GPU-accelerated simulations for eVTOL aerodynamic analysis

Pasquariello, Vito; Bunk, Y.L.; Eberhardt, S.; Huang, Pei-Hsuan ; Matheis, Jan; Ugolotti , Matteo ; Hickel, S.

DOI
10.2514/6.2023-2107

Publication date
2023

Document Version
Final published version

Published in
AIAA SciTech Forum 2023

AIAA SciTech Forum
23-27 January 2023, National Harbor, MD & Online
AIAA SCITECH 2023 Forum

10.2514/6.2023-2107

GPU-accelerated simulations for eVTOL aerodynamic analysis

Vito Pasquariello * †, Yannick Bunk ‡, Sebastian Eberhardt †, Pei-Hsuan Huang †, Jan Matheis † and Matteo Ugolotti†
Lilium GmbH, Claude-Dornier-Str. 1, 82234 Weßling, Germany

Stefan Hickel§
Delft University of Technology, 2629HS Delft, The Netherlands

The demand for fast, high-fidelity, scale-resolving computational fluid dynamics (CFD) simulations is continuously growing. Especially new emerging aviation technologies, such as electrical vertical take-off and landing aircraft (eVTOL), strongly rely on advanced numerical methods to retain development life-cycle costs and achieving design targets more quickly. This paper presents a cutting-edge large-eddy simulations (LES) solver developed to enable over-night turnaround times for full aircraft simulations on advanced graphics processing unit (GPU) architectures. The solver models weakly compressible fluid flows over complex three-dimensional bodies based on an immersed boundary method with geometry-based and flow-based automatic mesh adaption. Its high accuracy and unprecedented performance is demonstrated for high Reynolds number aerodynamic benchmark cases and compared to recent results from literature. In addition, the successful validation against experimental data for the Lilium Jet canard is discussed.

I. Introduction

Computational fluid dynamics (CFD) has a rich history of successful applications for aircraft analysis and design, and has radically changed the way aerospace systems are designed during the last decades. CFD enabled the aerospace community to go beyond incremental system performance improvement and allowed an in-depth understanding of the underlying flow-physics phenomena. Reynolds-averaged Navier-Stokes (RANS) models have been considered the standard for solving complex flows of industrial interest in the past years. However, the well-known inability of RANS turbulence models to properly capture highly separated regions has always been problematic [1, 2] This sets serious limitations in, e.g., further improving the aircraft drag-to-lift ratio at off-design conditions, increasing aero-engines propulsion efficiency, or reducing the aerodynamic losses. Required advances in fuel burn, noise emission, and new sustainable means of transportation [3] opened new challenges that can only be addressed with high-fidelity numerical technologies. In view of the rapid evolution of computer platforms with graphics processing units, direct numerical simulations (DNS) and large-eddy simulations (LES) are two possible high-fidelity methods that can accurately predict unsteady flows characterized by laminar-turbulent transition and boundary-layer separation. These techniques will also help to address multi-physics problems, such as acoustic analysis, aerodynamically induced low/high-frequency fluid-structure interaction (e.g. flutter), and aero-thermal interactions.

DNS resolving all temporal and spatial scales of turbulence are still intractable for most industrial applications; however, LES accurately resolves the dynamically most relevant flow structures in space and time with similar accuracy as DNS, thus drastically reducing the resolution requirements and the overall simulation cost. However, fully wall-resolved LES of high-Reynolds number flows is still not within the reach of modern High Performance Computing (HPC) systems. In wall-modeled LES (WMLES) the energetic turbulent scales within the near-wall boundary layer are not fully resolved but modeled, allowing for coarser meshes to be used [4–6]. The WMLES approach is chosen in this work for the analysis of electric vertical take-off and landing applications.

The accuracy of the physics-based predictive fluid dynamics models is only one of the findings listed in the NASA 2030 Grand Challenge [1, 2]; also the mesh generation process and mesh adaptation are considered a "significant bottleneck in the CFD workflow". The industry standard is the use of body-fitted meshes that require a considerable amount of time and expertise to be properly built and expensive re-meshing procedures to dynamically adapt to moving boundaries. The latter is an especially challenging bottleneck in aircraft flight-dynamics simulations where large surface

*Project Lead, Corresponding Author (vito.pasquariello@lilium.com)
†Aerodynamics Software Developer, Flight and Applied Sciences Department, Lilium GmbH, 82234, Weßling, Germany.
‡Aerodynamics Engineer, Flight and Applied Sciences Department, Lilium GmbH, 82234, Weßling, Germany.
§Professor of Aerodynamics, Faculty of Aerospace Engineering, Delft University of Technology, 2629HS Delft, The Netherlands.

deformations and displacements are common. The numerical framework presented in this work was thus designed to require close to zero user input for mesh generation, and the mesh adaptation is carried out fully automatically. In the framework of an immersed boundary method (IBM) [7, 8], the octree-based adaptive mesh refinement (AMR) capabilities of the p4est library [9] enable fast mesh adaption in our current framework.

The CFD 2030 Grand Challenge report emphasizes the need for the CFD algorithms to cope and adapt to new emerging HPC hardware capabilities. In this regard, several research groups developed or enabled the use of graphics processing units (GPUs) in their in-house CFD solvers [10–13]. This was always done by refactoring standard CFD methods and ad-hoc performance optimization based on the encountered algorithmic bottlenecks. The presented LES code aims to achieve RANS-like turnaround timing and, therefore, is built from scratch specifically for modern NVIDIA GPUs like the Ampere architecture. The purpose of this paper is to demonstrate the capability and performance of this Lilium in-house GPU-based LES code, and to present results for innovative eVTOL configurations featuring complex flow phenomena.

This paper is organized as follows: Section II introduces the baseline numerics and the immersed boundary and wall-modeling framework. Section III presents results for three validation cases: a turbulent channel flow, an inclined turbulent boundary layer flow and a realistic aircraft model in landing configuration, the so-called Japanese Aerospace Exploration Agency Standard Model (JSM). Section IV summarizes important optimization aspects of the present solver and compares computational costs of the JSM simulations to recently published results for another GPU-accelerated flow solver. Finally, in Section V, we present LES results of our Lilium 6-engine canard configuration, for which an experimental test campaign has been performed in the past.

II. Numerical Framework

The numerics used for the present solver are specifically chosen and tailored for high-performance scale-resolving simulations on GPUs. In the following, the baseline governing equations, the finite-volume discretization scheme, the immersed boundary method, and the meshing strategy are described on a high level.

A. Navier-Stokes equations and barotropic fluid model

We perform LES of the three-dimensional compressible Navier-Stokes equations on adaptive block-Cartesian grids in a finite volume framework for subsonic flows. The Navier-Stokes equations written in conservative form are

$$\partial_t \mathbf{U} + \boldsymbol{\nabla} \cdot \mathbf{F}(\mathbf{U}) - \boldsymbol{\nabla} \cdot \mathbf{D}(\mathbf{U}) = 0\,, \tag{1}$$

with the state vector $\mathbf{U} = [\rho, \rho u_1, \rho u_2, \rho u_3]$ consisting of density ρ and linear momentum ρu_i for a barotropic fluid. In the above equation, the total flux is split into an inviscid part $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3]^T$ following

$$\mathbf{f}_i(\mathbf{U}) = [u_i\rho, u_i\rho u_1 + \delta_{i1}p, u_i\rho u_2 + \delta_{i2}p, u_i\rho u_3 + \delta_{i3}p]^T\,, \tag{2}$$

and a viscous contribution $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3]^T$ following

$$\mathbf{d}_i(\mathbf{U}) = [0, \tau_{i1}, \tau_{i2}, \tau_{i3}]^T\,, \tag{3}$$

where u_i are the components of the velocity vector and τ_{ij} is the viscous stress tensor, which according to the Stokes hypothesis for a Newtonian fluid is

$$\tau_{ij} = \mu\left(\partial_j u_i + \partial_i u_j - 2/3\,\delta_{ij}\partial_k u_k\right)\,. \tag{4}$$

For LES with an eddy viscosity subgrid-scale (SGS) turbulence model, the dynamic viscosity coefficient $\mu = \mu_{air} + \mu_{SGS}$ is the sum of the molecular viscosity and the eddy viscosity. The molecular viscosity is modeled with Sutherland's law

$$\begin{aligned} \mu_{air}(T_w) &= \mu_{cal}\left(\frac{T_w}{T_{cal}}\right)^{3/2}\frac{T_{cal}+S_{cal}}{T_w+S_{cal}}\,, \\ T_{cal} &= 273.15\text{K}\,, \\ \mu_{cal} &= 17.16\times10^{-6}\text{Pa s}\,, \\ S_{cal} &= 110.4\text{K}\,, \end{aligned} \tag{5}$$

based on the expected adiabatic wall temperature

$$T_w = T_{ref} + \frac{u_{ref}^2}{2c_p(T_{ref})}, \tag{6}$$

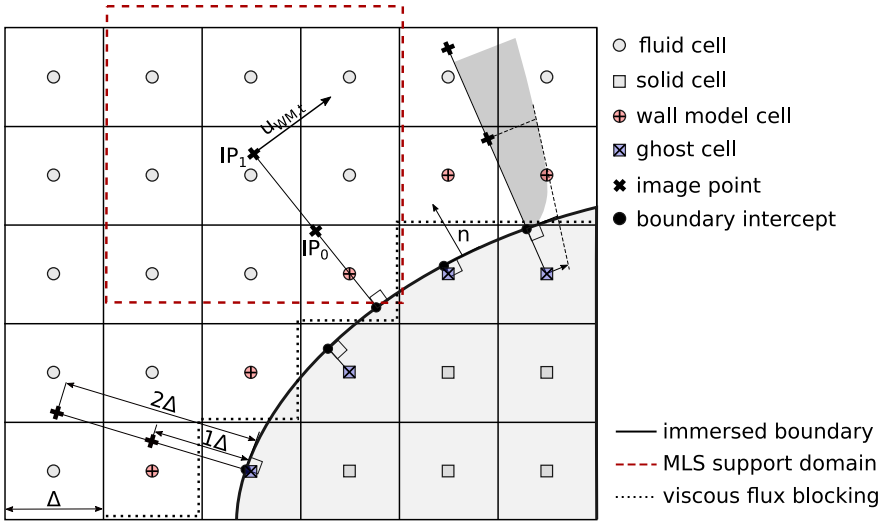


Fig. 1 Schematic of the ghost-cell immersed boundary method and the wall-model framework.

for a given reference velocity u_{ref} and reference temperature T_{ref}. Assuming subsonic flow, the system of equations in Eq. 1 is closed with the following linear barotropic equation of state (EOS)

$$p(\rho) = A\rho + B \,, \tag{7}$$

with

$$A = \gamma_{ref} R\, T_{ref} \,, \tag{8a}$$

$$B = \left(1 - \gamma_{ref}\right) p_{ref} \,. \tag{8b}$$

Equations 7-8 result from linearizing the ideal gas model around a reference state. Slope A and integration constant B are obtained from the specific gas constant of air $R = 287.05\,\mathrm{J/(kg\,K)}$ and by evaluating the specific heat capacity c_p with the NASA Glenn polynomials [14] at the reference state, which yields $\gamma_{ref} = c_p(T_{ref})/(c_p(T_{ref}) - R)$.

