

Combined Feature-Driven Richardson-Based Adaptive Mesh Refinement for Unsteady Vortical Flows

S. J. Kamkar,* A. M. Wissink,† and V. Sankaran‡

U.S. Army Aeroflightdynamics Directorate, Moffett Field, California 94035

and

A. Jameson§

Stanford University, Stanford, California 94305

DOI: 10.2514/1.J051679

An adaptive mesh refinement strategy that couples feature detection with local error estimation is presented. The strategy first selects vortical regions for refinement using feature detection, and then terminates refinement when an acceptable error level has been reached. The feature detection scheme uses a local normalization of the Q -criterion, which allows it to properly identify regions of swirling flow without requiring case-specific tuning. The error estimator relies upon a Richardson extrapolation-like procedure to compute local solution error by comparing the solution on different grid levels. Validation of the proposed approach is carried out using a theoretical advecting vortex and two practical cases, namely, tip vortices from a NACA 0015 wing and the wake structure of a quarter-scale V22 rotor in hover.

I. Introduction

ADAPTIVE mesh refinement (AMR) is a useful approach for computational fluid dynamic (CFD) simulations that contain isolated relevant features like shocks or tip vortices, which are small in size with respect to the surface geometry but have a profound impact on the resulting flowfield. The fixed-wing aerodynamics community has used adaptively refined grids for high-fidelity solutions of transonic and supersonic flows [1–3]. In addition to shocks, trailing tip vortices occur in fixed-wing flight but are of greater interest for rotorcraft flight, in which the vortices shed from the blade tips can dominate the unsteady dynamics of the turbulent wake and can significantly impact vehicle performance, vibration, and noise. Accurate wake resolution can therefore lead to improvements in the prediction of rotor performance metrics [4], such as the *figure of merit*, a nondimensional parameter that represents the efficiency of a rotor in hover. Additionally, wake modeling is important because rotorcraft fly in their own wake, which may become entrained during hover and interact with the fuselage during forward flight [5]. However, despite the need to accurately resolve tip vortices, the rotorcraft community has not exercised AMR to the degree that the fixed-wing community has to model shocks, mainly due to the complexities in the unsteadiness of rotary-wing problems. Similar to shock modeling for fixed-wing cases, the spatial scales of trailing vortices are relatively smaller than the chord distance, thereby requiring relatively fine meshes and making the use of uniformly fine grids largely impractical [5]. Therefore, in this work, we develop an unsteady AMR strategy that targets vortical features with the goal of enhancing the resolution for both fixed- and rotary-wing problems.

In particular for rotorcraft, complexities involving the inherently unsteady flowfield and the relative motion between the rotor and the fuselage make the implementation of efficient adaptive schemes

especially challenging. Moreover, high-fidelity rotorcraft CFD are highly unsteady and require time-accurate simulations. Adjoint-based AMR has shown promise for steady CFD applications [1,6,7], but time accurate solutions require the adjoint problem to be fully solved backwards in time, which is intractable for large-scale rotorcraft simulations that can involve 10^5 to 10^6 time-steps. Therefore, in this work we seek an alternative error-based refinement approach that specifically targets the vortex cores in a local manner without having to solve the full adjoint. Our approach first identifies the vortex cores using feature detection, and then the level of mesh resolution is set according to local solution error.

In our earlier work [8,9], we developed four locally normalized methods that appropriately guided the AMR process based upon popular methods by the feature detection community [10–13]. A major goal of this development was to eliminate the parameter tuning that is required for common (dimensional) approaches, e.g., vorticity-based. Whereas dimensional approaches require highly tuned thresholds to select regions for refinement, the normalized schemes are able to mark regions with key vortical features using a fixed threshold, regardless of vortical strength, size, and/or resolution. However, while these nondimensional schemes effectively deal with the issue of identifying regions for refinement, the degree of mesh resolution still needs to be specified by the user. To reduce user dependency and improve computational efficiency, in this study, we examine a method of automatically setting the degree of mesh resolution by using the solution error as a guide.

The objective of the current paper is to develop a solution-based error estimator that can be coupled with the nondimensional feature-based AMR. The error estimator is used to limit the amount of applied grid resolution so that additional refinement will be halted once the solution error is sufficiently low. Similar to a global functional, which is commonly used by adjoint approaches, our approach uses a local functional that is based upon quantities of interest to vortical motion. In effect, we aim to reduce the local error estimate through additional mesh refinement. Moreover, the Richardson estimator is quite practical because it is relatively simple to implement and efficient to execute.

The remainder of the paper is organized as follows. A description of the adaptive overset grid-based CFD approach used for the present work is presented first in Sec. II. Thereafter, Sec. III briefly reviews the nondimensional feature-based approach, which is used to identify candidate regions for mesh refinement. Section IV offers a theoretical analysis of the Richardson error estimator, along with spatial accuracy validation tests. Then, the coupled AMR strategy that combines the feature identification with the Richardson

Presented at the 49th AIAA Aerospacess Conference, Orlando, FL, January 4–7, 2011; received 14 October 2011; revision received 30 April 2012; accepted for publication 7 May 2012. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. Copies of this paper may be made for personal or internal use, on condition that the copier pay the \$10.00 per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923; include the code 0001-1452/12 and \$10.00 in correspondence with the CCC.

*Post-Doctoral Researcher; skamkar@merlin.arc.nasa.gov. AIAA Member.

†Aerospace Engineer; andrew.m.wissink@us.army.mil. AIAA Member.

‡Aerospace Engineer; vsankaran@merlin.arc.nasa.gov. AIAA Member.

§Professor; jameson@baboon.stanford.edu. AIAA Fellow.

estimator is presented in Sec. V. Section VI applies the combined strategy to an unsteady advecting vortex, a lifting NACA 0015 wing, and the tilt rotor aeroacoustic model (TRAM) in hover. Lastly, Sec. VII summarizes the major findings and discusses areas for additional work.

II. Computational Infrastructure

This work is developed as part of the CFD/computational structural dynamics (CSD) rotary-wing analysis package Helios [14], which employs a dual-mesh overset-based flow solution paradigm. It comprises an unstructured solver operating on a near-body prismatic/tetrahedral grid and a high-order accurate adaptive solver operating on an off-body Cartesian domain [15,16]. The two solvers are coupled using an overset domain connectivity algorithm that applies implicit hole-cutting and adds fringe regions to allow for relative grid motion [14]. The near-body grids can rotate and deform with the rotating grid system, while the off-body grids remain stationary (Fig. 1). Further description of Helios's main components are given elsewhere [14].

The purpose of this mixed (near/off-body) meshing strategy is to apply unstructured grids near the surface to resolve complex geometry and boundary layer effects and block-structured Cartesian off-body grids to resolve the far-field wake using a combination of high-order numerics and AMR. The OVERFLOW [17,18] code implements a similar paradigm, but the near-body grid system uses a structured curvilinear system. The work in the present paper is largely motivated by the need to automate the off-body mesh refinement process in Helios by specifically targeting vortical features in the wake.

A. Off-Body Solver Overview

SAMARC [19] is the off-body Cartesian solver that combines structured adaptive mesh refinement application infrastructure (SAMRAI) [20–22] with ARC3DC [23], a block-structured

Cartesian inviscid Euler solver with high-order spatial and temporal discretizations. SAMRAI manages the generation of meshes around flow features and geometries and handles the mesh partitioning, domain decomposition, and the message-passing interface (MPI)-based parallel communication on distributed memory computer systems. The off-body domain comprises a multilevel Cartesian grid system stored as a union of rectangular blocks in the approach proposed by Berger et al. [24,25]. New-level construction occurs from coarsest to finest, and there is an agglomeration operation that groups regions of the same level together to create block-based refinement. Cells tagged for refinement will be organized in such a manner as to form rectangular blocks that contain finer grid levels above their coarser equivalents (Fig. 2). The use of the block-structured paradigm along with algorithms that preserve computational efficiency in the SAMRAI library make adaptive solutions relatively cheap. Considering a comparable unstructured grid, the Cartesian-based block-structured grid storage requirement is considerably smaller, which allows for fast and efficient refinement on large parallel machines, even for highly unsteady problems with refinement occurring at every time-step [26,27]. Additionally, the cost of the structured Cartesian grid solver is more than an order of magnitude less on a per-node basis compared to unstructured solvers. Additional details regarding the time-stepping algorithm and multilevel solver approach are given in the next section.

B. Block Solution Procedure

ARC3DC is a higher-order, finite-difference code that solves the Euler equations

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} + \frac{\partial \mathbf{h}}{\partial z} = 0 \quad (1)$$

where the state and flux vectors are

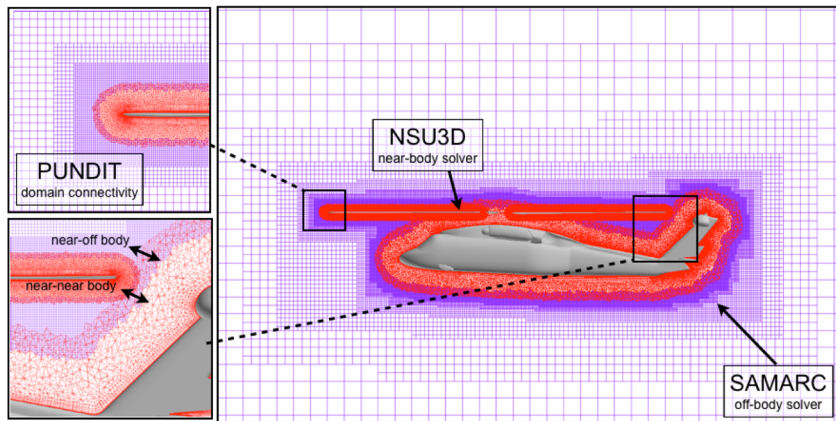


Fig. 1 The Helios dual-mesh paradigm with unstructured near-body grids and the block-structured Cartesian grids in the off-body.

