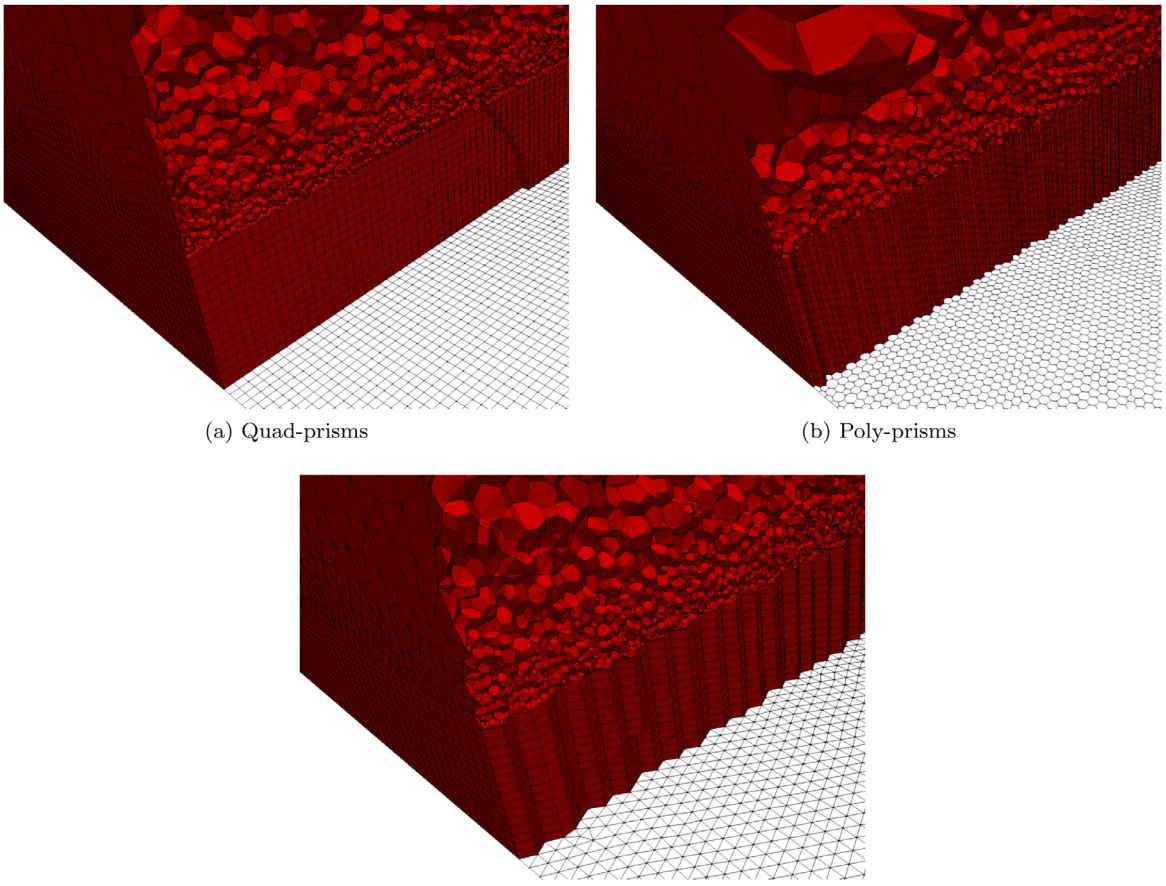

Fig. 11. Structure of the three grids used in the TBL simulations. In each plot, the white elements indicate the wall-surface grid, and the volume grid is cut for visualization.

