Verification of Unstructured Grid Adaptation Components

Marshall C. Galbraith,* Philip C. Caplan,† and Hugh A. Carson¶
Massachusetts Institute of Technology, Cambridge, MA 02139, USA

Michael A. Park,‡ Aravind Balan,§ and W. Kyle Anderson¶
NASA Langley Research Center, Hampton, VA 23681, USA

Todd Michal‖ and Joshua A. Krakos‖
Boeing Research & Technology, St. Louis, MO, USA

Dmitry S. Kamenetskiy‖
Boeing Research & Technology, Seattle, WA, USA

Adrien Loseille** and Frédéric Alauzet**
INRIA Paris-Saclay, Alan Turing Building, 91120 Palaiseau, France

Loïc Frazza††
Sorbonne Universités, UPMC Paris 06, 4 place Jussieu 75252 Paris cedex 05, France

Nicolas Barral‡‡
Imperial College London, South Kensington Campus, London SW7 2AZ, UK

Adaptive unstructured grid techniques have made limited impact on production analysis workflows where the control of discretization error is critical to obtaining reliable simulation results. Recent progress has matured a number of independent implementations of flow solvers, error estimation methods, and anisotropic grid adaptation mechanics. Known differences and previously unknown differences in grid adaptation components and their integrated processes are identified here for study. Unstructured grid adaptation tools are verified using analytic functions and the Code Comparison Principle. Three analytic functions with different smoothness properties are adapted to show the impact of smoothness on implementation differences. A scalar advection-diffusion problem with an analytic solution that models a boundary layer is adapted to test individual grid adaptation components. The scalar problems illustrate known differences in a grid adaptation component implementation and a previously unknown interaction between components. Laminar flow over a delta wing is verified with multiple,

*Research Engineer, Department of Aeronautics & Astronautics, AIAA Member.
†Graduate Research Assistant, Department of Aeronautics & Astronautics, AIAA Student Member.
‡Research Scientist, Computational AeroSciences Branch, AIAA Associate Fellow.
§NASA Postdoctoral Fellow, AIAA Member.
¶Senior Research Scientist, Computational AeroSciences Branch, AIAA Associate Fellow.
‖Engineer, AIAA Senior Member.
**Researcher, GAMMA3 Team, AIAA Member.
††PhD student, GAMMA3 Team, AIAA Member.
‡‡Research Associate, Department of Earth Science and Engineering.

independent grid adaptation procedures to show consistent convergence to fine-grid forces and pitching moment.

I. Introduction

The use of Reynolds-averaged Navier–Stokes (RANS) equations with a turbulence model has become a critical tool for the design of aerospace vehicles. However, the issues that affect the grid convergence of three dimensional ($3D$) configurations are not completely understood, as documented in the AIAA Drag Prediction Workshop series [1–3]. This led to an effort to verify the turbulence models with the Turbulence Modeling Resource (TMR) website [4]. Morrison, Kleb, and Vassberg [5] identified that "the DPW series does not have the systematic build up and definition on both the computational and experimental side that is required for detailed verification and validation." This has led to a focus on benchmark problems of increasing difficulty by Diskin et al. [6–9]. These benchmark problems provide ideal examples to evaluate unstructured grid adaptation methods because well-resolved solutions are available from a number of independent flow solvers on a series of carefully constructed, uniformly-refined grids. The International Workshop on High-Order CFD Methods [10] also provides a range of cases that are suitable to computing highly accurate solutions for verifying fluid simulations. Grid adaption methods have the potential for increasing the automation and discretization error control of RANS solutions to impact the aerospace design and certification process. To fulfill this potential, grid adaptation must also undergo a rigorous verification process by demonstrating spatial convergence behavior.

Alauzet and Loseille [11] documented the dramatic progress made in the last decade for solution-adaptive methods that includes the anisotropy to resolve simulations with shocks and boundary layers, and identify where improvements are needed for complex simulations. Park et al. [12] documented the current state of solution-based anisotropic grid adaptation and motivated further development with the impacts that improved capability would have on aerospace analysis and design in the broader context of the CFD Vision 2030 Study by Slotnick et al. [13]. The realization of the CFD Vision 2030 Study includes automated management of errors and uncertainties of physics-based, predictive modeling that can set the stage for ensuring a vehicle is in compliance with a regulation or specification using analysis without demonstration in flight test (i.e., certification by analysis).

Park et al. [14] separated the solution-adaptive process into a number of components (introduced in Section II) that can be independently evaluated and improved. Refining and documenting the evaluation methods is equally important as the test case descriptions to encourage old and new entrants in the grid adaptation development community to benchmark and compare implementation choices in a uniform and repeatable manner. An informal Unstructured Grid Adaptation Working Group (UGAWG) has been formed to continue this process as described in their first benchmark article [15], which focused on evaluating adaptive grid mechanics for analytic metric fields on planar and simple curved domains. The first benchmark contains a list of future directions, which includes the focus of this paper: separating the

solution-adaptive process into components (e.g., flow solver, error estimate, mesh adaptation mechanics) that can be examined by modifying or replacing one component with the remaining components fixed. Details of these components are provided in Sections III–VI.

The UGAWG evaluated the hemisphere cylinder and ONERA M6 wing cases of the Three Dimensional Benchmark Turbulent Flows [9] as documented by Park et al. [16, 17]. Michal et al. [18] also made an application of multiple anisotropic error estimation techniques to the ONERA M6 wing. Both of these references compared the results of integrated adaptation processes composed of different flow solvers, error estimation techniques, and adaptive grid mechanics. The convergence of forces, moments, and grid properties were compared. Grids resulting from one integrated adaptation process were examined with different flow solvers. This comparison has yielded an understanding of the properties of the integrated adaptation components and has allowed some best practices to be identified. However, to gain a deeper understanding of implementation choices and details, individual components of the process must be verified.

The verification and validation process is described in detail by Oberkampf and Roy [19]. Verification and validation is part of the AIAA Engineering Standards for CFD and Complex Aerospace Systems [20]. Oberkampf and Trucano specialize the discussion to CFD in Ref. [21]. The primary focus of this article is verification, where the question is asked, "has the model been implemented correctly?" The other question of validation, "is this the right model to use for this prediction?" has less emphasis in the current investigation. Trucano, Pilch, and Oberkampf [22] caution the use of the code-to-code comparisons in the context of rigorous verification exercises, but also indicate that there are benefits when the Code Comparison Principle (CCP) is used appropriately, i.e., comparison of two or more distinct and substantively different codes with the same algorithms. "Even given the philosophical limitations that we have stressed, benefits achieved from the use of the CCP for verification of complex problem numerical accuracy would likely increase if a rational methodology was consistently applied. [22]" The current work targets a rational application of code-to-code comparisons, where the reproducibility aspect of the scientific method is supported by documenting the outputs of multiple codes for a specified set of inputs [23, 24].

Verifying methods by comparison to a known set of inputs and outputs or independent implementations sets the stage for the further development of existing error estimation techniques and the creation of new methods. Oberkampf and Trucano [25] detail the desired properties of verification and validation benchmarks. The TMR website provides a successful model for the value of these benchmarks when they are provided in a manner that reduces the barriers to verification and validation exercises. A repository has been initiated for unstructured grid adaptation verification (`https://ugawg.github.io/`) that can be further refined and improved by the results of this effort.

A verification process for controlling the interpolation error of scalar functions is described in Section VII. This verification process is extended to a scalar advection-diffusion partial differential equation (PDE) in Section VIII to verify the convergence order properties of interpolation and output error. These interpolation and output error

verification processes are further extended to a wing example with laminar flow in Section IX, where an analytic function is not available for the solution. The force and moment trajectories of the solution-adaptive processes illustrate the convergence of the adaptive processes to low discretization error levels.

II. Integrated Grid Adaptation Method

The components of metric-based anisotropic unstructured grid adaptation are shown on Fig. 1. Starting with an initial grid, a flow solution (and optionally, an adjoint solution) is computed. The information from these solutions are used to estimate discretization error and specify a new grid resolution request via an anisotropic metric field $\mathcal{M}$. If the estimated errors are larger than limits specified by the practitioner, the current grid system is modified by grid mechanics to conform to the anisotropic metric $\mathcal{M}$. Once the adapted grid is available, the previous flow solution is optionally interpolated to the new grid to provide an initial condition for the flow solver that approximates the converged solution. This improved initial condition can decrease the execution time and improve the robustness of the flow solution calculation, but standard initialization is also possible. The process is repeated until exit criteria are met (e.g., accuracy requirement, resource limit). There are potential interactions between each of these elements that impact the overall convergence and efficiency of the adaptation process. Details of the specific implementations of these components is detailed in the following sections.

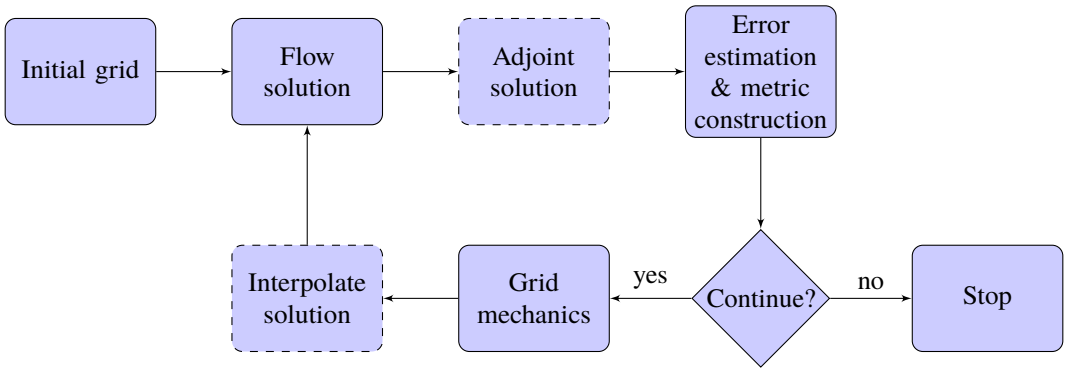


Fig. 1 Solution-based grid adaptation process with optional components indicated by dashed outlines.

A metric field, $\mathcal{M}$, can be constructed several ways to encode anisotropic information. Loseille and Alauzet [26] provide a thorough introduction to the metric tensor field. Here we consider two kinds of metric fields: multiscale and output-based metric fields. The multiscale metric controls the L^p norm of the interpolation error of a solution scalar field and forms the foundation of some output-based metrics [26, 27]. Output-based metric formulations [28] include the adjoint solution, which allows the targeting of a goal or functional output (e.g., lift, drag).

Grid adaptation is an inherently nonlinear process. The robustness of the adaptive procedure is enhanced by optimizing the grid at a fixed-complexity, which allows for control of Degrees of Freedom (DOF). The complexity, C,

of a continuous metric field, $\mathcal{M}$, is defined as the integral,

$$C(\mathcal{M}) = \int_{\Omega} \sqrt{\det(\mathcal{M}(x))}\, dx, \tag{1}$$

and is evaluated on the discrete grid and metric. The relationship between C and the number of vertices and elements in the adapted grid is shown theoretically in Ref. [26] and experimentally in Refs. [14, 29].

An adaptive series of grids is created by optimizing the metric at a fixed complexity and then increasing the target complexity in a series of steps. The progression of forces and moments during these steps is referred to as a trajectory in the result sections. Typically, 5 to 10 fixed-complexity adaptation iterations are performed before increasing the complexity to the next target. These fixed-complexity adaptations are referred to as subiterations in Michal et al. [18], where only the final force or moment value is plotted for the fixed-complexity target. Alternatively, an average of a number of fixed-complexity adaptations can be used to reduce the scatter in trajectories.

III. Flow Solvers

Multiple flow solvers are employed to compute the flow solution in the solution-based grid adaptation process. The details of the discretization and nonlinear solution scheme can impact the performance of the flow solver component on the integrated grid adaptation process, in particular, on highly anisotropic grids. A strong nonlinear update scheme can positively impact the overall robustness of the process and a low-diffusion discretization can approximate numerical solutions on coarser grids better than higher-diffusion schemes.

A. SANS

Solution Adaptive Numerical Simulator (SANS) [30], currently under development at the Massachusetts Institute of Technology, is a general framework for solving discrete finite-element approximations to advection-diffusion-reaction type PDEs, such as scalar advection-diffusion, Navier-Stokes, and RANS equations. A range of finite-element methods are currently implemented in SANS, including high-order discontinuous (DG), hybridized discontinuous (HDG), and continuous (CG) Galerkin finite-element methods. Boundary conditions are weakly enforced and forces are computed using the residual balance at the boundary.

The nonlinear system of equations is solved using pseudotime continuation (PTC) damped Newton's method with a line search to ensure residuals decrease. The complete linearization of the residual is computed via operator overloaded automatic differentiation [31]. The PTC algorithm computes an element local time step based on the characteristic speed, element size, and a Courant-Friedrichs-Lewy (CFL) number. The inverse CFL is driven toward zero such that a Newton-like convergence rate is recovered. The Portable, Extensible Toolkit for Scientific Computation (PETSc) [32–34] framework is used to solve the linear system for each PTC iteration with restarted generalized minimal residual

(GMRES) [35] preconditioned with an Incomplete Lower Upper (ILU) factorization. Parallel computations use the restricted additive Schwarz preconditioner with a single layer of overlap. The ILU preconditioner is applied to each subdomain, and restarted GMRES is applied to the global system. Adjoint systems are solved using the same linear solver as the primal. All discrete solutions are converged to near machine-zero residuals.

B. FUN3D-SFE

FUN3D-SFE [36] is a continuous Stabilized Finite-Element discretization within the FUN3D (Fully-Unstructured Navier-Stokes 3D) framework [37, 38]. The stabilization options include the Streamlined Upwind Petrov-Galerkin (SUPG) scheme [39, 40], Galerkin Least-Squares (GLS) [41], and variational multiscale methods [42]. In the results shown here, only the SUPG scheme is considered. The boundary conditions are enforced weakly and forces are computed directly from the residual routines within FUN3D-SFE. A linear nodal basis is used in this study, which is designed to be 2^{nd}-order accurate in space. The current implementation includes the capability for computing on tetrahedra, hexahedra, pyramids, and prisms, although all the results shown in the present paper are for purely-tetrahedral grids.

To advance the solution of the Navier-Stokes equations to a steady state, density, velocities, and temperature are updated with a Newton-type solver described by Anderson, Newman, and Karman [36]. Here, an initial update to the flow variables is computed using a locally varying time-step parameter that is later multiplied by the current CFL number, which is adjusted during the iterative process as described in the next paragraph. At each iteration, a linearized residual matrix is formed and solved using the GMRES algorithm with a preconditioner based on an ILU decomposition with two levels of fill [43] and a Krylov subspace dimension of 300.

