Progress in CFD Discretizations Algorithms and Solvers for Aerodynamic Flows

Dimitri J. Mavriplis [*]

Department of Mechanical Engineering, University of Wyoming, Laramie, WY 82071

An overview of developments in computational fluid dynamic discretizations and solution techniques for aerodynamic applications over the past five to six year time frame is given in this paper. We begin with a survey of community efforts that have contributed to advancing the state-of-the-art of computational aerodynamics. Subsequently, an overview of developments in Reynolds-averaged Navier-Stokes (RANS) methods is given, including the use of higher-order discretizations. This is followed by a look at recent developments in scale-resolving methods from an algorithmic point of view. The paper concludes with prospects for further advances towards long term milestones and Grand Challenge projects as outlined in the CFD2030 report.

I. Introduction

In March 2014 the NASA commissioned Vision CFD2030 report was issued as a NASA Contractor Report (CR),[1] which provided a bold vision for future computational capabilities and their potential impact on aerospace engineering and design. The report formulated a series of four Grand Challenge problems designed to demonstrate the potential impact that advanced computational capabilities will have on aerospace engineering. One of the key features of the Vision 2030 report was the publication of a technology road map, with notional goals and milestones for specific technology achievements, tied to pacing aeronautical application problems, over the 16 year period from 2014 to 2030.

Within the AIAA, the recently formed CFD2030 Integration Committee (IC) is charged with overseeing and coordinating activities related to advancing CFD research and development activities along the lines proposed in the report. As one of its first activities, the IC sponsored the "Future CFD Technologies Workshop",[2] held in January 2018, which highlighted advances being made towards the CFD2030 Vision goals specifically in algorithmic technologies as they relate to improved simulation capabilities for aerodynamic applications. The advent of the year 2019 corresponds to the five year mark since the issuance of the Vision 2030 report, and the IC felt that a look back at progress made over the last five years in the various technological areas outlined in the report would serve as a useful exercise to gauge the rate of progress of technology towards the long term goals as outlined in the report. In this paper, we provide a survey of recent advances made in the area of CFD algorithms, namely discretizations and solution techniques. The technology road map produced in the CFD2030 Vision report, reproduced in Figure 1, tracks various technology areas that will be critical for advancing overall computational simulation capabilities over the next 10 to 15 years. The focus of the current paper is on the area denoted as "Algorithms" in the road map, as depicted in Figure 1. Thus, the current focus is principally on CFD discretizations and solver technology, and we do not attempt to cover related areas such as computational mesh algorithms (i.e. overset meshes, adaptive mesh refinement (AMR)), or other areas such as visualization and data extraction, physical modeling (turbulence, transition) and advances in HPC, except where these impact the performance and technological suitability of CFD discretizations and solution techniques.

The importance of algorithmic developments for computational methods cannot be overstated. Numerous reports have been produced over the last several decades emphasizing the fact that advances in computational algorithms have contributed equally or more compared to advances afforded by the sustainement of Moore's law for high-performance computing (HPC) hardware over the last four decades.[1,3–6] Perhaps even more

[*] Professor; email: mavriplis@uwyo.edu

importantly, since algorithmic complexity is most often an asymptotic property, the benefits of superior algorithms grow with problem size and the need for optimal algorithms becomes increasingly important as larger problem sizes are attempted on increasingly capable HPC hardware. However, algorithmic research and development requires a union of skills that covers not only the disciplinary application area (i.e. aerospace engineering in this case), but also the fundamental disciplines of applied mathematics and computer science. Often, developments in fundamental algorithmic technology have been neglected in favor of incorporating more capability or physics into existing legacy codes. As the CFD2030 report notes, many current CFD codes are based on algorithms that are over 20 years old, which in turn are being strained by the demands of supporting increased multidisciplinary applications as well as much larger problem sizes.

The road map illustrated in Figure 1 contains technology milestones which feed into notional technology demonstration problems over the 16 year period covered by the report. Within the context of algorithmic developments for CFD, the technology milestones include the development of "Automated Robust Solvers" by the current time frame (around 2018-2019), the appearance of "Scalable Optimal Solvers" around the year 2022, and the use of "Production scalable entropy stable solvers" by the year 2030. Although some of these milestones appear to be aspirational (especially their timelines), the reality is that significant progress in these areas will be required to fully realize the future computational capabilities envisioned in the report as well as to successfully tackle the Grand Challenge problems. The technology demonstrations most closely associated with algorithmic developments in the road map include the demonstration of realistic complex geometry applications using of the order of 10 billion grid points or degrees of freedom (per field equation) in 2020, 100 billion grid points or degrees of freedom (dofs) in 2025, and an even larger more complex simulation operating an order of magnitude above the sustained exaflop scale in the 2030 time frame.

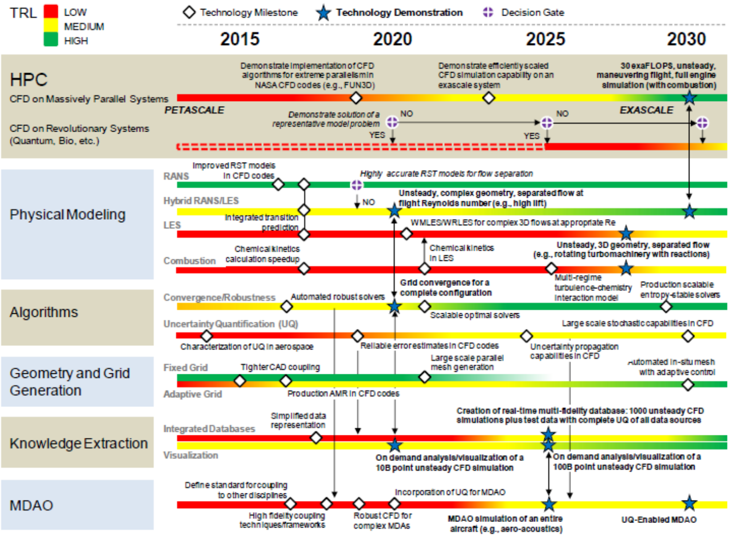

TRL
LOW
MEDIUM
HIGH
Technology Milestone
Technology Demonstration
Decision Gate
2015
2020
2025
2030
HPC
CFD on Massively Parallel Systems
Demonstrate implementation of CFD algorithms for extreme parallelism in NASA CFD codes (e.g., FUN3D)
Demonstrate efficiently scaled CFD simulation capability on an exascale system
30 exaFLOPS, unsteady, maneuvering flight, full engine simulation (with combustion)
PETASCALE
EXASCALE
CFD on Revolutionary Systems (Quantum, Bio, etc.)
Demonstrate solution of a representative model problem
NO
YES
NO
YES
Physical Modeling
RANS
Improved RST models in CFD codes
Highly accurate RST models for flow separation
NO
Hybrid RANS/LES
Unsteady, complex geometry, separated flow at flight Reynolds number (e.g., high lift)
LES
Integrated transition prediction
WMLES/WRLES for complex 3D flows at appropriate Re
Combustion
Chemical kinetics calculation speedup
Chemical kinetics in LES
Unsteady, 3D geometry, separated flow (e.g., rotating turbomachinery with reactions)
Multi-regime turbulence-chemistry interaction model
Algorithms
Convergence/Robustness
Automated robust solvers
Grid convergence for a complete configuration
Scalable optimal solvers
Production scalable entropy-stable solvers
Uncertainty Quantification (UQ)
Characterization of UQ in aerospace
Reliable error estimates in CFD codes
Uncertainty propagation capabilities in CFD
Large scale stochastic capabilities in CFD
Geometry and Grid Generation
Fixed Grid
Adaptive Grid
Tighter CAD coupling
Production AMR in CFD codes
Large scale parallel mesh generation
Automated in-situ mesh with adaptive control
Knowledge Extraction
Integrated Databases
Visualization
Simplified data representation
Creation of real-time multi-fidelity database: 1000 unsteady CFD simulations plus test data with complete UQ of all data sources
On demand analysis/visualization of a 10B point unsteady CFD simulation
On demand analysis/visualization of a 100B point unsteady CFD simulation
MDAO
Define standard for coupling to other disciplines
High fidelity coupling techniques/frameworks
Robust CFD for complex MDAs
Incorporation of UQ for MDAO
MDAO simulation of an entire aircraft (e.g., aero-acoustics)
UQ-Enabled MDAO

Figure 1. Technology development roadmap from Vision CFD2030 Report.

Reynolds-averaged Navier-Stokes (RANS) methods are now firmly established as the dominant approach for computational aerodynamics and their predictive ability for on-design conditions, with little or no regions or separated flow, is well documented.[7–11] The challenge for CFD in the coming years will lie increasingly in the prediction of off-design conditions involving large regions of turbulent separated flow, including the prediction of incipient separation.[1,12,13] Over the last decade, there has been a growing view that the successful prediction of separated flow cases will require the use of scale-resolving methods such as large-eddy simulations (LES) and/or detached-eddy simulations (DES)[14] and that continued investment in RANS turbulence modeling will only provide diminishing returns. This view is embodied in the recent external research funding awards under the NASA Revolutionary Computational Sciences (RCA) program, which mostly focus on techniques for scale-resolving methods. These trends are important for the current survey because steady (and unsteady) RANS approaches and time-dependent scale-resolving methods have significantly different requirements in terms of both discretizations and solution algorithms. However, due to the step increase in computational expense required by scale-resolving methods, RANS methods can be expected to remain the dominant approach for years to come, particularly when used as components in multidisciplinary simulations (i.e. aerostructural) or design optimization studies. Furthermore, the drive to LES methods will most likely be gradual, relying mostly on hybrid RANS-LES methods with increasing regions of LES for the foreseeable future. In this paper, we address progress and prospects for technology developments in both RANS (steady and unsteady) methods as well as scale-resolving methods, while highlighting the different requirements of both approaches.

In the remainder of the paper, we begin with an overview of progress in the state of the practice as assessed from various community workshops held over the last five year period. We then discuss in more detail specific technological advances for RANS methods, where we distinguish between second-order accurate discretizations and higher-order accurate methods, followed by a survey of progress in techniques for scale-resolving methods. We conclude with an assessment of the rate of progress and provide a future outlook towards meeting the CFD2030 objectives.

II. Community Activities

The highly successful Drag Prediction Workshop series came to a conclusion with the sixth workshop (DPW6) held in in Washington DC in June 2016.[11,15,16] Over its 15 year history, the DPW series documented the continuous progress of steady-state RANS methods at predicting drag at fixed lift conditions over industry relevant transonic cruise configurations. Some of the notable trends over this time period include the emergence and refinement of unstructured mesh solvers, a steady increase in the size of grids used for drag calculations, and the establishment of best practices for grid resolution requirements as well as grid refinement studies. Additionally, the DPW series played a large part in the creation of the Common Research Model (CRM) geometry,[17] which has been extended for other applications (e.g. high lift,[18] aerostructural[19]), and which continues to serve as a benchmark configuration for CFD validation studies. The workshop series has concluded that "the state-of-the-practice is to the point now where the absolute aerodynamic performance of a well-designed transonic transport at cruise conditions can be more accurately predicted by CFD than can be measured by a build-up from wind-tunnel data with corrections to flight Reynolds numbers".[20]

The high-lift prediction workshop series held its third and most recent workshop (HLPW3) in 2017 which served as a follow-on to the previous HLPW2 workshop held in 2013. While focusing on similar configurations (HL-CRM for HLPW2 and DLRF11 for HLPW2), it was noted that little overall improvement in our ability to capture $C_{L_{max}}$ for transport aircraft high-lift landing configurations was observed between the two workshops. Furthermore, the size of the finest grids employed did not increase appreciably between HLPW2 and HLPW3, which is perhaps surprising considering the four year period between the two workshops during which Moore's law could be expected to provide at least a four-fold increase in computational capability. On the other hand, some success was observed using scale-resolving simulations based on the Lattice Boltzmann approach.[21,22] These mixed results have led to a rethinking of the strategies and objectives for a follow-on HLPW4 workshop. Two principal thoughts are that the future workshop should focus on much finer grids and also seriously consider examining the predictive ability of scale-resolving methods over traditional steady-state RANS methods, particularly in the $C_{L_{max}}$ region.

The Hover Prediction Workshop (HPW) is a virtual workshop which consists of special sessions with published papers at AIAA meetings. Special sessions have been held at AIAA Scitech meetings every year from 2014 through 2019, totaling over 70 technical papers. The stated goal of the HPW is "...to inspire collaboration among industry, governments, and academia for the development of computational methods to predict all aspects of hovering flight efficiently, practically, and accurately".[23] The HPW studies have focused on standard helicopter rotor configurations (PSP rotor and the S-76) with different blade tip geometries, evaluating the accuracy of CFD to predict coefficients of thrust, power and figure of merit (FM) for a hovering rotor with near-transonic blade tip. A fundamental characteristic of the HPW is the fact that the focal problem is one of unsteady flow with moving geometry with a targeted emphasis on accurate rotor wake capturing. Contributions include results produced using unsteady RANS (URANS) solvers, as well as DES/LES methods, and combinations of these methods, often in conjunction with dynamic AMR.[24] A steady increase in capability has demonstrated the ability to capture hover performance within 1% accuracy in a relative sense. Still, several challenges remain: absolute performance predictions within 0.5% accuracy required by rotor-blade designers are still elusive. Contributing factors to the computational uncertainty include grid dependencies/convergence, computed vortex wake breakdown, accurate spanwise blade loading, installation effects, and transition effects. In addition, the lack of hover test data that accurately provides integrated performance, blade loads and unsteady vortex-wake structures hampers standardized code validation efforts.[25] The forthcoming NASA/Army HVAB hover tests promise to alleviate this need for comprehensive rotor-in-hover test data and will form the the basis for future HPW evaluations.

The second aeroelastic prediction workshop (AePW2) was held in January 2016[26] as a follow-on to the first AePW workshop held in May 2012.[27] The objective of the workshop was to assess the capability of current CFD tools to predict the onset of flutter over a simplified configuration consisting of a rigid wing (Benchmark Supercritical Wing or BSCW) mounted on a flexible pitch and plunge (PAP) apparatus. Since a precursor to accurate aeroelastic predictions is the ability to compute unsteady flows, a significant component of AePW2 (and the main focus of AePW1) was devoted to assessing the predictive ability of unsteady RANS CFD calculations through the inclusion of a prescribed pitching motion case for the BSCW. In this sense, both the AePW and HPW are relevant to any survey of CFD technologies, since they provide a focus on unsteady CFD capabilities which is lacking in the DPW and HLPW series. In these cases, not only spatial discretization errors must be considered (through grid refinement studies), but also temporal resolution and algebraic error (due to incomplete convergence at each implicit time step) must be studied to assess their impact on overall predictive ability.

Recently, there has been a growing interest in the possibility of the increased use of computational methods in general and CFD in particular for reducing the time and expense for aircraft certification. Within the AIAA, a Community of Interest (CoI) on Certification by Analysis (CbA) has been formed. One of the near term objectives of the CoI is the creation of a Recommended Practices standards document which would be used to guide the use of CbA by industry and streamline the acceptance of CbA by regulatory agencies in the determination of compliance. Additionally, NASA has issued a contract to a Boeing-led consortium of industry, academic and government partners to study the potential for CbA and outline a roadmap for the development of future technologies required to make CbA a reality. Notably, many of the candidate cases for CbA involve dynamic maneuver simulations, which will fully exercise the unsteady moving geometry CFD capabilities discussed above, possibly requiring the inclusion of aeroelastic effects.

One of the impediments outlined in the CFD2030 report is the fact that our ability to generate increasingly finer meshes has lagged behind advances in computational capability both from an HPC hardware standpoint and from CFD solvers themselves. As noted in the results from the high-lift workshop series, commonly used grid sizes have not increased substantially between the last two workshops and have plateaued in the region of 100 million points. The need for much finer computational meshes for difficult cases such as predicting $C_{L_{max}}$ has been discussed, and given the CFD2030 objective of computing on meshes with (O)10B to 100B points or degrees of freedom in the near future, significant advances in this area will be required. This call to action has been taken up by the AIAA Meshing, Visualization and Computational Environments TC, with two sponsored Geometry and Mesh Generation workshops (GMGW)now completed. The most recent, GMGW-2,[28] held in January 2019, included a session on exascale grid generation, with the objective of making production generation of grid sizes of the order of 30B cells or points a reality. Significant progress was demonstrated in GMGW-2, with several participants achieving the generation of grids of the order of 1B cells on the High-Lift CRM geometry using a variety of hardware platforms ranging from simple desktop machines to larger scale HPC hardware. As an alternative approach towards achieving more accurate simulations, higher-order accurate discretizations may be employed on coarser meshes with higher numbers of dofs per grid point or cell. However, to achieve their full potential, higher-order finite-element discretizations require the use of curved mesh elements which respect the underlying geometry. Thus the GMGW workshops have also promoted the development of mesh curving technologies,[29] which in turn have positively impacted the progress of high-order discretizations in CFD.

