17th World Congress on Computational Mechanics (WCCM)
10th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS)
19 - 24 July 2026, Munich, Germany

CAD-BASED NONCONFORMING MESH ADAPTATION FOR EXTREMELY COARSE INITIAL MESHES

MAXIMILIAN HÖCHEL[*§], THOMAS HAFEMANN[*], MARCO CRISTOFARO[*], MATTHIAS ORLT[†], AND MALTE WEGENER[‡]

[*] German Aerospace Center, Institute for Software Methods for Product Virtualization
Nöthnitzer Str. 46b, 01187 Dresden, Germany

[§] RWTH Aachen University, Schinkelstr. 2, 52062 Aachen, Germany
e-mail: maximilian.hoechel@dlr.de

[†] German Aerospace Center, Institute for Aerodynamics and Flow Technology
Bunsenstr. 10, 37073 Göttingen, Germany

[‡] German Aerospace Center, Institute for Aerodynamics and Flow Technology
Lilienthalplatz 7, 38108 Braunschweig, Germany

Key words: `FlowSimulator`, CFD, Adaptive Mesh Refinement, CAD, Mesh Deformation

Summary. *High-fidelity simulations are a key tool to improve the aerodynamic design of aircraft, but the required target accuracy often demands very fine meshes, especially for high-Reynolds-number configurations. Nonconforming hierarchical mesh refinement based on cell subdivision is a powerful means to reduce this cost, as it refines only where the solution requires it and works natively with the linear hybrid meshes commonly used in industry. To be maximally efficient, it should start from an extremely coarse initial mesh. However, when linear meshes are refined, the curved surface of the design is lost, so the mesh has to be kept faithful to the original CAD geometry. To this end, a CAD-driven mesh adaptation workflow is proposed: The next-generation, highly parallel CFD solver computes the flow field, and cells are refined locally, driven by the solution. Inside the boundary layer, refinement is performed on entire cell stacks, preserving the anisotropic layering and keeping the subsequent volume mesh deformation well-posed. The solution and the mesh-to-CAD mapping are transferred from the previous mesh efficiently, exploiting the hierarchical nature of the refinement. New surface node positions are computed on the CAD geometry by a cell-stack-dependent directional projection and supplied to a multi-step volume mesh deformation, which propagates them into the mesh interior while preserving the nonconformity. All steps are performed in memory inside the `FlowSimulator` framework, enabling their use in high-performance computing environments. The workflow is demonstrated for the transonic ONERA M6 wing test case, starting from an extremely coarse initial mesh of only 2112 cells. Without geometry integration, the refinement converges to a solution that is inconsistent with the actual design, whereas the proposed workflow yields geometry-consistent aerodynamic results. Compared to global refinement, for a given accuracy, the adaptive approach currently offers only a modest gain in cell count, since the stack refinement introduces a significant overhead, which future work aims to reduce.*

1 INTRODUCTION

Aerospace vehicles shall operate as cost-effective as possible, which saves fuel and thus reduces exhaust gas. To improve the design of its geometries accordingly, high-fidelity simulations are employed by the corresponding industry. Perspectively, these could even be used for a certification by analysis. But they demand high accuracy, which often necessitates a high resolution for the underlying numerical discretization. This effort can be reduced through adaptive mesh refinement (AMR): It reduces the required number of degrees of freedom needed for a given accuracy, avoids the human effort of repeatedly using the mesh generator in search of a suitable mesh, and also makes a mesh-independence study automatized.

The specific type of AMR investigated here is the nonconforming hierarchical one based on cell subdivision. In contrast to the often used metric-based refinement of purely tetrahedral meshes [1], it works natively for meshes with mixed element types, is algorithmically more efficient, more scalable [2], and reversible to perform rapid coarsening. However, it is rarely used for industrially relevant aeronautic flow simulations, where the very recent paper [3] is an exception. This is partly caused by the complexity of integrating curved wall shapes into it: For industrially relevant finite-volume computations, the meshes used are commonly linear. When refining linear meshes, the curved geometry has to be recovered. Otherwise, artificial corners and edges are modeled, converging to a different solution under mesh refinement (see Figure 1 for a simplified example). Spurious effects occur, as in [4], and the refinement is

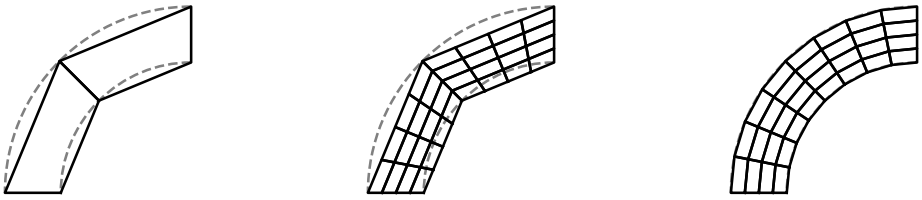


Figure 1: A mesh for an annulus quarter geometry, originally with two cells (left), is refined twice, either without (center) or with respecting the geometry (right).

restricted to reasonably sized initial meshes or to only three or four refinement levels. And even if the numerical artifacts are reduced by smoothing the surface mesh during refinement [5], the modeled geometry will still diverge from the actual design geometry, given by its Computer-Aided Design (CAD) description.