B. Finite Volume Discretization for LES

The compressible Navier-Stokes equations are discretized with a conservative finite-volume method on a block-Cartesian mesh with block-based adaptive local mesh refinement. The convective terms are discretized with a second-order skew-symmetric central scheme, while the viscous flux is discretized using a second-order central difference scheme. The discretization method has essentially second-order convergence rate in space and time. For improved robustness on very coarse meshes, the central scheme is blended with a small amount of local upwinding at block interfaces with hanging nodes as well as close to the immersed boundary. A second-order explicit time-integration scheme is used. Several SGS models for turbulence closure are implemented in the present solver. Vreman's model [15] is used for all simulations presented in this paper.

C. Immersed Boundary and Wall Modeling Framework

We follow the basic ghost-cell immersed boundary methodology introduced by Mittal et al. [16], where boundary conditions on the surface are implicitly imposed by specifying fluid variables in so-called ghost cells. This enables the simulation of flows around arbitrary moving geometries on a Cartesian grid. The cells of the computational mesh are tagged as either fluid, solid, ghost or wall model cell based on their signed distance to the specified geometry, see also Fig. 1. Solid and ghost cells (negative signed distance) are both inside an object and excluded from the time integration, where the latter type of cells is used by the spatial discretization scheme. For the second-order scheme used in this work only one layer of ghost cells is needed. We identify ghost cells as cells with a negative signed distance and at least one fluid-cell neighbor. Similarly, wall model cells are those with a positive signed distance and at least one ghost cell neighbor.

While it is well-known that ghost-cell IBM are suited to sharply represent the immersed boundary [17], a special wall-treatment is needed for high Reynolds number flows in order to overcome the severe mesh-resolution requirements of wall-resolving simulations. For the LES presented in this paper, a simple equilibrium wall-stress model is used to estimate the wall-shear stress τ_w based on the instantaneous wall-parallel velocity magnitude $u_{WM,t}$ at a constant off-wall distance. Image points IP_1 are used as the wall-model exchange location and placed at 2Δ normal to the immersed boundary, in accordance with the suggestions mentioned in [5] to avoid the log-layer mismatch. A moving least squares (MLS) method [18] with linear basis functions is used for the interpolation of fluid variables at this location. The MLS support domain is highlighted in Fig. 1, resulting in possibly 19 fluid cells (9 in 2D) used for the regression. We have opted for this interpolation method because it offers flexibility to exclude solid cells from the interpolation, which is important at corners and in narrow gaps. The shear stress itself is imposed through a source term in all wall model cells, modifying the wall-parallel momentum according to the computed wall-shear stress τ_w. As the wall function replaces the (under)resolved viscous stresses, the resolved viscous flux is blocked at the cell faces between wall-model cells and ghost cells, see also the dotted line in Fig. 1.

Following suggestions of Tamaki and Kawai [19], we apply a partial-slip wall-boundary condition to calculate the state of the ghost cells. The wall-tangential velocity is obtained by extrapolation using the analytical derivative of the wall function in the wall-normal direction at the image point IP_0, which is placed at a constant off-wall distance of 1Δ, see also Fig. 1. In wall-normal direction, a zero-penetration boundary condition is imposed based on the wall normal velocity at IP_0, which is reconstructed with the MLS method. This partial-slip boundary condition reduces numerical errors close to the immersed boundary for under-resolved grids by removing the non-linearity of the velocity profile close to the wall, which is key for maintaining the shear-stress balance in WMLES on non-body-conforming grids [19].

D. Mesh Refinement

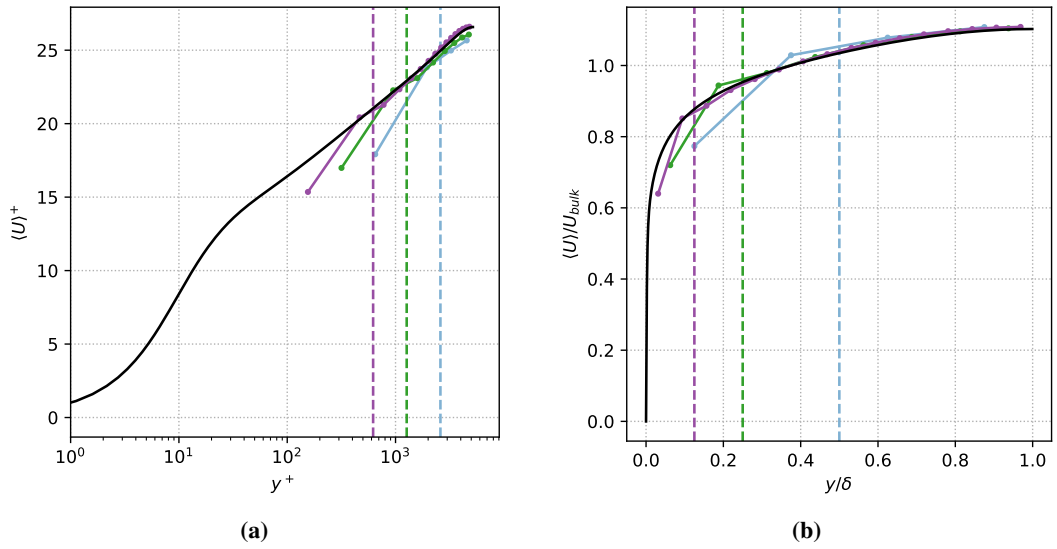
The present solver is based on a structured multi-block grid approach and relies on the p4est library [9] to generate the grid blocks. p4est is a software library for parallel adaptive mesh refinement that enables the dynamic management of a forest of adaptive octrees. Local mesh refinement is employed close to immersed boundaries based on user-defined wall-cell sizes for individual geometry parts. Furthermore, a solution-adaptive mesh refinement strategy based on different refinement criteria is also available. We discretize all blocks in the grid with the same number of cells in each direction, which allows for optimized algorithms with improved GPU occupancy. To optimally balance the blocks and parts of the immersed boundary we use the METIS library [20] with a multi-objective optimization target, see also Section IV for more details. This achieves an even distribution of regular computational domain cells and at the same time an even distribution of cells which are cut by the geometry across all processes.

III. Validation

A. Turbulent Channel Flow

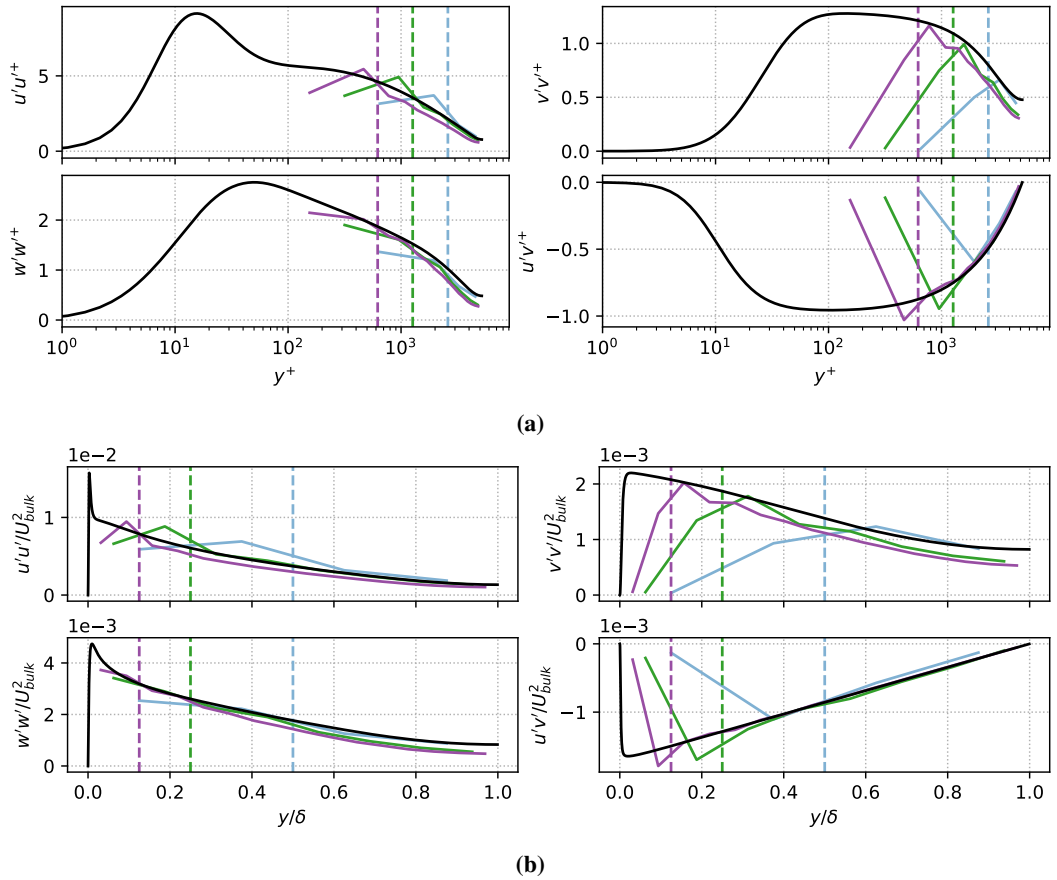
The new flow solver was rigorously verified and validated. In this section, we show selected results for a periodic turbulent channel flow (TCF) at $\mathrm{Re}_\tau = 5200$. Results are compared against DNS data of Lee and Moser [21]. The Mach number was set to 0.2 such that the present simulation can be considered as incompressible. A volume source term together with a PI controller is used to maintain a constant bulk Reynolds number $\mathrm{Re}_b = U_b\delta/\nu$ of 125000, where U_b is the area-averaged mean streamwise velocity, δ is the channel half height, and ν the kinematic viscosity. The computational domain size is $2\pi\delta \times 2\delta \times \pi\delta$ and discretized with a uniform Cartesian grid. Periodic boundary conditions are applied in the streamwise and spanwise directions of the computational domain. We show results for three different grid resolutions with 8 (coarse), 16 (medium) and 32 (fine) uniformly distributed grid points across the channel height 2δ.

Figure 2 shows the mean velocity profiles in inner (left) and outer scaling (right) for the three meshes. The vertical dashed line represents the matching location between wall function and LES solution and may be interpreted as effective y^+ value below which a match between DNS data and LES solution cannot be expected. The obtained y^+ values are 621, 1267 and 2597 for the fine, medium and coarse mesh, respectively. A very good agreement is observed for the fine mesh with DNS data, while we see some deviations in the outer layer of the profile on the coarse and medium mesh. Figure 3 shows the Reynolds stress tensor in inner and outer scaling. Up to the matching height a good agreement is observed for all components of the Reynolds stress tensor. It is remarkable that the coarse and medium mesh with only 8 and 16 cells across the channel height yield good results for both first and second order moments. However, the TCF is grid aligned



(a) (b)

Fig. 2 (a) averaged streamwise velocity in inner scaling, and (b) averaged streamwise velocity in outer scaling at $Re_\tau \approx 5200$: (——) DNS of Lee and Moser [21], (——) present fine mesh, (——) present medium mesh, (——) present coarse mesh.



(a)

(b)

Fig. 3 Reynolds stress components (a) in inner scaling and (b) in outer scaling at $Re_\tau \approx 5200$: (——) DNS of Lee and Moser [21], (——) present fine mesh, (——) present medium mesh, (——) present coarse mesh.

and therefore not representative for a realistic geometry immersed in a Cartesian grid. The validation case discussed in the next paragraph aims to demonstrate that the presented framework yields also satisfactory statistics for a turbulent flow with arbitrary cut cells.

