



HPC Requirements of High-Fidelity Flow Simulations for Aerodynamic Applications

Axel Probst¹(✉), Tobias Knopp¹, Cornelia Grabe¹, and Jens Jägersküpper²

¹ German Aerospace Center (DLR), Institute of Aerodynamics and Flow Technology, 37073 Göttingen, Germany
axel.probst@dlr.de

² German Aerospace Center (DLR), Institute of Aerodynamics and Flow Technology, 38108 Braunschweig, Germany

Abstract. This paper relates the computational demand of turbulence-resolving flow simulations for aircraft aerodynamics to the parallel scalability of the DLR flow solvers TAU and THETA, as well as the new CODA solver optimized for many-core HPC systems. Based on existing lower-fidelity simulations, the computational requirements for wall-resolved LES are first estimated for single aircraft components at wind-tunnel scale. It is shown that such simulations at reduced Reynolds numbers would be realizable within days to weeks with the current methods, if the largest available HPC clusters with more than 100,000 cores were used. However, an extrapolation to the HPC requirements of LES for a full 3D wing of an aircraft at flight Reynolds numbers highlights the urgent need for larger HPC resources and adapted parallel code designs, as well as more efficient numerical algorithms and physical models.

Keywords: High-Performance Computing · Large-Eddy Simulation · Aircraft aerodynamics · TAU · THETA · CODA

1 Introduction

Today's aerodynamic design, optimization and analysis of modern transport aircraft relies on a combination of flight test, wind tunnel testing and numerical simulation. Depending on the aircraft operation mode the contribution of numerical simulation, i.e. Computational Fluid Dynamics (CFD) to these disciplines varies within the flight envelope: while current CFD methods exhibit high maturity in cruise flight close to the design point of the aircraft, less confidence exists for low-speed take-off and landing configurations as well as for certain high-speed operation conditions [10]. State-of-the-art CFD codes employ a Finite Volume discretization of the Navier-Stokes equations. At present, turbulent flow is usually modelled with a Reynolds-Averaged Navier-Stokes (RANS) ansatz through additional transport equations. These RANS turbulence models were derived

and calibrated mainly for attached and steady flow conditions, as appear in cruise flight. During low-speed take-off and landing, the turbulent flow near the aircraft's surface experiences strong adverse pressure gradients, leading to flow separation. In these situations standard RANS turbulence models fail to predict the flow with sufficient accuracy [7]. For high-speed operation, the flow around the aircraft is susceptible to the buffet phenomenon, an unsteady periodic shock wave oscillation involving shock-induced separation. The accurate prediction of buffet is of significant importance, as the latter marks the limit of operation before structural damage of the aircraft may occur. Solutions obtained using RANS turbulence models to predict these flow conditions are strongly dependent on simulation parameters and grid resolution [3] and suitability or correct application of these models are still under investigation [10].

Recently, NASA published the *CFD Vision 2030 Study* [9] to highlight the current status of CFD in aerodynamics and to set up a roadmap for meeting present and future requirements in the field. The *Vision* emphasizes the shortcomings of current RANS turbulence models for certain flow phenomena and identifies the need for developing suitable turbulence resolving methods, e.g. Large Eddy Simulation (LES) methods, in which turbulence is partly modelled and partly resolved. To underline the importance of this development goal, the first of a total of four Grand Challenges that were proposed in the Vision is the LES of a powered aircraft configuration across the entire flight envelope. However, while the application of LES promises accurate predictions at the border of the flight envelope, it also requires large High Performance Computing (HPC) resources and significantly increases the computational effort compared to RANS turbulence models.

This paper gives an overview of the different turbulence modelling approaches applied at DLR to perform computations on industrially relevant aerodynamic configurations. The basics of these methods are introduced, as well as DLR's CFD codes and their respective HPC concepts. The HPC requirements for the simulation of flows around two aircraft components are assessed depending on the employed turbulence modelling approach. This assessment is used for estimating the computational effort of wall-resolved LES applied to a full 3D aircraft wing, both on wind-tunnel scale and for the real aircraft in different flight conditions. The paper concludes with required improvements and developments to achieve these simulations with acceptable computational cost.

2 Aerodynamic Flow Simulation

The computational effort of a flow simulation is governed by the resolution requirement of the spatial and temporal discretization of the partial differential equations to be solved. One key factor is the treatment of turbulence, i.e. small quasi-random unsteady fluctuations which emerge above a certain Reynolds number (defined as $Re = U_\infty l_{ref} / \nu$) and which strongly affect the aerodynamic performance of air vehicles (e.g. drag force, stall behaviour).