As a first step to include the CAD geometry, new positions on it must be computed. In [6], the authors interpolate the positions in the surface parametric space. This is only possible for simple geometries, as a surface cell must be covered entirely by a single CAD patch, which complicates the use of extremely coarse initial meshes. In addition, to get a smooth surface mesh, the patch should be uniformly parameterized, which is not the case for *non*-uniform rational B-splines (NURBS). As an alternative, in [3, 7], node positions are determined by projecting orthogonal to the surface mesh, based on a linear interpolation on the unrefined mesh. For that

projection, the intersection of the given direction with the geometry can be computed based on a tessellation of the geometry, that is, a very fine triangle-mesh approximation of it [8]. However, projecting onto a tessellation yields only an approximation of the true geometry, and requires a suitable resolution. A more direct solution is to use the mathematical descriptions of the patches of the CAD geometry themselves [3, 7]. In this way, since some mesh generators simplify the input CAD geometry as a preliminary step [9], the initial mesh can also be harmonized with the used CAD geometry.

As the second step to include the CAD geometry, when the positions of the new surface nodes are known, the displacements have to be propagated into the interior of the volume mesh. The approach of [6] is to adapt the coordinate mapping of the first volume cell at the mesh boundary, which must decay the entire deformation inside it. For this, the initial mesh must be very coarse in wall-normal direction so that the new node position is covered by the first volume cell, which rules out anisotropic cell layers in most cases. An alternative is to locally modify the coordinates of the nodes of multiple cells in the vicinity of the boundary [5]. However, this approach requires certain structures for these cells, and it cannot deal with nonconforming meshes. The local modification of some cells can, however, be replaced by a general-purpose volume mesh deformation that takes the surface displacements as input. This approach was adopted in reference [3] (in a file-based manner and with a non-parallelized part) and is the closest procedure to the one presented in this paper. However, the initial meshes were not particularly coarse, as only three refinements had to be carried out until mesh convergence. But the coarser the initial mesh, the more freedom the refinement indicator has to concentrate the refinement where it is needed.

To maintain the robustness of the flow solver and involved mesh deformation, this paper introduces two novelties: the refinement of cell stacks, and the projection with a mesh-dependent direction. With this, nonconforming AMR is applied to extremely coarse initial meshes for curved geometries. Thus, automatic adaptivity can be fully exploited in the sense that, in the converged adapted grid, every initial cell has been refined at least once. This ensures the final grid achieves the maximum coarseness everywhere, because the initial mesh determines the minimum resolution. At the same time, the proposed approach is efficient, as it (A), maintains the mesh nonconformity, saving degrees of freedom, thanks to the capabilities of the used flow solver, and (B), leverages the benefits of the hierarchical nature of the mesh refinement for the mesh-to-CAD mapping. Finally, the workflow is demonstrated to be robust and to exhibit the high parallel efficiency necessary for industrial applicability.

2 METHOD

The workflow proceeds iteratively until a given convergence criterion is met: The CFD solution is computed, AMR splits cells (creating new nodes at the mesh boundary), the updated surface node positions computed from the geometry serve as input to the volume mesh deformation, which propagates the surface deformation into the interior of the mesh.

As hierarchical refinement creates nonconforming meshes, some parts of this algorithm require modifications and additional steps.

2.1 Mesh Refinement with Cell Stacks

Aerospace applications typically involve high Reynolds numbers and thin boundary layers at the aircraft surfaces. To resolve the most important flow features, the commonly created type of CFD mesh is hybrid: At the walls, very thin prismatic layers of cells are stacked with increasing height toward the farfield, and the rest of the domain is filled with tetrahedra. When the wall is curved, the required node projection can span multiple tens of cell layers of the initial mesh (see Figure 2). For a conforming mesh, the subsequent deformation can move

