Strategies for Accelerating Newton Method Continuation in CFD Problems

Dimitri J. Mavriplis *
Behzad Ahrabi †
Michael Brazell ‡
Department of Mechanical Engineering, University of Wyoming, Laramie, WY 82071

Various techniques for accelerating global convergence of pseudo-transient continuation Newton methods are proposed and demonstrated in this paper. A residual smoothing technique is constructed, which is motivated by the effectiveness of local nonlinear smoothers at overcoming strong nonlinear transients. This method is demonstrated on applied aerodynamic CFD problems of practical interest for finite-volume and finite-element steady-state discretizations, where significant gains in computational efficiency are demonstrated. In a second part, a two-level non-linear multigrid strategy is developed and implemented within the context of a Newton-Krylov scheme. This approach is also demonstrated for finite-volume and finite-element discretizations and is shown to accelerate significantly the continuation phase of Newton methods.

I. Introduction

Newton methods have become popular strategies for solving large-systems of non-linear equations. In the field of computational fluid dynamics (CFD) for aerodynamics, Newton methods have enjoyed a resurgence in popularity, largely due to their ability to provide deep convergence levels for stiff systems of equations such as those resulting from emerging continuous and discontinuous Galerkin discretizations particularly for highly resolved steady-state problems.[1–4] In the final stages of convergence, when the iterative solution state is close to the exact nonlinear solution, Newton methods converge in a small number of nonlinear steps, and each step can often be solved effectively using preconditioned Krylov methods which are generally robust.[5–7] However, for most cases, a continuation strategy must be employed to iteratively approach the nonlinear state where fast convergence is obtained. A typical strategy consists of employing a pseudo-transient approach where a pseudo-time term is added to the diagonal of the Jacobian matrix with a variable pseudo-time step.[4,8,9] In the initial phases of convergence, when the pseudo-time step is small, the method approximates a pseudo-time explicit scheme, and in the final stages, when the pseudo-step becomes large, the exact Newton method is recovered.

Although pseudo-transient continuation for Newton methods is widely used for CFD type problems, experience has shown that this approach can be slow to converge and often evolves through nonlinear states that produce convergence stagnation or even unrealizable solutions. A notable mode of failure or inefficiency occurs when isolated residuals in the nonlinear system remain large and retard the entire global solution. This has prompted research into methods such as nonlinear elimination,[10,11] or nonlinear preconditioning[12–15] which attempt to break up the problem into smaller more local nonlinear problems, based on the observation that local nonlinear solution methods can often overcome the difficulties encountered by global continuation Newton methods.

Alternatively, the ability of localized residuals to retard the entire global nonlinear problem may be attributed to a non-smooth residual distribution, since for a perfectly smooth residual distribution, it should be possible to advance the global problem uniformly. Therefore, a residual smoothing technique is proposed

*Professor; email: mavriplis@uwyo.edu
†Research Associate; email: brezaahr@uwyo.edu
‡Postdoctoral Researcher; email: mbrazell@uwyo.edu

for accelerating the continuation process of a pseudo-transient continuation Newton-Krylov method. Residual smoothing can be achieved through a variety of approaches, such as using well-known local nonlinear smoothers including block-Jacobi and line smoothers. In the limit of small pseudo-time steps, the nonlinear iterations of the smoothed Newton continuation scheme correspond to a nonlinear implicit solver, while in the limit of large pseudo-time steps, the exact Newton scheme is recovered, along with its quadratic convergence properties. A second complementary approach is proposed based on a nonlinear two-level multigrid strategy. In this approach, rather than splitting the global nonlinear problem into smaller more local nonlinear problems, a surrogate nonlinear problem with fewer degrees of freedom is formed by transferring the nonlinear equations to a coarser mesh (in h or p) which presumably is easier to solve than the fine level problem. This coarse mesh surrogate problem is designed to be consistent with the fine level discrete solution using the standard full-approximation-storage (FAS) multigrid defect-correction approach. By using a two level scheme and combining this strategy with Newton method continuation, the robustness deficiencies of traditional nonlinear multigrid methods for stiff problems are minimized.

In the following section, we begin with the motivation for the residual smoothing approach and describe the formulation and its implementation in detail. In section III, results for this approach using a second-order accurate finite-volume discretization of the Reynolds averaged Navier-Stokes (RANS) equations for a transonic aerodynamic problem are presented and discussed. In section IV, results using the residual smoothing approach applied to second-order accurate (p=1) and third-order accurate (p=2) Streamwise Upwind Petrov Upwind Galerkin (SUPG) and Discontinuous Galerkin (DG) RANS discretizations on standard aerodynamic test cases are shown. In section V, the two-level nonlinear multigrid strategy is discussed and results using this approach are shown for a third-order accurate (p=2) SUPG discretization and for a second-order accurate finite-volume discretization on the same problem discussed previously in Section III. Conclusions and prospects for further improvements in Newton solver efficiency and robustness are given in Section VI.

II. Formulation of Residual Smoothing Approach

II.A. Motivation of the approach

We are interested in solving a non-linear set of equations denoted as

$$R(w) = 0 \tag{1}$$

where R represents the residual vector which is a nonlinear function of the state vector w for which we seek the solution. In this work, the residual $R(w)$ may arise from the spatial discretization of a steady-state CFD problem, or from the equations to be solved at each step of an implicit time-dependent CFD problem. The Newton scheme for this problem corresponds to solving multiple linear problems of the form

$$\left[\frac{\partial R(w^n)}{\partial w^n}\right] \Delta w^n = -R(w^n) \tag{2}$$

with nonlinear updates given as

$$w^{n+1} = w^n + \alpha \Delta w^n \quad with\ 0 \leq \alpha \leq 1 \tag{3}$$

where the parameter α is determined by a line search which seeks to minimize the L2 norm of the residual vector $R(w^{n+1})$. In most cases, a continuation method is required in order to advance the solution through initial nonlinear transients in order to reach a state close enough to the final solution where the quadratic convergence of Newton's method is observed. Pseudo-transient continuation (PTC) is an often used approach which seeks to mimic the physical time evolution of the solution by adding a pseudo-time term of the form:

$$\frac{M}{\Delta\tau}(w^{n+1} - w^n) + R(w^{n+1}) = R_t(w^{n+1}) = 0 \tag{4}$$

Here M denotes a suitable mass matrix, $\Delta\tau$ represents the pseudo-time step and R_t corresponds to the augmented pseudo time-dependent residual. Since the objective is to obtain the solution of $R(w) = 0$, pseudo-time accuracy is not a concern and a simple first-order pseudo-time discretization (BDF1) is suitable. Newton's method is applied to equation (4) in the standard manner, by linearizing $R_t(w)$ about the current state and solving for the update as:

$$\left[\frac{M}{\Delta\tau} + \left[\frac{\partial R(w^n)}{\partial w^n}\right]\right] \Delta w^n = -R_t(w^n) \tag{5}$$

Here we note that the pseudo-unsteady residual evaluated at the current state $R_t(w^n)$ is equal to the original residual and the above equation can be rewritten as:

$$\left[\frac{M}{\Delta\tau} + \left[\frac{\partial R(w^n)}{\partial w^n}\right]\right]\Delta w^n = -R(w^n) \tag{6}$$

Finally, the nonlinear update proceeds as determined previously by equation (3) where α can be determined by a line search. In this case, the line search seeks to minimize the L2 norm of the pseudo-unsteady residual, i.e. $F(\alpha) = ||R_t(w + \alpha\Delta w)||_2$, with

$$R_t(w + \alpha\Delta w) = \frac{M}{\Delta\tau}\alpha\Delta w^n + R(w^n + \alpha\Delta w^n) \tag{7}$$

Here we emphasize that it is the (pseudo) time-dependent residual R_t that should be minimized by the line search, instead of the residual R itself. In fact, it is well known that the line search is guaranteed to produce a descent direction for the L2 norm of this residual provided an exact linearization is used and equation (6) is solved exactly. This can be shown by denoting the Jacobian matrix as $A = \left[\frac{M}{\Delta\tau} + \left[\frac{\partial R(w^n)}{\partial w^n}\right]\right]$, and premultiplying equation (5) by $\Delta w^T A^T$, obtaining (dropping the n-superscripts for clarity):[9]

$$\Delta w^T A^T A \Delta w = -\Delta w^T A^T R_t \tag{8}$$

Since the left-hand side of the above equation is a positive scalar quantity, it follows that

$$\Delta w^T A^T R_t(w) = \frac{1}{2}\Delta w^T \frac{d||R_t(w)||_2}{dw} \approx \frac{1}{2}\Delta||R_t(w)||_2 < 0 \tag{9}$$

Therefore, the solution Δw of the linear problem arising from Newton's method in equation (5) or (6) is a descent direction for $||R_t(w)||_2$ and a decrease in this residual is guaranteed for small enough α, thus ensuring the success of the line search.

In the PTC approach, the pseudo-time step $\Delta\tau$ can either represent a global or local time step. Local time-stepping is most appealing, since we are not interested in time accuracy and since this approach should result in faster convergence rates, particularly for problems with large variations in mesh cell sizes as is typical in the applications considered in this work. In these cases, a pseudo-time step at each mesh cell i is computed as:

$$\Delta\tau_i = CFL_{NK}.\Delta\tau_{explicit_i} \tag{10}$$

where $\Delta\tau_{explicit_i}$ represents an estimate of the explicit time step limit in each cell and CFL_{NK} is a global scalar that controls the magnitude of all local time steps. (i.e. a CFL number). During the continuation process, the CFL number starts out at a low value, of the order of 1, and grows at each nonlinear step, such that it reaches a large value at the end of the convergence process, thus recovering an exact Newton scheme for the steady residual problem.