While all relevant physics of turbulence can be captured by a *Direct Numerical Simulation* (DNS) based on the Navier-Stokes equations, the requirement to

fully resolve the turbulent fluctuations in both space and time yields extremely large number of grid points (i.e. small mesh-cell sizes) and small physical time steps. Moreover, DNS resolution increases dramatically with the Reynolds number, i.e. the number of grid points N scales with $\sim Re^{37/14}$ in wall-bounded flow [1]. At relevant (flight) Re-numbers of aircraft ($Re = 10^7 - 10^8$) DNS will therefore remain beyond available computing capacities in the foreseeable future, see Fig. 1. This encouraged the development of less demanding methods to compute turbulent flow, which are briefly outlined in the following.

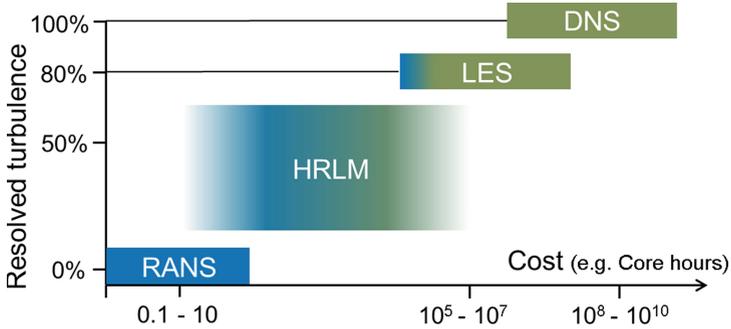


Fig. 1. Schematic relation of turbulence resolution level and computational cost in different flow simulation approaches.

RANS Modelling. The *Reynolds-averaged Navier-Stokes* (RANS) approach applies a temporal averaging to the Navier-Stokes equations which converts the unsteady turbulent fluctuations into mean statistical quantities, i.e. the Reynolds stresses. These stresses describe the effect of turbulence on the (steady) mean flow and need to be modelled by additional physical closure assumptions, called turbulence models. Such models usually consist of partial differential equations which pose no particular resolution requirements compared to the steady RANS equations. However, they often lack accuracy in complex turbulent flows, e.g. near the border of the flight envelope of aircraft.

Large-Eddy Simulation. In *Large-Eddy Simulations* (LES) the Navier-Stokes equations are formally filtered (often using a top-hat filter based on the local mesh-cell size) so that turbulence is decomposed into an unsteady resolved part and a smaller modelled part, the latter being usually provided by algebraic sub-grid scale models. LES requires high temporal and spatial resolution of the resolved turbulent fluctuations to meet the commonly accepted criterion [5] of at least 80% resolved turbulent kinetic energy (see Fig. 1). For wall-bounded flow, this leads to the number of grid points scaling roughly as $N \sim Re^{13/7}$ [1]. However, due to the reduced empirical modelling, the mean-flow predictions of LES often approach DNS accuracy.

Hybrid RANS/LES. *Hybrid RANS/LES Methods* (HRLM) combine the specific benefits of both methods by locally adapting the applied modelling to the flow topology and available grid resolution. For instance, the well-known *Detached-Eddy Simulation* aims to apply RANS modelling, where usually sufficient (i.e. attached near-wall flow) and LES, where needed (i.e. separated flow). More elaborate variants of DES offer wall-modelled LES capabilities [8] by applying RANS close to the wall and consistent transition to LES in the outer boundary layer, thus reducing the Re-number-dependency to $N \sim Re$ [1]. Note that the physical time step is not easily adapted to the local modelling, i.e. the time resolution in the local LES regions dictates the global time step. Due to the flexible localization of RANS and LES regions, HRLM covers a wide intermediate regime of turbulence resolution and effort, as depicted in Fig. 1.

3 Parallel Scaling Characteristics of DLR Flow Solvers

Numerical simulation tools for aeronautical flows classically rely on the finite-volume approach. At DLR two different finite-volume solvers for unstructured meshes have been developed: the TAU code for compressible flow and the THETA code for incompressible flow. Although both codes are written in C and share common data structures as well as similar implementations of the modelling approaches described in Sect. 2, the specific solution algorithms for flows with variable and constant density lead to different parallel scalability characteristics.