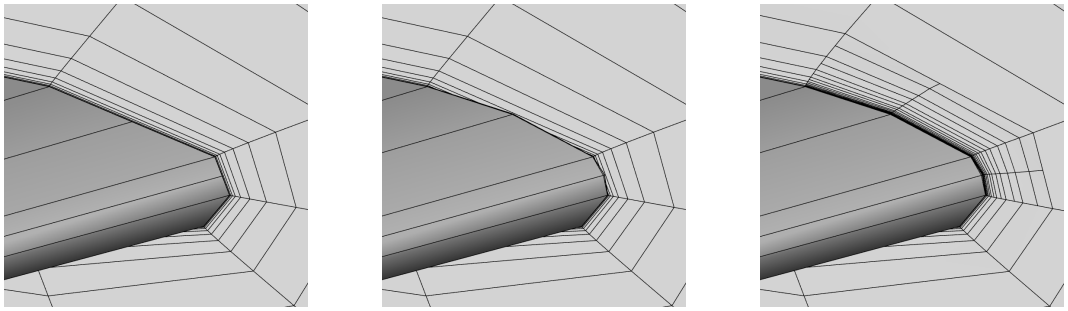


Figure 2: If the first cell at the wall is refined (left), a projection onto the geometry would move the new surface node across multiple tens of cell layers (center). This is handled by refining further cells (right).

the node to the correct position, as the displacement decays quickly in the local vicinity. For a nonconforming mesh, the volume mesh deformation is constrained by the hanging nodes: A hanging node must lie on the respective large cell, which cannot be bent or curved. This introduces internal rigidity into the volume mesh, propagating the deformation farther away, in particular for strongly anisotropic cells. If the first volume cell at the boundary is refined and has a hanging node at its top, the deformation will affect many cells that were not involved in the refinement in the first place. The resulting mesh will be strongly distorted if a solution to this numerically stiff deformation problem can be found at all. As a side effect, the spatial CFD discretization is affected, which should be avoided; this enables the refinement indicator to be as effective as possible in changing the mesh.

To address this, this paper proposes to refine multiple cells as a stack beginning at the wall, if the bottom cell is refined during the standard refinement procedure. The height of the refined cell stack is determined from the surface node displacement: Cells are collected upward until the top cell has a height in the projection direction that exceeds the displacement magnitude by a given factor (set to 4.0 for this paper). In this way, the hanging node at the top of the stack is moved away from the curved surface, and the neighboring cells are no longer strongly affected by the deformation.

As a first step, in this work, the refinement of a cell stack is triggered by the refinement of *any* of its cells, not just its bottom one. Hence, per stack, all cells have the same refinement level, which simplifies the implementation. In addition, cell stacks are preserved continuously during refinement in favor of the used flow solver, which uses line-implicit methods.

2.2 Surface Node Projection Direction

The hanging node poses further problems if the cell stacks are strongly curved or tilted, which often happens even for only moderately complex geometries. A tilted or curved cell stack might have to be straightened, strongly deforming the surrounding cells, as no other displaced configuration can satisfy the required node projection position without producing inverted cells. Note that the issue vanishes if conforming meshes are produced or if only global refinement is applied. The issue is amplified by the fact that there are hanging nodes on the surface, which must not be projected but must be constrained to follow the surrounding deformation, to produce a watertight nonconforming surface mesh. To balance not only the surface cell sizes but also the volume cell sizes in the stack, this work proposes to prescribe the projection direction based on the volume mesh, see Figure 3. If the cells are stacked at the wall, the path of the stack can be identified, and the point where it crosses the CAD geometry can be found. In practice, this point is determined by prescribing a projection direction accordingly (dashed arrow in Figure 3). This approach violates the goal of equally sized surface cells and introduces an adaptation path

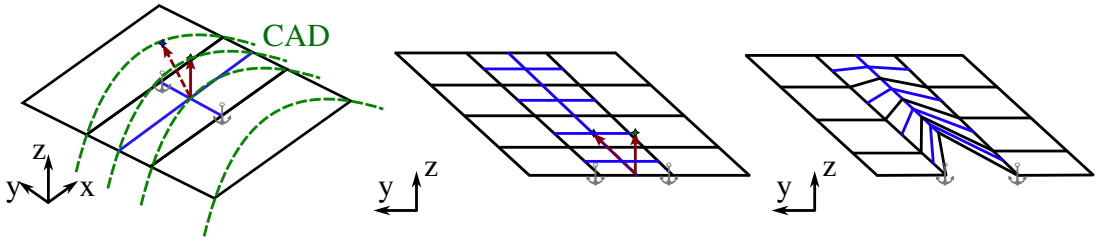
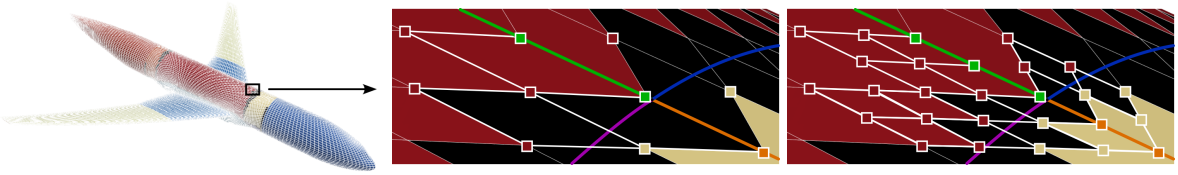