In practice, growth of the CFL number is controlled based on the results of the line search process. If a full non-linear update ($\alpha = 1.0$) is taken or if $\alpha > 0.75$, the CFL value is amplified by the factor $\beta_{CFL1} > 1$, whereas for small values $\alpha \leq 0.1$, the update is rejected and the CFL is reduced by the factor $\beta_{CFL2} < 1$. For intermediate values of α the CFL value remains unchanged. Based on experience, the values $\beta_{CFL1} = 1.5$ and $\beta_{CFL2} = 0.1$ are generally used, since these have been found to provide a good compromise between speed of convergence and robustness, although other settings are possible.

By its very construction, this CFL-controlled PTC Newton method can never result in solver divergence. However, stagnation can occur when the solver gets into a nonlinear state from which it cannot recover, and the CFL is continuously reduced to very small values ($CFL_{NK} << 1$) producing vanishingly small nonlinear updates. As an example, this may occur when the solution approaches a non-physical state which would result in localized negative pressure or density values in computational fluid dynamics problems.

In order to better understand the behavior of the PTC Newton method, we consider the limiting cases of large and small $\Delta\tau$ values. Clearly, for large $\Delta\tau$, equation (6) approaches equation (2) and an exact Newton scheme is recovered. On the other hand, for small $\Delta\tau$, equation (6) becomes

$$\left[\frac{M}{\Delta\tau}\right]\Delta w^n = -R(w^n) \tag{11}$$

which results in updates of the form

$$\Delta w^n = -\Delta\tau M^{-1} R(w^n) \tag{12}$$

For a finite-volume scheme where the mass matrix corresponds to the cell volume, this reduces to an explicit time-stepping scheme (using local time steps in this case). Even though this should be (linearly) stable for values of $CFL_{NK} \simeq 1$, there are many situations where transient nonlinear states coupled with the PTC controller result in values $CFL_{NK} << 1$ thus producing solver stagnation. Surprisingly, for many cases where solver stagnation occurs, experience has shown that commonly used local nonlinear solvers (i.e. non-linear Jacobi, Gauss-Seidel or line solvers) converge reliably and without difficulty. In some sense, equation (12) represents one of the weakest and most inefficient approaches for converging a nonlinear problem. For example, when solving coupled systems of PDE's, a point implicit approach is much more effective than a scalar explicit approach. A simple point implicit approach can be written as

$$\Delta w^n = -D^{-1} R(w^n) \tag{13}$$

where D^{-1} represents the inverse of the diagonal block-Jacobian at each mesh point or cell coupling all equations together. The justification for the PTC approach is that the non-linear solver will spend very little time in this regime and that the CFL value will grow quickly, transitioning the solver to a more implicit approach. However, experience has shown that stagnation at small CFL values is a common failure mode, and even for problems that converge reliably, a large portion of the overal computational time is spent in the initial transient regime at low to moderate CFL values. Furthermore, we speculate that updates of the form given by equation (12) result in poor error smoothing properties leading to nonlinear states that produce linear systems for Newton's method that are ill-conditioned (difficult to solve) as well as isolated residuals that impede global convergence and sometimes lead to stagnation. On the other hand, local nonlinear solvers can be designed specifically for good error smoothing properties, as is usually done for nonlinear multigrid problems. A simple example is provided by the Runge-Kutta time-stepping schemes devised for FAS multigrid methods in CFD.[16,17] Particularly, for problems with highly anisotropic meshes (or matrix coefficients), explicit methods as well as point-implicit methods are known to have poor smoothing properties and directional or line solvers are required.[18–20]

Although Newton methods provide a reliable approach for attaining deep convergence levels for stiff nonlinear problems, the issues encountered with these methods for evolving the solution through initial nonlinear transients, coupled with the success of simpler local nonlinear solvers in this regime, have prompted research into nonlinear preconditioning methods and other approaches that attempt to break up the problem into smaller more local nonlinear problems.[12–15] A simpler approach consists of initiating the solution process with a local nonlinear solver, and then switching to a Newton scheme at some point along the solution process. Aside from lacking elegance, this approach raises various questions such as how to determine the optimum point for switching solvers, and whether or when to return to the nonlinear solver, should the Newton solver fail to converge.[6]

In the context of PTC, one would like to design a solver that behaves as a local nonlinear solver rather than an explicit scheme in the limit of small $\Delta\tau$, while still recovering an exact Newton solver in the limit of large $\Delta\tau$. Using equation (13) to denote a generic local nonlinear solver, where D may represent a reduced Jacobian (i.e. block diagonal, lower triangular or tridiagonal line structure) or more generally any preconditioner that approximates the Jacobian, one approach may be to modify or replace the mass matrix in equation (6) as:

$$\left[\frac{D}{\Delta\tau} + \left[\frac{\partial R(w^n)}{\partial w^n}\right]\right] \Delta w^n = -R(w^n) \tag{14}$$

However, in the limit of small $\Delta\tau$, this results in updates of the form

$$\Delta w^n = -D^{-1}\Delta\tau R(w^n) \tag{15}$$

Unfortunately, the updates are still proportional to the pseudo-time step $\Delta\tau$, with the result that stagnation may still occur. In essence, what is needed is a formulation that reduces to equation (13) in the limit of small $\Delta\tau$ while still recovering Newton's method for large $\Delta\tau$. This can be achieved using a scheme of the form:

$$\left[\alpha(\Delta\tau)D + \beta(\Delta\tau)\left[\frac{\partial R(w^n)}{\partial w^n}\right]\right] \Delta w^n = -R(w^n) \tag{16}$$

where $\alpha(\Delta\tau)$ and $\beta(\Delta\tau)$ are constructed as :

$$\alpha(\Delta\tau) = \frac{1}{1+\Delta\tau} \tag{17}$$

$$\beta(\Delta\tau) = \frac{\Delta\tau}{1+\Delta\tau} \tag{18}$$

in order to have the desired asymptotic behavior

$$\alpha(\Delta\tau) \to 1 \quad \beta(\Delta\tau) \to 0 \quad for\ \Delta\tau << 1 \tag{19}$$

$$\alpha(\Delta\tau) \to 0 \quad \beta(\Delta\tau) \to 1 \quad for\ \Delta\tau >> 1 \tag{20}$$

There are two undesirable issues with the above formulation. Firstly, recalling that D may represent any suitable preconditioning matrix, this approach results in a modified left-hand-side matrix, the properties of which may be different than the original Jacobian matrix, particularly in regions $\Delta\tau \sim 1$. This may require a redesign of the linear iterative solution techniques used to (approximately) invert the Jacobian, which constitutes a drawback from an implementation viewpoint. Secondly, and perhaps more importantly, the left-hand side Jacobian in equation (16) no longer corresponds to an exact linearization of the right-hand side, with the result that the line search procedure cannot be guaranteed to produce a descent direction for the residual $R(w^n)$.

An alternative approach consists of retaining the same left-hand side as in equation (6) but modifying the right-hand side as:

$$\left[\frac{M}{\Delta\tau} + \left[\frac{\partial R(w^n)}{\partial w^n}\right]\right] \Delta w^n = -R(w^n) - \frac{M}{\Delta\tau} D^{-1} R(w^n) \tag{21}$$

Clearly, with this formulation, when $\Delta\tau << 1$, the local nonlinear update form of equation (13) is recovered, while in the limit $\Delta\tau >> 1$, the exact Newton scheme is obtained. The fact that this formulation can be interpreted as a residual smoothing approach can be seen by re-writing equation (21) as:

$$\left[\frac{M}{\Delta\tau} + \left[\frac{\partial R(w^n)}{\partial w^n}\right]\right] \Delta w^n = -\left[I + \frac{M}{\Delta\tau} D^{-1}\right] R(w^n) \tag{22}$$

and noting that the term $\frac{M}{\Delta\tau} D^{-1}$ is non-dimensional, since both D and $\frac{M}{\Delta\tau}$ scale as the Jacobian matrix. Therefore, the right-hand side in equation (22) is seen to be an average of residuals with the averaging determined by the sparsity pattern of D^{-1}. This approach has the advantage that the left-hand side Jacobian of the system remains identical to that used in the original PTC continuation Newton method, with the result that the same linear solver techniques can be used without modification. Rather, the only change is the addition of a constant source term on the right hand side constructed as the scaled corrections resulting from a local nonlinear solution step. Therefore, given an existing Newton approach, the implementation of this scheme is relatively straight-forward.

Additionally, the line search procedure is still guaranteed to yield a descent direction for the relevant smoothed residual vector R_{sm} given as:

$$R_{sm}(w + \alpha\Delta w) = \frac{M}{\Delta\tau}\alpha\Delta w^n + R(w^n + \alpha\Delta w^n) + \frac{M}{\Delta\tau} D^{-1} R(w^n) \tag{23}$$

Here it is important to note that the smoothing term which corresponds to the scaled nonlinear updates is evaluated at the start of the Newton step and held constant throughout the line search. Therefore, this term drops out in the linearization process and the left-hand side Jacobian remains an exact linearization of the smoothed residual vector defined above, thus guaranteeing that the solution of the Newton step corresponds to a descent direction for R_{sm}.

