Proceedings of the Canadian Society for Mechanical Engineering International Congress
Computational Fluid Dynamics Canada International Congress
CSME - CFD-SC2023
May 28-31, 2023, Sherbrooke, QC, Canada

High-Order Simulation of the Caradonna and Tung Rotor in Hover

Ramin Ghoreishi , Brian C. Vermeire

Department of Mechanical, Industrial, and Aerospace Engineering, Concordia University, Montreal, QC, Canada
*ramin.ghoreishi@concordia.ca

***Abstract*—**In this paper, we validate a previously proposed high-order method for simulating unsteady flows for a helicopter rotor in hover. To demonstrate the performance and efficiency of this strategy, three cases have been studied. The first case involves a three-dimensional circular cylinder at the onset of the shear-layer transition regime with a Reynolds number of $Re = 1000$ and Mach number of $M = 0.2$, while the second case examines the turbulent flow over a 3D SD 7003 airfoil undergoing heaving and pitching motions with a Reynolds number of $Re = 10000$ and Mach number of $M = 0.1$. These cases aim to illustrate the accuracy and efficiency of the routine when applied to transitional and turbulent flows. Finally, a hovering model of the Caradonna and Tung helicopter rotor with a tip Mach number of $M_t = 0.526$, $Re = 2.358 \times 10^6$, an angular velocity of $\Omega = 29.9237$ radian per second, and blade pitching angle of $\theta = 8^\circ$ is studied. This strategy is validated and compared against numerical and experimental reference data in terms of accuracy and computational cost, considering functional targets such as lift, drag, and thrust coefficients of the simulations. Results demonstrate that the algorithm can track regions of interest, such as boundary layers and wake regions, and yields a considerable speed-up when applied to parallel simulations. Qualitative and quantitative results showed equivalent levels of accuracy with significant speed-up when applied to parallel simulations. Hence, the proposed algorithm is an effective and accurate approach for simulating unsteady transitional and turbulent flows.

***Keywords-component*—polynomial adaptation; moving and deforming domains; dynamic load balancing; high-order**

I. Introduction

With the advent of digital computers, CFD became a vital instrument for engineers to understand, design and analyse aerospace products. CFD allows for comprehensive access compared to an experimental study and also significantly reduces design costs as it does not require a physical model or prototype and experimental apparatus. Furthermore, with the recent development of Micro Air Vehicles (MAVs) and more flexible airframes, the wind tunnel is becoming more incapable of providing the required conditions to study an object experimentally. Aerospace applications benefit from high-fidelity unsteady solution techniques, such as Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS). Recent research has led to the development of high-order unstructured spatial discretizations suitable for LES and DNS, such as the Discontinuous Galerkin (DG) [1], Spectral Volume (SV) [2], Spectral Difference (SD) [3], and Flux Reconstruction (FR) approaches [4]. It has been demonstrated previously that these methods are particularly appealing for scale-resolving LES/DNS simulations of unsteady turbulent flows [5], and that they are particularly well-suited for modern many-core hardware architectures [6, 7].

In the context of massively separated turbulent flows, such as slats, flaps, and landing gear, a-priori information about the flow-field and its related mesh resolution requirements are often not readily available. Furthermore, resolution requirements may change as the simulation evolves. Hence, a prohibitively large number of Degrees of Freedom (DOF) is required to capture the complex physics governed by the Navier–Stokes equations, making such simulations inherently computationally expensive. In practice, a relatively large number of DOF is needed only in regions where higher resolution is required for an accurate discrete approximation of the solution. Hence, the number of DOF can be adapted locally in order to minimize the overall computational cost, yet attaining almost the same level of accuracy. This can be achieved by locally increasing or decreasing the solution polynomial degree to adjust resolution and order of accuracy, a practice which is called polynomial adaptation or *p*-adaptation. However, combining polynomial adaptation with a massively parallel computation can cause overhead because *p*-adaptation tends to change the degree of solution polynomials locally, leaving the computational domain unbalanced. To maintain parallel efficiency, Dynamic

Load Balancing (DLB) techniques have been exploited [8, 9].