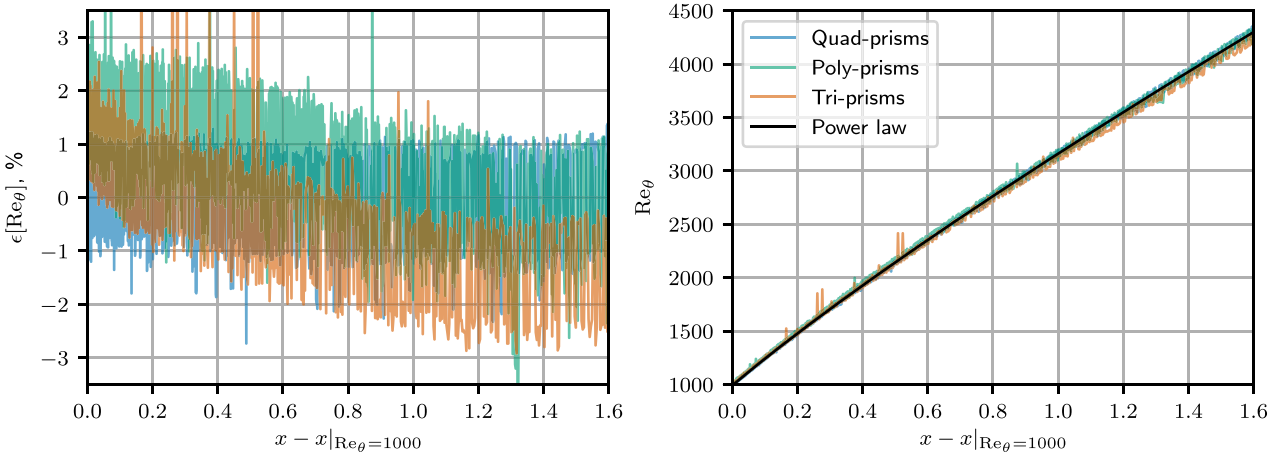

Fig. 12. *Left*: The relative errors in the values of Re_θ obtained in the TBL simulations, as compared to a power-law estimate. *Right*: The values of Re_θ obtained in the TBL simulations and the reference power-law estimate.

4.2.3. Results

The results of the conducted simulations are now presented. We note that, compared to the channel flow case, comparison with benchmark data is more difficult, because the outer scales of the flow are not defined by the geometry of the computational domain. As a consequence, the choice of the scales and the procedure to compute them can have a significant influence on the level of agreement with reference data. For the outer length scale, the most robust choice is generally considered to be the momentum thickness, θ. However, in the case of WMLES this is not necessarily the case, since computing θ involves integration across the whole mean velocity profile, which can exhibit erroneous values in the inner layer. Additionally, θ depends on the value of the free stream velocity U_0, which also serves as the outer velocity scale. In principle, U_0 is a fixed input parameter, however, in practice, the mean velocity profiles may locally exhibit values larger than U_0, and it is thus more robust to compute U_0 as function of x as the local maximum of $\langle u(y) \rangle$.

To simultaneously evaluate the procedures to compute the outer scales and the growth rate of the boundary layer, the evo-

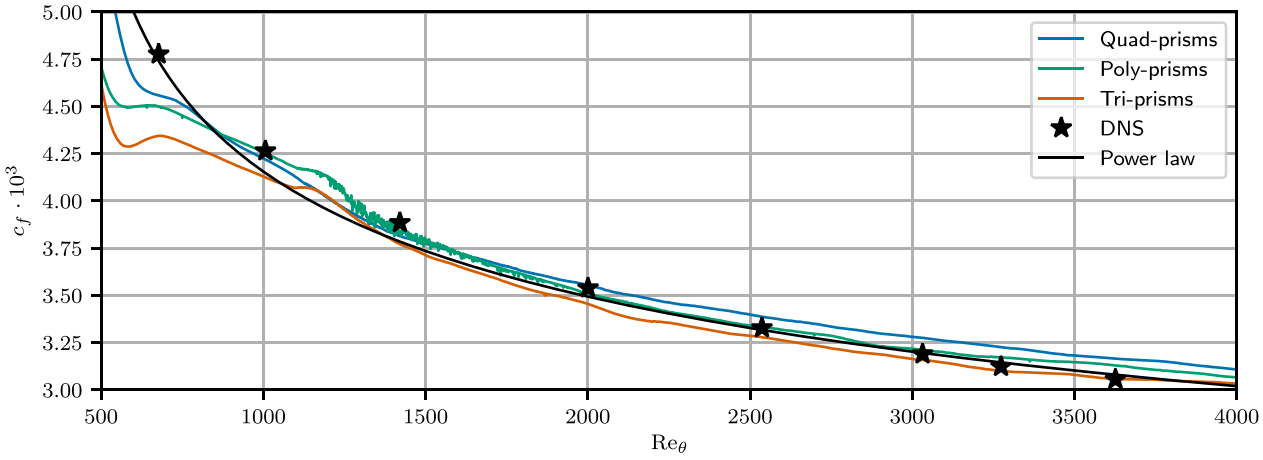

Fig. 13. The values of the skin-friction coefficient, c_f, obtained in the TBL simulations.