Figure 3: Perspectives of a surface mesh at a curved geometry (green), with cells tilted in y-direction. After refinement (blue), the new free node (no anchor symbol) is *not* projected orthogonally (solid red arrow), but in stack direction (dashed arrow). The final mesh is shown on the right.

dependency for the resulting mesh node coordinates. However, the refinement indicator, which guides the AMR workflow, will adapt to it over the refinement iterations.

2.3 Mesh-to-CAD Mapping Strategy

In general, the mapping from mesh nodes to the trimmed CAD geometry patches is found by searching for the nearest projection, shown exemplarily in Figure 4a. For the later extension



(a) Zoom location **(b)** Zoomed view **(c)** Zoomed view with refined cells

Figure 4: Initial and updated mesh-to-CAD mappings for the High-speed NASA Common Research Model, where mesh cell/node colors denote their CAD associations, and black denotes "unmappable".

of the mapping during hierarchical refinement, mesh edges and faces are also mapped to CAD curves or surfaces. A mesh edge or face is mapped to a CAD curve or surface if all edge/face nodes were mapped to it or its bounding curves/vertices. Sometimes, the mesh does not fit the CAD geometry completely, as mesh generators perform some CAD post-processing; e.g., they omit some CAD curves for the mesh generation [9]. In this case, a mesh edge/face is unmappable (black in Figure 4b), which is acceptable and only affects the hierarchical extension of the mapping, not the node projection. As a preprocessing step of the AMR workflow, if the geometry and the mesh stem from the different sources, the mesh is deformed once onto the CAD geometry so that the nodes fit it perfectly. Such a situation can arise if the exact, potentially manipulated, CAD geometry used for generating the grid is no longer available.

After adaptive mesh refinement, the mapping of the new mesh nodes is determined as follows (see Figure 4c): If the new mesh node is located on a parent mesh edge/face, it will inherit its CAD curve/surface association. As some mesh edges and faces have no mapping assigned, some nodes will still lack a CAD patch association. For these, the nearest CAD patch has to be searched for, using a consistently prescribed projection direction. When the CAD association is found for the new nodes, the mapping for the new mesh edges/faces is derived in the same way as for the initial mapping before any refinement.

2.4 Simulation Framework

The proposed geometry-adapted nonconforming AMR workflow is implemented within the `FlowSimulator` [10], see Figure 5. It runs completely in memory, with all parts paral-

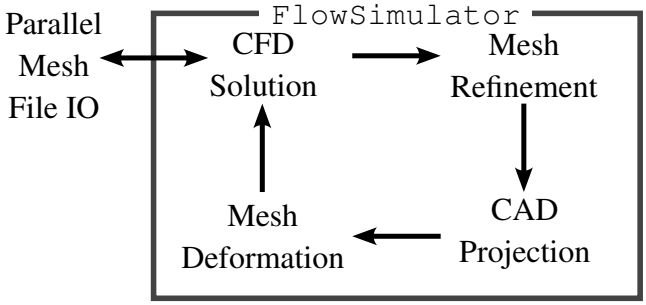


Figure 5: The investigated `FlowSimulator` workflow for geometry-adapted mesh refinement.

lelized using distributed memory, and time-consuming parts with additional shared memory parallelization.

For the CFD simulation, the next-generation "CFD Software by ONERA, DLR and Airbus" (`CODA`) [11, 12] is used[1]. The flow problem is spatially discretized by a finite volume method, solving the Reynolds-Averaged Navier-Stokes equations with the Spalart-Allmaras tur-

[1] `CODA` is the computational fluid dynamics (CFD) software being developed as part of a collaboration between the French Aerospace Lab ONERA, the German Aerospace Center (DLR), Airbus, and their European research partners. `CODA` is jointly owned by ONERA, DLR and Airbus.

bulence model in the negative formulation [13], using a second-order Roe-upwind convection scheme. For the steady-state problem, a multi-stage Linearized implicit Euler method with local time-stepping is employed, using, as the linear solver, a restarted GMRES with a Jacobi preconditioner using a tri-diagonal line implicit diagonal, from the corresponding `Spliss` [14] library.