In addition to the above activities, there have been various community efforts that more directly impact CFD technologies. For example, the objectives of the International High-order Methods Workshop series include “providing an open and impartial forum for evaluating the status of high-order methods (order of accuracy > 2) in solving a wide range of flow problems” and “assessing the performance of high-order methods through comparison to production second-order CFD codes widely used in the aerospace industry with well-defined metrics”.[30] The first workshop was held in January 2012, with follow-up workshops held in 2013, 2015, 2016, and the most recent (fifth) workshop held in January 2018. The workshops traditionally have consisted of a series of test cases ranging from simple canonical flows up to more challenging industrial aerodynamic configurations. Of particular interest to this survey is the inclusion of the CRM at transonic cruise conditions, taken from the DPW series, and high-lift test cases (DLR-F11 and HL-CRM) taken from the HLPW series. In the most recent workshop (HiOCFD5),[30] various participants demonstrated solutions using continuous finite-element (Streamwise Upwind Petrov Galerkin or SUPG) and discontinuous Galerkin (DG) discretizations up to third (p=2) and fourth-order accuracy (p=3) on the transonic CRM and high-lift cases. The workshop results firmly established the feasibility of using these high-order discretizations for industrial aerodynamic CFD problems, while demonstrating impressive accuracy gains on relatively coarse meshes, particularly in going from second-order (p=1) to third-order accurate (p=2) discretizations. On the other hand, the workshop did not focus strongly on computational expense, and it is widely acknowledged that these high-order accurate solutions are relatively expensive compared to traditional second-order accurate methods on a cost per degree-of freedom basis. While the true metric of importance is cost per equivalent level of accuracy, such metrics have not been thoroughly studied to date. On the other hand, the workshop also demonstrated significant advantages for high-order methods when used as scale-resolving methods for wake capturing and large-eddy simulations (LES) of canonical problems with well-defined accuracy metrics such as the Taylor Green Vortex problem. Some of the conclusions that can be drawn from the workshop series are that, for steady-state RANS problems, improved linear and non-linear solution algorithms are pacing items that are needed to make these approaches more competitive, while for scale-resolving methods nonlinear stability at high order remains a concern.

Within the AIAA, a CFD solver discussion group was formed several years ago, which has been very active in creating a forum for discussion and assessment of various aspects of RANS solver technology, including the assessment of existing capabilities, identification of critical technology gaps, and establishing methodologies for assessing advanced methods for RANS solutions. In particular, the group has established a set of well resolved reference solutions for verification of RANS solvers,[31] which can be found on the NASA turbulence modeling resource web site.[32] The Discussion Group has also established a consistent methodology for the assessment of grid convergence for both structured and unstructured grids using different element types, and has created and distributed efficient grid generation and coarsening programs for the benchmark cases in support of this endeavor.[33] Finally, the Group continues to provide assessments of new discretization schemes, mesh adaptation approaches, and advanced iterative convergence techniques through special sessions organized at SciTech-2015, SciTech-2016, SciTech-2018 and a special section publication in AIAA Journal (2017).[34]

The above overview of community activities provides a high-level view of advances made in the state-of-the-art of CFD methods for aerodynamic problems. In the following sections we provide a more detailed look at some specific technological advances and challenges in CFD discretizations and solution techniques. As the above discussion has shown, these two areas are intertwined and must be considered in unison. For practical purposes, the following discussion is separated into a first part which focuses on RANS and unsteady RANS (URANS) methods, and a second part which considers scale-resolving methods. This is due to the fact that the critical technologies, limiting factors and outstanding challenges are significantly different for these two approaches, both of which will play a significant role in computational aerodynamics for the foreseeable future.

III. Reynolds-Averaged Navier Stokes (RANS) Methods

One of the fundamental challenges in CFD in the current era is extending the success CFD has had in the prediction of on-design conditions, as documented in the DPW series, to the outer edges of the flight envelope where cases with significant areas of turbulent separated flow must be predicted accurately. Recently, several workshops have been held focusing on the successes and limitations of turbulence modeling for separated flows, including the University of Michigan workshop on Advances in Turbulence modeling held in July 2017[35] and the NASA Sponsored CFD Prediction Error Assessment Workshop[36] held in March 2018. Part of the outcome of these activities is a growing realization that scale-resolving methods may be required to achieve suitable predictive accuracy for such cases. However, RANS methods can be expected to remain the dominant approach in computational aerodynamics over at least the next decade, due to the step increase in computational expense required in moving to DES or LES methods. Furthermore, RANS methods will remain as essential building blocks in multidisciplinary analyses (.e.g. aero-elastic, aero-servo-elastic, aero-thermal-elastic etc.) as well as in single and multidisciplinary optimization problems. At the same time, current RANS technologies are being stressed by the drive to higher accuracy and larger meshes, all of which expose deficiencies particularly in linear and nonlinear solution technologies, and in the efficiency with which these port to emerging HPC architectures. For all of these reasons, it is important to continually evaluate the capabilities of new and emerging RANS technologies and to advocate for further development in these areas.

In the following section, we first look at second-order accurate RANS methods, including both finite-volume and finite element (p=1 SUPG) discretizations. In a second part, we look at higher order SUPG and DG discretizations and solvers applied to RANS aerodynamic problems.

A. Second-Order RANS Methods

The inclusion of p=1 SUPG methods alongside traditional second-order accurate finite-volume (FV) and finite-difference (FD) methods is motivated by the fact that, on the same computational mesh, these methods result in the same number of degrees of freedom. Additionally, SUPG methods do not require mesh curving at p=1 and can therefore operate on the same meshes as traditional methods. (We note that this is not the case for p=1 DG discretizations, which are relegated to the following section on higher-order methods). Additionally, while FV (and FD) methods dominate production CFD RANS methods, p=1 SUPG methods are also being used successfully in production mode.[37,38]

Over the last several years, there has been growing interest in SUPG discretizations for production RANS CFD codes.[38–43] Earlier work[44] pointed to the efficiency advantages of SUPG discretizations at low p orders over corresponding DG discretizations on equivalent meshes. These advantages have now become more well accepted with further comparisons of schemes for increasingly complex RANS problems.[45] The lower cost of SUPG methods is a direct result of the larger number of degrees of freedom inherent in a DG p=1 discretization compared to a p=1 SUPG discretization on the same mesh. SUPG discretizations at p=1 have conclusively demonstrated higher accuracy compared to traditional second-order accurate finite-volume discretizations on equivalent meshes.[32,46,47] Furthermore, SUPG discretizations appear to be less sensitive to grid irregularities, as witnessed by the ability of these discretizations to produce competitive solutions for high-Reynolds number RANS problems using fully tetrahedral elements in the highly stretched boundary layer regions of the mesh, versus the requirement of using prismatic elements in these regions for finite-volume based approaches. Although the availability of meshes with prismatic layers in boundary-layer regions is now widespread, the ability to employ tetrahedral elements throughout the computational domain offers advantages particularly for use with anisotropic AMR techniques, where most progress has been made using fully simplicial (i.e. tetrahedral in 3D) elements.[48,49]

On the other hand, it can be said that, in many cases, the computational cost in solving steady-state RANS problems using p=1 SUPG discretizations is still significantly higher than what can be achieved with state-of-the-art unstructured mesh finite-volume solvers. This may be due in large part to the fact that SUPG discretizations are most often converged to steady state using robust Newton-Krylov solvers, since weaker solvers such as point-implicit or nonlinear multigrid methods often fail to converge for these discretizations. In fact, when used in conjunction with finite-volume discretizations, Newton-Krylov solvers also incur substantial additional expense over simpler solvers in what is a classic trade-off between robustness and efficiency.[50,51] Whether the enhanced accuracy of SUPG discretizations is worth the added cost, and/or whether more efficient solvers can be devised for these discretizations remain open research questions today. Nevertheless, research on solvers for SUPG (and DG discretizations) has had a beneficial effect on the development of efficient and robust solvers for traditional finite-volume discretizations, as Newton-Krylov techniques originally developed for these finite-element discretizations have been applied increasingly to FV discretizations.

In general, solution approaches for steady and unsteady RANS problems can be formulated using local nonlinear solvers, inexact Newton schemes with associated linear solvers, and as exact Newton or Newton-Krylov methods with nonlinear continuation techniques. To describe these methods we begin by writing the equations to be solved as:

$$\frac{\partial w}{\partial t} + R(w) = 0 \tag{1}$$

where w represents the flow variables or degrees of freedom (dofs) and R(w) represents the spatial residual which is a nonlinear operator or function of w. The first step consists of adding a pseudo-time term to the equations resulting in

$$\frac{\partial w}{\partial \tau} + \frac{\partial w}{\partial t} + R(w) = 0 \tag{2}$$

For time-dependent problems solved using implicit time stepping, the physical time derivative $\frac{\partial w}{\partial t}$ must also be discretized, and this is usually accomplished using a second-order backward-difference discretization

(BDF2), although other temporal discretizations such as implicit Runge-Kutta and time-spectral methods have also been used. For simplicity, for now we consider the steady-state case in which the physical time derivative vanishes. In this case we can write a general implicit solution method as:

$$\frac{M(w^{n+1} - w^n)}{\Delta\tau} + R(w^{n+1}) = 0 \tag{3}$$

Solving this via a Newton scheme yields

$$\begin{aligned} \left[\frac{M}{\Delta\tau} + \frac{\partial R}{\partial w}\right] \Delta w^{n+1} &= -R(w^n) & (4) \\ w^{n+1} &= w^n + \alpha \Delta w^{n+1} & (5) \end{aligned}$$

where M is a suitable mass matrix (which reduces to the cell volume for FV schemes), and α is a parameter between 0 and 1 which is used to under-relax the update as needed. At each step of the Newton scheme, the left-hand side of equation (4) is inverted (approximately) to obtain the update Δw^{n+1}, which in turn is used to advance the nonlinear solution. For an exact Newton scheme the $\frac{\partial R}{\partial w}$ matrix represents an exact linearization of the right-hand-side residual operator $R(w)$, whereas for inexact Newton methods, an approximate linearization is used. For finite-volume schemes, inexact Newton schemes using first-order Jacobian approximations based on a nearest-neighbor stencil represent the most common approach. Generally the left-hand-side is inverted iteratively, by splitting the Jacobian as

$$\left[\frac{M}{\Delta\tau} + [D]\right] \Delta w^{k+1} = -R(w^n) - [O]\,\Delta w^k \tag{6}$$

where k denotes the index of the inner (linear) iteration process, which converges such that $\Delta w^k \rightarrow \Delta w^{n+1}$ for $k >> 1$. The splitting $\frac{\partial R}{\partial w} = [D] + [O]$ is constructed such that $[D]$ is easily inverted. For example if $[D]$ is chosen as the block diagonals of the Jacobian matrix, the above is easily solved through local inversion of each block matrix associated with each grid point or control-volume. Alternatively, $[D]$ may represent a block tridiagonal matrix resulting from the identification of line structures in the mesh, which is also easily inverted using the Thomas algorithm.[52] Simple local nonlinear solvers may be constructed by performing a single subiteration on the linear system with Δw^0=0, obviating the need to store the $[O]$ matrix, and updating the nonlinear residual at each step as:

$$\begin{aligned} \left[\frac{M}{\Delta\tau} + [D]\right] \Delta w^{n+1} &= -R(w^n) & (7) \\ w^{n+1} &= w^n + \alpha \Delta w^{n+1} & (8) \end{aligned}$$

Alternatively, equation (6) can be partially solved using a small number of linear iterations while the pseudo-time step is gradually increased at each outer nonlinear iteration in order to obtain a solver which approaches the steady-state residual equation R(u) = 0 at high values of $\Delta\tau$ (or CFL, which is used to scale the pseudo-time step).

For a Newton-Krylov scheme, equation (4) is solved using a Krylov method,[53] which only requires the product of the left-hand side Jacobian matrix with a vector. This facilitates use of the exact linearization particularly for FV schemes with extended stencils since the matrix-vector product can be assembled efficiently on the fly from stored matrix subcomponents or by finite-differencing the residual vector.[54] Since SUPG and DG methods employ a nearest neighbor stencil, Newton schemes with exact linearizations are usually employed directly.

Because the success of Krylov methods rests on the ability to employ an efficient preconditioner, the linear solver techniques described above are generally used as preconditioners for the Krylov method. Whether this approach is used to construct an iterative nonlinear solver, an inexact Newton scheme, or a Newton-Krylov scheme, the splitting $\frac{\partial R}{\partial w} = [D] + [O]$ represents the weakest component of the solution strategy, since this corresponds to a local technique (either as a nonlinear solver, linear solver or preconditioner), and the overall solution strategy cannot be expected to scale optimally as the problem size is increased. Using an ILU factorization of the Jacobian matrix (possibly with varying levels of fill-in) represents a more global approach to the linear solution or preconditioner phase of the solver. However, ILU techniques are inherently sequential and are most often implemented individually on each mesh partition for distributed memory parallel applications. When large numbers of processors are used, the effectiveness of the ILU approach is diminished for these reasons. Additionally, use of ILU with high levels of fill incurs substantial memory overheads making these methods ill-suited for use on emerging massively parallel computer architectures. On the other hand, shared memory ILU implementations have shown some success as preconditioners for Newton-Krylov methods applied to SUPG discretizations.[55]

Alternatively, the local nonlinear solvers, linear solvers or preconditioners may be used as smoothers within a multigrid algorithm. For example, a nonlinear full approximation storage (FAS) multigrid solver based on a nonlinear line-smoothing algorithm has been developed by the author in the 1990's and used successfully for unstructured mesh RANS problems.[56–59] Nonlinear multigrid methods were initially very popular in both structured and unstructured CFD RANS solvers, but have fallen out of favor lately due to robustness issues with nonlinear multigrid methods for difficult problems. Alternatively, an analogous linear multigrid solver can be used to accelerate the solution (or preconditioning) of the linear system to be solved at each step of a Newton solver making the overall Newton solver more scalable.[50,51]

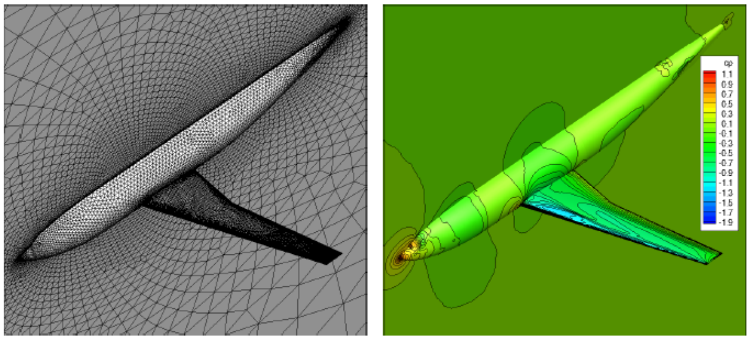
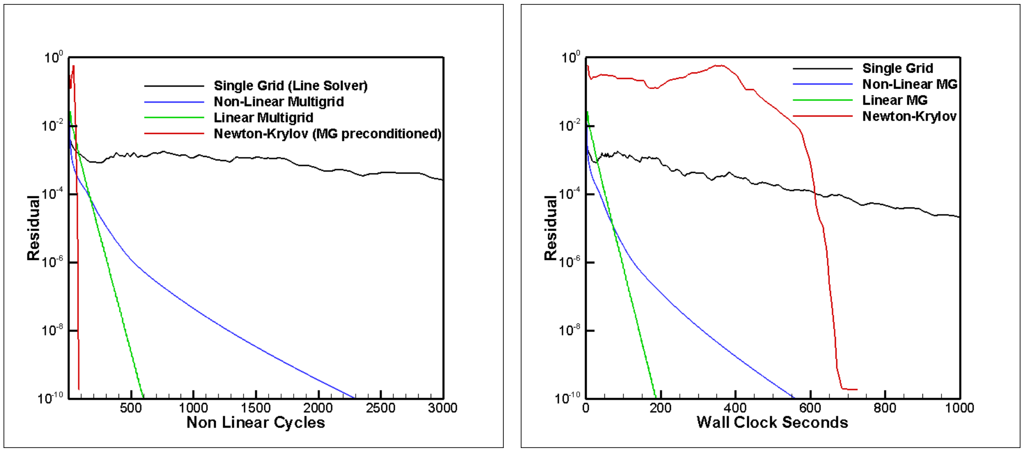


Figure 2. Illustration of unstructured mixed element mesh of approximately 1.2 million points and steady-state transonic flow solution for DLRF6 wing-body test case for Mach=0.75, Re=3 million, Incidence = 1^o

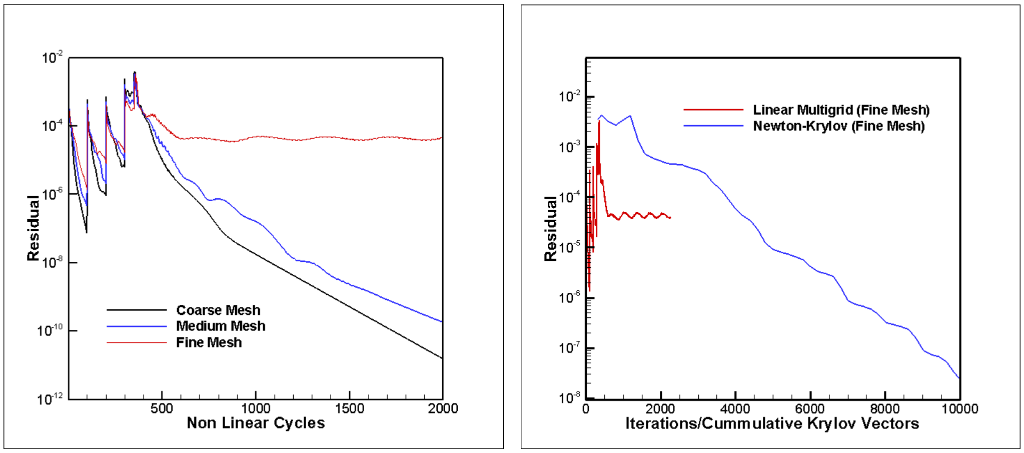


(a)

(b)

Figure 3. Convergence histories for various solvers in terms of (a) nonlinear cycles and (b) wall-clock time for DLRF6 test case.