II. Methodology

A. Governing Equations

In this paper we solve the ALE form of the Navier-Stokes equations [10]

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{F} + \boldsymbol{u}\left(\nabla \cdot \boldsymbol{v}_g\right) = 0, \tag{1}$$

where $\boldsymbol{v}_g$ is the local mesh velocity and $\boldsymbol{u}$ is the vector of conserved variables

$$\boldsymbol{u} = \begin{bmatrix} \rho \\ \rho u_i \\ \rho E \end{bmatrix}, \tag{2}$$

where ρ is the density, ρu_i is a component of the momentum, and ρE is the total energy. The total flux $\boldsymbol{F} = \boldsymbol{F}_e - \boldsymbol{F}_v + \boldsymbol{F}_g$ is the sum of the inviscid Euler fluxes, the viscous Navier-Stokes fluxes, and the ALE fluxes, respectively. The Euler fluxes are

$$\boldsymbol{F}_{e,j}(\boldsymbol{u}) = \begin{bmatrix} \rho u_j \\ \rho u_i u_j + \delta_{ij} p \\ u_j\left(\rho E + p\right) \end{bmatrix}, \tag{3}$$

and the pressure is determined from the ideal gas law

$$p = (\gamma - 1)\rho\left(E - \frac{1}{2} u_k u_k\right), \tag{4}$$

where $\gamma = c_p / c_v$ is the ratio of specific heats, c_p is the specific heat at constant pressure, and c_v is the specific heat at constant volume. The viscous fluxes for the Navier-Stokes equations are

$$\boldsymbol{F}_{v,j}(\boldsymbol{u}, \nabla \boldsymbol{u}) = \begin{bmatrix} 0 \\ \tau_{ij} \\ -q_j - u_i \tau_{ij} \end{bmatrix}, \tag{5}$$

where the heat flux is

$$q_j = -\frac{\mu}{Pr}\frac{\partial}{\partial x_j}\left(E + \frac{p}{\rho} - \frac{1}{2} u_k u_k\right), \tag{6}$$

Pr is the Prandtl number, and μ is the dynamic viscosity. The viscous stress tensor is given by

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right). \tag{7}$$

To recover the Euler equations these viscous terms are neglected. In the case of a moving mesh the ALE fluxes are given by

$$\boldsymbol{F}_g(\boldsymbol{u}, \boldsymbol{v}_g) = -\boldsymbol{v}_g \boldsymbol{u}. \tag{8}$$

The additional source term $\boldsymbol{u}\left(\nabla \cdot \boldsymbol{v}_g\right)$ is added to Equation 2 whenever the divergence of the velocity field is not zero, which is generally the case for mesh deformation other than solid body translation. For all cases considered in this study, we take $\gamma = 1.4$ and $Pr = 0.71$.

B. Flux Reconstruction

Following the flux reconstruction approach the solution is represented by a discrete approximation on each element such that [11, 12]

$$\boldsymbol{u}(\boldsymbol{x}, t) \approx \boldsymbol{u}^h(\boldsymbol{x}, t) = \bigoplus_{i=1}^{N_e} \boldsymbol{u}_i^h(\boldsymbol{x}, t), \tag{9}$$

where $\boldsymbol{u}^h(\boldsymbol{x}, t)$ is the global piecewise continuous approximation of the solution and $\boldsymbol{u}_i^h(\boldsymbol{x}, t)$ is a continuous representation of the solution on one of N_e elements in the domain. We take the approximate solution on each element to be a polynomial nodal basis representation such that

$$\boldsymbol{u}_i^h(\boldsymbol{x}, t) = \sum_{j=1}^{N_s} \boldsymbol{u}_{i,j}(t)\phi_{s,i,j}(\boldsymbol{x}), \tag{10}$$

where $\boldsymbol{u}_{i,j}(t)$ is the value of the solution at one of N_s solution nodal basis points on a given element and $\phi_{s,i,j}(\boldsymbol{x})$ is its corresponding nodal basis function. This approach ensures the solution is continuous on each element, but allows the solution to be discontinuous on the interfaces between elements [11]. We also use a polynomial nodal basis representation to map the mesh velocities from the element mapping nodes to the interior of the element as

$$\boldsymbol{v}_{g,i}^h(\boldsymbol{x}, t) = \sum_{j=1}^{N_g} \boldsymbol{v}_{g,i,j}(t)\phi_{g,i,j}(\boldsymbol{x}), \tag{11}$$

where N_g is the number of mapping points that define the element and $\phi_{g,i,j}(\boldsymbol{x})$ is the corresponding nodal basis function of the mapping points.