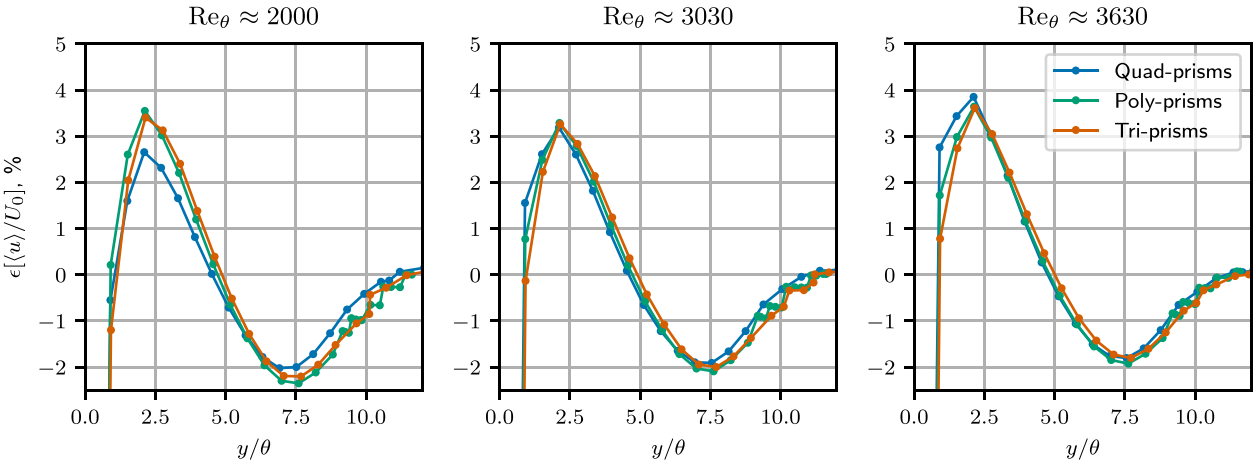

Fig. 14. Error profiles in the mean streamwise velocity in TBL simulations.

lution of $\mathrm{Re}_\theta = \theta U_0/\nu$ in the streamwise direction is considered. The following power-law estimate connecting Re_θ and Re_x is used as reference data [34]

$$\mathrm{Re}_\theta = 0.0167 \cdot \mathrm{Re}_x^{0.846} + 373.83. \tag{4}$$

To compare the growth across the three simulations, the origin of the x-axis is shifted to the point where $\mathrm{Re}_\theta \approx 1000$ for each of the simulations. The computed relative errors in Re_θ with respect to the power law are shown in Fig. 12. Due to the difficulties in robustly computing θ discussed above they exhibit high-frequency oscillations. The agreement with the power law is remarkably good, the magnitudes of the errors rarely exceed 3%. No significant dependency on the type of employed grid is observed. The values of Re_θ themselves along with the power-law estimate employed are shown in the right plot of Fig. 12.

The accurate prediction of Re_θ warrants using it as the independent variable when analysing the distributions of other flow quantities across the length of the plate. However, it is beneficial to first filter out the high-frequency errors in the obtained data so that they do not propagate into the subsequent results. To that end, the raw Re_θ values were least-square fit to a power-law function based on (4): $a_0 \cdot \mathrm{Re}_x^{0.846} + a_1$, where a_0 and a_1 are determined by the fit. In the remainder of this section, Re_θ will refer to the fitted values of the quantity.

Attention is now turned to the performance of the wall model, which is evaluated by considering the values of the skin-friction coefficient, c_f. The obtained values are shown in Fig. 13. The DNS data by Schlatter and Örlü [35] and a power law estimate

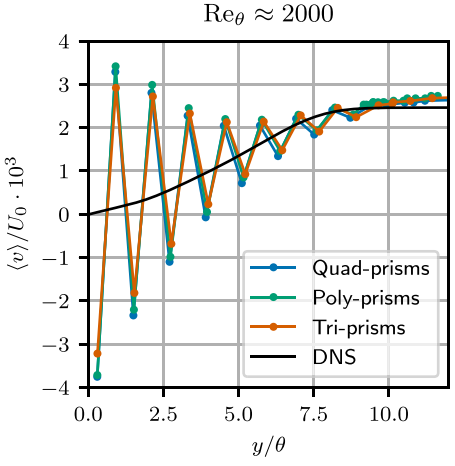

Fig. 15. Profiles of the mean wall-normal velocity in TBL simulations.

from [34] are used as reference. Excellent agreement with both is found. The results from all three grids exhibit a similar level of accuracy, however on the poly-prism grid some oscillations in the values of the wall-shear stress are observed up to $\mathrm{Re}_\theta \approx 2000$, as reflected in the corresponding c_f curve in the figure.

