High-order scale resolving entropy stable CFD for large scale applications

L. Galimberti[*,**], R. Nuca[*], L. Dalcin[*], A. Guardone[**] and M. Parsani[*]
Corresponding author: luca.galimberti@kaust.edu.sa
[*] King Abdullah University of Science and Technology (KAUST), Computer Electrical and Mathematical Science and Engineering Division (CEMSE), Thuwal, Saudi Arabia.
[**] Department of Aerospace Science and Technology, Politecnico di Milano, Building B12, Via La Masa 34, Milano, MI 20156, Italy.

Abstract: High-order numerical methods offer greater accuracy and efficiency than traditional low-order approaches, but their practical adoption can be limited by stability and meshing challenges. Ensuring nonlinear stability is key to making these methods robust and reliable for real-world engineering problems, especially in situations involving moving meshes, such as fluid-structure interactions, moving parts, and adaptive grids. The compressible Navier–Stokes (CNS) equations possess a convex entropy function, related to thermodynamic entropy. This function acts as an admissibility criterion for identifying physically relevant weak solutions and provides a means of demonstrating nonlinear (entropy) stability by bounding the solution in the L^2 norm. Maintaining entropy stability for CNS equations has been a major focus of research. Recent advances show entropy stability can be preserved even as computational grids deform, using a discontinuous Galerkin split form of the arbitrary Lagrangian Eulerian (ALE) approach. This work builds upon these results, demonstrating the effectiveness of high-order entropy stable DG methods in large-scale simulations with mesh motion. In particular, a canonical test case for mesh motion validation is presented, along with an oscillating airfoil in dynamic stall and the start of a propeller flow.

Keywords: High-order, Entropy stability, Moving grids, Scale resolving simulations.

1. Introduction

High-order methods are particularly attractive because they can deliver a given level of accuracy with fewer degrees of freedom (DOFs) than low-order methods. They typically exhibit improved dispersion and dissipation properties; however, their robustness depends critically on the design of provably stable formulations. Ensuring nonlinear stability is key to making these methods robust and reliable for real-world engineering problems, specifically, in situations involving moving meshes, such as fluid-structure interactions, moving parts, and adaptive or deforming grids. The Arbitrary Lagrangian Eulerian (ALE) method has been widely adopted over the past century, beginning with the seminal work of Hirt [1]. Many different formulations have applied the method to incompressible flows [2], fluid-structure interaction problems [3], focusing the analysis on linear stability [4], geometric conservation laws (GCL) [5], and high-order spectral behavior [6, 7]. More recently, the ALE formulation has been successfully applied to summation by parts discretizations of the Discontinuous Galerkin Method [8, 9, 10]. These works proved entropy stability and entropy conservation properties for the compressible Euler and Navier–Stokes equations (CNS). These hyperbolic systems of equations possess a convex entropy function, related to thermodynamic entropy. This function serves as an admissibility criterion for identifying physically relevant weak solutions and provides a means of demonstrating nonlinear (entropy) stability by bounding the solution in the L^2 norm. This work applies a similar formulation to large scale applications involving different types of moving grids, geometries, and flow physics. Section 2 briefly introduces the system of equations and the numerical discretization characteristics. Section 3 presents the three test cases, compares them to reference solutions, and comments on the physical results.

2. Methodology

Considering the Navier–Stokes equations in conservation form, we define a space-time mapping that accounts for both curvilinear coordinates and mesh motion. The mapping transforms each element to the reference element for which all the computations are performed.

$$\frac{\partial \mathbf{U}}{\partial t}+\sum_{m=1}^{3}\frac{\partial \mathbf{F}^{I}_{x_m}}{\partial x_m}=\sum_{m=1}^{3}\frac{\partial \mathbf{F}^{(v)}_{x_m}}{\partial x_m}, \tag{1}$$

where

$$\mathbf{Q} = [\rho, \rho U_1, \rho U_2, \rho U_3, \rho E]^T \tag{2}$$

$$\mathbf{F}^{I}_{x_m} = \left[\rho U_m, \rho U_m U_1 + \delta_{m,1}P, \rho U_m U_2 + \delta_{m,2}P, \rho U_m U_3 + \delta_{m,3}P, \rho U_m H\right]^T \tag{3}$$

$$\mathbf{F}^{V}_{x_m} = \left[0, \tau_{1,m}, \tau_{2,m}, \tau_{3,m}, \sum_{i=1}^{3}\tau_{i,m}U_i - \kappa\frac{\partial T}{\partial x_m}\right]^T \tag{4}$$

are the conservative variables, the inviscid fluxes, and the viscous fluxes.

$$\tau_{i,j} = \mu\left(\frac{\partial U_i}{\partial x_j}+\frac{\partial U_j}{\partial x_i}-\delta_{i,j}\frac{2}{3}\sum_{n=1}^{3}\frac{\partial U_n}{\partial x_n}\right) \tag{5}$$

are the viscous stresses. The system of equations is closed by the thermodynamic constituent relations:

$$H = c_P T + \frac{1}{2}\mathbf{U}^T\mathbf{U}, \qquad P = \rho R T, \qquad R = \frac{R_u}{M_w}, \tag{6}$$

where c_P is the specific heat capacity at constant pressure, T is the temperature, R is the gas constant and M_w is the molecular weight of the fluid.
Considering a generic coordinate mapping from the reference domain $\xi \in \bar{\Omega} = [-1,1]^3$ at time τ, to the physical domain $\mathbf{x} \in \Omega$ at time t. Where the Jacobian of the transformation is defined as $\mathcal{J} = \left|\frac{\partial(t,\mathbf{x})}{\partial(\tau,\xi)}\right|$. It is possible to derive the generic form of the Navier–Stokes equations as

