Conference on Modelling Fluid Flow (CMFF'25)
The 19th International Conference on Fluid Flow Technologies
Budapest, Hungary, August 26-August 29, 2025

ODTLES: Large-Eddy Simulation with Autonomous Stochastic Subgrid-Scale Modeling Applied to Turbulent Duct Flow

Pavle Marinković[1,2,3], Juan A. Medina Méndez[2], Marten Klein[2,3], Heiko Schmidt[2,3]

[1] Corresponding Author. Tel.: +49 (0) 355 69 6034, Fax: +49 (0) 355 69 4891, E-mail: marinkov@b-tu.de
[2] Chair of Numerical Fluid and Gas Dynamics, Faculty of Mechanical Engineering, Electrical and Energy Systems, Brandenburg University of Technology Cottbus-Senftenberg. Siemens-Halske-Ring 15A, 03046 Cottbus, Germany
[3] Scientific Computing Lab, Energy Innovation Center, Brandenburg University of Technology Cottbus-Senftenberg. Universitätsstraße 22, 03046 Cottbus, Germany

ABSTRACT

In this work, we discuss the application of the One-Dimensional Turbulence-based (very) Large-Eddy Simulation model, abbreviated as ODTLES, to turbulent duct flow. ODTLES is a multi-scale flow model in which an autonomous stochastic One-Dimensional Turbulence (ODT) model, capable of simulating the full bandwidth of time and length-scales in a 1-D domain, is supplemented with large-scale 3-D information coming from a very large eddy simulation (VLES) grid. ODTLES is more expensive than any other VLES, but could be cheaper than highly resolved LES or, naturally, than Direct Numerical Simulation (DNS). Unlike Reynolds-Averaged Navier–Stokes (RANS) and VLES, ODTLES does neither need a wall model, nor a damping function. The correct near-wall behavior is naturally obtained from one SGS ODT domain that is locally wall-normal. The proposed hybrid (3-D/1-D) approach allows the resolution of all relevant scales, modeling certain aspects of 3-D turbulence on the SGS scale. Here, turbulent duct flow is considered as an example, which poses a moderate challenge for traditional LES due to emerging secondary flows that manifest themselves by corner vortices that crucially depend on the accurate capturing of small and large scale motions. Preliminary results indicate a reasonable match with DNS for mean velocity profiles, although capturing secondary flow structures remains a challenge at this stage. Further refinements of the solver and modeling approach are ongoing to improve accuracy and predictive capabilities.

Keywords: Duct flow, Large Eddy Simulation (LES), Multiscale modeling, One-Dimensional Turbulence (ODT), ODTLES, Turbulence modeling

NOMENCLATURE

C	[–]	eddy rate parameter
$C_i^{j->k}$	$[\frac{m}{s^2}]$	Coupling term from grid j to grid k for velocity component i
F_i^k	$[\frac{N}{m^3}]$	i-th component of the forcing field residing on an ODT line indirection k
K	[-]	kernel function
P	$[Pa]$	pressure field on the LES level
$P_{a,eddy}$	[–]	eddy acceptance probability
Re_{tau}	[-]	friction Reynolds number
T	$[s]$	simulation time
U_i	$[\frac{m}{s}]$	i-th component of the velocity field on the LES level
Z	[–]	small-scale suppression parameter
c_i	[-]	kernel coefficient
$f(l)$	[-]	assumed exponential distribution of eddy sizes
$g(x_{k,0})$	$[\frac{1}{m}]$	eddy position PDF
$h(l)$	$[\frac{1}{m}]$	eddy size PDF
l	$[m]$	eddy size
l_k^{-1}	[-]	Deconvolution operator
l_k	[-]	Convolution operator
u_i^k	$[\frac{m}{s}]$	i-th component of the velocity field on an ODT line in k direction
x_k	[-]	ODT aligned direction
$x_{k,0}$	[m]	eddy position
$f(x_k)$	[-]	mapping function
$\Delta\tau_\nu$	$[s]$	viscous time-scale
Δt_{VLES}	[s]	time step for VLES
Δt_{samp}	$[s]$	eddy sampling interval
ν	$[\frac{m^2}{s}]$	kinematic viscosity
ρ	$[\frac{kg}{m^3}]$	density
τ	$[s]$	local time scale of a mapping candidate
τ^{-1}	$[\frac{1}{s}]$	eddy frequency
H	[m]	duct height
L	[m]	duct length

Re_τ	[–]	friction Reynolds number
u_τ	[m/s]	friction velocity
$\mathcal{M}\{u_i^k(x_k, t)\}$	[-]	eddy mapping event

Subscripts and Superscripts

e	eddy event
i	component of vector
k	direction of ODT line
$*$	intermediate values (non-divergence-free values)

1. INTRODUCTION

Turbulence, known for its intricate and chaotic flow patterns, is a phenomenon of great importance in various engineering and scientific fields. From optimizing the aerodynamic performance of aircraft to enhancing mixing processes in chemical reactors, a thorough understanding of turbulent flows is crucial for improving design efficiency, predicting system behavior, and ensuring operational safety. Turbulent flows impact the effectiveness of transportation systems, the spread of pollutants in the environment, and the thermal management in industrial applications. Despite its ubiquity, accurately modeling turbulence remains one of the most challenging aspects of fluid dynamics.