The cell subdivision is implemented in `FSAdaptationNG` [15]. To fit the environment of the `FlowSimulator`, it uses the standard explicit mesh storage instead of space-filling curves (as in [6], for example): The mesh is stored as a list of node coordinates and a list of cells, each as a list of node IDs, so that node coordinates can be modified individually without additional indirections, as required by the mesh deformation. To describe hanging node relations in the nonconforming mesh, `FSAdaptationNG` inserts so-called pseudo cells: A mesh-consistent description of a hanging node connection that can be thought of as a flat volume cell. The faces of the pseudo cell are connected to the large and small neighbor volume cells, respectively.

The CAD projection is performed with `FSOCCT` [16], which is based on and extends the OpenCascade technology (OCCT) CAD kernel [17].

The volume mesh deformation is performed with `FSMeshDeformation` [18], which uses elastic analogy for the deformation. To use it for nonconforming distributed meshes, a multi-step approach is applied: In the first deformation, pseudo cell connections are ignored: The boundary displacements are satisfied, but cells in the mesh interior intersect, and hanging nodes are pushed away from the faces/edges they belong to. Next, the coordinates of these hanging nodes are overwritten with the (bi-)linear interpolation values of the edges/faces. As a final step, while holding all nodes of the pseudo cells in their updated positions, the previous deformation is repeated from the otherwise undeformed state, yielding the final watertight mesh.

3 RESULTS

The AMR is applied to the ONERA M6 free tapered swept wing test case [19], which is a relatively simple three-dimensional geometry. With a Mach number of $\mathrm{Ma} = 0.84$, transonic shocks occur, and with a Reynolds number of $\mathrm{Re} = 14.6 \cdot 10^6$, strongly anisotropic cells are required in the boundary layer. The initial grids are hybrid; they were created based on reference [20] with the Fortran tool available on the NASA Turbulence Modeling Resource webpage, using the recommended input files[2]. The CAD geometry was downloaded from a further NASA Turbulence Modeling Resource webpage[3].

The AMR is driven by a scaled version of the following general-purpose residual-based refinement indicator [21, 22]:

$$\eta_\kappa^{\mathrm{res}} = h_\kappa \left\| \mathbf{R}_\kappa \left(\mathbf{u}_h \right) \right\|_\kappa + h_\kappa^{1/2} \left\| \mathbf{r}_{\partial\kappa} \left(\mathbf{u}_h \right) \right\|_{\partial\kappa}$$

Here, κ denotes the element, h the element size, $\left\| \mathbf{R}_\kappa \left(\mathbf{u}_h \right) \right\|_\kappa$ the strong residual of the partial

[2] https://www.nasa.gov/wp-content/uploads/2026/04/wing-release-072319.zip, retrieved on September 3rd, 2026

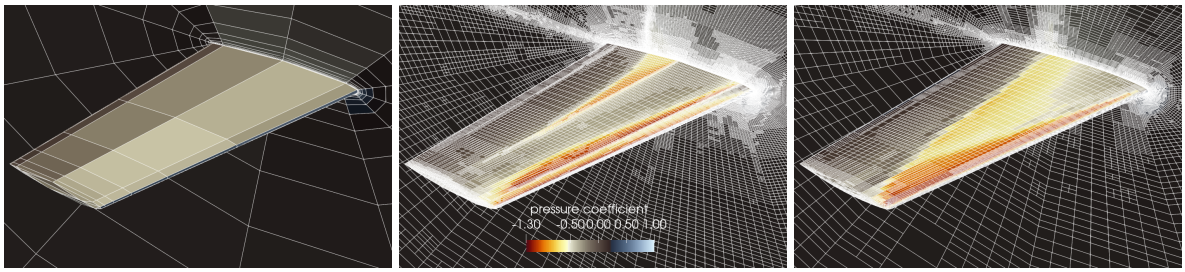
[3] https://tmbwg.github.io/turbmodels/Onerawingnumerics_grids/AileM6_with_sharp_TE.igs, retrieved on September 3rd, 2026

differential equation (PDE) on the element κ, and $\|\mathbf{r}_{\partial\kappa}(\mathbf{u}_h)\|_{\partial\kappa}$ the face residual on the boundary of the element κ. The definitions of the residuals are devised in [21, 22]. The indicator can be computed cheaply from the CFD solution and is an indication of the local discretization error. The indicator is used to guide refinement by selecting the cells with the highest values until the cell count after refinement roughly doubles. However, starting at a distance to the wing surface of eight root chord lengths, the indicator is scaled down exponentially, reaching 1% at the farfield boundary. By this, the refinement is gently preferred in the wing vicinity, as the complete wing wake would consume a lot of the refinement otherwise.

Once the refined mesh is created, the solution is transferred to it cheaply by copying the variable values from a parent cell to its children, serving as initial guess for the iterative flow solution.