$$\frac{\partial\left(\mathcal{J}\mathbf{Q}\right)}{\partial \tau}+\sum_{l,m=1}^{3}\frac{\partial}{\partial \xi_l}\left(\mathcal{J}\frac{\partial \xi_l}{\partial x_m}\left(\mathbf{F}^{\mathbf{I}}_{\mathbf{x_m}} - V_m\mathbf{Q} - \mathbf{F}^{\mathbf{V}}_{\mathbf{x_m}}\right)\right) = 0, \tag{7}$$

$$\frac{\partial \mathcal{J}}{\partial \tau}-\sum_{l,m=1}^{3}\frac{\partial}{\partial \xi_l}\left(\mathcal{J}\frac{\partial \xi_l}{\partial x_m}V_m\right) = 0, \tag{8}$$

$$\sum_{l=1}^{3}\frac{\partial}{\partial \xi_l}\left(\mathcal{J}\frac{\partial \xi_l}{\partial x_m}\right) = 0, \qquad m = 1,2,3. \tag{9}$$

where (8) and (9), are the GCL equations in space and time. Yamaleev [10] and Schnucke [9] proved that this system of equations is entropy stable provided adequate boundary conditions. The proofs demonstrate both at the continuous and semi-discrete level that entropy conservation is achievable for the system of equations. In this work, we solve the compressible Navier–Stokes equations [11] with a high-order entropy-stable discontinuous collocated Galerkin method. We use summation-by-parts operators to compute spatial derivatives. Fluxes are evaluated with the two-point flux function from [12], with a penalty type of approach between cells and at boundary conditions [13, 14]. The two-point flux function is modified and takes into account mesh motion. For this set of test cases, we also consider a rigid motion of the full grid. This allows us to guarantee that both the space and time GCL equations are always verified. It also reduces the computational cost since there is no need to solve a discretized Jacobian in time together with the hyperbolic system, like proposed by [10, 9]. On the other hand, this approach is limited to rigid movements and prevents any type of grid deformation. The SSDC solver is built on top of the highly scalable Portable and Extensible Toolkit for Scientific Computing (PETSc) [15], its mesh topology abstraction (DMPlex), and its scalable differential–algebraic equation solver components. The method handles unstructured curvilinear meshes, which allow to simulate complex shapes. Quadrilateral and hexahedral elements are handled natively, whereas triangular and tetrahedral meshes are converted on the fly by splitting. The collocation nodes inside each element are distributed according to the Gauss–Legendre–Lobatto quadrature points; in addition, both h and p adaptation are handled by the method. Explicit time integration is performed using the third-order Runge–Kutta scheme of Bogacki–Shampine [16] with four stages, with the first-same-as-last property. Adaptive time stepping is achieved with the embedded second-order method of this Runge–Kutta scheme.

3. Numerical Results

The following Section reports 3 different test cases with increasing difficulty. First, a 2d airfoil is compared to state of the art high-fidelity solvers. Then, a canonical 3d LES computation for an airfoil undergoing dynamic stall is performed. Finally, a novel LES computation of a starting propeller in forward flight is performed to showcase the applicability and the scalability of the method.

3.1 Heaving-pitching Airfoil

We consider the flow about a NACA-0012 airfoil undergoing a heaving and pitching motion. This test case has been considered for the High-fidelity CFD verification workshop [17, 18]. The flow is at $Re = 1000$, and $Ma = 0.2$, with a unitary chord length. The airfoil motion is defined by 2 degrees of freedom: the height $h(t)$ with respect to the initial position, and the pitching angle $\theta(t)$ with respect to the point at $\frac{1}{3}$ of the chord length.

$$h(t) = \frac{1}{2}t^3 - \frac{3}{16}t^4 \tag{10}$$

$$\theta(t) = \frac{4\pi}{9}\left(-t^6 + 6t^5 - 12t^4 + 8t^3\right) \tag{11}$$

We consider the more complicated motion that combines both heaving and pitching. Moreover, the native grid from the workshop is used to reduce the variability compared to the reference data. Figure 1 shows a close-up of the grid, few structured layers are then transitioned into a fully unstructured grid. Each cell is discretized with third order polynomials collocated on the LGL points. Even though the flow is laminar, this specific configuration is chosen since the abrupt polynomial motion and the high variation of the angle of attack generate a strong force variation and allow for proper testing of the mesh motion capabilities of the solver. The computation is started from the converged static solution of the airfoil. Figure 2 shows the forces in the streamwise and perpendicular direction along time. A total of 2 convective time units is simulated to perform the maneuver. Overall, there is strong agreement compared to other solvers from [18]. Similarly, considering an instantaneous snapshot of the vorticity field, Figure 3 phenomenologically resembles the results from [18].

3.2 Dynamic stall of a SD7003 airfoil

A famous test case characterised by airfoil motion is the SD7003 in dynamic stall due to plunging, originally investigated by [19]. The airfoil has a unitary chord length and is placed in a uniform free stream. The flow conditions are $Re = \frac{\rho_\infty U_\infty c}{\mu_\infty} = 4.0 \times 10^4$, $Ma = 0.1$, with a fixed geometric angle of attack $\alpha_0 = 4^\circ$. A sinusoidal plunging motion is imposed in the direction normal to the incoming flow.