In Fig. 14, the obtained relative error profiles of the mean streamwise velocity are presented at selected values of Re_θ. DNS data [35] is used as reference, and similarly to the WMLES data, the velocity is normalized with its maximum value at the con-

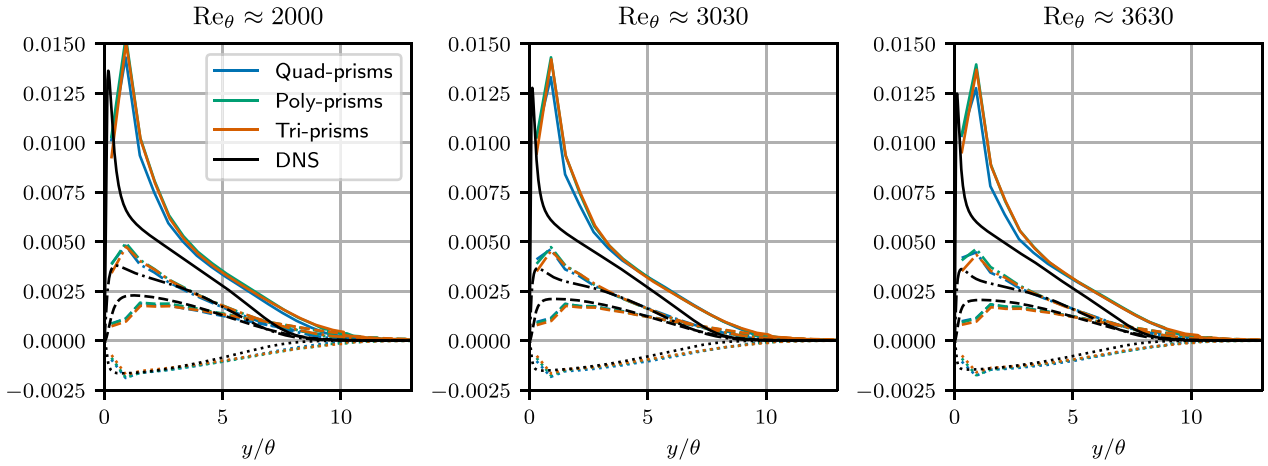

Fig. 16. Profiles on non-zero components of the Reynolds stress tensor obtained in the TBL simulations. Solid lines: $\langle u'u' \rangle$; dashed lines: $\langle v'v' \rangle$; dash-dotted lines: $\langle w'w' \rangle$; dotted lines: $\langle u'v' \rangle$.

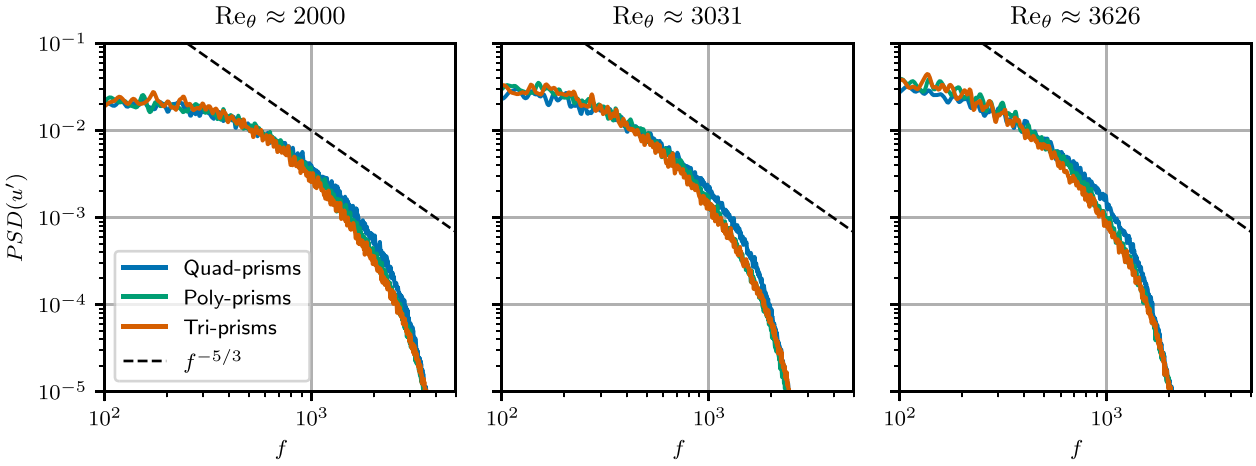

Fig. 17. Power spectral densities of the streamwise velocity in the TBL simulations.

