

ANISOTROPIC MESH ADAPTATION PIPELINE FOR THE 3D LAMINAR FLOW OVER A DELTA WING

Christos Tsolakis and Nikos P. Chrisochoides

CRTC Lab Computer Science Department
School of Sciences
Old Dominion University
{ctsolakis,nikos}@cs.odu.edu

ABSTRACT

In this work, a mesh adaptation tool is loosely coupled with a metric based error estimator and a CFD solver into a computational pipeline. The pipeline is validated using a laminar flow over a delta wing. Drag and lift coefficients are computed and compared to similar simulations from the literature. The proposed anisotropic adaptation method delivers the same accuracy with an order of magnitude fewer resources than the more commonly used isotropic mesh adaptation.

Keywords: Anisotropic Mesh Generation, Metric Adaptation, Mesh Adaptation

1 INTRODUCTION

Computational Fluid Dynamics (CFD) is concerned with evaluating characteristics and quantities related to the flow of a fluid around a body. The computational aspect of the method is composed mainly of three parts. First, is the Geometry Definition and Geometry Discretization, Solution of the underline equations and finally visualization. While not new, the requirements of this pipeline have undergone significant changes following the increasing demands of industry and academia for better solution resolution, see (Alauzet and Loseille 2016). As a consequence, any new tool needs to be thoroughly tested and verified against previous methods and datasets. In this work, a pipeline consisting of a mesh adaptation software a suite of error estimator and interpolation tools, as well as a CFD solver, are brought together, and the configuration is tested against a well-documented case of a 3D laminar flow over a delta wing.

In this study, mesh adaptation is built upon previous work (Tsolakis, Drakopoulos, and Chrisochoides 2018). More specifically, metric-based adaptation is employed. The crux of this approach is to create an anisotropic mesh by mapping distance and quality evaluation to the metric space. This mapping \mathcal{M} is given by a 3×3 positive definite matrix called metric tensor and it can be shown that it induces an inner product $\langle u, v \rangle_{\mathcal{M}} := u^T \mathcal{M} v$. Based on the inner product the familiar formulas for distance, angle and volume evaluation can be rewritten so that they take into account the information encapsulated in \mathcal{M} . Equipped with these size evaluation tools, the mesh adaptation algorithm will attempt to create elements that optimize the spacing and the quality in the metric space. A high-level description of the general adaptation procedure is given in Figure 1.

Section 4 demonstrates the effectiveness of this approach where quantitative characteristics like the lift and drag coefficients of the flow are evaluated with lower error using an order of magnitude fewer vertices and

consequently an order of magnitude less computational resources than the more commonly used isotropic mesh adaptation.

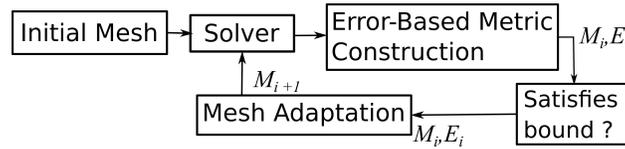


Figure 1: Mesh Adaptation pipeline : At each iteration, the solver provides feedback to the mesh generation module, about the areas of the mesh that need refinement and or coarsening.

2 METHODS

In this work, the mesh is adapted using CDT3D (Drakopoulos 2017) a multi-threaded mesh generation software based on advancing front point creation, direct point insertion and a speculative scheme for local reconnection (Drakopoulos, Tsolakis, and Chrisochoides 2017). SU2 (Economon, Palacios, Copeland, Lukaczyk, and Alonso 2016) is used as a CFD solver, it was chosen due to its ease-of-use and open availability. Finally, the `refine` suite of mesh mechanics (Park 2018) is used for evaluating the multiscale metric and interpolating the solution at each iteration to the new, adapted mesh.

2.1 Mesh Adaptation

The mesh adaptation software used in this work is CDT3D (Drakopoulos 2017, Drakopoulos, Tsolakis, and Chrisochoides 2017). CDT3D is designed to be the speculative component of the telescopic approach to mesh generation presented in (Chrisochoides 2016). As part of meeting the expectations of the CFD 2030 study (Slotnick, Khodadoust, Alonso, Darmofal, Gropp, Lurie, and Mavriplis 2014) which frames the future needs of the simulation and meshing community, metric adaptation was added (Tsolakis, Drakopoulos, and Chrisochoides 2018) and evaluated (Tsolakis, Chrisochoides, Park, Loseille, and Michal 2019) next to well-know and tested mesh adaptation methods.