Using the full update of the variables, the L^2 norm of the unsteady residual is compared to its value at the beginning of the iteration. If the L^2 norm after the update is less than one half of the original value, the CFL number is doubled and the iterative process continues to the next iterative cycle. If the L^2 reduction target for the residual is not met, a line search is conducted to determine an appropriate relaxation factor. Here, the L^2 norm of the residual is determined at four locations along the search direction and the optimal relaxation factor is determined by locating the minimum of a cubic polynomial curve fit through the samples. After the line search, the solution is updated using the relaxation factor and the CFL number is neither increased nor decreased.

C. GGNS

GGNS (General Geometry Navier-Stokes) is a Boeing-developed flow solver built upon the SUPG finite-element discretization. The code uses piecewise linear finite elements resulting in a 2^{nd}-order accurate discretization. The solver uses unstructured grids of mixed-element type (tetrahedrons, prisms, and pyramids) as well as purely-tetrahedral grids. The number of DOF for the 2^{nd}-order SUPG scheme is equal to the number of vertices in the computational grid. The discretization is vertex-based in the sense that it is conservative over the dual volumes of an unstructured grid.

More details on discretization used in the GGNS solver, including the particular choices of discretization variables and special treatment of the essential boundary conditions via the Lagrange-multiplier based technique [44], can be found in Kamenetskiy et al. [45]. Forces are computed using the residual balance at the boundary.

The discrete nonlinear solver in the GGNS code implements a variant of the Newton-Krylov-Schwarz algorithm using PETSc. Time stepping is employed to drive to the steady state solution. On each time step, an exact Jacobian matrix for the discretization is formed by an automatic differentiation technique. The linear system arising from the Newton's method is approximately solved using GMRES with a drop-tolerance-based block-ILU preconditioner (locally on subdomains) implemented in the context of the additive Schwarz method with minimal overlap [43]. Right preconditioning is employed to maintain consistency between the nonlinear and linear residuals. The compact stencil property of the SUPG scheme helps to reduce the fill-in levels in the approximate factorization, thereby reducing the memory footprint.

A line search is applied along the direction provided by the approximate solution of the linear system. Residual decrease and physical realizability of the updated state are tracked during the line search. A heuristic feedback algorithm is implemented to communicate failure of the line search back to the time-stepping algorithm, so that the CFL number can be increased or decreased as necessary. There is no upper preset limit for the CFL number in the time-marching algorithm; so Newton-type quadratic convergence (or, at least, superlinear, due to inexact linear solves) is routinely achieved at steady state.

IV. Grid Mechanics

The following anisotropic grid mechanics packages are used to modify the grid to conform to a given metric field $\mathcal{M}$. The goal is to create a unit grid [26], where the edges are unit-length and the elements are unit-volume with respect to the given metric. These tools adopt a local modification approach, where a valid input grid is transformed into a more metric conforming valid output grid through a series of local operators. These tools differ in the local operations that are considered (e.g., element split, element collapse, element reconnection, vertex relocation) and the criteria used to apply these local operations.

A. avro

`avro` is a dimension-independent anisotropic mesh adaptation package under development at the Massachusetts Institute of Technology and has been demonstrated on up to $4D$ mesh generation problems [46]. It is based on local cavity operators inspired by the work of Coupez [47] and Loseille [48] and uses a combination of edge split, edge collapse, edge swaps, facet ($(D-1)$-simplex) swaps and vertex smoothing to conform to a prescribed metric. The emphasis in `avro` is to construct edge lengths that conform to a requsted metric field for an input mesh with bounded edge length bounds, which are often satisfied in an adaptive framework. In particular, it is designed to accept input edge

lengths that are within $[1/\alpha, \alpha]$ for $\alpha = 2$, but also works well for $\alpha = 4$. `avro` associates each mesh vertex with an EGADS [49] geometry entity in order to check the topological validity of each mesh operator and for projecting vertices to the geometry. Operators are sequenced by collapsing short edges followed by splitting long edges such that splits do not create short edges. Swaps are interleaved within splits and collapses to improve the quality of the mesh and escape topological configurations that restrict the ability to split or collapse. These operators are also scheduled such that the target metric complexity is also recovered.

B. `refine`

The `refine` open source anisotropic grid adaptation mechanics package was developed at NASA. It is available via `github.com/NASA/refine` under the Apache License, Version 2.0. The current version under development uses the combination of edge split and collapse operations proposed by Michal and Krakos [50]. Vertex relocation is performed to improve adjacent element shape. A new ideal vertex location of the vertex is created for each adjacent element. A convex combination of these ideal vertex locations is chosen to yield a new vertex location update that improves the element shape measure in the anisotropic metric [51]. The vertices of elements with poor shape measures are also relocated with the nonsmooth optimization of Frietag and Ollivier-Gooch [52]. Geometry is accessed through the EGADS application program interface, and parallel execution is facilitated by EGADSlite [53].

C. PRAgMaTIc

PRAgMaTIc (Parallel anisotRopic Adaptive Mesh ToolkIt) is an open source $2D$ and $3D$ anisotropic adaptation package developed as a C++ library at Imperial College London, `https://meshadaptation.github.io`. PRAgMaTIc modifies the input grid through a series of local edge-based grid manipulations [15]. First, iterative applications of coarsening (edge collapse), edge/face swapping, and refinement (edge splitting) is used to optimize the resolution and the quality of the grid. Then, an element-shape-constrained Laplacian smoothing step fine-tunes the grid element shape measure. PRAgMaTIc aims at generating quality grids for a wide range of numerical simulations, notably for geophysics applications, and it has been integrated with the PETSc and the Firedrake solver suite [54, 55].

D. EPIC

The EPIC anisotropic grid adaptation package developed at Boeing provides a modular framework for anisotropic grid adaptation that can be linked with external flow solvers [50]. EPIC relies on repeated application of edge break, edge collapse, element reconnection and vertex movement operations to modify a grid such that element edge lengths match a given anisotropic metric tensor field. The EPIC-ICS variant includes only edge insertion, edge collapse, and element swaps. The EPIC-ICSM variant adds vertex movement. The metric field on the adapted grid is continuously interpolated from the initial metric field. Several methods are available to preprocess the metric so as to limit minimum and maximum local metric sizes, control stretching rates of metric size and/or anisotropy, and ensure smoothness of the

resulting distribution. In addition, the metric distribution can be limited relative to the initial grid and/or to the local geometry surface curvature. The surface grid is maintained on an IGES geometry definition with geometric projections and a local regridding procedure.

E. FEFLO.A

FEFLO.A is a $2D$, $3D$, and surface mesh adaptation tool. It uses a combination of generalized standard operators (e.g., insertion, collapse, swap of edges and faces). The generalized operators are based on recasting the standard operators in a cavity framework [48, 56]. The cavity operator allows a simultaneous application of multiple standard operator combinations. Quality improvements are attained with the cavity operator that are not possible through a sequential application of standard operators. To increase robustness, the surface and volume mesh are modified simultaneously and each local modification is checked to verify that a valid mesh is maintained. For the volume, validity consists of checking that each newly created element has a strictly positive volume. For the surface, validity is checked by ensuring that the deviation of the geometric approximation with respect to a reference surface mesh remains within a given tolerance. During surface remeshing, new vertex locations are either evaluated with a cubic surface representation or an EGADS geometry query.

V. Multiscale Metric

The multiscale metric controls the L^p-norm of the interpolation error of a solution scalar field [57]. The (reconstructed) Hessian, $\mathcal{H}$, of a scalar field, is locally scaled by the Hessian determinant and globally scaled to a specified target to get the metric field,

$$\mathcal{M}_{L^p} = D_{L^p} \det(\mathcal{H})^{\frac{-1}{2p+d}} |\mathcal{H}|, \tag{2}$$

where the global scaling D_{L^p},

$$D_{L^p} = \left(\frac{C_t}{C\left(\det(\mathcal{H})^{\frac{-1}{2p+d}} |\mathcal{H}| \right)} \right)^{2/d}, \tag{3}$$

corrects the complexity of the locally scaled Hessian to produce $\mathcal{M}_{L^p}$ with specified target complexity C_t. Both scaling operations depend on the dimensionality of the domain, which is $d = 3$ in this case. A grid conforming to $\mathcal{M}_{L^p}$ provides optimal control of the scalar field interpolation error in the p-norm. A lower p-norm targets weaker variations of the scalar field and a larger p-norm targets rapid variations of the scalar field. A number of multiscale metric calculation methods are described below that have subtle differences in how the multiscale metric is formed (e.g., Hessian reconstruction, limits on metric gradation).

A. `refine` Multiscale Metric

To form the metric, a Hessian of the scalar field can be reconstructed by a k-exact quadratic reconstruction [58] or recursive application of L^2-projection [27]. The k-exact reconstruction is formed over vertices that are neighbors of neighbors of the vertex having its Hessian reconstructed. The least-squares system is solved with QR factorization, where the linear system is decomposed into an orthogonal matrix Q and an upper triangular matrix R [37]. If the least-squares system is underdetermined or poorly conditioned, the reconstruction stencil is iteratively grown one additional layer of vertex neighbors until a well-conditioned least-squares system is formed. No special boundary treatment is employed, but the one-sided stencil created on boundaries can contribute to poor conditioning that is mitigated with reconstruction stencil growth.

The Hessian can also be reconstructed by recursive application of the L^2-projection gradient reconstruction scheme. The gradient is computed in each element and a volume-weighted average is collected at each vertex [27]. The 2^{nd}-derivative Hessian terms are formed by computing the reconstructed gradients of these gradients formed in the first pass. The mixed derivative terms of the Hessian are averaged. A special boundary treatment is employed. The reconstructed Hessian on the boundary is replaced with an extrapolation from neighboring interior vertices, which have a well-formed stencil.

The reconstructed Hessian is then diagonalized into eigenvalues and eigenvectors. The absolute value of the Hessian is formed by recombining the absolute value of the eigenvalues with eigenvectors to ensure the Hessian is symmetric positive definite. The Hessian at each vertex is scaled to control the L^p norm [27] with Eq. (2). The gradation of the metric field is limited isotropically in the metric space with the "metric-space-gradation" of Ref. [59]. The complexity is computed, and the metric is globally scaled to set its complexity to a specified value. The complexity Eq. (1) is evaluated discretely by assuming it is piecewise constant in each median dual.

B. Firedrake+PRAgMaTIc Multiscale Metric

The PRAgMaTIc remeshing library was integrated with the Firedrake solver suite [54, 55]. In this study, PRAgMaTIc is called via its Firedrake interface, and the metric computation is done with Firedrake. A weak finite-element formulation for the Hessian $\mathcal{H}$ of the scalar field u is written:

$$\mathcal{H} = \nabla^2 u\,,$$

hence

$$\int_\Omega \left(\sigma \cdot \mathcal{H} - \sigma \cdot \nabla^2 u\right) \mathrm{d}V = 0\,, \forall \sigma \in \Sigma\,,$$

where $\sigma \in \Sigma$ are test functions. This is then integrated by parts into:

$$\int_\Omega (\sigma \cdot \mathcal{H} + \mathrm{div}(\sigma) \cdot \nabla u)\, \mathrm{d}V = \int_{\partial\Omega} \mathbf{n} \cdot (\sigma \cdot \nabla u) \mathrm{d}S,$$

where $\mathbf{n}$ is the outward normal. The problem is specified using Firedrake's high-level Unified Form Language (UFL) and discretized automatically. The numerical problem is then solved using the PETSc library. There is currently no additional treatment of the Hessian on the boundary.

The Hessian is then symmetrized by taking the average value of opposite nondiagonal elements, and made positive by taking the absolute value of the eigenvalues. Then the multiscale metric described in Section V is computed, and the eigenvalues of the metric are truncated to a specified minimal size. Gradation of the metric sizes can be limited following the isotropic method from [59]; however, no gradation was performed in this study.

C. GGNS+EPIC Multiscale Metric

The Mach Hessian for each element is evaluated from the flow solution by using a least-squares approach on an extended stencil in GGNS. GGNS then passes the Hessian at each element to EPIC, which converts it to adaptation metrics via an element-centered modification of Alauzet and Loseille [27], which minimizes the L^p norm of interpolation error of the scalar field for a given grid complexity. In this modification, each elemental Hessian is scaled to control the L^p norm with Eq. (2). The global scaling factor, D_{L^P}, is initialized as Eq. (3). When enabled, the metric gradation is limited as detailed in the EPIC description, Section IV.D. The complexity, Eq. (1), of the resulting elemental adaptive metric is computed and the global scale factor, D_{L^P}, is adjusted to better match the requested value. The metric is then iteratively recomputed until the computed complexity is within a specified tolerance of the requested value. A continuous metric field is generated by Log-Euclidean [60] interpolation of the elemental metrics to the grid vertices.

VI. Output-based Metric

The metric field can be formulated to target a specific goal or output of the simulation by using information from the dual or adjoint problem. Output-based grid adaptation approaches are reviewed by Fidkowski and Darmofal [28] and a number of optimal-goal or output-based metric construction methods are examined here.

A. MOESS Output-based Metric

The Metric Optimization via Error Sampling and Synthesis (MOESS) [61, 62] adaptation algorithm is based on the continuous mesh framework developed by Loseille and Alauzet [26, 29]. Galbraith, Allmaras, and Darmofal [30] detail the implementation of MOESS in SANS. The output error estimation method used within MOESS is the Dual Weighted Residual (DWR) method, as originally devised by Becker and Rannacher [63]. DWR consists of computing an adjoint solution for an output functional in an enriched solution space, and weighting it against the residual to give an estimate

of $\mathcal{J}(u) - \mathcal{J}(u_{h,p})$ for some output functional $\mathcal{J}(u)$. Originally devised for the CG discretization, this approach was extended by Carson et al. [64] for the DG Bassi-Rebay (BR2) discretization wherein an additional adjoint for the BR2 lifting operator was developed to account for errors in the lifting operator residual. The SANS CG local error estimate is based on the work of Richter and Wick [65], where a nodal partition of unity localizes the DWR error estimate.