sidered streamwise location. Consequently, the values of the errors go towards zero at the edge of the TBL. The curves from the three simulations are very similar. Inspection of the values close to the free stream reveals slight oscillations in the solution. They are caused by the fact that the boundary layer extends above the prismatic layers, and represent a significant contribution to the oscillations in θ along with the errors in the inner layer. Agreement with the DNS is reasonably good and comparable to what is reported for channel flow at a similar resolution, see Fig. 6. The error pattern is also similar. The reference data is first over-predicted and towards the edge of the TBL under-prediction is observed instead.

Unlike channel flow, in the case of the TBL the mean wall-normal velocity is also non-zero, although several orders of magnitude smaller than the streamwise component. Unfortunately, this quantity could not be predicted reliably in the conducted WMLES, and the obtained fields exhibit heavy oscillations, see Fig. 15. It is speculated that a denser grid is necessary to diminish them. It is interesting that in spite of these oscillations the growth of the TBL could be predicted correctly.

The profiles of the components of the Reynolds-stress tensor are analysed next. We do not consider the relative errors here due to the fact that the stresses decay to zero quite fast towards the edge of the TBL, and the relative error becomes a poor metric. The obtained results are shown in Fig. 16. As with the other quantities, the accuracy appears to be independent of the type of grid employed. The least accurately predicted component is $\langle u'u' \rangle$, whereas the other two diagonal components are captured quite well. The turbulent shear stress exhibits a different error pattern compared to channel flow. This is due to the fact that here $\langle u'v' \rangle$ is not directly coupled to the values of u_τ. The near-wall peak is predicted quite accurately, but the values further above are slightly under-predicted.

Finally, the PSDs of the streamwise velocity fluctuations are considered. The same computational procedure as for channel flow has been used to obtain the estimates. Fifty probes uniformly distributed along the spanwise direction were used to collect the data for each considered streamwise location. The wall-normal position of the probes is $y \approx 0.5\delta(x)$. The results are shown in Fig. 17, with the plots zoomed in on the inertial subrange. No substantial dependency of the PSD on the grid type is observed, although a marginally smaller level of dissipation is obtained on the quad-prism grid.

4.2.4. Summary

Largely, the outcomes of the TBL study confirm what was previously observed for channel flow. All three unstructured grids led to very similar results with respect to all the considered quantities. Agreement with WMLES results of the same flow reported in [15] is also good. Furthermore, even though flat-plate TBL flow is more difficult to predict than channel flow due to the distribution of δ being a simulation outcome instead of a geometric pa-

rameter, the achieved accuracy is similar. Since no grid type led to improved predictions, it can be argued that the selection should be made based on computational cost, meaning that the tri-prism grid constitutes the optimal choice. Finally, the inability of the WMLES to capture $\langle \nu \rangle$ raises a concern. It would be interesting to see if the issue persists at grid resolutions higher than $d = \delta/15$.

5. Conclusions

This study presents a careful evaluation of the predictive accuracy of WMLES on unstructured grids. To that end, we propose: *i)* A mesh construction strategy that relates the resolution of the mesh to the local outer scale of the flow. *ii)* A mesh resolution metric based on cell-centre distance that is suitable for comparative analysis of results obtained on meshes with different cell topology.

Results from channel flow and flat-plate TBL simulations clearly show that unstructured prismatic meshes, constructed according to the proposed strategy, can be used for WMLES without any significant impediment to accuracy as compared to results on structured hexahedral meshes.

Three candidates for the base shape of the prisms are considered in the study: triangles, quadrilaterals and hexagons. The choice of the shape did not have a significant impact on the results, although in the case of channel flow using triangular prisms did lead to slight deterioration of accuracy. Consequently, the selection of the shape can be guided by practical concerns, of which, arguably, the most important one is the simulation cost. From that perspective, triangular prisms constitute the superior choice because it minimizes both the number of cells and the number of faces per cell. Another important guideline for choosing the base shape is simply the quality of the produced grid for the geometry at hand. Here it was only necessary to discretize planar surfaces, therefore excellent quality was easy to achieve with all the considered shape options, but for a practical case the situation is likely to be different.