Following the flux reconstruction approach [11] and its extension to simplex element types [12], the physical conservation law that must be satisfied in the discrete sense on each element is

$$\frac{\partial \boldsymbol{u}_i^h}{\partial t} + \nabla \cdot \boldsymbol{F}_i^h + \boldsymbol{\delta}_i + \boldsymbol{u}_i^h\left(\nabla \cdot \boldsymbol{v}_{g,i}^h\right) = 0, \tag{12}$$

where $\boldsymbol{F}_i^h = \boldsymbol{F}_i^h(\boldsymbol{u}_i^h, \nabla \boldsymbol{u}_i^h)$ and $\boldsymbol{\delta}_i$ is a correction field on the element that is in the same polynomial space as the solution. This correction field is analogous to the divergence of the penalty functions introduced in the original FR scheme for tensor product elements [11]. Finally, applying the conservation law at each of the solution points we obtain

$$\frac{d\boldsymbol{u}_{i,j}^h}{dt} + \left(\nabla \cdot \boldsymbol{F}_i^h\right)\Big|_{\boldsymbol{x}_{i,j}} + \boldsymbol{\delta}_{i,j} + \boldsymbol{u}_{i,j}^h\left(\nabla \cdot \boldsymbol{v}_{g,i}^h\right)\Big|_{\boldsymbol{x}_{i,j}} = 0, \tag{13}$$

where $\boldsymbol{x}_{i,j}$ is the corresponding solution point location and, following the FR formulation

$$\boldsymbol{\delta}_{i,j} = \frac{1}{|\Omega_i|}\sum_{f \in S}\sum_{l} \alpha_{i,j,f,l}[\hat{\boldsymbol{F}}]_{i,f,l} S_f, \tag{14}$$

where $|\Omega_i|$ is the element volume, f is one of the number of faces on the element surface S, l is one of the flux points, $\alpha_{i,j,f,l}$ is a constant lifting coefficient, $[\hat{\boldsymbol{F}}]_{i,f,l}$ is the difference between a common Riemann flux at the flux point and the

value of the internal flux, and S_f is the area of the face. Depending on the specification of these lifting coefficients, a number of different energy stable schemes can be obtained for general element types, including the Spectral Difference (SD), Spectral Volume (SV), and Discontinuous Galerkin (DG) methods. In this study we use lifting coefficients based on the nodal basis functions to recover the DG method [11]. For more detail on implementation of FR, please refer to [9, 13].

C. Vorticity-Based Adaptation

This paper explores the utility of a novel non-dimensional vorticity-based indicator for p-adaptation when solving the unsteady Navier-Stokes equations. We begin by computing the maximum non-dimensional vorticity magnitude at any solution point within an element according to

$$\kappa_i = \max_{1 \leq j \leq N_s} \overline{\omega}_{i,j}, \tag{15}$$

where $\overline{\omega}_{i,j}$ is the non-dimensionalized counterpart of vorticity magnitude

$$\overline{\omega}_{i,j} = \frac{|\omega_{i,j}| \Delta_{max,i}}{U_\infty}, \tag{16}$$

where $\omega_{i,j}$ is the vorticity at solution point j on element i, $\Delta_{max,i}$ is the maximum mesh dimension of the element, N_s is the number of solution points on the element, and U_∞ is the free-stream velocity. We then define the solution polynomial degree on any element using Algorithm 1 mentioned in [13], where ϵ is a constant resolution threshold, ϑ is a constant that controls the relative threshold for different solution polynomial degrees and k_m is the maximum polynomial degree. Following this approach, elements with large vorticity magnitudes relative to the effective mesh resolution are adapted to higher-degree polynomials, increasing their resolving power.

D. Dynamic Load Balancing

In this study, a load balancing technique is employed using the repartitioning routine of the ParMETIS Parallel Graph Partitioning and Sparse Matrix Ordering library developed by Karypis et al. [14] to circumvent the overhead due to an imbalanced mesh. Using this approach, the mesh is considered as a weighted graph, where elements and the computational burden are its vertices and weight accordingly. The computational burden varies based on the element type, solution polynomial degree, number of dimensions, and number of conserved variables. The compute time for a single element is used as the reference to define the element weight.