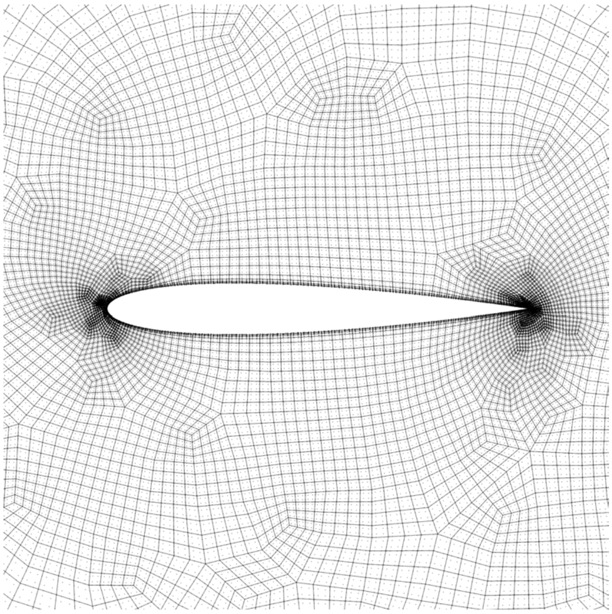
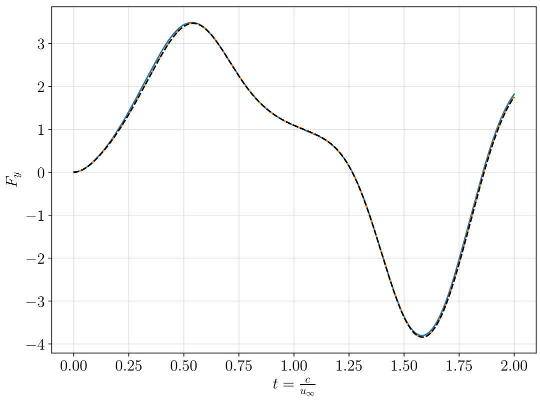
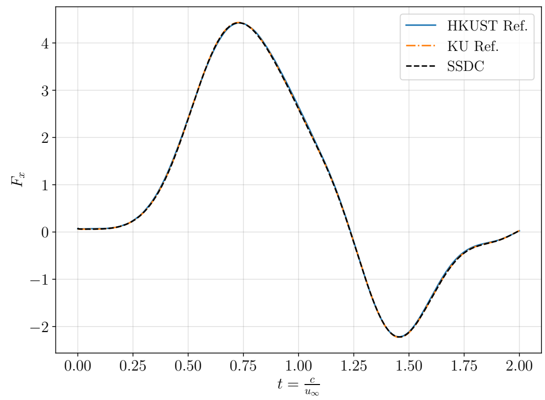


Figure 1: Computational grid with LGL collocation points, degree 3. The grid is the same as the workshop [18].



(a) Aerodynamic force in the y direction F_y comparison with digitized reference computation.



(b) Aerodynamic force in the x direction F_x comparison with digitized reference computation.

Figure 2: Time series of the load lift and drag aerodynamic coefficients.

The vertical displacement of the airfoil is prescribed as

$$h(t) = h_0 \sin\left(2kF(t)t\right), \tag{12}$$

where (h_0) is the plunging amplitude and (k) is the reduced frequency. The reduced frequency is defined as $k = \frac{\pi f c}{U_\infty}$, with ($f$) denoting the plunging frequency. In the present configuration, the motion parameters are $\frac{h_0}{c} = 0.05$, $k = 3.93$.

The function ($F(t)$) is used to smoothly increase the oscillation amplitude from the initially developed static-airfoil flow to the fully periodic plunging motion. It is written as

$$F(t) = 1 - \exp\left(-at\right), \qquad a = 9.2. \tag{13}$$

This ramp avoids an impulsive start of the motion and reduces the strength of artificial transients during the initial cycles. The plunging velocity modifies the instantaneous effective incidence of the airfoil.

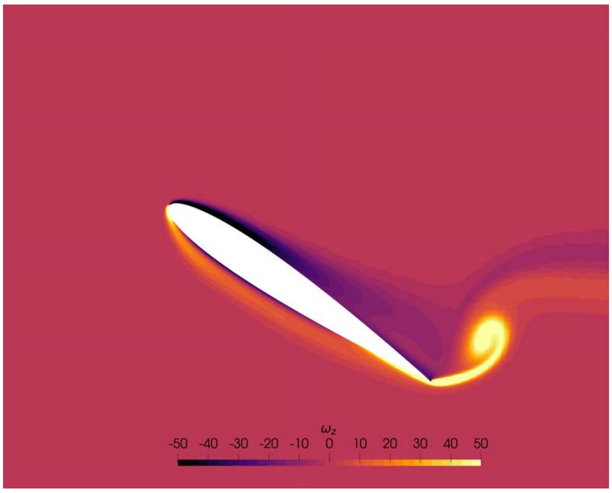


Figure 3: Instantaneous vorticity field for the heaving+pitching motion at $t = 0.5$.