Figures 2 and 3 illustrate the convergence obtained using the NSU3D code developed by the author for a relatively easy problem, namely steady-state transonic flow over a wing-body configuration on a mixed element (prism-pyramid-tetrahedra) unstructured mesh of 1.2 million points.[51] The baseline solver is a line-implicit method, where $[D]$ in equation (6) corresponds to a block-tridiagonal matrix obtained by identifying line structures in the boundary layer regions of the mesh.[58] When used as a simple nonlinear solver, convergence is relatively slow, requiring over close to 10,000 iterations to reach machine zero residuals (although force coefficients converged to engineering precision are obtained at a fraction of this cost). However, when this approach is used as a smoother for a nonlinear full-approximation storage (FAS) multigrid method,[60] convergence is much faster, achieving ten orders or magnitude reduction in the residuals in just over 2000 multigrid cycles. Alternatively, the analogous line smoother may be used to drive a linear multigrid solver employed within an inexact Newton scheme (based on a first-order Jacobian approximation in equation (4)). Using four linear line-smoothing sweeps on each mesh level, and three linear multigrid cycles for each nonlinear update, convergence to the same level (10 orders) is obtained in just 600 nonlinear steps. Finally, using an exact Newton-Krylov approach, where the linear system is preconditioned with the aforementioned linear multigrid solver, convergence to steady state (ten order reduction) is obtained in just 80 nonlinear updates. Figure 3(b) compares these different solution techniques in terms of cpu time, running on 128 cores of a modern day computer cluster, showing that the inexact Newton scheme with linear multigrid provides the fastest solution time, while the Newton-Krylov method provides rapid final convergence but requires more time overall due to its slow initial convergence.



(a)

(b)

Figure 4. Convergence histories for linear multigrid on coarse, medium and fine meshes and (b) comparison of fine mesh convergence for linear multigrid versus Newton-Krylov method for DLR-F11 high-lift test case from HLPW2.

As mentioned previously, for more difficult cases the multigrid algorithms often fail to fully converge. This is illustrated in the example depicted in Figure 4, for a high-lift case taken from HLPW2. Here we use a sequence of three progressively finer meshes for the DLR-F11 Rev 2 (no flap track fairings) configuration, consisting of 10 million, 30 million and 76 million grid points for the coarse, medium and fine meshes respectively. These meshes were supplied as workshop meshes (and are available at the workshop website).[61] For this case, the nonlinear multigrid algorithm fails to fully converge on the medium and fine meshes. The linear multigrid method converges well for the coarse and medium mesh, providing near grid independent convergence rates. On the finest mesh, initial convergence follows that achieved on the coarse and medium meshes, but eventually stalls out and fails to reach full convergence. Conversely, the multigrid preconditioned Newton-Krylov solver can be used to fully converge the fine mesh problem, demonstrating the increased robustness of this approach. However this comes at significant additional cost, as shown in Figure 4(b) where the linear multigrid and Newton-Krylov convergence histories for the fine mesh are plotted in terms of iterations for the linear multigrid solver and cumulative Krylov vectors for the Newton-Krylov solver, which provides a comparative measure of computational effort for both schemes. Using the Newton-Krylov solver to fully converge this problem requires four to five times more computational resources that would have been required by the linear multigrid solver if it had not stalled.

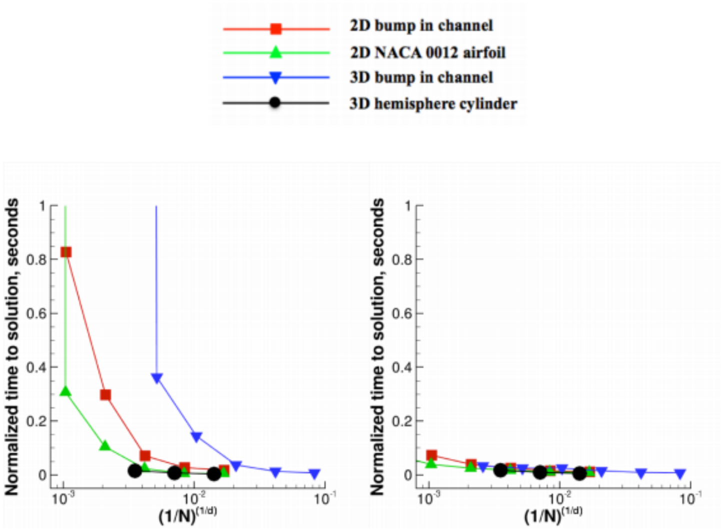
The above examples illustrate a hierarchy of increasingly robust solvers, progressing from nonlinear multigrid to fully implicit exact Newton methods. However, this robustness often comes at a cost in computational resources as shown in this example. Furthermore, the number of solver parameters that are used to tune the solvers increases as one proceeds along this hierarchy. At the lower end, the nonlinear multigrid algorithm requires virtually no tuning parameters (except for the number of non-linear cycles per grid level), while the inexact Newton solver requires additional settings for the number of linear cycles or a linear system exit tolerance, as well as a growth strategy for the pseudo-time step. Finally, the exact Newton-Krylov solver requires setting the number of Krylov vectors, or a linear system solution exit tolerance, in addition to the number of sweeps for the linear multigrid preconditioner, and a prescription of a suitable continuation strategy. In this case a pseudo-transient continuation with nonlinear line search has been used as discussed below. While these tuning parameters allow for optimization of the solver for different test cases, they can also easily lead to excessively long solver run times, for example by requiring over-solving of the linear system at particular phases along the nonlinear solution path.

However the Newton-Krylov approach has some built-in advantages as an overall solution strategy. In the final phases of convergence, when the residual and nonlinear update are small, the obtained reductions in the linear system residual are translated into nearly equivalent reductions in the nonlinear residual, since the neglected higher order terms in the Taylor series expansion at the core of the derivation of equation (4) become truly small. Thus over-solving of the linear system becomes less of a concern. Furthermore, the underlying linear iterations are more cost-effective than equivalent nonlinear iterations (for example linear multigrid compared to nonlinear multigrid) because linear solver iterations can be formulated as a matrix-vector product, which amortizes initial costs of matrix factorization and avoids recomputing any costly nonlinear terms in the residual vector.[50] Additionally, Krylov methods can be used to guarantee monotone linear system convergence[62] (although stagnation may occur), whereas no such proofs exist for nonlinear systems. On the other hand, the principal disadvantage of Newton-Krylov methods is that they provide slow initial convergence and spend most of their time in the nonlinear continuation phase. Thus, in order to develop more efficient and robust solvers, improvements in both linear and nonlinear solution techniques are required.

1. Linear solvers and Preconditioners

Efficient linear solvers are required to solve the linear problem arising at each non-linear step in inexact or exact Newton solvers. For finite-volume discretizations, a common approach consists of using an approximate Jacobian matrix corresponding to the linearization of a first-order discretization based on a nearest neighbor stencil. This simplified linear system can then be inverted approximately using iterative techniques or matrix factorization methods. Alternatively, a Newton-Krylov method can be used to solve the linear problem arising from the exact linearization based on a Krylov method. Since Krylov methods only require the evaluation of Jacobian-vector products, this can be achieved either by finite-differencing the residual,[54] or by storing subcomponents of the Jacobian matrix which can be used to assemble exact Jacobian-vector products for extended stencil discretizations. In this context, the same linear iterative or matrix factorization solver techniques are employed indirectly as preconditioners for the Krylov method. A principal advantage of p=1 SUPG discretizations is that they result in a nearest neighbor stencil, and the usual practice is to employ a Newton method with iterative or matrix factorization preconditioners based on the exact Jacobian matrix. For finite-volume schemes, simple iterative linear solvers such as block Jacobi, colored block-Gauss-Seidel, or block line solvers are often used. Significant efficiency gains have been reported with the use of more implicit linear solvers such as block line solvers, particularly when these are used as linear solvers within a Newton scheme rather than directly as local nonlinear solvers. As an example, Figure 5 illustrates the time required for solution convergence of various standard test cases for a point-implicit versus a line-implicit iterative linear solver strategy used as a preconditioner within the Hierarchical Adaptive Nonlinear Iterative Method (HANIM) solver for the NASA USM3D code.[63] HANIM employs a Newton-Krylov strategy with robust pseudo-transient continuation (PTC) strategy. The horizontal axis in the figure corresponds to a measure of grid size or resolution, and the vertical axis corresponds to solution time normalized by the number of grid points or mesh size. For all three test cases, the line-implicit approach is seen to deliver convergence rates which are much less sensitive to increases in grid resolution compared to the point-implicit approach.

As an alternative to local iterative schemes, sparse matrix factorization techniques have also been employed extensively as linear solvers for CFD problems. Incomplete LU factorizations with varying levels of fill (ILU(k)) have been applied to FV discretizations[64–66] and are perhaps now the most common linear solver or preconditioning approach used for the solution of p=1 and higher SUPG and DG discretizations.[38, 39, 42, 45, 67, 68] These methods are more global in nature than the aforementioned local iterative solvers. The principal drawback is that they are inherently sequential algorithms and are usually used in a partition-local sense in a distributed memory parallel computing environment. This leads to a breakdown of numerical convergence effectiveness for large processor counts, although Schur complement approaches [66] or using overlap within an alternating Schwarz strategy can mitigate this effect to some degree.[69, 70] Additionally, ILU methods are often unstable when applied iteratively and are therefore most often used as a preconditioner for Newton-Krylov methods. One advantage of ILU methods is that one can easily select



(a) point-implicit HANIM

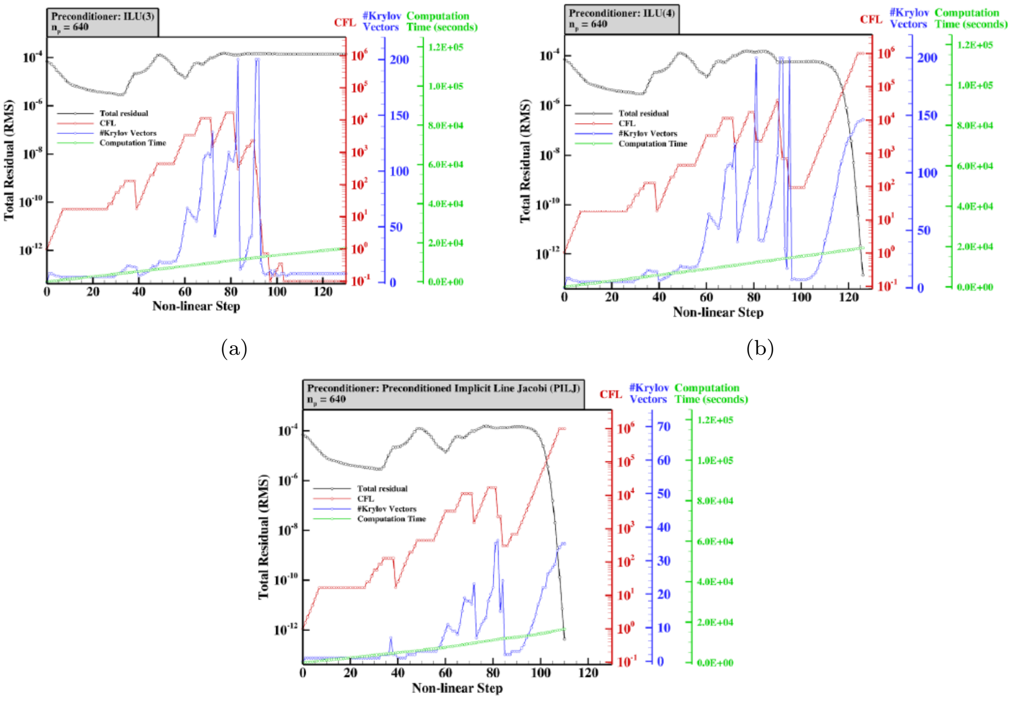
(b) line-implicit HANIM

Figure 5. Variations of normalized CPU time to solution with respect to degrees of freedom (N) for point and line-implicit solver in NASA USM3D code. The CPU time is normalized by N. The parameter d is case dimension.Reproduced from[63]

increasing levels of fill to produce a stronger (more global) preconditioner, although this comes at the expense of increased memory overheads. Figure 6 illustrates the solution of a p=1 SUPG discretization, for transonic flow over the DPW CRM configuration on a fully tetrahedral mesh of approximately 6.2 million vertices (36.6 million elements), using ILU(k) distributed over 640 processors or cores. At this level of partitioning, the effectiveness of the ILU preconditioner is weakened. Using k=2 results in a non-convergent solver, and three levels of fill (k=3) are needed to produce a convergent algorithm as shown in Figure 6.

As mentioned previously, one approach to improving the scalability of ILU methods is to use overlap within a Schwarz method[69] or a Schur complement technique[66] to solve more exactly across partition boundaries. As an alternate strategy, we have explored the use of iterative line-implicit preconditioners for use with p=1 SUPG discretizations.[71] Devising stable iterative solvers for SUPG discretizations has proven to be problematic, likely due to the lower diagonal dominance of the Jacobian matrix resulting from the linearization of the SUPG discretization (compared to FV). Our approach consists of formulating the preconditioner as a defect-correction scheme, where the left-hand side implicit matrix to be inverted is constructed using the Jacobian elements along lines formed in the unstructured mesh with an added pseudo-time step on the diagonal based on a lower CFL value (or pseudo-time step) determined for stability of the iterative scheme. In this formulation, the right-hand side of the defect-correction scheme consists of the residual of the linear system to be solved by the Krylov method, which employs a larger CFL value or pseudo-time step sizes determined by the pseudo-transient continuation of the Newton solver. This dual CFL approach is found to provide a stable iterative preconditioner which delivers convergence rates which are independent of the number of processors used (provided the partitioning does not break any of the line structures in the mesh), and incurs substantially lower memory overheads compared to the ILU preconditioner. Results from a study on the aforementioned CRM test case are given in the table presented in Figure 7, where the ILU preconditioner is seen to require increasing levels of fill to produce a convergent solver at high processor counts, while also being constrained at low processor counts by available memory. In comparison, the iterative line preconditioner requires approximately the same time to solution as the ILU preconditioner, but is more scalable, in that it produces the same convergence (shown in Figure 6) for all processor counts and is not memory constrained.

Noting that the ILU factorization of a tridiagonal matrix is exact (i.e. no fill in occurs), further work has explored the idea of defining sub-structures within the local Jacobian matrices that can be factorized either exactly or approximately with minimal fill-in using the ILU(k) approach. The explored method can be viewed as a generalization of the implicit line-smoothing technique by extending the groups of implicitly-solved unknowns from lines to blocks, while the blocks are formed by partitioning the computational domain

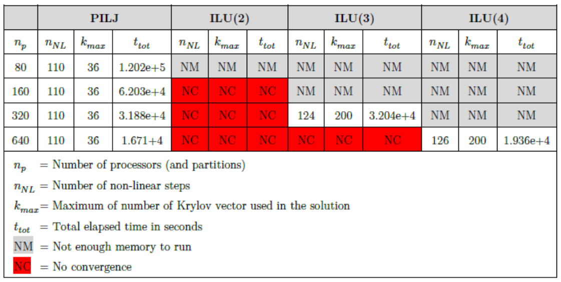


(a)

(b)

(c)

Figure 6. Convergence histories for DPW4 test case using SUPG p=1 discretization for (a) Newton-Krylov solver with ILU(2) preconditioner, (b) ILU(3) preconditioner and (c) Implicit Line Jacobi preconditioner (PILJ)



	PILJ			ILU(2)			ILU(3)			ILU(4)		
n_p	n_{NL}	k_{max}	t_{tot}	n_{NL}	k_{max}	t_{tot}	n_{NL}	k_{max}	t_{tot}	n_{NL}	k_{max}	t_{tot}
80	110	36	1.202e+5	NM	NM	NM	NM	NM	NM	NM	NM	NM
160	110	36	6.203e+4	NC	NC	NC	NM	NM	NM	NM	NM	NM
320	110	36	3.188e+4	NC	NC	NC	124	200	3.204e+4	NM	NM	NM
640	110	36	1.671+4	NC	NC	NC	NC	NC	NC	126	200	1.936e+4

n_p = Number of processors (and partitions)
n_{NL} = Number of non-linear steps
k_{max} = Maximum of number of Krylov vector used in the solution
t_{tot} = Total elapsed time in seconds
NM = Not enough memory to run
NC = No convergence

Figure 7. Convergence characteristics of ILU Newton-Krylov method using increasing fill levels on larger numbers of partitions

such that the strong connections between unknowns are not broken by the partition boundaries.[72] The resulting method is referred to as the block ILU smoothing method. Figure 8 illustrates the convergence obtained on a grid of 2 million points (12 million tetrahedra) for the High-Lift CRM using the block ILU smoother. In this case the blocks contain approximately 2000 mesh points and ILU(0) factorization was used. The block ILU smoother is applied iteratively, using 200 smoothing passes per Krylov vector, and convergence to machine precision is obtained in just under 80 nonlinear steps.

Although iterative techniques including line solvers have been successful in accelerating convergence of production CFD codes for many cases, these methods still correspond to local techniques, and cannot be expected to deliver optimal convergence for problems on increasingly finer meshes. One approach to extending the efficiency of iterative linear solvers is to use them as smoothers on each level of a linear multigrid algorithm, which is used in turn as a solver for inexact Newton methods, or as a preconditioner for Newton-Krylov solvers. This is the approach taken in the results presented in the previous section using NSU3D, where the inexact Newton solver is driven by a linear agglomeration multigrid method using block line-smoothers on each level, while this same approach is used as a preconditioner for the Newton-

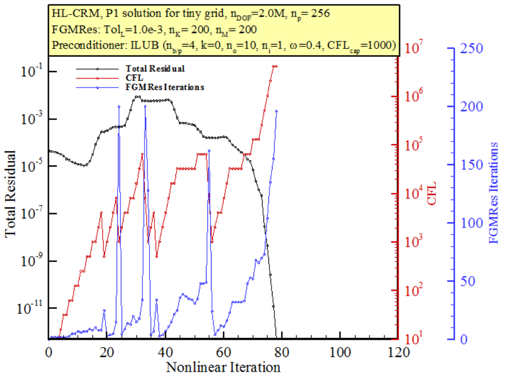


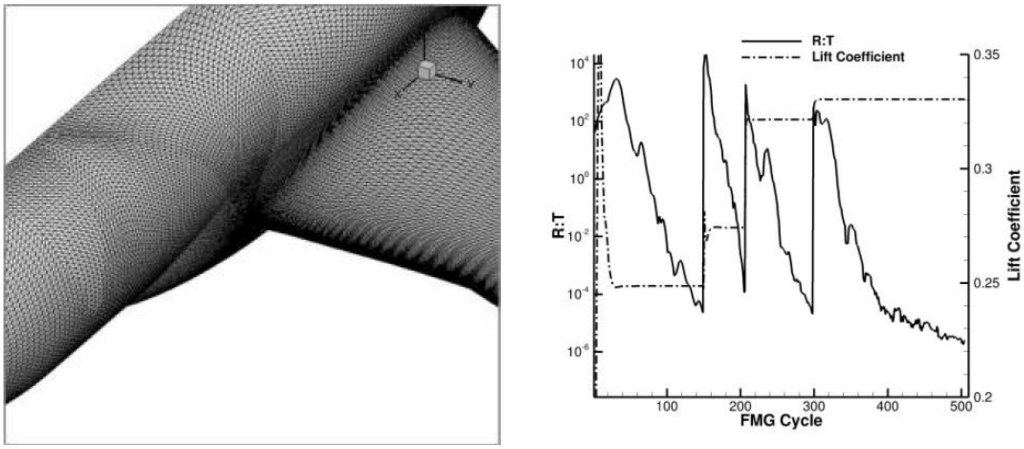
Figure 8. Convergence history of ILUB method for HL-CRM test case on mesh of 2 million points for p=1 SUPG discretization.

