Energie-Innovationszentrum Cottbus

Brandenburgische Technische Universität Cottbus - Senftenberg

Ongoing development of a hybrid reduced order stochastic/LES solver for turbulent flows

Pavle Marinković [1,2], Juan A. Medina Méndez [1], Heiko Schmidt [1,2], Marten Klein [1,2]
[1]Scientific Computing Lab of the EIZ, [2]Chair of Numerical fluid and gas dynamics, BTU Cottbus-Senftenberg

Introduction

Scaling energy systems poses computational challenges, especially in turbulent flow simulations like the atmospheric boundary layer. While Direct Numerical Simulations (DNS) offer accuracy, they are computationally prohibitive, and Large Eddy Simulations (LES) remain costly. This drives the need for efficient algorithms like the One-Dimensional Turbulence-based Large Eddy Simulation (ODTLES), balancing accuracy and efficiency.
Reliable, maintainable, and extensible code is crucial for effective computational tools in solving complex problems. Rigorous code refactoring enhances turbulent flow modeling capabilities and overall computational simulation efficiency in energy systems and beyond.

Code Refactoring

Software development in computational fields is iterative and prone to change. Our refinement of the ODTLES solver involves extensive code refactoring, focusing on reliability, maintainability, and extensibility. Originally in Fortran, the codebase was rewritten in C++, with ongoing significant changes and Python supplementation for flexibility and future expansions. This meticulous approach ensures adaptability, enabling rapid modifications. Guided by core principles—extraction, abstraction, and unit testing—we systematically extract reusable components, clarify complex functionalities, and rigorously test our codebase for robustness and scalability.