The MOESS algorithm [62] constructs a set of elemental local models that approximate the change in a localized error estimate for an output of interest as a function of a step matrix change to the implied metric of the grid. These local models are fit from local solves where an element κ is locally refined, either isotropically or by splitting an edge, and an approximate solve is performed. A local solve consists of fixing the DOF outside of a patch, $\omega_\kappa \supseteq \kappa$ whilst allowing those in the patch to vary. For a DG local solve, $\omega_\kappa \equiv \kappa$ but for a CG local solve, ω_κ consists of the elements attached to the vertices of κ. The localized error estimate is then reevaluated using the result of this local solve, which produces a change in the error estimate as a function of the step matrix change to the element. The local solve is repeated for multiple different refinements of the element and the results for all the local refinements are synthesized into a local model for that element. This is repeated for all the elements of the mesh and the sum of these local models is optimized subject to a maximum DOF constraint as well as constraints on the magnitude of the step matrices. The optimized step matrices are applied to the implied metric to create the new metric request that is supplied to a grid mechanics tool.

B. GGNS+EPIC Output-based Metric

The GGNS output metric was introduced by Michal et al. [18]. For the evaluation of the local error estimate, piecewise linear reconstruction is used to represent the adjoint solution and to obtain its gradient. For the 2^{nd}-order derivatives, Zienkiewicz-Zhu type patch recovery reconstruction for the gradients [66] is first obtained at the vertices of the grid and then these are linearly interpolated inside the elements. The interpolation error for the primal solution is not evaluated directly, instead, we rely on a semiheuristic approach when the Hessian of the Mach number field is provided to the adaptation module. The Mach field is the only source of metric anisotropy in the present approach; in particular, any information about the adjoint solution enters the error indicator through the isotropic weight. To evaluate the Hessian of the primal solution, the k-exact least-square approach based on an extended stencil reconstruction is involved resulting in the quadratic reconstruction. The local error indicator along with the described implementation has strong similarities to the well known Venditti–Darmofal [67] approach but is not exactly equivalent to the latter.

VII. Verification of Adaptation to Scalar Fields

To verify the implementation of the multiscale metric, the MOESS algorithm, and mesh mechanics implementations, three scalar analytic functions are chosen that stress the adaptation process to match both isotropic and anisotropic metric fields. The three functions `sinfun3`, `tanh3`, and `sinatan3`, progress in anisotropy to challenge the algorithms of each tool in different ways. For each analytic scalar field, s, a discrete approximation, $s_{h,p}$, is computed via an L^2

approximation,

$$0 = \int_{\Omega} \phi \left(s - s_{h,p} \right) \quad \forall \phi \in \mathcal{V}_{h,p}, \tag{4}$$

where p signifies the polynomial degree of the discrete approximation. Sequences of successively refined grids are used to verify that the L^2-error

$$\eta = \sqrt{\int_{\Omega} \left(s - s_{h,p} \right)^2}, \tag{5}$$

asymptotically decays at a rate of $p + 1$. All functions are evaluated on a unit cube domain $\Omega \equiv [0, 1] \times [0, 1] \times [0, 1]$.

The multiscale metric estimates the quadratic interpolation error of linear elements via a Hessian reconstruction. Therefore, all results in this section use a continuous linear, $p = 1$, polynomial approximation of the scalar functions to compute multiscale metric fields. The `refine` multiscale metrics are computed in the 2-norm with gradation limited to 5 based on either k-exact or L^2-projection Hessian reconstruction.

Rather than targeting interpolation error, the MOESS algorithm seeks to minimize an error functional. Typically, the DWR is used as the error functional, but this only provides an approximate error estimate. To eliminate the approximation in the error functional for this verification exercise, the MOESS algorithm is modified to minimize the square L^2-error directly. The local solves requires the error functional to be localized to each element, and the square L^2-error localized to the element κ is simply,

$$\eta_{\kappa} = \int_{\kappa} \left(s - s_{h,p} \right)^2 . \tag{6}$$

The MOESS algorithm is exercised for both continuous and discontinuous finite-element solution spaces. For a discontinuous solution space, the L^2 approximation is decoupled between elements and thus, the local solves are exact (up to quadrature error). However, for a continuous solution space, L^2 approximation is a global operation, and there remains some approximation in the local solve. As the MOESS algorithm does not make any assumption about the polynomial degree, all sequences of grids for each scalar function are produced with polynomial degrees of $p = 1$, $p = 2$, and $p = 3$.

A series of target complexities is used to demonstrate the asymptotic $p + 1$ convergence rate for both the multiscale- and MOESS-generated metric fields. There are no specific requirements on how this series of complexities is constructed, but the meshes must be fine enough to capture the asymptotic convergence rate. Here, the sequence of element counts are derived using a discontinuous space from a sequence of DOF counts. As shown in Table 1, the number of elements for a given DOF count decreases with increasing p. The target complexity is set by multiplying an element count by the volume of a unit-length tetrahedron, i.e., $\sqrt{2}/12$. Ideally, if the adapted meshes have perfect unit-length tetrahedron as measured under the metric, the resulting grid should match the target number of elements. However, because unit-length tetrahedron do not tile or fill space, the observed ratio of complexity to elements in practice is typically $1/12$–$1.1/12$ as

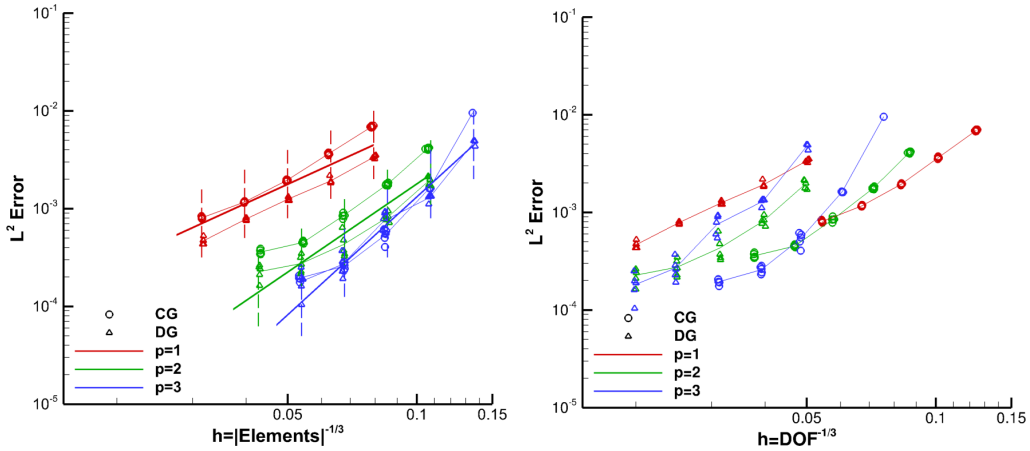
Table 1 DG element counts for given DOF counts.

	DG Elements		
DOF	$p = 1$	$p = 2$	$p = 3$
4,000	1,000	400	200
8,000	2,000	800	400
16,000	4,000	1,600	800
⋮	⋮	⋮	⋮
2,048,000	512,000	204,800	102,400

demonstrated in Refs. [29] and [14]. Thus, the element count of the adapted meshes is not expected to perfectly match the target element count.

Nearly all the results in the following sections use SANS to compute the L^2 projections. The exceptions are the meshes adapted with PRAgMaTIc, where the L^2 projection is computed with Firedrake. Thus, some differences between PRAgMaTIc and other mesh adaptation tools can be attributed to differences between SANS and Firedrake. The multiscale metrics are all generated using the algorithm implemented with `refine` or Firedrake, and MOESS metrics are generated using the algorithm implemented in SANS. All adaptation sequences for a given complexity start with a uniform mesh with 750 elements (uniform $5 \times 5 \times 5$ hexes divided into tetrahedron). The adaptation loop is repeated 30 times for each target complexity.

Two reasonable choices as the representative length scale to measure the error convergence rate are $\text{DOF}^{-1/3}$ and $|\text{Elements}|^{-1/3}$. For a finite-element discretization, the DOF count represents the cost to solve the linear system of an implicit discretization, and the element count is proportional to the residual or Jacobian evaluation cost. Since the purpose of this section is the verification of the metric calculation and mesh adaptation tools, the convergence rate is illustrated using the number of elements. The element count clusters the data when comparing continuous and discontinuous solution spaces, as illustrated in Fig. 2. In addition, the target element count is exactly represented with the dashed vertical lines in Fig. 2(a), whereas the target DOF count for the continuous space can only be approximated.



(a) Element count as reference length.

(b) DOF count as reference length.

Fig. 2 DOF vs. Elements as reference length.

The figures in this section that include multiple mesh adaptation tools show the averages of last five L^2-error values in the adaptation sequence. The figures that only include one mesh adaptation tool show the last five L^2-error values in the adaptation sequence as demonstrated in Fig. 2. The lines in these figures also connect the average error values between target complexities. All figures with L^2-errors include reference lines that illustrate the expected $p+1$ convergence rates.

A. `sinfun3`

The `sinfun3` function,

$$
\begin{aligned}
xyz &= (x-0.4)(y-0.4)(z-0.4) \\
s &= \begin{cases} 0.1\sin(50.0\,xyz) & \text{if} \quad xyz \le -1.0\pi/50.0 \\ \sin(50.0\,xyz) & \text{if} \quad xyz \le 2.0\pi/50.0 \\ 0.1\sin(50.0\,xyz) & \text{else} \end{cases},
\end{aligned} \tag{7}
$$

shown in Fig. 3, has a smooth variation that is mostly isotropic. The convergence of the L^2-errors for all the mesh adaptation tools and the multiscale algorithms are shown in Fig. 4. The multiscale algorithms differ mostly in the way the Hessian is evaluated. As discussed in Section V, the Hessian can be calculated by recursive L^2-projection, k-exact reconstruction or by a finite element formulation as implemented in Firedrake. The L^2-errors are all converging at the expected 2$^{\text{nd}}$-order rate. Small differences can be observed between the mesh adaptation tools or the two multiscale algorithms implemented with `refine` and the multiscale metric computed with Firedrake, as is expected on this smooth problem. Similar results are observed for the MOESS algorithm as shown in Fig. 5. All L^2-errors are converging at the

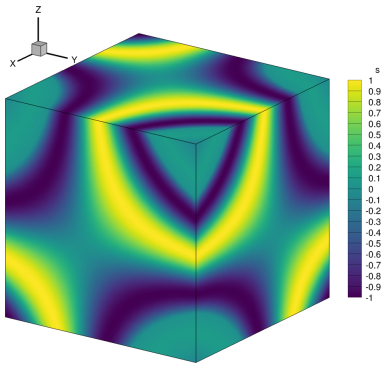
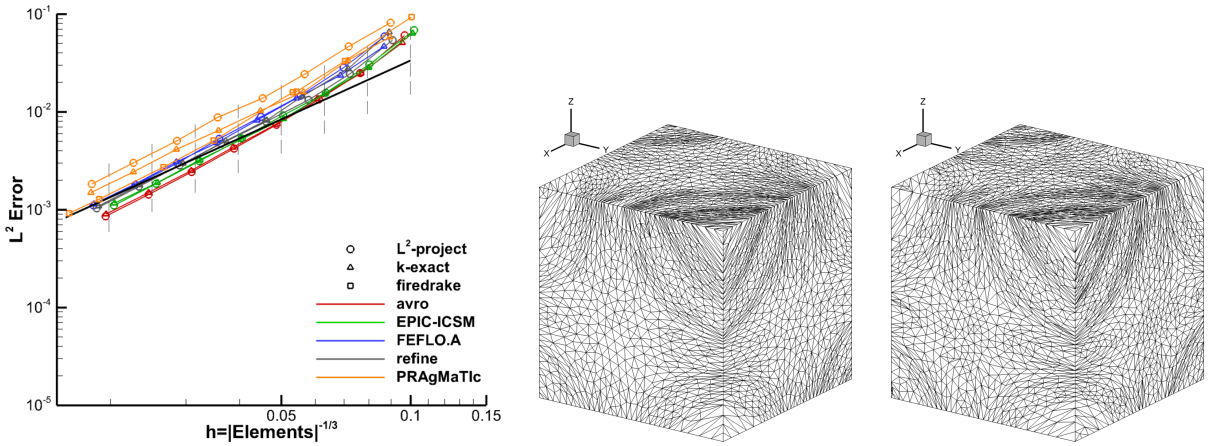


Fig. 3 `sinfun3` scalar function on a cube.

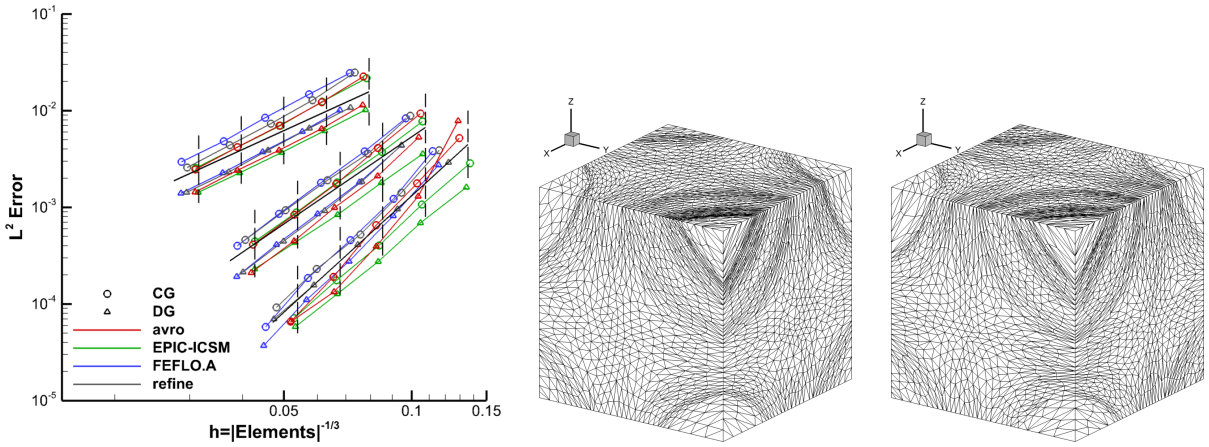
expected rate of $p + 1$. Separate convergence rates and grids adapted with the respective tools are shown in the appendix of Ref. [17].

While the expected convergence rates are achieved, the grids adapted to the `sinfun3` created neighbor of neighbor k-exact Hessian reconstruction stencils that were ill-conditioned. This prompted an extension of the k-exact reconstruction algorithm to improve the conditioning by growing the stencil to include additional layers in the `refine` multiscale metric implementation.



(a) Convergence: reference line slope of 2. (b) L^2 target 128,000 elements. (c) k-exact target 128,000 elements.

Fig. 4 `sinfun3` L^2-error multiscale metric convergence. Example meshes adapted with `refine`.