The pipeline of CDT3D adaptation for this work is shown in Figure 2 can be divided into four main steps: (a) boundary adaptation which is performed using the mesh library MMGS (Dobrzynski and Frey 2009), (b) initial mesh construction which is starting from a surface mesh builds a tetrahedral mesh that conforms to the boundary, (c) mesh refinement introduces the new points improving the quality of the mesh along the way, and finally mesh quality optimization.

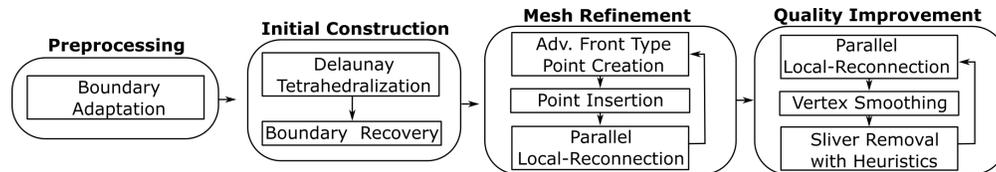


Figure 2: CDT3D mesh generation pipeline

2.2 CFD Solver

SU2 (Economon, Palacios, Copeland, Lukaczyk, and Alonso 2016) is a general framework for solving sets of governing equations for multi-physics problems. It is a vertex-based solver and uses dual control volumes for determining the required quantities. It can use either a finite volume method or finite element method

with an edge-based data structure. Both centered and upwind spatial integration schemes are provided. Moreover, it includes features like agglomeration multigrid and preconditioners for low-speed applications which are both utilized in this study. SU2 utilizes MPI to exploit parallelism in distributed memory environments.

2.3 Mutliscale Metric

The mutliscale metric (Loseille, Dervieux, Frey, and Alauzet 2007) is used in order to control the L^p -norm of the interpolation error of a given scalar field. The multiscale metric is evaluated based on the reconstruction the Hessian \mathcal{H} of the given scalar field.

It has been shown experimentally (Loseille and Alauzet 2011b) and theoretically (Loseille and Alauzet 2011a) that a mesh conforming to \mathcal{M}_{L^p} provides optimal control of the scalar field interpolation error in the p -norm.

In this study, the implementation of `refine`'s multi-scale metric is used (Park 2018). In particular, the L^2 -projection gradient reconstruction scheme is used for Hessian reconstruction and the order of the metric norm is set to $p = 2$.

3 EXPERIMENTAL SET-UP

In order to meet the ever-evolving and growing demands of the CFD community, a simulation pipeline should be able to integrate a plethora of different tools. The T-infinity project (O'Connell, Druyor, Thompson, Jacobson, Anderson, Nielsen, Carlson, Park, Jones, Biedron, Lee-Rausch, and Kleb 2018), demonstrates a series of different use-cases where a high-level Python interface can be used to build complicated pipelines. In this work, due to the small scale (in terms of computational effort) and complexity of the application, a more straightforward pipeline was built communicating data solely through files. However, the API that the T-infinity project is proposing will be the goal for future applications.