Direct Numerical Simulation (DNS) offers full resolution of all turbulent scales and structures. The computational demands of DNS are, therefore, extremely high when flows are highly turbulent (the highest Reynolds number achieved to date with DNS for turbulent channel flow is $Re_\tau \approx 10{,}000$ [1]). In contrast, Reynolds-Averaged Navier–Stokes (RANS) simulations and Large Eddy Simulations (LES), which are more commonly used in practical engineering applications, often struggle to achieve the necessary accuracy, particularly in complex flow scenarios. This is especially true for wall-bounded flows, where traditional turbulence models rely on simplifying assumptions such as small-scale isotropy or local equilibrium of the near-wall flow, which has notable limitations due to which alternative approaches are still being actively researched (e.g. [2]). These assumptions, including low-order statistical closure models and the law-of-the-wall, are inadequate when the details of small-scale turbulent fluctuations are critical. To overcome the limitations of wall models and diffusive or filter-based turbulence models, alternative approaches such as stochastic turbulence models have been developed. One such model is the One-Dimensional Turbulence-based Large Eddy Simulation (ODTLES) model [3, 4] introduced more than a decade ago, and further developed more recently by [5, 6]. ODTLES aims to achieve the fidelity of DNS by resolving all relevant time and length scales within columnar stacks of dimensionally reduced subdomains. In these stacks, the three-dimensional aspects of turbulence are modeled using a stochastic process on a one-dimensional domain, significantly reducing computational costs while maintaining full-scale resolution. Although still in its developmental stages, ODTLES has demonstrated promising results, particularly in simulating turbulent channel flows with friction Reynolds numbers up to $Re_\tau \approx 2040$ on a single Banana Pi M64 computer [6].

2. PRELIMINARY STUDY CASE

As a preliminary test case, we apply the ODTLES model to turbulent flow in a square duct and validate the results against Direct Numerical Simulation (DNS) data provided by Zhang et al. [7]. The geometric dimensions of the duct are chosen to match those used in the DNS study, and are illustrated in Figure 1.

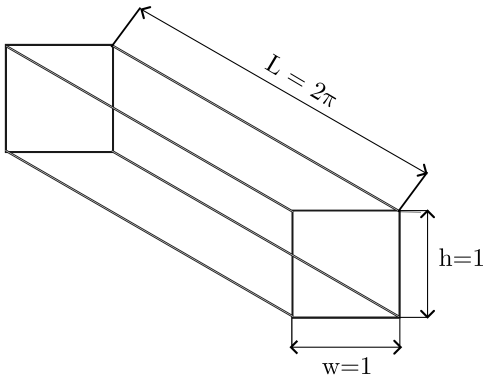

Figure 1. Computational domain of the square duct configuration.

No-slip boundary conditions are applied at the duct walls, while periodic boundary conditions are imposed at the inlet and outlet in the streamwise direction. To drive the flow, a constant streamwise pressure gradient is applied. The magnitude of the pressure gradient is prescribed as $\frac{u_\tau^2}{H}$, where u_τ is the friction velocity and H is the duct width. The value of the kinematic viscosity is then calculated as $\nu = \frac{u_\tau \frac{H}{2}}{Re_\tau}$.

Simulations were performed for friction Reynolds numbers $Re_\tau = 300$ and 600. In both cases, the base LES grid has a resolution of $10 \times 10 \times 10$, while the auxiliary ODTLES grids (see figure 3) are refined to ensure that the first cell sizes in the y and z directions correspond to wall units y^+ and z^+ less than one.

This very coarse base LES resolution has been utilized because we are still working on the algorithm, although in previously it has been shown that using a base resolution that is slightly higher (16x16 resolution in the cross section) to be sufficient for moderately turbulent duct flows [5].

3. ONE-DIMENSIONAL TURBULENCE

Kerstein's One-Dimensional Turbulence model (ODT) [8] is a stochastic modeling approach that utilizes dimensionally reduced map-based advection modeling. It is conceptualized as a standalone turbulence model that operates on a one-dimensional domain, akin to the visualization of a turbulent flow field from within a turbulence line-of-sight, see also [8, 9]. The model simulates the effects of three-dimensional turbulence through a discrete sequence of stochastic events, which are commonly known as eddy events. These eddy events are represented by a model representation of the effect that vortical motions cause in 1-D scalar profiles within turbulent flows. Assuming a 1-D domain aligned in direction x_k, where $k \in \{1, 2, 3\}$, eddy events are implemented as a triplet map of one-dimensional scalar profiles, $\psi(x_k) \rightarrow \psi[f(x_k)]$, where $f(x_k)$ is the mapping function. The eddy events are sampled in time with an interval Δt_{samp}, where $\Delta t_{samp} \ll \tau_v$, with τ_v being the viscous time-scale. This guarantees full time-scale resolution of the turbulent flow. Operationally, the sampling follows a Poisson stochastic process, achieving a pre-specified mean acceptance probability and a corresponding mean eddy rate. Details of the sampling process can be found in [10]. The mean acceptance probability is calculated as the online average of the different acceptance probabilities for each eddy event candidate, $P_{a,eddy}$, which is estimated as

$$P_{a,eddy} = \frac{\Delta t_{samp}}{l^2 \tau(y_0, l, t) h(l) g(y_0)}. \tag{1}$$