(a) Convergence: reference line slopes of $p + 1$. (b) $p = 1$ CG target 128,000 elements. (c) $p = 1$ DG target 128,000 elements.

Fig. 5 `sinfun3` L^2-error MOESS convergence. Example meshes adapted with FEFLO.A.

B. `tanh3`

The `tanh3` function,

$$s = \tanh((x + 1.3)^{20}(y - 0.3)^{9}z), \tag{8}$$

shown in Fig. 6, has strong anisotropic regions including a boundary layer feature on the $z = 0$ face. The convergence

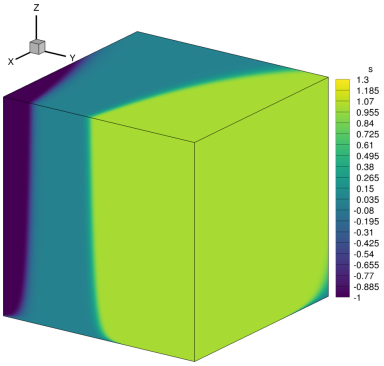


Fig. 6 `tanh3` scalar function.

of the L^2-error for all the mesh adaptation tools with the multiscale algorithms are shown in Fig. 7. Due to the anisotropy in the `tanh3` function, larger grid sizes are required to reach the 2nd-order asymptotic rate. The outliers are the meshes adapted with PRAgMaTIc and the Firedrake multiscale algorithm. From inspection of the grids shown in Fig. 8, PRAgMaTIc with the Firedrake multiscale metric produces lower anisotropy on the boundary than PRAgMaTIc with `refine` L^2-projection or `refine` k-exact multiscale metrics. Since Hessian recovery degrades on the boundary, interpolation error may be higher on the boundary without a specialized treatment of the recovered Hessian on the boundary. Metric gradation helps to create an anisotropic boundary mesh, but at the cost of an increased

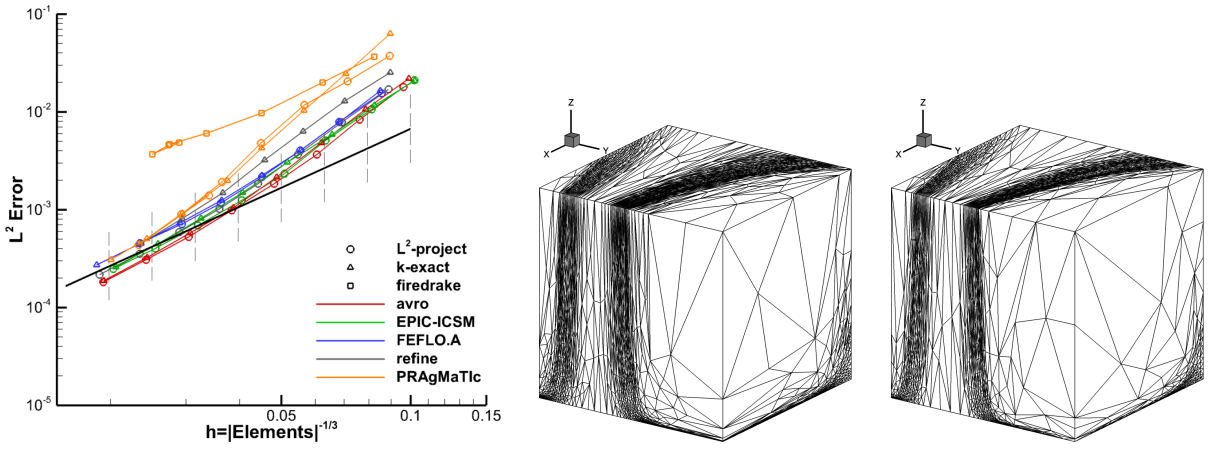
DOF count. Despite this deficiency, the L^2-error using the Firedrake multiscale metric converges at a 2^{nd}-order rate, but at significantly higher errors than using the `refine` multiscale metrics, which extrapolate the interior Hessian recovered with L^2-projection to the boundary or use k-exact reconstruction. This demonstrates the value of code-to-code comparisons to aid in identifying deficiencies.

The L^2-errors using the MOESS algorithm are shown in Fig. 9. The L^2-errors cluster reasonably well around the 2^{nd}-order reference line. The clustering is not as clear around the 3^{rd}- and 4^{th}-order lines, but average convergence rates match expectations. The convergence rates for each mesh adaptation tool for both multiscale and MOESS metrics are shown in the appendix of Ref. [17].

The errors from `refine` in the CG space are significantly higher relative to the other mesh adaptation tools. As an example, the convergence rates obtained with `refine` compared with those obtained with EPIC-ICSM are shown in Fig. 10. Aside from the last point, the error rates computed with EPIC-ICSM agree well with the expected $p + 1$ rates. The `refine` results using the CG solution space are the only ones that exhibit a degraded rate and increased errors.

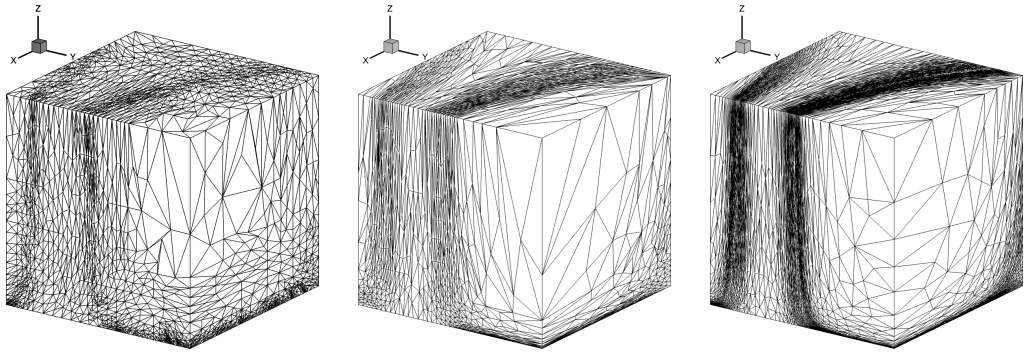
These higher error values appear to be due to isotropic elements in grids adapted with `refine` near the $z = 0$ boundary as shown in Fig. 10. Notably, the elements in this region are anisotropic using `refine` with the multiscale metric as shown in Fig. 7. The other mesh adaptation tools do give the expected anisotropy using the MOESS metrics, e.g., the grid adapted by EPIC-ICSM in Fig. 10. Thus, the issue here is likely due to the mesh adaption algorithm in `refine` rather than an issue with the MOESS algorithm.

One possible cause for this differing behavior with `refine` might be attributed to how the metric fields are generated with the multiscale algorithm relative to the MOESS algorithm. The multiscale metric is roughly a fixed target metric, as the multiscale algorithm uses Hessians from the given scalar field that is mostly influenced by the accuracy of the scalar approximation. The MOESS algorithm always computes a metric that is a perturbation from its interpretation of the implied metric of the background grid. Possible causes of `refine` failing to create the grid specified by the MOESS algorithm are differences in interpretations of metric edge lengths or that the MOESS requests are within a deadband of the `refine` metric-conforming tolerance.



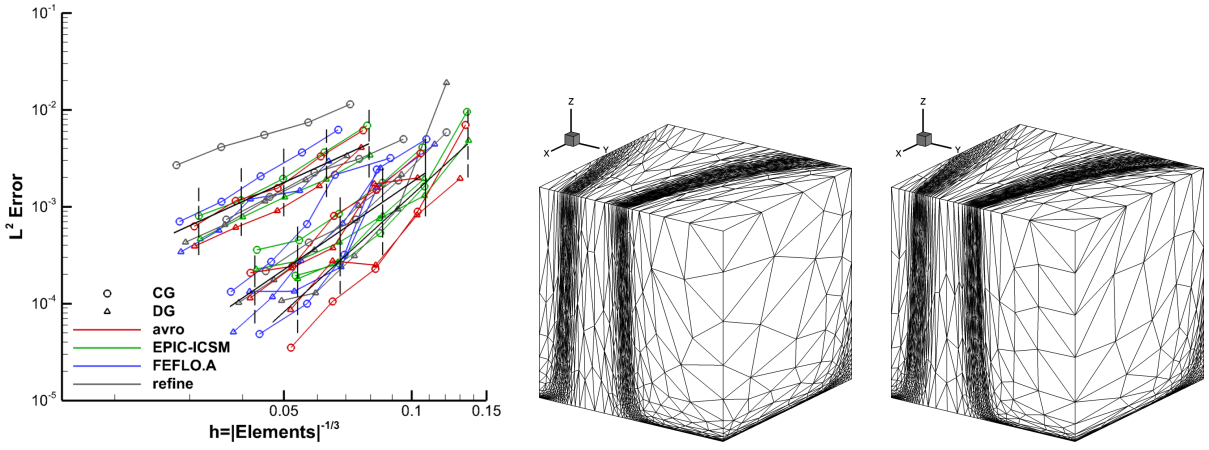
(a) Convergence: reference line slope of 2. (b) L^2 target 128,000 elements. (c) k-exact target 128,000 elements.

Fig. 7 `tanh3` L^2-error multiscale metric convergence. Example meshes adapted with `refine`.



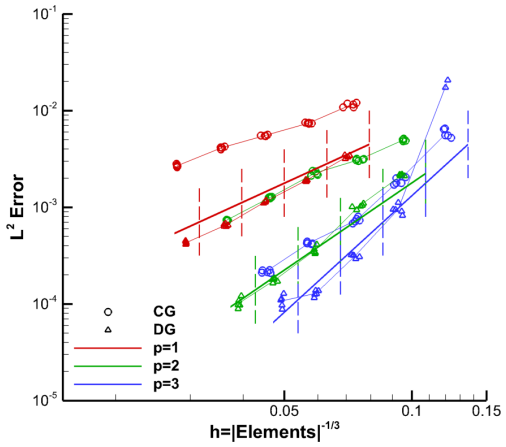
(a) Firedrake. (b) `refine` L^2. (c) `refine` k-exact.

Fig. 8 Pragmatic adapted meshes using multiscale metrics with target 128,000 elements.

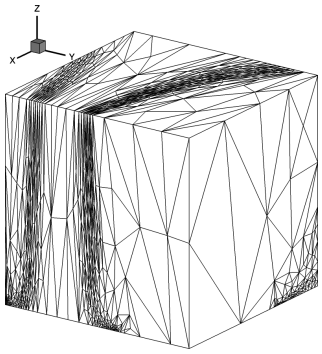


(a) Convergence: reference line slopes of $p+1$. (b) $p=1$ CG target 128,000 elements. (c) $p=1$ DG target 128,000 elements.

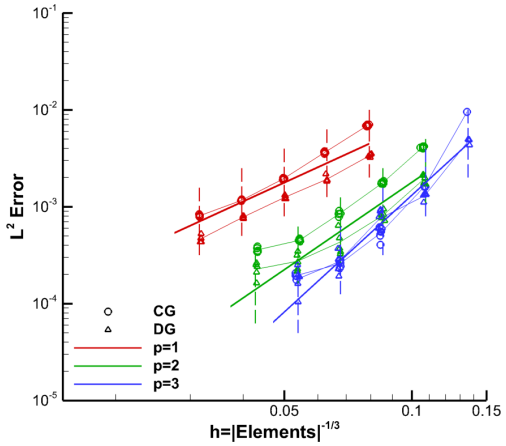
Fig. 9 `tanh3` L^2-error MOESS convergence. Example meshes adapted with FEFLO.A.



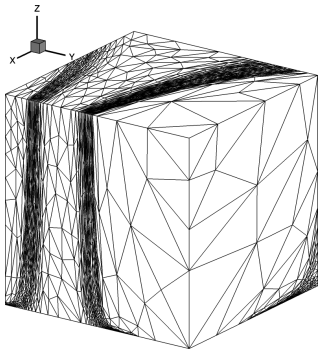
(a) `refine`.



(b) `refine` $p = 1$ CG target 128,000 elements.



(c) EPIC-ICSM.



(d) EPIC-ICSM $p = 1$ CG target 128,000 elements.

Fig. 10 `tanh3` L^2-error MOESS convergence rates using `refine` vs. EPIC-ICSM.

C. `sinatan3`

The `sinatan3` function,

$$s = 0.1\sin(50xz) + \tan^{-1}(0.1/(\sin(5y) - 2xz)), \tag{9}$$

shown in Fig. 11 has a strong curved anisotropic region with a smooth, low amplitude background variation. The L^2-error convergence using the multiscale metric is shown in Fig. 12. The scale of the vertical axis differs from the previous figure as the L^2-error is larger for the `sinatan3` function. The strong anisotropy in the `sinatan3` function again requires larger grid sizes to observe the asymptotic 2^{nd}-order rate. For the grids shown in Fig. 12, the k-exact multiscale metric does not appear to detect the smooth background variation. However, previous calculations with finer

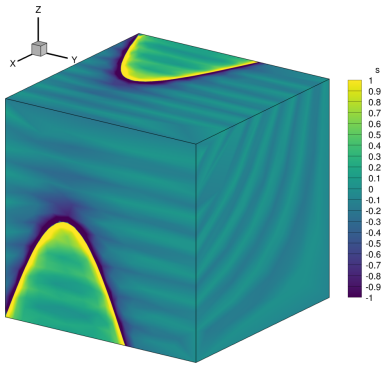
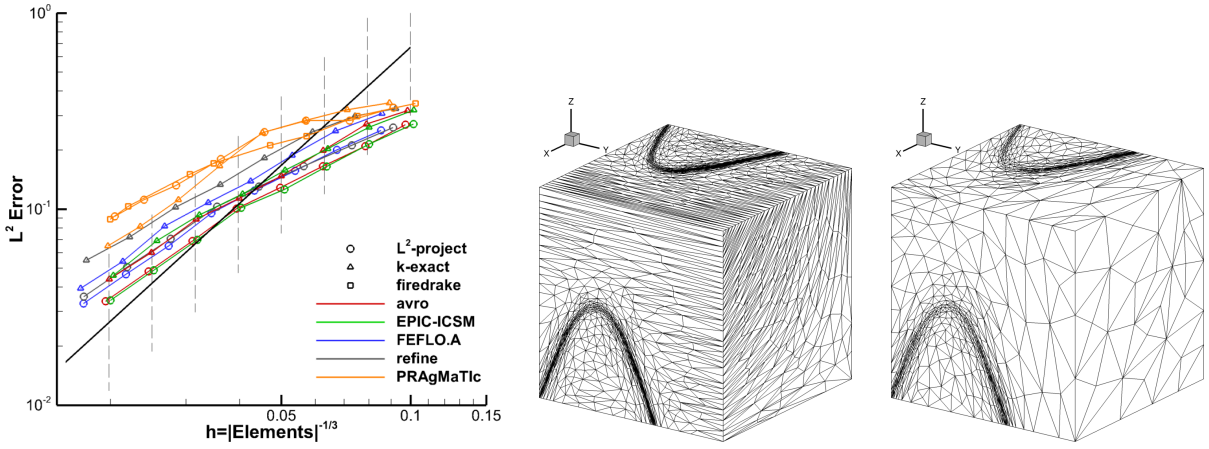


Fig. 11 `sinatan3` scalar function on a cube.