II.B. Generalization and Interpretation

In the above discussion, the residual smoothing term proportional to $D^{-1}R(w^n)$ was presented as the result of a local nonlinear update, where D represents a suitable preconditioning matrix which can be either block diagonal (block-Jacobi), lower triangular (Gauss-Seidel), tridiagonal (line-solver) or any suitable approximation of the full Jacobian $\frac{\partial R(w)}{\partial w}$ which is known to have good smoothing properties. In practice, the additional

source term is formed by first computing an update produced by a local nonlinear solver as determined by equation (13), multiplying by $\frac{M}{\Delta\tau}$ and adding this term to the unsmoothed residual $R(w^n)$. Under this description, the nature of this smoothing source term can be broadened to include any sequence of nonlinear updates applied to the original system of nonlinear equations $R(w) = 0$. For example, the operator D^{-1} may be designed as a sequence of preconditioned nonlinear Runge-Kutta (RK) stages designed for smoothness following previous work[17,20,21] as:

$$\begin{aligned} w^0 &= w^n \\ w^1 &= w^0 - \alpha_1 [P]^{-1} R(w^0) \\ w^2 &= w^0 - \alpha_2 [P]^{-1} R(w^1) \\ w^3 &= w^0 - \alpha_3 [P]^{-1} R(w^2) \\ &... \\ w^k &= w^0 - \alpha_k [P]^{-1} R(w^{k-1}) \\ w^{n+1} &= w^k \end{aligned} \tag{24}$$

for a k-stage scheme where α_k represent the R-K scheme coefficients and [P] represents a local preconditioning matrix. In this case, the final nonlinear update is given as

$$\Delta w^n = w^{n+1} - w^n = w^k - w^0 = -D^{-1}(R(w^n)) \tag{25}$$

which implicitly defines the composite nonlinear operator D^{-1}. In this example, the RK scheme corresponds to a sequence of nonlinear updates, and the matrix [P] may be held fixed or evolved as a function of the nonlinear state w^k at each stage. In practice, any number of nonlinear iterations may be used to form the smoothing source term, and any nonlinear solver may be used, including multiple passes of preconditioned RK, nonlinear block-Jacobi, Gauss-Seidel, line solvers, or even linear or non-linear (FAS) multigrid. Presumably, more passes will provide smoother residual terms and accelerated non-linear convergence overall.

The proposed solution strategy may be interpreted in two different manners. On the one hand, it can be seen as a more elegant strategy (compared to switching solvers) for smoothly transitioning from local nonlinear smoothers, which are well known to provide good initial convergence, to an increasingly exact Newton method, which is generally more successful in obtaining deep final convergence, as the pseudo-time step is increased.

The second interpretation is that of residual smoothing, where the process ensures a smooth field of residuals on the right-hand-side of equation (22), thus producing smooth updates to the solution vector at each Newton step, thereby avoiding cases where isolated residuals hold up global convergence and avoiding evolution of the solution state into unphysical non-smooth states that either cause failure, are difficult or costly to recover from using the typical PTC controller strategy of lowering $\Delta\tau$, or which may drive the pseudo-time step to zero and cause stagnation. Indeed, using the current line-search based controller, if the state does become ill-conditioned causing the pseudo-time step to be reduced, the scheme reverts to the local nonlinear smoother $\Delta w^n = -D^{-1}(R(w^n))$ even in the limit $\Delta\tau << 1$, which results in a lower bound on the degree to which corrections are reduced with smaller pseudo-time steps, and which is inherently self-correcting.

III. Residual Smoothing Results for Finite-Volume Discretization

We seek solutions of the Reynolds-averaged Navier-Stokes (RANS) equations, which consist of the compressible form of the Navier-Stokes equations augmented with the one-equation Spalart-Allmaras turbulence model.[22,23] In three-dimensions, this yields a system of 6 coupled PDE's describing conservation of mass (1), momentum (3), energy (1), and turbulence convection/diffusion/production (1). For the results in this section, we employ a vertex-based finite-volume discretization on unstructured meshes of mixed element types, using isotropic tetrahedral elements in off-body regions and highly anisotropic prismatic elements in near-body regions to capture thin boundary layer gradients, as is typical for external aerodynamics CFD problems. The finite-volume discretization employs a matrix-based artificial dissipation formulation which is second-order accurate and which employs a distance-2 (neighbor of neighbor) stencil.[6,24] This results in a relatively large bandwidth Jacobian matrix which is generally considered impractical to store in memory. However, exact Jacobian-vector products are available through hand-differentiation of the residual routine

and are used in the Newton-Krylov solver. Alternatively, the Jacobian of the corresponding first-order accurate discretization has a nearest-neighbor stencil and may be stored for use as a preconditioner or for direct use in the local nonlinear solvers as described below.

The baseline local nonlinear solver consists of a 3-stage line-preconditioned Runge-Kutta scheme designed for good error smoothing properties.[20,21] This corresponds to the nonlinear update scheme depicted in equation (25) where[P] is a piecewise block-tridiagonal matrix, as determined by sets of lines constructed in the mesh using a graph algorithm. The line structures are used to relieve the stiffness associated with high mesh stretching in near-wall regions. The lines are constructed based on an algorithm that is initiated in regions of high mesh anisotropy and proceeds in a greedy fashion towards regions of lower anisotropy, with lines terminating when nearly isotropic mesh elements are encountered.[20,21,25] This graph algorithm results in a set of lines of varying length that do not span the entire mesh, as illustrated in Figure 1. For each identified edge joining points ij in the line set, the corresponding first-order accurate Jacobian off-diagonal block matrix entries $[O_{ij}]$ and $[O_{ji}]$ are retained in the [P] matrix, while all other diagonal entries are dropped. The mesh vertices and edges are then reordered resulting in a local tridiagonal Jacobian matrix structure for each line. The implementation naturally handles lines of varying length, where lines of length 0 (1 vertex, zero edges) reduce to a block-diagonal preconditioning approach. In the current implementation, the local tridiagonal matrices are factorized and frozen for all stages of the Runge-Kutta solver. This line-preconditioned RK scheme can either be used directly as a nonlinear solver, or as a smoother for a non-linear FAS multigrid scheme based on coarse agglomerated meshes.[20,21]

Alternatively, a pseudo-transient continuation (PTC) Newton-Krylov method can be used to solve the same problem. In this case, a right-preconditioned GMRES algorithm is used to solve the linear system described by equation (6) using exact Jacobian-vector products. Preconditioning is achieved by approximately inverting the equivalent first-order Jacobian matrix using a small number of linear multigrid iterations driven on each level by a linear block-tridiagonal line solver which is analogous to the nonlinear variant described previously.

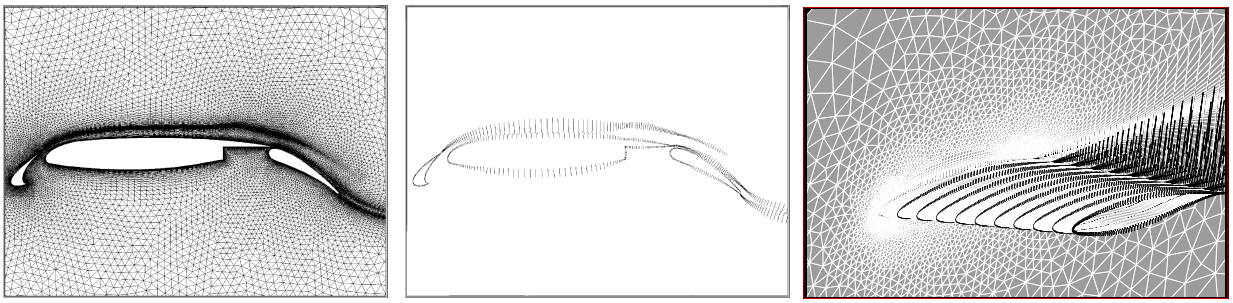
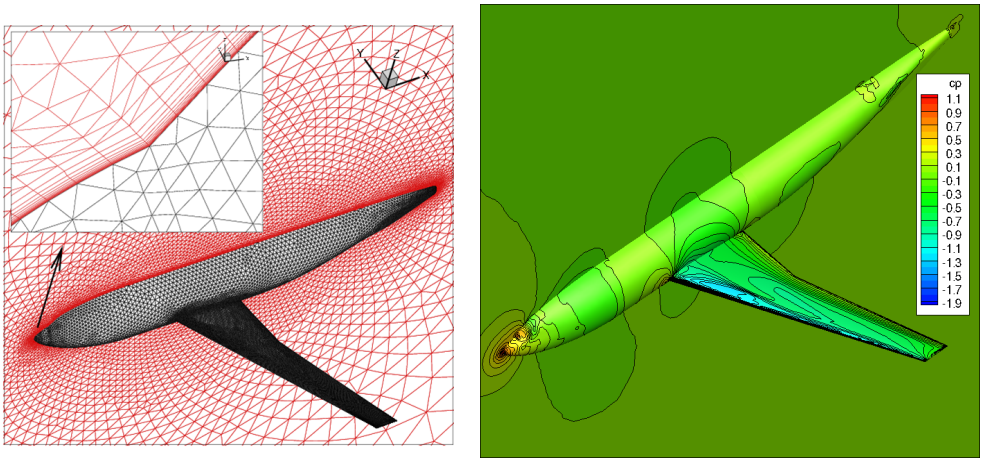


Figure 1. Illustration of implicit line structures extracted from unstructured meshes in two and three dimensions

The considered test case consists of the steady transonic flow over an aircraft wing-body geometry which has been the subject of previous drag-prediction-workshop (DPW) accuracy studies[26,27] as well as convergence efficiency studies.[6] An unstructured mesh of 1.2 million vertices is used for this case, as shown in Figure 2(a) where the highly anisotropic prismatic elements in the near wall regions are depicted. The aspect ratio of these elements can be of the order of 10^4. The freestream flow conditions include a Mach number of 0.75, a flow incidence of 0 degrees, and a Reynolds number based on the mean aerodynamic chord of the wing of 3 million. The steady-state solution produced by all the solution techniques is illustrated in Figure 2(b), where a weak shock wave on the wing is observed. This is considered a relatively well behaved (easy to converge) case by industrial standards.