In this equation, $h(l)$ and $g(x_{k,0})$ are presumed probability density functions (PDFs) for eddy sizes l and eddy positions $x_{k,0}$, respectively. Indeed, $g(x_{k,0})$ is a uniform PDF, given that it is assumed that turbulent eddies can occur with equal probability everywhere along the 1-D domain. Conversely, $f(l)$ is an assumed exponential distribution of eddy sizes, see details in [9]. The local time scale of an individual mapping candidate is denoted by τ. The term "candidate" is used because not all mappings yield a physical time-scale or eddy turnover time τ. This is best explained when defining the eddy frequency τ^{-1} based on dimensional arguments for extractable kinetic energy. Operationally, τ^{-1} considers the following integral kinetic energy balance for the eddy candidate with size l, where a proportionality coefficient C has been inserted in order to indicate the direct proportionality, see details in [8, 9]

$$\tau^{-1} = C \frac{\nu}{l^2} \sqrt{\left[\frac{l}{\nu} \sum_{i}^{3} \frac{1}{l^2} \int_l u_i^k(t, f(x_k)) K(x_k) \mathrm{d}x_k \right]^2 - Z} \tag{2}$$

In order for τ^{-1} to be a real number, the quantity within the square root must be positive. To that extent, the parameter Z acts as a dimensionless coefficient for eddy suppression. Specifically, it restricts the implementation of small eddies, and can take the form of either a Reynolds number or a viscous coordinate following algebraic manipulation of Equation 2. Both C and Z are then turbulence model coefficients indicating the empiricism associated with the ODT model. Note that $u_i^k(t, f(x_k))$ represents the mapped profile for the velocity component u_i (which is represented in the domain aligned in x_k) at time t, while $K = x_k - f(x_k)$ is a kernel function (see [8]). Note that we also use $i \in \{1, 2, 3\}$, similar to k. Reasons for the apparent notation redundancy regarding index k will be clarified later. Acceptance of an eddy event candidate, i.e., the mapping $u_i^k(x_k) \rightarrow u_i^k(f(x_k))$, also implies a linear transformation of the post-mapped profile $u_i^k(f(x_k)) + c_i K$, where c_i is a (uniform) kernel coefficient which depends on the form of the scalar velocity profile. Details of the calculation of c_i can be found in [11, 9].

For an ODT line oriented in the x_k-direction, there is also a 1-D governing partial differential equation, which in the case of low Mach constant property flow can be written as

$$\left.\frac{\partial u_i^k}{\partial t}\right|_{\text{ODT}} = \int_0^T \sum_e \mathcal{M}\{u_i^k(x_k, t)\} \delta(t - t_e) \mathrm{d}t + \nu \frac{\partial^2 u_i^k}{\partial x_k^2} + F_i^k \tag{3}$$

where $\mathcal{M}\{u_i^k(x_k, t)\}$ is one transformation of the form $u_i^k(x_k, t) \rightarrow u_i^k(f(x_k), t) + c_i K(x_k)$. This transformation can be implemented several times in an intermittent fashion during the numerical time integration of the 1-D PDE, which is the reason why a summation over eddy events e appears in Equation 3. The delta function in Equation 3 is only written for the purpose of its integral identity returning a discrete value (discrete eddy events in time), whereas said integral implies the entire simulation time T. As commented before, eddy events are evaluated using a predictive approach based on the current flow state and a sampled eddy size l and position $x_{k,0}$. When an eddy event is accepted, the diffusive (and forcing) terms of Equation 3 are advanced in time up to the moment in which the eddy event trial is set to occur. This process is commonly referenced as a catch-up diffusion event in ODT. Additionally, in Equation 3, F_i^k is an acceleration term due to a corresponding body force.

To illustrate the impact of eddy events and the subsequent diffusion process on the velocity profile, we present a plot showing an initial velocity profile, the profile after the mapping of an eddy event, and the profile after the catch-up diffusion in Figure 2.