With the adopted sign convention, the effective angle of attack can be expressed as

$$\alpha_{\text{eff}}(t) = \alpha_0 - \tan^{-1}\left(\frac{\dot{h}(t)}{U_\infty}\right). \tag{14}$$

For the selected reduced frequency and amplitude, the variation of (α_{eff}) is sufficiently large to produce leading edge separation and dynamic-stall vortex formation during part of the plunging cycle. An unstructured grid is generated starting from a structured boundary layer and an unstructured farfield circular domain, positioned at 50 chord lengths. Similar spacings as in [8] are considered. A uniform degree equal to 7 is considered, with a spanwise extent of the computational domain $L_z = 0.2c$, and periodic boundary conditions are imposed in the spanwise direction. The outer boundaries are placed far from the airfoil and are assigned freestream conditions, while the airfoil surface is treated as an adiabatic no-slip wall. The mesh motion is imposed analytically as a rigid motion of the full computational domain. The simulation is initialized from the fully developed flow around the static airfoil. The computed flow field shows the characteristic sequence associated with dynamic stall. During the downward motion of the airfoil, the effective angle of attack increases, and the suction-side boundary layer separates near the leading edge. A coherent leading-edge vortex forms and is convected downstream over the airfoil surface. As this vortex grows, it becomes unstable and breaks down into three-dimensional structures, producing a localized transitional region above the airfoil. Further downstream, vortical structures generated in previous cycles remain visible in the wake. During the upward part of the cycle, the effective angle of attack decreases, and the leading edge region reattaches. A similar separation process occurs on the pressure side during the opposite part of the motion, but the resulting vortical structures are weaker because of the positive mean angle of attack.

The aerodynamic force coefficients are defined as

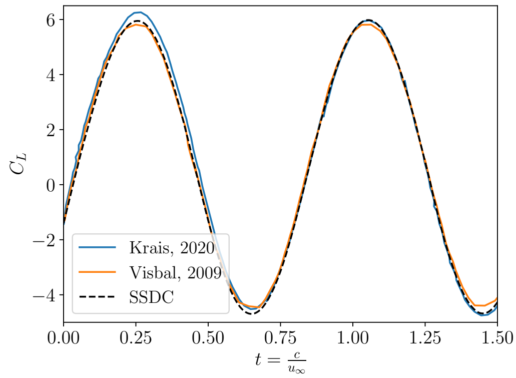
$$C_D = \frac{F_D}{\frac{1}{2}\rho_\infty U_\infty^2 S}, \tag{15}$$

and

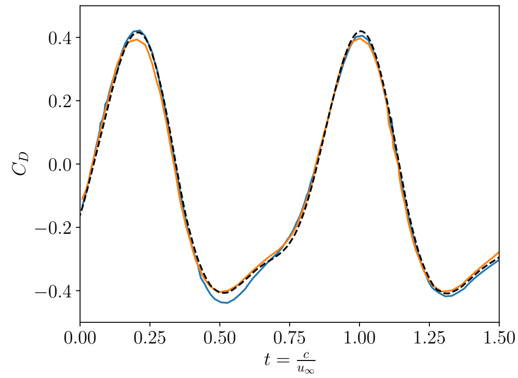
$$C_L = \frac{F_L}{\frac{1}{2}\rho_\infty U_\infty^2 S}, \tag{16}$$

where (F_D) and (F_L) are the streamwise and transverse force components, respectively, and (A) is the reference airfoil surface area.

The mean value of ($C_D = -0.0823$) is negative, and it is consistent with the original value from [19]. The obtained thrust level is close to that reported in the reference computation, and the temporal histories of both (C_D) and (C_L) exhibit the expected periodic behavior, as shown in Figure 4. Finally, Figure 5 shows a 3-dimensional visualization of the minimum and maximum lift conditions, which respectively correspond to the bottom and the top of the trajectory at two of the instantaneous peaks captured in the computation. Overall, the agreement in the aerodynamic coefficients, together with the close phenomenological correspondence of the vortical structures, demonstrates that the present numerical framework accurately reproduces the main physical mechanisms of the plunging-airfoil benchmark.

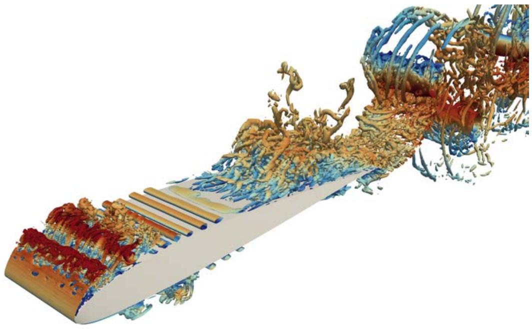


(a) Lift coefficient C_L comparison with digitized reference computation.

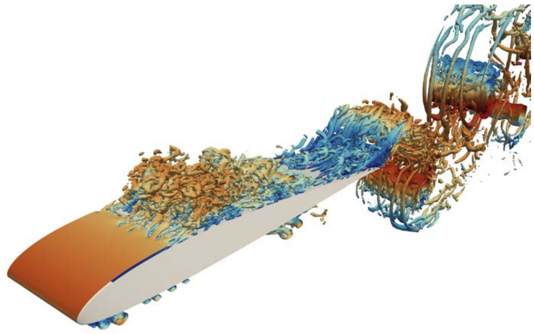


(b) Drag coefficient C_D comparison with digitized reference computation.

Figure 4: Time series of the lift and drag aerodynamic coefficients.



(a) Minimum lift condition.



(b) Maximum lift condition.

Figure 5: Instantaneous flow visualization at different phases of the plunging motion. Iso-surface of the second invariant of the velocity gradient tensor ($Q = 50$) colored by streamwise velocity (u).