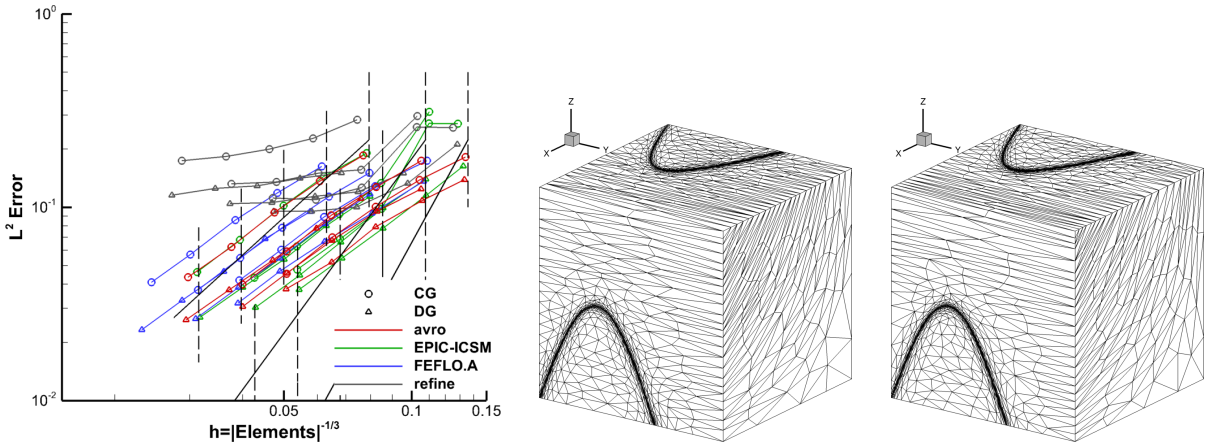
grids show that the k-exact metric does eventually capture the background variation. The L^2-projection multiscale metric does detect the background variation, which is likely why the L^2-error values for the L^2-projection metric is slightly smaller relative to the k-exact L^2-error.

As shown in Fig. 13, this function is likely not regular enough for the higher-order solution spaces to converge at the expected $p+1$ rate. As a result, the L^2-error converges at an approximately 2^{nd}-order rate independent of p. Like the `tanh3` function, the results from `refine` are outliers. As shown in Fig. 14, `refine` is not creating meshes with the expected anisotropy with the MOESS metric. This again appears to be an issue with combining MOESS with `refine` as the grids adapted with EPIC-ICSM using MOESS metrics exhibit the expected anisotropy as shown in Fig. 14, and the grids adapted with `refine` using multiscale metrics have the expected anisotropy as shown in Fig. 12. The convergence rates for each mesh adaptation tool using both multiscale and MOESS metrics are shown in the appendix of Ref. [17].



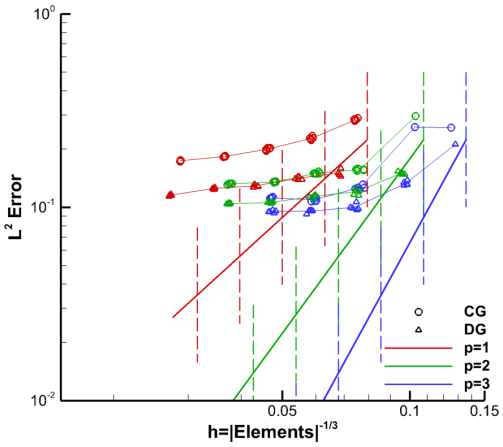
(a) Convergence: reference line slope of 2. (b) L^2 target 128,000 elements. (c) k-exact target 128,000 elements.

Fig. 12 `sinatan3` L^2-error multiscale metric convergence. Example meshes adapted with `refine`.

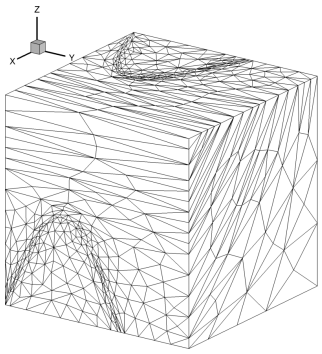


(a) Convergence: reference line slopes of $p + 1$. (b) $p = 1$ CG target 128,000 elements. (c) $p = 1$ DG target 128,000 elements.

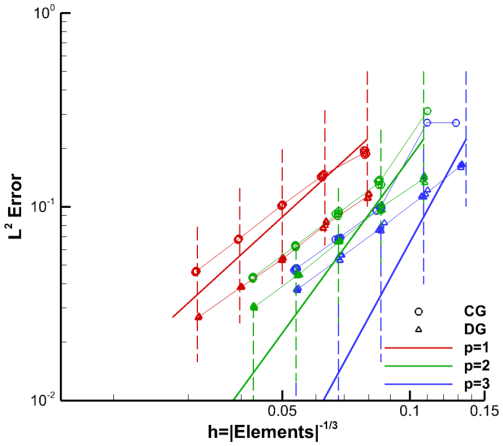
Fig. 13 `sinatan3` L^2-error MOESS convergence. Example meshes adapted with FEFLO.A.



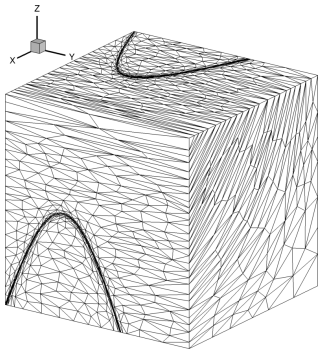
(a) `refine`.



(b) `refine` $p = 1$ CG target 128,000 elements.



(c) EPIC-ICSM.



(d) EPIC-ICSM $p = 1$ CG target 128,000 elements.

Fig. 14 `sinatan3` L^2-error MOESS convergence rates using `refine` vs. EPIC-ICSM.

VIII. Verification of Adaptation to Triple Boundary Layer (`TripleBL`) Scalar PDE

The previous three scalar cases allow isolated tests of the metric construction and mesh adaptation mechanics components on an analytic function. In this section, a scalar PDE is used to model the discretization error of a fluid flow solver and show how this form of error may impact error convergence rates. The triple boundary layer (`TripleBL`) solution, u, represents a simplified version of a boundary layer, which often occurs in fluid flow simulations. The anisotropic features of u are ideal tests for anisotropic metric-based adaptation, where a correctly implemented metric-based grid adaptation scheme should to be able to accurately resolve the features.

The (`TripleBL`) solution (shown in Fig. 15) is given by the expression,

$$u(x,y,z) = 1 - \frac{1-\exp\left(\frac{-a(x-1)}{\nu}\right)}{1-\exp\left(\frac{-a}{\nu}\right)} \frac{1-\exp\left(\frac{-b(y-1)}{\nu}\right)}{1-\exp\left(\frac{-b}{\nu}\right)} \frac{1-\exp\left(\frac{-c(z-1)}{\nu}\right)}{1-\exp\left(\frac{-c}{\nu}\right)}, \tag{10}$$

where $\nu = \frac{1}{30}$ and $a = b = c = 1$. Unlike the previous scalar functions, the `TripleBL` is an analytic solution to the linear $3D$ advection-diffusion PDE,

$$\nabla \cdot \left(\vec{V}u - \nu \nabla u\right) = 0, \qquad \vec{V} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}, \tag{11}$$

with Dirichlet boundary conditions on all boundaries of the unit cube $\Omega \equiv [0,1] \times [0,1] \times [0,1]$. This corresponds to $u(1,y,z) = u(x,1,z) = u(x,y,1) = 1$ on three faces of the unit cube with $u \approx 0$ at the origin. A rapid variation in u occurs close to the boundary $x = y = z = 1$ boundaries and the solution is smoothly varying elsewhere.

The discrete `TripleBL` scalar field, $u_{h,p}$, is computed by solving the advection-diffusion PDE using SANS with either an unstabilized CG discretization or a DG discretization with the BR2 viscous stabilization. The boundary conditions are enforced weakly, which means they are very poorly resolved on the initial coarse grid (uniform $5 \times 5 \times 5$ hexes divided into 750 tetrahedra).

The MOESS algorithm minimizes the DWR output functional error estimate. Two output functionals are considered here for the DWR. The first functional, $\mathcal{J}_u$, is a volume integral of the solution,

$$\mathcal{J}_u(w) = \int_\Omega w, \tag{12}$$

which for the analytic solution is

$$\mathcal{J}_u(u) = 1 - \frac{1}{abc} \frac{\left(e^{a/\nu}(a-\nu)+\nu\right)}{\left(e^{a/\nu}-1\right)} \frac{\left(e^{b/\nu}(b-\nu)+\nu\right)}{\left(e^{b/\nu}-1\right)} \frac{\left(e^{c/\nu}(c-\nu)+\nu\right)}{\left(e^{c/\nu}-1\right)}. \tag{13}$$

The second output functional, $\mathcal{J}_b$, is an integral of the flux on the $x = 1$ boundary of the domain, i.e.,

$$\mathcal{J}_b(w) = \iint_{\partial\Omega} \left(\vec{V}w(1,y,z) - \nu \nabla w(1,y,z)\right) \cdot \vec{n}\, dydz, \tag{14}$$

which for the analytic solution is

$$\mathcal{J}_b(u) = a\left(1 - \frac{e^{a/\nu}\left(e^{b/\nu}(b-\nu)+\nu\right)\left(e^{c/\nu}(c-\nu)+\nu\right)}{bc\left(e^{a/\nu}-1\right)\left(e^{b/\nu}-1\right)\left(e^{c/\nu}-1\right)}\right). \tag{15}$$

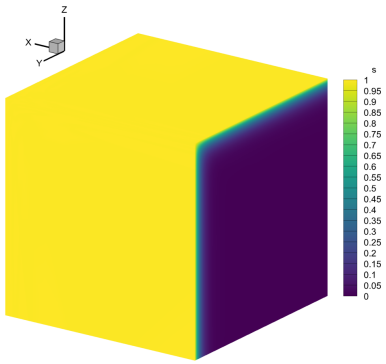


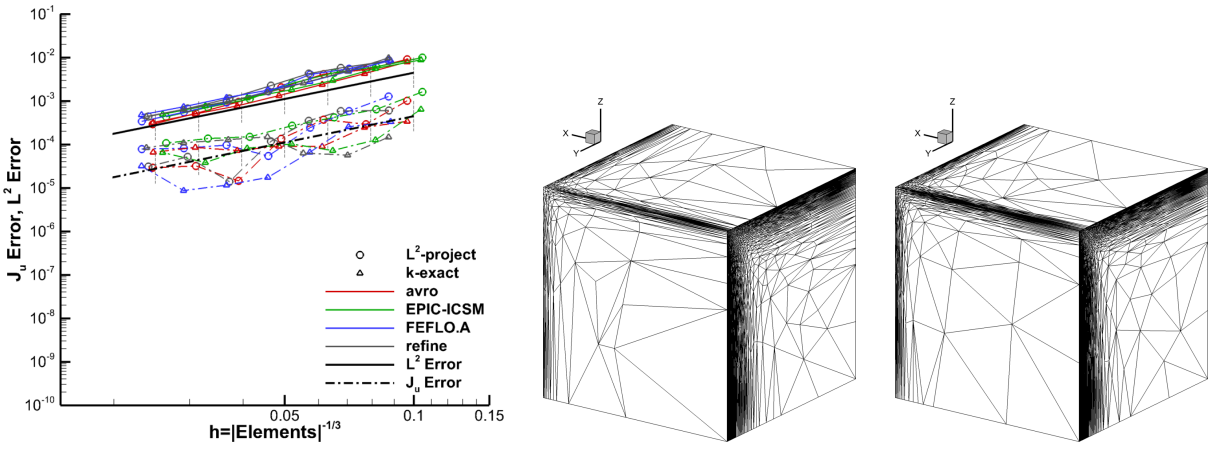
Fig. 15 `TripleBL` scalar solution.

For adjoint consistent discretizations, a convergence rate of $2p$ is expected for output functionals [68].

To verify convergence rates, multiscale metrics are computed from the discrete $p = 1$ unstabilized CG solutions at a series of target complexities. Both L^2-error as well as errors in $\mathcal{J}_u$, i.e., $|\mathcal{J}_u(u) - \mathcal{J}_u(u_{h,p})|$, are expected to converge at a 2^{nd}-order rate. The convergence rates of the errors computed from grids adapted with `avro`, EPIC-ISCM, FEFLO.A, and `refine` using the `refine` L^2-projection and k-exact multiscale metrics are shown in Fig. 16. The L^2-error converges for all mesh adaptation tools at a 2^{nd}-order rate. The convergence rate for the $\mathcal{J}_u$-error is a bit more noisy, but the trend is generally 2^{nd}-order. The meshes adapted with `refine` shown in Fig. 16 are representative of the meshes created with the other mesh adaptation tools. Convergence rates for the individual mesh adaptation tools as well as figures of the adapted meshes are shown in the appendix of Ref. [17].