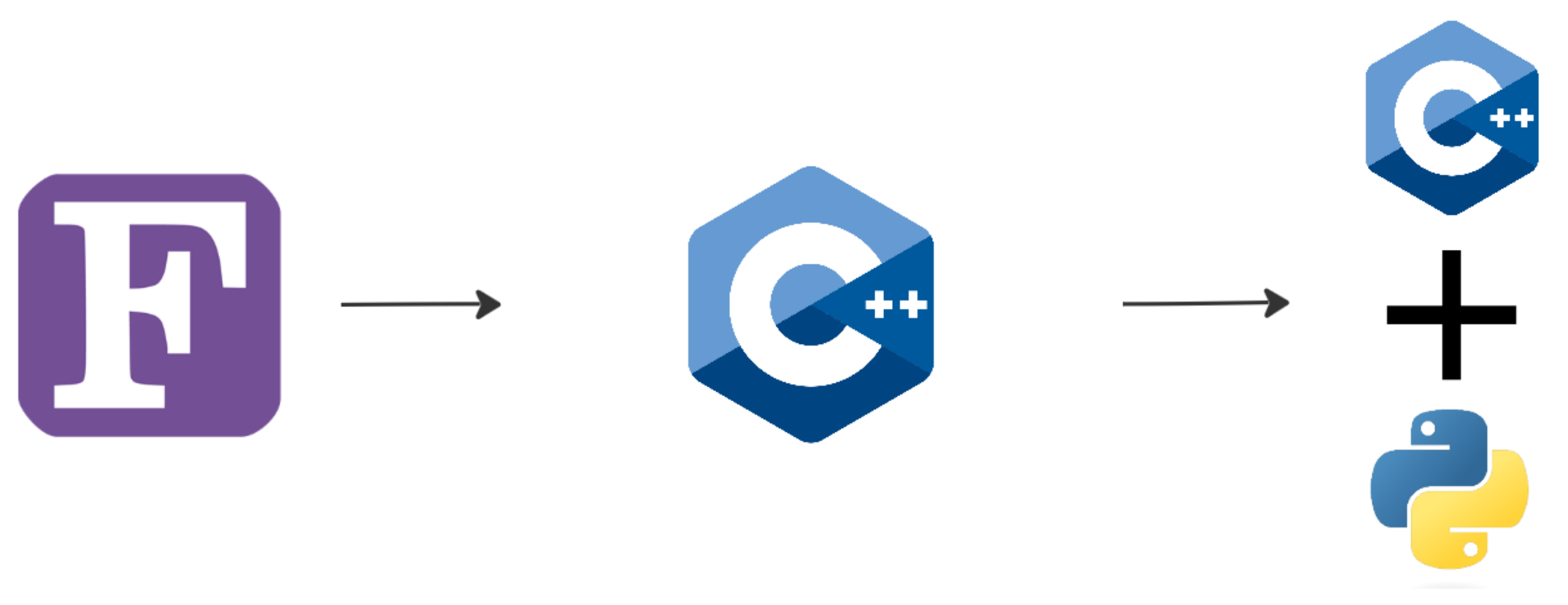

Figure 1: Code timeline

Extraction

In software development, extraction refers to the process of isolating and extracting specific functionalities, components, or patterns from existing code, such as classes or methods. This is done to create more modular, reusable, or maintainable software structures by breaking down larger entities into smaller, more manageable units. Below you can see an example from our current work.