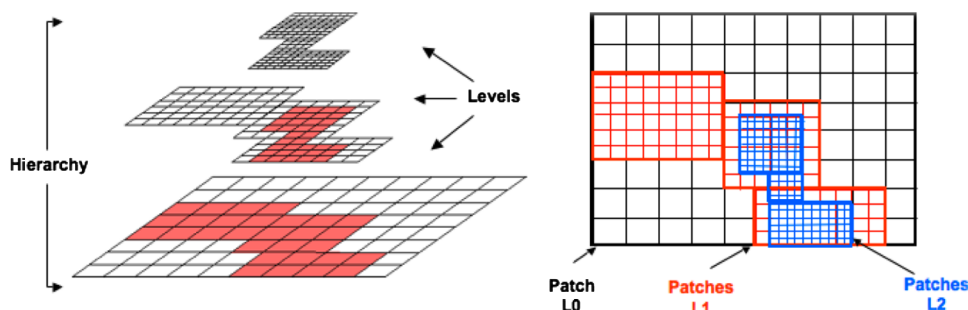


Fig. 2 Block-structured AMR grid containing a hierarchy of nested refinement levels.

$$\mathbf{u} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho uH \end{pmatrix}$$

$$\mathbf{g} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho vH \end{pmatrix}, \quad \mathbf{h} = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho wH \end{pmatrix}$$

where ρ is density, p is pressure, E is the total energy, H is the stagnation enthalpy, and u , v , and w represent the x -, y -, and z -velocities. Solving the Euler equations is acceptable for rotorcraft flow in the off-body domain, where the average cell-Reynolds numbers are large and viscous effects are small.

The full computational domain, which may comprise thousands of separate blocks, is solved in parallel on a block-by-block basis. During runtime, SAMRAI supplies the individual blocks to ARC3DC, which has been specifically designed to capitalize on these lightweight, Cartesian-based grids. The block-solution procedure facilitates the implementation of ARC3DC's higher-order explicit spatial and temporal discretizations. Fifth-order spatial and third-order temporal differencing is applied and both formulations are discussed below.

For demonstrative purposes, a one-dimensional, semi-discrete form of the Euler equations [Eq. (1)] using central differencing can be written as

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\mathbf{f}_{j+\frac{1}{2}} - \mathbf{f}_{j-\frac{1}{2}}}{\Delta x} = 0 \quad (2)$$

where Δx represents the spacing in the x -dimension and the $j \pm \frac{1}{2}$ locations represent the fluxes at the half-cell. For a sixth-order accurate scheme, the flux at the cell interface is

$$\mathbf{f}_{j+\frac{1}{2}}^{6C} = \frac{1}{60} (\mathbf{f}_{j+3} - 8\mathbf{f}_{j+2} + 37\mathbf{f}_{j+1} + 37\mathbf{f}_j - 8\mathbf{f}_{j-1} + \mathbf{f}_{j-2}) \quad (3)$$

ARC3DC uses a central-plus-artificial-dissipation that was first proposed by Jameson [28]. Specifically, the discretization scheme [29] relies upon a scalar-based formulation in which the dissipation is determined by the eigenvalues of the flux Jacobian, $A_{j+\frac{1}{2}}$. Although additional coefficients may be employed to achieve a higher-order accurate formulation, in its simplest form it can be written as

$$\mathbf{d}_{j+\frac{1}{2}} = \frac{\varepsilon}{2} |\sigma(A_{j+\frac{1}{2}})| \Delta \mathbf{u}_{j+\frac{1}{2}} \quad (4)$$

Using fifth-order dissipation (fifth-order accurate), the overall discretization scheme in ARC3DC becomes

$$\mathbf{f}_{j+\frac{1}{2}} = \mathbf{f}_{j+\frac{1}{2}}^{6C} - \frac{\varepsilon}{60} |\sigma(A_{j+\frac{1}{2}})| (\mathbf{u}_{j+3} - 5\mathbf{u}_{j+2} + 10\mathbf{u}_{j+1} - 10\mathbf{u}_j + 5\mathbf{u}_{j-1} - \mathbf{u}_{j-2}) \quad (5)$$

For unsteady flows, high spatial accuracy is essential, but it is of limited value without also maintaining temporal accuracy. For this purpose, Runge–Kutta (RK) time-integration methods [30–32] are particularly suitable as they exhibit high accuracy and efficiency and have encountered widespread success [33–35]. Thus, ARC3DC uses the third-order explicit Runge–Kutta method (RK3) due to Wray [36]. Writing the discretized Euler equation as a representative scalar ordinary differential equation (ODE), $du/dt = f(u, t)$, where the time-rate change of u is dependent upon both space and time, the explicit RK3 solution procedure is

$$u^* = u^N + \frac{\Delta t}{4} f[u^N, N\Delta t], \quad u^{**} = u^* + \frac{8\Delta t}{15} f[u^*, N\Delta t]$$

$$u^{***} = u^* + \frac{5\Delta t}{12} f[u^{**}, (N + \frac{8}{15})\Delta t]$$

$$u^{N+1} = u^* + \frac{3\Delta t}{4} f[u^{***}, (N + \frac{2}{3})\Delta t]$$

Large rotorcraft simulations also benefit from the minimal storage cost, as the RK3 scheme requires only one temporary solution vector during a solution update.

C. Adaptive Multi-Level Implementation

This section describes the RK3 scheme implemented on the multilevel Cartesian structured AMR grid system. For each RK substep, we first update level boundaries using interpolated values from the coarser level, then perform the function evaluation on the level to compute a solution, and finally inject that solution into the coarser level wherever there is overlap. A more detailed explanation follows.

Consider the first substep in the RK3 algorithm, in which u_m^* is computed using u_m^N . The m subscript is used to designate level number, such that $m = 0$ designates the coarsest level and each additional integer represents a new mesh level with resolution that halves the grid spacing. The multilevel implementation of this substep would proceed as follows:

$$u_0^* = u_0^N + \frac{\Delta t}{4} f[u_0^N, N\Delta t]$$

for $m = 1, \text{ finest}$ **do**

$$u_m^N \leftarrow u_{m-1}^N \text{ at boundaries}$$

$$u_m^* = u_m^N + \frac{\Delta t}{4} f[u_m^N, N\Delta t]$$

$$u_m^* \rightarrow u_{m-1}^* \text{ in overlap regions}$$

end for

The first step computes u^* on all blocks of the coarsest level. Next, the algorithm loops through the levels from coarsest to finest. It first sets boundary conditions for u_m^N using interpolated values from the coarser level u_{m-1}^N (Fig. 3a). Then it applies the functional evaluation $f(u_m^N)$ to compute u_m^* . Finally, it injects the new u_m^* solution into u_{m-1}^* to overwrite the coarser level solution with the more accurate finer solution wherever grid levels overlap (Fig. 3b). Parallel implementation of this algorithm is straightforward and is discussed in more detail elsewhere [19].

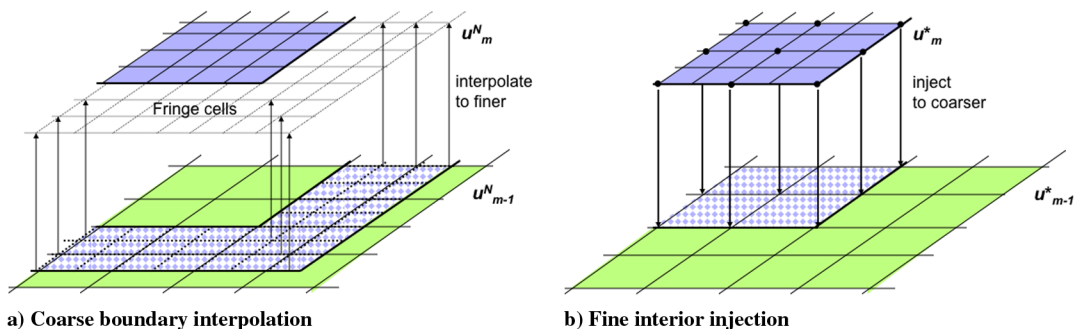


Fig. 3 Coarse-fine operations for multilevel RK3 implementation: a) interpolation of u^N from coarser level to boundary fringes on finer level and b) injection of u^* from finer level to overlapped points on coarser level.

The hierarchical block-structured storage maintains the solution across all levels, which simplifies the computation of the Richardson-based error estimate. Before the final step of the algorithm, the $u_m^* \rightarrow u_{m-1}^*$ injection, a solution error between levels $e_m = f(u_m^*, u_{m-1}^*)$ may be computed. Additional details on how the error is computed and combines with the feature detection to drive refinement are given in a subsequent section.

III. Feature Detection

In our earlier work [37], a family of feature-detection methods was proposed that employed a nondimensionalization based on local shear-strain rates, which emphasized generality and allowed for automation. Four methods were developed based upon the Q -criterion [10], λ_2 [11], λ_{ci} [12], and λ_+ [13]. The four methods demonstrated similar and desirable tagging behavior, as each identified nearly identical regions of vortical flows using a fixed threshold of unity. In this work, the nondimensional Q method is used for all the tests and its formulation is briefly discussed as follows.

The Q -criterion [10] is generally written as $Q = \frac{1}{2}(\|\Omega\|^2 - \|S\|^2)$, which represents the difference between the strain (S) and rotation (Ω) components of the velocity gradient tensor. While Q measures the relative rotational strength, it is still dependent upon the characteristic length and velocity scales of the specific problem. Therefore, to yield a suitable nondimensional form, we divide by $\|S\|^2$,

$$\tilde{Q} = \frac{1}{2} \left(\frac{\|\Omega\|^2}{\|S\|^2} - 1 \right) \quad (6)$$

Using the shear-strain rate tensor is a particularly appropriate normalization for two reasons: it has the same units as Q , and its functional behavior is well behaved with respect to Q . The shear-strain value is zero at the center of an ideal vortex, and it is commensurate with Q near the radial extent of the vortex.

Using a threshold of unity, i.e., marking regions for refinement when $\tilde{Q} > 1$, automatically detects the entire core region without tagging the surrounding area outside of the core. Moreover, using a threshold value of unity remains satisfactory, regardless of vortical strength, size, or type [9]. Conditions may arise wherein $\|S\| \rightarrow 0$, thereby leaving \tilde{Q} unbounded. However, at the center of a vortex, where this is likely to occur, an infinite \tilde{Q} value is acceptable because the core will be routinely tagged as it is greater than unity. Additionally, near-zero strain rates may occur in freestream regions that contain minor rotational perturbations. For these cases, a small noise filter can be applied that does not affect robustness or limit overall accuracy [37].

IV. Error Estimation

The normalized feature detection approach is highly automated, but it cannot indicate when sufficient mesh resolution has been attained. In this section, we formulate a Richardson estimator, suitable for both unsteady and steady flows, which can serve as a termination criterion for refinement.

A. Local Truncation Error

Before constructing the error estimator, we first examine the local truncation error (LTE) for unsteady flows for two reasons. First, the LTE serves as a verification of the spatial and temporal order-of-accuracy convergence rates of the numerical discretization scheme. Second, it represents the best attainable spatial error convergence and can therefore be used as a performance benchmark for the Richardson estimator.