For all solvers, the flow field is initialized impulsively as a uniform freestream field that does not satisfy the wall boundary conditions. Figure 3 depicts the convergence history achieved by the nonlinear line-preconditioned RK solver used alone on the fine mesh, as well as that achieved using this solver as a smoother for the nonlinear multigrid scheme with 4 coarser agglomerated mesh levels. Convergence is monitored as a function of the L2 norm of the total residual and the history of the computed lift coefficient, which represents a global integrated quantity of engineering interest. In both cases, a near monotonic residual convergence is observed, with the multigrid scheme achieving significantly faster overall convergence, as expected.

Figure 4 depicts the convergence history achieved by the PTC Newton-Krylov solver. In this case, a



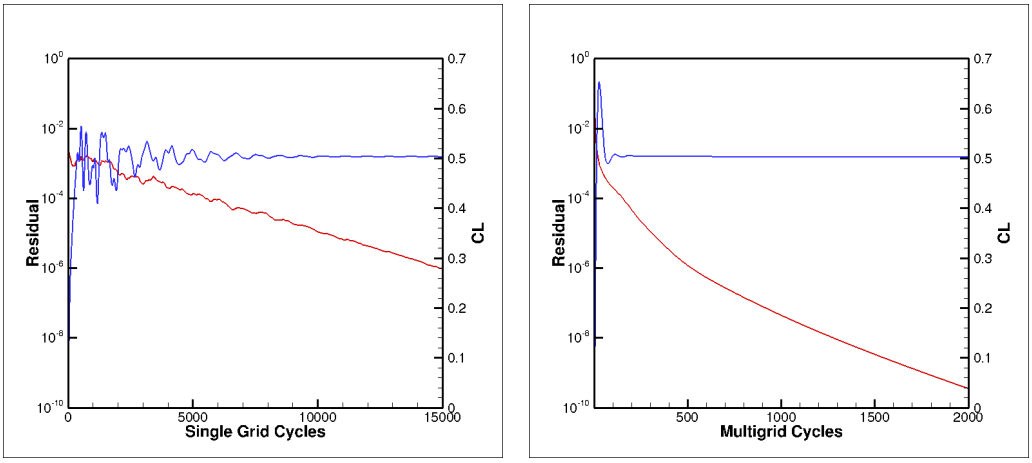
(a) Unstructured mesh with near wall detail

(b) Steady state solution in terms of computed surface pressure contours

Figure 2. Illustration of steady-state wing-body CFD test case

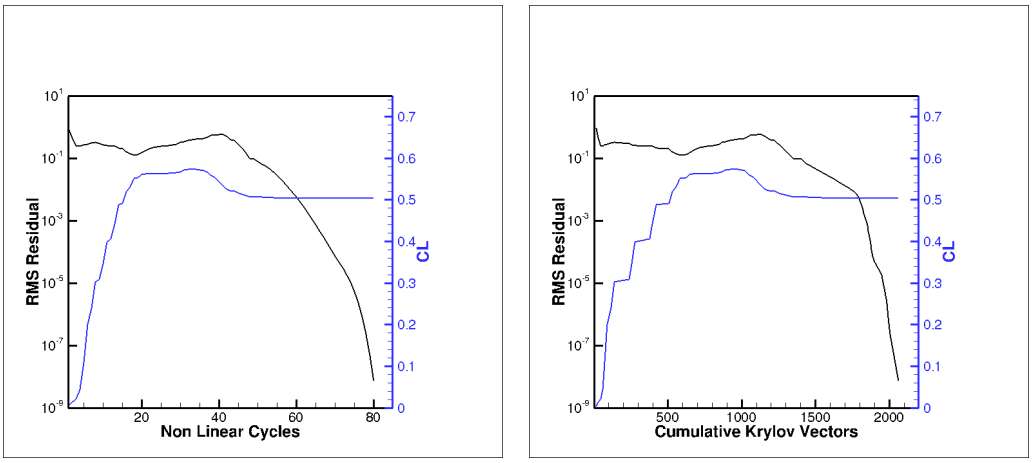
linear system convergence tolerance of two orders of magnitude reduction in the linear residual is requested and a maximum of 100 Krylov vectors are allowed. If the requested convergence tolerance is not met within the maximum number of Krylov vectors, the linear solver exits and records a failure, which prompts the PTC controller to reject the nonlinear update and lower the pseudo-time step by the factor β_{CFL2}. Each Krylov vector is preconditioned using 3 linear multigrid cycles with 4 line-smoother passes on each level. The computational cost of a Krylov vector is approximately equal to the cost of a single nonlinear multigrid cycle or two nonlinear single grid cycles in the previous case. The PTC controller uses an initial pseudo-time step value of CFL=10, and increases or reduces the CFL number depending on the success of the line search operation based on the amplification factors $\beta_{CFL1} = 1.5$ and $\beta_{CFL2} = 0.1$, respectively. From Figure 4(a), the residual is seen to decrease by 8 orders of magnitude over 80 nonlinear cycles (Newton steps). However, for the first 40 or more cycles, the residual does not decrease substantially from its initial value, and only begins to decrease rapidly near the end of the calculation, as is typically observed for Newton methods. Figure 4(b) reproduces the same convergence histories plotted in terms of cumulative Krylov vectors, which is a better representation of computational wall-clock time, further illustrating how the slow initial convergence of the Newton scheme consumes more than half of the Krylov vectors or compute time. Additionally, the lift coefficient is also slow to converge initially, and only comes close to its final value in the quadratic convergence region of the Newton method. This is in contrast to the rapid initial convergence of this quantity in the nonlinear multigrid solver, as shown in Figure 3, which may seem surprising given the fact that the Newton-Krylov scheme uses an analogous linear multigrid solver as a preconditioner. We note that rapid partial convergence of engineering quantities such as integrated force coefficients can have important practical implications for production use of solvers.

In order to gain a better understanding of the Newton-Krylov convergence process, Figure 5 plots, in addition to the nonlinear residual, the number of Krylov vectors per Newton step and the CFL value as determined by the PTC controller. We first note that the initial value CFL=10 is too large for the impulsive initial condition, and is reduced by the PTC controller on the first step to CFL=1.0. Thereafter the CFL value rises slowly at first, but stagnates and remains relatively low over the first 50 nonlinear steps (out of 80), which corresponds to the first 1400 cumulative Krylov vectors (out of 2063), after which it resumes steady growth to large values that enable the final quadratic convergence. Another point to note is that the hardest linear systems to solve, i.e. those that require the largest number of Krylov vectors, are those at the beginning of the continuation process, when the CFL or pseudo-time step is small, and the Jacobian is thought to be strongly diagonally dominant. Evidently, the nonlinear state about which these Jacobians are linearized results in stiff linear problems, particularly compared to the final state where quadratic nonlinear convergence occurs. In this sense, the pseudo-transient continuation approach is doubly inefficient. On the one hand it is slow to grow the CFL value to produce a more strongly implicit scheme, while on the other hand it results in linear problems early in the continuation process that are more difficult and costly to solve than even the final Newton step.



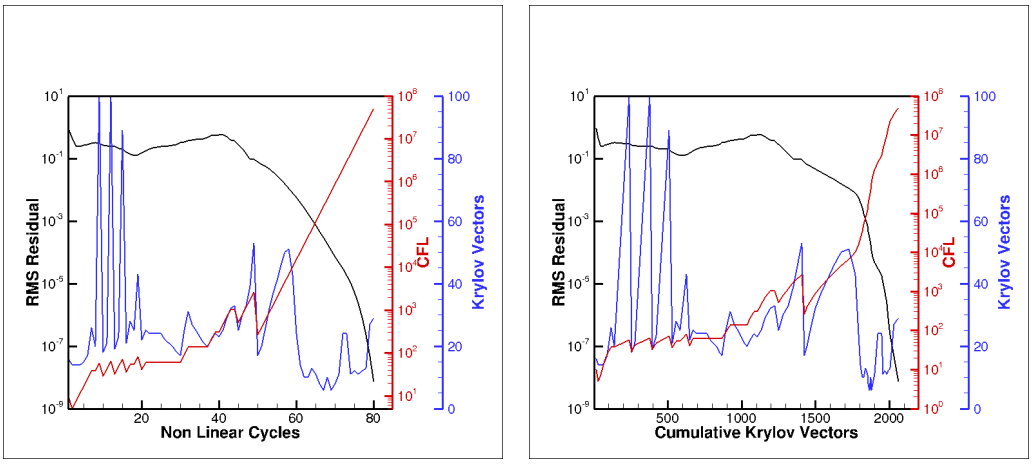
(a) Single grid solver (b) Multigrid solver

Figure 3. Convergence history of (a) single grid and (b) multigrid nonlinear solvers. Note difference in horizontal scales



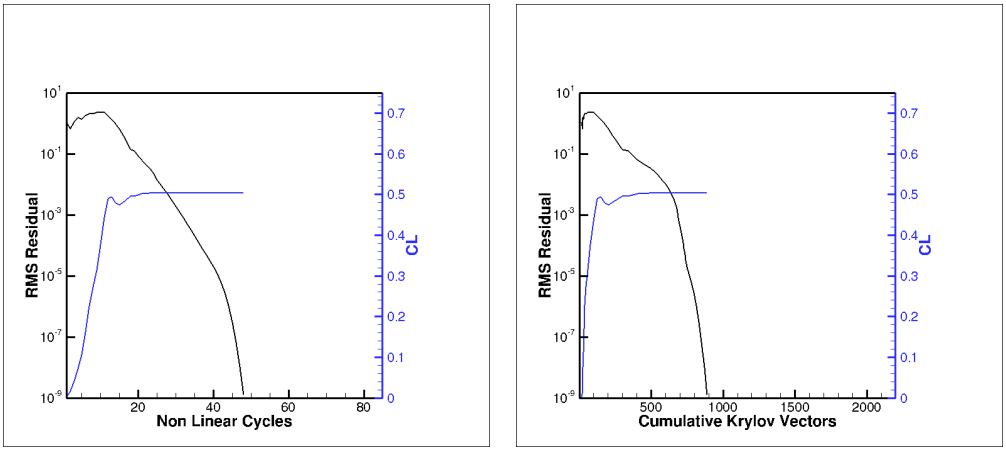
(a) (b)