3.1 Parallel Scalability of Present Codes TAU and THETA

For decades, MPI-based domain decomposition has been the dominating parallelization paradigm for mesh-based CFD to make use of HPC resources for flow simulation. The computational domain is statically split into a number of parts, called domains. Each domain maps one-to-one with an MPI process. The domain-local calculations are often stencil-based, for instance the flux balance for each finite (control) volume. As a consequence, data from neighboring domains are needed for control volumes touching a domain boundary. These data are often called the domain's halo. Halo data allow for a complete stencil for each domain-local control volume. As the simulation proceeds, halo data need to be frequently updated. TAU as well as THETA use a lock-step approach: before each stencil loop, halo data gets communicated using point-to-point communication. Namely, each process (domain) posts an `MPI_Irecv` for each neighboring domain, followed by an `MPI_Isend` to each neighbor, and finalized by an `MPI_Waitall` on all these immediate MPI calls. The number of neighbors of a domain can be considered limited, say around the order of 10, independent of the number of domains used (given the mesh is large). The smaller the domains, the fewer halo data is contained in a message, resulting in halo communication becoming latency and/or message-rate bound. In contrast to point-to-point communication for the halo update, global reduce operations result in a communication time growing logarithmically in the number of processes involved. And what is more (actually

worse), load imbalances accumulate at such process-synchronization points. In particular for THETA, which makes use of full implicit time-integration schemes in combination with multi-grid acceleration, global reductions limit the parallel scalability. Due to the different time-integration schemes used for the compressible equations, TAU's scalability is less affected in this regard.

The scalability of both TAU and THETA is examined in exemplary LES computations of the periodic turbulent channel flow at $Re_\tau = 395$ on the same mesh. The goal of this study was to determine the lowest achievable wall-clock time for one physical time step on currently installed HPC hardware at DLR (Intel Xeon@2.8GHz, Ivy Bridge EP, Infiniband Connect). Due to its more efficient solution algorithm for this incompressible flow, THETA needs more than a magnitude less time to compute one time step on a given domain decomposition (number of grid points per CPU core) than TAU, see Fig. 2 (left).¹ With increasing parallelization, i.e. fewer points per core, both codes initially yield a linear reduction in wall-clock time before gradually departing from the ideal scalability curve. With minimum values of about 2 s for TAU and 0.15 s for THETA at 4800 points/core, TAU slightly reduces the initial (relative) wall-time margin thanks to its somewhat better scalability. This is also reflected in the relative parallel efficiency (reciprocal of wall-clock time \times number of CPU cores) of both codes, as depicted in Fig. 2 (left).

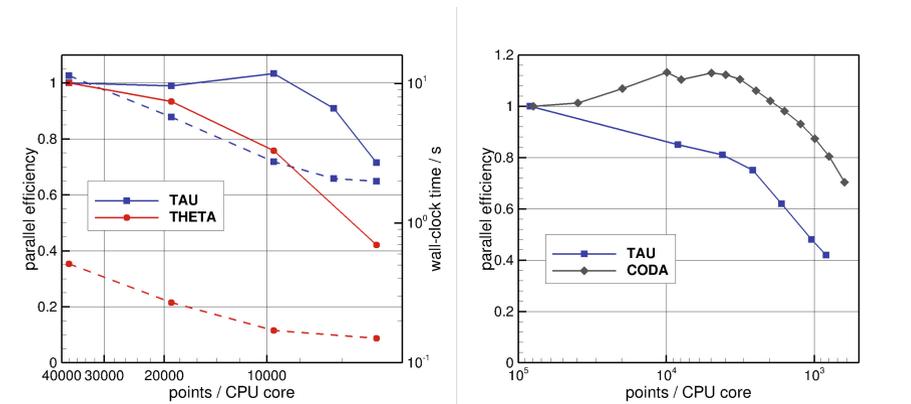


Fig. 2. *Left:* Parallel efficiency (solid lines) and wall-clock time per simulated time step (dashed lines) from LES of a channel flow with DLR-codes TAU and THETA. *Right:* Parallel efficiency of TAU and CODA in a generic (non-LES) benchmark case.

¹ Note on the other hand, that incompressible solvers are not generally applicable to aeronautical flows.