B. Inclined Turbulent Boundary Layer

Simulations of nominally zero-pressure-gradient fully developed flat-plate turbulent boundary layers (TBL) are conducted to validate the present wall-model framework for non-body-conforming Cartesian grids. This study is motivated by a validation case shown by Tamaki and Kawai [19]. Figure 4 shows an overview of the computational domain, where α is the angle of the plate relative to the x-axis. The thin black lines represent block edges. Every block contains 16^3 cells. The size of the computational domain is $L_x \times L_y \times L_z = 90\delta \times 61\delta \times 1.2\delta$, where δ refers to the approximate target boundary layer thickness at which profiles are extracted. The Mach number is set to 0.2. Local mesh refinement is applied in the vicinity of the flat plate such that the boundary layer edge does not extend

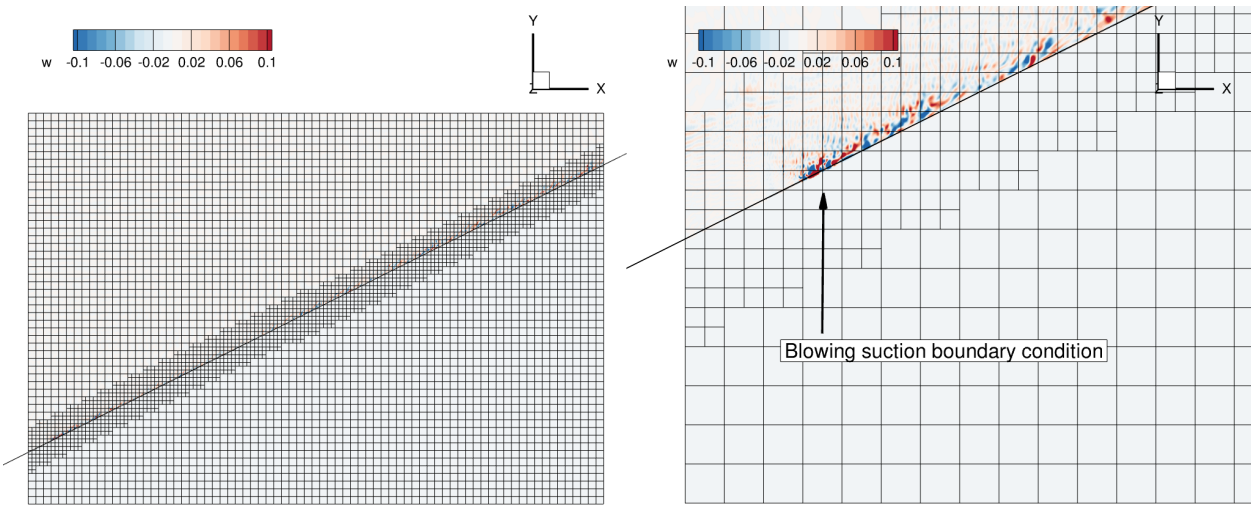


Fig. 4 Overview of the computational setup with $\alpha = 26.56°$. The medium resolution mesh is shown. Thin black lines represent block edges. Every block contains 16^3 cells. Contours of the spanwise velocity component w are depicted to visualize the TBL.

into coarser regions of the mesh. At the inflow and upper boundary patch a constant velocity is prescribed using a non-reflective boundary condition based on Riemann invariants. The static pressure is prescribed at the outflow boundary. A blowing-suction strategy at the immersed boundary is employed close to the inflow to force rapid transition of the flow to a turbulent state, see Fig. 4. Results for three series of numerical experiments are presented:

1) A mesh convergence study at fixed plate angle α and fixed target Reynolds number.
2) A plate angle study at fixed target Reynolds number on a given mesh.
3) A Reynolds number study at fixed plate angle α on a given mesh.

An overview of the simulated cases is given in Tab. 1.

1. Mesh study at $Re_\theta = 4981$ (cases 1, 2, and 3)

Figure 5 shows results of a mesh study at fixed plate angle $\alpha = 26.56°$ and fixed target Reynolds number based on the momentum thickness $\mathrm{Re}_\theta = 4981$. LES predictions are compared to reference DNS data from Schlatter and Örlü [22]. The mesh resolutions for the coarse, medium and fine mesh are given in Table 2. Figure 5a shows the velocity profiles in inner scaling for the three meshes. The vertical dashed lines indicate the coupling location of the wall-stress model with the LES solution and may therefore be interpreted as effective y^+ value at the wall. Above the matching height, a very good agreement is observed with both experimental and numerical reference data.

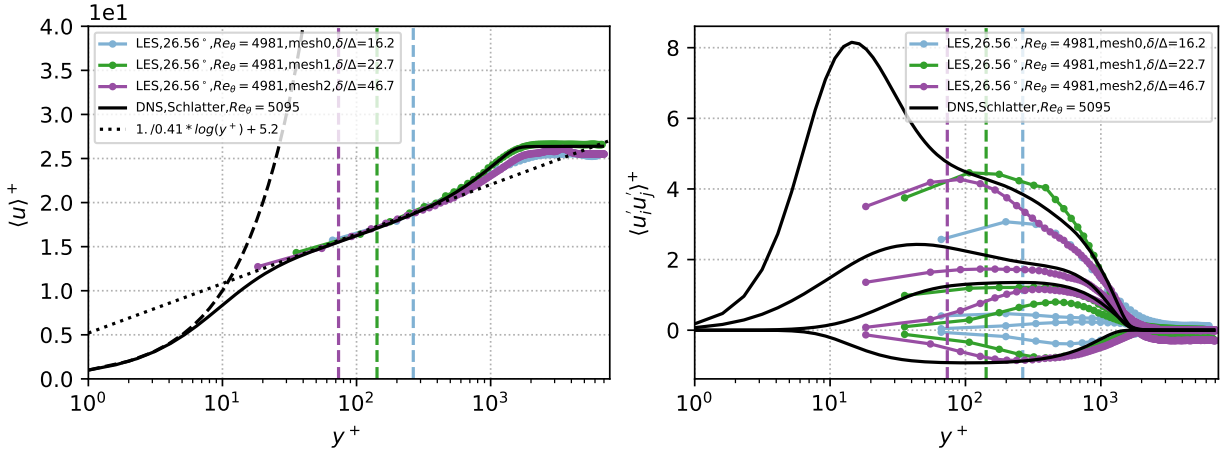
Figure 5b shows the Reynolds stress profiles in inner scaling. Good agreement with the reference data is observed for both the medium and fine mesh. The coarse mesh shows larger discrepancies for $\langle u'v' \rangle$, $\langle v'v' \rangle$ and $\langle w'w' \rangle$. It

Case	α	Mesh	Unit Reynolds number	Re_θ
1	26.56°	coarse	10 000	$\approx$ 4981
2	26.56°	medium	10 000	$\approx$ 4981
3	26.56°	fine	10 000	$\approx$ 4981
4	7.50°	medium	10 000	$\approx$ 4981
5	15.00°	medium	10 000	$\approx$ 4981
6	26.56°	medium	50 000	$\approx$ 28 390
7	26.56°	medium	250 000	$\approx$ 123 268
8	26.56°	medium	500 000	$\approx$ 235 785

Table 1 Overview of simulated cases.

	Δ	δ/Δ
mesh0 (coarse)	0.375	$\approx$ 15
mesh1 (medium)	0.1875	$\approx$ 25
mesh2 (fine)	0.093 75	$\approx$ 50

Table 2 Mesh settings for inclined flat plate TBL.



(a) **(b)**

Fig. 5 Velocity profiles (a) and Reynolds stresses (b) in inner scaling at target Reynolds number $\mathrm{Re}_\theta = 4981$ using a coarse, medium and fine mesh.

appears that $\approx$ 16 points per boundary layer thickness are not sufficient to accurately predict the Reynolds stresses on non-body-conforming meshes, probably because the location of the matching height on the coarse mesh is $> 0.1\delta$, i.e., outside the inner layer of the boundary layer.

2. Angle study at $Re_\theta = 4981$ using the medium resolution mesh (cases 2, 4, and 5)

Next the robustness of the proposed method is investigated for different plate angles $\alpha = \{7.50°, 15°, 26.56°\}$. The angles have been selected such that the flat plate does not cut the upper boundary patch. Figure 6 shows the mean streamwise velocity and Reynolds stress profiles in inner scaling for the cases 2, 4, and 5. The profiles collapse reasonably well for these cases and show good agreement with the experimental and numerical reference data. From these results we conclude that the predictions of the WMLES approach are robust for different cell-cut conditions.

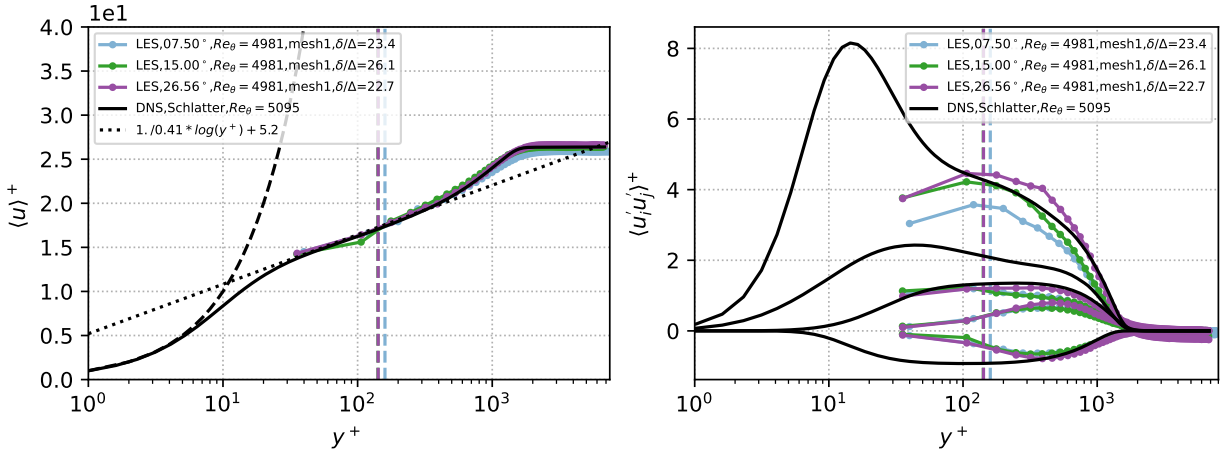


Fig. 6 Velocity profiles (a) and Reynolds stresses (b) in inner scaling at target Reynolds number $Re_\theta = 4981$ using the medium resolution mesh. The flat plate angle α is varied from $7.5°$ to $26.56°$.