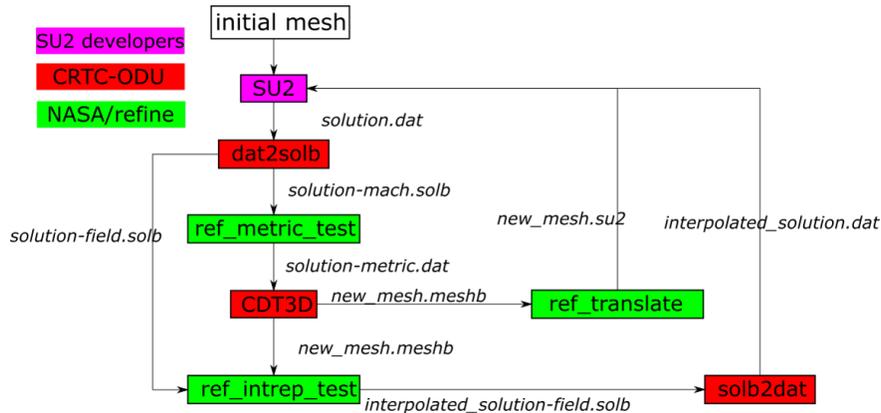


Figure 3: Execution pipeline for the simulation in this study

The absence of common file formats for meshes and solutions required the creation some converters. The final pipeline can be seen Figure 3. `dat2solb` and `solb2dat` are used to convert the solution format between SU2 and the `refine` tools. `ref_metric_test` creates the metric tensor out of the current solution. `ref_intrep_test` interpolates the old solution to the new mesh. This step allows the solver to restart the calculation from a state closer to the final solution than starting from the freestream values which is the default. The values of the previous solution are interpolated using linear interpolation. Finally,

`ref_translate` converts the mesh from the `.meshb` format to the `.su2` so that the solver can process it. Although, at first this pipeline may seem highly tailored to the specific format, the flow of data follows the abstract pipeline of mesh adaptation of Figure 1.

The simulation chosen for this study is a laminar flow over a delta wing. The flow conditions have been set so that they match the case used in the first three High-Order Workshops (Wang, Fidkowski, Abgrall, Bassi, Caraeni, Cary, Deconinck, Hartmann, Hillewaert, Huynh, Kroll, May, Persson, Leer, and Visbal 2013). In particular, the freestream conditions are 0.3 Mach, 4000 Reynolds number based on a root chord length of 1 and 12.5° angle of attack. The wing surface is modeled as an isothermal no-slip boundary with the freestream temperature. The Prandtl number is 0.72. The viscosity is assumed constant.

As initial mesh, a model from the Unstructured Grid Adaptation Working Group (UGAWG)¹ was used. This mesh can be seen in Figure 4. The computations were performed on a 24 core workstation (2 x Intel[®] Xeon[®] E5-2697 v2 @ 2.70GHz) with 757GB RAM.

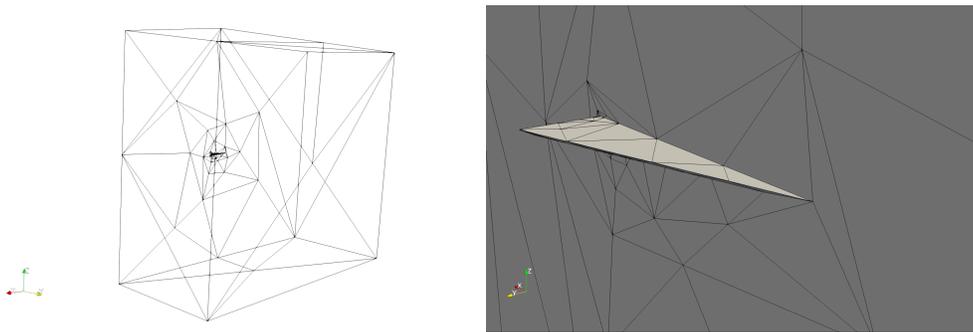


Figure 4: Initial mesh, # vertices 143 # tetrahedra 434, available at <https://github.com/UGAWG/solution-adapt-cases>

4 RESULTS

In this section, both quantitative and qualitative results are presented.

Figure 9 in appendix A depicts the evolution of the solution after each adaptation iteration. The generated Mach contour lines capture the features of the vortex and match the ones presented in (Leicht and Hartmann 2010).

Another important issue when simulating flow where vortices are present is that the vortex created by the flow should be resolved in the wake region of the wing. Since this is the expected behavior for these simulation parameters both in physical and computational tests. Figure 5 depicts the evolution of the solution at a distance of 4 chord lengths from the trailing edge.

The structure of the vortex core can also be seen in Figure 6 where streamlines are used to visualize the flow through the vortex.