4. MODEL EXTENSION: 3-D ODTLES

In order to supplement VLES information to ODT, and extend the model to its 3-D counterpart, ODTLES, it is necessary to consider modifications to the standalone ODT implementation. Instead of representing a 3-D velocity field with three velocity components, as it is usual in the standalone ODT vector formulation, see [11], the ODT module in ODTLES represents only two velocity components within the 1-D domain. To that extent, we redefine the ODT velocity field u_i^k by making i and k permuta-

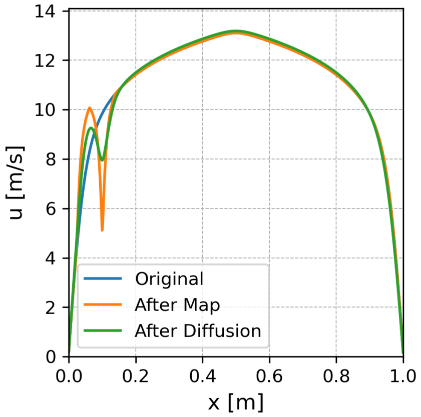

Figure 2. Visualization of map-based advection and diffusion operators for a smooth initial profile, the profile after eddy mapping, and the profile after catch-up diffusion.

tions of $\{1, 2, 3\}$, with $i \neq k$. Thus, the velocity component aligned in the direction of the ODT domain is not governed by Equation 3. Having said that, in order to develop a coherent interpretation of ODTLES, it is necessary to define operators relating the different scales and numerical grids (e.g., the VLES and the ODT grid). The one-dimensional filtering operator $[l_k]$ generates a coarsely resolved velocity variable from the finely resolved velocity in the x_k direction:

$$U_i(X_k, t) = [l_k]u_i^k(x_k, t) = \frac{1}{\Delta X_k} \int_{X_k - \frac{\Delta_{X_k}}{2}}^{X_k + \frac{\Delta_{X_k}}{2}} u_i^k \, dx_k \quad (4)$$

Here, X_k is the resulting coarsened grid (previously x_k) with grid cell sizing Δ_{X_k}, while U_i is the corresponding large-scale velocity field.

4.1. Spatial discretization strategy

ODTLES is a multi-scale model that operates on meshes with different levels of refinement. The coarsest resolution or large-scale information level is represented in a (V)LES grid, whose resolution is set according to the standard practices used in VLES or coarse RANS simulations. Additional to the VLES grid, we introduce three supplementary grids, which we reference as ODTLES grids. Each of these three grids are equivalent to the VLES grid, but feature refinements in each of the three principal Cartesian directions, respectively. The level of said refinements is set according to the Direct Numerical Simulation (DNS) resolution requirements for the smallest turbulent scales in each respective direction. To that extent, the memory requirements for ODTLES can be estimated as approximately $\sim 3N_{\text{VLES}}^2 N_{\text{DNS}}$, where N_{VLES} and N_{DNS} are the VLES and ODT grid resolutions, respectively. This is significantly lower than the memory requirements for DNS, which can be roughly estimated as $\sim N_{\text{DNS}}^3$. Consequently, for a fixed LES resolution, the cost of ODTLES for increasing Reynolds numbers varies linearly, in contrast to the cubic variation expected for DNS [12]. Overall, each ODTLES grids is the result of an embedding of a set of parallel 1-D ODT domains, as seen in Figure 3. This geometric configuration restricts the application of the current ODTLES model to structured meshes. Although ODTLES is not fundamentally restricted to equidistant meshes, the current implementation prioritizes equidistant meshes for simplicity (grid discretization in every direction is equidistant, although it may be different across directions).

In ODTLES, the velocity components u_i^k correspond to velocity fields represented in an ODTLES grid refined in direction x_k. Conversely, the coarsely resolved pressure and velocity variables in the VLES grid are denoted as P and U_i, respectively.

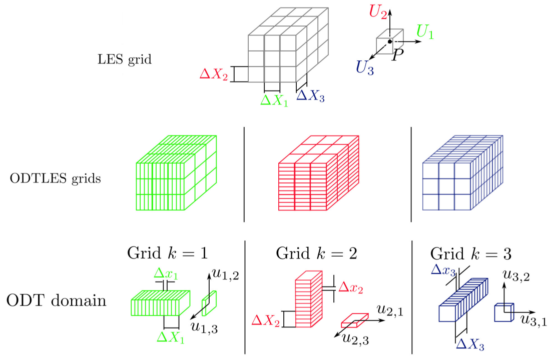

Figure 3. ODTLES model structure, showing the primary LES grid and the three supplementary ODTLES grids with refinements in the principal Cartesian directions.

A consistency condition needs to be fulfilled between the velocity fields, such that

$$U_i = [l_k]u_i^k, \quad \text{for every } k \quad (5)$$

The latter consistency condition implies that the filtered velocity field representation of all ODTLES grid represented velocity fields should be equivalent, see also [5].