3. Unit Reynolds number study (cases 2, 6, 7, and 8)

In the following paragraph, the unit Reynolds number is increased from 10000 up to 500000 without changing the mesh resolution. All simulations use the medium resolution *mesh1*. This leads to effective y^+ values of > 5000 while fixing the matching height at $< 0.1\delta$. Boundary layer profiles are extracted at $\approx 66\delta$ downstream of the blowing-suction boundary condition, which yields Re_θ of up to 235k, see also Tab. 1. Figure 7 shows the velocity profiles in inner (left) and outer scaling (right). Note that the matching height in outer scaling remains almost constant for all cases since the matching location is specified in terms of Δ. A very good agreement with the logarithmic law of the wall is observed for all cases, even for very high values of $y^+ > 5000$. Reynolds stress profiles show the expected shape and magnitude;

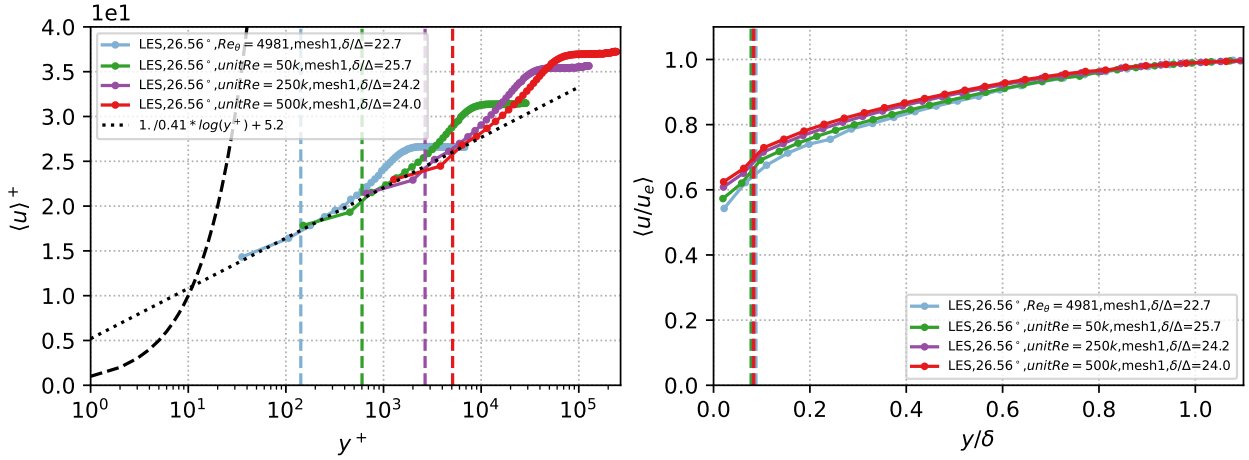


Fig. 7 Velocity profiles in inner and outer scaling at different Reynolds numbers using *mesh1* and a plate angle of $26.56°$.

however, given the lack of reference data we omit the presentation here. We summarize the obtained results by plotting the skin friction coefficient C_f as function of Re_θ for cases 2, 6, 7, and 8 in Fig. 8. Results obtained with the present LES method lay within the uncertainty range of the Kármán-Schoenherr (KS) empirical relation. The results indicate that the proposed WMLES framework can predict the viscous drag at high Reynolds numbers accurately.

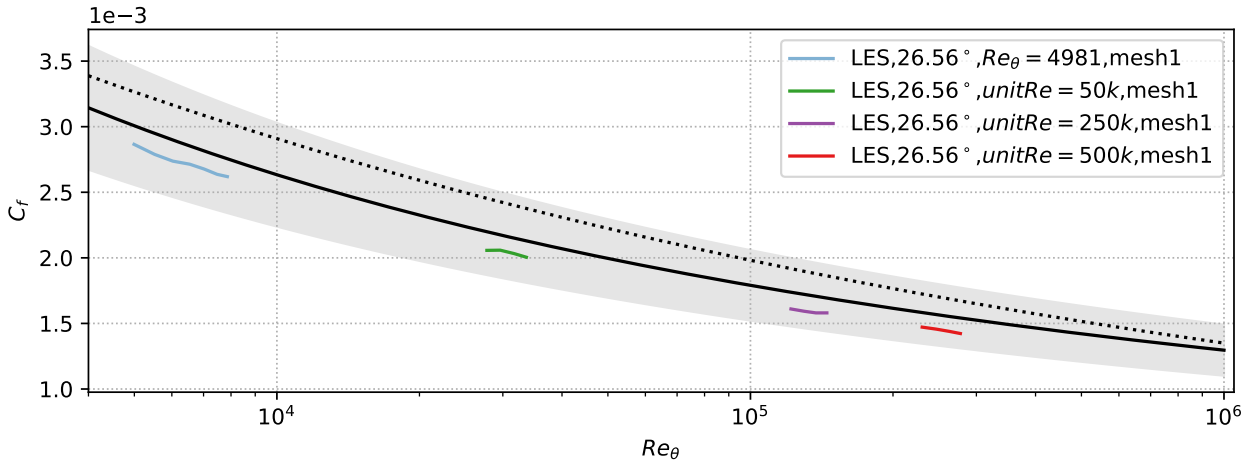


Fig. 8 Skin friction as a function of Re$_\theta$. (——) The KS empirical relation $Cf = [17.08\log_{10}(Re_\theta)^2 + 25.11\log_{10}(Re_\theta) + 6.012]^{-1}$; (·····) power law $Cf = 0.0135Re_\theta^{-1/6}$. The gray shaded region represents the $+-15\%$ range from the KS relation. References for the empirical relations can be found in Tamaki and Kawai [19].

C. High Lift 3 Prediction Workshop

In order to assess the accuracy of the newly developed flow solver for complex cases, WMLES are carried out for a realistic aircraft model in landing configuration, the so-called *JAXA Standard Model* (JSM). This validation case has received considerable attention since it was the focus of the *3rd High Lift Prediction Workshop* [23]. Most of the

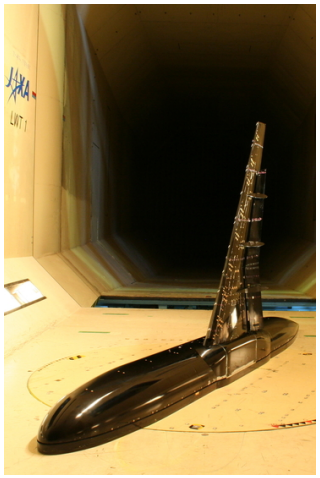


Fig. 9 JAXA Standard Model mounted in the $6.5\,m \times 5.5\,m$ low-speed wind tunnel. Image retrieved from [23].

contributions to the workshop were steady RANS simulations, however, a few groups also used unsteady methods such as unsteady RANS or lattice Boltzmann very-large-eddy simulation [24]. More recently, WMLES results were presented by Goc et al. [25, 26], demonstrating not only a high level of accuracy with respect to the prediction of aerodynamic coefficients across the full angle of attack range, but also turnaround times of less than a day on supercomputers available to the aviation industry.

Figure 9 shows an image of the model (nacelle off) mounted in the wind tunnel. The JSM model has a mean aerodynamic chord of 529.2 mm and a wing half-span of 2300 mm. The experiments were conducted with a free stream Mach number of 0.172 and a chord-based Reynolds number of 1.96×10^6. The experimental data has originally been collected by Yokokawa et al. [27, 28] and is available through the homepage of the *3rd High Lift Prediction Workshop* [23].

Name	Min cell size [$\times 10^{-3}$ mm]	Number of blocks	Number of cells [$\times 10^{6}$]
mesh1	3.05	17 522 (15 390)	71 (63)
mesh2 (baseline)	1.52	37 682 (32 402)	154 (132)
mesh3	0.76	120 884 (98 919)	495 (405)

Table 3 Mesh settings for simulation of the JSM. The block and cell count in parentheses excludes grid blocks fully immersed in the solid region.

1. Simulation set-up

We consider the JSM model in free air, i.e., without including the wind tunnel geometry. The fluid domain extends over 50.0 m × 50.0 m × 50.0 m. The obtained simulation results for the lift, drag and moments are therefore compared with the experimental measurements that have been corrected for the blockage due to the wind tunnel walls. Three different computational meshes are considered for which the settings are summarized in Tab. 3. An overview of the *mesh2* grid is shown in Fig. 10. The figure shows the grid blocks, where each block has 16^3 cells, and highlights the local mesh refinement. Every part of the aircraft is refined uniformly except for the fuselage, which is kept at *mesh1* resolution for all three mesh configurations. A symmetry boundary condition is applied at the domain boundary aligned

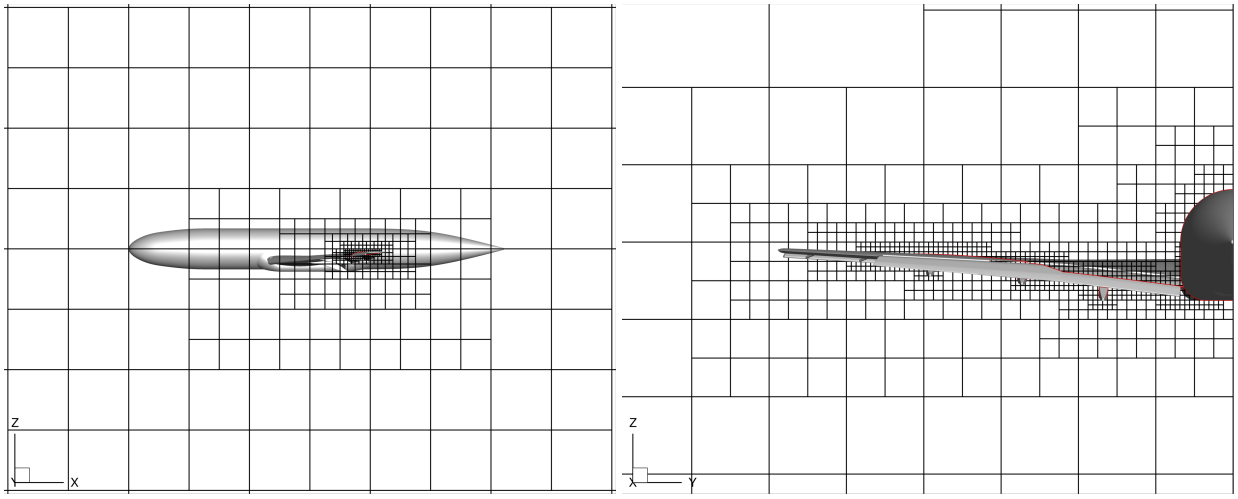


Fig. 10 Grid blocks for simulation of the JSM on *mesh2*. Black lines are block edges, while red lines are cut lines through the geometry. Each block contains 16^3 cells.

with the airplane symmetry plane, a constant static pressure is imposed at the downstream domain boundary (domain outlet), and non-reflective boundary conditions based on Riemann invariants are imposed on the remaining four domain sides. A full angle of attack sweep, comprising $\alpha = \{4.36°, 14.54°, 18.58°, 21.57°, 22.54°\}$, has been performed for *mesh1* and *mesh2*. WMLES on the third mesh, *mesh3*, have been performed for angles $\alpha = \{4.36°, 18.58°, 21.57°\}$.

2. Results

Statistical convergence is assessed by plotting the time history of the lift coefficient C_L in Fig. 11 for *mesh2*. All flow statistics are averaged over 10 convective flow through times over $l_{ref} = 1$ m in the non-dimensional time-interval $t\, u_{ref}/l_{ref} \in [20, 30]$, see also the grey shared area in Fig. 11. Contours of mean non-dimensional wall distance $\langle y^+ \rangle$ for all mesh configurations at $\alpha = 21.57°$ are shown in Fig. 12.