The general formulation of the LTE is derived first for the one-dimensional (1-D) wave equation for which an exact solution is available. Following this, the LTE of an unsteady advecting vortex is examined. In the simple 1-D linear wave equation,

$$\frac{\partial u}{\partial t} + \lambda \frac{\partial u}{\partial x} = 0 \quad (7)$$

u is a general scalar that represents the exact solution, λ represents the wave-speed, and t and x represent the temporal and spatial dimensions, respectively. To solve Eq. (7) numerically, it first must be discretized, so that a discrete solution variable w is defined. For illustrative purposes, Eq. (7) is temporally discretized with the Forward Euler method and spatially discretized with first-order upwinding. The discrete difference equation at an arbitrary node j can be written as

$$w_j^{n+1} = u_j^n - \frac{\lambda \Delta t}{\Delta x} (u_{j+1}^n - u_j^n) \quad (8)$$

where Δt and Δx represent the time-step and cell size, respectively.

The LTE measures how well the discrete equation models the exact solution for a single time-step, and therefore it may be estimated using the exact and numerical solutions. Because w^{n+1} is guaranteed to satisfy the difference equation, it is likely that the exact solution (u^{n+1}), when inserted into $\mathcal{R}(\cdot)$, which is the residual of Eq. (8), will not satisfy it. The resulting non-zero term is the LTE, which for Eq. (8) can be written as

$$\text{LTE}(x, t) = \frac{u(x, t + \Delta t) - u(x, t)}{\Delta t} + \frac{\lambda}{\Delta x} (u_{j+1} - u_j)$$

In turn, we note that this is equivalent to

$$\text{LTE} = \frac{u^{n+1} - w^{n+1}}{\Delta t} \quad (9)$$

Because the LTE is dependent upon space and time, it is usually expressed in terms of their delta quantities when discussing the numerical order-of-accuracy. For example, if $\text{LTE} = \mathcal{O}(\Delta t^q) + \mathcal{O}(\Delta x^p)$, the numerical scheme is considered to be q th- and p th-order accurate in time and space, respectively. Although the aforementioned derivation used the Forward Euler method for time discretization, it can be shown that this analysis is equally applicable to multistage Runge–Kutta time integration schemes, such as those used in the ARC3DC solver [38].

Having expressed the LTE for a general unsteady case, we now examine how spatial and temporal refinement will work to reduce the error present for an advecting Shu–Erlebacher–Hussaini [39] vortex. The vortex is advanced for a single time-step from its initial state, which enables a complete analysis of the spatial and temporal convergence as a function of the mesh size and time-step. The exact solution is found by translating the initial solution at a velocity set by the freestream, and the following information explains the computational solution procedure. The vortex is placed at the center of a domain defined by $x \in [0, 32]$ and $y \in [0, 32]$, and the parametric study is conducted with $\Delta t = \{5, 5/2, 5/4, \dots, 5/512\}$ and $\Delta x = \{2, 1, 1/2, \dots, 1/32\}$. For reference, the vortex convects through the domain at 0.1 Mach, which, given the size of the applied time-steps, translates the vortex at core widths ranging between 0.5 and 9.76×10^{-3} grid units. Regarding grid resolution, the vortex core radius is approximately 1.2, so that the coarsest and finest grids contain 1 and 75 points along the diameter, respectively. Examining such a wide range of scales ensures that the asymptotic solution limit is attained for both Δt and Δx , so that the error behavior can be fully assessed.

Figure 4a contains the spatial accuracy study, where the x -momentum error is plotted as a function of Δx . A series of time-steps are applied and the resulting LTE is shown. For the larger time-steps, the temporal error dominates the spatial error, causing the LTE to stall in the limit of small spatial resolution. However, when Δt is small enough or when $\frac{\Delta t}{\Delta x}$ is approximately less than or equal to unity, fifth-order accuracy is clearly demonstrated. This ratio also happens to represent the Courant–Friedrichs–Lewy number ($CFL = \frac{\Delta t}{\Delta x}$) number used in ARC3DC and is bounded by RK3's CFL limit of $\sqrt{3}$, showing that smaller time-steps are required for both accuracy and stability.

The complementary temporal accuracy study is shown in the convergence plot in Fig. 4b, in which a series of different spatial

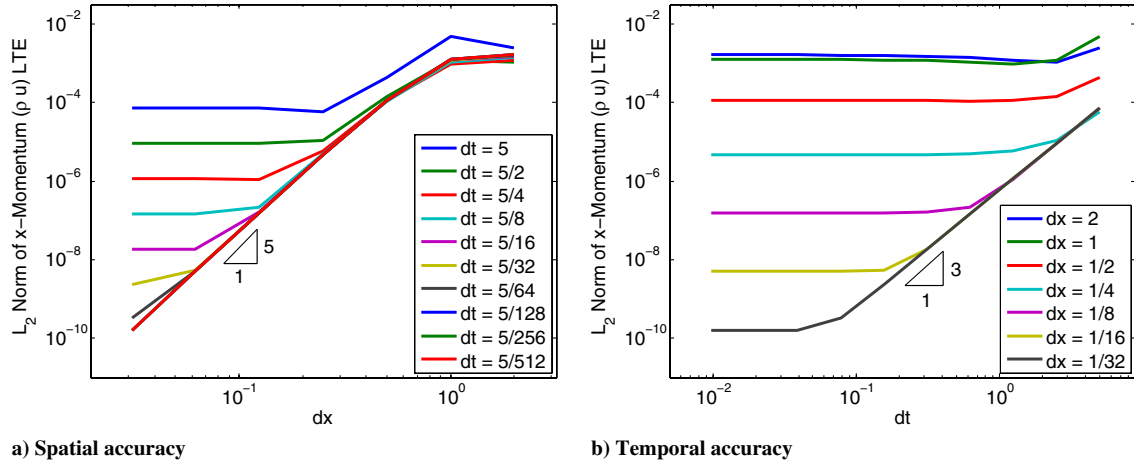


Fig. 4 Convergence of the local truncation error (LTE) for the advecting vortex case. For $\frac{\Delta t}{\Delta x} \approx 1$, the scheme is fifth- and third-order accurate in space and time, respectively.

resolutions are applied across a range of time-steps. Although it showcases the same data as that presented in Fig. 4a, the lines now represent constant Δx to illustrate temporal convergence. Third-order accuracy is achieved in regions where $\frac{\Delta t}{\Delta x}$ is approximately equal to or greater than unity, and stalled convergence occurs when Δt is large compared to Δx . This study thus verifies that the numerical discretization scheme used is third-order accurate in time and fifth-order accurate in space.

B. Richardson Extrapolation

Although the LTE represents a convenient form of the discretization error, it cannot be used practically because the exact solution must be known. If the asymptotic solution exhibits logarithmic error convergence, which was shown in the previous section, recursively finer meshes will better approximate the exact solution. This idea is key to the Richardson estimator, where finer discrete solutions are used in place of the exact solution to compute the error. Note that this computed error is a *relative* error between the fine and coarse grid levels, but its convergence will be shown to be comparable to the LTE. Before presenting its formulation, we first cover the basic assumptions of Richardson extrapolation, which, as discussed by Roy [40], must include 1) uniform and systematic refinement, 2) smooth and asymptotic solutions, and 3) dominant discretization error:

1) The block-wise refinement used in SAMARC ensures the assumption of uniform refinement. Cell aspect ratio, cell skewness, and cell-to-cell stretching all remain constant because of the systematic and isotropic refinement on each level provided by the Cartesian grid system.

2) The condition of smooth and asymptotic solutions is generally valid for well-resolved rotorcraft flowfields, as they do not contain discontinuities, e.g., shocks. It is expected that while the coarse baseline mesh may not completely resolve the wake dynamics, the applied sequence of increasingly refined meshes will tend towards asymptotic behavior. Recall that the error estimator is being used to *terminate* refinement, which is likely to only occur after the vortex cores are sufficiently resolved.

3) Discretization error measures how well the discrete solution, generated through an iterative or equivalent time-marching process, is able to approximate the actual physical system for the matching set of continuous equations. The ability of the discretization scheme to achieve this was shown in the earlier wave equation formulation.

1. Derivation

A Taylor series approximation can be used to express the exact solution in terms of the discrete solution and higher-order terms. For example, the discrete solution w_h is computed on a grid with spacing $h = \Delta x$,

$$u = w_h + \sum_{i=0}^{\infty} C_i h^{p+i} \quad (10)$$

where w_h denotes the discrete solution, p refers to the applied spatial order-of-accuracy, and C_i represents a set of unknown constants. Applying Landau notation, the higher-order terms can be grouped together so that a single term is extracted from the infinite sum,

$$u = w_h + C_0 h^p + \mathcal{O}(h^{p+1}) \quad (11)$$

If the higher order term is dropped, Eq. (11) represents a single linear equation. This equation can be used to create a system of linearly independent equations for each grid level available, which, when combined, can be used to develop a better approximation of u . Equation (11) uses a grid size of h , and so for demonstrative purposes, it is assumed that a finer solution is available on $h/2$,

$$u = w_{h/2} + C_0 \left(\frac{h}{2}\right)^p + \mathcal{O}\left[\left(\frac{h}{2}\right)^{p+1}\right] \quad (12)$$

With some algebraic manipulation, Eqs. (11) and (12) can be combined to form an approximation of u ,

$$u = \frac{2^p w_{h/2} - w_h}{2^p - 1} + \mathcal{O}(h^{p+1})$$

Note that the constants C_i cancel and the aforementioned expression has improved the order-of-accuracy by one, as compared to the original discrete representation in Eq. (10).

Beyond improving accuracy of the functional estimate, a similar technique can be used to compute the error. Solving for C_0 in Eq. (11), the error on $w_{h/2}$ can be defined as the difference between the exact and computed solutions, i.e., $u - w_{h/2}$

$$\mathcal{E}_{h/2} = \frac{w_{h/2} - w_h}{2^p - 1} + \mathcal{O}(h^{p+1}) \quad (13)$$

Note that the denominator is a function of p (applied spatial order-of-accuracy) and is therefore constant on all grid levels. As such, the error can be simply expressed as the difference between the coarse and fine solutions.