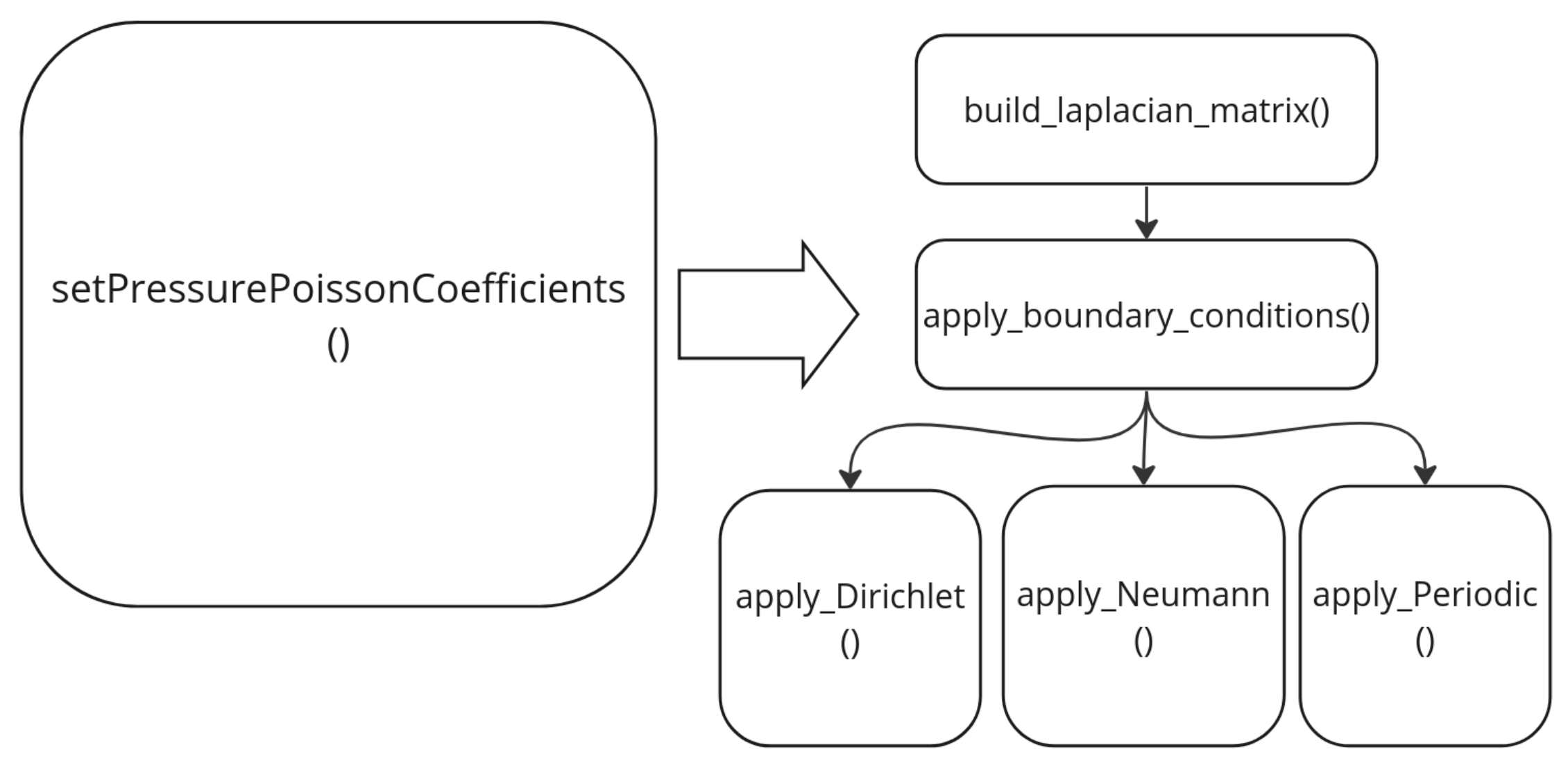

Figure 2: Extraction from the setPressurePoissonCoefficients() method

Abstraction

Abstraction in software development reduces code complexity by increasing conceptual complexity. In this refactor, we heightened the conceptual complexity of our momentum equation representation and boundary condition handling. Initially, we directly manipulated indices (1) (only diffusion shown for brevity), necessitating numerous conditional statements to identify boundary indices and execute specific boundary condition operations.

$$\rho\frac{\partial U_{pq}}{\partial t}\Delta x\Delta y = \left(\nu\frac{\partial U}{\partial y}\bigg|_{p+\frac{1}{2},q} - \nu\frac{\partial U}{\partial y}\bigg|_{p-\frac{1}{2},q}\right)\Delta x + \left(\nu\frac{\partial U}{\partial y}\bigg|_{p,q+\frac{1}{2}} - \nu\frac{\partial U}{\partial y}\bigg|_{p,nq-\frac{1}{2}}\right)\Delta x \qquad (1)$$

We have transitioned to a more abstract for of the momentum equation, that is the so called compact form (2).

$$a_p^U U_p + \sum_{\text{faces}} a_n^U U_n = b + S^U \qquad (2)$$

With this formulation, we now loop over all faces of a cell implicitly and boundary conditions are handled by the neighbour coefficients a_n^U, owner cell coefficient a_p^U and the source term S^U at lower levels. This was possible to achieve by also increasing the conceptual complexity of our Mesh class that now contains a method for if a face belongs to a boundary.

Unit testing

Unit tests are code snippets designed to validate the behavior of individual components or units of software, ensuring they meet specified requirements and perform as expected. In our current work we have put emphasis on writing and maintaining extensible unit tests. Below you can see a visual representation of a test case for the aforementioned method of the Mesh class.