3.3 Propeller in forward flight

We consider the flow about a starting propeller in forward flight. Similar high-fidelity computations have been performed for hovering cases in the works of [20, 21, 22]. In this case, we consider a simplified propeller geometry based on a NACA-0012 airfoil with a blunt trailing edge. The twist is linearly distributed along the chord from 15° at the root to 5° at the tip. The propeller has a radius of 0.15, and the blade sections have a fixed chord length of 0.02. We consider an angular velocity $\Omega = 6000$ rpm, and an advance ratio $J = \frac{u_\infty}{nd} = 0.4$, where u_∞ is the freestream velocity, n is the angular velocity in rotations per second, and d is the propeller diameter. In practice, $\frac{u_\infty}{n}$ represents the distance advanced by the propeller with a single revolution. The propeller doesn't have a hub; the two blades are separate and start at 20% in the radial direction of the blade. We perform a wall-resolved LES simulation of the starting propeller; the freestream Mach number is $Ma = 0.035$, whereas the Reynolds number based on the diameter and freestream velocity is $Re = 240000$. The grid is designed to achieve a streamwise spacing equivalent to

$\Delta x^+ = 30$, $\Delta y^+ = 1$, and $\Delta z^+ = 10$ at the wall. The wall region is extruded from a triangular surface discretization into prisms. Outside of the extruded region, the volume is filled with tetrahedral cells up to a spherical farfield with radius $33d$. As a result, an initial grid of mixed triangular prisms and tets is generated. The solver automatically handles this topology by splitting both prisms and tetrahedra into hexahedra. A uniform degree of 3 is used on all the elements. To kickstart the computation, at first, a coarser resolution is used to pass the initial acoustic perturbation; subsequently, a uniform h discretization defines the final grid for the computation. A total of 76 million cells, corresponding to $4.9 \cdot 10^9$ nodes, defines the computational domain. The timestep is automatically adapted to $\approx 10^{-7}$. Figure 6 shows a section of the grid topology for the lower refined grid. The grid shows a representation of cells defined by local LGL points. The fully refined computation has 2^3 times the number of nodes.

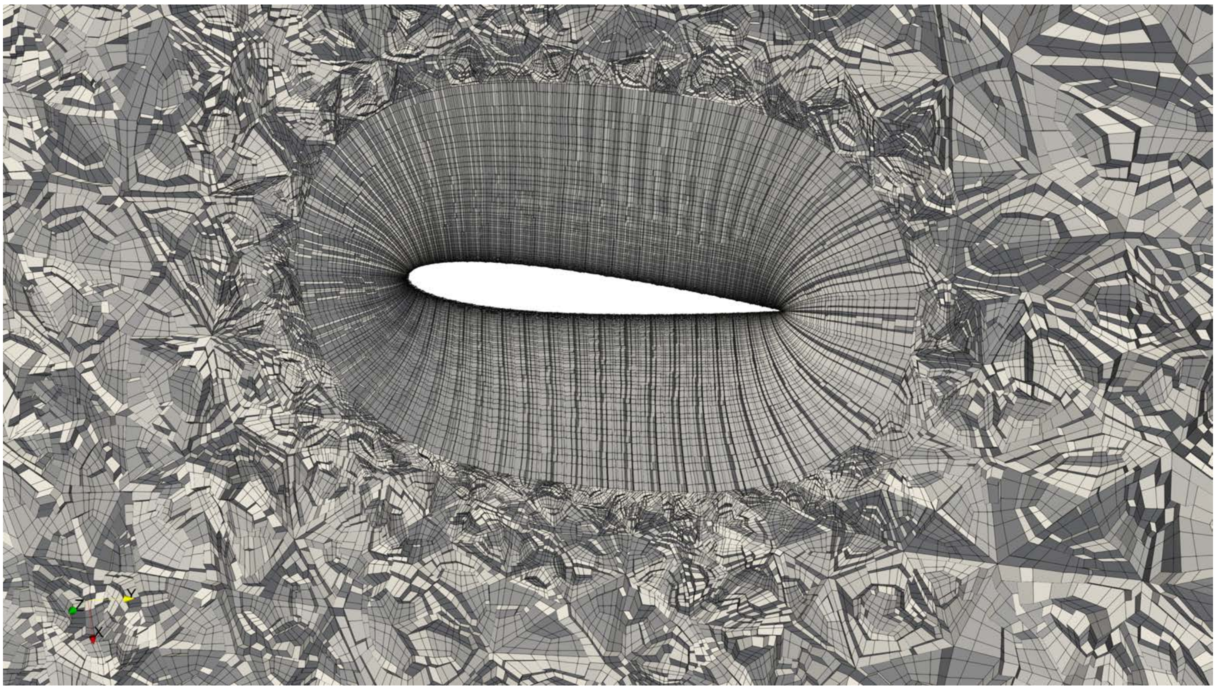


Figure 6: Slice of a blade section at 0.75% in the radial direction. Grid refinement at the beginning of the computation.

Preliminary results of the computation capture the first half of the revolution. In detail, figure 7 shows the pressure coefficient distribution on the suction side of the blade. Proceeding towards the trailing edge, it is possible to see significant pressure fluctuations due to turbulent eddies detaching from the surface of the blade. Considering the skin friction coefficient distribution, the tip vortex and root vortex are more visible, as well as the turbulent fluctuations on the surface. Figure 9 shows a close-up of the tip region; the tip vortex, given the abrupt geometry of the blade, has a strong impact on the flowfield on the suction side. Instantaneous visualizations of the flowfield show the development of the starting vortex and the turbulent structures on the suction side of the blade. Figure 11 shows part of the tip vortex disrupting the flow as it progresses towards the trailing edge of the airfoil.