Figure 4. Convergence history for Newton-Krylov solver (unsmoothed)



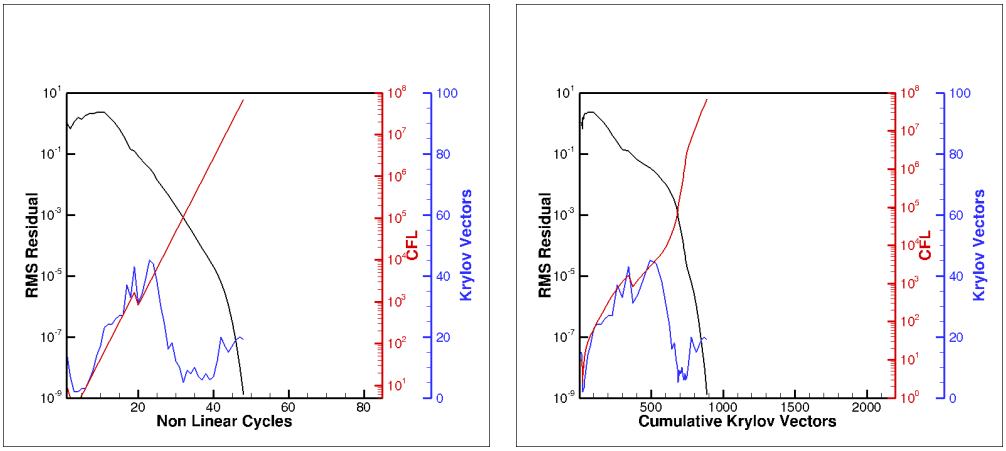
(a) (b)

Figure 5. Convergence history details of pseudo-transient continuation process for Newton-Krylov solver (unsmoothed)



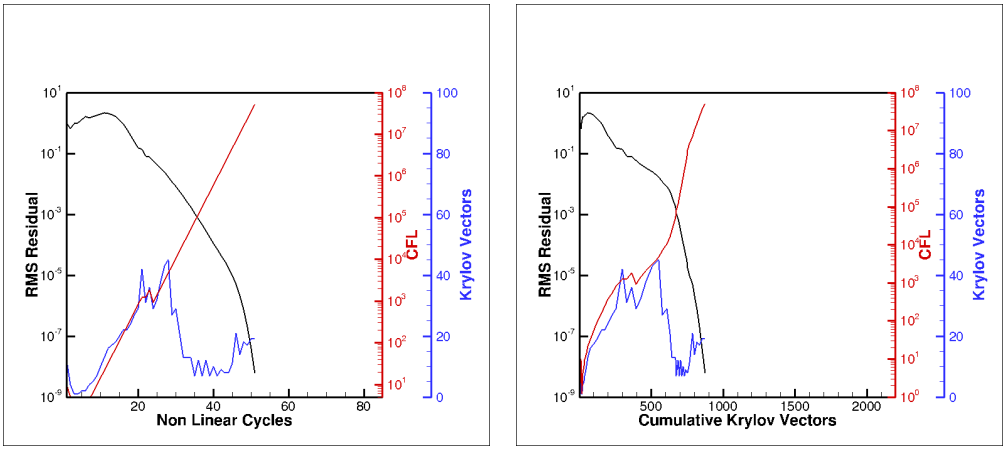
(a) (b)

Figure 6. Convergence history for residual-smoothing Newton-Krylov solver (single grid RK smoother)



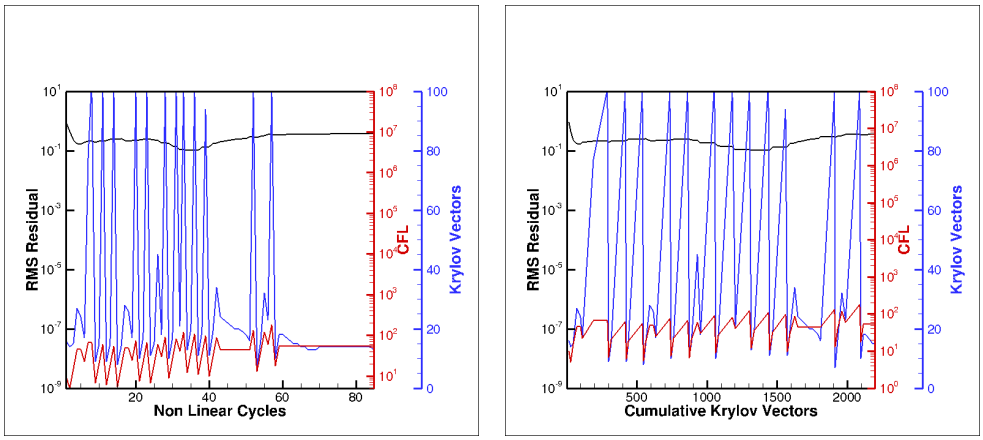
(a) (b)

Figure 7. Convergence history details of pseudo-transient continuation for smoothed Newton-Krylov solver (single grid RK smoother)



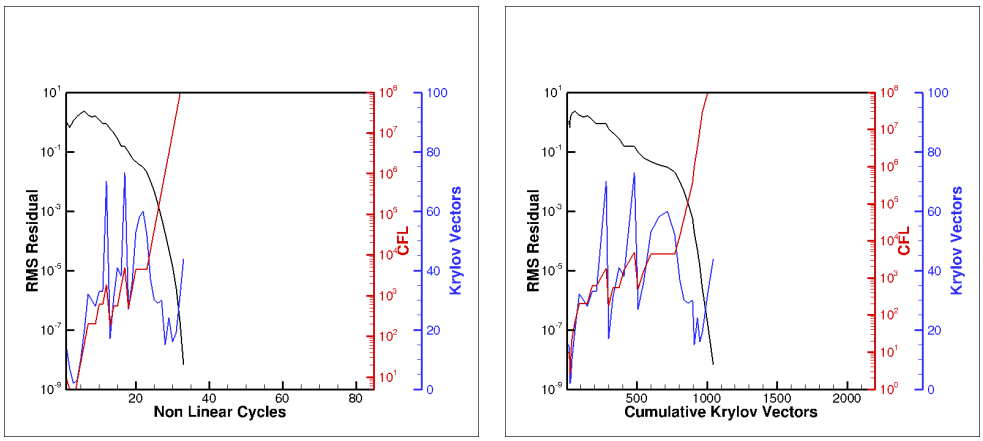
(a) (b)

Figure 8. Convergence history details of pseudo-transient continuation for smoothed Newton-Krylov solver using multigrid smoother



(a) (b)

Figure 9. Convergence history details of pseudo-transient continuation for unsmoothed Newton-Krylov solver using more aggressive CFL growth factor $\beta_{CFL1} = 3$ (single grid RK smoother)



(a) (b)

Figure 10. Convergence history details of pseudo-transient continuation for smoothed Newton-Krylov solver using more aggressive CFL growth factor $\beta_{CFL1} = 3$ (single grid RK smoother)

Figure 6 depicts the convergence history achieved by the residual smoothing approach applied to the PTC Newton-Krylov solver. In this case, 5 nonlinear cycles of the 3-stage line-preconditioned RK scheme are used to generate the smoothing source term, which is added to the right-hand side as given in equation (21), with all other Newton-Krylov solver parameters remaining identical to those used in the previous case. As the figure illustrates, the number of nonlinear cycles required to achieve the same residual reduction of 8 orders of magnitude is reduced from 80 to 48. Perhaps more importantly, the number of cumulative Krylov vectors is reduced from 2063 to 888. Given that the overhead of the residual smoothing process accounts for roughly 10% of the total solver time, the overall gain is roughly a factor of 2. Furthermore, the initial convergence of the lift coefficient is more rapid, particularly in terms of the number of cumulative Krylov vectors, displaying convergence behavior which is closer to that observed in Figure 3 using the nonlinear multigrid approach. Figure 7 depicts additional details of the solution process for this case. As in the previous case, the initial value CFL=10 is determined by the line search to be too large for the impulsive initial condition, and is reduced to CFL=1 at the first step. However, the CFL value grows almost monotonically thereafter, quickly reaching large values and enabling earlier onset of quadratic nonlinear convergence. At the same time, the number of Krylov vectors required to solve each linear system is rarely more than double the required number at the final state, presumably due to smoother nonlinear states achieved throughout the continuation process.

In Figure 8 the residual smoothing approach is repeated, although the residual smoothing term is now generated using 5 nonlinear multigrid cycles (using the 3-stage line-preconditioned RK scheme as a smoother on each level). As demonstrated in Figure 3, the nonlinear multigrid solver is known to be a much more

effective nonlinear solver than the single grid line-RK smoother alone. However, from Figure 8 it is seen that the convergence behavior using the multigrid scheme in the place of the single grid nonlinear solver for residual smoothing purposes does not improve the overall convergence behavior. In this case, the equivalent 8 order of magnitude residual drop is achieved in 51 nonlinear cycles and 874 cumulative Krylov vectors, compared to 48 and 888 respectively for the previous case. This case provides evidence that the effective mechanism of the proposed approach is indeed smoothing rather than simply advancing the solution nonlinearly through the application of additional nonlinear solver steps, as would be the case if a solver switching strategy were to be used.