In contrast with the precepts of the filtering operation defined in Equations 4 and 5, it is in general not possible to perfectly reconstruct a field u_i^k from its large-scale counterpart. That is, the filtering operator has no unique inverse, such that, in general, $[l_k^{-1}][l_k] \neq 1$. Nevertheless, Schmidt et al. [3] proposed a numerical approximation for the refinement of coarsely resolved information, i.e., an algorithm for the deconvolution operator of coarse fields $[l_k^{-1}]$. Said algorithm was later improved by Glawe [5] to address the issue of artificial extrema generation near no-slip boundary conditions. The algorithm in [3, 5] is high-order accurate and avoids the introduction of noticeable discontinuities. It also fulfills an important integral constraint, namely,

$$U_i = [l_k][l_k^{-1}]U_i \tag{6}$$

An example of the application of $[l_k^{-1}]$ with k being the wall normal direction in our duct case is visualized in Figure 4.

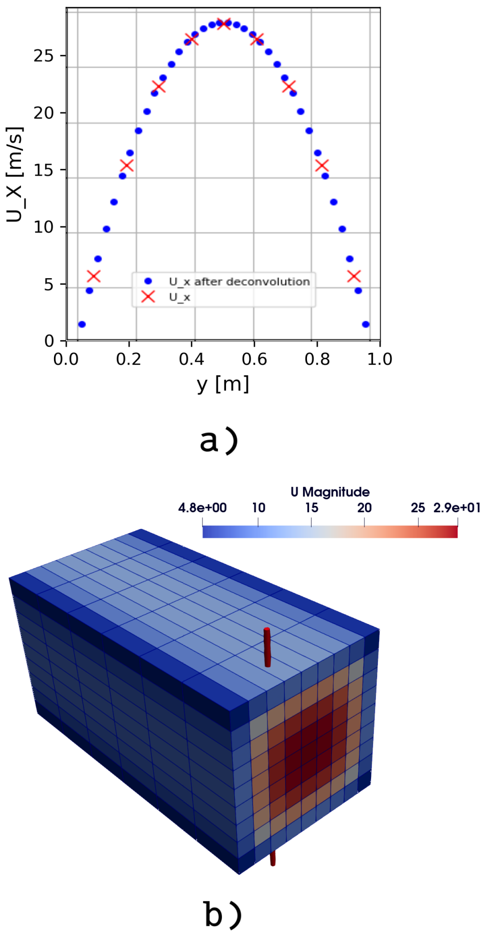

Figure 4. Example of the deconvolution algorithm at work. a) The deconvoluted fine-scale-resolved velocity together with the coarse-scale velocity. b) Visual representation of the VLES grid and the selected ODT domain along which the deconvolution was done (indicated by the extended local y axis shown in red).

4.2. ODTLES governing equations

The 3-D extension of the standalone ODT model with LES support, ODTLES, can be written as follows for the velocity component u_i, $i \in \{1, 2, 3\}$ in a grid refined in direction x_k, $k \neq i$, $k \in \{1, 2, 3\}$

$$\frac{\partial u_i^k}{\partial t} = -[l_k^{-1}]\left\{\frac{\partial(U_iU_i)}{\partial X_i} + \frac{\partial\left(U_jU_i\right)}{\partial X_j} + \frac{\partial(U_kU_i)}{\partial X_k}\right\} + \nu\frac{\partial^2 u_i^k}{\partial X_i^2} + \left.\frac{\partial u_i^k}{\partial t}\right|_{\mathrm{ODT}} + C_i^{j\to k} - \frac{1}{\rho}\frac{\partial P}{\partial X_i} \tag{7}$$

Note that the Einstein summation convention is omitted in equation 7, implying that there are actually six equations similar to 7 which must be solved simultaneously (2 velocity components in 3 ODTLES grids). The corresponding ODTLES continuity equation, or zero divergence condition, is only satisfied at the VLES level, see [5], and it is written as

$$\frac{\partial U_i}{\partial X_i} + \frac{\partial U_j}{\partial X_j} + \frac{\partial U_k}{\partial X_k} = 0 \tag{8}$$

In equations 7 and 8, ρ denotes the density. Also, the term $\partial u_i^k/\partial t|_{\mathrm{ODT}}$ is the entire ODT acceleration (or time-advancement) given by Equation 3. Such term considers the standalone ODT time-advancement of one ODT *stack* within the ODTLES grid, during $T = \Delta t_{\mathrm{VLES}}$, the latter being a time-step required for stable numerical integration on the coarse VLES grid level. This implies that the numerical time-advancement of Equation 3 utilizes an eddy sampling procedure $\Delta t_{\mathrm{samp}} \ll \tau_v$, and catch-up diffusion events with time-stepping $\Delta t_{\mathrm{ODT}} \ll \Delta t_{\mathrm{VLES}}$, where again, Δt_{ODT} is a numerical time-step required for stable time-integration in the ODT domain (hence, compliant with the corresponding CFL condition). We note that the standalone ODT time-advancement imposes a limit on the maximum eddy size to be sampled from the presumed eddy size PDF $h(l)$, see [10, 13]. This limit is maintained in the standalone ODT advancement which is part of ODTLES.