4. Conclusions and Outlook

Overall, this work presents an analysis of several applications of the entropy stable ALE discontinuous Galerkin method with SBP operators, specifically aimed at achieving scale resolving computations of turbulent flows. The primary objective is to facilitate stable, accurate, and scalable simulations of turbulent flows that involve complex geometries and moving boundaries, which are often encountered in engineering applications. The current results support a validation of the framework with canonical test cases of increasing difficulty. The 2d heaving-pitching NACA-0012 establishes a common ground

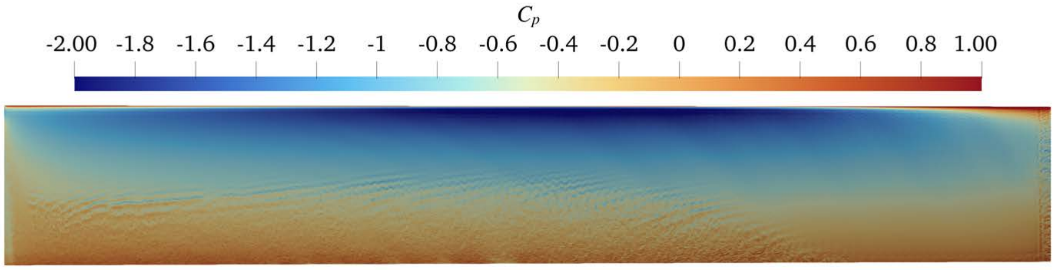


Figure 7: Surface distribution of the pressure coefficient C_p. Tip on the left, root on the right.

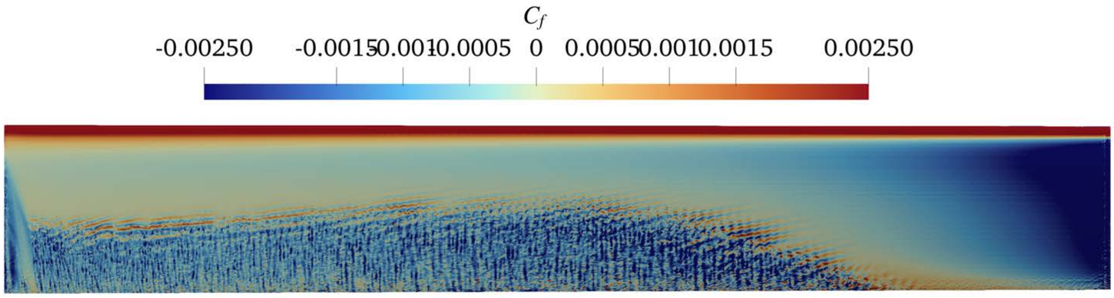


Figure 8: Surface distribution of the pressure coefficient C_f. Tip on the left, root on the right.

with similar solvers for simple motions and unstructured grids. A 3d iLES computation of the plunging SD7003 airfoil shows that the solver is scalable and can compute complex separation phenomena, with accurate prediction of flow structures and aerodynamic loads. The dynamic stall phenomenon is accurately reproduced when compared to reference data. Finally, the starting propeller shows a preliminary application of the method to a real-world test case, with a large number of degrees of freedom and complex flow physics. All in all, the computations are satisfactory, and the solver is tested with different polynomial degrees and types of motion. The propeller test case will be further investigated, as this is the first part of the computation, and possible future directions include but are not limited to aerodynamic, flowfield, and aeroacoustic analysis. A possible route to improve performance and reduce the overall cost is to consider hp adaptation throughout the computation.

Acknowledgements

The authors gratefully acknowledge the HPC resources provided for this effort by the Supercomputing Laboratory at King Abdullah University of Science & Technology (KAUST).

References

[1] C. W. Hirt, A. A. Amsden, and J. L. Cook. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14(3):227–253, 1974.

[2] Thomas J. R. Hughes, Wing Kam Liu, and Thomas K. Zimmermann. Lagrangian–Eulerian finite element formulation for incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 29(3):329–349, 1981.

[3] J. Donea, S. Giuliani, and J. P. Halleux. An arbitrary Lagrangian–Eulerian finite element method for transient dynamic fluid–structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33(1–3):689–723, 1982.

[4] Luca Formaggia and Fabio Nobile. Stability analysis of second-order time accurate schemes for