Figure 13 shows lift curve, drag polar and pitching moment curve obtained by all participants of the *3rd High Lift Prediction Workshop* (grey curves) and experimental reference data (black lines). It is worth noting that all CFD data predict higher drag coefficients than the blockage-corrected wind tunnel experiment, see Fig. 13b, and that prediction of the different CFD models enclose a wide scatter range, especially near and beyond $C_{L,max}$. Results obtained by the present numerical framework are shown as box and whiskers to visualize the interquartile range together with minimum and maximum recorded values. The predictions for the lift coefficient are shown in Fig. 13a. The mesh sensitivity study

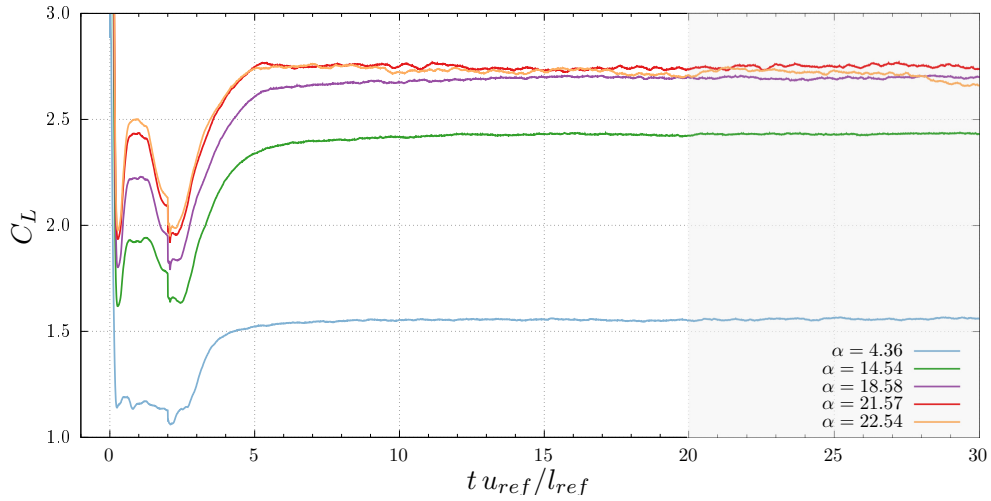


Fig. 11 C_L **as function of time on *mesh2*. Statistics are gathered in the time interval** $tu_{ref}/l_{ref} \in [20, 30]$**.** (—) $\alpha = 4.36°$, (—) $\alpha = 14.54°$, (—) $\alpha = 18.58°$, (—) $\alpha = 21.57°$ (—) $\alpha = 22.54°$

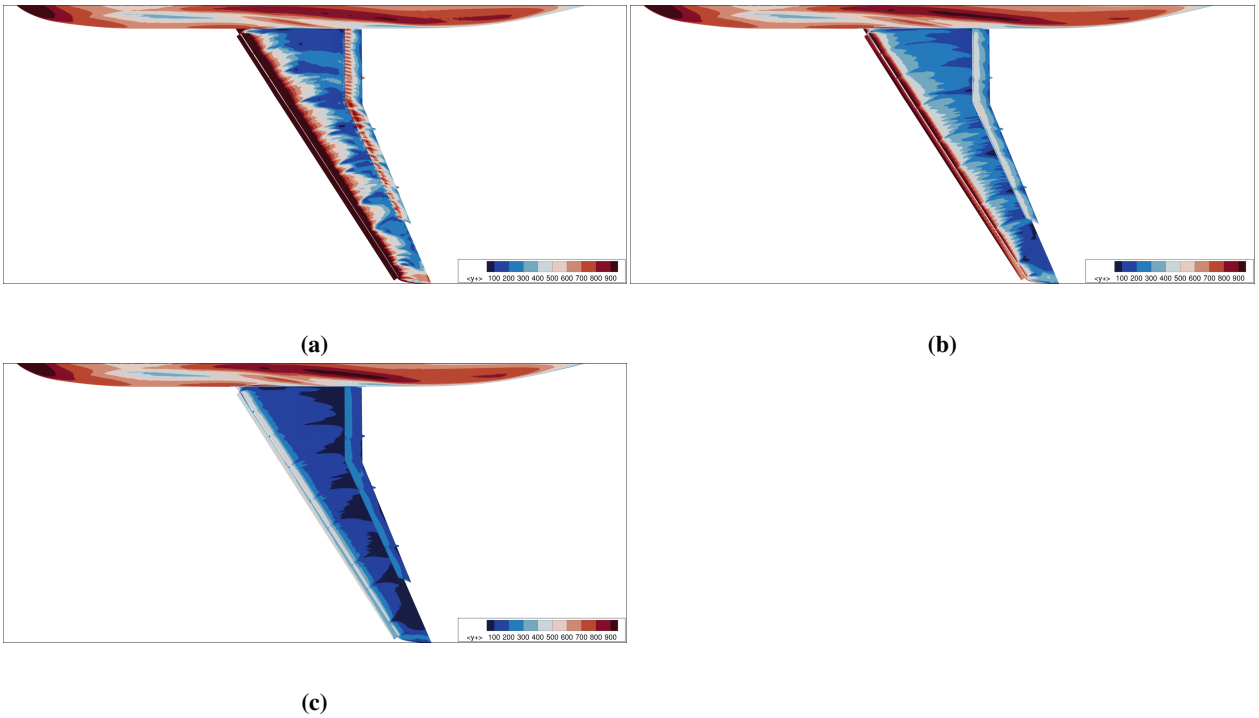
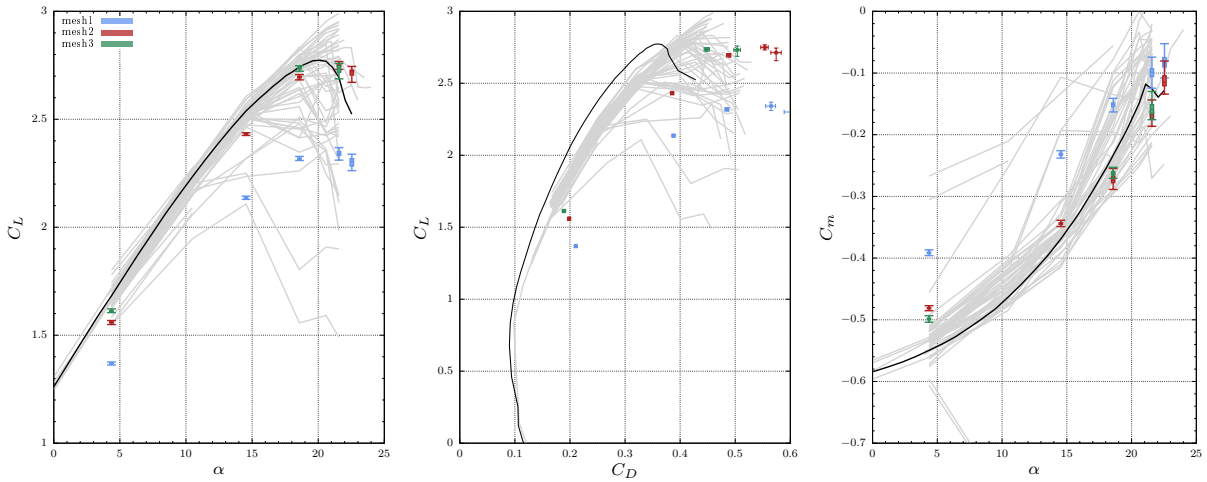


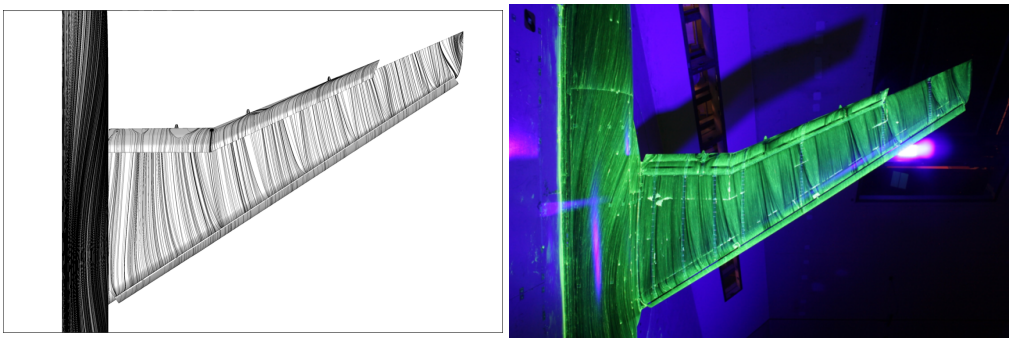
Fig. 12 Contour plot of mean non-dimensional wall distance $\langle y^+ \rangle$ **for** $\alpha = 21.57°$ **on (a) *mesh1*, (b) *mesh2*, (c) *mesh3*.**



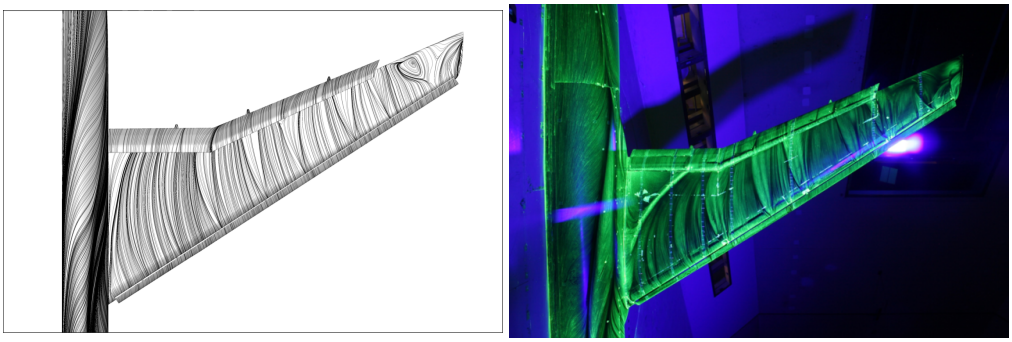
(a) **(b)** **(c)**

Fig. 13 (a) Lift curve, (b) drag polar and (c) pitching moment curve: (——) other workshop participants, (——) experimental results, boxes with whiskers represent present solver.

reveals that *mesh1* is too coarse to predict the correct lift magnitude, while *mesh2* already yields a reasonable agreement with the experimental measurements, which is further improved on *mesh3*. The drag polar shown in Fig. 13b shows a shift of $\Delta C_D \approx 0.05 - 0.15$. We expect to overestimate the drag since our wall model applies a turbulent wall-shear stress independent of the flow state and no transition model is used. However, it appears that the observed shift in drag is large compared to several CFD references. According to Goc et al. [26], it is required to include wind tunnel effects and to compare against uncorrected data to overall improve the drag prediction. The pitching moment coefficient is shown in Fig. 13c. Our numerical prediction on the baseline mesh closely follows the experimental curve. A feature that is missing in the numerical prediction is the kink in the post-stall region that leads to an almost constant C_m in the experimental data. According to Goc et al. [26], this kink is related to an inboard separation that can only be accurately captured by including wind tunnel effects. The under-prediction of the inboard stall mechanism in our simulations can also be seen in Fig. 14, where we compare numerical and experimental oil flow images at two selected angles of attack. While the inboard separation appears to be underestimated in the numerical results, the outboard separation is qualitatively matched well at $\alpha = 21.57°$. Overall, a good agreement is found between numerical and experimental oil flow images. The inboard and outboard separation can also be visualized using the Q-criterion. Figure 15b clearly shows the development of large-scale vortical structures in these regions. The pressure coefficient C_p is compared to experimental reference data for two angles of attack (4.36°, 21.57°) and all mesh configurations in Fig. 17 and Fig. 18. The corresponding wing sections are given in Fig. 16. For both angles of attack (attached and post-stall flow state), sectional pressure predictions are in very good agreement with experimental data for the baseline *mesh2* and the fine *mesh3* up to midspan. Moving further outboards, the suction peak is missed for all mesh configurations, indicating that an even finer mesh would be required to accurately resolve the thin laminar boundary layer on the slat.