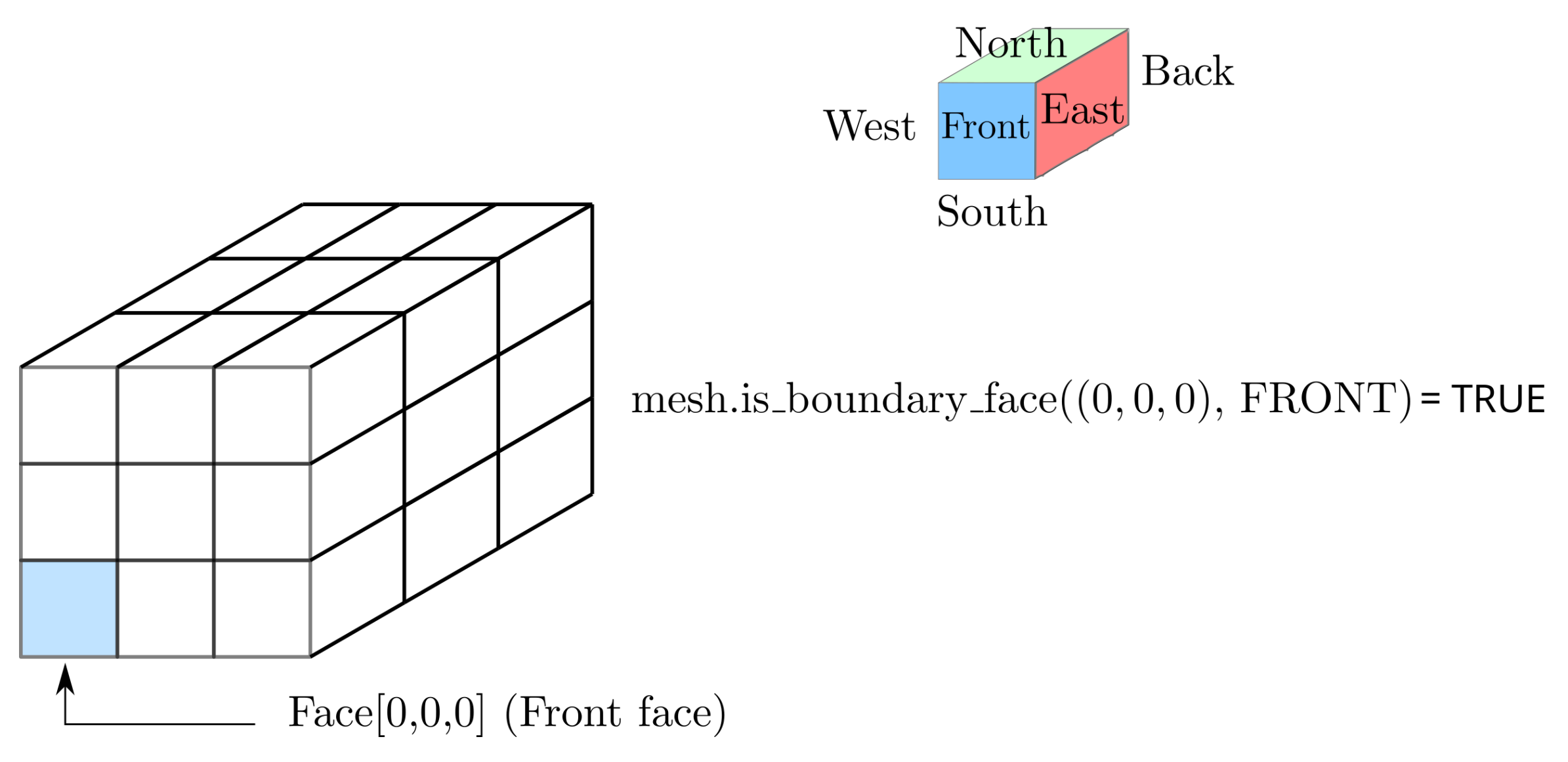

Figure 3: Illustration of a test case for the `is_boundary_face()` method

ODTLES/XLES

ODTLES is a model with high potential for highly turbulent flows, and from a methodological point of view is in between DNS and LES. In ODTLES a set of 1D ODT models us embedded in a coarse grained 3D LES, where on the ODT scale, the turbulent advection is modeled as a sequence of stochastic eddy events, while the other terms are fully resolved in space and time [1].