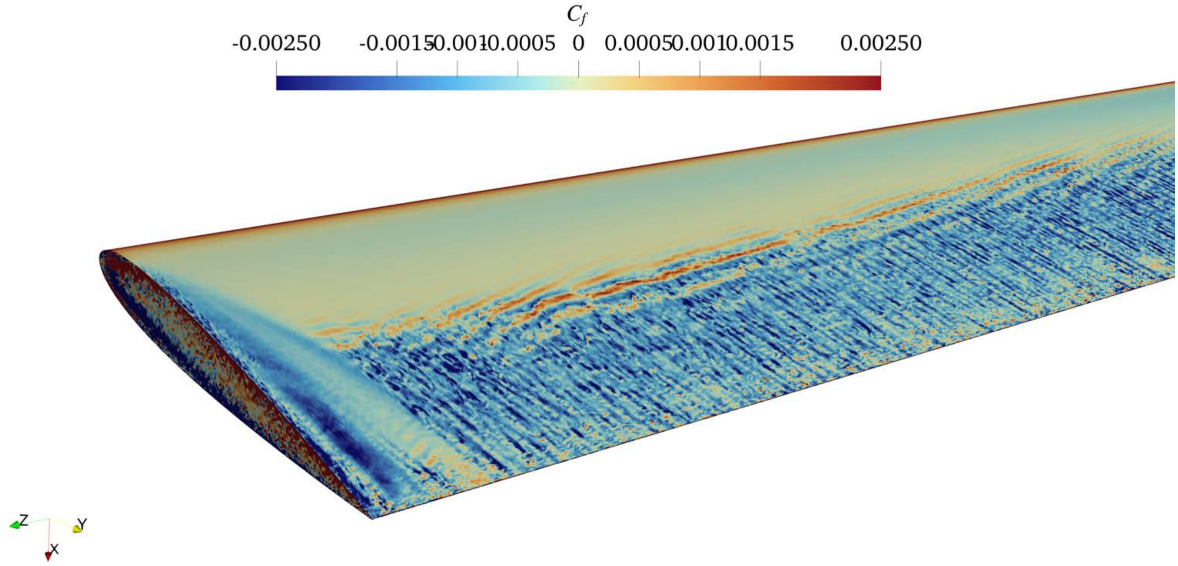


Figure 9: Surface distribution of the pressure coefficient C_f.

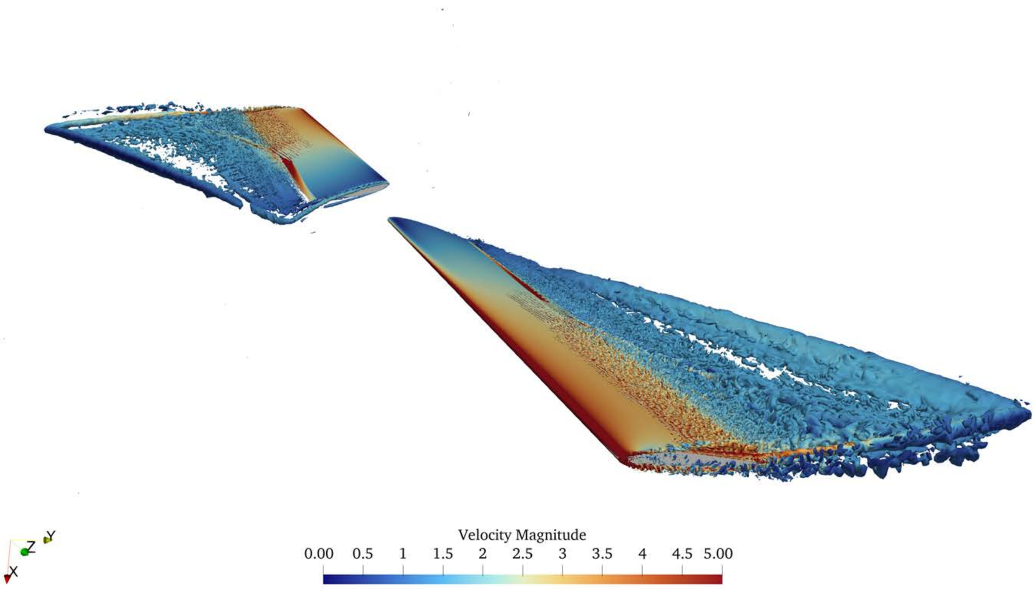


Figure 10: Iso-surface of the second invariant of the velocity gradient tensor ($Q = 50$) colored by streamwise velocity (u).

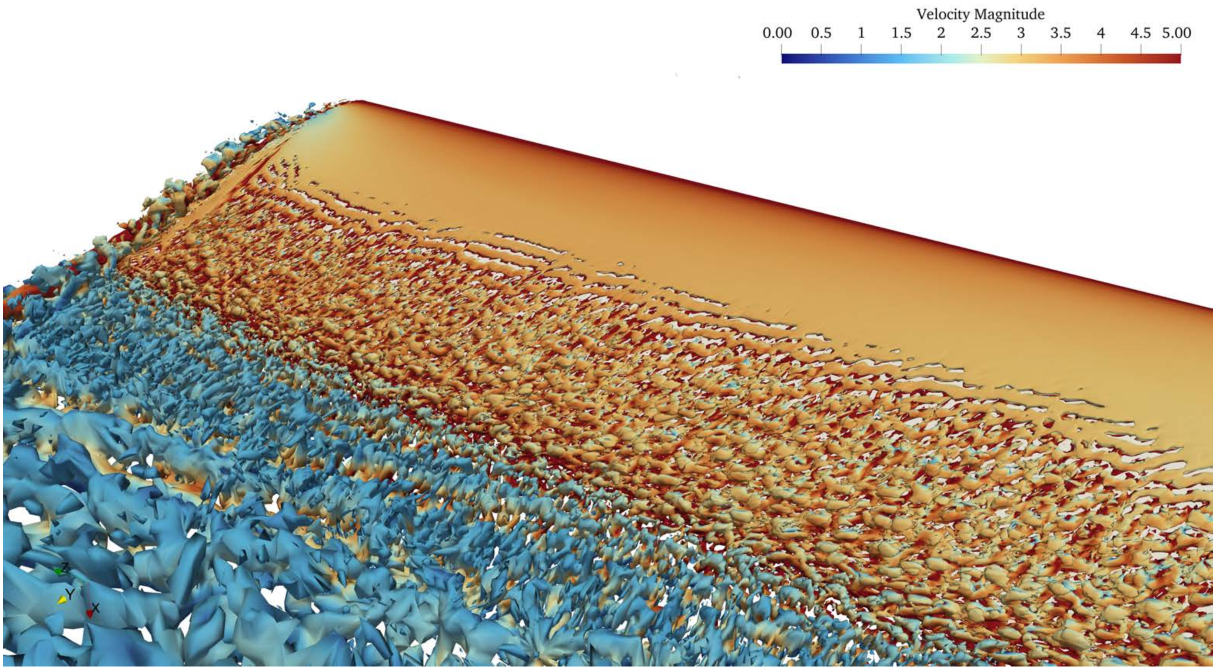


Figure 11: Iso-surface of the second invariant of the velocity gradient tensor ($Q = 100$) colored by streamwise velocity (u).