III. Numerical Results

A. 3D Dynamic Stall of a SD 7003 Airfoil

A three-dimensional computational domain, with its origin at the leading edge, consisting of 56,000 structured hexahedral elements with high-order solution polynomials in each element, moderately refined near the trailing and leading edges, was used. Riemann invariant boundary conditions were applied at the far-field, periodic boundary conditions were applied in the span-wise direction, and a no-slip adiabatic wall boundary condition was applied at the surface of the airfoil. The domain extends to $19c$ upstream, $20c$ above, below, and downstream of the airfoil, and $0.4c$ in the span-wise, where c is the airfoil chord. Quadratically curved elements were used at the boundaries to match the airfoil geometry. The mesh was initially partitioned over 120 processors.

The translation of the coordinates of the center of oscillation and the pitching function are defined as

$$\begin{aligned} x(t) &= 0, \\ y(t) &= A\cos\left(2\pi f_e t\right), \\ \theta(t) &= \theta_0 + \theta_e \cos(2\pi f_e t + \phi_e), \end{aligned} \tag{17}$$

where A is the heaving oscillating amplitude, which is non dimensionalized by the airfoil chord length, f_e is the excitation frequency which is defined in terms of the reduced frequency $f_K = \pi f_e c/U_\infty$, θ_0 is the mean pitch angle value, θ_e is the pitch amplitude, and ϕ_e is the phase shift between the heaving and pitching motions. Simulation were carried out at $Re = 10000$, $M = 0.1$, $A/c = 0.5$, $f_K = 0.25$ resulting in a period of oscillation of $T_e = 4\pi/U_\infty$, $\theta_0 = 8°$, $\theta_e = -8.42°$, and $\phi_e = 90°$ in order to match the experimental study of Baik et al. [15] and Ol et al. [16]. The total simulation time is set to $8T_e$. This corresponds to $\approx 125t_c$, where t_c is the time required for the flow to traverse one airfoil chord. Time integration was with the classical $RK_{4,4}$ scheme, with a non-dimensional time step of $\Delta t^* = 6.14 \times 10^{-5}$, which is a fraction of the oscillation period. A three-level adaptive simulation was performed to verify the utility of the adaptation algorithm and load balancing, which was achieved via the adaptive repartitioning algorithm in ParMETIS [14]. The adaptation parameters K_m, ϵ, and $\vartheta_{1:5}$ are set to 3, 2.5×10^{-1}, and $[0.25, 0.25, 1]$, respectively. The adaptation and the DLB routines are called every 100 and 1000 iterations respectively.

Figure 1 shows the non-dimensional x-component of the velocity profile averaged through the span and four last cycles for the instant of $2T_e/4$ at different chord locations. Comparing these plots confirms that the three-level adaptive simulation correlates well with the experiment results. Table I shows the simulation time per one iteration T_{itr} and the speed-up factor $S_A = T_U/T_A$ for uniform and the three-level adaptive simulations. The speed-up factor is 2.73 which makes the adaptive simulation even faster than a uniform $K = 2$ simulation. Table II shows the scalability of the parallel algorithm for different numbers of cores, where N_n is the number of nodes, C is the number of cores, $S_N = T_N/T_C$ is the speed-up factor based on a one node simulation, $E_s = S_N/C$ is the efficiency, T_N is the compute time of a parallel simulation on one node, and T_C is the compute time of a parallel simulation on C cores. These results verify the scalability of the algorithm and the capability of the DLB to distribute uniform computation load among processors.

B. Turbulent Flow Over a 3D Circular Cylinder

A three-dimensional computational domain, with its origin located at the centre of the cylinder, containing a total of

TABLE. I. Computation time of the dynamic load-balanced adaptive algorithm for the heaving and pitching SD 7003 airfoil case.

Degree	Number of Nodes	Number of Cores	T_{itr}	S_A
1	1	40	3.832×10^2	-
2			8.629×10^2	-
3			1.750×10^3	-
$1-3$			6.416×10^2	**2.73**

TABLE. II. Scalability metrics of the dynamic load-balanced adaptive algorithm for the heaving and pitching SD 7003 airfoil case.