A possible direction of future work is to see how strongly the accuracy of the results deteriorates when the quality of the prisms is worsened, i.e. when the underlying surface mesh contains skewed and non-orthogonal cells. Also, robustness with respect to the dissipativity of the employed numerics would be interesting to study. Both to see whether good accuracy can still be achieved with more dissipative schemes, and whether simulations could be kept stable when less dissipation is introduced. The next step in the complexity of the flow problem would be TBLs over curved surfaces and affected by pressure gradients.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

T. Mukha: Conceptualization, Methodology, Investigation, Visualization, Data curation, Writing - original draft. **R.E. Bensow:** Funding acquisition, Resources, Supervision, Writing - review & editing. **M. Liefvendahl:** Conceptualization, Methodology, Supervision, Writing - review & editing.

Acknowledgements

This work was supported by grant number P38284-2 from the Swedish Energy Agency. The simulations were performed on resources at Chalmers Centre for Computational Science and Engineering (C3SE).

References

[1] Juretić F, Gosman AD. Error analysis of the finite-volume method with respect to mesh type. Numer Heat Transf Part B 2010;57(6):414–39. doi:10.1080/10407791003685155.
[2] Kallinderis Y, Fotia S. A priori mesh quality metrics for three-dimensional hybrid grids. J Comput Phys 2015;280:465–88. doi:10.1016/j.jcp.2014.09.036.
[3] Nishikawa H. A face-area-weighted centroid' formula for finite-volume method that improves skewness and convergence on triangular grids. J Comput Phys 2020;401:109001. doi:10.1016/j.jcp.2019.109001.
[4] Jalali A, Sharbatdar M, Ollivier-Gooch C. Accuracy analysis of unstructured finite volume discretization schemes for diffusive fluxes. Comput Fluids 2014;101:220–32. doi:10.1016/j.compfluid.2014.06.008.
[5] Ghoreyshi M, Bergeron K, Seidel J, Lofthouse AJ, Cummings RM. Grid quality and resolution effects for aerodynamic modeling of ram-air parachutes. J Aircr 2016;53(4):1087–105. doi:10.2514/1.C033391.
[6] Komen EMJ, Shams A, Camilo LH, Koren B. Quasi-DNS capabilities of OpenFOAM for different mesh types. Comput Fluids 2014;96:87–104. doi:10.1016/j.compfluid.2014.02.013.
[7] Mahesh K, Constantinescu G, Moin P. A numerical method for large-eddy simulation in complex geometries. J Comput Phys 2004;197(1):215–40. doi:10.1016/j.jcp.2003.11.031.
[8] Larsson J, Kawai S, Bodart J, Bermejo-Moreno I. Large eddy simulation with modeled wall-stress: recent progress and future directions. Mech Eng Rev 2016;3(1):1–23. doi:10.1299/mer.15-00418.
[9] Bose ST, Park GI. Wall-modeled large-eddy simulation for complex turbulent flows. Annu Rev Fluid Mech 2018;50:535–61. doi:10.1146/annurev-fluid-122316-045241.
[10] Aljure DE, Calafell J, Baez A, Oliva A. Flow over a realistic car model: wall modeled large eddy simulations assessment and unsteady effects. J Wind Eng Ind Aerodyn 2018;174:225–40. doi:10.1016/j.jweia.2017.12.027.
[11] Zhang C, Sanjosé M, Moreau S. Wall-modeled large eddy simulation with adverse pressure gradients: application to bluff bodies. CFD Canada 2018. Winnipeg, Canada; 2018.
[12] Zhang C, Sanjosé M, Moreau S. Improvement of the near wall treatment in large eddy simulation for aeroacoustic applications. 2018 AIAA/CEAS Aeroacoustics conference 2018. doi:10.2514/6.2018-3795.
[13] Liefvendahl M, Johansson M. Wall-modeled LES for ship hydrodynamics in model scale. 32nd Symposium on naval hydrodynamics, Hamburg, Germany; 2018.