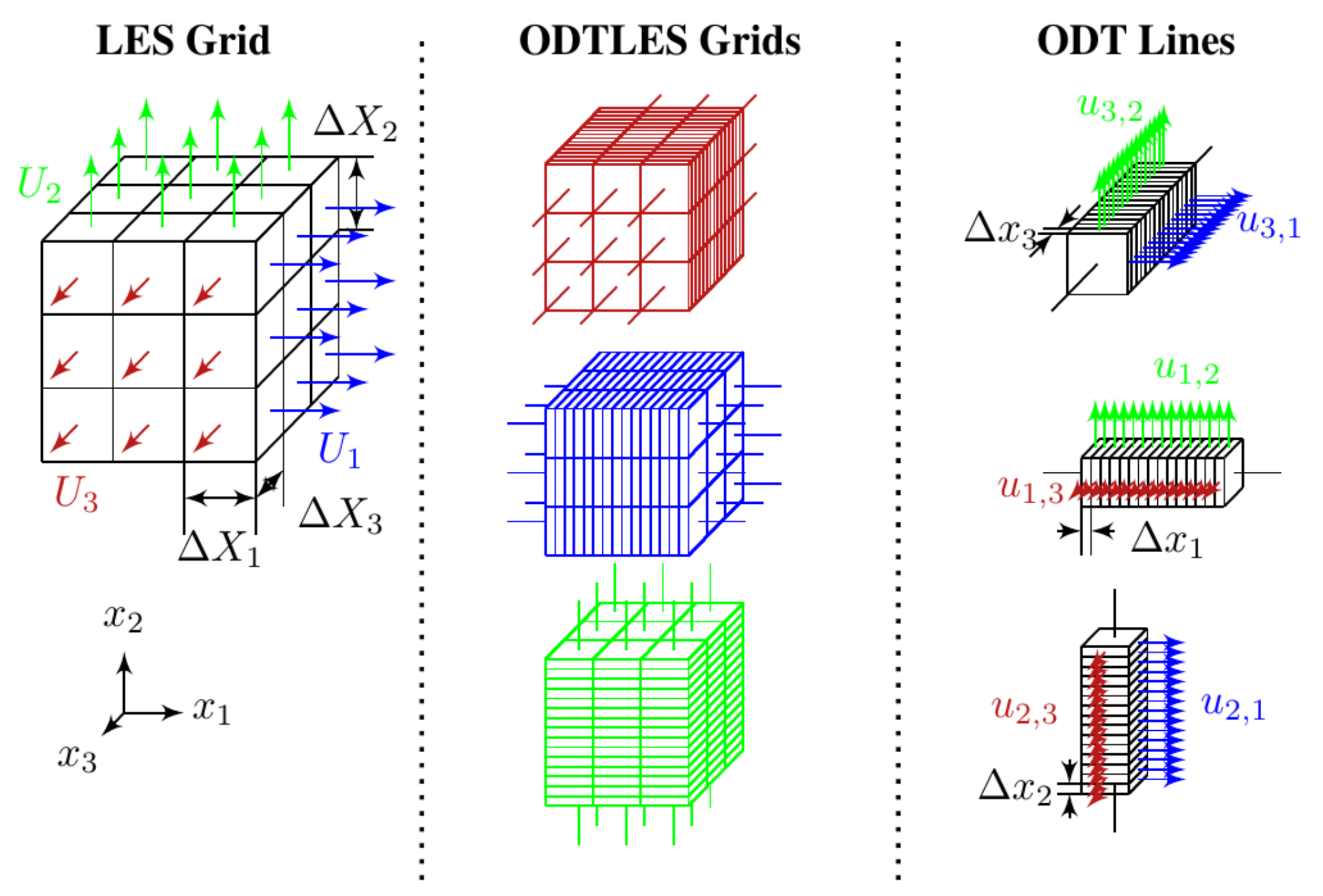

Figure 4: ODTLES model structure [2]

The ODTLES momentum equation can be written as:

$$\begin{aligned} 0 =& \frac{\partial u_{k,i}}{\partial t} + \frac{1}{\rho}\frac{\partial P}{\partial X_i} \\ &+ \frac{\partial}{\partial x_k}u_{k,k}\cdot u_{k,i} + \frac{\partial}{\partial X_i}u_{k,i}\cdot u_{k,i} + \frac{\partial}{\partial X_j}u_{k,j}\cdot u_{k,i} + C^{LES}_{j\to k,i} \\ &+ (eddy_{k,i} - \nu\frac{\partial^2}{\partial x_k^2}u_{k,i} - F_i) - \nu\frac{\partial^2}{\partial X^2}u_{k,i} + C^{ODT}_{j\to k,i} \end{aligned} \qquad (3)$$

where $\{i,j,k\}$ are positive permutations of $\{1,2,3\}$ with $i \neq k$. The turbulent ODT advancement $eddy_{k,i}$ represents an ODT eddy and $C^{LES}_{j\to k,i}$ and $C^{ODT}_{j\to k,i}$ are coupling terms used for communication between grids [1].

Results and outlook

The current state of the code has successfully produced results that closely align with equivalent OpenFOAM simulations for laminar 3D channel and cavity flows. The ongoing process of integrating ODT with the newly refactored LES base holds promise for further advancements. Upon completion of this coupling, it will pave the way for the exploration of new ideas and the continued development of the model.