Number of nodes	Number of cores	S_N	E_s
$N_n = 1$	40	1.000	1.000
$N_n = 2$	80	1.979	0.989
$N_n = 4$	160	3.946	0.986
$N_n = 8$	320	7.669	0.959
$N_n = 16$	640	13.722	0.861
$N_n = 32$	1280	24.258	0.758

78176 hexahedral elements with 14 elements in the span-wise direction was used with Riemann invariant boundary conditions at the far-field, periodic boundary condition in the span-wise direction, and a no-slip adiabatic wall boundary condition at the surface of the cylinder. The domain extends to $40D$ above, below, and upstream of the cylinder, $80D$ downstream, and $2\pi D$ in the span-wise direction, where D is the cylinder diameter. Initially partitioned over 320 processors, the mesh is moderately refined near the wall and in the wake region, and uses quadratically curved elements at the boundaries to match the cylinder geometry.

Simulations were run at Mach number $M = 0.2$, for a total of $400t_c$, where $t_c = D/U_\infty$ is the time required for the flow to traverse one cylinder diameter. Time integration was carried out with a fourth-order twelve-stage optimized Runge-Kutta scheme [17], and the non-dimensional time step is set to $\Delta t^* = \Delta t U_\infty / D = 8.0 \times 10^{-4}$. By having more stages, this Runge-Kutta scheme allows for a larger time step size which relatively reduces the computational cost. A total of six simulations were carried out to verify the utility of the dynamically load-balanced adaptation algorithm, including five uniform simulations with solution polynomials of degree $K = 1$ to $K = 5$ and a five-level adaptive simulation. The adaptation parameters K_m, ϵ, and $\vartheta_{1:5}$ are set to 5, 1.25×10^{-1}, and $[0.25, 0.25, 1, 4, 16]$ respectively. The adaptation and the DLB routines are called every 100 and 1000 iterations, respectively. Simulations were carried out on eight nodes of the Niagara cluster, each node consists of 2 sockets with 20 Intel Skylake cores (2.4 GHz, AVX512), for a total of 40 cores per node, and a total of 202 GB of RAM.

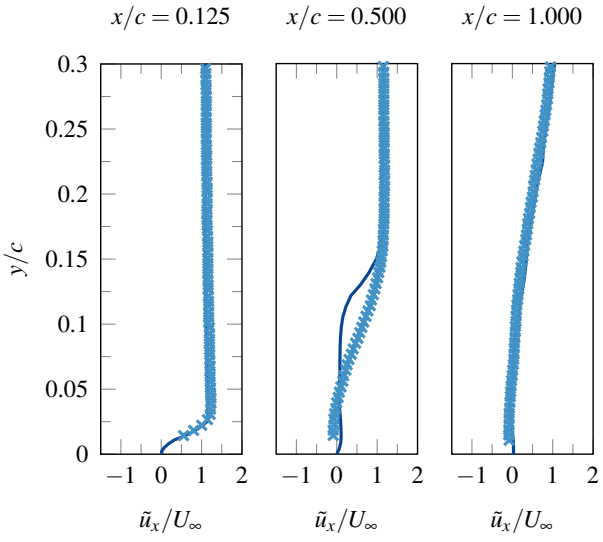

Figure. 1. Phase averaged x-component of the velocity profile at different chord locations at time instant of $t = 2T_e/4$ (solid lines correspond to this study, and the crosses correspond to the experiment [15]) for the SD 7003 airfoil case.

TABLE. III. Numerical values of $\bar{C_D}$, St, T_{itr} and S_A averaged over the 100 convective times for the circular cylinder case.

Degree	C_D	St	T_{itr}	S_A
1	1.195	0.204	7.42×10^2	-
2	1.083	0.208	2.12×10^3	-
3	1.008	0.210	4.74×10^3	-
4	1.018	0.205	9.75×10^3	-
5	1.014	0.205	1.90×10^4	-
$1-5$	1.055	0.209	4.10×10^3	**4.63**
Pereira	1.064	-	-	-
Zhao	1.092	0.202	-	-

Figure 2 shows the contours of the Q-criterion colored by the velocity magnitude, the polynomial degree distribution, and the repartitioned mesh for the five-level adaptive simulation $(K = 1-5)$ for the instant of maximal lift. This verifies that the adaptation algorithm successfully tracks elements with large vorticity magnitude relative to their size. We observe that a relatively fewer number of high-order elements are allocated to processors near the cylinder and in the wake region, while processors located farther away from the cylinder and the wake contain a larger number of lower-order elements. This illustrates the ability of the DLB algorithm to maintain the load balance. Table III reports the quantitative values of $\bar{C_D}$, St, T_{itr}, and S_A averaged during $100t_c$ and compares them against the reference data. These results quantitatively show agreement within 4.02% in the $\bar{C_D}$ value between the adaptive and uniform $K = 5$ simulations, while the former is 4.63 times faster.