One of the most significant advantages using metric-based anisotropic mesh adaptation is that the same level of accuracy can be achieved with a smaller mesh reducing thus the running time of all the components in the computational pipeline. To illustrate this result for this study, two more configurations were tested both using the same mesh adaptation tool. First, as a baseline, a uniform mesh refinement was used. In the uniform refinement case the size of all the elements was fixed and for every iteration it was approximately reduced to half. Second, an isotropic size was derived from the metric tensor by disregarding the directional

¹<https://ugawg.github.io/>

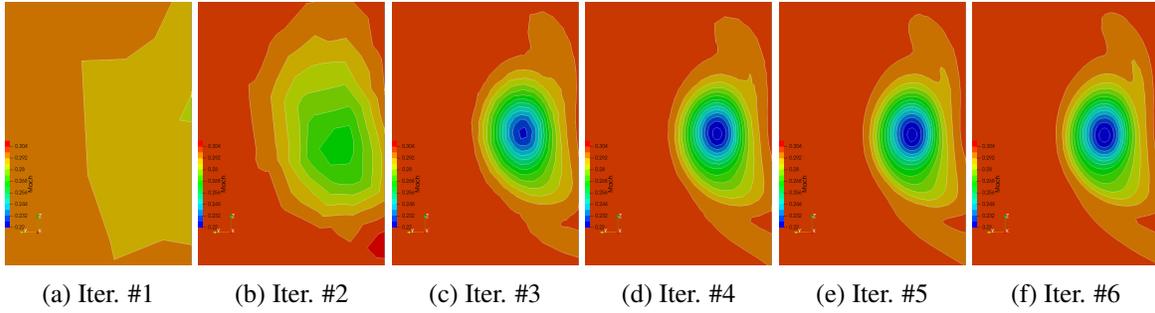


Figure 5: Slice of the vortex core in the wake region 4 cord lengths from the trailing edge at each iteration.

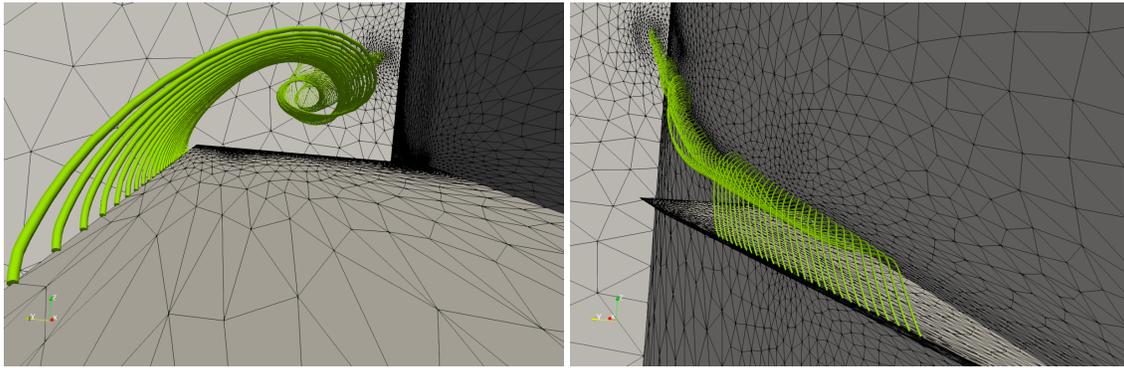


Figure 6: Streamlines of the final solution, # vertices 122,384 # tetrahedra 714,018

information (i.e. its eigenvectors) and using the smallest directional size (i.e. the largest eigenvalue) to control the local sizing.

The three mesh refinement approaches were compared against each other using the lift and drag coefficients as calculated by SU2 (see Figure 7). As a reference, the values $C_l^{ref} = 0.347$ and $C_d^{ref} = 0.1658$ have been used (Leicht and Hartmann 2010).

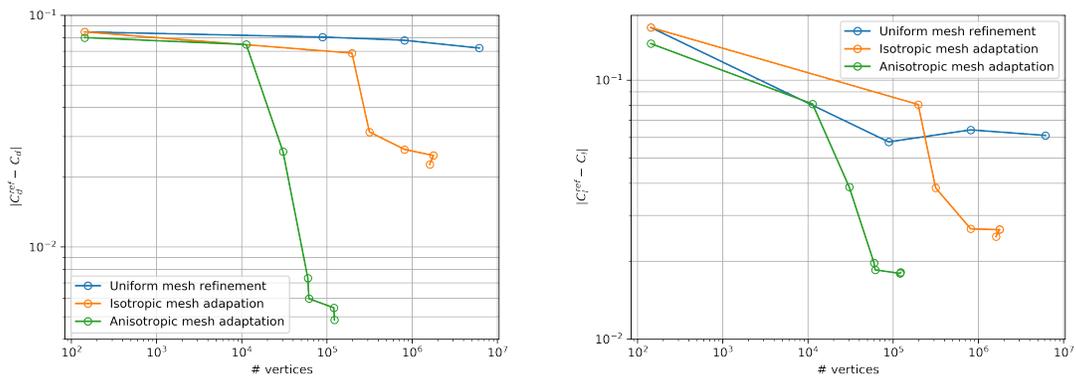


Figure 7: Absolute error of lift and drag coefficients, for three different types of mesh refinement.