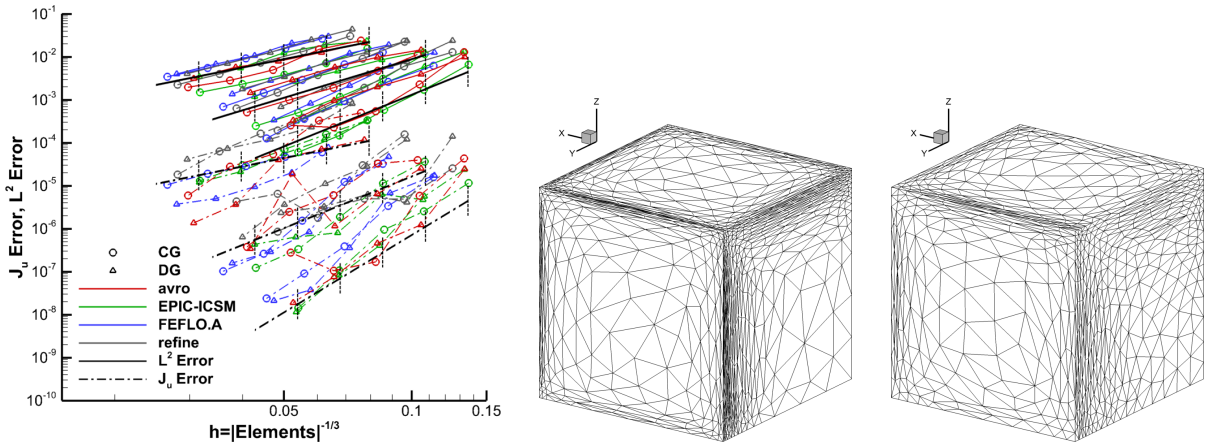
Convergence rates of L^2- and $\mathcal{J}_u$-error computed using $\mathcal{J}_u$ as the functional for the DWR error estimate in the MOESS algorithm for each mesh adaptation tool are shown in Fig. 17. The reference lines for the L^2-error have slopes of $p + 1$, and the reference lines for $\mathcal{J}_u$-error have slopes of $2p$. Unfortunately, the errors are too noisy to decipher the trends of all mesh adaptation tools at once. Focusing on the errors computed with the EPIC-ICSM mesh adaptation tool in Fig. 18, the slopes in the $\mathcal{J}_u$-error roughly follow the expected rates of $2p$. The L^2-error also converges at the expected rate of $p + 1$, because $\mathcal{J}_u$ is a volume functional and MOESS is minimizing the error in this functional. Error convergence figures for each mesh adaptation tools are shown in the appendix of Ref. [17].

Errors in L^2 and $\mathcal{J}_b$ using $\mathcal{J}_b$ as the functional for the DWR error estimate in the MOESS algorithm for each mesh adaptation tool are shown in Fig. 19. Again focusing on results with the EPIC-ICSM in Fig. 20, the $\mathcal{J}_b$-error converges approximately at the expected rate of $2p$. However, unlike the volume functional $\mathcal{J}_u$, minimizing the error in the boundary functional $\mathcal{J}_b$ results in grids clustered near the $x = 1$ domain boundary as shown in Figs. 19 and 20. As a result, the L^2-error that includes the entire volume is not expected to converge.



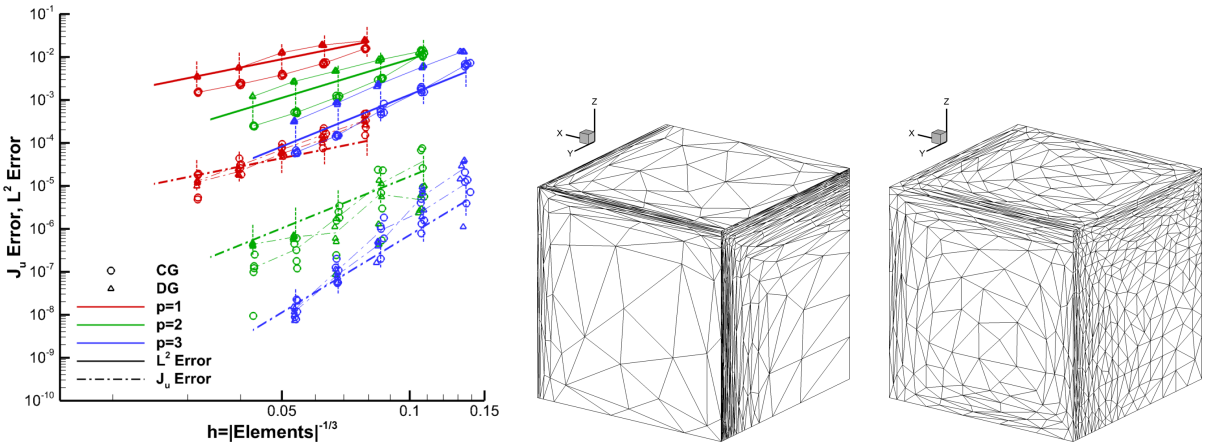
(a) Convergence: reference line slope of 2. (b) L^2 target 128,000 elements. (c) k-exact target 128,000 elements.

Fig. 16 L^2-error and $\mathcal{J}_u$-error multiscale metric convergence. Example meshes adapted with `refine`.



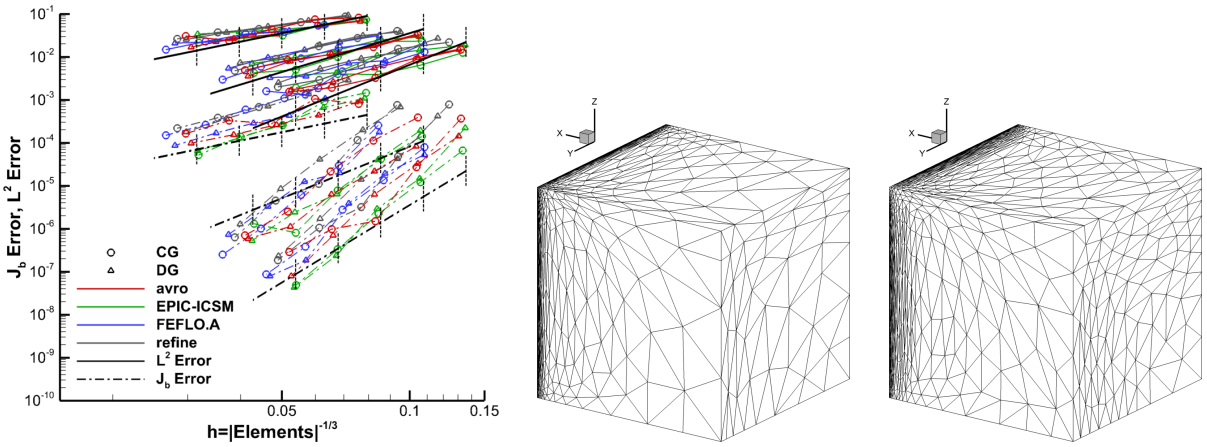
(a) Convergence: reference line slopes of $p+1$ and $2p$. (b) $p=1$ CG target 128,000 elements. (c) $p=1$ DG target 128,000 elements.

Fig. 17 `TripleBL` L^2-error and $\mathcal{J}_u$-error MOESS convergence. Example meshes adapted with FEFLO.A.



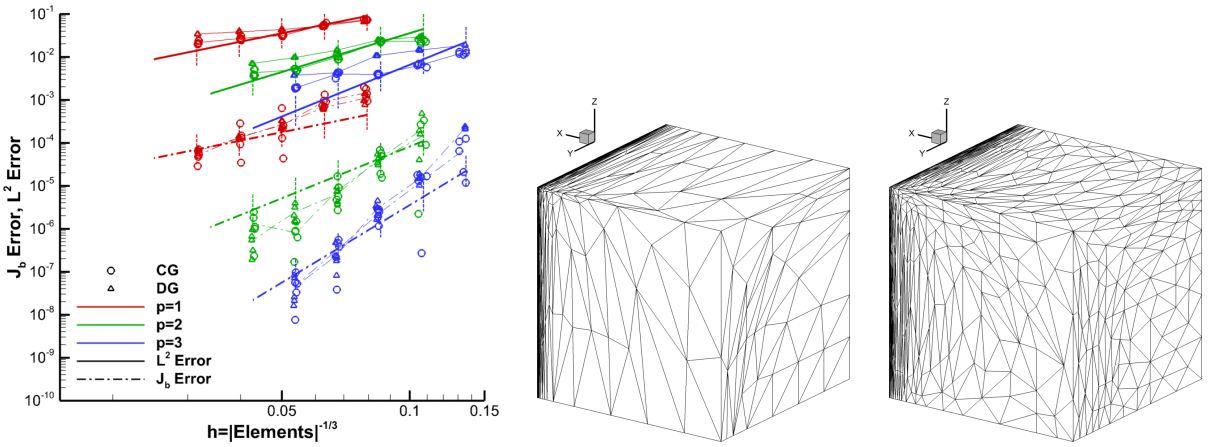
(a) Convergence: reference line slopes of $p+1$ and $2p$. (b) $p=1$ CG target 128,000 elements. (c) $p=1$ DG target 128,000 elements.

Fig. 18 `TripleBL` L^2-error and $\mathcal{J}_u$-error MOESS convergence and grids using EPIC-ICSM.



(a) Convergence: reference line slopes of $p+1$ and $2p$. (b) $p=1$ CG target 128,000 elements. (c) $p=1$ DG target 128,000 elements.

Fig. 19 `TripleBL` L^2-error and $\mathcal{J}_b$-error MOESS convergence. Example meshes adapted with FEFLO.A.



(a) Convergence: reference line slopes of $p+1$ and $2p$. (b) $p=1$ CG target 128,000 elements. (c) $p=1$ DG target 128,000 elements.

Fig. 20 `TripleBL` L^2-error and $\mathcal{J}_b$-error MOESS convergence and grids using EPIC-ICSM.

IX. Laminar Delta Wing

The $3D$ Laminar Delta Wing case was described by Wang et al. [10] and was used in the first three International Workshops on High-Order CFD Methods (HIOCFD). This case was denoted BTC3 in the ADIGMA (Adaptive Higher-order Variational Methods for Aerodynamic Applications in Industry) project [69, 70] where adaptive results are detailed in Hartmann et al. [71]. Adaptive results were presented by Leicht and Hartmann [72]. Yano [61] published grid adapted results with the MOESS metric. The geometry was modified from the original description in Klaij, van der Vegt, and van der Ven [73] based on the experiment of Riley and Lowson [74] to match the HIOCFD and ADIGMA cases.

The dimensions of the Delta Wing are provided in Table 2. The domain is constructed from planar facets, which eliminates the need for curved boundary recovery from the adaptive grid mechanics and curved elements for higher-order solution representations. The freestream conditions are 0.3 Mach, 4,000 Reynolds number based on the root chord length, and 12.5° angle of attack. Isothermal noslip boundary conditions are set to freestream temperature. The Prandtl number is 0.72. The viscosity is assumed to be constant to match the workshop case. An example adapted solution is shown in Fig. 21 computed by FUN3D-SFE. The symmetry plane is colored with Mach number, the delta wing is colored with coefficient of pressure. The footprint of the leading edge vortex can be seen in the upper surface pressure coefficient. A transparent Mach=0.2 isosurface shows the vortex roll up downstream of the configuration.

Table 2 Reference geometry for the Delta Wing.

Root Chord	1.0
Span	0.267949223
Thickness	0.0244161374
Reference Area	0.133974596

The drag and lift coefficient adaptation trajectories are shown in Fig. 22 with a vertical range that shows all adapted-grid trajectories. The spacing h is estimated as $h = DOF^{-1/3}$. The descriptions in the legend are the flow solver, grid mechanics, and metric construction method joined with the + symbol. The SANS adaptive trajectories show an average force coefficient value for a number of fixed-complexity adaptations, but the other methods show the last grid at a given complexity. The SANS-GLS cases use the `refine` multiscale metric with L^2-projection Hessian reconstruction. The trajectories that extend to $h < 0.01$ appear to be converging toward the same fine grid values. The trends of some of the trajectories that end with $h > 0.01$ are less consistent. The reference values of the drag and lift coefficients computed for the ADIGMA project are 0.16608 and 0.34865 in Ref. [71] and 0.1658 and 0.347 in Ref. [72]. The trajectories here indicate a slightly lower drag and and a higher lift than the two sets of reference values. Finer grids are used for the adaptive results here than the previously published results from HIOCFD or ADIGMA.

A zoom of the finer adapted grid solutions is shown in Fig. 23. The SANS-GLS-P1+EPIC-ICSM+`refine`-L2 and

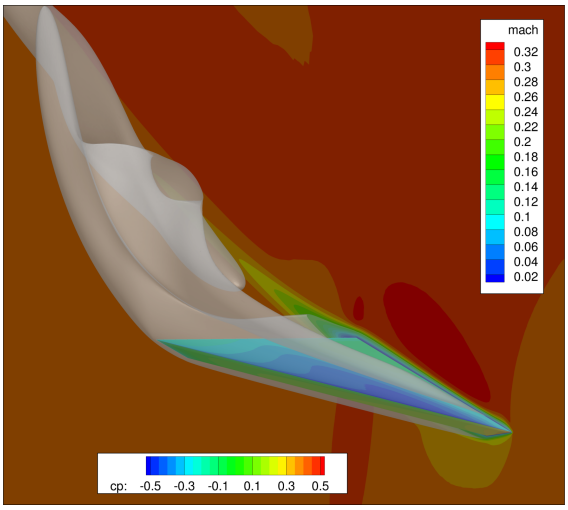
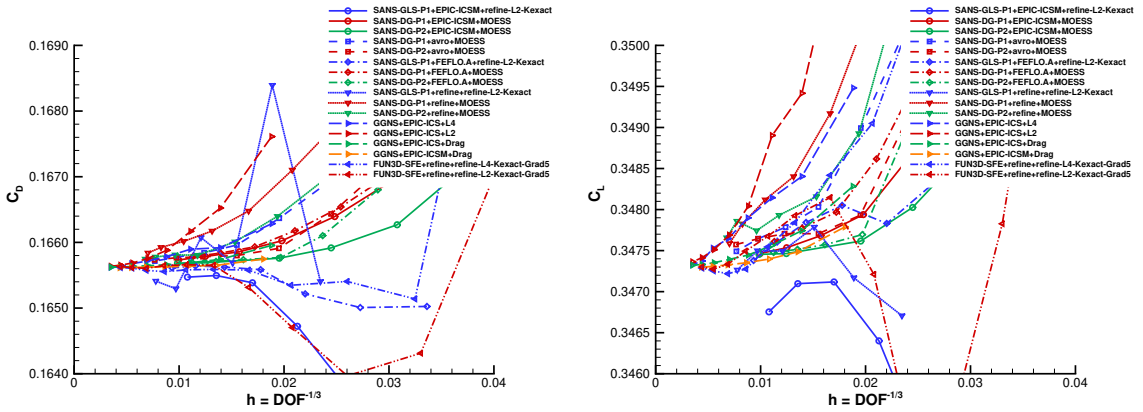


Fig. 21 Laminar Delta Wing, Symmetry plane colored with Mach, Surface colored with pressure coefficient, transparent Mach=0.2 isosurface.