Krylov solver in those examples. Although there is little evidence of significant new developments in linear multigrid methods in the aerospace engineering literature, there has been a sustained effort within the applied mathematics community and the US Department of Energy (DoE) communities to develop general purpose scalable linear solvers based on algebraic multigrid (AMG).[73–76] Preliminary attempts at using some of these libraries for second-order accurate CFD discretizations have been encouraging, but overall performance still lags that achieved with custom implementations.[71] Part of this may be due to optimization of solver tuning parameters in these libraries and studies are on-going. However, in many cases, existing AMG libraries have struggled to produce competitive solutions on the linear systems arising from compressible flow CFD problems. In particular, the coarse grid operators resulting from the linearization of CFD discretizations have often been found to be unstable for the usual iterative solvers such as Gauss-Seidel.[77] These difficulties may be due to the fact that some of the heuristics used in custom CFD linear solvers, which often arise from intuition or physical reasoning, are not available for general purpose linear solvers. However, it is also well known that many commercial CFD products make use of algebraic multigrid linear solvers as components in their overall solution strategy.[78] These issues point to the need for more targeted research for the development of scalable linear solvers for compressible flow problems.

2. Nonlinear solver techniques

There have been significantly fewer developments for nonlinear solution techniques over the last decade compared to linear solvers. This may be due to the fact that linear solvers can be more easily abstracted out from application codes and addressed as applied mathematics problems. Furthermore, advances in linear solver techniques can be incorporated into libraries which can be reused by diverse sets of application codes.[79] On the other hand, the success or failure of nonlinear solution techniques is often intricately tied to the specific application at hand.

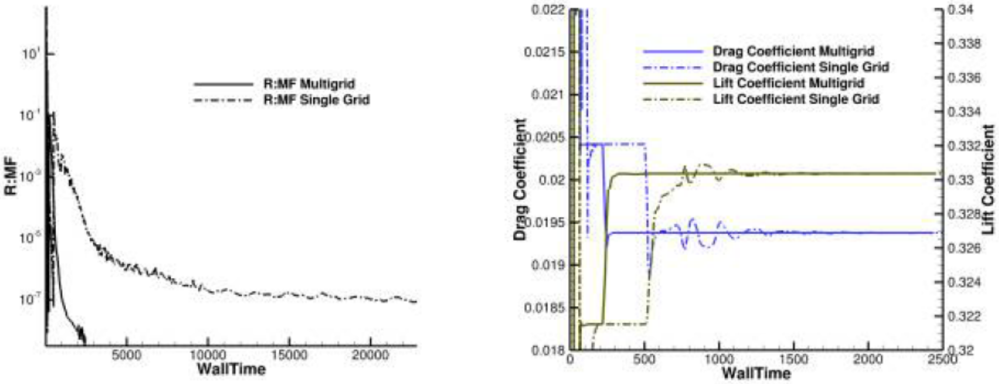
Nonlinear multigrid solvers such as the full-approximation storage (FAS) multigrid method are appealing because in many instances they require few ad-hoc parameters and are known to be optimal solvers for model problems.[60,80] Research in these methods for CFD problems has decreased significantly since key advances were made more than 20 years ago, and a search of multigrid methods in AIAA publications over the last 6 years returns only 10 entries. Nevertheless, many production CFD codes contain a nonlinear multigrid solver capability including the author's NSU3D code,[51] the DLR tau code,[81] the ONERA elSA code ,[82] the JAXA FaSTAR code[83] and the NASA production codes FUN3D and Overflow .[84,85] In particular, the multigrid solver developed in the NASA FUN3D code[84] differs from the usual multigrid literature in that a strong Newton-Krylov solver with associated linear preconditioners is used on each level of the multigrid sequence, as opposed to the more common approach of using simple local nonlinear smoothers on each level.[80] Figure 9 reproduced from reference[84] demonstrates consistent rapid convergence for the transonic CRM test case from DPW5 on a sequence of progressively finer meshes with the finest mesh containing 5.2 million points. In this case, the multigrid algorithm has been optimized for structured meshes, although an agglomeration multigrid method for general unstructured meshes is also described. Figure 10 illustrates the speedup in wall-clock time obtained using the multigrid algorithm compared to the single grid solver showing close to an order of magnitude efficiency gain for this test case. However, as with most production CFD codes, whether these gains can be maintained for increasingly complex industrial test cases on finer unstructured meshes remains to be demonstrated.



(a)

(b)

Figure 9. Illustration of structured surface mesh for CRM configuration and convergence history on sequence of progressively finer meshes using nonlinear multigrid approach implemented in FUN3D code. Reproduce from[84]



(a)

(b)

Figure 10. Comparison of single versus multigrid convergence for CRM test case. Reproduced from[84]

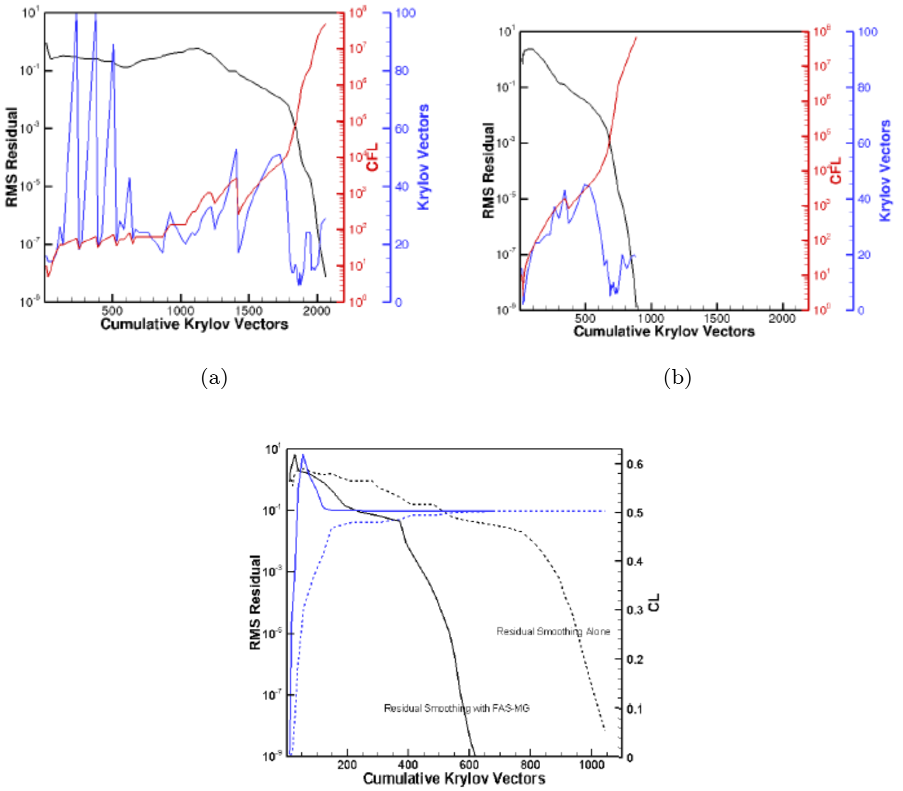
With regards to solver robustness, one of the important developments over the past several years has been the more widespread adoption of a line-search technique coupled with pseudo-transient continuation (PTC) for Newton-Krylov methods. Although line search techniques have been well known for optimization methods,[86] their use for PTC Newton-Krylov solvers in aerodynamic CFD problems appears to have been implemented first in the Boeing GGNS code[37] and later refined by Ceze et al.[87] and others.[46] The general idea is to choose a value of $0 \leq \alpha \leq 1$ in equation (5) which maximizes the decrease in the L_2 norm of the nonlinear unsteady residual $R_{unsteady}(w^{n+1}) = \frac{M(w^{n+1}-w^n)}{\Delta\tau} + R(w^{n+1})$. Provided $\frac{\partial R}{\partial w}$ represents the exact linearization of R, a reduction in the L_2 norm of the unsteady residual is guaranteed for small enough values of $\alpha \; \epsilon \; [0,1]$.[87] The optimal value of alpha is usually determined by evaluating the residual vector $R_{unsteady}(w^n + \alpha\Delta w^{n+1})$ for several values of alpha, possibly using a curve fit to minimize the functional $\|R_{unsteady}(\alpha)\|_2$. The line search is linked with the pseudo-transient continuation algorithm by increasing the pseudo-time step when the full update is taken (i.e. $\alpha = 1$) and decreasing the pseudo-time when $\alpha \ll 1$. This approach has been found to be much more reliable than earlier strategies based on residual norm alone[88] and is one of the key developments that has enabled more widespread use of Newton-Krylov solvers for computational aerodynamics.

However, in spite of the success of line-search based PTC methods, Newton-Krylov solvers still expend a significant portion of their time in the nonlinear continuation phase of the algorithm for more difficult cases and additional techniques for accelerating this phase of the solver are desirable. More recently there has been research in the use of homotopy-based methods for accelerating the continuation phase of Newton-Krylov solvers with some success, particularly for stiff problems.[89, 90] Other methods have been proposed, such as nonlinear preconditioning, where the global nonlinear problem is broken up into smaller nonlinear problems that are solved individually,[91, 92] although little success has been observed with this approach for practical CFD problems in our own experience. In recent work, we have proposed a residual smoothing technique for accelerating nonlinear continuation.[93, 94] In this approach, the right hand side of equation (4) is augmented with a smoothed residual as:

$$\left[\frac{M}{\Delta\tau} + \frac{\partial R}{\partial w}\right] \Delta w^n \quad = \quad -R(w^n) - \frac{M}{\Delta\tau}\left[D\right]^{-1} R(w^n) \tag{9}$$

where the last term on the right hand side consists of an update formed by application of a local nonlinear smoother $[D]^{-1} R(w^n)$ (i.e. nonlinear Jacobi, Gauss-Seidel or line solver) rescaled by the factor $\frac{M}{\Delta\tau}$ for consistency. Figure 11 illustrates the application of this method to the wing-body transonic problem described earlier in Figure 2. The smoothing operator is formed by taking the aggregate solution update resulting from 5 nonlinear line-implicit smoothing steps. Using this simple modification, the Newton-Krylov scheme converges in 44 nonlinear steps compared to 80 for the unsmoothed case. In Figure 11 the convergence histories are plotted in terms of the cumulative Krylov vectors used, which provides a better measure of required computational time. In this case, the addition of residual smoothing reduces the total number of cumulative Krylov vectors employed by a factor of 2, resulting in a corresponding reduction in overall cpu time to solution. Another approach for accelerating the nonlinear continuation phase is to use a nonlinear (FAS) multigrid algorithm, where periodic visits to a coarser grid are performed to compute nonlinear corrections to the Newton-Krylov solver operating on the fine grid. This approach exhibits some similarities to the FUN3D multigrid strategy discussed previously in Figures 9 and reference,[84] although in this case the algorithm is designed to recover an exact Newton solver in the final phases of convergence. This approach has demonstrated increases in nonlinear solution efficiency when used on its own[95] and when used in conjunction with the residual smoothing approach discussed previously as shown in Figure 11(c) taken from reference.[94]

It should be noted that Newton-Krylov methods for computational aerodynamic problems were introduced early on by Wigton,[54] and later as an alternate solver in the NASA FUN3D code, which won the 1999 Gordon Bell prize,[65] but did not become widely adopted as mainstream production solvers until more recently. This is in contrast to their early and extensive use in other application areas particularly within the DoE. Part of the reason for the resurgence of these solvers in computational aerodynamics may be due to the increased availability of exact linearizations, the success of line-search pseudo-transient continuation methods and the increased need for convergence to low residual tolerances for more demanding problems. Krylov methods on their own are useful techniques for accelerating the solution of linear problems, and they are being increasingly adopted for the solution of adjoint problems which arise in the context of design optimization and are linear problems. Additionally, Krylov methods can be used to provide tighter coupling for multidisciplinary problems following the physics-based preconditioning approach originally discussed in references.[96, 97] In this approach, a loose coupling approach where each discipline is solved in turn while lagging the other discipline(s) (i.e. disciplinary Gauss-Seidel) is used as a preconditioner for a Newton-Krylov scheme based on the linearization of the fully coupled multidisciplinary problem. One of the first demonstrations of this approach for aero-structural problems can be found in reference,[98] which has been used extensively in subsequent work.[99] Figure 12 reproduces a comparison taken from reference[100] between the convergence obtained for a loosely coupled approach versus the fully coupled Newton-Krylov strategy for a steady-state aerostructural problem based on the CRM, which requires the coupled solution of the flow equations, mesh deformation equations and structural model equations. In this particular case, the fully coupled approach achieves close to a factor of 2 in overall reduction in solution cost as shown from the convergence histories of the individual disciplinary residuals. The reference also shows how the benefits of the fully coupled approach increase for more flexible structures or higher dynamic pressures. In addition to the analysis problem, this same approach can be used for solution of the corresponding coupled adjoint problem. General strategies



(a)

(b)

(c)

Figure 11. Convergence of (a) baseline Newton-Krylov method and (b) Newton-Krylov method with residual smoothing and (b) Newton-Krylov with residual smoothing and FAS multigrid continuation for F6 wind-body test case in terms of cumulative Krylov vectors. Reproduced from[94]

for fully coupled multidisciplinary Newton-Krylov solvers have been implemented in the NASA OpenMDAO framework.[101]

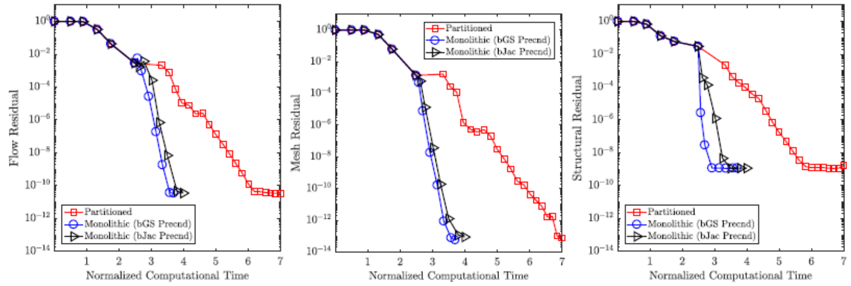


Figure 12. Convergence of aero-structural analysis problem using Newton-Krylov approach applied to fully coupled problem versus loosely coupled approach for CRM aerostructural test case. Reproduced from[100]

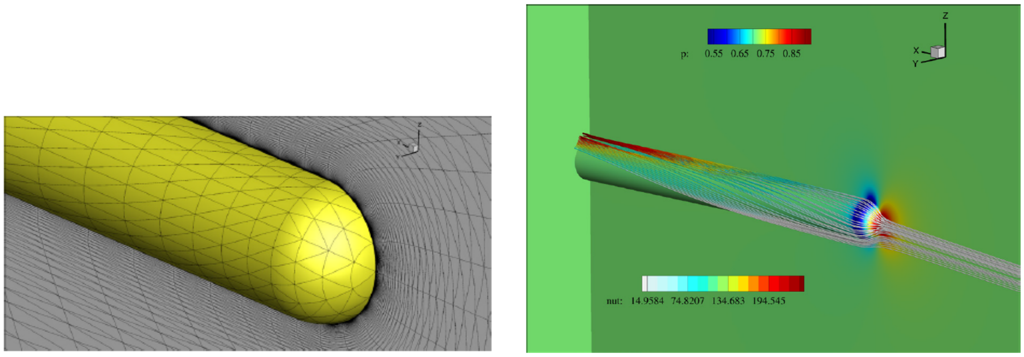
Finally, preconditioned Newton-Krylov methods for time-spectral problems have been demonstrated where the Newton-Krylov scheme is applied to the linearization of the full space-time system of equations for periodic problems in time, showing increases in computational efficiency of the order of 2 over the standard approximate factorization scheme.[102–104]

B. Higher-Order RANS Methods

In this section we consider $p \geq 1$ DG discretizations and $p \geq 2$ SUPG discretizations. As mentioned previously, p=1 DG discretizations are included herein due to the fact that they incur a larger number of degrees of freedom on the same mesh compared to second-order accurate FV schemes, and require the use of curved mesh elements to fully capture the full accuracy of the underlying formulation.