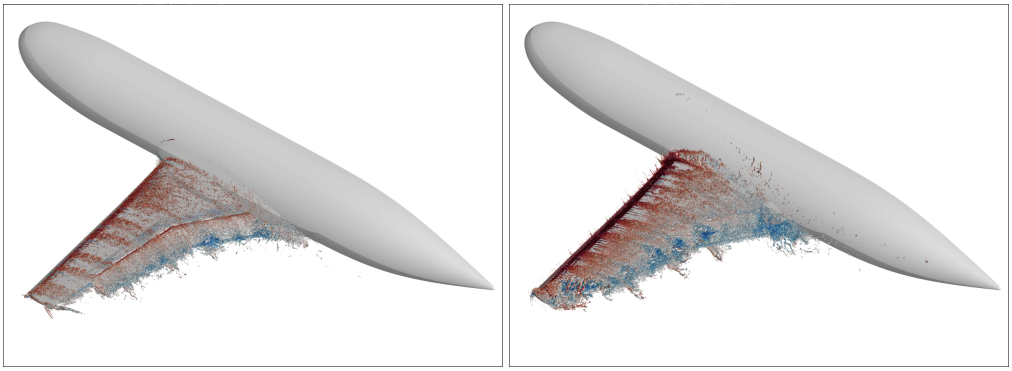


(a) Numerical results at $\alpha = 4.36°$. **(b) Experimental data at $\alpha = 4.36°$.**



(c) Numerical results at $\alpha = 21.57°$. **(d) Experimental data at $\alpha = 21.57°$.**

Fig. 14 Comparison of numerical and experimental oil flow images: (top) $\alpha = 4.36°$, (bottom) $\alpha = 21.57°$.



(a) **(b)**

Fig. 15 Iso-surfaces of Q-criterion colored by streamwise velocity at (a) 4.36° and (b) 21.57° angle of attack.

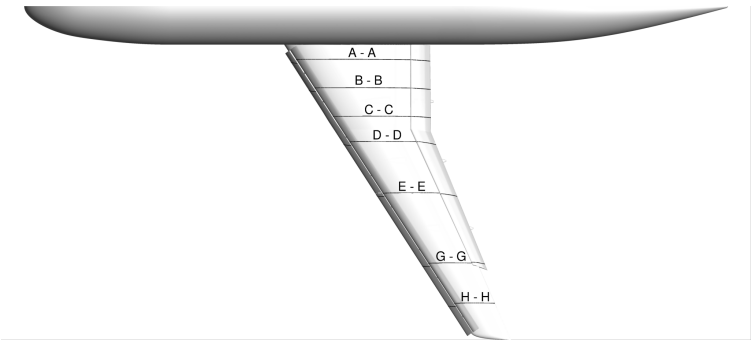


Fig. 16 Wing section cuts used for pressure coefficient comparison.

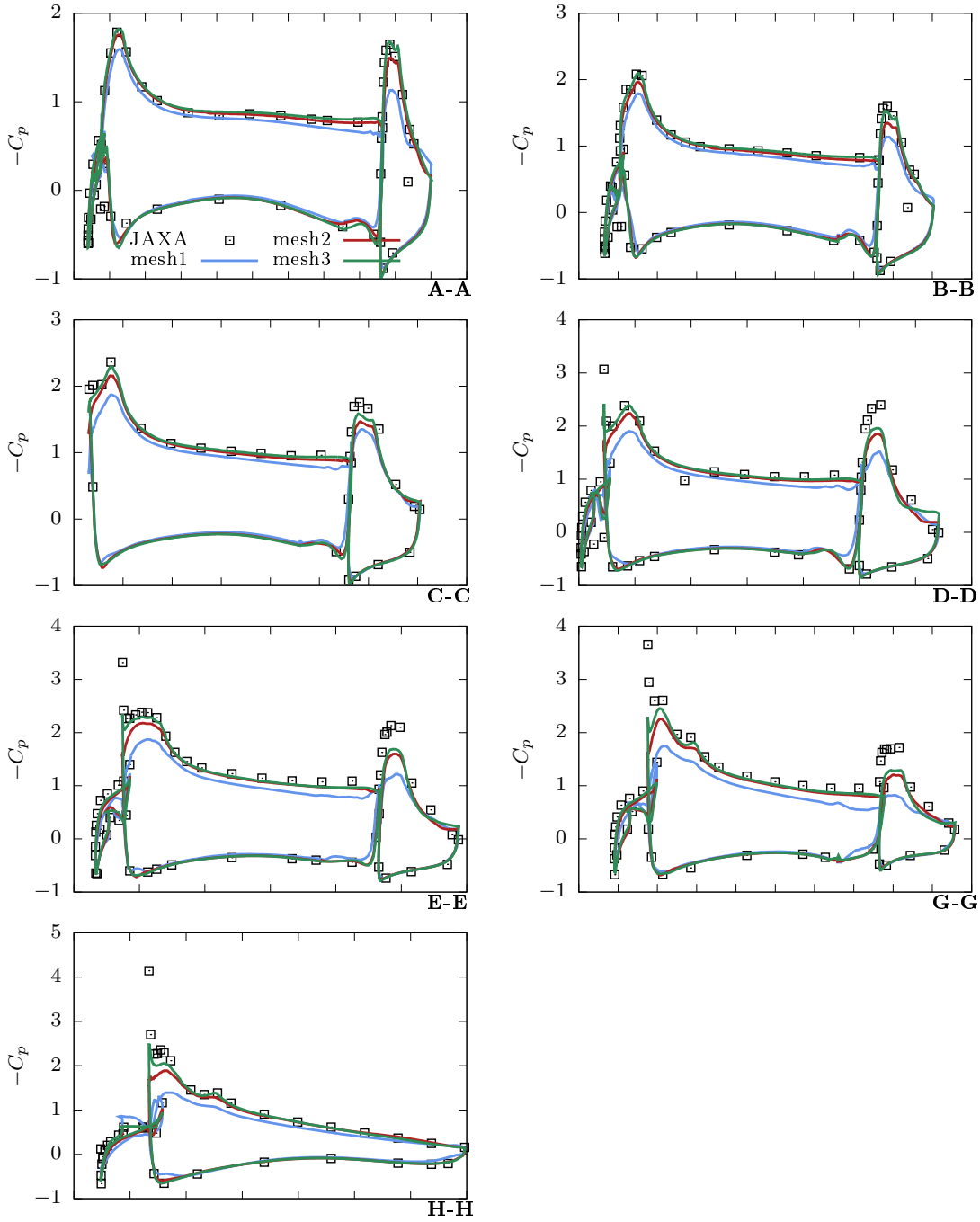


Fig. 17 Comparison of pressure coefficient with experimental reference data for different wing sections at $\alpha = 4.36°$.

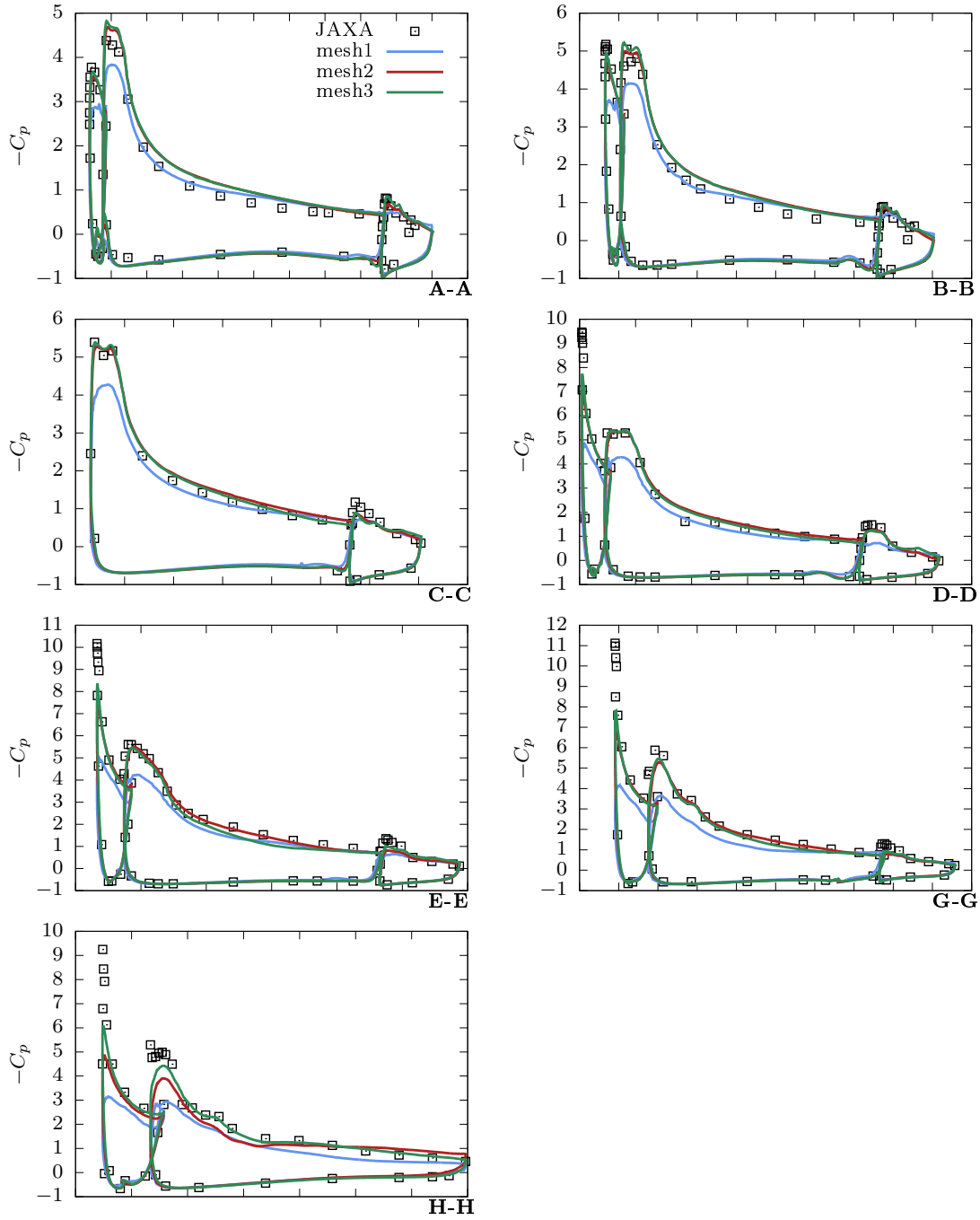


Fig. 18 Comparison of pressure coefficient with experimental reference data for different wing sections at $\alpha = 21.57°$.