It is also noted that we indicated lowercase and uppercase variables distinctively in Equations 7 and 8. On one hand, lowercase variables indicate the finely resolved fields or directions, while, on the other hand, uppercase variables denote the VLES fields and directions. As a notation example, we note $\partial^2 u_i^k/\partial X_i^2$ as being a coarse diffusion operator on u_i^k, the finely resolved velocity field defined on grid k (finely resolved in x_k direction); said coarse diffusion operator considers the VLES grid cell size ΔX_i for the diffusion estimation (recall that $i \neq j \neq k$). Also, note that all advection terms are defined at VLES level, with the VLES velocity components U_i, U_j, U_k. All advection effects are simply deconvoluted from the VLES grid to each ODTLES grid.

The term $C_i^{j\to k}$ in Equation 7 is a coupling term that facilitates the communication between ODTLES grids. This allows the coupling of the ODT-modeled turbulent advection from the array of 1-D domains which conform the ODTLES grid refined in x_j direction to the ODTLES grid refined in x_k direction. In order to avoid double-counting of the forcing term from the standalone ODT advancement in ODTLES grid j in the ODTLES grid k, the forcing term is subtracted from the time-advancement. In fact, the acceleration term $\partial u_i^k/\partial t|_{\mathrm{ODT}}$ is better understood as a velocity change in a time interval $T = \Delta t_{\mathrm{VLES}}$, for the purpose of the coupling represented by $C_i^{j\to k}$. These considerations lead to the definition of $C_i^{j\to k}$ as

$$C_i^{j\to k} = [l_k^{-1}][l_j]\left(\left.\frac{\partial u_i^j}{\partial t}\right|_{\mathrm{ODT}} - F_i^j\right) \tag{9}$$

Figure 5 shows a visualization of the instantaneous vortical structures, represented by Q-criterion iso-surfaces, alongside the instantaneous velocity magnitude.

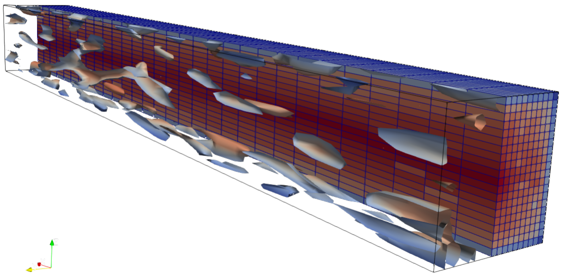

Figure 5. Instantaneous flow structures in the square duct. The left half of the domain shows iso-surfaces of the Q-criterion highlighting vortical structures, while the right half displays the instantaneous velocity magnitude. Visualization performed on a mesh of a higher resolution for visualization quality.

Equation 7 is solved using a finite-volume method (FVM) discretization with the standard midpoint rule for locating all flow variables at cell centers. Standard upwind interpolation schemes are used for the advective fluxes, and linear interpolation schemes are used for the diffusion fluxes. It is noted that this discretization method is different from previous ODTLES implementations relying on a staggered grid logic, see [6, 5]. That is, we have transitioned to a collocated approach. The numerical time integration method is defaulted to an explicit Euler scheme. However, we note that an implicit-explicit scheme for better stability was proposed in the staggered grid logic in [6].

Finally, we comment on the pressure gradient term $\partial P/\partial X_i$ in Equation 7. It is important to note that the pressure variable P is only defined on the VLES grid level. Applying a projection method, we split the time-advancement of Equation 7 into a predictor and a corrector step. The predictor step considers all momentum contributions other than the pressure gradient. This causes a change in the ODTLES velocity fields, e.g., $u_i^{k,n}$ to $u_i^{k,*}$ where we have used the superindex n to indicate a time discretization, such that time-advancement takes place from t^n to t^*, with $\Delta t_{\mathrm{VLES}} = t^* - t^n$. Note that all advection effects are calculated on the VLES grid at time t^n, e.g., using the velocity field U_i^n. Following the predictor step, we obtain the intermediate non-divergence-free velocity fields u_i^{k*}. Due to $C_i^{j\to k}$, these velocity fields are consistent across ODTLES grids, such that we can obtain the VLES intermediate velocity field using the filtering operator,

$$U_i^* = [l_k]u_i^{k*} \tag{10}$$

The intermediate predicted VLES velocity fields are then used for the corrector step in order to enforce zero-divergence at the VLES level by solving the following pressure Poisson equation,

$$\begin{aligned}\frac{\partial^2 P}{\partial X_i^2}+\frac{\partial^2 P}{\partial X_j^2}+\frac{\partial^2 P}{\partial X_k^2} \\ = \frac{\rho}{\Delta t_{\mathrm{VLES}}}\left(\frac{\partial U_i^*}{\partial X_i}+\frac{\partial U_j^*}{\partial X_j}+\frac{\partial U_k^*}{\partial X_k}\right)\end{aligned} \tag{11}$$

After P is found from equation 11, the VLES velocity field U_i is updated to the next time-step for advancement t^{n+1} as

$$U_i^{n+1} = U_i^* - \frac{\Delta t_{\mathrm{VLES}}}{\rho}\frac{\partial P}{\partial X_i} \tag{12}$$

The change $U_i^{n+1} - U_i^*$ over Δt_{VLES} is considered a source term (after deconvolution) to update $u_i^{k,*}$ to $u_i^{k,n+1}$. This updates the information in the ODTLES grids. Note that we obtain the mass-fluxes required to evaluate the divergence of U_i^* in equation (11) by linear interpolation. Said face-centered mass-fluxes are also updated with equation 12 using a face-centered pressure gradient (calculated using the collocated pressure values). They are stored and used later for the calculation of the upwinded VLES advection effects in equation 7.