On a given computational mesh, the accuracy gains observed in raising the order of accuracy of a finite-element (FEM) discretization (DG or SUPG) are much larger than those obtained for high-order finite-difference discretizations. This is due to the fact that the number of degrees of freedom increases for the FEM discretizations as the p-order is raised on the same mesh. By contrast, finite-difference discretizations employ the same degrees of freedom but achieve higher-order accuracy through increased stencil size. In addition to these accuracy advantages, the compact stencils at higher order, particularly for DG discretizations, are seen as an advantage for obtaining exact linearizations for use in Newton-Krylov solvers, for the formulation of adjoint methods in design optimization, as well as for facilitating adaptive mesh refinement and overset mesh techniques which require stencil modifications at inter-grid boundaries.[105,106]

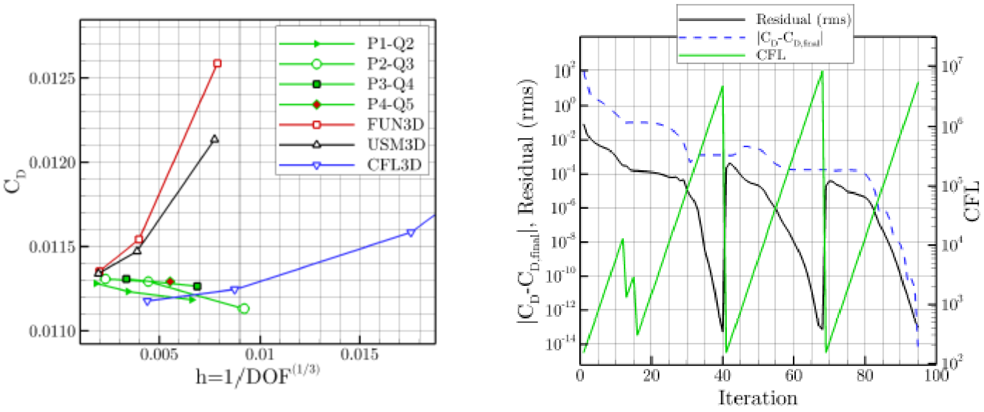
Evidence from the High-Order CFD Workshop (HiOCFD) series has shown that higher-order accurate DG and SUPG discretizations achieve superior accuracy with lower numbers of degrees of freedom compared to their lower order counterparts. These benefits are most apparent for simple problems with smooth solutions. For example, Figures 13 and 14 illustrates the solutions for a DG discretization at various orders of accuracy up to fourth-order (p=3) for the hemispherical cylinder flow problem defined in references,[31,32] using suitably curved mesh elements. In this case, the most accurate solution overall is obtained at p=3 on a grid of approximately 446,000 cells (74K prisms and 372K hexahedra) using fewer degrees of freedom (26.8 million at p=3) than most of the lower order solutions on finer meshes shown for comparison in Figure 14(a). Furthermore, the steady-state solution at p=3 for this case was obtained using a Newton-Krylov method with ILU(0) preconditioning, which was easily converged using p continuation with 25 nonlinear steps at the final p=3 discretization, as shown in Figure 14(b).



(a) (b)

Figure 13. Illustration of hemispherical cylinder test case and solution at Mach=0.6, Incidence=5^o, Re=0.35million

For more complex industrial cases, high-order discretizations yield significant accuracy benefits but generally result in very stiff systems of equations which can be difficult and costly to converge to steady-state. This is well illustrated by the more difficult test cases attempted in the latest HiOCFD5 workshop. Figure 15 depicts a sample mesh and p=2 SUPG solution computed for the CR1 test case in HiOCFD5. The configuration consists of the CRM geometry from DPW5 with two flow conditions examined: Mach=0.3 and Mach=0.85, with an flow incidence of 2.75^o and Reynolds number of 5 million for both cases. Solutions were provided by five contributors using either SUPG discretizations at p=1 (second) and p=2 (third) orders of accuracy, or DG discretizations up to p=3 (fourth-order) accuracy. A sequence of progressively finer fully tetrahedral meshes was generated for the workshop with quadratically (q=2) curved elements, which is suitable for p=2 SUPG and p=1 DG discretizations (noting that p-order DG discretizations require q=p+1 curvature to achieve their full accuracy benefit). While one of the p=1 SUPG contributions was performed on the workshop supplied meshes, the two other SUPG contributions were run on fully tetrahedral meshes with resolutions ranging from 180,000 points to 1.8 million points, which are much coarser than the workshop



(a)

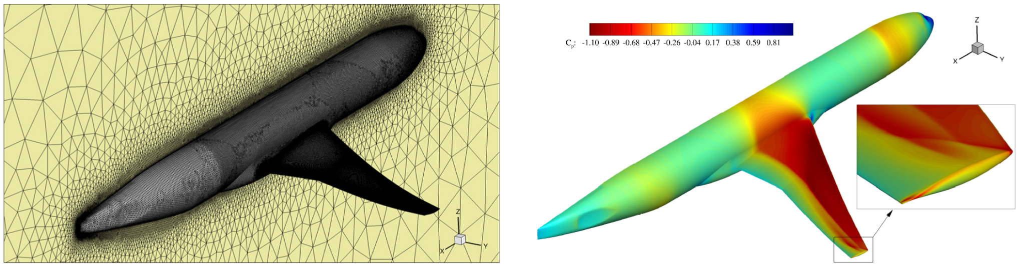
(b)

Figure 14. (a) Comparison of computed drag coefficient for various discretizations as a function of the number of degrees of freedom and (b) convergence of DG discretization at orders p=1, p=2 and p=3 for hemispherical cylinder test case.

supplied meshes. This is evident in the plots of drag convergence in Figure 16 for both cases, where the two sets of p=1 SUPG results (i.e. legend HOMA and COFFE) lie to the right of the collective DPW workshop results, indicating the use of much coarser grids. However, the p=2 SUPG results on these same meshes are seen to lie well within the asymptotic drag values obtained on finer grids using finite-volume methods. Furthermore, the variation of drag values with grid size at p=2 is much lower than at p=1, with even the coarsest mesh giving relatively accurate drag values. Similar conclusions can be drawn for the DG results (DG3D and SANS labels), although these would benefit further at p=2 and above from more accurate mesh curving. Figure 17(a) depicts the convergence histories for the DG3D code for the Mach=0.3 case at p=1, p=2 and p=3, while Figure 17(b) shows the convergence history at Mach=0.85 for the HOMA SUPG code at p=1 and p=2. For both codes, a PTC-Newton-Krylov method was used, with ILU(0) preconditioner for the DG3D code, and the dual-CFL line-preconditioner with 200 line sweeps per Krylov vector for the HOMA code. In both cases, convergence to low residual levels was achieved for the higher p orders, with the SUPG results being obtained with fewer nonlinear iterations overall. For the transonic case, convergence of the DG3D code was problematic and required restarting from lower Mach number solutions to achieve full convergence.

Moving on to more complex configurations, the DLR-F11 high-lift geometry, which was the subject of the HLPW2 workshop, was chosen as a challenging case for high-order methods in HiOCFD4. Figure 18 illustrates a p=2 DG solution obtained on this configuration by the DLR-PAGE code[67] and the associated convergence history. In this case the mesh consists of approximately 3.5 million cells, with cubic (q=3) curved elements near the wall. The resulting p=1 and p=2 solutions contain approximately 14 million and 35 million degrees of freedom, respectively. The solution scheme consists of a PTC-Newton-Krylov method with ILU(0) preconditioning, similar to that employed in the previous case by the DG3D code. The figure shows rapid initial convergence at p=0 (first-order accuracy), followed by somewhat slower p=1 convergence, and final convergence at p=2 to a residual level of 1.e-10 achieved in approximately 250 nonlinear cycles. These results are significant as they represent one of the first demonstrations of higher-order DG discetizations on a relatively difficult aerodynamic CFD problem of industrial interest.

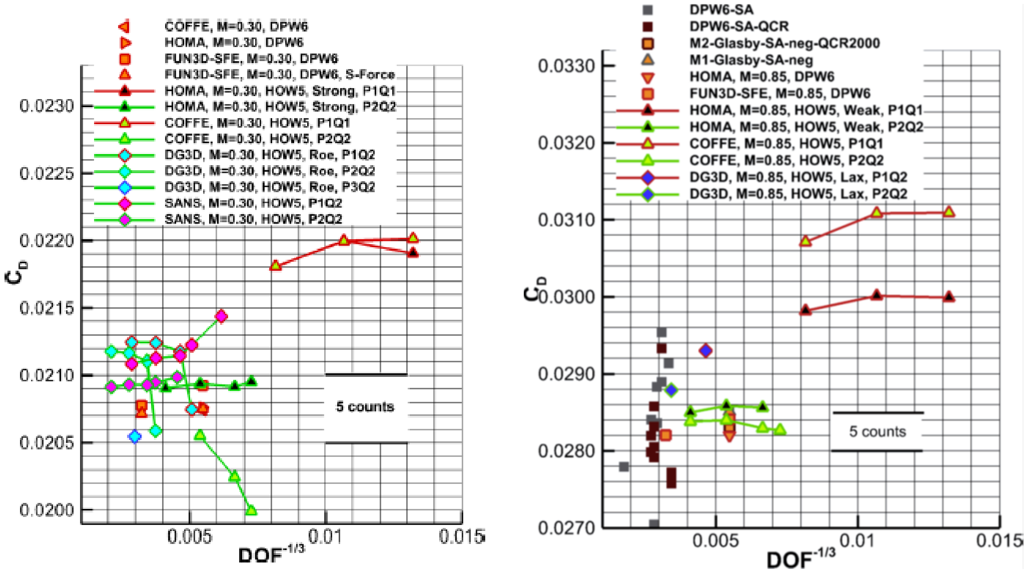
In the HiOCFD5 workshop, a follow-up high-lift test case denoted ($MC1-0$) was proposed based on the HL-CRM geometry taken from HLPW3. Various unstructured meshes were generated with quadratic (p=2) element curving. Figure 19 illustrates a fully tetrahedral mesh of 5.5 million cells and 933,000 points and the corresponding p=2 SUPG solution (which results in 7.4 million degrees of freedom) obtained on this mesh using the HOMA code.[72] A total of four participants submitted results, consisting of two SUPG contributions and two DG contributions at orders p=1 and p=2. In terms of accuracy the results mirrored those reported previously for the CR1 (based on the CRM configuration) test case, with the p=2 results showing dramatically increased accuracy on the same grids compared to the p=1 solutions. Figures 20 and 21 focus on the solution procedure and convergence results for the HOMA SUPG code at p=1 and p=2 for this case, which we believe to be representative of a competitive solution strategy. The convergence history


Z
Y
X
C_p: -1.10 -0.89 -0.68 -0.47 -0.26 -0.04 0.17 0.38 0.59 0.81
Z
X
Y

(a)

(b)

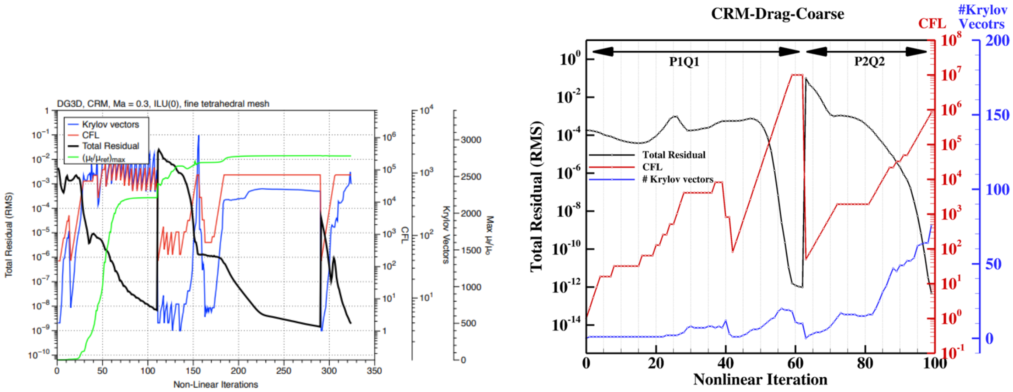
Figure 15. Illustration of tetrahedral mesh for CRM configuration and p=2 SUPG solution for CR1 test case from HiOCFD5



(a)

(b)

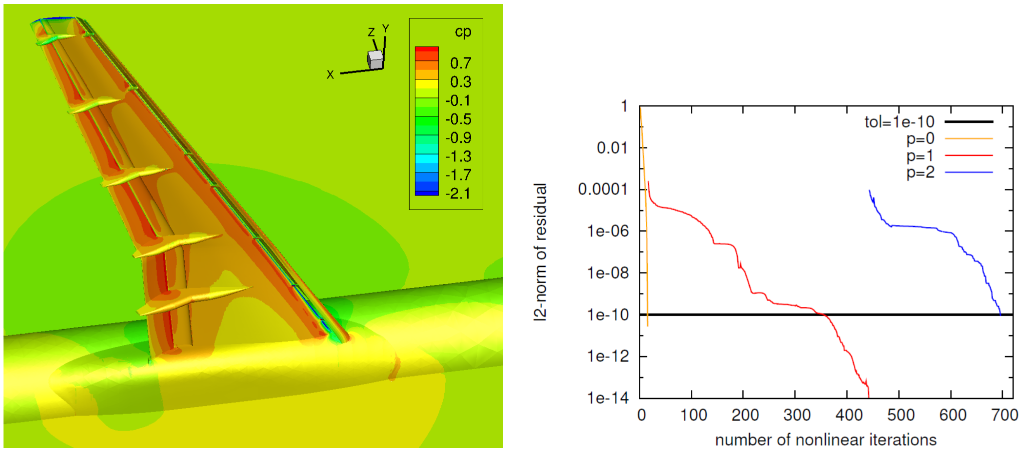
Figure 16. Illustration of computed drag coefficient as a function of grid resolution for (a) M=0.3 and (b) M=0.85 CR1 test case from HiOCFD5



(a)

(b)

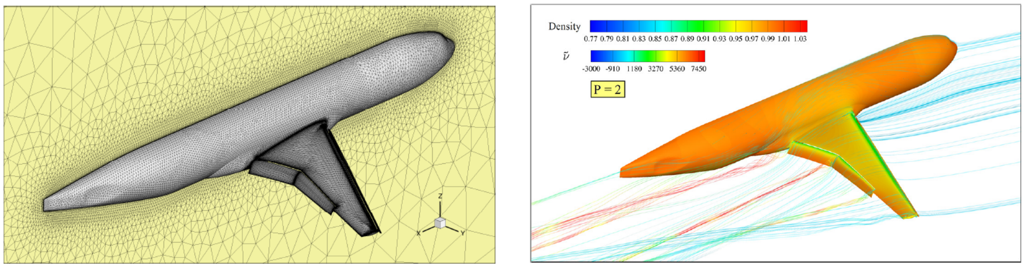
Figure 17. Convergence history of DG3D code at p=1, p=2 and p=3 for Mach=0.3 test case on coarse mesh and (b) Convergence of SUPG HOMA code at p=1 and p=2 for Mach=0.85 test case on coarse mesh.



(a) (b)

Figure 18. (a)DG solution obtained on DLRF11 high-lift configuration using DLR PAGE code and (b) Convergence history for p=0, p=1 and p=2 solutions in terms of nonlinear iterations using an ILU(0) preconditioned GMRES solver. Reproduced from[67]

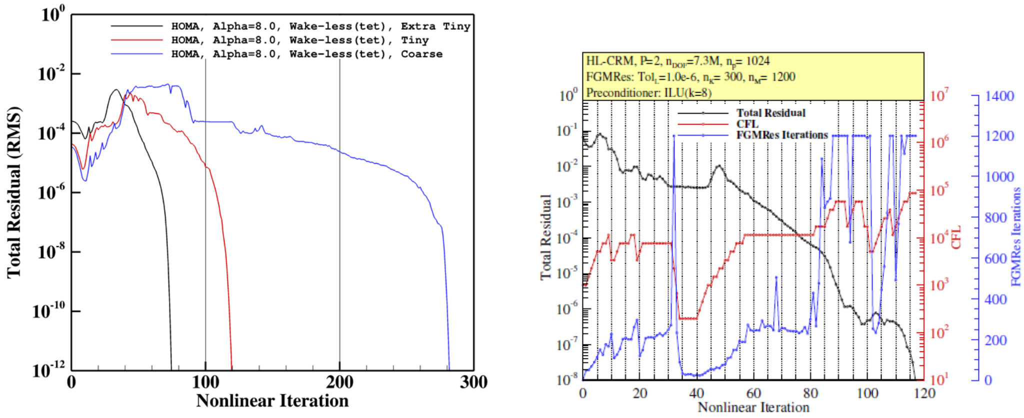
for the p=1 discretization on three meshes of increasing resolution is shown in Figure 20(a) where the total number of nonlinear iterations to achieve machine zero convergence is seen to increase substantially with finer mesh resolution. This highlights the need for improved solution strategies that are more optimal and capable of delivering grid independent convergence rates. Subsequent work in reference[72] has shown benefits using a residual smoothing technique to reduce this grid dependence of the convergence history, although further work is required to be able to consistently guarantee more optimal convergence rates. The convergence history for the p=2 discretization is shown in Figure 20(b), illustrating the difficulties in converging this discretization on complex cases. In this case, an ILU(k) with eight levels of fill (k=8) was required to obtain a convergent algorithm. In an attempt to reduce the overall cost of the solution, the block ILU smoother previously described was applied to this case. Using smaller block sizes of approximately 2,000 mesh points with ILU(k) with two levels of fill (k=2), and 200 smoothing sweeps per Krylov vector, convergence in 80 nonlinear steps was achieved a shown in Figure 21(a). Overall, this approach resulted in a reduction of cpu time to solution by approximately 60%, as shown in Figure 21(b).