Since the cells to refine are usually clustered, the adaptation has an inherent load imbalance, so the AMR overhead must be considered for the overall performance. For the considered cases, it is approximately 5 %, counting everything except `CODA`.

The AMR workflow is started from the extremely coarse initial mesh level 6 of the grid family (2112 cells), with only seven chord-wise and eight span-wise nodes, and a non-dimensional first cell height of $y^+ \approx 42$ on the upper wing surface center. For comparison, the level 5 mesh is used as well, which corresponds to one global refinement of level 6. The AMR workflow was run six times, see Figure 6 for the final meshes with and without geometry integration. It



(a) Initial grid **(b)** AMR without geometry **(c)** AMR with geometry

Figure 6: The pressure coefficient distributions for the initial and adaptively refined mesh level 6.

can be seen that neglecting the geometry leads to completely different shock positions, showing multiple separate shocks, starting at the initial mesh edges.

The convergence of aerodynamic forces is presented in Figure 7, where the results are compared to the results obtained with the NASA grid family. In addition to the common lift and drag coefficients, the y-moment coefficient is added to give a complete picture of the strong influence of the shock positions. Mesh convergence can not be achieved with only six refinements, but was already impossible to achieve with the NASA grid family, despite the considerable compute resources required for the finest mesh with almost 70 million cells (confirming [23]). Nevertheless, it is evident that neglecting the geometry influence makes the computation converge to forces inconsistent with the actual geometry. Also, it leads to a non-monotonic behavior, questioning the assessment of the convergence of geometry-free AMR simulations in general. But

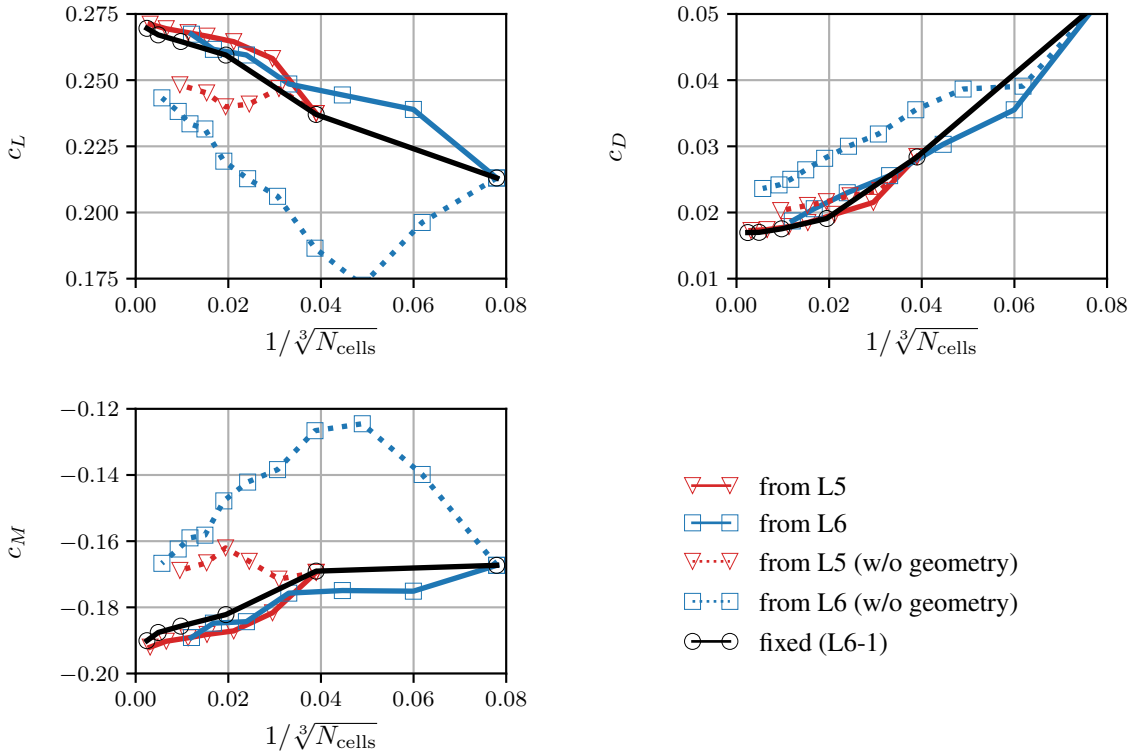


Figure 7: Lift c_L, drag c_D, and moment coefficients c_M for different cell counts N_{cells} for the ONERA M6 test case, comparing geometry-free and geometry-based approaches with global refinement.

surprisingly, even when including the geometry, the AMR performance in terms of the number of cells is only slightly better than global refinement. This shows a side-effect of the stack-refinement: The median of the strongly varying overhead of the stack-oriented refinement can be quantified as around 55% of the cells, both for starting from the level 5 or the level 6 meshes. Thus, the indicator looses significant freedom in driving the adaptation process efficiently, challenging the decision to trigger stack refinement by any of its cells, rather than just its bottom one. Furthermore, starting from finer level 5 yields better results than starting from the coarser level 6. It is unclear if this is due to the level 6 mesh being too coarse to compute a refinement indicator meaningful for the final solution, or a worse quality of the adapted level 6 mesh after repeated deformation.