5. SOLVER IMPLEMENTATION

The current ODTLES codebase represents the second C++ rewrite of the original Fortran code developed by Christoph Glawe in [6]. The initial C++ codebase, a porting used in a previous publication in [6], was developed in order to modernize the original code and achieve high modularity and expandability. The current version is a highly refactored iteration of that codebase. The strategy and experiences of this process have been documented in [14], highlighting the importance and benefits of clean code principles. By adhering to these practices, we have completed a comprehensive refactor of the remaining parts of the algorithm, which we anticipate will facilitate quick and relatively easy improvements and expansions of the current ODTLES algorithm.

As discussed in [12], ODTLES greatly benefits from parallelization. In its current version, we utilize OpenMP to parallelize major parts of the algorithm. Most notably, we advance the ODT part of the momentum equations in parallel for each ODT line. Running the ODTs in parallel is observed to have excellent parallel efficiency due to the numerous individual tasks, each taking roughly the same amount of time, with tasks that are substantial enough such that the overhead of thread creation is not significant.

Similarly, all deconvolution operations, being one-dimensional, are parallelized using the same logic. These operations are also observed to exhibit high parallel efficiency, although we expect slightly worse performance as in the ODT part, since the tasks are not as computationally expensive.

For the advancement of the remaining parts of the momentum equations, we have divided the available threads into three groups—one for each ODTLES grid—and perform the explicit advancement in parallel within each group. This approach is also observed to yield a good parallel efficiency.

In the current code version, the pressure Poisson problem of Equation 11 is solved using the Eigen library's SparseLU solver [15]. While a strict performance measurement has not been carried out for this part of the algorithm, we anticipate that it will greatly benefit from further parallelization efforts. In the future, we plan to utilize Hypre's multigrid solvers [16] as in the older ODTLES versions in [5, 6], which are designed to run in parallel and are well-suited for large-scale linear systems. This transition is expected to further enhance the efficiency and scalability of our solver, although we stress that the bottleneck of the algorithm should not be in the Poisson problem anyway, since this is only solved at the large-scale VLES grid level.

6. PRELIMINARY RESULTS

We have conducted simulations of turbulent flow in a square duct at friction Reynolds numbers Re_τ = 300 and 600.

In the current setup, we are actively fine-tuning the value of the model's C parameter, while keeping the Z parameter fixed at the standard value commonly used in standalone ODT studies. These adjustments aim to optimize the balance between model fidelity and computational efficiency.

So far, preliminary results demonstrate that the refactored ODTLES implementation is capable of producing mean streamwise velocity profiles along the wall bisector that agree reasonably well with reference DNS data by Zhang et al. [7], as shown in Figure 6.

While previous work has shown that ODTLES can capture the emergence of secondary flows in duct geometries [5]), the present solver—undergoing substantial algorithmic restructuring—has not yet successfully reproduced these secondary motions. This limitation is currently under investigation.

Furthermore, we are exploring additional algorithmic improvements, particularly concerning the treatment of advection terms within the ODTLES framework, with the aim of enhancing accuracy while not significantly decreasing performance.

7. SUMMARY AND OUTLOOK ON ONGOING AND FUTURE WORK

In this study, we presented the implementation of a recent version of the One-Dimensional Turbulence-based Large-Eddy Simulation (ODTLES) model within a newly refactored C++ codebase. This framework provides increased flexibility, maintainability, and testing capabilities, allowing for the use of arbitrary boundary conditions on a structured finite-volume collocated grid.

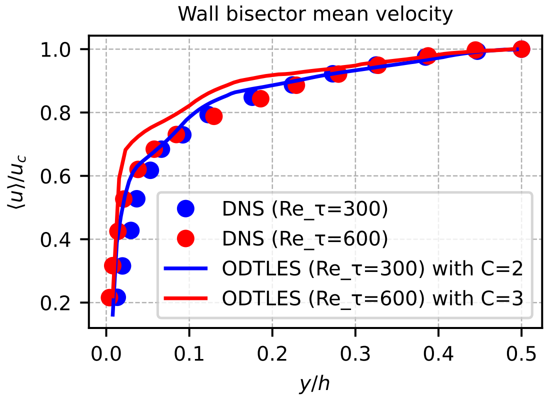

Figure 6. Comparison of mean streamwise velocity profiles from ODTLES and DNS [7] at Re_τ = 300 and 600.