For steady problems, Eq. (13) can be directly applied to compute the error between two systematically refined grids. However, because our prime interest is in developing a well-behaved unsteady error estimation technique, we now consider how a time-accurate solution influences the spatial and temporal errors. Let us assume that a fine and a coarse solution are separated by n levels of refinement, so that the coarse and fine solutions exist on grids with spacing h and $h/2^n$, where $n = \{1, 2, 3, \dots\}$. The wave equation is reconsidered [Eq. (7)], and if a discrete second-order accurate solution is computed, the difference is

$$w_f^{n+1} - w_c^{n+1} = u_f^n - \lambda \Delta t \left. \frac{\partial u_f^n}{\partial x_f} \right|_d + \frac{\lambda^2 \Delta t^2}{2} \left. \frac{\partial^2 u_f^n}{\partial x_f^2} \right|_d - u_c^n + \lambda \Delta t \left. \frac{\partial u_c^n}{\partial x_c} \right|_d - \frac{\lambda^2 \Delta t^2}{2} \left. \frac{\partial^2 u_c^n}{\partial x_c^2} \right|_d \quad (14)$$

Here, we also assume that the size of the time-step taken on the coarse and fine grids is identical and that the flow solution has advanced only a single time-step from the initialization of u_f and u_c , so that $u_f^n - u_c^n = 0$. Recall from Sec. II.C that the multilevel algorithm injects the finest solution available into its overlapping coarser level before the start of each time-step, and so this assumption remains valid. Consequently, the computed error represents the error generated during a single time-step. The fine-coarse difference can now be rewritten as

$$w_f^{n+1} - w_c^{n+1} = -\lambda \Delta t \left(\left. \frac{\partial u_f^n}{\partial x_f} \right|_d - \left. \frac{\partial u_c^n}{\partial x_c} \right|_d \right) + \frac{\lambda^2 \Delta t^2}{2} \left(\left. \frac{\partial^2 u_f^n}{\partial x_f^2} \right|_d - \left. \frac{\partial^2 u_c^n}{\partial x_c^2} \right|_d \right) \quad (15)$$

For illustrative purposes, second-order accurate derivatives can be used so that the first bracketed term on the right hand side will resemble

$$\left. \frac{\partial u_f^n}{\partial x_f} \right|_{\text{II}} - \left. \frac{\partial u_c^n}{\partial x_c} \right|_{\text{II}} = \frac{\Delta x_f^2}{6} \frac{\partial^3 u_f^n}{\partial x_f^3} - \frac{\Delta x_c^2}{6} \frac{\partial^3 u_c^n}{\partial x_c^3} + \mathcal{O}(\Delta x_f^3) + \mathcal{O}(\Delta x_c^3)$$

Applying the assumption that $\Delta x_f = \Delta x_c/2^n$, this equation can be expressed as

$$\left. \frac{\partial u_f^n}{\partial x_f} \right|_{\text{II}} - \left. \frac{\partial u_c^n}{\partial x_c} \right|_{\text{II}} = \left(1 + \frac{1}{2^n} \right) \left[\frac{\Delta x_c^2}{6} \left(\frac{\partial^3 u_f^n}{\partial x_f^3} - \frac{\partial^3 u_c^n}{\partial x_c^3} \right) + \mathcal{O}(\Delta x_c^3) \right] \quad (16)$$

When the flow is smooth, the higher-order derivatives will contain errors that are significantly smaller than Δx_c . For a general p th-order accurate derivative, Eq. (16) may be approximated as

$$\left. \frac{\partial u_f^n}{\partial x_f} \right|_p - \left. \frac{\partial u_c^n}{\partial x_c} \right|_p \approx \Delta x_c^p$$

Now, the fine-coarse difference expressed by Eq. (15) can be order-approximated by the following series:

$$w_f^{n+1} - w_c^{n+1} = \mathcal{O}(\Delta x_c^p) \sum_{k=1}^{\infty} \mathcal{O}(\Delta t^k) \quad (17)$$

Note that term on the right-hand side includes both Δx_c^p and Δt^k , indicating that the spatial and temporal effects are still coupled.

Comparing Eq. (17) with the analogous LTE formulation [Eq. (9)], we note that the two uncoupled spatial and temporal terms no longer exist for the Richardson estimator. Rather, the Richardson estimate comprises only coupled space-time terms, which will effectively remove the small- Δt stalled convergence behavior observed in the LTE analysis. We note that this result is naturally a consequence of using the same physical time-step between different grid levels, but this assumption is generally valid in practical simulations.

If the time-integration scheme involves time refinement in conjunction with the spatial refinement, and the spatial operator order-of-accuracy is p , whereas the time order-of-accuracy is q , a larger coarser level time-step may be applied and still maintain temporal error convergence as long as

$$\Delta t_{\text{coarse}} \leq \Delta t_{\text{fine}} \left(\frac{\Delta x_{\text{coarse}}}{\Delta x_{\text{fine}}} \right)^{p/q} \quad (18)$$

The time integration scheme used in this work, however, applies a uniform time-step across all grid levels; time refinement is not used.

2. Validation

To validate the Richardson estimator, we consider the same advecting vortex problem from our earlier LTE study. Note that the number of grid levels considered will be one fewer because the extrapolation process can only generate $n - 1$ error estimates from n solutions. Moreover, two separate studies will be conducted in this section. The first study will compute error on the coarse level using the *finest* solution available. This essentially represents the ideal case, because the finest solution is a reasonably good approximation to the exact solution. In the second case, for each grid level, the *next-finer* solution will be used, which represents what would be used in a practical implementation.

Figure 5a plots the L_2 norm of Richardson x -momentum error on different levels as compared to the *finest* solution available ($\Delta x = \frac{1}{32}$), using a form of Eq. (17) that is normalized by Δt . Here, as $\Delta x \rightarrow 0$, fifth-order spatial convergence is observed for all Δt and the error decay rate does not stall, even for large Δt . With the exception of $\Delta t = 5$ on the two coarsest grids ($\Delta x = 1, 2$), which is numerically unstable, the spatial convergence is generally unaffected by variations to time-step size and exhibits smooth monotonic convergence. These rates of convergence are in agreement with the LTE analysis presented earlier.

Next, Fig. 5b contains the same Richardson error quantity, but it is computed using the solution from the *next-finer* level. All time-step plots match the general behavior as found in Fig. 5a. Spatial error convergence is relatively asymptotic, even for the highly under-resolved cases. Moreover, the monotonic profiles illuminate the inherent numerical stability associated with the Richardson estimator, allowing it to be a robust control mechanism for grid

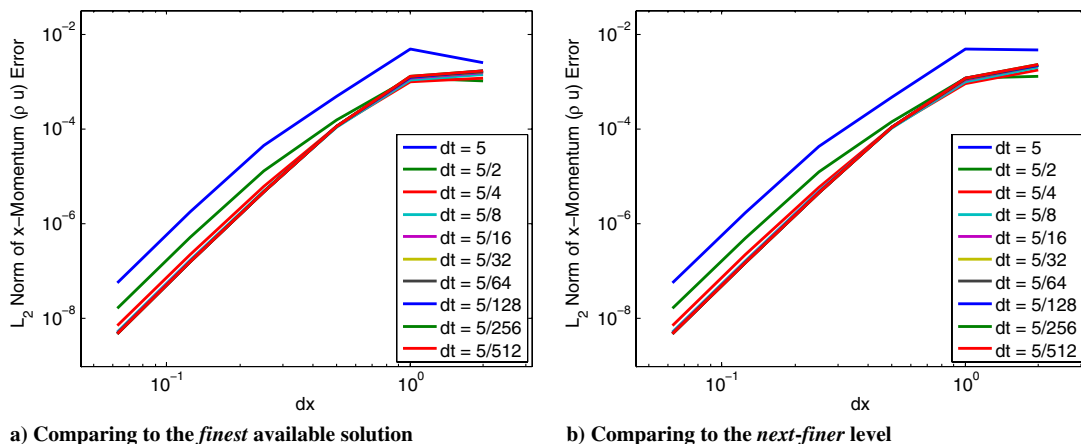


Fig. 5 Spatial convergence of the Richardson estimator, compared to a) finest solution available and ($\Delta x = \frac{1}{32}$) and b) solution on next-finer level for 5th-order spatial discretization.

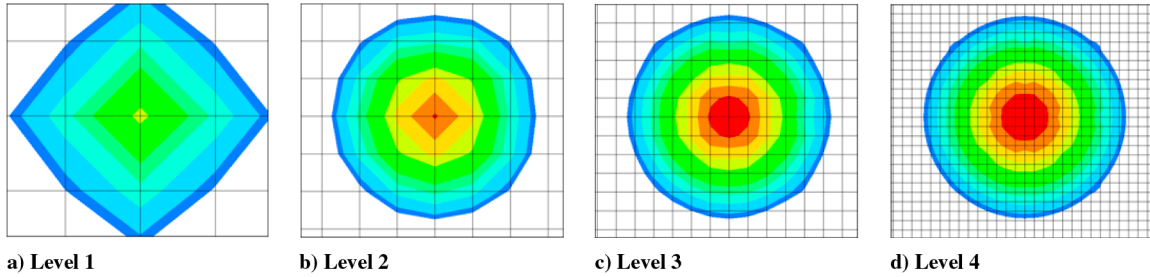


Fig. 6 Initial contours of vorticity that depict vortical resolution for the time-accurate error analysis study. Relative to the maximum magnitude found on L4, the lack of flow resolution on L1 has reduced its strength by about 30%.

adaptation. Temporal accuracy studies presented elsewhere [38] similarly demonstrate ideal convergence.

The aforementioned two-part study verifies that the reported error is strongly dependent upon the solution on *coarser* grid, rather than the finer grid. While Eq. (17) indicated that the computed error is dominated by Δx_c , this test confirms that using a very fine solution, which closely approximates the exact solution, offers no additional benefit as compared to using the solution from the next-finer level. In practice, error on the finer grid will be computed by comparing it to its coarser equivalent, although the error converges according to the coarse mesh solution. For those fine mesh points that lie between coarse mesh points, interpolation will be used.

3. Time-Accurate Behavior

The previous spatial convergence studies demonstrated that the Richardson estimator is suitable for unsteady flows using a single time-step. We next examine the error during a full time-accurate simulation. Furthermore, to understand how convective and dissipative errors individually contribute to total error, two analogous time-dependent cases, one in which the vortex remains stationary, and the other in which it advects through the domain, are studied. It is desired that the error from a non-stationary vortex is comparable to a stationary one, so that varying advection rates do not influence reported error. Both cases use five levels of refinement where $h_L = \{8/3, 4/3, 2/3, 1/3, 1/6\}$, allowing the coarsest and finest levels to contain about one and seventeen points across the core, respectively. Contour plots of vorticity for the four finest grids, illustrating core resolution, are shown in Fig. 6. Adaptively refined meshes are not yet applied, as this fundamental test computes error on uniformly refined meshes.