[14] Lozano-Durán A, Bose ST, Moin P. Prediction of trailing edge separation on the NASA Juncture Flow using wall-modeled LES. AIAA Scitech 2020 Forum. Orlando, FL; 2020. doi:102514/62020-1776.
[15] Mukha T, Johansson M, Liefvendahl M. Effect of wall-stress model and mesh–cell topology on the predictive accuracy of LES of turbulent boundary layer flows. In: 7th European conference on computational fluid dynamics, Glasgow, UK; 2018.
[16] Mukha T, Rezaeiravesh S, Liefvendahl M. A library for wall-modelled large-eddy simulation based on OpenFOAM technology. Comput Phys Commun 2019;239:204–24. doi:10.1016/j.cpc.2019.01.016.
[17] Liefvendahl M, Fureby C. Grid requirements for LES of ship hydrodynamics in model and full scale. Ocean Eng 2017;143:259–68. doi:10.1016/j.oceaneng.2017.07.055.
[18] Remacle J-F, Henrotte F, Carrier-Baudouin T, Béchet E, Marchandise E, Geuzaine C, et al. A frontal Delaunay quad mesh generator using the L^∞ norm. Int J Numer Methods Eng 2013;94(5):494–512. doi:10.1002/nme.4458.
[19] Geuzaine C, Remacle J-F. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. Int J Numer Methods Eng 2009;70(11):1309–31.
[20] Remacle J-F, Lambrechts J, Seny B. Blossom-Quad: a non-uniform quadrilateral mesh generator using a minimum-cost perfect-matching algorithm. Int J Numer Methods Eng 2012;89:1102–19. doi:10.1002/nme.3279.
[21] Frey PJ, George PL. Mesh generation. ISTE; 2008. doi:101002/9780470611166.
[22] Rebay S. Efficient unstructured mesh generation by means of Delaunay triangulation and Bowyer-Watson algorithm. J Comput Phys 1993;106(1):125–38. doi:10.1006/JCPH.1993.1097.
[23] Schöberl J. NETGEN an advancing front 2D/3D-mesh generator based on abstract rules. Comput Vis Sci 1997;1(1):41–52. doi:10.1007/s007910050004.
[24] Nicoud F, Ducros F. Subgrid-scale stress modelling based on the square of the velocity gradient tensor. Flow Turbul. Combust. 1999;62(3):183–200. doi:10.1023/A:1009995426001.
[25] Rezaeiravesh S, Mukha T, Liefvendahl M. Systematic study of accuracy of wall-modeled large eddy simulation using uncertainty quantification techniques. Comput Fluids 2019;185:34–58. doi:10.1016/j.compfluid.2019.03.025.
[26] Temmerman L, Leschziner MA, Mellen CP, Fröhlich J. Investigation of wall-function approximations and subgrid-scale models in large eddy simulation of separated flow in a channel with streamwise periodic constrictions. Int J Heat Fluid Flow 2003;24(2):157–80. doi:10.1016/S0142-727X(02)00222-9.
[27] Spalding DB. A single formula for the 'law of the wall'. J Appl Mech 1961;28(3):455–8.
[28] Kawai S, Larsson J. Wall-modeling in large eddy simulation: length scales, grid resolution, and accuracy. Phys. Fluids 2012;24(1):015105. doi:10.1063/1.3678331.
[29] Jasak H. Error analysis and estimation for the finite volume method with applications to fluid flows. PhD thesis; 1996.

[30] Issa RI. Solution of the implicitly discretised fluid flow equations by operator-splitting. J Comput Phys 1986;62(1):40–65.
[31] Lee M, Moser RD. Direct numerical simulation of turbulent channel flow up to $Re_\tau \approx 5200$. J Fluid Mech 2015;774:395–415.
[32] Pope SB. Turbulent flows. Cambridge University Press; 2000.
[33] Fröhlich J, Mellen CP, Rodi W, Temmerman L, Leschziner MA. Highly resolved large-eddy simulation of separated flow in a channel with streamwise periodic constrictions. J Fluid Mech 2005;526:19–66. doi:10.1017/S0022112004002812.
[34] Rezaeiravesh S, Liefvendahl M, Fureby C. On grid resolution requirements for LES of wall-bounded flows. ECCOMAS Congress 2016. Crete, Greece; 2016. https://www.eccomas2016.org/proceedings/pdf/7105.pdf.
[35] Schlatter P, Örlü R. Assessment of direct numerical simulation data of turbulent boundary layers. J Fluid Mech 2010;659:116–26. doi:10.1017/S0022112010003113.