4 SUMMARY AND OUTLOOK

The workflow for integrating the geometry in an adaptive mesh refinement includes the flow solver, cell subdivision, design geometry node projection, and volume mesh deformation. It is extended to work with maximally coarse initial meshes using nonconforming refinement. To this end, the mesh refinement is extended to preserve cell stacks, up to a height dependent on the surface node displacement. Furthermore, the new nodes are projected onto the geome-

try directionally, considering the cell-stack structure. The mapping between mesh entities and computational geometry, found by search and stored as metadata in the mesh object, is updated quickly during refinement by exploiting the hierarchical nature of the adaptive mesh refinement. The algorithms are implemented within the `FlowSimulator` environment, combining next-generation "CFD Software by ONERA, DLR and Airbus" (`CODA`), `FSAdaptationNG`, `FSOCCT`, and `FSMeshDeformation`, the latter applied in a multi-step approach to enable the deformation of nonconforming meshes.

The proposed methodology is applied successfully to the transonic flow over a free tapered swept wing, enabling a start from an extremely coarse initial mesh and yielding geometry-consistent results. The improvement over the global refinement is limited, caused by the overhead of the stack refinement. Furthermore, starting the from the coarsest of the tested meshes did not improve the AMR performance compared to the next finer mesh of the grid family. Potential reasons for this are being investigated.

More work is required to make the workflow applicable to more complex geometries. This is because even meshes optimized for anisotropic boundary layers might not contain cell-stacks everywhere, e.g., at concave surface intersections, depending on the mesh generator. In these cases, a generalization of the refinement of stacks to arbitrarily connected cells is required, which may simultaneously eliminate the need for the directional projection. In addition, especially for the (potentially generalized) stack refinement, considering anisotropic refinement might save additional degrees of freedom. For instance, when refining a cell stack, a horizontal split would be enough to move the hanging node further away from the curved geometry, toward the top of the stack. Finally, this improved AMR workflow should be tested for flows around actual industrial, more complex geometries.

Acknowledgments

Simulations were carried out on the CARO HPC cluster. The authors gratefully acknowledge the scientific support and HPC resources provided by the German Aerospace Center (DLR). The HPC system CARO is partially funded by “Ministry of Science and Culture of Lower Saxony” and “Federal Ministry for Economic Affairs and Climate Action”.

REFERENCES

[1] Frederic Alauzet et al. 5th AIAA CFD High Lift Prediction Workshop Results Using Metric-Based Anisotropic Mesh Adaptation. In *AIAA SCITECH Forum*, 2025. doi:10.2514/6.2025-0497.

[2] Johannes Holke et al. t8code - Modular Adaptive Mesh Refinement in the Exascale Era. In *SIAM International Meshing Round Table*, 2023. doi:10.21105/joss.06887.

[3] Dimitri J. Mavriplis, Zhi Yang, and Andrew C. Kirby. Dynamic Adaptive Mesh Refinement for WMLES on Complex Configurations. In *AIAA SCITECH Forum*, 2026. doi:10.2514/6.2026-0703.

[4] Arjun J. Vedam and William A. Engblom. Exploring a Novel Adaptive Mesh Refinement Strategy for Transonic Flows. *International Journal for Numerical Methods in Engineering*, 2025. doi:10.1002/nme.70226.

[5] Thomas Alrutz. Hybrid Grid Adaptation in TAU. In Norbert Kroll and Jens K. Fassbender, editors, *MEGAFLOW - Numerical Flow Simulation for Aircraft Design*, 2005. doi:10.1007/3-540-32382-1_7.

[6] Sandro Elsweijer, Johannes Holke, Jan Kleinert, and Dirk Reith. Constructing a Volume Geometry Map for Hexahedra with Curved Boundary Geometries. In *SIAM International Meshing Roundtable Workshop*, 2022. URL https://elib.dlr.de/186570/.

[7] Luca Heltai, Wolfgang Bangerth, Martin Kronbichler, and Andrea Mola. Propagating Geometry Information to Finite Element Computations. *ACM Trans. Math. Softw.*, 47(4), 2021. doi:10.1145/3468428.