C. Caradonna and Tung Rotor in Hover

The geometry is taken from the Caradonna and Tung experimental study [18] with an aspect ratio of $AR = 6$, non-dimensional chord length of $c = 1$, hub radius of $R_h = 0.5c$, and blade root location of $R_r = c$. A three-dimensional computational domain containing a total of 6.945×10^6 hybrid elements was used. Using a C-topology in the stream-wise direction, the domain extends to $30c$ upstream, $60c$ downstream, and a radius of $30c$ in the span-wise direction, where c is

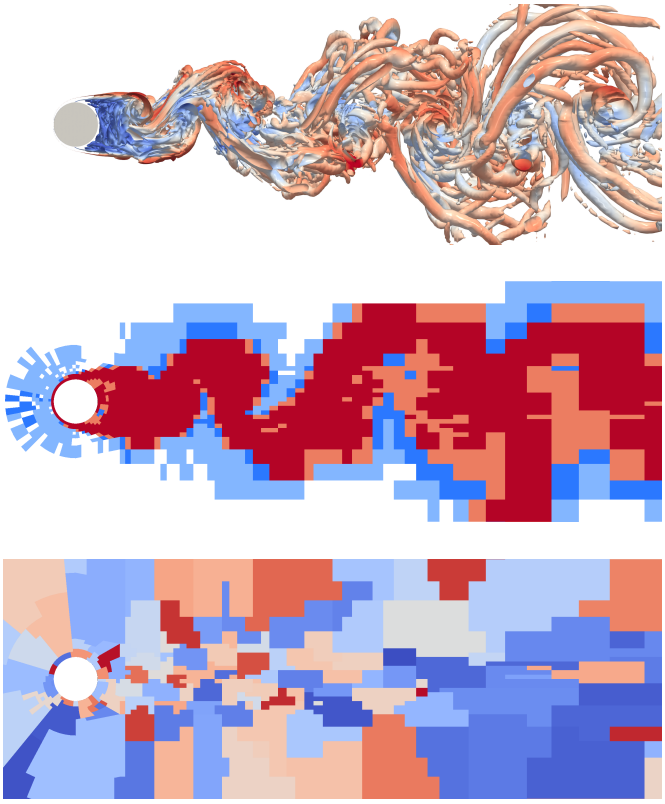

Figure. 2. Contours of Q-criterion (top), polynomial distribution (middle), and dynamically balanced mesh partition (bottom) for the adaptive computation ($K = 1-5$) at the times corresponding to maximal lift for the the circular cylinder case.

the blade chord length. Only one blade is simulated with a rotational periodic boundary conditions at the symmetry plane. The mesh is moderately refined near the blade, rotor hub, and in the wake region. Incompressible Multiple Reference Frame (MRF) in OpenFOAM was used with pressure inlet and outlet, slip wall at the far-field and the propeller hub, and no-slip adiabatic wall boundary condition at the surface of the blade. Cyclic Arbitrary Mesh Interface (AMI) was used on the symmetry plane. The domain was divided into two rotational and static regions, where the former rotates with the blade angular velocity.

The simulation was run with tip Mach number of $M_{tip} = 0.526$, tip Reynolds number of $Re_{tip} = 2.358 \times 10^6$, angular velocity of $\Omega = 29.9237$ radian per second, blade pitching angle of $\theta = 8°$, and ambient density of $\rho = 1.2389$. $k-\omega$ turbulence model was used. The simulation was assumed steady state with 2000 iterations.