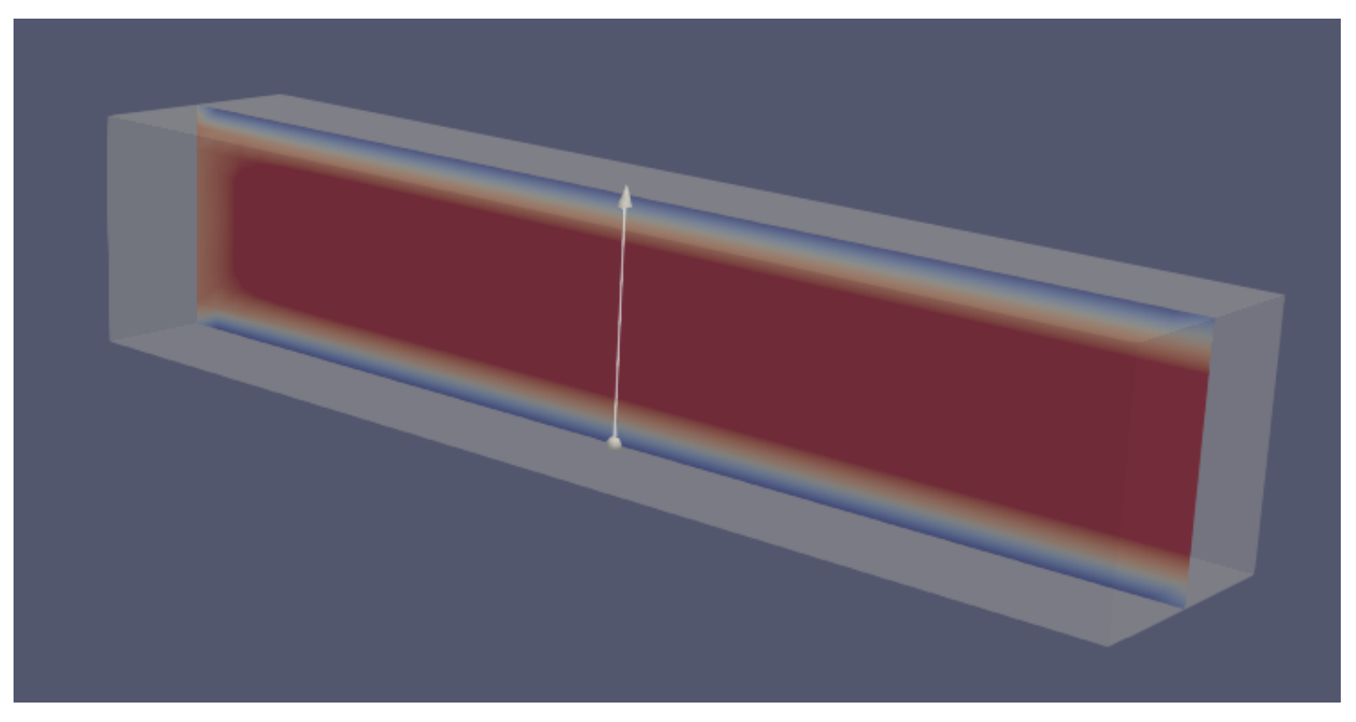

Figure 5: Laminar Channel flow results

References

1. C. Glawe, J.A. Medina Méndez, and H. Schmidt. IMEX based multi-scale time advancement in ODTLES. *Zeitschrift für Angewandte Mathematik und Mechanik*, 98:1907-1923, 2018.
2. Juan A. Medina Méndez, Christoph Glawe, Tommy Starick, Mark S. Schöps, and Heiko Schmidt. IMEX-ODTLES: A multi-scale and stochastic approach for highly turbulent flows. *Proc. Appl. Math. Mech.*, 19:e201900433, 2019.

This project is funded by the German government with funds from the Structural Development Act (Strukturstärkungsgesetz) for coal-mining regions and co-financed with funds from the state of Brandenburg.

Supported by: The Federal Government on the basis of a decision by the German Bundestag

LAND BRANDENBURG

SPONSORED BY THE Federal Ministry of Education and Research