Returning to the original residual smoothing results plotted in Figure 7, since the CFL grows almost monotonically, it can be argued that a faster CFL growth rate may lead to faster nonlinear convergence. In Figures 9 and 10 the original unsmoothed and the smoothed Newton-Krylov solvers discussed in Figures 5 and 7 are rerun using a larger CFL growth factor of $\beta_{CFL1} = 3$. As seen in Figure 9, the unsmoothed Newton-Krylov solver fails to converge with this more aggressive CFL growth factor, entering a limit cycle leading to repeated reduction in the CFL values, followed by resumed growth. On the other hand, the smoothed approach achieves convergence (defined as 8 orders of magnitude residual reduction) in 33 nonlinear cycles corresponding to 1044 cumulative Krylov vectors. In spite of the fast CFL growth rate and resulting lower number of nonlinear cycles, the total number of Krylov vectors is slightly higher and the overall efficiency is slightly lower than the convergence discussed previously and shown in Figures 6 and 7. On the one hand, this test case demonstrates the ability of the smoothed approach to overcome convergence stalling that occurs for the nonsmoothed approach, thus providing an additional level of robustness. However, this case also reveals the delicate balance that must be achieved between local smoothing and the continuation process in order to maximize overall solution efficiency.

IV. Application to finite-element discretizations

In this section, the residual smoothing strategy is applied to the solution of finite-element discretizations including Streamwise Upwind Petrov Galerkin (SUPG) and discontinuous Galerkin (DG) discretizations at second (p=1) and third (p=2) orders of accuracy. As previously, all problems considered solve the steady-state RANS equations using the Spalart-Allmaras turbulence model. Although finite-element discretizations have proven accuracy advantages over more traditional finite-volume methods in CFD, particularly at higher order, these methods most often result in stiff nonlinear systems of equations that can be difficult to solve efficiently and robustly. For this reason, the use of Newton-Krylov methods has become widespread for the solution of SUPG and DG discretizations, particularly when applied to steady-state RANS problems. However, as discussed previously, these solution techniques are much more costly in terms of cpu time and memory footprint than the simpler nonlinear iterative methods often used for finite-volume schemes, and improving the efficiency of solvers for these discretizations is critical for enabling their adoption in applied CFD for industrial problems.

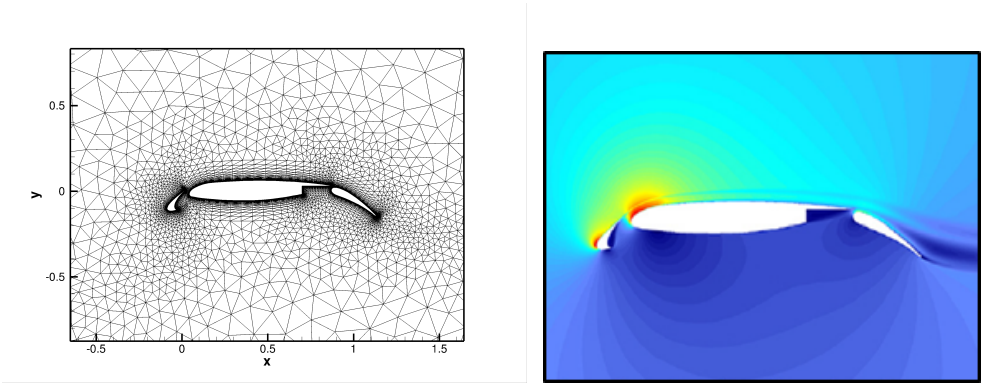
One of the principal difficulties in applying the residual smoothing strategy to SUPG and DG discretizations has been the ability to construct stable local nonlinear iterative methods for these discretizations. In general, the Jacobian matrices that arise from the linearization of finite-element based residual operators are significantly less diagonally dominant than those encountered in traditional finite-volume based schemes, and direct application of commonly used local nonlinear smoothers, such as block or line Jacobi or Gauss-Seidel, has been found to be unstable. In order to construct a stable local nonlinear iterative scheme, the local operator D in equations (21) and/or (22) is augmented with a second pseudo-time step as

$$D_{stable} = \frac{M}{CFL_{local}\Delta\tau} + D \tag{26}$$

where M is a suitable mass matrix and $\Delta\tau$ is an estimate of the local explicit time step, as previously, and CFL_{local} is another parameter which is either constant or set as

$$CFL_{local} = min(CFL_{NK}, CFL_{local_{max}}) \tag{27}$$

Here $CFL_{local_{max}}$ is a value which limits the growth of the CFL_{local} to values which ensure stability of the nonlinear iterative scheme. These values are usually much smaller than CFL_{NK} which tends to infinity as the Newton solver approaches quadratic convergence. In addition to the use of a local CFL number, the

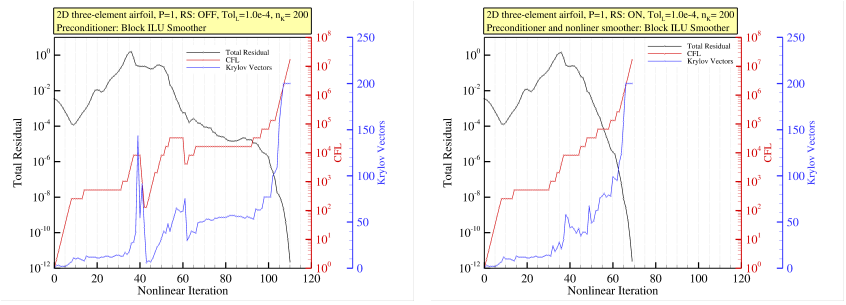


(a) Computational mesh (b) Computed flow field

Figure 11. Computational mesh and computed flow field for 2D steady-state high lift case using p=1 SUPG discretization

individual local nonlinear updates produced in the smoothing scheme can be under-relaxed to further aid with overall stability.

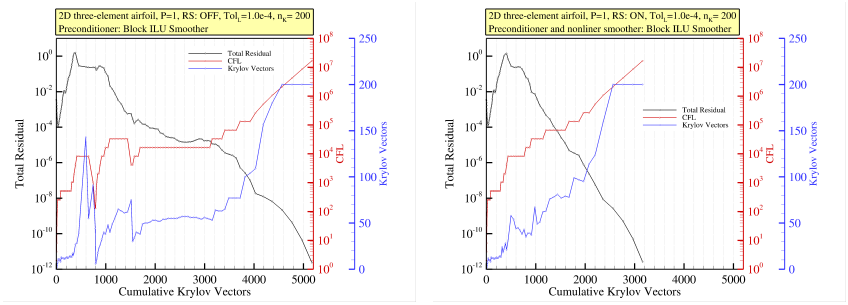
Figures 11 through 13 illustrate results achieved using a second-order accurate (p=1) SUPG discretization applied to a two-dimensional high-lift problem. This case consists of the flow over a three-element airfoil with a freestream Mach number of 0.2, an incidence of 16^o, and a Reynolds number of 9 million. The computational mesh and computed flowfield in terms of Mach number contours are displayed in Figure 11. For this case, the baseline Newton-Krylov scheme employs a block ILU preconditioner[28] and solves the linear system at each Newton step to a relative linear residual L2-norm of 1.E-04. For the residual smoothing case, the smoothing operator was constructed using a single sweep of the block ILU preconditioner used as a nonlinear solver, with $CFL_{local_{max}} = 250$ and an under-relaxation factor of 0.2. The convergence plots in Figure 12 indicate that the application of residual smoothing in this case reduces the total number of nonlinear cycles in the Newton method from 110 to 70. Furthermore, the jumps in the number of Krylov vectors needed for the solution of linear systems in the continuation region of the unsmoothed case are eliminated in the smoothed case. Overall, application of residual smoothing reduces the total number of Krylov vectors required from 5168 to 3162, as shown in Figure 13.



(a) Original Convergence (b) Residual smoothed convergence

Figure 12. Comparison of convergence for baseline and residual smoothed Newton scheme for p=1 SUPG discretization of 2D steady-state high lift case in terms of nonlinear cycles

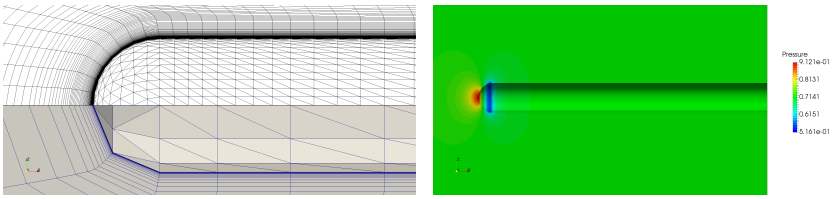
The next test case illustrates the use of residual smoothing for p=1 and p=2 DG discretizations. In this case we consider the flow over a hemispherical cylinder, which is a standard test case used for verification purposes of high-order CFD methods.[4,29] The flow conditions include a farfield Mach number of 0.6, a flow incidence of 0^o, and a Reynolds number of 350,000 based on the cylinder diameter. A relatively coarse grid of 4320 elements was used, with appropriately curved elements for the p=2 DG discretization, as shown in Figure 14. In this case, a p=1 solution was first computed on this mesh and used to initialize the Newton scheme for the solution of the p=2 discretization. The Newton solver was configured to request a relative reduction in the linear residual of 1.e-03 at each Newton step, and used 50 sweeps of a block line solver as preconditioner, where lines were constructed normal to the wall and the block size corresponds to all