3.2 Potential of New CODA Solver

In 2014, DLR started from scratch a new flexible unstructured CFD solver (“Flucs”) [4], which has by now become the basis for a common CFD capability for/of Onera (the French Aerospace Lab), DLR, and Airbus (“CODA”). Just as TAU/THETA, also CODA uses domain decomposition to make use of distributed-memory parallelism. In contrast to TAU/THETA, however, CODA features overlapping halo-data communication with computation to hide network latencies and, thus, improve scalability. As an alternative to MPI, the GASPI implementation GPI-2 can be used for halo communication. This Partitioned Global Address Space (PGAS) library features highly efficient one-sided communication, minimizing network traffic as well as latencies. Moreover, CODA features sub-domain decomposition, i.e., each domain can again be partitioned into subdomains to make use of shared-memory parallelism. Each subdomain is processed by a dedicated software thread that is mapped one-to-one to a hardware thread, maximizing data locality. In contrast to the 1st level decomposition, this 2nd level (sub)decomposition does not use halo data (but makes use of shared memory). Just as on process level, also for the thread level, the SPMD paradigm is applied, trying to reduce thread synchronization to a minimum. For CODA, running one process per multicore chip has turned out best practice. If a chip features multiple (Non-)Uniform Memory Access ((N)UMA) domains, however, depending on the performance of the cache-coherence logic, running one process per UMA domain may be beneficial. For each process, as many threads (subdomains) are run as hardware threads are available. CODA’s 2-level domain decomposition just described in combination with communication/computation overlap allows for a significantly improved parallel efficiency and scalability, see Fig. 2 (right). Note that a parallel efficiency of more than 100% corresponds to a super-linear speed-up. This effect, in particular observed for CODA in Fig. 2 (right), is due to the distributed simulation fitting into the total L3 cache memory of the HPC-cluster nodes utilized. Also note that a wall-clock time analysis of LES with CODA as in Fig. 2 (left) has not yet been conducted.

4 HPC Requirements of LES for Aircraft - Case Studies

As discussed in Sect. 1, the increasing demand for accurate border-of-envelope simulations has stimulated an interest in LES of full aircraft configurations. To approach the question of feasibility, we first consider two exemplary aircraft components in wind-tunnel conditions and estimate the respective computational effort for LES. Then we discuss possible extrapolations to a full 3D wing of an aircraft in different regimes of the flight envelope (low-/high-speed borders).

4.1 Delta Wing

The nacelle strake is an important component for the aircraft performance at high-lift. We consider a delta wing as a generic generator of a longitudinal vortex with similar properties as for a strake vortex.

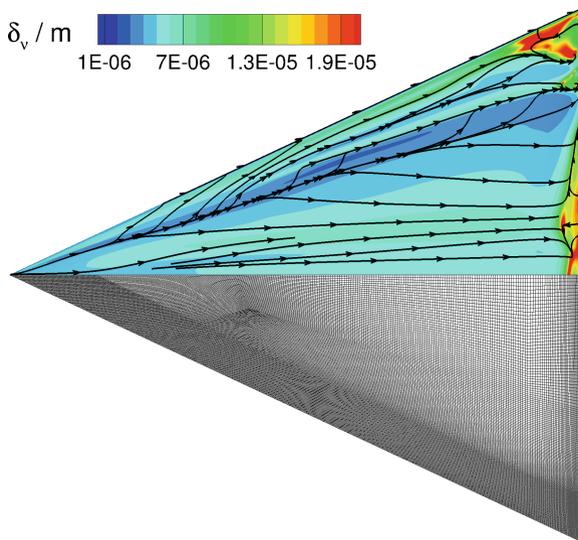


Fig. 3. Top view on the delta wing: Distribution of the viscous length scale and surface streamtraces (top); surface mesh for the RANS computation (bottom).

For a large-eddy simulation, the largest computational costs arise for the resolution of the smallest vortices in the boundary layers of the delta wing. Their size is of the order of the viscous length scale $\delta_\nu = \nu/u_\tau$, which is computed from the kinematic viscosity ν and the local friction velocity u_τ . The surface flow pattern together with the viscous length scale computed with RANS are shown in Fig. 3. The aim is to use an anisotropic surface mesh in the region of attached flow in the inner part of the wing and an isotropic surface mesh in the region of the separated flow in the outer part of the wing. To estimate a spatial resolution sufficient to resolve the near-wall vortices, the grid spacings Δx_i are expressed in wall units, i.e. $\Delta x_i^+ = \Delta x_i/\delta_\nu$. Based on commonly accepted criteria for wall-resolved LES [5] the estimated normalized grid spacings in streamwise direction Δx^+ , wall-normal direction Δy^+ and spanwise direction Δz^+ as well as the associated number of mesh points N_x , N_y , and N_z are given in Table 1. The estimate is based on a chord length $c = 0.3$ m, onflow velocity $U_\infty = 55.5$ m/s and an average viscous length scale of $\delta_\nu = 5 \times 10^{-6}$ m. The total number of mesh points in the boundary layers on both sides of the delta wing is $N_{pnt} = 6.75 \times 10^8$.