We applied the model to turbulent flow in a square duct, a challenging test case due to the presence of secondary flows and corner vortices. While preliminary results show reasonable agreement with DNS data in terms of mean velocity profiles along the wall bisector, the current solver has not yet been able to fully capture the complex secondary flow structures observed in experiments and previous ODTLES studies [5].

This limitation is attributed to ongoing algorithmic restructuring, especially concerning the treatment of advection terms, and highlights the need for further refinement. Future work will focus on addressing these challenges and performing additional tests and comparisons, with the goal of achieving robust and accurate predictions of three-dimensional flow features in duct geometries.

ACKNOWLEDGEMENTS

This research is supported by the German Federal Government, the Federal Ministry of Education and Research and the State of Brandenburg within the framework of the joint project EIZ: Energy Innovation Center (project numbers 85056897 and 03SF0693A) with funds from the Structural Development Act for coal-mining regions.

REFERENCES

[1] Hoyas, S., Oberlack, M., Alcántara-Ávila, F., Kraheberger, S. V., and Laux, J., 2022, "Wall turbulence at high friction Reynolds numbers", *Phys Rev Fluids*, Vol. 7, p. 014602, URL `https://link.aps.org/doi/10.1103/PhysRevFluids.7.014602`.

[2] Heinz, S., and Fagbade, A., 2025, *A critical review of hybrid RANS-LES concepts: Continuous eddy simulation versus classical methods*, American Institute of Aeronautics and Astronautics, Inc.

[3] Schmidt, R. C., Kerstein, A. R., and McDermott, R., 2010, "ODTLES: A multi-scale model for 3D turbulent flow based on one-dimensional turbulence modeling", *Comput Methods Appl Mech Eng*, Vol. 199 (13), pp. 865–880.

[4] Gonzalez-Juez, E. D., Schmidt, R. C., and Kerstein, A. R., 2011, "ODTLES simulations of wall-bounded flows", *Phys Fluids*, Vol. 23, p. 125102.

[5] Glawe, C., 2015, "ODTLES: Turbulence Modeling Using a One-Dimensional Turbulence Closed Extended Large Eddy Simulation Approach", Ph.D. thesis, Freie Universität Berlin.

[6] Glawe, C., Méndez, J. A. M., and Schmidt, H., 2018, "IMEX based multi-scale time advancement in ODTLES", *Z Angew Math Mech*, Vol. 98, pp. 1907–1923.

[7] Zhang, H., Trias, F. X., Gorobets, A., Tan, Y., and Oliva, A., 2015, "Direct numerical simulation of a fully developed turbulent square duct flow up to Re=1200", *International Journal of Heat and Fluid Flow*, Vol. 54, pp. 258–267, URL `https://www.sciencedirect.com/science/article/pii/S0142727X15000685`.

[8] Kerstein, A., 1999, "One-Dimensional Turbulence: Model formulation and application to homogeneous turbulence, shear flows, and buoyant stratified flows", *J Fluid Mech*, Vol. 392, pp. 277–334.

[9] McDermott, R. J., 2005, "Toward one-dimensional turbulence subgrid closure for large-eddy simulation", Ph.D. thesis, University of Utah.

[10] Lignell, D. O., and Rappleye, D. S., 2012, "One-dimensional-turbulence simulation of flame extinction and reignition in planar ethylene jet flames", *Combust Flame*, Vol. 159 (9), pp. 2930–2943.

[11] Kerstein, A. R., Ashurst, W. T., Wunsch, S., and Nilsen, V., 2001, "One-Dimensional Turbulence: Vector formulation and application to free shear flows", *J Fluid Mech*, Vol. 447, p. 85–109.

[12] Méndez, J. A. M., Glawe, C., Starick, T., Schöps, M. S., and Schmidt, H., 2019, "IMEX-ODTLES: A multi-scale and stochastic approach for highly turbulent flows", *Proc Appl Math Mech*, Vol. 19, p. e201900433.

[13] Klein, M., Schmidt, H., and Lignell, D. O., 2022, "Stochastic modeling of surface scalar-flux fluctuations in turbulent channel flow using one-dimensional turbulence", *Int J Heat Fluid Flow*, Vol. 93, p. 108889.

[14] Marinković, P., Medina, J. A., Schöps, M. S., Klein, M., and Schmidt, H., 2025, "Experiences From the Bottom-Up Development of an Object-Oriented CFD Solver with Prospective Hybrid Turbulence Model Applications", *Proc Appl Math Mech*, Vol. 25 (1), p. e202400190, `https://onlinelibrary.wiley.com/doi/pdf/10.1002/pamm.202400190`.

[15] Guennebaud, G., Jacob, B., et al., 2010, "Eigen v3", http://eigen.tuxfamily.org.

[16] "*hypre*: High Performance Preconditioners", `https://llnl.gov/casc/hypre`, `https://github.com/hypre-space/hypre`.