	Mesh Refinement Method		
	Uniform (3 iterations)	Adaptive Isotropic (6 Iterations)	Adaptive Anisotropic (6 Iterations)
Solver Time	817.28 min	151.95 min	93.05 min
Mesh Adaptation Time	23.75 min	14.31 min	12.05 min

Table 1: Simulation time comparison for the three mesh refinement approaches

Figure 8 depicts the upper surface of the wing of the final iteration for each mesh refinement approach. The anisotropic approach packs more efficiently the points along the pressure contour lines. The uniform mesh is also included for reference.

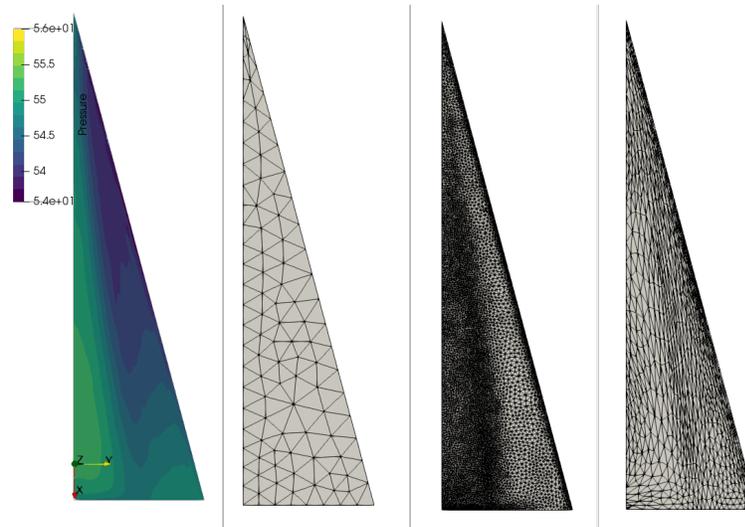


Figure 8: From the left : Pressure contours on the surface of the wing. Final mesh for the uniform, isotropic and anisotropic case. Respective number of vertices : 6.140.030, 1.609.346 , 122.384

The anisotropic mesh adaptation outperforms the other two approaches, in particular it can achieve lower error with an order of magnitude fewer vertices. Table 1 indicates that these gains are also reflected in the total time of the simulation.

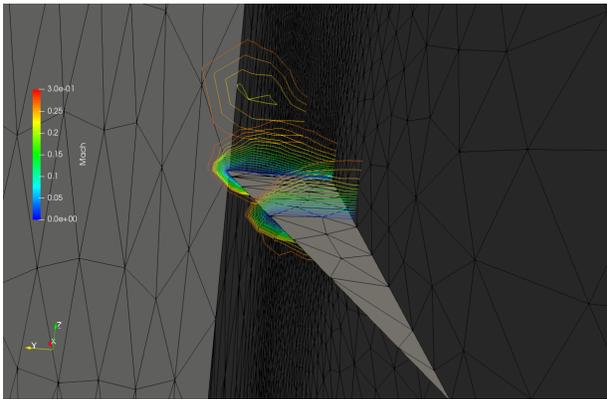
5 CONCLUSION

In this work, three different software components were brought together and a simulation pipeline was built. The simulation results are promising and agree with already published work. The use of anisotropic mesh adaptation enables lower error with an order of magnitude fewer elements, that can also be extrapolated to an order of magnitude lower use of resources, power and time optimizing thus user-productivity. In the future, the same case will be executed at higher complexity to explore both the solution convergence as well as to study the scalability of all three components.