(a) (b)

Figure 19. Illustration of tetrahedral mesh and p=2 SUPG solution for HL-CRM test case (MC1-0) from HiOCFD5

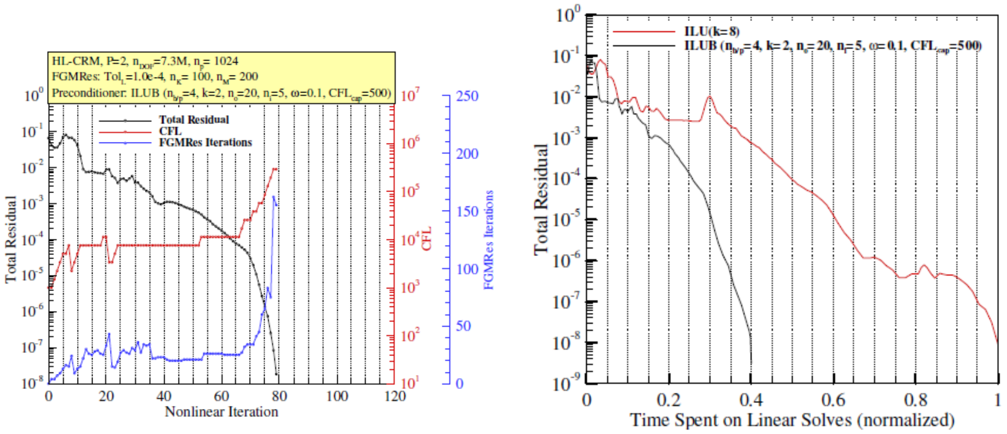
These results serve to demonstrate the potential advanced solution techniques can have on making these higher-order discretizations more competitive and thus enabling the superior accuracy properties inherent in these formulations. However, it should be noted that, in all the results presented in this section, the convergence to steady-state of these high-order accurate discretizations comes at a computational expense that is significantly higher than what can be achieved with current lower-order production codes on finer meshes. Partly for this reason, it appears that the focus in higher-order methods for industrial RANS problems has become more centered on capitalizing on the accuracy gains obtained in going from p=1 to p=2, particularly for SUPG discretizations. Indeed, in this lower p range, SUPG discretizations can be



(a)

(b)

Figure 20. (a) Convergence histories for p=1 SUPG HOMA code on meshes of increasing resolution and (b) p=2 SUPG convergence history (restarted from p=1 solution) on coarsest mesh using ILU(8) preconditioner



(a)

(b)

Figure 21. (a) Improved convergence history for p=2 SUPG solution using iterative ILU(2) on graph-based blocks and (b) comparison of this solver with ILU(8) preconditioner in terms of wall-clock time

expected to be more efficient simply due to the lower number of degrees of freedom compared to equivalent DG discretizations.[44] Earlier goals of developing p=3 and higher-order accurate discretizations for RANS problems are now seen as aspirational by many practitioners, as the difficulties in solving the discrete RANS equations resulting from these higher-order discretizations has become more evident. Nevertheless, even at p=2, SUPG and DG methods offer impressive accuracy gains on relatively coarse meshes which could transform RANS CFD codes if these can be made competitive in terms of cpu time to solution for a given accuracy level. Additionally, the availability of curved element meshes will be key to realizing the full potential of these discretizations. Naturally, these efforts must be weighed against the cost of solving traditional finite-volume methods on significantly finer meshes, which may entail their own difficulties given the less than optimal scaling of current finite-volume RANS solvers as discussed in the previous section.

Finally, the success of higher p discretizations on simpler problems with smooth solutions, such as the hemisphere flow results presented above in Figure 14, provide evidence that higher p discretizations are indeed more effective at delivering high accuracy in regions away from singularities. Therefore, for optimal accuracy and efficiency, combined adaptive h-p refinement will need to be considered. Although h-p refinement methods have been demonstrated for simple problems, their use for complex industrial problems such as high-lift RANS problems remains to be demonstrated.

C. Unsteady RANS Methods

Discretization and solution techniques for unsteady RANS problems are not discussed in detail in this paper. In general, the spatial discretizations used in steady-state problems translate directly to unsteady problems, and second-order BDF2 temporal discretizations are most often used. However, higher order temporal discretizations have been used increasingly, including optimized versions of BDF,[107] diagonally implicit Runge-Kutta schemes,[108] and more recently fully-implicit Runge-Kutta schemes.[109] Furthermore, harmonic balance or time-spectral methods have gained wide acceptance as temporal discretizations for problems which are periodic in time[110, 111] and extensions of these methods to quasi-periodic problems have been proposed.[112] Solution techniques for time-implicit unsteady RANS problems generally carry over from the steady-state case. In fact the solution of an individual implicit time step in an unsteady RANS problem is most often less demanding than the solution of the corresponding steady-state problem due to the reduction in stiffness afforded by the temporal terms in the time-implicit Jacobian matrix, which increases the diagonal dominance of the matrix as the time step is reduced. Although partial convergence of the residuals in time-dependent problems has been a common practice due to computational cost, convergence to deeper levels at each implicit time step is becoming more important for problems involving long time integration, since local errors at each time step accumulate over time. Furthermore, convergence to low tolerances is required for time-dependent adjoint problems in order to obtain accurate sensitivities and/or error estimates.[113] These growing requirements make the use of stronger solvers including Newton-Krylov methods more desirable for unsteady RANS problems. Newton methods in particular are well suited for these purposes because one of their main drawbacks, notably the slow nonlinear continuation phase, is often avoided in time-dependent problems since a good initial solution is obtained from the state at the previous time step. Advances in techniques for fully-implicit Runge-Kutta methods and time-spectral methods also enable the possibility of using time parallelism to accelerate the overall wall-clock time to solution.[104, 114]

IV. Scale-Resolving Methods

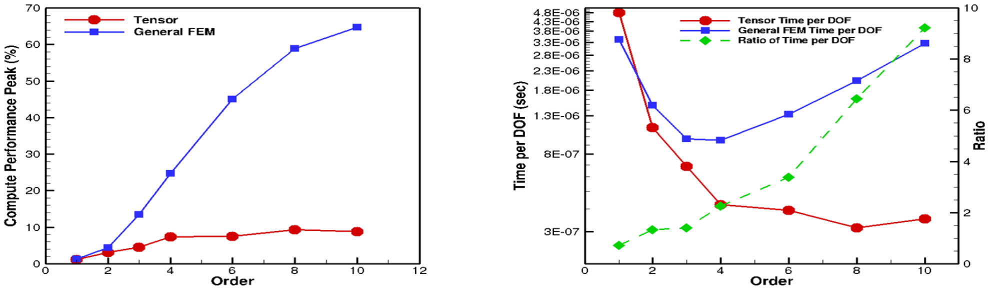
The requirements for the construction of effective scale-resolving methods for computational aerodynamics can be considerably different compared to the requirements of steady-state and unsteady RANS methods. Here scale-resolving methods refer principally to Large Eddy Simulation (LES) methods, either in stand-alone mode or as a component of a hybrid RANS-LES method such as Detached Eddy Simulation (DES)[14] or overset mesh RANS-LES methods.[106, 115] For scale-resolving methods, the spatial scales of interest (resolved turbulence eddies) are frequently of the same order as the temporal scales of interest, and explicit schemes are often employed. In this case, the solution techniques become trivial (residual evaluation) although particular considerations must be taken into account for the construction of suitable discretizations. The spatial discretization must incur low levels of numerical dissipation in order to avoid dissipating eddies prematurely at resolved scales. As always, high spatial accuracy remains an important objective, but this may be achieved either by using second-order accurate discretizations on relatively fine meshes or higher-order accurate discretizations on coarser meshes.

Lattice Boltzmann methods (LBM)[116] fall into the first category. These methods are based on kinetic theory and correspond to a discretization of the Boltzmann equation, which has been shown to converge to the Navier-Stokes equations in the continuous limit. LBM methods are explicit methods in time, which are second-order accurate in space and first-order accurate in time with particularly low dissipative properties and a demonstrated advantage for modeling accurate convection of vorticity.[117] LBM methods have shown promise for predicting off-design conditions for aerodynamic problems such as $C_{L_{max}}$ for high-lift configurations in the recent HLPW workshops.[21, 22] Similarly, second-order accurate low dissipation finite-volume schemes operating on structured or unstructured meshes have been used extensively as scale-resolving methods over the last decade.[118, 119] In both instances, the principal advances in these methods today revolve around physical modeling, for example for near-wall regions[120] and subgrid scale models,[121] which remain beyond the scope of this paper.

The use of higher-order accurate discretizations for scale-resolving methods dates back several decades with the advent of spectral methods and their extension to spectral element methods, principally for incompressible flows.[122] More recently, high-order DG methods have been used increasingly for scale resolved compressible flow problems. Numerical experiments on canonical test problems such as the Taylor-Green vortex (TV) problem[123] have demonstrated the advantages of using very high-order accurate discretizations for DNS and LES problems, although as the summary from the first HiOCFD workshop points out, presently this can only be said for spatial error using explicit time-stepping.[124]

Numerous high-order discretizations have been proposed for use as scale-resolving schemes, including spectral difference, spectral volume,[125] discontinuous Galerkin,[126] residual distribution[127] and flux reconstruction methods.[128] Many of these schemes are related and can be cast as special cases within a general flux reconstruction framework,[129] including generalized residual distribution schemes.[130] In many cases, these insights can be used to construct total kinetic energy or entropy preserving schemes with nonlinear stability properties.[131] A principal advantage of these discretizations is that they result in dense computational kernels that are well suited for emerging HPC architectures including GPUs. This has led to the construction of highly optimized software libraries designed to enable efficient execution of high-order scale-resolving applications on heterogeneous HPC architectures.[132–134]

On quadrilateral elements in two dimensions and hexahedral elements in three dimensions, a tensor-product formulation may be used to drastically reduce the cost of assembling the residual for these high-order discretizations[135] (and extensions have been proposed for other element types[136]). In this approach, the computational complexity of the basic operations, such as evaluating solution values at quadrature points within a cell, is reduced from $O((p+1)^{2d})$ to $O((p+1)^{d+1})$ where p+1 is the order of accuracy of the discretization and d represents the number of spatial dimensions (d=3 in 3D). As an example, Figure 22 depicts the computational rates achieved as a function of p-order accuracy of a residual evaluation for a DG discretization of the Navier-Stokes equations on Cartesian mesh hexahedral elements.[137] The first implementation consists of a standard FEM approach, which can be formulated as a matrix-matrix multiplication operation, which in turn has been implemented using the BLAS Level 3 optimized numerical libraries. This implementation is seen to achieve increasingly higher computational rates as the p-order is increased up to p=10. The second implementation uses the aforementioned tensor-product formulation, which is seen to achieve lower (although still respectable) computational rates, which also increase moderately with p-order. However, the tensor-product formulation is seen to provide close to an order of magnitude decrease in wall-clock time required per degree of freedom at high p orders due to the lower computational complexity of this approach.



(a)

(b)

Figure 22. (a)Illustration of computational rates achieved for standard finite-element DG discretization versus tensor-product formulation as a function of p-order of accuracy and (b) Corresponding time per degree of freedom as a function of p-order of accuracy.

The high accuracy and lower cost (per degree of freedom) of tensor-product formulations makes them ideal candidates for use as scale-resolving methods. However, the additional issues of numerical stability and explicit time step restrictions must also be considered. The use of very high-order discretizations often encounters problems of numerical stability due to aliasing errors, which arise from the inability to integrate exactly the flux functions present in the Euler and Navier-Stokes equations on mesh elements and faces. Thus, most approaches resort to over-integration and/or filtering techniques for preventing numerical instabilities, with varying degrees of success. An alternate approach consists of developing high-order discretizations with nonlinear stability properties such as total kinetic energy (TKE) preserving[126, 138] or entropy stable schemes.[131, 139–141] These discretizations incur very low levels of numerical dissipation making them ideally suited for use as scale-resolving methods. Indeed, DG discretizations using standard numerical flux functions are known to be overly dissipative for the smallest resolved scales even at high order, which may account for the common practice of performing implicit LES (ILES) simulations with DG, where no subgrid-scale model is used in order to prevent excessive dissipation at the smallest resolved scales. By contrast, the use of TKE preserving or entropy stable discretizations at high order offers the possibility of more carefully controlling the dissipation of the smallest resolved scales using a physically derived subgrid-scale model rather than through excessive numerical dissipation.[138]

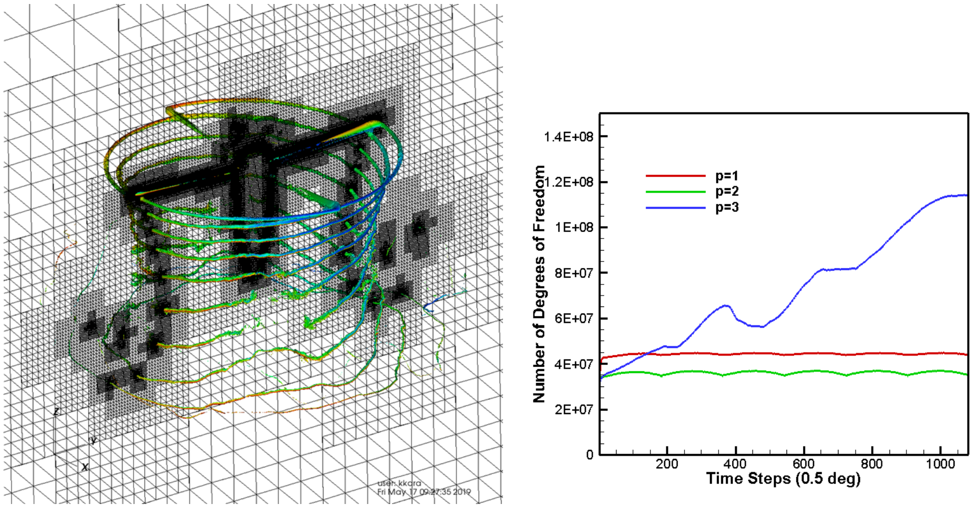
While many DG-based LES methods rely exclusively on explicit time-stepping procedures, there is a growing interest in the use of time-implicit methods for such problems. Taking an implicit approach in the temporal direction may be driven by excessively small explicit time step restrictions, which may arise from low Mach number acoustic limits or viscous diffusion limits, restrictions due to very high p-spatial order, widely varying grid spacing, possibly due to the use of adaptive mesh refinement (AMR) and higher order temporal discretizations. Higher-order time discretizations used in conjunction with high-order spatial discretizations have been proposed using diagonally and fully-implicit Runge-Kutta methods[114] and space-time discretizations using an analogous discontinuous Galerkin discretization in time.[141] In these examples, the requirement of solving simultaneously for multiple quadrature points in the time direction leads to an additional degree of parallelism in time which may be exploited by the temporal solver.

One of the principal issues in formulating an efficient implicit solver for time-implicit high-order DG methods is that the evaluation and approximate inversion of the local element Jacobian using a straightforward approach becomes impractical at high p-orders due to the size of the resulting dense block matrices. A practical approach to circumvent this difficulty has been proposed in references,[142,143] where a preconditioned Newton-Krylov method is proposed using a tensor-product based preconditioner for space-time DG discretizations at high order. As an alternate strategy, a p-multigrid approach may be invoked, using tensor-product operations at various p-orders within a linear multigrid strategy. p-multigrid methods were used successfully for Euler and RANS-based DG discretizations, but have fallen out of favor more recently due to robustness issues for more difficult steady-state RANS problems. These methods may prove to be better suited for scale-resolving simulations, given the closer alignment between temporal and spatial scales in such cases.

Finally, there exist several examples of research or production codes that combine scale-resolving methods with adaptive mesh refinement (AMR) methods.[106,115] Since scale-resolving methods are inherently time-dependent, these necessarily involve dynamic AMR procedures for which various library packages are available or under development.[144–146] Figure 23 illustrates an overset mesh simulation using a high-order DG discretization in the off-body region for the S76 rotor using an adaptive h-p refinement strategy in the wake using the methodology described in references.[106,137] In this case, the h-p refinement is driven simply by a prescription which causes full refinement (to a predetermined h and/or p level) when Q-criterion exceeding a user set threshold level is detected. In general, the strategy produces high p-order cells in wake regions away from the body, and smaller cells with lower p-order in regions where the off-body mesh overlaps with the near body mesh, since resolution and p-order matching are prescribed in these regions. Figure 23(b) illustrates the number of degrees of freedom contained in the cells according to their p-order over the first 1.5 revolutions of the rotor using a 0.5 degree time step. As the wake develops, the number p=3 cells grows while the number of lower p-order cells remains close to constant, since these represent the near overlap regions with the moving near-body mesh. Experience has shown that using higher-order p cells results in equivalent or higher accuracy results at lower cost, compared to the use of lower p-order cells. For example, in this case, after 7 rotor revolutions the p=1,2,3 simulation resulted in 210 million degrees of freedom and required 35.9 seconds per time step, while the analogous p=1,2 simulation resulted in 193 million degrees of freedom and required 53 seconds per time step. Naturally, this is more of a qualitative rather than quantitative assessment of accuracy and efficiency, but nevertheless demonstrates the potential of high-order methods in complex time-dependent simulations. Ultimately, formal error estimators must be developed for both h and p refinement, in both the spatial and temporal directions for scale resolved simulations in order to drive the adaptive process in an optimal manner.

V. Conclusions and Future Work

Looking back over the last five to six years, it becomes evident that significant progress has been made in CFD discretizations and solvers. The fact that advanced discretizations which extend to higher-order accuracy can be used regularly to solve problems of industrial interest including transonic cruise and high-lift RANS cases represents a significant development compared to a decade ago when most of these methods were being exercised for simple two-dimensional problems. At the same time, a clearer picture of how these methods can be used as scale-resolving techniques is also coming into focus. Much of this progress



(a)

(b)

Figure 23. (a)Illustration of rotor wake and adaptive mesh and (b) time history of second (p=1), third (p=2) and fourth-order accurate (p=3) cells in off-body rotor wake mesh for first 1.5 revolutions of S76 rotor.

is due to community efforts which have been instrumental in establishing important baselines, including reference solutions, and building resources to enable verification and validation of existing and emerging CFD technologies. Nevertheless, many of the discretizations and solution techniques discussed in this paper are still not sufficiently mature in terms of efficiency and robustness to displace existing production tools. Rather, a gradual adoption of particular capabilities, possibly through inclusion into existing workflows can be expected to occur over the next several years. Finally, in order to meet the milestones as set out in the CFD2030 roadmap, and to successfully tackle the proposed Grand Challenge problems, a concerted effort will be required not only from CFD algorithms, but from the other important contributing technologies such as geometry, mesh generation and adaptivity, physical modeling and HPC. Promoting these types of interactions and coordination between these diverse communities will be important for achieving the envisioned capabilities and will be a central focus of the CFD2030 Integration Committee.