Figure 3 shows the pressure coefficient distribution at different radial locations. The discrepancy near the blade root is due to the blade attachment used in the experimental study. Figure 4 shows the contours of Q-criterion colored by the velocity magnitude. As mentioned earlier, the presence of a root vortex due to the lack of blade attachment is responsible for the discrepancy in pressure distribution at the radial position of

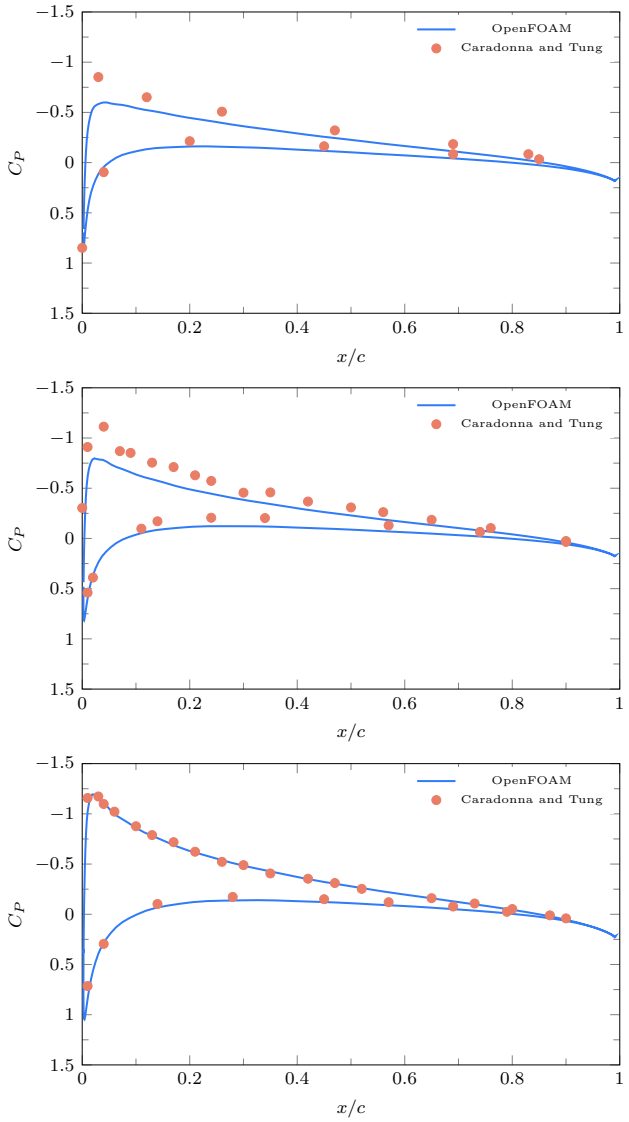

Figure. 3. Pressure coefficient distribution at different radial locations of $r/R = 0.5$, $r/R = 0.80$, and $r/R = 0.89$ from top to bottom for the Caradonna and Tung rotor in hover.

$r/R = 0.5$. Approximately $400°$ of a tip vortex age is resolved in the current simulation. Figure 5 shows the tip vortex descent rates, which is accurately predicted by the current simulation. Future work will explore high-fidelity simulations of this case.

IV. Conclusion

Results demonstrated that the algorithm is capable of tracking salient turbulent features. Qualitative results showed equivalent levels of accuracy between the adaptive and high-order solutions, with a significant reduction in simulation time. Results from the adaptive simulations also showed good quantitative agreement with parallel high-order simulations and reference numerical, experimental, and analytical results.

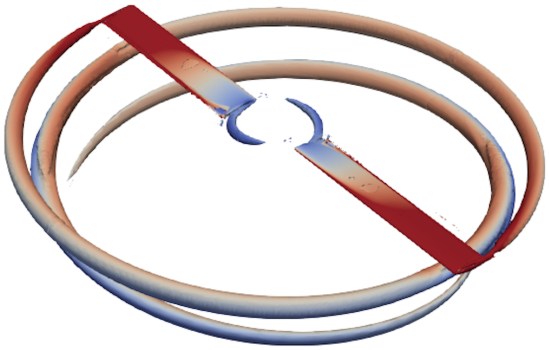

Figure. 4. Contours of Q-criterion coloured by velocity magnitude for the Caradonna and Tung rotor in hover.