(a) Original Convergence (b) Residual smoothed convergence

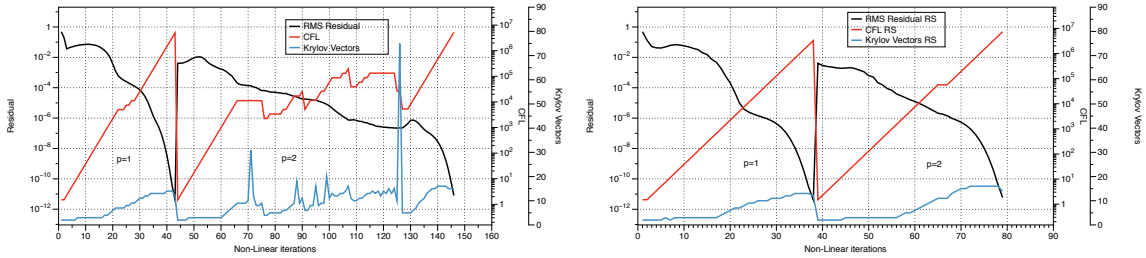
Figure 13. Comparison of convergence for baseline and residual smoothed Newton scheme for p=1 SUPG discretization of 2D steady-state high lift case in terms of cumulative Krylov vectors

degrees of freedom within a mesh cell for both p=1 and p=2 discretizations. The nonlinear smoother was constructed using 30 nonlinear sweeps of the same block line solver with the addition of a local CFL value set as $CFL_{local} = 1000$. Both the preconditioner and the nonlinear smoother employed an under-relaxation factor of 0.7. Convergence histories for both p=1 and p=2 discretizations are shown in Figure 15 in terms of nonlinear cycles for both the unsmoothed and residual-smoothed Newton solvers. In both cases, the p=1 convergence history is relatively well behaved and the application of residual smoothing only shows modest improvement, reducing the number of nonlinear cycles from 42 to 38, and the number of cumulative Krylov vectors from 300 to 250. However, the p=2 discretization proves to be much more difficult to converge for the unsmoothed solver, with spikes in the number of Krylov vectors occurring near nonlinear iterations 70 and 125. By contrast, the convergence of the residual-smoothed Newton solver is much more well behaved with no significant jumps in the required number of Krylov vectors. Overall, the total number of nonlinear cycles for this problem is reduced from 146 to 79, while the cumulative number of Krylov vectors used is reduced from 1320 to 570, as shown in Figure 16.



(a) Mesh (b) Flow solution

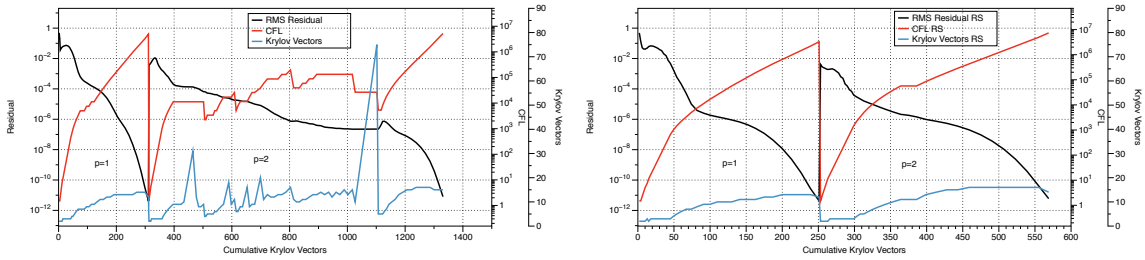
Figure 14. Coarse unstructured mesh showing linear and (subdivided) curved elements and p=2 DG flow solution for hemispherical test case



(a) Unsmoothed (b) Residual Smoothed

Figure 15. Comparison of convergence of p=1 and p=2 DG hemisphere test case for baseline (unsmoothed) and residual smoothed Newton scheme in terms of nonlinear cycles

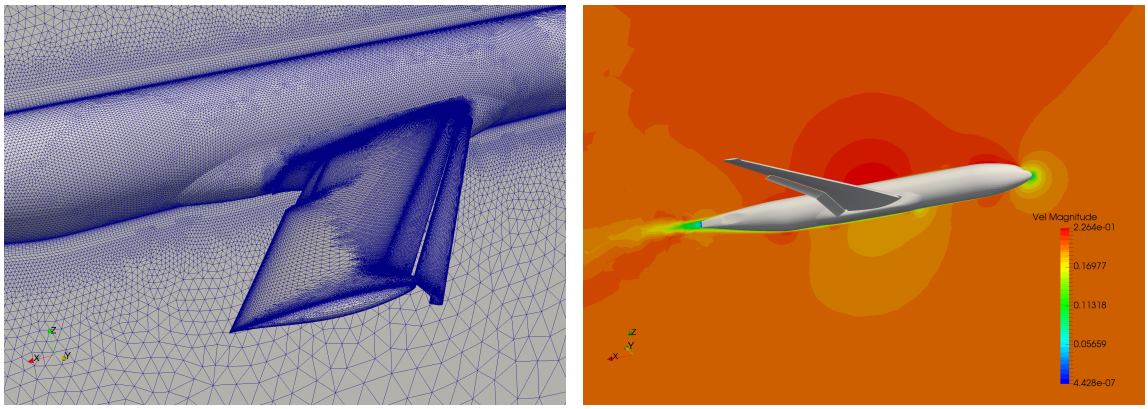
A more demanding test case is considered in Figure 17 where a p=1 DG discretization is used to compute the flow over a 3D high-lift configuration based on the high-lift CRM geometry.[30] A mixed-element mesh consisting principally of prisms and tetrahedra with a total of 2,265,022 elements is used, with no curved elements. The lack of curving in the mesh for p=1 discretizations is known to degrade the accuracy of the



(a) Unsmoothed (b) Residual Smoothed

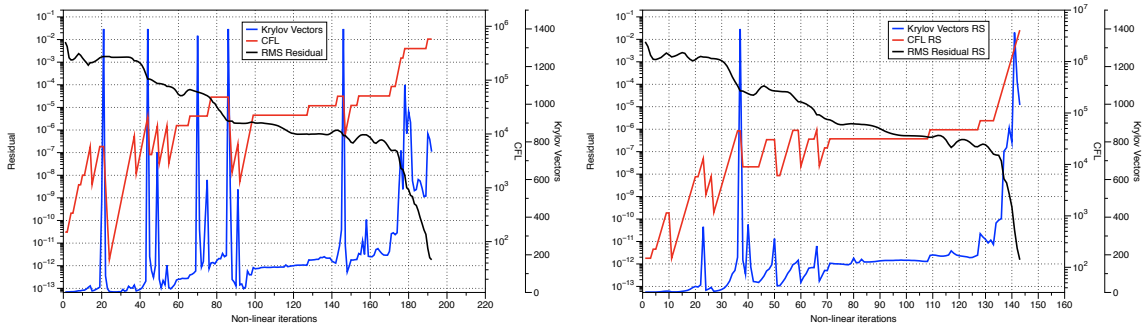
Figure 16. Comparison of convergence of p=1 and p=2 DG hemisphere test case for baseline (unsmoothed) and residual smoothed Newton scheme in terms of cumulative Krylov vectors

solution and to increase the difficulty of obtaining a full converged steady-state solution, providing a good test for solver robustness. The flow conditions include a far field Mach number of 0.2, a flow incidence of 8^o and a Reynolds number of 3.25 million. The Newton solver used a linear system tolerance of 2.e-03, and the preconditioner was constructed using 50 sweeps of a (cell-wise) block Jacobi algorithm with under-relaxation of 0.7. The nonlinear residual smoothing operator was constructed using 10 nonlinear cycles of the block Jacobi algorithm with a local CFL limited as $CFL_{local_{max}} = 100$ and an under-relaxation factor of 0.5. However, in this case, the nonlinear residual was also updated between each residual-smoothed Newton update using the nonlinear smoothing algorithm alone. As shown in Figure 18, the number of Newton steps is reduced from 191 to 142, while the number of cumulative Krylov vectors is reduced from 42,000 to 26,000 as seen in Figure 19.



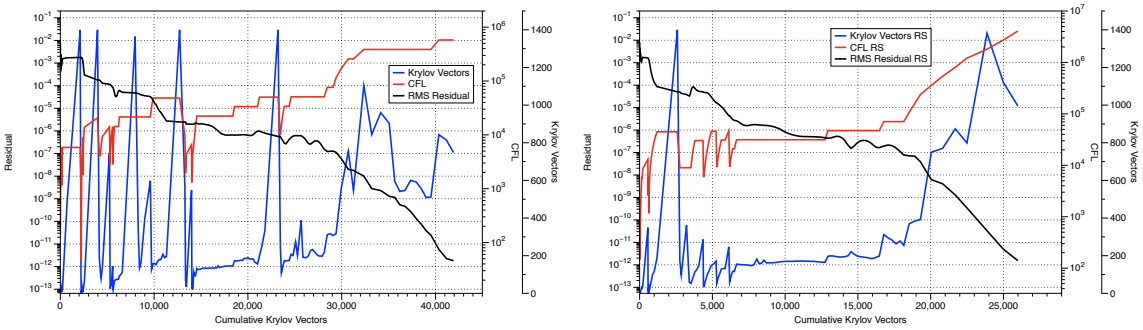
(a) (b)

Figure 17. Steady-state p=1 DG solution for High-Lift CRM test case



(a) Unsmoothed (b) Residual smoothed

Figure 18. Comparison of convergence of p=1 DG high lift test case for baseline (unsmoothed) and residual smoothed Newton scheme in terms of nonlinear cycles



(a) Unsmoothed (b) Residual smoothed

Figure 19. Comparison of convergence of p=1 DG high lift test case for baseline (unsmoothed) and residual smoothed Newton scheme in terms of cumulative Krylov vectors