VI. Acknowledgments

This work was partially funded by NASA grant NNX15AU23A under the Tranformational Tools and Technologies (T^3) project. We are grateful for computer time provided by the NCAR-Wyoming Supercomputer Center (NWSC) and by the University of Wyoming Advanced Research Computing Center (ARCC). Special thanks for input and assistance provided by Behzad Ahrabi, Michael Brazell, John Chawner, Boris Diskin, Rob Falgout, Nathan Hariharan, Ralf Hartmann, Kursat Kara, Li-Shi Luo, Joaquim Martins, Jeff Slotnick, Michael Stoellinger, Ray Tuminaro and David Zingg.

References

[1] 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, NF1676L-18332.

[2] "FUTURE CFD TECHNOLOGIES WORKSHOP," https://scientific-sims.com/cfdlab/WORKSHOP/workshop.html.

[3] "Computational Science: Ensuring America's Competitiveness," *Report to the President, President's Information Technology Advisory Committee*, June 2005.

[4] "A Science-Based Case for Large-Scale Simulation, Volume I," *Office of Science, US Department of Energy*, July 2003, 76pp.

[5] Mavriplis, D. J., Darmofal, D., Keyes, D., and Turner, M., "Petaflops Opportunities for the NASA Fundamental Aeronautics Program," AIAA-Paper 2007-4048, Invited paper at the AIAA Computational Fluid Dynamics Meeting, Miami FL.

[6] Malik, M. R. and (eds.), D. B., "Role of Computational Fluid Dynamics and Wind Tunnels in Aeronautics R & D," NASA/TP-2012-217602.

[7]Levy, D. W., Zickuhr, T., Vassberg, J., Agrawal, S., Wahls, R. A., Pirzadeh, S., and Hemsch, M. J., "Data Summary from the First AIAA Computational Fluid Dynamics Drag Prediction Workshop," *Journal of Aircraft*, Vol. 40, No. 5, 2003, pp. 875–882.

[8]Vassberg, J. C., Tinoco, E. N., Mani, M., Brodersen, O. P., Eisfeld, B., Wahls, R. A., Morrison, J. H., Zickuhr, T., Laflin, K. R., and Mavriplis, D. J., "Abridged Summary of the Third AIAA Computational Fluid Dynamics Drag Prediction Workshop," *Journal of Aircraft*, Vol. 45, No. 3, 2008, pp. 781–798.

[9]Vassberg, J. C., Tinoco, E. N., Mani, M., Rider, B., Zickuhr, T., Levy, D., Brodersen, O. P., Eisfeld, B., iS. Crippa, Wahls, R. A., Morrison, J. H., Mavriplis, D. J., and Murayama, M., "Summary of the Fourth AIAA CFD Drag Prediction Workshop," *Journal of Aircraft*, Vol. 51, No. 4, June 2014, doi.org/10.2514/1.C032418.

[10]Levy, D. W., Laflin, K. R., Vassberg, J. C., Tinoco, E., Mani, M., Rider, B., Brodersen, O., Crippa, S., Rumsey, C. L., Wahls, R. A., Morrison, J., Mavriplis, D. J., and Murayama, M., "Summary of Data from the Fifth AIAA CFD Drag Prediction Workshop," *Journal of Aircraft*, Vol. 51, No. 4, 2014, doi.org/10.2514/1.C032389.

[11]Tinoco, E. N., Brodersen, O. P., Keye, S., Laflin, K. R., Feltrop, E., Vassberg, J. C., Mani, M., Rider, B., Wahls, R. A., Morrison, J., Hue, D., Roy, C. J., Mavriplis, D. J., and Murayama, M., "Summary Data from the Sixth AIAA CFD Drag Prediction Workshop: CRM Cases," *Journal of Aircraft*, Vol. 55, No. 4, 2018, doi.org/10.2514/1.C034409.

[12]Rumsey, C. L., Slotnick, J. P., and Sclafani, A. J., "Overview and Summary of the Third AIAA High Lift Prediction Workshop," *Journal of Aircraft*, Vol. 56, No. 2, 2019, doi.org/10.2514/1.C034940.

[13]Slotnick, J. and Heller, G., "Emerging Opportunities for Predictive CFD for Off-Design Commercial Airplane Flight Characteristics," *Proceedings of the 54th 3AF International Conference, Paris, France*, 2019.

[14]Spalart, P. R., Deck, S., Shur, M. L., Squires, K. D., M.Strelets, and Travin, A., "A New Version of Detached-eddy Simulation Resistant to Ambiguous Grid Densities," *Theoretical and Computational Fluid Dynamics*, Vol. 20, No. 3, 2006, pp. 181–195, DOI: 10.1007/s00162-006-0015-0.

[15]Roy, C. J., Rumsey, C. L., and Tinoco, E. N., "Summary Data from the Sixth AIAA Computational Fluid Dynamics Drag Prediction Workshop: Code Verification," *Journal of Aircraft*, Vol. 55, No. 4, 2018, doi.org/10.2514/1.C034856.

[16]Keye, S. and Mavriplis, D., "Summary of Case 5 from Sixth Drag Prediction Workshop: Coupled Aerostructural Simulation," *Journal of Aircraft*, Vol. 55, No. 4, 2018, doi.org/10.2514/1.C034427.

[17]Vassberg, J. C., DeHaan, M. A., Rivers, S. M., and Wahls, R. A., "Development of a Common Research Model for Applied CFD Validation Studies," AIAA Paper 2008-6919.

[18]Lacy, D. S. and Sclafani, A. J., "Development of the High Lift Common Research Model (HL-CRM): A Representative High Lift Configuration for Transonic Transports," *AIAA Paper 2016-0308, 54th AIAA Aerospace Sciences Meeting, San Diego, CA*, Jan. 2016.

[19]Brooks, T. R., Kenway, G. K., and Martins, J., "Undeflected Common Research Model (uCRM): An Aerostructural Model for the Study of High Aspect Ratio Transport Aircraft Wings," *AIAA Paper 2017-4456, 35th AIAA Applied Aerodynamics Conference, Denver, CO*, June 2017.

[20]Vassberg, J. C., "Challenges and Accomplishments of the AIAA CFD Drag Prediction Workshop Series," *AVT-246 Progress and Challenges in Validation Testing for Computational Fluid Dynamics, Avila, Spain, 26-28 September 2016*, 2016.

[21]Konig, B., Fares, E., Murayama, M., and Ito, Y., "PowerFLOW Simulations for the Third AIAA High-Lift Prediction Workshop," *AIAA Paper 2018-1255*, Jan. 2018, doi.org/10.2514/6.2018-1255.

[22]Trapani, G., Brionnaud, R., and Holman, D., "XFlow Contribution to the Third High-Lift Prediction Workshop," *AIAA Paper 2018-2847*, June 2018, doi.org/10.2514/6.2018-2847.

[23]"AIAA Rotorcraft Hover Prediction Workshop," https://aiaahover.wixsite.com/website-6.

[24]Jain, R., "Hover Predictions on the S-76 Rotor with Tip Shape Variation Using Helios," *Journal of Aircraft*, Vol. 55, No. 1, Jan. 2018, doi.org/10.2514/1.C034075.

[25]Hariharan, N. S., Narducci, R. P., and Egolf, T. A., "AIAA Standardized Hover Simulation: Hover Performance Prediction Status and Outstanding Issues," *AIAA Paper 2017-1429*, Jan. 2017, doi.org/10.2514/6.2017-1429.

[26]Heeg, J., Chwalowski, P., Raveh, D., Jirasek, A., and Dalenbring, M., "Overview and Data Comparisons from the 2nd Aeroelastic Prediction Workshop," Jan. 2016, AIAA Paper 2016-3121.

[27]Heeg, J., Chwalowski, P., Florance, J., Wieseman, C., Schuster, D., and Perry III, B., "Overview of the Aeroelastic Prediction Workshop," Jan. 2013, AIAA Paper 2013-783, 51st Aerospace Sciences Meeting, Grapevine, TX.

[28]"AIAA Geometry and Mesh Generation Workshop," http://www.gmgworkshop.com/.

[29]Karman, S. L. and Erwin, J. T., "High Order Meshes for the Geometry and Mesh Generation Workshop I," *AIAA Paper 2018-0661*, Jan. 2018.

[30]"5th International Workshop on High-Order CFD Methods," https://how5.cenaero.be/.

[31]Diskin, B., Thomas, J. L., Rumsey, C. L., and Pandya, M. J., "Reference Solutions for Benchmark Three Dimensional Turbulent Flows," *AIAA Paper 2016-0858*, Jan. 2016, doi.org/10.2514/6.2016-0858.

[32]Rumsey, C., "Turbulence Modeling Resource Website," http://turbmodels.larc.nasa.gov.

[33]Nishikawa, H. and Diskin, B., "Customized Grid Generation Codes for Benchmark Three-Dimensional Flows," *AIAA Paper 2018-1101*, Jan. 2018, doi.org/10.2514/6.2018-1101.

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

[35]"UMich/NASA Symposium on Advances in Turbulence Modeling," http://turbgate.engin.umich.edu/symposium/.

[36]"NASA CFD Prediction Error Assessment Workshop 2018," https://turbmodels.larc.nasa.gov/nasa40percent.html.

[37]Kamensrskiy, D. S., Bussoletti, J. E., Hilmes, C. L., Venkatakrishnan, V., Wigton, L. B., and Johnson, F. T., "Numerical Evidence of Multiple Solutions for the Reynolds-Averaged NavierStokes Equations," *AIAA Journal*, Vol. 52, No. 8, Aug. 2014, doi.org/10.2514/1.J052676.

[38]Glasby, R. S. and Erwin, J. T., "Introduction to COFFE: The Next-Generation HPCMP CREATETM-AV CFD Solver," *AIAA Paper 2016-0567*, Jan. 2016, doi.org/10.2514/6.2016-0567.

[39]Anderson, W. K., Newman, J. C., and Karman, S. L., "Stabilized Finite Elements in FUN3D," *Journal of Aircraft*, 2017, doi.org/10.2514/1.C034482.

[40]Michal, T. R., Babcock, D., Kamenetskiy, D. S., Krakos, J., Mani, M., Glasby, R., Erwin, T., and Stefanski, D., "Comparison of Fixed and Adaptive Unstructured Grid Results for Drag Prediction Workshop 6," *AIAA Paper 2017-0961*, Jan. 2017, doi.org/10.2514/6.2017-0961.

[41]Ahrabi, B., Brazell, M., and Mavriplis, D. J., "An Investigation of Continuous and Discontinuous Finite-Element Discretizations on Benchmark 3D Turbulent Flows," *AIAA Paper 2019-0100*, Jan. 2019, doi.org/10.2514/6.2019-0100.

[42]Galbraith, M. C., Allmaras, S. R., and Darmofal, D. L., "SANS RANS Solutions for 3D Benchmark Configurations," *AIAA Paper 2018-1570*, Jan. 2018, doi.org/10.2514/6.2018-1570.

[43]Balin, R. and Jansen, K. E., "A Comparison of RANS, URANS, and DDES for High Lift Systems from HiLiftPW-3," doi.org/10.2514/6.2018-1254.

[44]Glasby, R., Burgess, N., Anderson, W. K., Wang, L., Allmaras, S., and Mavriplis, D., "Comparison of SUPG and DG Finite-Element Techniques for the Compressible Navier-Stokes Equations on Anisotropic Unstructured Meshes," *AIAA Paper 2013-0691*, Jan. 2013, doi.org/10.2514/6.2013-691.

[45]Ahrabi, B. R., Brazell, M. J., and Mavriplis, D. J., "An Investigation of Continuous and Discontinuous Finite-Element Discretizations on Benchmark 3D Turbulent Flows," *AIAA Paper 2018-1569, Presented at the 57th AIAA Aerospace Sciences Meeting, Kissimmee FL*, Jan. 2018.

[46]Anderson, W. K., Ahrabi, B. R., and Newman, J. C., "Finite Element Solutions for Turbulent Flow over the NACA 0012 Airfoil," *AIAA Journal*, Vol. 54, No. 9, 2016, doi.org/10.2514/1.J054508.

[47]Diskin, B., Anderson, W. K., Pandya, M., Rumsey, C. L., Thomas, J., Liu, Y., and Nishikawa, H., "Grid Convergence for Three Dimensional Benchmark Turbulent Flows," *AIAA Paper 2018-1102*, Jan. 2018, doi.org/10.2514/6.2018-1102.

[48]Michal, T. R., Kamenetskiy, D. S., and Krakos, J., "Anisotropic Adaptive Mesh Results for the Third High Lift Prediction Workshop (HiLiftPW-3)," *AIAA Paper 2018-1257*, Jan. 2018, doi.org/10.2514/6.2018-1257.

[49]Alauzet, F., Loseille, A., Marcum, D., and Michal, T., "Assessment of Anisotropic Mesh Adaptation for High-Lift Prediction of the HL-CRM configuration," *AIAA Paper 2017-3300*, June 2017, doi.org/10.2514/6.2017-3300.

[50]Mavriplis, D. J., "An Assessment of Linear versus Non-Linear Multigrid Methods for Unstructured Mesh Solvers," *Journal of Computational Physics*, Vol. 175, Jan. 2002, pp. 302–325.

[51]Mavriplis, D. J. and Mani, K., "Unstructured Mesh Solution Techniques using the NSU3D Solver," AIAA Paper 2014-081, 52nd Aerospace Sciences Meeting, National Harbor, MD.

[52]Isaacson, E. and Keller, H. B., *Analysis of Numerical Methods*, John Wiley and Sons, New York, NY, 1966.

[53]Saad, Y. and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM J. Sci. Stat. Comput.*, Vol. 7, No. 3, 1986, pp. 856–869.

[54]Wigton, L. B., Yu, N. J., and Young, D. P., "GMRES Acceleration of Computational Fluid Dynamic Codes," *Proceedings of the 7th AIAA CFD Conference*, July 1985, pp. 67–74, AIAA Paper 85-1494-CP.

[55]Wood, S., Burdyshaw, C. E., Erwin, J. T., Stefanski, D. L., and Peterson, G., "Strategy for Fine-Grained Parallelism in Multi-Level Computational Engineering Solvers," *AIAA Paper 2018-0397*, Jan. 2018, doi.org/10.2514/6.2018-0397.

[56]Mavriplis, D. J. and Venkatakrishnan, V., "A Unified Multigrid Solver for the Navier-Stokes Equations on Mixed Element Meshes," *International Journal for Computational Fluid Dynamics*, Vol. 8, 1997, pp. 247–263.

[57]Mavriplis, D. J., "Multigrid Strategies for Viscous Flow Solvers on Anisotropic Unstructured Meshes," *Journal of Computational Physics*, Vol. 145, No. 1, Sept. 1998, pp. 141–165.

[58]Mavriplis, D. J., "Directional Agglomeration Multigrid Techniques for High-Reynolds Number Viscous Flow Solvers," *AIAA Journal*, Vol. 37, No. 10, Oct. 1999, pp. 1222–1230.

[59]Mavriplis, D. J. and Pirzadeh, S., "Large-Scale Parallel Unstructured Mesh Computations for 3D High-Lift Analysis," *AIAA Journal of Aircraft*, Vol. 36, No. 6, Dec. 1999, pp. 987–998.

[60]Brandt, A., "Multigrid Techniques with Applications to Fluid Dynamics:1984 Guide," *VKI Lecture Series*, March 1984, pp. 1–176.

[61]"Second AIAA High Lift Prediction Workshop." San Diego, CA. http://highliftpw.larc.nasa.gov.

[62]Saad, Y., *Iterative Methods for Sparse Linear Systems*, PWS Series in Computer Science, PWS Publishing Company, Boston, MA, 1996.

[63]Pandya, M. J., Diskin, B., Thomas, J. L., and Frink, N. T., "Assessment of Preconditioner for a USM3D Hierarchical Adaptive Nonlinear Iteration Method (HANIM)," .

[64]Zingg, D. and Pueyo, A., "An Efficient Newton-GMRES Solver for Aerodynamic Computations," *Proceedings of the 13th AIAA CFD Conference, Snowmass, CO*, June 1997, pp. 712–721, AIAA Paper 97-1955-CP.

[65]Anderson, W. K., Gropp, W. D., Kaushik, D. K., Keyes, D. E., and Smith, B. F., "Achieving High Sustained Performance in an Unstructured Mesh CFD Application," *Bell Prize award paper, Special Category, in the Proceedings of SC'99, IEEE, Los Alamitos, CA.*, 1999.

[66]Reist, T. and Zingg, D. W., "Performance of a NewtonKrylovSchur Algorithm for Solving Steady Turbulent Flows," *AIAA Journal*, Vol. 54, No. 9, 2016.

[67]Hartmann, R., McMorris, H., and Leicht, T., "Curved grid generation and DG computation for the DLR-F11 high lift configuration," *Proceedings of the ECCOMAS Congress 2016, Crete Island, Greece, eds. M. Papadrakakis, V. Papadopoulos, G. Stefanou and V. Plevris*, June 2016.

[68]Ceze, M. and Fidkowski, K. J., "Pseudo-transient Continuation, Solution Update Methods, and CFL Strategies for DG Discretizations of the RANS-SA Equations," June 2013, AIAA Paper 2013-2686 presented at the 21st AIAA Computational Fluid Dynamics Conference, San Diego CA.

[69]Cai, X., Keyes, D. E., and Venkatakrishnan, V., "Newton-Krylov-Schwarz: An implicit solver for CFD," *Proceedings of Eighth International Conference on Domain Decomposition Methods (R. Glowinski et al., eds.), Wiley, New York*, 1997, pp. 387–400.