For the temporal resolution, we assume a total simulation time of 25 CTUs (convective time units) with $\text{CTU} = c/U_\infty$. For the resolution of the boundary layers we adopt a normalized time step size of $\Delta t^+ = 0.4$ as suggested in [2], leading to $\Delta t = \Delta t^+ \nu u_\tau^{-2} = 1.6 \times 10^{-7}$ s. The number of physical time steps becomes $N_t = 8 \times 10^5$.

To estimate the HPC requirements and wall-clock times for such a simulation with TAU and THETA², we consider the LES of a channel flow from Sect. 3 and assume that the parallel efficiency and wall-clock times per physical time step from Fig. 2 (left) apply to any LES with these flow solvers. For appropriate HPC usage, we further demand $\geq 90\%$ parallel efficiency, which allows distributing 6,500 points/core with TAU, and 20,000 points/core with THETA. The resulting wall-clock times and total core usage for an LES of the delta wing would be 19.3 days on 104,000 cores with TAU, and 2.3 days on 34,000 cores with THETA.

Table 1. Estimated spatial resolution for LES of a delta wing (upper and lower side).

Grid region	Mesh type	Δx^+	Δy^+	Δz^+	N_x	N_y	N_z	N_{pnt} in 10^6
Inner wing	Anisotropic	40	1	20	1500	100	1500	225
Outer wing	Isotropic	20	1	20	3000	100	1500	450

4.2 High-Lift Wing Section

Due to the low speed of an aircraft during take-off and landing, the lift of the wings needs to be increased by deploying slats at the leading edge and flaps at the trailing edge. Simulating the flow around such a wing in high-lift configuration is a crucial but challenging task, since complex interactions of turbulent boundary layers, co-fluent shear layers and flow separation may occur. In this case study, we consider a section of the 3-element DLR-F15 airfoil [12] at wind-tunnel conditions ($Re = 2.1 \cdot 10^6$ and $M = 0.15$), which is inclined with an angle of attack of $\alpha = 6^\circ$.

Requirements of RANS and Hybrid RANS/LES. Present experience on this case comprises TAU simulations using the RANS approach with different turbulence models as well as hybrid RANS/LES with Improved Delayed DES (IDDES) applying wall-modelled LES in the attached boundary layers.

For an unswept wing section (airfoil), the RANS approach can be applied on a two-dimensional mesh in the xy-plane, yielding just about 0.2×10^6 grid points in the given case. Moreover, the simulation can be conducted in steady mode, i.e. omitting temporal resolution of the RANS equations. For illustration, Fig. 4 (top-left) shows the normalized vorticity and the streamlines from the RANS simulation in the vicinity of the deployed flap.

In contrast, the IDDES requires adequate spanwise resolution of the resolved three-dimensional turbulence and a sufficient spanwise domain extent in order not to restrict the resolved structures, visible in Fig. 4 (bottom-left). The largest occurring structures can be estimated by the maximum boundary layer thickness (including displacement due to separation), leading in this case to a minimum span of 8% of airfoil chord in combination with periodic boundary conditions.

² With an onflow Mach number of $M = 0.16$, the flow is mostly incompressible.

The spatial discretization for wall-modelled LES using IDDES requires full wall-normal resolution down to the wall, i.e. $\Delta y^+ \approx 1$, whereas the tangential (stream- and spanwise) spacing scales with the boundary layer thickness. Figure 4 (right) shows the resulting distribution of the wall-tangential (stream- and spanwise) spacings normalized in wall units for a block-structured grid used in [6] on all three airfoil elements. This adequate (yet not highly-resolved) IDDES grid contains $N_{xy} = 0.27 \times 10^6$ grid points in the xy-plane and $N_z = 100$ equidistant spanwise layers in z-direction, yielding $N_{pnt} = 27 \times 10^6$ points in total, cf. Table 2.