First, for the stationary case, the vortex is placed at the origin and experiences no mean-flow. The solution is advanced 100 time-steps of $\Delta t = 0.8$. The time-history of a pressure-based error, which represents the static pressure difference between two grid levels, on the four finer meshes during the simulation is presented in Fig. 7a. Although the result is somewhat benign, the constant error across all

levels represents ideal behavior. Recall that at the end of each time-step, the finer solution is injected into their coarser parents. This, combined with the higher-order spatial difference operators, prevents any error fluctuation, and the Richardson error is therefore constant. We also note that the sequentially refined meshes lead to greater reductions in error, thereby demonstrating asymptotic grid convergence.

For the next case, all run parameters are similar for the stationary vortex case, but now the vortex convects about four vortex core widths from $\mathbf{x}_i = \{-4, 0\}$ to $\mathbf{x}_f = \{4, 0\}$. Figure 7b shows a plot of the error behavior. The magnitudes are comparable, but some oscillatory behavior is observed. This occurs because the error is computed at the overlapping grid points, allowing the vortical center, which contains the largest error, to be adequately resolved, depending upon its relative position to the nearest node. Interestingly, the number of cycles is predictable, as the L1 error exhibits three periods during the simulation equal to the number of L1 grid points that it eclipses. The same trend is observed for the L2 error, except the period is cut in half due to the Δ -to- $\frac{\Delta}{2}$ refinement strategy employed by the grid system. Additional refinement produces higher frequency content but at increasingly smaller amplitudes. However, despite these oscillations, the error remains periodically constant over the course of the unsteady simulation. The maximum, minimum, and average error amplitudes shown in Table 1 are comparable to the stationary case.

C. Selecting an Error Function

An advantage of the Richardson estimator is that any locally computed function can be easily used to measure solution error. Nevertheless, one must be careful in selecting the proper functional value to characterize the local error. Specifically, the following criteria are applied to help design a suitable error functional.

First, the functional error should be inherent to the flow and not derived, i.e., quantities directly computed from the state variables are acceptable whereas vorticity is not. This criterion is imposed to prevent discretization error, which is likely to occur in

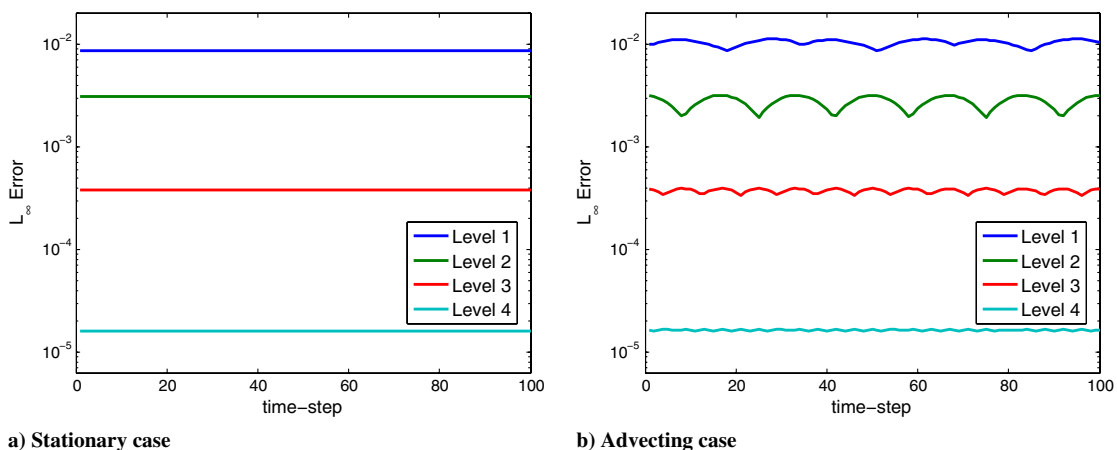


Fig. 7 Mass flux error, as computed by the Richardson estimator on each level during 100 time-steps for the stationary and moving vortex cases.

Table 1 Comparison of L_∞ error norm, as computed by the Richardson estimator, for a time-accurate solution of the modified-EHS vortex when either stationary or moving through the domain

	Stationary case		Advecting case	
	(constant)	(min)	(max)	(avg)
L1	8.68×10^{-3}	8.63×10^{-3}	1.13×10^{-2}	1.04×10^{-2}
L2	3.10×10^{-3}	1.94×10^{-3}	3.19×10^{-3}	2.74×10^{-3}
L3	3.80×10^{-4}	3.38×10^{-4}	3.94×10^{-4}	3.75×10^{-4}
L4	1.61×10^{-5}	1.60×10^{-5}	1.66×10^{-5}	1.64×10^{-5}

under-resolved vortices, from affecting the measurement of the solution error. Second, the functional error must be Galilean invariant. This property ensures that moving reference frames do not influence the computed value, which is necessary to ensure that background flowfields do not affect either the magnitude or profile of the error. Lastly, the functional error should be well scaled. By this we mean that the error should scale with respect to the convective scales inherent to the specific problem. Therefore, for a given problem, the error levels should remain constant, even if the vortical strength is increased or decreased, possibly indicating that the body is generating more or less lift.

To compare different error functions, we use the same case setup from Sec. IV.B.3 and compute the error on L3 by comparing the solutions on L2 and L3. Note that a stationary vortex is tested, as the low-Mach limit case proved more challenging for some of the selected candidates. Figure 8 plots the error reported by three

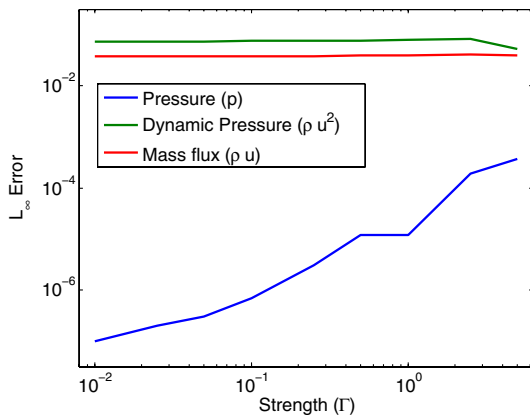


Fig. 8 Maximum Richardson pressure error, dynamic pressure error, and mass flux error, over a range of strengths for a single vortex.

estimators: pressure, dynamic pressure, and mass flux. In each case, the error, computed as the difference between the fine and coarse grids, is normalized by its associated global maximum value, e.g., $p_{\text{error}} = (p_{\text{fine}} - p_{\text{coarse}})/p_{\text{max}}$, and is plotted for a range of vortical strengths, $\Gamma \in [0.05, 5.0]$. A local normalization is not applied because it leads to strong error overshoot conditions in regions of stagnant flow. Figure 8 demonstrates that pressure error is strongly correlated with vortical strength, whereas dynamic pressure and mass flux are significantly less. Moreover, in the limit that the vortex contains sonic flow ($\Gamma \rightarrow 5$), mass flux is slightly better behaved than dynamic pressure. Similar tests indicate the mass flux error is well behaved for different freestream Mach numbers as well. Additionally, there is no additional computational overhead in computing the mass flux because it is contained in the state vector. As such, it will be used to estimate error for all cases in Sec. VI. However, before presenting these results, the coupled refinement strategy that combines feature detection and error estimation is offered in the next section.

V. Refinement Strategy

The overall mesh refinement strategy that couples the feature-detection methodology with the Richardson error estimator is outlined in this section. The software package responsible for controlling the AMR process is called guided adaptive mesh refinement (GAMR) and operates by identifying vortical structures and adding refinement in regions where the solution error is unacceptably large. Although GAMR is implemented within SAMARC (see Fig. 9), the module can be easily integrated into any solver that uses a multi-level Cartesian-based AMR paradigm. The rest of this section specifically addresses the refinement and coarsening algorithm followed by GAMR.

To explain the refinement process, we begin with a sine-like feature on L_N , as illustrated by Fig. 10a. Let us assume that L_N represents the finest level in a particular region of the mesh so that new-level construction may occur above it. For example, if a grid system contained L0, L1, and L2 grids, the only permissible way of building a finer L3 region would be if the L2 solution required it.

Initially, the wave is identified by feature detection, as shown by Fig. 10b. Once complete, the computed nodes with unsuitably high error are marked for refinement, while those below the prescribed error tolerance are not refined. Figure 10c illustrates this process, as some of the previously tagged cells (green x's) are untagged (red x's). The tagging process is now complete and Fig. 10d shows the newly created L_{N+1} grid above L_N . Note that cells adjacent to the flagged nodes are refined because SAMARC performs all refinement operations in a cell-wise fashion. This general process is followed for each grid level, with the exception of the coarsest baseline grid (L0).

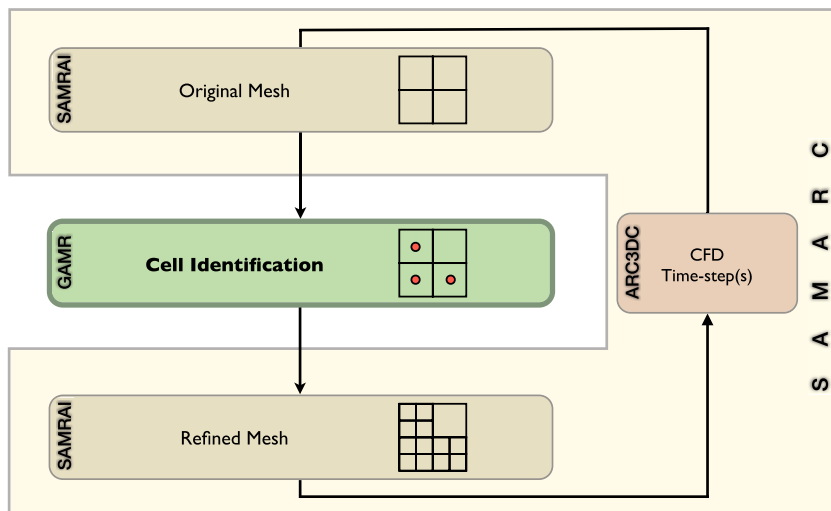


Fig. 9 Coupling GAMR with SAMARC, the AMR-capable, off-body flow solver.