[70] Hwang, F.-N. and Cai, X.-C., “A parallel nonlinear additive Schwarz preconditioned inexact Newton algorithm for incompressible Navier–Stokes equations,” *Journal of Computational Physics*, Vol. 204, No. 2, 2005, pp. 666–691.

[71] Ahrabi, B. A. and Mavriplis, D. J., “A scalable solution strategy for high-order stabilized finite-element solvers using an implicit line preconditioner,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 341, 2018, pp. 956–984.

[72] Ahrabi, B. and Mavriplis, D. J., “An Implicit Block ILU Smoother for Preconditioning of Newton-Krylov Solvers with Application in Finite-Element Discretizations,” *AIAA Paper 2019-0101*, Jan. 2019, doi.org/10.2514/6.2019-0101.

[73] Tuminaro, R. S. and Tong, C., “Parallel Smoothed Aggregation Multigrid : Aggregation Strategies on Massively Parallel Machines,” Proceedings of the 2000 ACM/IEEE conference on Supercomputing.

[74] Heys, J., Manteuffel, T., McCormick, S. F., and Olson, L., “Algebraic multigrid for higher-order finite elements,” *Journal of computational Physics*, Vol. 204, No. 2, 2005, pp. 520–532.

[75] Tsuji, P. and Tuminaro, R., “Augmented AMG-shifted Laplacian preconditioners for indefinite Helmholtz problems,” *Numerical Linear Algebra with Applications*, Vol. 22, No. 6, 2015, pp. 1077–1101.

[76] Vassilevski, P. S. and Yang, U. M., “Reducing communication in algebraic multigrid using additive variants,” *Numerical Linear Algebra with Applications*, Vol. 21, No. 2, 2014, pp. 275–296.

[77] Tuminaro, R., “Private Communication,” May 2019.

[78] Smith, W. A., “Multigrid Solution of Transonic Flow on Unstructured Grids,” *Recent Advances and Applications in Computational Fluid Dynamics*, Nov. 1990, Proceedings of the ASME Winter Annual Meeting, Ed. O. Baysal.

[79] Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W. D., Karpeyev, D., Kaushik, D., Knepley, M. G., May, D. A., McInnes, L. C., Mills, R. T., Munson, T., Rupp, K., Sanan, P., Smith, B. F., Zampini, S., Zhang, H., and Zhang, H., “PETSc Web page,” http://www.mcs.anl.gov/petsc, 2019.

[80] Briggs, W. L., Henson, V. E., and McCormick, S. F., *A Multigrid Tutorial*, SIAM, Philadelphia, PA, 2000.

[81] Schwamborn, D., Gerhold, T., and Heinrich, R., “The DLR TAU-code: recent applications in research and industry,” 2006.

[82] Cambier, L., Gazaix, M., Heib, S., Plot, S., Poinot, M., Veuillot, J., Boussuge, J., and Montagnac, M., “An Overview of the Multi-Purpose elsA Flow Solver.” *AerospaceLab*, , No. 2, 2011, pp. p–1.

[83] Hashimoto, A., Murakami, K., Aoyama, T., Ishiko, K., Hishida, M., Sakashita, M., and Lahur, P., “Development of FAST Unstructured CFD Code FaSTAR,” *ICAS 2012*, 2012.

[84] Diskin, B. and Nishikawa, H., “Evaluation of multigrid solutions for turbulent flows,” *AIAA Paper 2014-0082*, Jan. 2014.

[85] Pulliam, D. J. T. and Buning, P., “Recent Enhancements to OVERFLOW,” Jan. 1997, AIAA Paper 97-0644.

[86] Nocedal, J. and Wright, S. J., *Numerical Optimization*, Springer Series in Optimization Research, 2000.

[87] Ceze, M. and Fidkowski, K. J., “Constrained pseudotransient continuation,” *International Jounral for Numerical Methods in Engineering*, Vol. 102, No. 11, 2015, pp. 1683–1703.

[88] Kelley, C. T. and Keyes, D. E., “Convergence analysis of pseudo-transient continuation,” *SIAM Journal on Numerical Analysis*, Vol. 35, No. 2, April 1998, pp. 508–523.

[89] Brown, D. A. and Zingg, D. W., “Advances in Homotopy Continuation Methods in Computational Fluid Dynamics,” *AIAA Paper 2013-2944*, June 2013, doi.org/10.2514/6.2013-2944.

[90] Prosser, D. and Glasby, R. S., “Evaluation and Improvement of Robustness, Speed, and Accuracy of the COFFE CFD Solver,” *AIAA Paper 2019-1344*, Jan. 2019, doi.org/10.2514/6.2019-1344.

[91] Cai, X.-C. and Keyes, D. E., “Nonlinearly preconditioned inexact Newton algorithms,” *SIAM Journal on Scientific Computing*, Vol. 24, No. 1, 2002, pp. 183–200.

[92] Liu, L. and Keyes, D. E., “Field-split preconditioned inexact Newton algorithms,” *SIAM Journal on Scientific Computing*, Vol. 37, No. 3, 2015, pp. A1388–A1409.

[93] Mavriplis, D. J., “A Residual Smoothing Strategy for Accelerating Newton Method Continuation,” *arXiv:1805.03756 [math.NA]*, March 2018.

[94] Mavriplis, D. J., Ahrabi, B., and Brazell, M., “Strategies for Accelerating Newton Method Continuation for CFD Problems,” *AIAA Paper 2019-0100*, Jan. 2019, doi.org/10.2514/6.2019-0100.

[95] Ahrabi, B. R. and Mavriplis, D. J., “Scalable Solution Strategies for Stabilized Finite-Element Flow Solvers on Unstructured Meshes: Part II,” *AIAA Paper 2017-4257, Presented at the 23rd AIAA Computational Fluid Dynamics Conference, Denver CO*, June 2017.

[96] Mousseau, V. A., Knoll, D. A., and Rider, W. J., “Physics-based preconditioning and the Newton-Krylov method for non-equilibrium radiation diffusion,” *Journal of Computational Physics*, Vol. 160, No. 2, 2000, pp. 743–765.

[97] Knoll, D. A. and Keyes, D. E., “Jacobian-free NewtonKrylov methods: a survey of approaches and applications,” *Journal of Computational Physics*, Vol. 193, No. 2, 2004, pp. 357–397.

[98] Kenway, G. K. W., Kennedy, G. J., and Martins, J. R. R. A., “Scalable Parallel Approach for High-Fidelity Steady-State Aeroelastic Analysis and Adjoint Derivative Computations,” *AIAA Journal*, Vol. 55, No. 5, May 2014, doi.org/10.2514/1.J052255.

[99] Brooks, T. R., Kenway, G. K. W., and Martins, J. R. R. A., “Benchmark Aerostructural Models for the Study of Transonic Aircraft Wings,” *AIAA Journal*, Vol. 56, No. 7, July 2018, pp. 2840–2855.

[100] Zhang, Z. J. and Zingg, D. W., “Efficient Monolithic Solution Algorithm for High-Fidelity Aerostructural Analysis and Optimization,” *AIAA Journal*, Vol. 56, No. 3, March 2018, doi.org/10.2514/1.J056163.

[101] Gray, J. S., Hwang, J. T., Martins, J. R. R. A., Moore, K. T., and Naylor, B. A., “OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization,” *Structural and Multidisciplinary Optimization*, Vol. 59, No. 4, April 2019, pp. 1075–1104.

[102] Leffell, J., Sitaraman, J., Lakshminarayan, V. K., and Wissink, A. M., “Towards Efficient Parallel-in-Time Simulation of Periodic Flows,” *AIAA Paper 2016-0066*, Jan. 2016, doi.org/10.2514/6.2016-0066.

[103] Mundis, N. and Mavriplis, D., "Toward an Optimal Solver for Time-spectral Solutions on Unstructured Meshes," *AIAA Paper 2016-0069, Presented at the 54rd AIAA Aerospace Sciences Conference, San Diego CA*, Jan. 2016.

[104] Ramezaninan, D. and Mavriplis, D. J., "An Order NlogN Parallel Newton-Krylov Solver for Time Spectral Problems," *AIAA Paper 2017-4509*, Jan. 2019, doi.org/10.2514/6.2017-4509.

[105] Brazell, M. J., Sitaraman, J., and Mavriplis, D. J., "An overset mesh approach for 3D mixed element high-order discretizations," *Journal of Computational Physics*, Vol. 322, 2016, pp. 33–51.

[106] Kirby, A. C., Brazell, M., Yang, Z., Roy, R., Reza Ahrabi, B., Mavriplis, D., Sitaraman, J., and Stoellinger, M. K., "Wind farm simulations using an overset hp-adaptive approach with blade-resolved turbine models," *International Journal of High Performance Computing Applications*, 2019, doi.org/10.1177/1094342019832960.

[107] Vatsa, V., Carpenter, M., and Lockard, D., "Re-evaluation of an Optimized Second Order Backward Difference (BDF2OPT) Scheme for Unsteady Flow Applications," *AIAA Paper 2010-122*, June 2010, doi.org/10.2514/6.2010-122.

[108] Bijl, H., Carpenter, M. H., Vatsa, V. N., and Kennedy, C. A., "Implicit time integration schemes for the unsteady incompressible Navier-Stokes Equations: laminar flow," *Journal of Computational Physics*, Vol. 179, No. 1, 2002, pp. 313–329.

[109] Jameson, A., "Application of Dual Time Stepping to Fully Implicit Runge Kutta Schemes for Unsteady Flow Calculations," *AIAA Paper 2015-2753*, June 2015, doi.org/10.2514/6.2015-2753.

[110] Ekici, K. and Hall, K., "Nonlinear Analysis of Unsteady Flows in Multistage Turbomachines using Harmonic Balance," *AIAA Journal*, Vol. 45, No. 5, 2007, pp. 1047–1057.

[111] Gopinath, A. and Jameson, A., "Time Spectral Method for Periodic Unsteady Computations over Two- and Three-Dimensional Bodies," *AIAA-Paper 2005-1220*, No. Jan., 2005.

[112] Mavriplis, D. J. and Yang, Z., "Time Spectral Method for Periodic and Quasi-Periodic Unsteady Computations on Unstructured Meshes," *Mathematical Modeling of Natural Phenomena*, Vol. 6, No. 3, May 2011, pp. 213–236, DOI: http://dx.doi.org/10.1051/mmnp/20116309.

[113] Mishra, A., Mavriplis, D. J., and Sitaraman, J., "Time-dependent Aero-elastic Adjoint-based Aerodynamic Shape Optimization of Helicopter Rotors in Forward Flight," AIAA Paper 2015-1130, 56th AIAA Aerospace Sciences Meeting, Kissimmee, FL.

[114] Pazner, W. and Persson, P.-O., "Stage-parallel fully implicit Runge–Kutta solvers for discontinuous Galerkin fluid simulations," *Journal of Computational Physics*, Vol. 335, 2017, pp. 700–717.

[115] Wissink, A., Jayaraman, B., Datta, A., Sitaraman, J., Potsdam, M., Kamkar, S., Mavriplis, D., Yang, Z., Jain, R., Lim, J., and Strawn, R., "Capability Enhancements in Version 3 of the HELIOS High Fidelity Rotorcraft Simulation Code," AIAA Paper 2012-713, Presented at the 50th AIAA Aerospace Sciences Meeting and Exhibit, Nashville TN.

[116] Chen, S. and Doolen, G. D., "Lattice Boltzmann Method for Fluid Flows," *Annual Review of Fluid Mechanics*, Vol. 30, 1998, pp. 329–364.

[117] He, X. and Luo, L.-S., "A priori derivation of the lattice Boltzmann equation," *Phys. Rev. E*, Vol. 55, 1997, pp. 6333–6336.

[118] Bodart, J., Larsson, J., and Moin, P., "Large eddy simulation of high-lift devices," *AIAA Paper 2013-2724*, June 2013, doi.org/10.2514/6.2013-2724.

[119] Moin, P. and Verzicco, R., "On the suitability of second-order accurate discretizations for turbulent flow simulations," *European Journal of Mechanics-B/Fluids*, Vol. 55, 2016, pp. 242–245.

[120] Park, G. I. and Moin, P., "Numerical aspects and implementation of a two-layer zonal wall model for LES of compressible turbulent flows on unstructured meshes," *Journal of Computational Physics*, Vol. 305, 2016, pp. 589–603.

[121] Rozema, W., Bae, H. J., Moin, P., and Verstappen, R., "Minimum-dissipation models for large-eddy simulation," *Physics of Fluids*, Vol. 27, No. 8, 2015, pp. 085107.

[122] Maday, Y. and Patera, A. T., "Spectral element methods for the incompressible Navier-Stokes equations," *IN: State-of-the-art surveys on computational mechanics (A90-47176 21-64). New York, American Society of Mechanical Engineers, 1989, p. 71-143. Research supported by DARPA.*, 1989, pp. 71–143.

[123] Taylor, G. I. and Green, A. E., "Mechanism of the production of small eddies from large ones," *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences*, Vol. 158, No. 895, 1937, pp. 499–521.

[124] Wang, Z. J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H. T., et al., "High-order CFD methods: current status and perspective," *International Journal for Numerical Methods in Fluids*, Vol. 72, No. 8, 2013, pp. 811–845.

[125] Wang, Z. J., Liu, Y., Lacor, C., and Azevedo, J. L. F., "Spectral Volume and Spectral Difference Methods," *Handbook of Numerical Analysis*, Vol. 17, 2016, pp. 199–226.

[126] Gassner, G. J., Winters, A. R., and Kopriva, D. A., "Split form nodal discontinuous Galerkin schemes with summation-by-parts property for the compressible Euler equations," *Journal of Computational Physics*, Vol. 327, 2016, pp. 39–66.

[127] Abgrall, R. and Roe, P. L., "High order fluctuation schemes on triangular meshes," *Journal of Scientific Computing*, Vol. 19, No. 1-3, 2003, pp. 3–36.

[128] Vincent, P. E., Castonguay, P., and Jameson, A., "A new class of high-order energy stable flux reconstruction schemes," *Journal of Scientific Computing*, Vol. 47, No. 1, 2011, pp. 50–72.

[129] Huynh, H., Wang, Z. J., and Vincent, P. E., "High-order methods for computational fluid dynamics: a brief review of compact differential formulations on unstructured grids," *Computers & Fluids*, Vol. 98, 2014, pp. 209–220.

[130] Abgrall, R., Meledo, E. l., and Oeffner, P., "On the Connection between Residual Distribution Schemes and Flux Reconstruction," *arXiv preprint arXiv:1807.01261*, 2018.

[131] Abgrall, R., "A general framework to construct schemes satisfying additional conservation relations. Application to entropy conservative and entropy dissipative schemes," *Journal of Computational Physics*, Vol. 372, 2018, pp. 640–666.

[132] Chan, J., Wang, Z., Modave, A., Remacle, J.-F., and Warburton, T., "GPU-accelerated discontinuous Galerkin methods on hybrid meshes," *Journal of Computational Physics*, Vol. 318, 2016, pp. 142–168.

[133] "MFEM: A Modular Finite Element Library," https://mfem.org/.

[134] Witherden, F. D., Farrington, A. M., and Vincent, P. E., "PyFR: An open source framework for solving advection–diffusion type problems on streaming architectures using the flux reconstruction approach," *Computer Physics Communications*, Vol. 185, No. 11, 2014, pp. 3028–3040.

[135] Karniadakis, G. E. and Sherwin, S. J., *Spectral/hp Element Methods for CFD*, Oxford University Press, 1999.

[136] Sherwin, S. and Karniadakis, G. E., "A triangular spectral element method; applications to the incompressible Navier-Stokes equations," *Computer methods in applied mechanics and engineering*, Vol. 123, No. 1-4, 1995, pp. 189–229.

[137] Brazell, M., Kirby, A., and Mavriplis, D., "A High-order Discontinuous-Galerkin Octree-Based AMR Solver for Overset Simulations," AIAA Paper 2017-3944, 23nd AIAA Computational Fluid Dynamics Conference, Denver CO.

[138] Flad, D. and Gassner, G., "On the use of kinetic energy preserving DG-schemes for large eddy simulation," *Journal of Computational Physics*, Vol. 350, 2017, pp. 782–795.

[139] Carpenter, M. H. and Fisher, T. C., "High-order entropy stable formulations for computational fluid dynamics," *AIAA Paper 2013-2868, 21st AIAA Computational Fluid Dynamics Conference*, 2013, p. 2868.

[140] Carpenter, M., Fisher, T., Nielsen, E., Parsani, M., Svärd, M., and Yamaleev, N., "Entropy stable summation-by-parts formulations for compressible computational fluid dynamics," *Handbook of Numerical Analysis*, Vol. 17, Elsevier, 2016, pp. 495–524.

[141] Murman, S. M., Diosady, L., Garai, A., and Ceze, M., "A space-time Discontinuous-Galerkin approach for separated flows," *AIAA Paper 2016-1059, 54th AIAA Aerospace Sciences Meeting*, 2016, p. 1059.

[142] Pazner, W. and Persson, P.-O., "Approximate tensor-product preconditioners for very high order discontinuous Galerkin methods," *Journal of Computational Physics*, Vol. 354, 2018, pp. 344–369.

[143] Diosady, L. T. and Murman, S. M., "Scalable Tensor-Product Preconditioners for High-Order Finite-Element Methods: Scalar Equations," *Journal of Computational Physics*, 2019.

[144] Hornung, R., Wissink, A., and Kohn, S., "Managing Complex Data and Geometry in Parallel Structured AMR Applications," *Engineering with Computers*, 2006, doi:10.1007/s00366-006-0038-6.

[145] Burstedde, C., Wilcox, L. C., and Ghattas, O., "p4est: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees," *SIAM Journal on Scientific Computing*, Vol. 33, No. 3, 2011, pp. 1103–1133.

[146] "AMReX:A software framework for massively parallel, block-structured adaptive mesh refinement (AMR) applications," https://amrex-codes.github.io/amrex.