The physical time step was chosen as 5000 steps per CTU ($=c/U_\infty$) which yields a mean normalized timestep in wall-units of about $\Delta t^+ = 0.55$, but with maxima up to 1.6 near the leading edges. Due to the rather small-scale flow phenomena (boundary layers with local separations), which bear only limited impact on the global flow field, an overall simulation time of about 10 CTU is considered sufficient, corresponding to $N_t = 5 \times 10^4$ physical time steps.

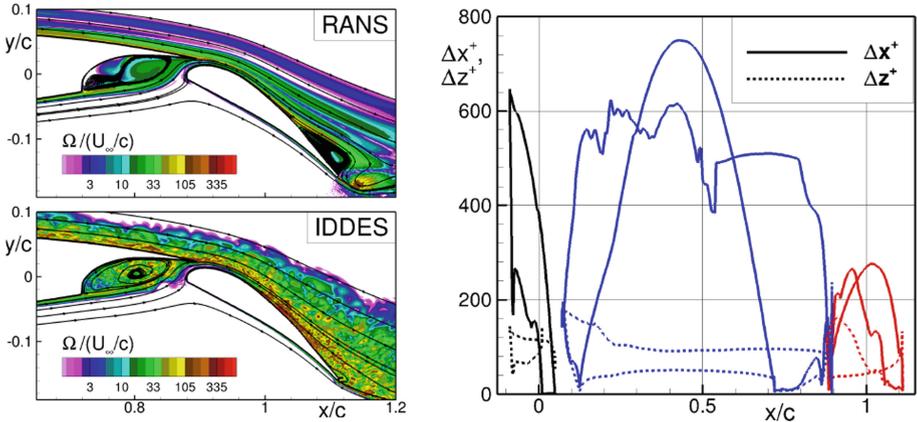


Fig. 4. DLR-F15 airfoil. *Left:* Snapshot of normalized vorticity and time-averaged streamlines around the flap from different modelling approaches. *Right:* Wall-tangential grid spacing in wall-units in the IDDES grid (colors indicate different airfoil elements). (Color figure online)

Extrapolation to LES. For a wall-resolved LES, the domain size and the overall simulation time can be kept as for IDDES. However, the wall-tangential spacing needs to be refined to meet common LES requirements for wall-bounded flow, i.e. $\Delta x^+ \approx 40$ and $\Delta z^+ \approx 20$. Considering Fig. 4 (right), this (roughly) leads to a 10× refinement in stream-, and a 5× refinement in spanwise direction, yielding $N_{pnt} = 1.35 \times 10^9$ total grid points for the same block-structured topology. To ensure a sufficient temporal resolution of $\Delta t^+ \leq 0.4$ in the all flow regions [2], the time step should be divided by 4 compared to IDDES, resulting in $N_t = 2 \times 10^5$ time steps for 10 CTU, cf. Table 2.

With these data and demanding $\geq 90\%$ parallel efficiency according to Sect. 3.1, a total number of around 210,000 cores could be used efficiently with TAU, and the simulation would take 4.9 days. For THETA, with a lower (efficient) core number of around 70,000 the simulation would take 0.6 days (if the local compressibility effects at this Mach number are neglected).

Table 2. Spatial and temporal resolution of different simulation approaches for the DLR-F15 airfoil.

Modelling	Δx^+	Δy^+	Δz^+	N_{xy} in 10^6	N_z	N_{pnt} in 10^6	$\Delta t/(c/U_\infty)$	N_t in 10^3
RANS	" ∞ "	1	" ∞ "	0.2	1	0.2	–	(steady)
HRLM	500	1	100	0.27	100	27	2×10^{-4}	50
LES	40	1	20	2.7	500	1350	5×10^{-5}	200

4.3 3D Wing of Aircraft in Flight

The previous sections show the enormous computational cost associated with wall-resolved LES of the flow around aircraft components at wind-tunnel scale. In this section we extrapolate the computational cost for the main aerodynamic device of a full aircraft in flight conditions, i.e. the wing. Details strongly depend on the operating point of the aircraft, e.g. low-speed take-off and landing or high-speed operation. Note that such a range of flow conditions can only be handled with compressible flow solvers like TAU (or CODA in the future).