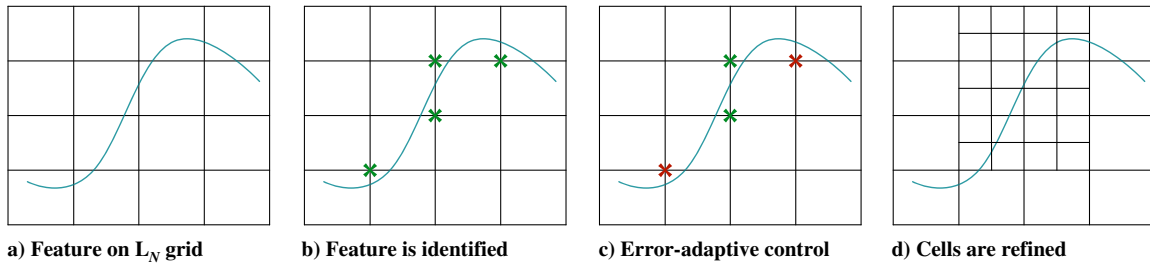


Fig. 10 Refinement process: a) feature on a particular grid block, b) initial tagging of feature by the detection algorithm, c) removal of tags that are under designated threshold error, and d) subsequent refinement.

Here, feature detection occurs as normal, but error-adaptive control cannot be applied because no error estimate exists because the Richardson technique requires an overlapping solution. In addition, during a single re-grid cycle, a maximum of one level can be added to an existing grid-level hierarchy. Although the relative error between L_N and L_{N-1} may be used to determine if L_{N+2} (and finer) should be constructed, such a strategy is likely to produce erratic mesh behavior. Error reductions may be highly nonlinear, making larger extrapolations less accurate. Instead, we prefer to have the mesh gradually evolve with the flow feature using a fast re-grid rate, which is practical due to the efficient AMR infrastructure.

In addition to refinement, coarsening operations are employed to remove refinement where no longer necessary. Recall that SAMARC operates one level at a time, and performs re-grid operations from coarse to fine. Therefore, if a feature has moved and is no longer contained in a cell situated in L_n or if the error drops below the tolerance, *all* the overlaid levels at or above L_{n+1} will be removed automatically.

As a final note, the relative CPU speeds of the feature detection and error estimation routines are quite small, on the order of one percent of a single off-body CFD iteration. This is significantly more economical than an adjoint solution, which is at least expensive as a single flow solution step. Moreover, these algorithms have been

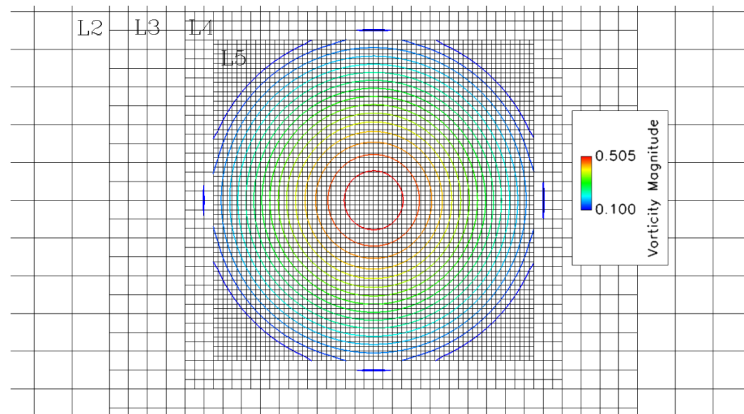
designed with high-performance computing in mind and are executed in parallel.

VI. Results

In this section, the coupled strategy is allowed to drive the off-body refinement process within Helios. The test cases include an advecting vortex, and the flow over a NACA 0015 wing, and a rotating quarter-scale V-22 (TRAM) rotor. In each case, the key objective is to demonstrate that the degree of applied grid resolution can be automatically controlled by the combined feature detection and error estimation approach. The associated tolerance will govern the level of solution fidelity, and the grid points will be distributed in a manner that targets regions containing highest error.

A. Advecting Vortex

In our previous work, the nondimensional feature detection methods were shown to effectively drive the AMR process for an unsteady advecting vortex. Now, the Richardson error estimator is coupled with the nondimensional Q method and is allowed to limit the amount of resolution applied to the vortex. To understand the impact of different error thresholds on the time-dependent solution,



a) Initial solution. Note: L1 and L0 surround the L2 grid system, but are not represented above

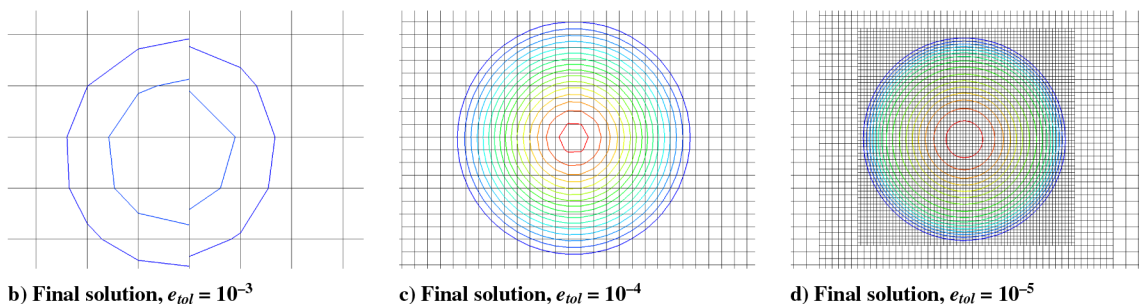


Fig. 11 Initial vortex solution on six levels of resolution, with RE error estimator adding or removing resolution at each adapt step; finest resolution capped at L5.

Table 2 The L_∞ solution error at the end of the simulation, number of required grid points, nodal multiplier factor for comparable uniform simulation, and the finest supplied grid level, when using various error tolerances to control grid resolution

e_{tol}	Error	Nodal count	AMR savings	Finest level
10^{-3}	3.2×10^{-1}	778	2.1x	L1
10^{-4}	1.3×10^{-2}	2,592	9.9x	L3
10^{-5}	5.8×10^{-3}	10,432	39.3x	L5

three different error tolerances of 10^{-3} , 10^{-4} , and 10^{-5} are each applied to the exact same initial solution. Figure 11a represents the starting vortex on the six overlapping grid regions, where the grid spacing is $h_L = \{1, 1/2, 1/4, 1/8, 1/16, 1/32\}$. The diameter of the vortex is about 2, and so the coarsest (L0) and finest (L5) resolutions contain about 3 and 64 points across the core, respectively.

For the simulation, the vortex is placed at the origin of a domain defined by $\mathbf{x} \in [-10, 10]$ and $\mathbf{y} \in [-10, 10]$. The vortex convects rightward ($+x$) at Mach = 0.1 for a distance of 20 core widths, with periodic boundary conditions used in x , and Neumann conditions in y . A single period is $t = 400$, and by enforcing a CFL of 1.0 on the finest level, $\Delta t = 0.01$. For all cases, a fixed re-grid period of thirty time-steps is employed. This frequency ensures that the mesh is adapted before the vortex enters and exits an L5 cell. Therefore, during the entire 40,000 time-steps, 1,333 re-grid operations are performed. Additionally, the maximum number of mesh levels is restricted to six (L5). This restriction provides an upper bound on the number of grid points, and maintains a reasonable Δt for ARC3DC's explicit solution procedure.

Figures 11b–11d display the final solutions when applying the three different error tolerances. Although each travels an equal distance, larger error tolerances provide less mesh refinement, which leads to increased numerical dissipation of the vortical core. For the largest tolerance of 10^{-3} , the vortex has encountered severe dissipative effects and only retains a small fraction of the starting vorticity. Note that the discontinuous vorticity contour lines in Fig. 11b do *not* represent the true solution, but occur because the visualization software occasionally encounters difficulty when computing vorticity on a multiblock grid. Next, when the tolerance is lowered to 10^{-4} , the solution dramatically improves, providing two additional mesh levels, which help retain the initial size and strength of the vortex. Finally, when it is lowered to 10^{-5} , more mesh levels are provided, but the relative improvement of the overall feature resolution, compared to the previous result, is less.

Table 2 compares the number of supplied grid points and the resulting solution error for three different error tolerances. The solution error represents the difference between the exact and computed solutions and is normalized with respect to the exact value. From the tabulated data, we arrive at two key conclusions. First, with regard to computational expense, there exists a strong correlation between the number of grid points and applied error tolerance. Each time the tolerance is dropped by an order of magnitude, about three to four times as many points are added. Second, with regard to computational accuracy, there exists a similar correlation between the applied error tolerance and the reported solution error. Lowering the error tolerance, results in comparably lower solution error.

While the task of selecting an appropriate error tolerance is dependent upon the particular application, the error tolerance of 10^{-4} provides a near-optimal solution for this case. With this tolerance, about 16 mesh points are placed along the vortical center and this level of resolution is sufficient according to previous results for similar vortex convection problems that use high-order spatial discretization [37].

B. NACA 0015 Wing

This test case is based on the original experiment of McAlister and Takahasi [41], where a series of vortex core measurements were taken at several downstream locations for different angles of attack, freestream velocities, and wing-tip shapes. The current test case

considers the steady flow around a full-span NACA 0015 square wing at a 12 deg angle of attack. The wing experiences uniform inflow with a Mach number of 0.1235 and a Reynolds number of 1.5×10^6 . Dirichlet conditions are imposed along the far-field boundary. The Helios dual-mesh infrastructure, outlined by Sec. II, is employed and the hybrid unstructured/structured-Cartesian grid system at the start of the simulation (before any solution-based refinement) is illustrated in Fig. 12. The unstructured grid surrounds the wing section and is embedded within the Cartesian mesh. The off-body domain pictured here is comprised of four mesh levels where $h_{L0} = 0.2$, $h_{L1} = 0.1$, $h_{L2} = 0.05$, and $h_{L3} = 0.025$, where the grid distance has been nondimensionalized by the chord length. In general, re-gridding is performed every 100 iterations and convergence is obtained after about 250 adaption steps.

Previous work simulated this case with feature-only adaption and it was demonstrated that feature-based AMR leads to significant improvements in far-field vortex resolution. As in the advecting vortex case, we now apply three error tolerances of 10^{-2} , 10^{-3} , and 10^{-4} to control the refinement process. To control computational costs, the maximum allowed number of grid levels was restricted to four. The Richardson error estimator is therefore only active through three grid levels; addition of finer levels beyond L3 is restricted. The final solutions and grids are shown in Fig. 13. As before, larger error tolerances limit the amount of mesh resolution applied, leading to increased numerical dissipation. Setting $e_{\text{tol}} = 10^{-2}$ only allows the vortex to travel about six chord lengths downstream, while a threshold of 10^{-3} doubles the distance to about twelve chords back. Further reducing the tolerance to $e_{\text{tol}} = 10^{-4}$ allows for the cores to reach the end of the domain, a distance of nearly twenty chords.