IV. Performance

Throughout the development of our LES solver, great effort has been put into achieving maximum performance on the chosen compute hardware. The code is highly optimized for running on NVIDIA GPUs. All performance measurements reported in the following were obtained on 8 A100 GPUs. The following optimization aspects have been considered during the code development cycle:

- **Memory management**: Since the code was developed from scratch to run on GPUs, we put special care into leaving all data in the GPUs' memory and only transfer large amounts of data during initialization and for writing results to disk. As part of the effort to leave all data in GPU memory, we employ a CUDA aware OpenMPI library for all inter GPU communication.
- **Single precision**: Another critical contribution to the overall performance of any flow solver is the numerical framework. All numerics are optimized to be run in single precision. As described in Section II we opted for a low-cost central scheme that is very memory efficient. The memory efficiency leads to numeric kernels being compute bound in contrast to other methods like Lattice Boltzmann which typically are memory bandwidth limited.
- **Remote communication**: To increase the efficiency of block connections communicating over MPI, all data which is sent from one process to another is packed into a single large data buffer. Further, remote and local communication overlap by using CUDA streams.
- **CUDA streams**: Besides utilizing several GPUs for parallelization, the use of CUDA streams and hiding memory transfers next to compute workloads provide additional layers of parallelism wherever feasible.
- **CPU loops**: For blocks and geometry handling we aim at avoiding any kind of looping on the CPUs, and rather aim for optimal numbers of CUDA threads and streams.
- **Load balancing**: Using the partitioning from the p4est library, which builds upon a z-ordering space-filling curve, a large compute imbalance with respect to the immersed boundary was observed while the number of blocks per partition is balanced. This has a large impact on the overall solver performance. We therefore employ the METIS library [20] for multi-objective partitioning based on the *(i)* number of grid cells and *(ii)* number of cut cells per partition.

	p4est			METIS		
Process	n_{blocks}	n_{cells}	$n_{cut-cells}$	n_{blocks}	n_{cells}	$n_{cut-cells}$
0	4710	19 292 160	324 487	4557	18 665 472	311 943
1	4710	19 292 160	326 453	4704	19 267 584	306 551
2	4710	19 292 160	467 870	4745	19 435 520	315 581
3	4711	19 296 256	376 217	4757	19 484 672	313 692
4	4710	19 292 160	375 026	4757	19 484 672	313 301
5	4710	19 292 160	279 943	4757	19 484 672	307 967
6	4710	19 292 160	166 916	4656	19 070 976	315 350
7	4711	19 296 256	182 834	4749	19 451 904	315 361

Table 4 Load balancing for the JSM *mesh2* simulation using p4est and METIS. The computational domain is decomposed in 37682 blocks with a total of 154×10^6 cells and 2.5×10^6 cut cells.

To highlight the importance of the load balancing, the partitioning obtained with p4est and METIS is shown in Tab. 4 for the JSM *mesh2*. The table indicates a perfect balancing of the number of blocks per process when using the p4est library, however, a large load imbalance is observed with respect to the number of cut cells (compare for instance process 2 and 6). Using the multi-objective partitioning capabilities of METIS, the computational load is more evenly distributed with respect to both number of blocks and cut cells.

Table 5 displays run times and computational mesh sizes of the JSM case discussed in Section III.C. The table shows how the solver scales with different mesh sizes running a single time step. It indicates that the present solver hasn't reached yet the scaling limit for the considered problem sizes.

A performance overview for a full JSM simulation based on 30 flow passes over the aerodynamic mean chord is given in Tab. 6. For this comparison the time step is non-dimensionalized using the flow pass time over the wing mean aerodynamic chord. We included performance data from the charLES solver [29] as a point of reference. To

Cell count $\times 10^6$	Wall-clock time per iteration [ms]	Cell count ratio	Wall-clock time ratio
72	3.8	1.0	1.0
154	6.7	2.15	1.8
495	18.6	6.9	4.9

Table 5 Scaling test of the present code using the High Lift 3 Prediction Workshop setup from Section III.C.

compare the performance of both solvers with each other, table rows with roughly the same time step have to be chosen, since this implies similar mesh resolution in the near wall region of the computational grid (both solvers use explicit time integration with a CFL-controlled time step). Based on the original *mesh3* we have included a grid-optimized *mesh3-opt*, which differs only in the wall resolution of the main wing pressure side without compromising accuracy significantly. This specific setup has been also run on 8 V100 GPUs. Please note that in general a different compute hardware has been used for the charLES solver, i.e. , 8 V100/A100 GPUs for the present solver and 96 V100 GPUs for charLES. We see that the A100 GPU is twice faster than the V100 GPU. Compared to charLES, the present solver achieves comparable turnaround times on a 12 times smaller system.

Solver	Cell count $\times 10^6$	Non-dim. time step	Wall-clock time to 30 flow passes	Hardware
present solver (*mesh1*)	72	$2.48 \cdot 10^{-4}$	7 min	8 A100
present solver (*mesh2*)	154	$1.22 \cdot 10^{-4}$	27 min	8 A100
present solver (*mesh3*)	495	$6.08 \cdot 10^{-5}$	2.5 h	8 A100
present solver (*mesh3-opt*)	226	$6.14 \cdot 10^{-5}$	1.4 h	8 A100
present solver (*mesh3-opt*)	226	$6.14 \cdot 10^{-5}$	2.9 h	8 V100
charLES	32	$6.47 \cdot 10^{-5}$	48 min	96 V100
charLES	157	$3.23 \cdot 10^{-5}$	7 h	96 V100

Table 6 Computational cost summary for 30 flow passes of the High Lift 3 Prediction Workshop setup from Section III.C at $\alpha = 18.58°$. Timings from charLES solver [29] are given as a point of reference.

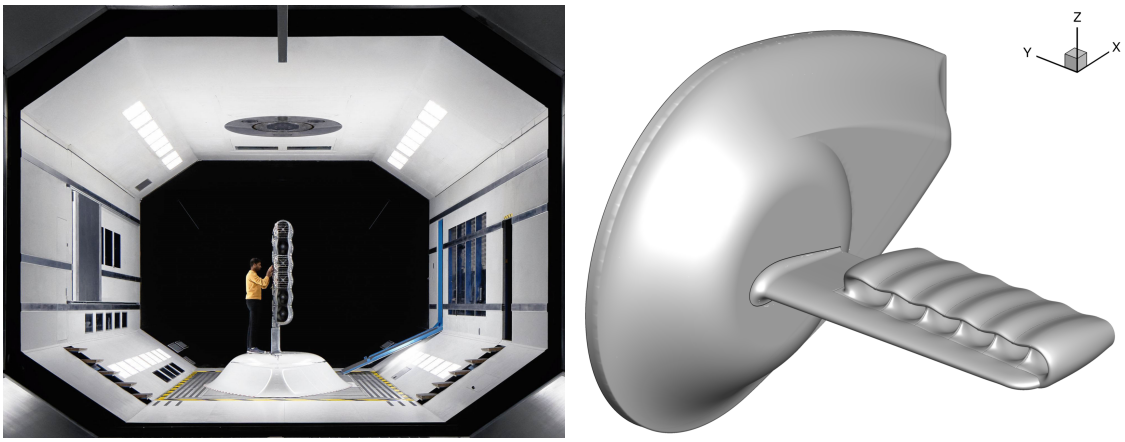
V. Application to Lilium's Experimental 6-Engine Jet Canard Setup

With the CFD solver presented in this work, we aim to enrich the understanding of complex flow features generated by the eVTOL Lilium Jet. The aircraft experiences very different flow regimes during the standard flight envelope, low speed propulsion driven flows, separated flow regions in transition flight and attached flow in cruise. To this end, the present LES solver provides a much-needed complementary tool to the more common RANS or unsteady-RANS simulations, the latter of which can only partially help analyzing the inherently unsteady aircraft aerodynamics in non-cruise conditions. In the following numerical results are compared with selected experimental data from the Lilium wind tunnel test campaign.

1. Experimental set-up

A full size Lilium Jet canard was tested in the RUAG LWTE facility. The tunnel setup is shown in Fig. 19a. The wind tunnel model consists of a fixed canard geometry with 6 electric jet engines mounted in individually actuated movable flaps. The balance measures integral forces and moments on the canard and flaps, excluding the fairing. The engine characteristics for a given operating condition, including the engine mass flow rate, are obtained from engine rakes that are mounted at the exit plane of three out of six engines.

The test condition selected in the following section is obtained from a slow sweep of the uniform flap bank whilst the engine rotation rate is kept constant throughout. The nature of the changing inlet flow leads to changes in engine thrust and mass flow rate throughout the flap angle range. Further, the velocity in the test section is set at the start of the run and consequently varies as the blockage changes throughout the flap angle change. The flow on the leading edge of the flaps is tripped using sand-paper.



(a) Experimental setup. **(b) Numerical free air setup.**

Fig. 19 Overview of the 6-engine Lilium Jet canard setup.

2. Simulation set-up

	Part	Min cell size [$\times 10^{-3}$]	Number of blocks	Number of cells [$\times 10^{6}$]
mesh1	Baseplate	3.90	11 124	$\approx$ 365
	Canard	1.95		
	Flap	1.95		
	Upper flap surface	1.95		
mesh2	Baseplate	3.90	20 812	$\approx$ 682
	Canard	1.95		
	Flap	1.95		
	Upper flap surface	0.98		

Table 7 Mesh settings for the canard simulation.

The canard is simulated in free air with a domain size of 32 m $\times$ 32 m $\times$ 32 m. The obtained simulation results for the lift are therefore compared with the experimental measurements that have been corrected for wind-tunnel blockage effects. The unit Reynolds number is Re/m $\approx 2.0 \times 10^6$ and the free stream Mach number is $Ma = 0.1$. Two different computational meshes are considered for which the settings are summarized in Tab. 7. Only the upper flap surface is further refined for *mesh2*, while the baseplate, canard and remaining flap surface are kept at the *mesh1* resolution. For this setup each block contains 32^3 cells.

The domain boundary from which the canard plinth extends out is defined as a symmetry plane, the downstream boundary as static pressure outlet, and the remaining domain sides as non-reflective boundary conditions based on Riemann invariants. The entire canard model, including the plinth, are defined as viscous wall-modeled surfaces. The electric jet engines are modeled by a volume source-term that avoids resolving the engine rotor blades and nozzle guide vanes. A PI controller is used to adjust the volume force dynamically in such a way that the target mass flow rate is achieved. A non-dimensional mass flow of $C_\mu = \dot{m}/(A_{fan} U_\infty \rho_\infty) = 2.65$ is used for the following simulations in accordance with the experimental setup.

The current wall-modeling approach is unable to accurately predict thin laminar boundary layers on coarse meshes. It is common practice to apply some form of tripping, forcing the boundary layer to be fully turbulent, such that turbulent wall models could be applied throughout [30]. This technique has been also applied to the leading edge of the flap and is in accordance with the experimental setup.

3. Results

Three different flap angles, $\{29.7°, 40.2°, 50.3°\}$, have been simulated on *mesh1* and *mesh2*. The lift coefficient is shown in Fig. 20 and compared to experimental data. An overall good agreement is found with a slight over prediction in the linear and post-stall regime. Results improve for high flap deflections on *mesh2*. The engine mass flow setting could be an error source, since the simulation assumes a constant C_μ for all engines independent of the flap angle, while experimental measurements revealed a flap-angle and engine dependent variation of the mass flow on the order of $5 - 10\%$. Flow visualizations in terms of instantaneous and mean streamwise velocity together with iso-surfaces of

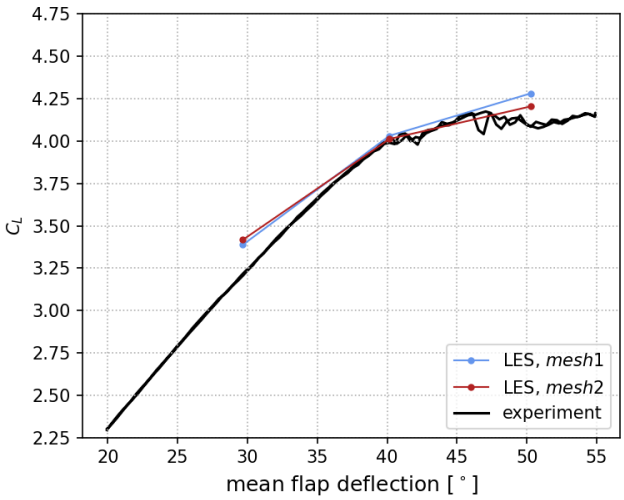


Fig. 20 Lift polar comparison for all mesh configurations.