First we extrapolate the above estimate for the three-element wing-section of 8% span for the small wind-tunnel Reynolds number of $Re = 2.1 \times 10^6$ to a full wing. For modern transport aircraft, the aspect ratio of wing half span to mean aerodynamic chord is around 4.5 to 5. Using the former, the number of nodes in spanwise direction on a single wing side increases by a factor of 56 compared to the 8%-span wing-section, leading to a total number of 7.6×10^{10} mesh points. Using the same number of time steps ($N_t = 2 \times 10^5$) and cores (210,000) as given in the previous section, the simulation time with TAU can be linearly extrapolated (i.e., assuming perfect weak scaling) to 250 days.

In the second step we include the Reynolds number effects. We assume a take-off speed of 77 m/s and a mean aerodynamic chord length of $c_{\text{MAC}} = 5.8$ m, close to the values for an A350, together with $\nu = 1.5 \times 10^{-5}$ m²/s. The Reynolds number is thus $Re_{c,\text{MAC}} = 30 \times 10^6$. Then the viscous length scale $\delta_\nu = \nu/u_\tau$ is decreasing according to $\delta_{\nu,\text{high}}/\delta_{\nu,\text{low}} \sim \sqrt{c_{f,\text{high}}/c_{f,\text{low}}} Re_{\text{high}}/Re_{\text{low}}$. For the Re-dependence of c_f we use the Coles-Fernholz correlation with $c_{f,\text{low}}/c_{f,\text{high}} \approx 1.5$ and together with $Re_{\text{high}}/Re_{\text{low}} = 15$ we obtain $\delta_{\nu,\text{high}}/\delta_{\nu,\text{low}} \approx 1./12.2$. This leads to an increase of the number of grid points by a factor of $(\delta_{\nu,\text{low}}/\delta_{\nu,\text{high}})^2 = 150$ for $N_x \times N_z$, and we neglect a possible small increase of N_y . This leads to a total number of 1.1×10^{13} grid points for a single wing. Note that this estimate agrees well with the Re-based extrapolation according to [1], see Sect. 2.

For temporal discretization, the condition $\Delta t^+ = 0.4$ [2] implies that the physical time step needs to be decreased by a factor of $(\delta_{\nu,\text{low}}/\delta_{\nu,\text{high}})^2 \approx 150$. Then the number of time steps becomes $N_t = 3 \times 10^7$ and the wall-clock time would rise by a factor of 150^2 compared to the low-Reynolds-number case, yielding more than 15 thousand years on 210,000 cores. But even with exclusive access to the largest existing cluster of Xeon-CPU's comparable to DLR's, i.e. "Tianhe-2A" with almost 5 million cores [13], such a TAU simulation would take around 650 years, when extrapolated linearly. As a final remark, for a full aircraft a simulation time of 10 CTUs may not be sufficient. Following [11], the trailing-vortex system needs to be resolved over $50 c_{\text{MAC}}$ downstream of the wing, increasing the simulation time by another factor of 5.

High-speed buffet occurs at even higher Reynolds numbers. The oscillating shock wave and the involved shock-induced flow separation not only requires highly robust numerical algorithms but extensive LES regions to resolve turbulence for the complete buffet region. For an estimate of the flow conditions representative for the boundary of the flight envelope in the high-speed regime, we assume $Ma_\infty = 0.9$ and that the local flow speed can reach $U_\infty = 400$ m/s in the supersonic flow regions above the wing. On the other hand, at 10 km altitude, the small density of air of around 0.41 kg/m³ leads to larger values of the kinematic viscosity of $\nu \approx 3.5 \times 10^{-5}$ m²/s. Therefore $Re_{c,\text{MAC}} \approx 66 \times 10^6$ is increased by a factor of approximately 2 compared to low-speed high-lift conditions. Therefore the mesh resolution is increased roughly by a factor of 2^2 , and the physical time step needs to be decreased by a factor of 2^2 .

5 Conclusion

Various aspects affecting the feasibility and HPC requirements of high-fidelity flow simulations around aircraft at the border of the flight envelope were discussed. Due to the large resolution requirements of wall-resolved LES at high Reynolds numbers, even the flow around isolated aircraft components at wind-tunnel scale, e.g. narrow wing sections or nacelle strakes, were shown to yield grid-point numbers in the order of 10^9 . Even though the present DLR codes with MPI-based inter-process communication could theoretically make use of up to $\sim 200,000$ CPU cores³ for such problem sizes (assuming perfect weak scaling), the time integration over sufficiently long simulation intervals still leads to wall-clock times in the order of days to weeks.