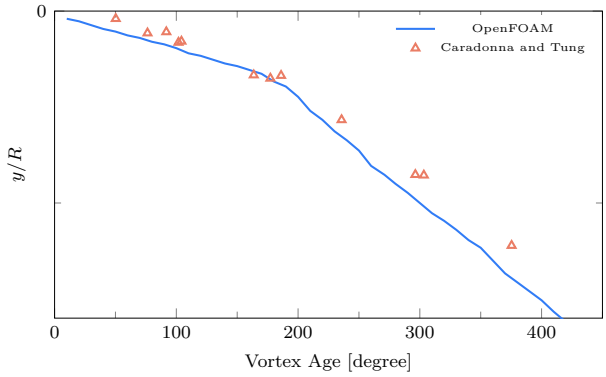

Figure. 5. Tip vortex descent for the Caradonna and Tung rotor in hover.

References

[1] B. Cockburn, S. Y. Lin, and C. W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems. 1988.

[2] Z. J. Wang. Spectral (Finite) Volume Method for Conservation Laws on Unstructured Grids. Basic Formulation:. *Journal of Computational Physics*, 178(1):210–251, May 2002.

[3] W. Liu, M. Vinokur, and Z. J. Wang. Spectral difference method for unstructured grids I: Basic formulation. *Journal of Computational Physics*, 216(2):780–801, 2006.

[4] H. T. Huynh. A Reconstruction Approach to High-Order Schemes Including discontinuous Galerkin for Diffusion. In *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, January 2009.

[5] B. C. Vermeire B. C., S. Nadarajah, and P. G. Tucker. Implicit large eddy simulation using the high-order correction procedure via reconstruction scheme. *International Journal for Numerical Methods in Fluids*, 82(5):231–260, 2015.

[6] B. J. Zimmerman, Z. J. Wang, and M. R. Visbal. High-order spectral difference: verification and acceleration using GPU computing. In *21st AIAA Computational Fluid Dynamics Conference*, page 2941, 2013.

[7] B. C. Vermeire, F. D. Witherden, and P. E. Vincent. On the utility of GPU accelerated high-order methods for unsteady flow simulations: A comparison with industry-standard tools. *Journal of Computational Physics*, 334:497–521, 2017.

[8] F. Bassi, A. Colombo, A. Crivellini, K. J. Fidkowski, M. Franciolini, A. Ghidoni, and G. Noventa. Entropy-adjoint p-adaptive discontinuous Galerkin method for the under-resolved simulation of turbulent flows. *AIAA Journal*, 58(9):3963–3977, 2020.

[9] R. Ghoreishi and B. C. Vermeire. Vorticity-based polynomial adaptation for moving and deforming domains. *Computers & Fluids*, 231:105160, 2021.

[10] K. Liu, Y. Lu, and C. You. High-order ALE method for the Navier–Stokes equations on a moving hybrid unstructured mesh using flux reconstruction method. *International Journal of Computational Fluid Dynamics*, 27(6-7):251–267, 2013.

[11] H. T. Huynh. A Flux Reconstruction Approach to High-Order Schemes Including discontinuous Galerkin Methods. In *18th AIAA Computational Fluid Dynamics Conference*, volume AIAA 2007-4079. American Institute of Aeronautics and Astronautics, 2007.

[12] H. Gao and Z. J. Wang. A High-order Lifting Collocation Penalty Formulation for the Navier-Stokes Equations on 2d Mixed Grids. In *19th AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, June 2009.

[13] Seyedramin Ghoreshilangrodi. *Vorticity-Based Polynomial Adaptation for Unsteady Flows*. PhD thesis, Concordia University, 2022.

[14] G. Karypis, K. Schloegel, and V. Kumar. Parmetis: Parallel graph partitioning and sparse matrix ordering library. 1997.

[15] Y. S. Baik, J. Rausch, L. Bernal, and M. Ol. Experimental investigation of pitching and plunging airfoils at Reynolds number between 1x10ˆ 4 and 6x10ˆ 4. In *39th AIAA fluid dynamics conference*, page 4030, 2009.

[16] M. V. Ol, L. Bernal, C. Kang, and W. Shyy. Shallow and deep dynamic stall for flapping low Reynolds number airfoils. In *Animal Locomotion*, pages 321–339. Springer, 2010.

[17] S. Hedayati Nasab, C. A. Pereira, and B. C. Vermeire. Optimal runge-kutta stability polynomials for multidimensional high-order methods. *Journal of Scientific Computing*, 89(1):1–30, 2021.

[18] Francis X Caradonna and Chee Tung. Experimental and analytical studies of a model helicopter rotor in hover. In *European rotorcraft and powered lift aircraft forum*, number A-8332, 1981.