[8] Anton Schotte, Jean-Baptiste Chapelier, Francesca Basile, Romain Laraufie, and Sébastien Pernet. An anisotropic unstructured mesh adaptation strategy for Discontinuous Galerkin methods with application to compressible turbulent flow problems. *Computers & Fluids*, 315, 2026. doi:10.1016/j.compfluid.2026.107138.

[9] Atul Thakur, Ashis Gopal Banerjee, and Satyandra K. Gupta. A survey of CAD model simplification techniques for physics-based simulation applications. *Computer-Aided Design*, 41(2):65–80, 2009. doi:10.1016/j.cad.2008.11.009.

[10] Lars Reimer, Ralf Heinrich, Markus Ritter, Andreas Krumbein, Sven Geisbauer, Stefan Goertz, and Tobias Leicht. Virtual Aircraft Technology Integration Platform: Ingredients for Multidisciplinary Simulation and Virtual Flight Testing. In *AIAA Scitech Forum*, 2021. doi:10.2514/6.2021-1202.

[11] Tobias Leicht, Jens Jägersküpper, Daniel Vollmer, Axel Schwöppe, Ralf Hartmann, Jens Fiedler, and Tobias Schlauch. DLR-Project Digital-X – Next Generation CFD Solver 'Flucs'. *DLR project report*, 2016. URL https://www.dglr.de/publikationen/2017/420027.pdf.

[12] Pedro Stefanin Volpiani, Jean-Baptiste Chapelier, Axel Schwöppe, Jens Jägersküpper, and Steeve Champagneux. Aircraft Simulations Using the New CFD Software from ONERA, DLR, and Airbus. *Journal of Aircraft*, 61(3):857–869, 2024. doi:10.2514/1.C037506.

[13] Steven Allmaras, Forrester Johnson, and Philippe Spalart. Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model. In *ICCFD7*, pages 1–11, 2012. URL https://www.iccfd.org/iccfd7/assets/pdf/papers/ICCFD7-1902_paper.pdf.

[14] Olaf Krzikalla, Arne Rempke, Alexander Bleh, Michael Wagner, and Thomas Gerhold. Spliss: A Sparse Linear System Solver for Transparent Integration of Emerging HPC Technologies into CFD Solvers and Applications. In *New Results in Numerical and Experimental Fluid Mechanics XIII*, pages 635–645, 2021. doi:10.1007/978-3-030-79561-0_60.

[15] Maximilian Höchel, Thomas Hafemann, Matthias Orlt, and Roberto Iker Sanchez Ortiz. Algorithms for Geometry-Adapted Mesh Refinement with Cell Stack Preservation in the FlowSimulator. Conference for Adaptive Mesh Refinement and Applications, September 2025. URL https://elib.dlr.de/216488/.

[16] Thomas Hafemann, Mladen Banovic, Adam Büchner, Simon Ehrmanntraut, Constantin Höing, Sebastian Gottfried, and Arthur Stück. A CAD-enabled MDAO Framework Approach for Gradient-based Aerodynamic Shape Optimization. In *9th ECCOMAS Congress*, October 2024. doi:10.23967/eccomas.2024.199.

[17] Open Cascade SAS. Open CASCADE technology. https://github.com/Open-Cascade-SAS/OCCT, 2026. Version 8.0.0.

[18] Arne Rempke. Deformation of CFD Meshes with Anisotropic Cells in a Viscous Boundary Layer Using Line-Implicit Methods. In *23rd STAB/DGLR Symposium on New Results in Numerical and Experimental Fluid Mechanics XIV*, volume 154, pages 273–283, 2023. doi:10.1007/978-3-031-40482-5_26.

[19] V. Schmitt and Françoise Charpin. Pressure distributions on the ONERA M6 wing at transonic Mach numbers. In *Report of the Fluid Dynamics Panel Working Group 04*, 1979.

[20] Hiroaki Nishikawa and Boris Diskin. Customized Grid Generation Codes for Benchmark Three-Dimensional Flows. In *AIAA Aerospace Sciences Meeting*, 2018. doi:10.2514/6.2018-1101.

[21] Tobias Leicht and Ralf Hartmann. Error estimation and anisotropic mesh refinement for 3D laminar aerodynamic flow simulations. *Journal of Computational Physics*, 229(19): 7344–7360, 2010. doi:10.1016/j.jcp.2010.06.019.

[22] M. Wegener. *Adaptive Methods for Flexible Basis Discontinuous Galerkin Methods and Finite Volume Methods*. PhD thesis, Technical University of Braunschweig, 2026. Dissertation (in preparation).

[23] Boris Diskin et al. Grid Convergence for Three Dimensional Benchmark Turbulent Flows. In *AIAA SciTech Forum*, 2018. doi:10.2514/6.2018-1102.