A quantitative comparison of the solutions is given in Table 3. A solution generated with feature-only refinement, i.e., without error-based control ($e_{\text{tol}} = 0$), is used as a benchmark, and the relative error for each case is computed at 6, 12, and 18 chord lengths downstream. The smallest error tolerance of 10^{-4} results in perfect agreement with the feature-only solution, and does so with about 4% fewer grid points. The next larger tolerance of 10^{-3} uses about two million fewer points, matches the solution very well at 6 chord lengths and only incurs an error of 13% at 12 chord lengths. However, near the twelfth chord length station, the finest grids are removed and this leads to an error increase to 42% at 18 chord lengths downstream. Lastly, when $e_{\text{tol}} = 10^{-2}$, the vortex is under-resolved at all the chord locations shown. The amount of error reported is 24% after the first six chord lengths, and it further increases to 57% over the next six chords. After this, the relative error decreases slightly to 48% over the next six chords. This demonstrates how crucial grid resolution is to vortex preservation in the wake.

Although these plots clearly demonstrate that a Richardson estimator is capable of controlling the amount of resolution that a vortical feature receives, the importance of the feature detection

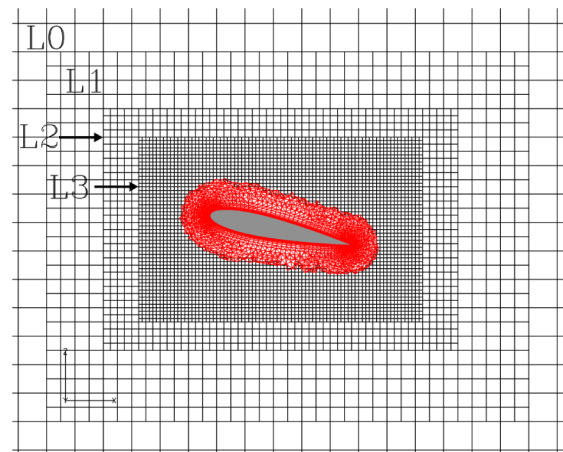


Fig. 12 The NACA 0015 near-body unstructured grid with the unadapted off-body Cartesian grid system.

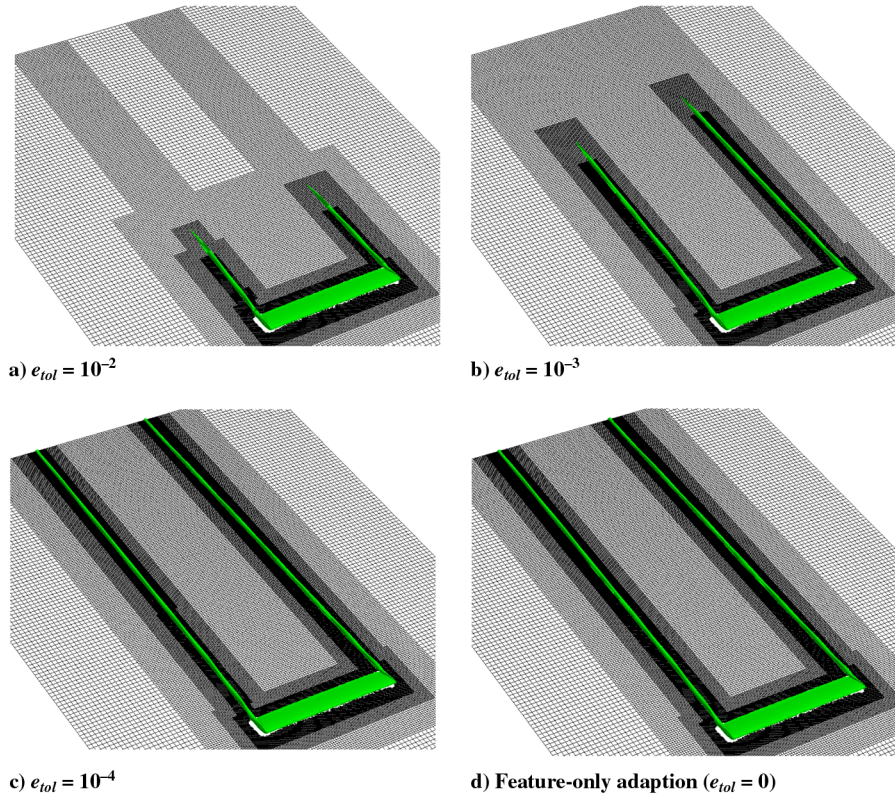


Fig. 13 Adaptive grids (black) and equal isosurfaces of vorticity magnitude (green) for the NACA 0015 case; Richardson error controls grid resolution in refinement regions detected by non-dimensional Q method.

should not be overlooked. By only allowing refining where features are found, and *not* to all regions with high error, the current approach avoids refining the entire turbulent wake region. According to the Richardson estimator, the wake contains high error but the feature detection procedure prevents this region from being flooded with points. Figure 14 demonstrates that by targeting the vortex cores, refinement is specifically directed towards them. We note that for certain problems, it may be necessary to further resolve the wake region as well, in which case it is possible to rely entirely upon the Richardson error to guide the AMR process. However, we do not consider such limited cases in the present study.

C. Tilt Rotor Acoustic Model (TRAM)

The tilt rotor acoustic model (TRAM) is a quarter-scale model of the Bell/Boeing V-22 Osprey tiltrotor aircraft right-hand 3-bladed rotor. The isolated TRAM rotor was tested in the Duits-Nederlandse Windtunnel Large Low-speed Facility (DNW-LLF) in the spring of 1998. Aerodynamic pressures, thrust, and power were measured along with structural loads and aeroacoustics data. Wake geometry, in particular the locations of tip vortices, was not part of the data collected. Further details on the TRAM experiment and extensive CFD validations with OVERFLOW can be found in the work of Potsdam and Strawn [4].

For the computational tests, $M_{tip} = 0.625$, $Re_{tip} = 2.1 \times 10^6$, and the collective is set to 14 deg. A noninertial reference frame is used, such that the rotor stays fixed. Although the freestream Mach number is low, the speed of the flow relative to the blade is high due to the rotational terms. The TRAM geometry is comprised of three blades and a center body. The near-body volume mesh contains 2.8×10^6 nodes and extends to approximately two chord lengths from the blade surface. The maximum number of off-body levels is seven, which corresponds to a length that is 10% of the rotor tip chord. The unstructured grid is shown embedded within the unadapted Cartesian mesh in Fig. 15.

To test the Richardson estimator the adaptive error-based control is turned off in regions where the computed error is below e_{tol} . Three different error tolerances, of 10^{-2} , 10^{-3} , and 10^{-4} , are applied. As in the NACA 0015 case, the maximum allowed number of grid levels was restricted to seven to control the computational cost. The Richardson error estimator is active only through the six levels of refinement; addition of finer levels beyond L6 was restricted. The resulting flow-fields and adaptive Cartesian grid systems are shown in Fig. 16. Again, the results clearly indicate that, by using the solution error computed by the Richardson estimator, the off-body mesh resolution can be effectively controlled. Similar to the NACA case, the amount of far-field refinement is continually reduced when the error tolerance is increased. When $e_{tol} = 10^{-2}$, only a relatively

Table 3 Solution fidelity, as a function of error tolerance and downstream distance from the trailing edge, for the NACA 0015 case. Maximum normalized swirl velocity (V_z/V_∞) measured at various downstream locations and is used to represent vortical strength, and percent-error is computed by using V_z/V_∞ . Percentage error is relative to solution without using error-based control ($e_{tol} = 0$)

$e_{tol} = 0$	Nodal count	6 chords		12 chords		18 chords	
		V_z/V_∞	% error	V_z/V_∞	% error	V_z/V_∞	% error
0	10.9×10^6	0.398	—	0.370	—	0.286	—
10^{-2}	6.3×10^6	0.302	24%	0.159	57%	0.147	48%
10^{-3}	8.3×10^6	0.398	0%	0.322	13%	0.164	42%
10^{-4}	10.5×10^6	0.398	0%	0.370	0%	0.286	0%

small amount of off-body refinement is applied and the cores remain coherent for just about one-half revolution. Lowering the tolerance to 10^{-3} preserves the helical vortex system for more than a revolution, and further decreasing e_{tol} to 10^{-4} produces a solution similar to the feature-only case.

It should also be mentioned that between the NACA and TRAM cases, similar error threshold levels resulted in similar off-body fidelity. This is especially notable because the maximum vorticity magnitude in the off-body of the NACA case is about two orders of magnitude greater than in the TRAM case. While additional comparative studies are necessary to understand the effects of different error thresholds upon rotor performance and wake prediction, it is verified here that the proposed error estimator procedure has the potential to guide refinement without laborious tuning of the threshold parameters.

VII. Conclusion

An adaptive mesh refinement (AMR) strategy is developed for unsteady vortex-dominated flows that couples feature detection with a Richardson-like error estimation to control grid resolution. The refinement strategy follows a two-part approach. First, feature detection is used to identify regions of the off-body domain for mesh refinement. Local solution error, as computed by the Richardson error estimator, is then used to control the amount of mesh resolution. The combined feature detection-error estimation approach restricts mesh refinement to important coherent structures in the flowfield, an aspect that is particularly relevant for tip vortex resolution in fixed-wing and rotorcraft applications. This paper focuses on the development of the Richardson estimator, and implementation in the context of Cartesian-based off-body refinement.

Richardson extrapolation is commonly used to improve discrete functional approximations, but our approach uses a form of *interpolation* to compute the error between two solutions from overlapping grid levels. The error estimation process requires a coarse and (relatively) finer solution, which have been generated using the same time-marching process. After each time-step, the error may be computed simply as the difference between the two solutions. This error calculation is akin to computing the local truncation error, which approximates the discretization error, but which requires an exact solution along with its discrete equivalent. For most practical computational fluid dynamic (CFD) problems of interest, the exact solution is unknown, so that our approach approximates the exact solution with the discrete fine solution. Spatial convergence tests reveal that the Richardson estimator is on par with the local truncation error, even for very coarse meshes where the vortex may be under-resolved. Additional tests verify that the estimator is well behaved for both steady and unsteady flows, and a mass flux error is shown to be relatively insensitive to variations of vortical strength, as compared to other error functionals (such as pressure).

The capability of the approach is demonstrated using three cases, an advecting vortex, tip vortices from a NACA 0015 wing, and the wake of a quarter-scale V22 tilt rotor acoustic model (TRAM) rotor. For all cases, an error tolerance of 10^{-4} was found to adequately preserve the vortex system and offer relatively small error far downstream. Moreover, the results are in very good agreement with results obtained using a feature-only process with a fixed number of mesh levels. Overall, the proposed method shows good potential for an automated procedure for resolving vortical features in aerodynamics applications.