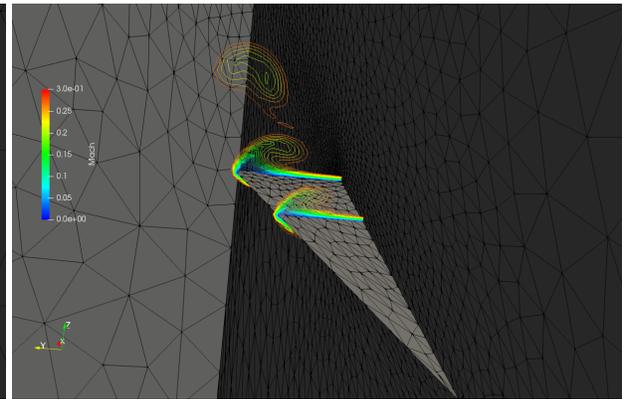
ACKNOWLEDGMENTS

This research was sponsored in part by the NASA Transformational Tools and Technologies Project (NNX15AU39A) of the Transformative Aeronautics Concepts Program under the Aeronautics Research Mission Directorate, NSF grant no. CCF-1439079, the Richard T. Cheng Endowment and the Modeling and Simulation fellowship of Old Dominion University.

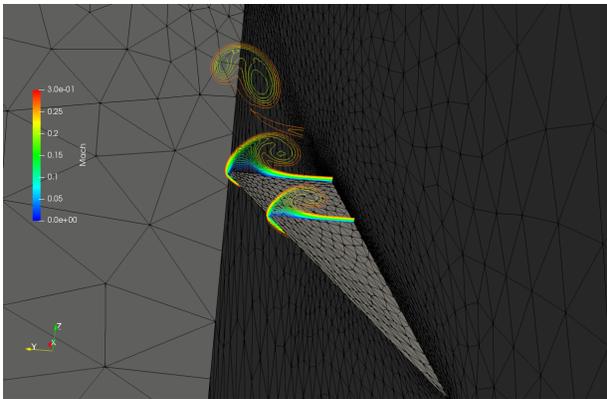
A APPENDIX



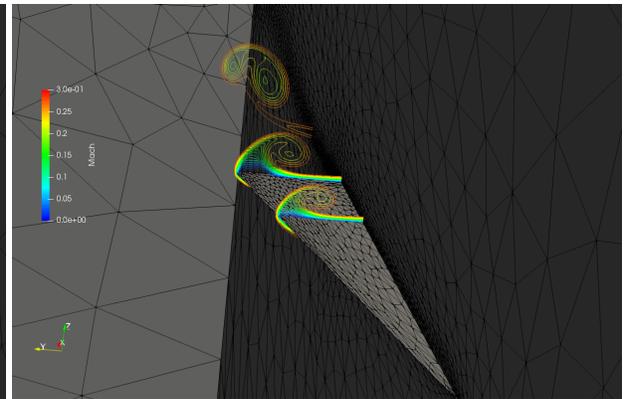
(a) # vertices 11,313 # tetrahedra 58,333



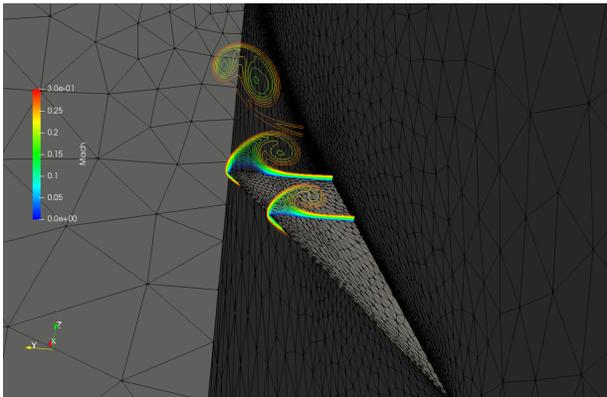
(b) # vertices 30,509 # tetrahedra 171,425



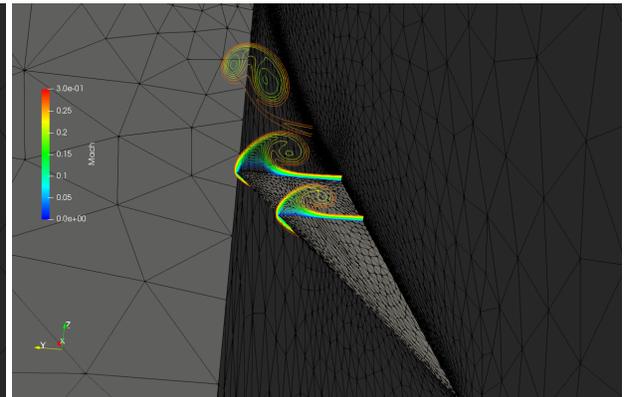
(c) # vertices 59,726 # tetrahedra 344,379



(d) # vertices 61,455 # tetrahedra 355,650



(e) # vertices 120,612 # tetrahedra 702,791



(f) # vertices 122,384 # tetrahedra 714,018

Figure 9: Adapted surface mesh and contour lines of the Mach number for each iteration.