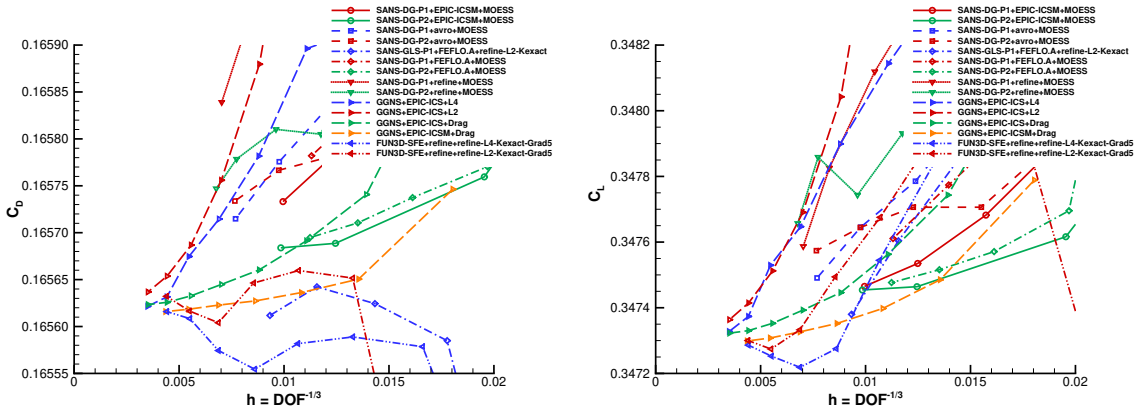
(a) Drag coefficient.

(b) Lift coefficient.

Fig. 22 Laminar Delta Wing, lift and drag coefficient, wide scale.

SANS-GLS-P1+`refine`+`refine`-L2 trajectories have been omitted, because they do not terminate in the narrow drag and lift coefficient axis range. The addition of vertex movement to GGNS+EPIC drag adaptation allows it to reach fine-grid force values earlier. There are no consistent trends when varying the norm exponent of the multiscale metric shown by GGNS+EPIC and FUN3D-SFE+`refine`. At $h = 0.0048$ (about 5,000,000 complexity or about 11,000,000 vertices), the drag coefficient is in the range 0.16561–0.16566 (half a drag count variation) and the lift coefficient is in the range 0.34722–0.34745 (0.07% variation).

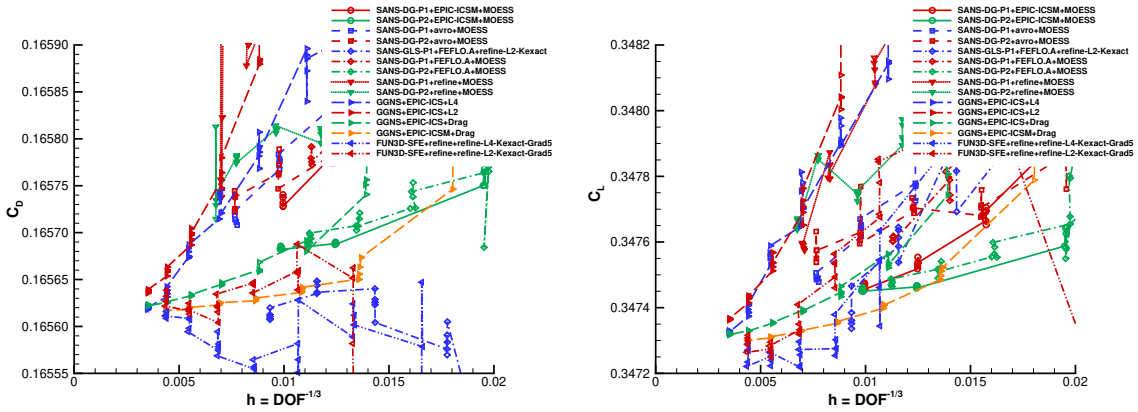
All the subiterations for the finer adapted grid solutions are shown in Fig. 24. The SANS-GLS-P1+EPIC-ICSM+`refine`-L2 and SANS-GLS-P1+`refine`+`refine`-L2 trajectories have been omitted, because they do not terminate in the narrow drag and lift coefficient axis range. The multiscale metric trajectories have more scatter at a



(a) Drag coefficient. (b) Lift coefficient.

Fig. 23 Laminar Delta Wing, lift and drag coefficient, narrow scale.

fixed complexity and a steeper slope than the lift and drag adaptation trajectories. The addition of vertex movement to GGNS+EPIC drag adaptation reduces scatter on coarser grids. The larger scatter shown by FUN3D-SFE+`refine` may be due to inadequate metric conformity at these low error levels. SANS-GLS with `refine` grid mechanics and `refine` multiscale metric has the largest scatter at a fixed-complexity iteration.



(a) Drag coefficient. (b) Lift coefficient.

Fig. 24 Laminar Delta Wing, lift and drag coefficient, narrow scale, with subiterations.

The convergence of Mach interpolation error is shown in Fig. 25 for the FUN3D-SFE Laminar Delta Wing cases. The "truth" solution used to measure interpolation error is the final mesh in the adaptive series. To compute the error, the Mach field on a candidate grid is linearly interpolated to the truth grid. The 2-norm and 4-norm (as shown in parenthesis of the figure legend) of the difference is computed for the series of multiscale metric adapted grids with 2 and 4 exponents. All trajectories show the expected 2nd-order convergence.

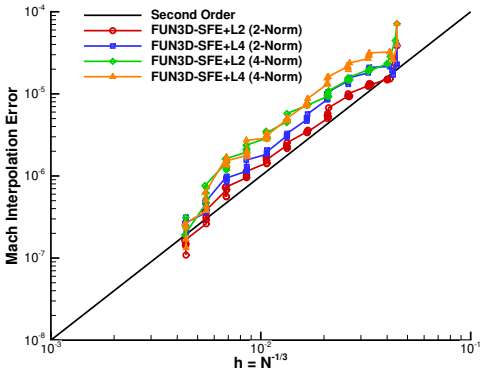


Fig. 25 Laminar Delta Wing, Mach interpolation error, with subiterations.

X. Conclusions

The verification of individual tools and integrated grid adaptation processes is critical to establishing the confidence in these tools, which encourages their use in production fluid simulation and certification by analysis workflows. Where possible, these tools are evaluated on analytic solutions to demonstrate design-order convergence. Where analytic solutions are not available, the Code Comparison Principle is exercised on a Laminar Delta Wing. These wing cases have been examined in past studies, and improvements to these tools and integrated grid adaptation processes are shown in the current study.

The interpolation error of grids adapted to three scalar functions is examined. The first field, `sinfun3`, shows design convergence order for the three implementations of the multiscale metric and the MOESS algorithm on this smooth function. The multiscale metric is tested with five grid mechanics tools and MOESS is tested with four grid mechanics tools.

The `tanh3` function has mild shock and boundary layer features. The Firedrake multiscale metric exhibited a degradation at the boundary that allows design-order convergence of error, but at a higher error level than other implementations. MOESS with `refine` adapting the continuous approximation space has errors that are significantly higher than other mesh modification tools and fails to resolve the `tanh3` boundary layer feature. These situations demonstrate the value of the Code Comparison Principle in identifying differences between implementations. Appendices of Ref. [17] contain the convergence plots of all tools and example surface meshes to illustrate potential causes of design-order behavior degradation for these scalar fields.

The `sinatan3` function has a strong, curved anisotropic region combined with a smooth, low amplitude background variation. At these complexity levels, the `refine` multiscale metric with k-exact Hessian reconstruction misses the low amplitude variation that the L^2-projection Hessian reconstruction resolves. The unresolved low amplitude variation due to an implementation difference results in a consistently higher L^2-error for all five grid mechanics tools with

k-exact Hessian reconstruction. The function is not regular enough for MOESS to converge at the expected $p + 1$ rate; a p-independent 2^{nd}-order rate is observed for `avro`, FEFLO.A, and EPIC-ICSM. The combination of `refine` with MOESS fails to resolve the curved anisotropic feature, which results in a nonconverging L^2-error.

A linear $3D$ advection-diffusion PDE with an analytical solution is used for testing the multiscale and MOESS metrics. Four grid mechanics tools give 2^{nd}-order convergence and very similar grids with the multiscale metric. MOESS with four grid mechanics tools converges at design-order for the volume and boundary output functionals targeted by MOESS.

FUN3D-SFE and the `refine` implementation of the multiscale metric is shown to produce Mach fields with interpolation errors that converge at 2^{nd}-order for the Laminar Delta Wing. Output-based metrics that control error estimates of lift and drag converged faster to fine-grid forces and moments than the multiscale metric for the Laminar Delta Wing. This motivates the need for continued development of goal-oriented and output-based grid adaptation metrics.

This study documents a clear improvement to existing grid adaptation mechanics and the introduction of new implementations. This effort demonstrates progress toward CFD Vision 2030 [13], and a number of the timeline elements proposed by Park et al. [12] continue to be accomplished. This work provides a benchmark for verifying the multiscale metric in integrated adaptive grid tools for scalar fields and wing aerodynamics. These verified processes will set the stage for the infusion of solution interpolation error and ultimately output-error-controlled simulations into production CFD workflows and certification by analysis.

Acknowledgements

Yi Liu provided the FUN3D-FV uniformly refined grid forces. Matt O'Connell improved the performance of the Parfait wall distance to permit larger adapted grid sizes. Bil Kleb, Matt O'Connell, and Beth Lee-Rausch provided feedback that improved this manuscript. This work was partially supported by the Transformational Tools and Technologies (TTT) Project of the NASA Transformative Aeronautics Concepts Program (TACP).

References

[1] Mavriplis, D. J., Vassberg, J. C., Tinoco, E. N., Mani, M., Brodersen, O. P., Eisfeld, B., Wahls, R. A., Morrison, J. H., Zickuhr, T., Levy, D., and Murayama, M., "Grid Quality and Resolution Issues from the Drag Prediction Workshop Series," *AIAA Journal of Aircraft*, Vol. 46, No. 3, 2009, pp. 935–950. doi:10.2514/1.39201.

[2] Levy, D. W., Laflin, K. R., Tinoco, E. N., Vassberg, J. C., Mani, M., Rider, B., Rumsey, C. L., Wahls, R. A., Morrison, J. H., Brodersen, O. P., Crippa, S., Mavriplis, D. J., and Murayama, M., "Summary of Data from the Fifth Computational Fluid Dynamics Drag Prediction Workshop," *AIAA Journal of Aircraft*, Vol. 51, No. 4, 2014, pp. 1194–1213. doi:10.2514/1.C032389.

[3] Morrison, J. H., "Statistical Analysis of the Fifth Drag Prediction Workshop Computational Fluid Dynamics Solutions," *AIAA Journal of Aircraft*, Vol. 51, No. 4, 2014, pp. 1214–1222. doi:10.2514/1.C032736.

[4] Rumsey, C. L., "Recent Developments on the Turbulence Modeling Resource Website," AIAA Paper 2015–2927, 2015.

[5] Morrison, J. H., Kleb, B., and Vassberg, J. C., "Observations on CFD Verification and Validation from the AIAA Drag Prediction Workshops," AIAA Paper 2014–202, 2014.

[6] Diskin, B., and Thomas, J. L., "Introduction: Evaluation of RANS Solvers on Benchmark Aerodynamic Flows," *AIAA Journal*, Vol. 54, No. 9, 2016, pp. 2561–2562. doi:10.2514/1.J054642.

[7] Diskin, B., Thomas, J. L., Rumsey, C. L., and Schwöppe, A., "Grid-Convergence of Reynolds-Averaged Navier–Stokes Solutions for Benchmark Flows in Two Dimensions," *AIAA Journal*, Vol. 54, No. 9, 2016, pp. 2563–2588. doi:10.2514/1.J054555.

[8] Diskin, B., Thomas, J. L., Pandya, M. J., and Rumsey, C. L., "Reference Solutions for Benchmark Turbulent Flows in Three Dimensions," AIAA Paper 2016–858, 2016.

[9] Diskin, B., Anderson, W. K., Pandya, M. J., Rumsey, C. L., Thomas, J. L., Liu, Y., and Nishikawa, H., "Grid Convergence for Three Dimensional Benchmark Turbulent Flows," AIAA Paper 2018–1102, 2018.

[10] Wang, Z. J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H. T., Kroll, N., May, G., Persson, P.-O., van Leer, B., and Visbal, M., "High-Order CFD Methods: Current Status and Perspective," *International Journal for Numerical Methods in Fluids*, Vol. 72, No. 8, 2013, pp. 811–845. doi:10.1002/fld.3767.

[11] Alauzet, F., and Loseille, A., "A Decade of Progress on Anisotropic Mesh Adaptation for Computational Fluid Dynamics," *Computer-Aided Design*, Vol. 72, 2016, pp. 13–39. doi:10.1016/j.cad.2015.09.005, 23rd International Meshing Roundtable Special Issue: Advances in Mesh Generation.

[12] Park, M. A., Krakos, J. A., Michal, T., Loseille, A., and Alonso, J. J., "Unstructured Grid Adaptation: Status, Potential Impacts, and Recommended Investments Toward CFD Vision 2030," AIAA Paper 2016–3323, 2016.

[13] Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D., "CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences," NASA CR-2014-218178, Langley Research Center, Mar. 2014. doi:2060/20140003093.

[14] Park, M. A., Loseille, A., Krakos, J. A., and Michal, T., "Comparing Anisotropic Output-Based Grid Adaptation Methods by Decomposition," AIAA Paper 2015–2292, 2015.

[15] Ibanez, D., Barral, N., Krakos, J., Loseille, A., Michal, T., and Park, M., "First Benchmark of the Unstructured Grid Adaptation Working Group," *Procedia Engineering*, Vol. 203, 2017, pp. 154–166. doi:10.1016/j.proeng.2017.09.800, 26th International Meshing Roundtable, IMR26, 18-21 Sept. 2017, Barcelona, Spain.

[16] Park, M. A., Barral, N., Ibanez, D., Kamenetskiy, D. S., Krakos, J., Michal, T., and Loseille, A., "Unstructured Grid Adaptation and Solver Technology for Turbulent Flows," AIAA Paper 2018–1103, 2018.

[17] Park, M. A., Balan, A., Anderson, W. K., Galbraith, M. C., Caplan, P. C., Carson, H. A., Michal, T., Krakos, J. A., Kamenetskiy, D. S., Loseille, A., Alauzet, F., Frazza, L., and Barral, N., "Verification of Unstructured Grid Adaptation Components," AIAA Paper 2019–1723, 2019.

[18] Michal, T., Kamenetskiy, D. S., Marcum, D., Alauzet, F., Frazza, L., and Loseille, A., "Comparing Anisotropic Error Estimates for ONERA M6 Wing RANS Simulations," AIAA Paper 2018–920, 2018.