Future work will apply the developed approach for more complex rotorcraft applications, in particular those involving multiple components such as fuselage and tail rotor interaction.

Acknowledgments

Development was performed at the High Performance Computing (HPC) Institute for Advanced Rotorcraft Modeling and Simulation (HI-ARMS) located at the U.S. Army Aeroflightdynamics Directorate at Moffett Field, CA. Material presented in this paper is a

product of the CREATE-AV element of the Computational Research and Engineering for Acquisition Tools and Environments (CREATE) Program sponsored by the U.S. Department of Defense HPC Modernization Program Office. The authors gratefully acknowledge the contributions to this work by Jay Sitaraman and Dimitri Mavriplis, of the University of Wyoming, and Thomas Pulliam of the NASA Ames Research Center.

References

- [1] Venditti, D. A., and Darmofal, D. L., "A Multilevel Error Estimation and Grid Adaptive Strategy for Improving the Accuracy of Integral Outputs," AIAA Paper 1999-3292, Jan. 2008.
- [2] Park, M. A., "Adjoint-Based, Three-Dimensional Error Prediction and Grid Adaptation," AIAA Paper 2002-3286, 2002.
- [3] Wintzer, M., Nemec, M., and Aftosmis, M., "Adjoint-Based Adaptive Mesh Refinement for Sonic Boom Prediction," AIAA Paper 2008-6593, Aug. 2008.
- [4] Potsdam, M. A., and Strawn, R. C., "CFD Simulations of Tiltrotor Configurations in Hover," *Journal of the American Helicopter Society*, Vol. 50, No. 1, 2005, pp. 82–94. doi:10.4050/1.3092845
- [5] Caradonna, F. X., "Developments and Challenges in Rotorcraft Aerodynamics," AIAA Paper 2000-0109, Jan. 2000.
- [6] Nemec, M., Aftosmis, M. J., and Wintzer, M., "Adjoint-Based Adaptive Mesh Refinement for Complex Geometries," AIAA Paper 2008-725, Jan. 2008.
- [7] Duraisamy, K., Alonso, J. J., and Chandrasekhar, P., "Error Estimation for High Speed Flows using Continuous and Discrete Adjoints," AIAA Paper 2010-128, Jan. 2010.
- [8] Kamkar, S. J., Jameson, A. J., and Wissink, A. M., "Automated Grid Refinement using Feature Detection," AIAA Paper 2009-1496, Jan. 2009.
- [9] Kamkar, S. J., Jameson, A. J., Wissink, A. M., and Sankaran, V., "Feature-Driven Cartesian Adaptive Mesh Refinement in the Helios Code," AIAA Paper 2010-0171, Jan. 2010.
- [10] Hunt, J. C. R., Wray, A. A., and Moin, P., "Eddies, Streams, and Convergence Zones in Turbulent Flows," *Studying Turbulence using Numerical Simulation Databases, 2. Proceedings of the 1988 Summer Program*, NASA, Dec. 1988, pp. 193–208.
- [11] Jeong, J., and Hussain, F., "On the Identification of a Vortex," *Journal of Fluid Mechanics*, Vol. 285, 1995, pp. 69–94. doi:10.1017/S0022112095000462
- [12] Chong, M. S., Perry, A. E., and Cantwell, B. J., "A General Classification of Three-Dimensional Flow Fields," *Physics of Fluids*, Vol. 2, No. 5, May 1990, pp. 765–777. doi:10.1063/1.857730
- [13] Horiuti, K., and Takagi, Y., "Identification Method for Vortex Sheet Structures in Turbulent Flows," *Physics of Fluids*, Vol. 17, No. 12, Dec. 2005, pp. 1–4. doi:10.1063/1.2147610
- [14] Sankaran, V., Sitaraman, J., Wissink, A. M., Datta, A., Jayaraman, B., Potsdam, M., Mavriplis, D., Yang, Z., O'Brien, D., Saberi, H., Cheng, R., Hariharan, N., and Strawn, R., "Application of the Helios Computational Platform to Rotorcraft Flowfields," AIAA Paper 2010-1230, Jan. 2010.
- [15] Wissink, A. M., Sitaraman, J., Sankaran, V., Pulliam, T., and Mavriplis, D., "A Multi-Code Python-Based Infrastructure for Overset CFD with Adaptive Cartesian Grids," AIAA Paper 2008-927, Jan. 2008.
- [16] Sitaraman, J., Katz, A., Jayaraman, B., Wissink, A. M., and Sankaran, V., "Evaluation of a Multi-Solver Paradigm for CFD using Unstructured and Structured Adaptive Cartesian Grids," AIAA Paper 2008-660, Jan. 2008.
- [17] Jespersen, D., Pulliam, T., and Buning, P., "Recent Enhancements to OVERFLOW," AIAA Paper 1997-0644, Jan. 1997.
- [18] Buning, P., Gomez, R., and Scallion, W., "CFD Approaches for Simulation of Wing-Body Stage Separation," AIAA Paper 2004-4838, Aug 2004.
- [19] Wissink, A. M., Kamkar, S. J., Sitaraman, J., and Sankaran, V., "Cartesian Adaptive Mesh Refinement for Rotorcraft Wake Resolution," AIAA Paper 2010-4554, June 2010.
- [20] Hornung, R. D., and Kohn, S. R., "Managing Application Complexity in the SAMRAI Object-Oriented Framework," *Concurrency and Computation: Practice and Experience*, Vol. 14, No. 5, 2002, pp. 347–368. doi:10.1002/cpe.652
- [21] Hornung, R. D., Wissink, A. M., and Kohn, S. R., "Managing Complex Data and Geometry in Parallel Structured AMR Applications,"

- Engineering with Computers*, Vol. 22, No. 3, 2006, pp. 181–195.
doi:10.1007/s00366-006-0038-6
- [22] Wissink, A. M., Hornung, R. D., Kohn, S. R., Smith, S. S., and Elliott, N., “Large Scale Parallel Structured AMR Calculations Using the SAMRAI Framework,” *Supercomputing '01: Proceedings of the 2001 ACM/IEEE conference on Supercomputing (CDROM)*, Association for Computing Machinery, New York, 2001, p. 6-6.
- [23] Pulliam, T. H., “High Order Accurate Finite-Difference Methods: As seen in OVERFLOW,” *20th AIAA Computational Fluid Dynamics Conference*, Honolulu, HI, AIAA Paper 2011-3851, June 2011.
- [24] Berger, M., and Olinger, J., “Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations,” *Journal of Computational Physics*, Vol. 53, No. 3, 1984, pp. 484–512.
doi:10.1016/0021-9991(84)90073-1
- [25] Berger, M. J., and Colella, P., “Local Adaptive Mesh Refinement for Shock Hydrodynamics,” *Journal of Computational Physics*, Vol. 82, No. 1, May 1989, pp. 64–84.
doi:10.1016/0021-9991(89)90035-1
- [26] Wissink, A. M., Hysom, D. A., and Hornung, R. D., “Enhancing Scalability of Parallel Structured AMR Calculations,” *Proceedings of the 17th ACM International Conference on Supercomputing (ICS03)*, Association for Computing Machinery, New York, June 2003, pp. 336–347.
- [27] Gunney, B. T. N., Wissink, A. M., and Hysom, D. A., “Parallel Clustering Algorithms for Structured AMR,” *Journal of Parallel and Distributed Computing*, Vol. 66, No. 11, 2006, pp. 1419–1430.
doi:10.1016/j.jpdc.2006.03.011
- [28] Jameson, A., “Analysis and Design of Numerical Schemes for Gas Dynamics 1: Artificial Diffusion, Upwind Biasing, Limiters and Their Effect on Accuracy and Multigrid Convergence,” *International Journal of Computational Fluid Dynamics*, Vol. 4, No. 3, 1995, pp. 171–218.
doi:10.1080/10618569508904524; also Research Inst. for Advanced Computer Science (RIACS) Technical Rept. 94.15.
- [29] Jameson, A., Schmidt, W., and Turkel, E., “Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes,” AIAA Paper 1981-1259, June 1981.
- [30] Runge, C., “Über die Numerische Auflösung von Differentialgleichungen,” *Mathematische Annalen*, Vol. 46, 1895, pp. 167–178.
- [31] Kutta, W., “Beitrag zur Naherungsweise Integration Totaler Differentialgleichungen,” *Zeitschrift fur Angewandte Mathematik und Physik*, Vol. 46, 1901, pp. 435–453.
- [32] Zingg, D., and Chisholm, T., “Runge-Kutta Methods for Linear Problems,” AIAA Paper 1995-1756, June 1995.
- [33] Agarwal, R., and Dese, J., “An Euler Solver for Calculating the Flowfield of a Helicopter Rotor in Hover and Forward Flight,” AIAA Paper 1987-1427, June 1987.
- [34] Jorgenson, P., and Chima, R., “Explicit Runge-Kutta Method for Unsteady Rotor/Stator Interaction,” *AIAA Journal*, Vol. 27, No. 6, 1989, pp. 743–749.
doi:10.2514/3.10174
- [35] Yee, H., and Sweby, P., “Some Aspects of Numerical Uncertainties in Time-Marching to Steady-State Numerical Solutions,” AIAA Paper 1996-2052, June 1996.
- [36] Wray, A. A., *Minimal Storage Time Advancement Schemes for Spectral Methods*, NASA Ames Research Center, Moffett Field, CA, 1986 (unpublished).
- [37] Kamkar, S. J., Wissink, A. M., Sankaran, V., and Jameson, A., “Feature-Driven Cartesian Adaptive Mesh Refinement for Vortex-Dominated Flows,” *Journal of Computational Physics*, Vol. 230, No. 16, 2011, pp. 6271–6298.
doi:10.1016/j.jcp.2011.04.024
- [38] Kamkar, S. J., “Mesh Adaption Strategies for Vortex-Dominated Flows,” Ph.D. Thesis, Stanford Univ., Stanford, CA, 2011.
- [39] Erlebacher, G., Hussaini, M. Y., and Shu, C.-W., “Interaction of a Shock with a Longitudinal Vortex,” *Journal of Fluid Mechanics*, Vol. 337, 1997, pp. 129–153.
doi:10.1017/S0022112096004880
- [40] Roy, C. J., “Review of Discretization Error Estimators in Scientific Computing,” AIAA Paper 2010-126, Jan. 2010.
- [41] McAlister, K. W., and Takahashi, R. K., “NACA 0015 Wing Pressure and Trailing Vortex Measurements,” NASA Technical Paper 3151, AVSCOM Technical Report 91-A-003, Nov. 1991.

Z. J. Wang
Associate Editor