REFERENCES

- Alauzet, F., and A. Loseille. 2016, March. “A decade of progress on anisotropic mesh adaptation for computational fluid dynamics”. *Computer-Aided Design* vol. 72, pp. 13–39.
- Chrisochoides, N. P. 2016. “Telescopic Approach for Extreme-Scale Parallel Mesh Generation for CFD Applications”. In *46th AIAA Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics.
- Dobrzynski, C., and P. Frey. 2009. “Anisotropic Delaunay Mesh Adaptation for Unsteady Simulations”. In *Proceedings of the 17th International Meshing Roundtable*, edited by R. V. Garimella, pp. 177–194. Springer Berlin Heidelberg.
- Drakopoulos, F. 2017. *Finite Element Modeling Driven by Health Care and Aerospace Applications*. Ph. D. thesis, Computer Science, Old Dominion University.
- Drakopoulos, F., C. Tsolakis, and N. P. Chrisochoides. 2017. “Fine-Grained Speculative Topological Transformation Scheme for Local Reconnection Methods”. In *AIAA Aviation Forum*, pp. accepted.
- Economon, T. D., F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso. 2016. “SU2: An Open-Source Suite for Multiphysics Simulation and Design”. *AIAA Journal* vol. 54 (3), pp. 828–846.
- Leicht, T., and R. Hartmann. 2010, September. “Error estimation and anisotropic mesh refinement for 3d laminar aerodynamic flow simulations”. *Journal of Computational Physics* vol. 229 (19), pp. 7344–7360.
- Loseille, A., and F. Alauzet. 2011a. “Continuous Mesh Framework Part I: Well-Posed Continuous Interpolation Error”. *SIAM Journal on Numerical Analysis* vol. 49 (1), pp. 38–60.
- Loseille, A., and F. Alauzet. 2011b, January. “Continuous Mesh Framework Part II: Validations and Applications”. *SIAM Journal on Numerical Analysis* vol. 49 (1), pp. 61–86.
- Loseille, A., A. Dervieux, P. Frey, and F. Alauzet. 2007, June. “Achievement of Global Second Order Mesh Convergence for Discontinuous Flows with Adapted Unstructured Meshes”. In *18th AIAA Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences*. American Institute of Aeronautics and Astronautics.
- O’Connell, M., C. Druyor, K. B. Thompson, K. Jacobson, W. K. Anderson, E. J. Nielsen, J.-R. Carlson, M. A. Park, W. T. Jones, R. Biedron, E. M. Lee-Rausch, and B. Kleb. 2018, June. “Application of the Dependency Inversion Principle to Multidisciplinary Software Development”. In *2018 Fluid Dynamics Conference, AIAA AVIATION Forum*. American Institute of Aeronautics and Astronautics.
- Park, Michael A. 2018. “refine GitHub site”. <https://github.com/nasa/refine>.
- Slotnick, J., A. Khodadoust, J. Alonso, D. Darmofal, W. Gropp, E. Lurie, and D. Mavriplis. 2014, March. “CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences”. Technical report.
- Tsolakis, C., N. Chrisochoides, M. A. Park, A. Loseille, and T. Michal. 2019. “Parallel Anisotropic Unstructured Grid Adaptation”. In *AIAA SciTech Forum 2019*.
- Tsolakis, C., F. Drakopoulos, and N. Chrisochoides. 2018, April. “Sequential Metric-Based Adaptive Mesh Generation”. In *Modeling, Simulation, and Visualization Student Capstone Conference*. Suffolk, VA.
- Wang, Z. J., K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, N. Kroll, G. May, P.-O. Persson, B. v. Leer, and M. Visbal. 2013. “High-order CFD methods: current status and perspective”. *International Journal for Numerical Methods in Fluids* vol. 72 (8), pp. 811–845.