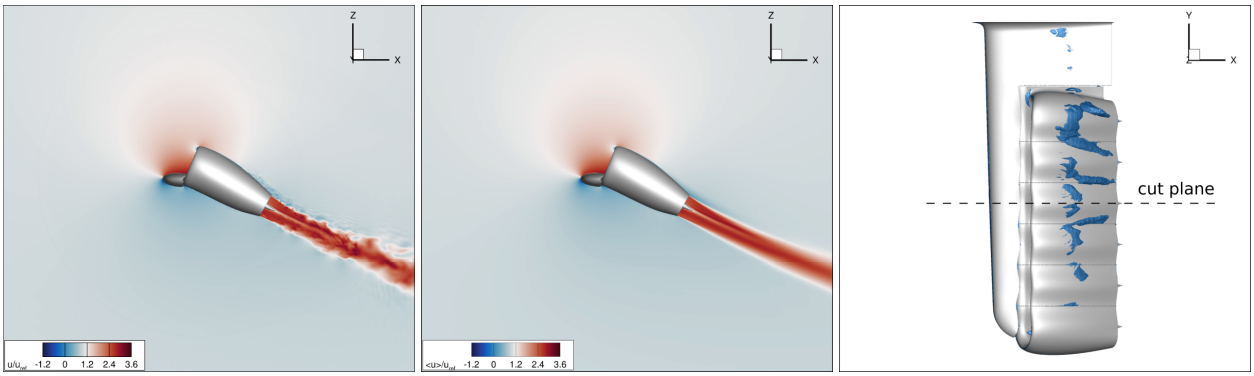
zero velocity are shown in Fig. 21. The side-views in the left and mid column represent cut planes through the center of the third inboard flap, see also the dashed horizontal lines in Figs. 21c, 21f and 21i. At the lowest flap angle, 29.7°, the flow is attached on the flap top surface, while 40.2° flap deflection indicates the onset of stall with a separation being formed on the second inboard flap. Further increasing the flap angle to 50.3° shows the presence of a large stall cell centered around the third inboard flap with the flow detaching right from the leading edge of the flap top surface.

VI. Conclusion

We have demonstrated that wall-modeled large-eddy simulations (WMLES) for practical aeronautical applications can be carried out within a few minutes or hours with very good accuracy in predicting design-critical properties, such as integrated forces. This is possible by leveraging modern GPU architectures such as the NVIDIA A100 together with carefully selected and tailored algorithms.

The present solver is based on a ghost-cell immersed boundary method (IBM) and coupled to a WMLES approach in a weakly-compressible finite-volume framework. A simple block-Cartesian adaptive grid with a fixed amount of cells per block has been selected, allowing for optimized algorithms with very high GPU occupancy. Special care was taken to equally balance the compute load of regular grid cells and the increased workload of the immersed boundary method among all partitions. This optimal load balancing is crucial for the overall performance and the scaling characteristics of the code, and has been achieved through a multi-objective partitioning in the present work.

The IBM-WMLES solver has been rigorously verified and validated against well-known academic test cases, such as the turbulent channel flow and the inclined turbulent boundary layer. Results have shown good accuracy of the present method on coarse meshes and robust performance independent of the cut cell configuration. WMLES are carried out for a realistic aircraft model in landing configuration, the so-called JAXA Standard Model (JSM). Sufficient accuracy with respect to the overall lift polar is achieved on a baseline mesh with turnaround times of less than an hour and using only 8 NVIDIA A100 GPUs, making WMLES of practical aeronautical applications accessible to the industry at an affordable cost. Finally, WMLES of Lilium's experimental 6-engine jet canard setup were presented, for which the lift coefficient at flap angles both in the attached and stalled flow regime agrees well with experimental measurements.



(a) u/u_{ref} **at** 29.7° **(b)** $\langle u \rangle / u_{ref}$ **at** 29.7° **(c)** $\langle u \rangle = 0$ **iso-surface at** 29.7°

(d) u/u_{ref} **at** 40.2° **(e)** $\langle u \rangle / u_{ref}$ **at** 40.2° **(f)** $\langle u \rangle = 0$ **iso-surface at** 40.2°

(g) u/u_{ref} **at** 50.3° **(h)** $\langle u \rangle / u_{ref}$ **at** 50.3° **(i)** $\langle u \rangle = 0$ **iso-surface at** 50.3°

Fig. 21 Instantaneous u/u_{ref} (left) and mean velocity $\langle u \rangle / u_{ref}$ contours (middle) for a y-normal slice cutting through the center of the third inboard flap. Iso-surfaces of $\langle u \rangle = 0$ (right). The flap angle is increased from top to bottom. Results are shown for *mesh2*.

Acknowledgments

The authors would like to thank the following people who are or have been involved in this project: Amar Bansal, Nele Bittel, Florian Krause and Chris Shields. A special thanks goes to Matthijs Van Ede and Kin Sam Lam who worked on validation tasks during their internship at Lilium. All Aerodynamics Software Developers contributed equally to the development of the present LES solver.

Bibliography

[1] Slotnick, J. P., Khodadoust, A., Alonso, J. J., Darmofal, D. L., Gropp, W. D., Lurie, E. A., and Mavriplis, D. J., "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences," Tech. Rep. NASA/CR-2014-218178, 2014.

[2] Cary, A. W., Chawner, J., Duque, E. P., Gropp, W., Kleb, W. L., Kolonay, R. M., Nielsen, E., and Smith, B., "Cfd vision 2030 road map: Progress and perspectives," *AIAA AVIATION 2021 FORUM*, 2021.

[3] Brelje, B. J., and Martins, J. R., "Electric, hybrid, and turboelectric fixed-wing aircraft: A review of concepts, models, and design approaches," *Progress in Aerospace Sciences*, Vol. 104, 2019, pp. 1–19.

[4] Piomelli, U., and Balaras, E., "Wall-layer models for large-eddy simulations," *Annual Review of Fluid Mechanics*, Vol. 34, No. 1, 2002, pp. 349–374.

[5] Bose, S. T., and Park, G. I., "Wall-modeled large-eddy simulation for complex turbulent flows," *Annual Review of Fluid Mechanics*, Vol. 50, 2018, pp. 535–561.

[6] Larsson, J., Kawai, S., Bodart, J., and Bermejo-Moreno, I., "Large eddy simulation with modeled wall-stress: Recent progress and future directions," *Mechanical Engineering Reviews*, Vol. 3, No. 1, 2016, pp. 15–00418.

[7] Peskin, C. S., "Flow patterns around heart valves: A numerical method," *Journal of Computational Physics*, Vol. 10, No. 2, 1972, pp. 252–271.

[8] Mittal, R., and Iaccarino, G., "Immersed boundary methods," *Annual Review of Fluid Mechanics*, Vol. 37, 2005, pp. 239–261.

[9] Burstedde, C., Wilcox, L. C., and Ghattas, O., "`p4est`: Scalable Algorithms for Parallel Adaptive Mesh Refinement on Forests of Octrees," *SIAM Journal on Scientific Computing*, Vol. 33, No. 3, 2011, pp. 1103–1133.

[10] Niemeyer, K. E., and Sung, C.-J., "Recent progress and challenges in exploiting graphics processors in computational fluid dynamics," *The Journal of Supercomputing*, Vol. 67, No. 2, 2014, pp. 528–564.

[11] MacCalla, W., and Kulkarni, S., "Utilizing GPUs to Accelerate Turbomachinery CFD Codes," Tech. Rep. NASA/TM-2016-218947, 2016.

[12] Jespersen, D. C., "Acceleration of a CFD code with a GPU," *Scientific Programming*, Vol. 18, No. 3-4, 2010, pp. 193–201.

[13] Reguly, I. Z., Mudalige, G. R., Bertolli, C., Giles, M. B., Betts, A., Kelly, P. H., and Radford, D., "Acceleration of a full-scale industrial CFD application with OP2," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 27, No. 5, 2015, pp. 1265–1278.

[14] McBride, B. J., Zehe, M. J., and Gordon, S., "Glenn coefficients for calculating thermodynamic properties of individual species," Tech. Rep. NASA/TP-2002-211556, 2002.

[15] Vreman, A. W., "An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications," *Physics of Fluids*, Vol. 16, 2004, pp. 3670–3681.

[16] Mittal, R., Dong, H., Bozkurttas, M., Najjar, F., Vargas, A., and Loebbecke, A., "A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries," *Journal of Computational Physics*, Vol. 227, 2008, pp. 4825–4852.

[17] Mittal, R., and Iaccarino, G., "Immersed boundary method," *Annual Review of Fluid Mechanics*, Vol. 37, No. 1, 2005, pp. 239–261.

[18] Vanella, M., and Balaras, E., "A moving-least-squares reconstruction for embedded-boundary formulations," *Journal of Computational Physics*, Vol. 228, No. 18, 2009, pp. 6617–6628.

[19] Tamaki, Y., and Kawai, S., "Wall modeling for large-eddy simulation on non-body-conforming Cartesian grids," *Physical Review Fluids*, Vol. 6, No. 114603, 2021. https://doi.org/10.1103/PhysRevFluids.6.114603.

[20] "A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs," *SIAM Journal on Scientific Computing*, Vol. 20, 1999, p. 359—392.

[21] Lee, M., and Moser, R. D., "Direct numerical simulation of turbulent channel flow up to $Re_\tau \approx 5200$," *Journal of Fluid Mechanics*, Vol. 774, 2015, p. 395–415. https://doi.org/10.1017/jfm.2015.268.

[22] Schlatter, P., and Örlü, R., "New DNS Data: Vortical and Large Scale Structures in a Turbulent Boundary Layer," *J. Fluid Mech.*, Vol. 659, 2010, pp. 116–126.

[23] NASA, "3rd AIAA CFD High Lift Prediction Workshop (HiLiftPW-3)," , 2017. URL https://hiliftpw.larc.nasa.gov/index-workshop3.html.

[24] Konig, B., Fares, E., Murayama, M., and Ito, Y., "PowerFLOW Simulations for the Third AIAA High-Lift Prediction Workshop," 2018.

[25] Goc, K., Bose, S., and Moin, P., "Wall-Modeled Large Eddy Simulation of an Aircraft in Landing Configuration," 2020.

[26] Goc, K. A., Lehmkuhl, O., Park, G. I., Bose, S. T., and Moin, P., "Large eddy simulation of aircraft at affordable cost: a milestone in computational fluid dynamics," *Flow*, Vol. 1, 2021, p. E14.

[27] Yokokawa, Y., Murayama, M., Ito, T., and Yamamoto, K., "Experiment and CFD of a High-Lift Configuration Civil Transport Aircraft Model," *25th AIAA Aerodynamic Measurement Technology and Ground Testing Conference*, 2006.

[28] Yokokawa, Y., Murayama, M., Kanazaki, M., Murota, K., Ito, T., and Yamamoto, K., "Investigation and Improvement of High-Lift Aerodynamic Performances in Lowspeed Wind Tunnel Testing," *46th AIAA Aerospace Sciences Meeting and Exhibit*, 2008.

[29] Goc, K. A., Lehmkuhl, O., Park, G. I., Bose, S. T., and Moin, P., "Large eddy simulation of aircraft at affordable cost: a milestone in computational fluid dynamics," *Flow*, Vol. 1, 2021, p. E14. https://doi.org/10.1017/flo.2021.17.

[30] Dauricio, E. T., and Azevedo, J. L. F., "A Wall Model for External Laminar Boundary Layer Flows Applied to the Wall-Modeled LES Framework," *Journal of Computational Physics*, 2022. https://doi.org/10.2139/ssrn.4142016.