[19] Oberkampf, W. L., and Roy, C. J., *Verification and Validation in Scientific Computing*, Cambridge University Press, 2010.

[20] Lee, H. B., Ghia, U., Bayyuk, S., Oberkampf, W., Roy, C. J., Benek, J., Rumsey, C. L., Powers, J., Bush, R. H., and Mani, M., "Development and Use of Engineering Standards for Computational Fluid Dynamics for Complex Aerospace Systems," AIAA Paper 2016–3811, 2016.

[21] Oberkamp, W. L., and Trucano, T. G., "Verification and Validation in Computational Fluid Dynamics," *Progress in Aerospace Sciences*, Vol. 38, No. 3, 2002, pp. 209–272. doi:10.1016/S0376-0421(02)00005-2.

[22] Trucano, T. G., Pilch, M., and Oberkampf, W. L., "On the Role of Code Comparisons in Verification and Validation," Tech. Rep. SAND2003-2752, Sandia National Laboratories, Aug. 2003.

[23] Kleb, W. L., and Wood, W. A., "CFD: A Castle in the Sand?" AIAA Paper 2004–2627, 2004.

[24] Kleb, W. L., and Wood, W. A., "Computational Simulations and the Scientific Method," AIAA Paper 2005–4873, 2005.

[25] Oberkamp, W. L., and Trucano, T. G., "Verification and Validation Benchmarks," *Nuclear Engineering and Design*, Vol. 238, No. 3, 2008, pp. 716–743. doi:10.1016/j.nucengdes.2007.02.032.

[26] Loseille, A., and Alauzet, F., "Continuous Mesh Framework Part I: Well-Posed Continuous Interpolation Error," *SIAM Journal on Numerical Analysis*, Vol. 49, No. 1, 2011, pp. 38–60. doi:10.1137/090754078.

[27] Alauzet, F., and Loseille, A., "High-Order Sonic Boom Modeling Based on Adaptive Methods," *Journal of Computational Physics*, Vol. 229, No. 3, 2010, pp. 561–593. doi:10.1016/j.jcp.2009.09.020.

[28] Fidkowski, K. J., and Darmofal, D. L., "Review of Output-Based Error Estimation and Mesh Adaptation in Computational Fluid Dynamics," *AIAA Journal*, Vol. 49, No. 4, 2011, pp. 673–694. doi:10.2514/1.J050073.

[29] Loseille, A., and Alauzet, F., "Continuous Mesh Framework Part II: Validations and Applications," *SIAM Journal on Numerical Analysis*, Vol. 49, No. 1, 2011, pp. 61–86. doi:10.1137/10078654X.

[30] Galbraith, M. C., Allmaras, S. R., and Darmofal, D. L., "SANS RANS Solutions for 3D Benchmark Configurations," AIAA Paper 2018–1570, 2018.

[31] Galbraith, M. C., Allmaras, S. R., and Darmofal, D. L., "A Verification Driven Process for Rapid Development of CFD Software," AIAA Paper 2015–818, 2015.

[32] Abhyankar, S., Brown, J., Constantinescu, E. M., Ghosh, D., Smith, B. F., and Zhang, H., "PETSc/TS: A Modern Scalable ODE/DAE Solver Library," *Computing Research Repository (CoRR)*, Vol. Numerical Analysis (math.NA), No. arXiv:1806.01437, 2018. URL `http://arxiv.org/abs/1302.6066`.

[33] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., McInnes, L. C., Rupp, K., Smith, B. F., Zampini, S., Zhang, H., and Zhang, H., "PETSc Users Manual," Tech. Rep. ANL-95/11 - Revision 3.8, Argonne National Laboratory, 2017.

[34] Balay, S., Gropp, W. D., McInnes, L. C., and Smith, B. F., "Efficient Management of Parallelism in Object Oriented Numerical Software Libraries," *Modern Software Tools in Scientific Computing*, edited by E. Arge, A. M. Bruaset, and H. P. Langtangen, Birkhäuser Press, 1997, pp. 163–202.

[35] Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimum Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal of Scientific and Statistical Computing*, Vol. 7, 1986, pp. 856–869.

[36] Anderson, W. K., Newman, J. C., and Karman, S. L., "Stabilized Finite Elements in FUN3D," *Journal of Aircraft*, Vol. 55, No. 2, 2018, pp. 696–714. doi:10.2514/1.C034482.

[37] Anderson, W. K., and Bonhaus, D. L., "An Implicit Upwind Algorithm for Computing Turbulent Flows on Unstructured Grids," *Computers and Fluids*, Vol. 23, No. 1, 1994, pp. 1–22. doi:10.1016/0045-7930(94)90023-X.

[38] Biedron, R. T., Carlson, J.-R., Derlaga, J. M., Gnoffo, P. A., Hammond, D. P., Jones, W. T., Kleb, B., Lee-Rausch, E. M., Nielsen, E. J., Park, M. A., Rumsey, C. L., Thomas, J. L., and Wood, W. A., "FUN3D Manual: 13.4," NASA TM-2018-220096, Langley Research Center, Oct. 2018. doi:2060/20180007519.

[39] Brooks, A. N., and Hughes, T. J. R., "Streamline Upwind/Petrov-Galerkin Formulation for Convection Dominated Flows with Particular Emphasis on Incompressible Navier-Stokes Equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 32, No. 1–3, 1982, pp. 199–259. doi:10.1016/0045-7825(82)90071-8.

[40] Shakib, F., Hughes, T. J. R., and Johan, Z., "A New Finite-Element Formulation for Computational Fluid Dynamics: X. The Compressible Euler and Navier-Stokes Equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 89, No. 1–3, 1991, pp. 141–219. doi:10.1016/0045-7825(91)90041-4.

[41] Hughes, T. J. R., Franca, L. P., and Hulbert, G. M., "A New Finite-Element Formulation for Computational Fluid Dynamics: VIII. The Galerkin Least Squares Method for Advective-Diffusion Equations," *Computer Methods in Applied Mechanics and Engineering*, Vol. 73, No. 2, 1989, pp. 173–189. doi:10.1016/0045-7825(89)90111-4.

[42] Bazilevs, Y., and Akkerman, I., "Large Eddy Simulation of Turbulent Taylor–Couette Flow Using Isogeometric Analysis and the Residual-Based Variational Multiscale Method," *Journal of Computational Physics*, Vol. 229, No. 9, 2010, pp. 3402–3414. doi:10.1016/j.jcp.2010.01.008.

[43] Saad, Y., *Iterative Methods for Sparse Linear Systems*, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2003.

[44] Allmaras, S. R., "Lagrange Multiplier Implementation of Dirichlet Boundary Conditions in Compressible Navier-Stokes Finite Element Methods," AIAA Paper 2005-4714, 2005.

[45] Kamenetskiy, D. S., Bussoletti, J. E., Hilmes, C. L., Venkatakrishnan, V., and Wigton, L. B., "Numerical Evidence of Multiple Solutions for the Reynolds-Averaged Navier-Stokes Equations," *AIAA Journal*, Vol. 52, No. 8, 2014, pp. 1686–1698. doi:10.2514/1.J052676.

[46] Caplan, P. C., Haimes, R., and Darmofal, D. L., "Extension of Local Cavity Operators to $3d + t$ Spacetime Mesh Adaptation," AIAA Paper 2019–1992, 2019.

[47] Coupez, T., "Génération de Maillage et Adaptation de Maillage par Optimisation Locale," *Revue européenne des éléments finis*, Vol. 9, 2000, pp. 403–423. doi:10.1080/12506559.2000.10511454.

[48] Loseille, A., Alauzet, F., and Menier, V., "Unique Cavity-Based Operator and Hierarchical Domain Partitioning for Fast Parallel Generation of Anisotropic Meshes," *Computer-Aided Design*, Vol. 85, 2017, pp. 53–67. doi:10.1016/j.cad.2016.09.008, 24th International Meshing Roundtable Special Issue: Advances in Mesh Generation.

[49] Haimes, R., and Drela, M., "On The Construction of Aircraft Conceptual Geometry for High-Fidelity Analysis and Design," AIAA Paper 2012–683, 2012.

[50] Michal, T., and Krakos, J., "Anisotropic Mesh Adaptation Through Edge Primitive Operations," AIAA Paper 2012–159, 2012.

[51] Alauzet, F., "A Changing-Topology Moving Mesh Technique for Large Displacements," *Engineering with Computers*, Vol. 30, No. 2, 2014, pp. 175–200. doi:10.1007/s00366-013-0340-z.

[52] Freitag, L. A., and Ollivier-Gooch, C., "Tetrahedral Mesh Improvement Using Swapping and Smoothing," *International Journal for Numerical Methods in Engineering*, Vol. 40, 1997, pp. 3979–4002. doi:10.1002/(SICI)1097-0207(19971115)40:21<3979::AID-NME251>3.0.CO;2-9.

[53] Haimes, R., and Dannenhoffer, J. F., III, "EGADSlite: A Lightweight Geometry Kernel for HPC," AIAA Paper 2018–1401, 2018.

[54] Rathgeber, F., Ham, D. A., Mitchell, L., Lange, M., Luporini, F., Mcrae, A. T. T., Bercea, G.-T., Markall, G. R., and Kelly, P. H. J., "Firedrake: Automating the Finite Element Method by Composing Abstractions," *ACM Transactions on Mathematical Software (TOMS)*, Vol. 43, No. 3, 2016, pp. 24:1–24:27. doi:10.1145/2998441.

[55] Barral, N., Knepley, M. G., Lange, M., Piggott, M. D., and Gorman, G. J., "Anisotropic Mesh Adaptation in Firedrake with PETSc DMPlex," *25th International Meshing Roundtable Research Notes*, Sandia National Laboratories, 2016.

[56] Loseille, A., and Löhner, R., "Cavity-Based Operators for Mesh Adaptation," AIAA Paper 2013–152, 2013.

[57] Loseille, A., Dervieux, A., Frey, P. J., and Alauzet, F., "Achievement of Global Second Order Mesh Convergence for Discontinuous Flows with Adapted Unstructured Meshes," AIAA Paper 2007–4186, 2007.

[58] Barth, T. J., "Recent Developments in High Order K-Exact Reconstruction on Unstructured Meshes," AIAA Paper 93–668, 1993.

[59] Alauzet, F., "Size Gradation Control of Anisotropic Meshes," *Finite Elements in Analysis and Design*, Vol. 46, No. 1–2, 2010, pp. 181–202. doi:10.1016/j.finel.2009.06.028.

[60] Arsigny, V., Fillard, P., Pennec, X., and Ayache, N., "Log-Euclidean Metrics for Fast and Simple Calculus on Diffusion Tensors," *Magnetic Resonance in Medicine*, Vol. 56, No. 2, 2006, pp. 411–421. doi:10.1002/mrm.20965.

[61] Yano, M., "An Optimization Framework for Adaptive Higher-Order Discretizations of Partial Differential Equations on Anisotropic Simplex Meshes," Ph.D. thesis, Massachusetts Institute of Technology, Jun. 2012. doi:1721.1/76090.

[62] Yano, M., and Darmofal, D. L., "An Optimization-Based Framework for Anisotropic Simplex Mesh Adaptation," *Journal of Computational Physics*, Vol. 231, No. 22, 2012, pp. 7626–7649. doi:10.1016/j.jcp.2012.06.040.

[63] Becker, R., and Rannacher, R., "A Feed-Back Approach to Error Control in Finite Element Methods: Basic Analysis and Examples," *East-West Journal of Numerical Mathematics*, Vol. 4, 1996, pp. 237–264.

[64] Carson, H. A., Darmofal, D. L., Galbraith, M. C., and Allmaras, S. R., "Analysis of Output-Based Error Estimation for Finite Element Methods," *Applied Numerical Mathematics*, Vol. 118, 2017, pp. 182–202. doi:10.1016/j.apnum.2017.03.004.

[65] Richter, T., and Wick, T., "Variational Localizations of the Dual Weighted Residual Estimator," *Journal of Computational and Applied Mathematics*, Vol. 279, 2015, pp. 192–208. doi:10.1016/j.cam.2014.11.008.

[66] Zienkiewicz, O. C., and Zhu, J. Z., "The Superconvergent Patch Recovery and A Posteriori Error Estimates. Part 1: The Recovery Technique," *International Journal for Numerical Methods in Engineering*, Vol. 33, No. 7, 1992, pp. 1331–1364.

[67] Venditti, D. A., and Darmofal, D. L., "Anisotropic Grid Adaptation for Functional Outputs: Application to Two-Dimensional Viscous Flows," *Journal of Computational Physics*, Vol. 187, No. 1, 2003, pp. 22–46. doi:10.1016/S0021-9991(03)00074-3.

[68] Pierce, N. A., and Giles, M. B., "Adjoint Recovery of Superconvergent Functionals from PDE Approximations," *SIAM Review*, Vol. 42, No. 2, 2000, pp. 247–264.

[69] Kroll, N., Bieler, H., Deconinck, H., Couaillier, V., Ven, H., and Sorensen, K., *ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications: Results of a Collaborative Research Project Funded by the European Union, 2006-2009*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Springer, 2010. doi:10.1007/978-3-642-03707-8.

[70] Kroll, N., "ADIGMA – A European Project on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications," AIAA Paper 2009–176, 2009.

[71] Hartmann, R., Held, J., Leicht, T., and Prill, F., "Error Estimation and Adaptive Mesh Refinement for Aerodynamic Flows," *ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace Applications*, edited by N. Kroll, H. Bieler, H. Deconinck, V. Couaillier, H. van der Ven, and K. Sørensen, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, pp. 339–353. doi:10.1007/978-3-642-03707-8_24.

[72] Leicht, T., and Hartmann, R., "Error Estimation and Anisotropic Mesh Refinement for 3D Laminar Aerodynamic Flow Simulations," *Journal of Computational Physics*, Vol. 229, No. 19, 2010, pp. 7344–7366. doi:10.1016/j.jcp.2010.06.019.

[73] Klaij, C. M., van der Vegt, J. J. W., and van der Ven, H., "Space–time Discontinuous Galerkin Method for the Compressible Navier–Stokes Equations," *Journal of Computational Physics*, Vol. 217, No. 2, 2006, pp. 589–611. doi:10.1016/j.jcp.2006.01.018.

[74] Riley, A. J., and Lowson, M. V., "Development of a Three-Dimensional Free Shear Layer," *Journal of Fluid Mechanics*, Vol. 369, 1998, pp. 49–89.