The subsequent extrapolation to a full 3D wing of an aircraft in take-off conditions (but still at wind-tunnel scale) adds a factor of >50 to either the core number or the wall-clock time, both being infeasible for nowadays industrial use. Finally, with wall-clock times of more than 600 years using one of the largest existing HPC clusters exclusively, the extrapolation to flight Reynolds numbers clearly reveals the inability of present methodologies to resolve all relevant turbulent scales with LES around aircraft flying at the border of the envelope.

³ Consuming more than half of Germany's top-ranked HPC cluster 'SuperMUC-NG' with 305,856 cores [13].

Based on these estimates the following complementary requirements for future turbulence-resolving simulations of aircraft can be formulated:

- Further expansion of available HPC resources, allowing the regular use of >100,000 computing cores in a single flow simulation.
- Modern parallel code designs beyond classical MPI (as the new CODA solver) to further increase parallel efficiency on many-core HPC hardware.
- More efficient and yet robust numerical algorithms to reduce the wall-clock time for one physical time step in a given distributed simulation.
- Finally, despite the appeal of minimized turbulence modelling, the local use of classical RANS in regions with less complex flow physics appears inevitable for industrial use. This leads to hybrid RANS/LES methods with wall modelling in resolved flow regions, possibly supplemented by model-based efficiency improvements like locally-embedded LES regions (using synthetic turbulence injection) or wall functions to bridge the near-wall region.

References

1. Choi, H., Moin, P.: Grid-point requirements for large eddy simulation chapman's estimates revisited. Center for Turbulence Research, Annual Research Briefs 2011, pp. 31–36 (2011)
2. Choi, H., Moin, P.: Effects of the computational time step on numerical solutions of turbulent flow. *J. Comput. Phys.* **113**(1), 1–4 (1994)
3. Illi, S., Lutz, T., Krämer, E.: On the capability of unsteady RANS to predict transonic buffet. In: Third Symposium Simulation of Wing and Nacelle Stall, 21–22 June, Braunschweig, Germany, pp. 1–13 (2012)
4. Leicht, T., et al.: DLR-project digital-X: next generation CFD solver 'Fluc', Deutscher Luft- und Raumfahrtkongress 2016, Deutsche Gesellschaft für Luft- und Raumfahrt (2017). <http://www.dglr.de/publikationen/2017/420027.pdf>
5. Pope, S.B.: *Turbulent Flows*. Cambridge University Press, New York (2000)
6. Probst, A., Löwe, J., Reuss, S., Knopp, T., Kessler, R.: Scale-resolving simulations with a low-dissipation low-dispersion second-order scheme for unstructured flow solvers. *AIAA J.* **54**(10), 2972–2987 (2016)
7. Rumsey, C.L., Slotnick, J.P., Scalfani, A.J.: Overview and Summary of the Third AIAA High Lift Prediction Workshop, AIAA-2018-1258, AIAA Aerospace Sciences Meeting, Kissimmee, Florida, 8–12 January 2018
8. Shur, M.L., Spalart, P.R., Strelets, M.K., Travin, A.K.: A hybrid RANS-LES approach with delayed-DES and wall-modelled LES capabilities. *Int. J. Heat Fluid Flow* **29**(6), 406–417 (2008)
9. Slotnick, J.P., et al.: CFD vision 2030 study: a path to revolutionary computational aerosciences, NASA/CR–2014-218178, March 2014
10. Slotnick, J.P., Heller, G.: Emerging opportunities for predictive CFD for off-design commercial airplane flight characteristics. In: 54th 3AF International Conference on Applied Aerodynamics, Paris, 25–27 March 2019
11. Spalart, P.R., Jou, W.H., Strelets, M., Allmaras, S.R.: Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach. In: Liu, C., Liu, Z. (eds.) *Advances in DNS/LES First AFOSR International Conference on DNS/LES*, Ruston, LA, 4–8 August 1997, Greyden, Columbus, OH, 1997 (1997)

12. Wild, J., Pott-Pollenske, M., Nagel, B.: An integrated design approach for low noise exposing high-lift devices. In: 3rd AIAA Flow Control Conference (2006)
13. TOP500 List: June 2019. <https://www.top500.org/lists/2019/06/>. Accessed 16 Sept 2019