ALE–FEM. *Computer Methods in Applied Mechanics and Engineering*, 193(39–41):4097–4116, 2004.

[5] Charbel Farhat, Philippe Geuzaine, and Céline Grandmont. The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. *Journal of Computational Physics*, 174(2):669–694, 2001.

[6] David A. Kopriva, Andrew R. Winters, Marvin Bohm, and Gregor J. Gassner. A provably stable discontinuous galerkin spectral element approximation for moving hexahedral meshes. *Computers & Fluids*, 139:148–160, 2016.

[7] Samira Nikkar and Jan Nordström. Fully discrete energy stable high order finite difference methods for hyperbolic problems in deforming domains. *Journal of Computational Physics*, 291:82–98, 2015.

[8] Nico Krais, Gero Schnücke, Thomas Bolemann, and Gregor J. Gassner. Split form ale discontinuous galerkin methods with applications to under-resolved turbulent low-mach number flows. *Journal of Computational Physics*, 421:109726, 2020.

[9] Gero Schnücke, Nico Krais, Thomas Bolemann, and Gregor J. Gassner. Entropy stable discontinuous galerkin schemes on moving meshes for hyperbolic conservation laws. *Journal of Scientific Computing*, 82(3):69, 2020.

[10] Nail K. Yamaleev, David C. Del Rey Fernández, Jialin Lou, and Mark H. Carpenter. Entropy stable spectral collocation schemes for the 3-d navier-stokes equations on dynamic unstructured grids. *Journal of Computational Physics*, 399:108897, 2019.

[11] Matteo Parsani, Radouan Boukharfane, Irving Reyna Nolasco, David C. Del Rey Fernández, Stefano Zampini, Bilel Hadri, and Lisandro Dalcin. High-order accurate entropy-stable discontinuous collocated Galerkin methods with the summation-by-parts property for compressible CFD frameworks: Scalable SSDC algorithms and flow solver. *J. Comput. Phys.*, 424:109844, 2021.

[12] Praveen Chandrashekar. Kinetic energy preserving and entropy stable finite volume schemes for compressible euler and navier-stokes equations. *Communications in Computational Physics*, 14(5):1252–1286, 2013.

[13] Mark H. Carpenter, David Gottlieb, and Saul Abarbanel. Time-stable boundary conditions for

finite-difference schemes solving hyperbolic systems: Methodology and application to high-order compact schemes. *Journal of Computational Physics*, 111(2):220–236, 1994.

[14] Mark H Carpenter, Jan Nordström, and David Gottlieb. A stable and conservative interface treatment of arbitrary spatial accuracy. *Journal of Computational Physics*, 148(2):341–365, 1999.

[15] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Steven Benson, Jed Brown, Peter Brune, Kris Buschelman, Emil Constantinescu, Lisandro Dalcin, Alp Dener, Victor Eijkhout, Jacob Faibussowitsch, William D. Gropp, Václav Hapla, Tobin Isaac, Pierre Jolivet, Dmitry Karpeev, Dinesh Kaushik, Matthew G. Knepley, Fande Kong, Scott Kruger, Dave A. May, Lois Curfman McInnes, Richard Tran Mills, Lawrence Mitchell, Todd Munson, Jose E. Roman, Karl Rupp, Patrick Sanan, Jason Sarich, Barry F. Smith, Hansol Suh, Stefano Zampini, Hong Zhang, Hong Zhang, and Junchao Zhang. PETSc/TAO users manual. Technical Report ANL-21/39 - Revision 3.22, Argonne National Laboratory, 2024.

[16] P. Bogacki and L.F. Shampine. A 3(2) pair of runge - kutta formulas. *Applied Mathematics Letters*, 2(4):321–325, 1989.

[17] Nathan A. Wukie, Krzysztof Fidkowski, Per-Olof Persson, and Zhi J. Wang. High-fidelity cfd verification workshop 2024: Mesh motion. In *AIAA SCITECH 2023 Forum*, 2023.

[18] Nathan A. Wukie, Krzysztof Fidkowski, Per-Olof Persson, Takeshi R. Fujimoto, Zhi J. Wang, James Gabbard, and Wim van Rees. High-fidelity cfd verification workshop 2024: Post-workshop mesh motion summary. In *AIAA AVIATION FORUM AND ASCEND 2024*, 2024.

[19] Miguel R. Visbal. High-fidelity simulation of transitional flows past a plunging airfoil. *AIAA Journal*, 47(11):2685–2697, 2009.

[20] Christopher Thurman, D. Douglas Boyd Jr., Pieter Buning, Gabriel Reboul, and Christophe Benoit. *NASA/ONERA Collaboration on Small Hovering Rotor Broadband Noise Prediction Using Lattice-Boltzmann Method and Structured Navier-Stokes Solvers*. 2024.

[21] D.V. Shenoy, R. Gojon, T. Jardin, and M.C. Jacob. Aerodynamic and acoustic study of a small scale lightly loaded hovering rotor using large eddy simulation. *Aerospace Science and Technology*, 150:109219, 2024.

[22] Damiano Casalino, Gianluca Romani, Raoyang Zhang, and Hudong Chen. Lattice-boltzmann calculations of rotor aeroacoustics in transitional boundary layer regime. *Aerospace Science and Technology*, 130:107953, 2022.