V. Nonlinear Two-Level Multigrid Approach

Another strategy for accelerating Newton method continuation is to form and solve a surrogate nonlinear problem which is easier to solve than the original nonlinear problem. For example Newton solver continuation may be accelerated by initializing the Newton solver with a solution obtained on a coarser mesh. However, this strategy can only be applied as an initial condition and cannot be used to further accelerate the Newton solver after initialization. Alternatively, a coarse mesh nonlinear problem which is consistent with the fine mesh solution can be constructed using the standard full-approximation multigrid (FAS) approach.[31] The formulation of a consistent coarse level problem enables the use of repeated visits to the coarse level to accelerate the nonlinear progression of the fine level Newton scheme at any point in the convergence history. Following the FAS multigrid formulation, the fine level problem is written as:

$$L_h u_{exact} = f_h \tag{28}$$

where L_h represents the nonlinear equations on the fine level denoted by subscript h, u_{exact} refers to the exact solution and f_h is a source term which vanishes for our case on the fine level. For an approximate solution on the fine level u_h we can write the non-zero fine level residual operator as:

$$R_h(u_h) = L_h u_h - f_h \tag{29}$$

A consistent coarse level problem can then be constructed as

$$L_H u_H - L_H \tilde{u}_H = -I_h^H R_h(u_h) \tag{30}$$

where L_H denotes the nonlinear discretization equations on the coarse level, $\tilde{u}_H$ represents an approximate solution on the coarse level, obtained for example by interpolating the fine level solution onto the coarse level, and $I_h^H R_h(u_h)$ represents the fine level residuals restricted to the coarse level. Alternatively, the coarse level problem may be written as:

$$L_H u_H = \tau_H \tag{31}$$

$$\tau_H = L_H \tilde{u}_H - I_h^H R_h(u_h) \tag{32}$$

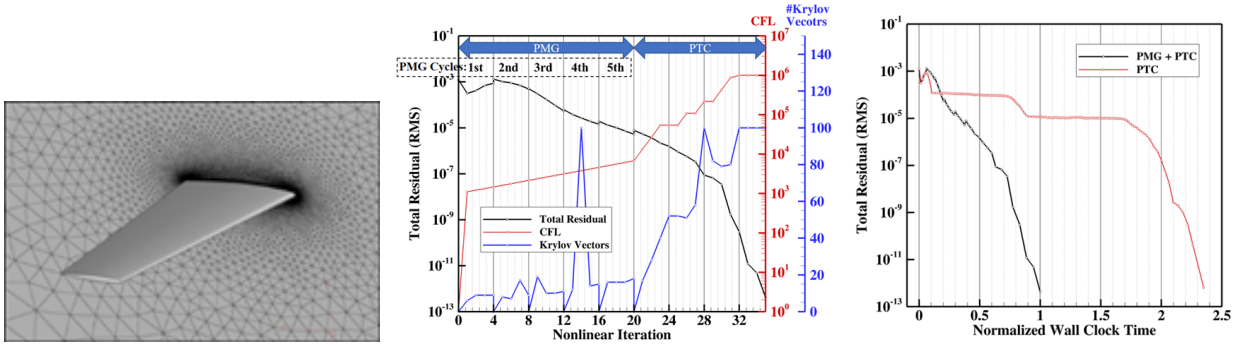
where τ_H is constant on the coarse level since it is a function only of fine level quantities. After solution of equation (31), the fine level solution is updated as:

$$u_h = u_h + \alpha I_H^h (u_H - \tilde{u}_H) \tag{33}$$

where I_H^h represents the operator which interpolates or prolongates the corrections $u_H - \tilde{u}_H$ back to the fine level and $0 \leq \alpha \leq 1$ is an optional parameter which may be controlled by the line search. This coarse level formulation is consistent with the fine level problem in that, when the fine level residuals vanish, the corrections supplied by the coarse level also vanish, as can be seen from equation (30). Applied recursively on a sequence of coarser meshes, nonlinear multigrid has been demonstrated successfully for a wide variety

of CFD problems. However, nonlinear multigrid solvers lack the robustness attributes of Newton solvers and often fail to converge to small tolerance levels for more difficult test cases and higher-order discretizations.[6] The approach taken in this work is to augment a fine level Newton-Krylov solver with a small number of visits to a single coarser level (i.e. a 2 level method), where the problem is solved to some prescribed tolerance, and the updates are used to accelerate the continuation phase of the PTC Newton method.

In Figure 20 this approach is applied to a p=2 SUPG discretization for flow over an ONERA M6 wing at Mach = 0.2, 0 degrees incidence, and Reynolds number of 5 million. In this case, the coarse level consists of the corresponding p=1 discretization on the same mesh, and thus no coarse physical mesh is required. The fine level Newton scheme is initialized with a precomputed p=1 solution, and a total of 5 coarse (p=1) level solutions are performed during the PTC Newton continuation process, after which the solution procedure resumes with the standard PTC Newton method. The coarse level solutions are obtained using the analogous Newton method on that level, and the total cost of the coarse level visits is a small fraction of the overall solution cost due to the fact that the p=1 discretization contains 8 times fewer degrees of freedom compared to the fine p=2 level. The comparison of convergence histories for the baseline Newton scheme and the two-level multigrid-enhanced Newton scheme shows a more rapid nonlinear convergence in the initial phases of the solution process and an overall savings in terms of cpu time of approximately 2.



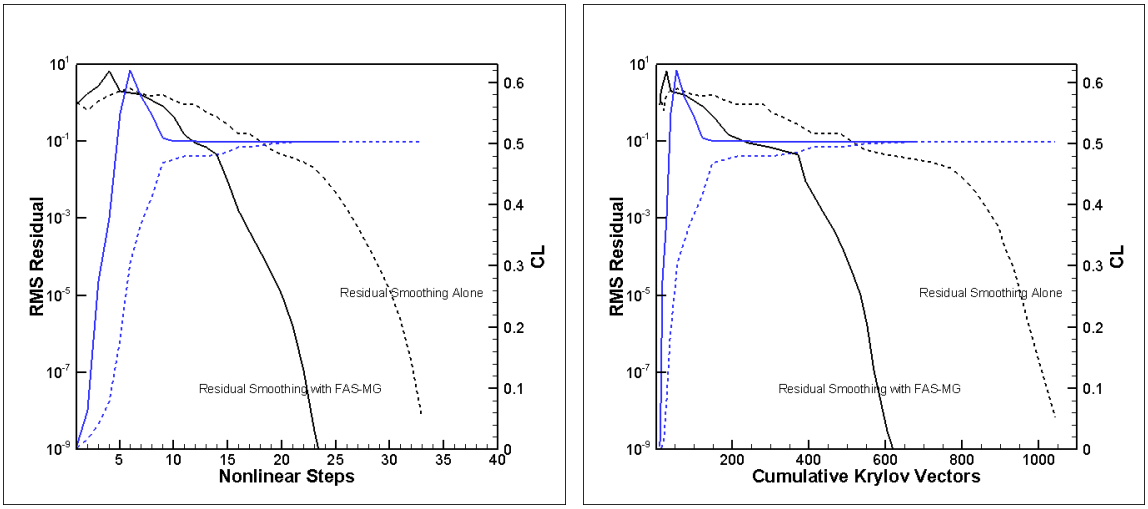
(a) (b) (c)

Figure 20. Illustration of nonlinear convergence acceleration for SUPG p=1 discretization of flow over ONERA M6 wing using 5 sweeps of 2 level FAS multigrid combined with PTC Newton scheme

This nonlinear multigrid approach is complementary to the residual smoothing approach discussed previously and the two methods can be used simultaneously. In Figure 21, we return to the transonic wing-body test case computed using the NSU3D finite-volume discretization and apply the two-level multigrid approach to the residual-smoothed PTC Newton scheme discussed in Section III. In this case, the coarse level is constructed using an agglomeration process which is incorporated into the NSU3D solver[6] and the coarse level problem is discretized only to first-order accuracy, in contrast to the second-order accurate discretization used on the fine level. Visits to the coarse level are performed after each Newton update of the residual-smoothed PTC Newton-Krylov solver on the fine level, and the coarse level problem is solved approximately using the recursive multigrid algorithm available in the NSU3D code. As shown in the convergence history plots, the number of fine level Newton steps is reduced from 34 to 23 and the number of cumulative Krylov vectors requires is reduced from 1050 to 620. Precise wall clock time comparisons are not made due to the fact that the coarse level solver has not been optimized. Nevertheless, the combined effect of residual smoothing and nonlinear multigrid provides over a factor of 3.5 reduction in the number of cumulative Krylov vectors required for the solution of this problem, showing that the benefits of both approaches can be multiplicative.

VI. Conclusions and Future Work

In this work we have proposed two complementary strategies for addressing the problem of slow initial convergence in pseudo-transient continuation Newton methods. The first approach is based on residual smoothing, were we argue that, provided a smooth distribution of residuals can be maintained, the global nonlinear problem should be able to be advanced efficiently through strong nonlinear transients. Using a formulation that combines local nonlinear smoothers with a global Newton scheme, significant gains in efficiency of the overall nonlinear solution process have been demonstrated for realistic CFD problems.



(a) Nonlinear cycles

(b) Cumuative Krylov vectors

Figure 21. Illustration of nonlinear convergence acceleration for finite-volume (NSU3D) discretization using 2 level FAS multigrid combined with PTC Newton scheme for test case described in Figure 2

Similarly, using a two-level nonlinear multigrid approach applied judiciously during the continuation phase of the Newton scheme, efficiency gains of the order of a factor of 2 have been demonstrated. Although these methods have shown promise over a wide range of discretizations and test cases, significant challenges remain in the construction of optimal robust nonlinear solvers for CFD problems. The nonlinear smoothers employed in the construction of the residual smoothing operator must be stable and convergent, properties which may be difficult to guarantee for all cases. The nonlinear multigrid-augmented Newton solver, as currently formulated, does not revert to the baseline Newton scheme in the final stages of convergence when the CFL number tends towards infinity. This may cause loss of quadratic convergence, or worse divergence. Although the multigrid updates can be turned off after a period of initial convergence, a more elegant approach for recovering the exact Newton scheme in the final phases of convergence is desirable. Future work will focus on improving both approaches separately and on combining these techniques to provide larger overall gains in efficiency for Newton solvers.

VII. Acknowledgments

This work was partially funded by NASA grant NNX15AU23A under the Tranformational Tools and Technologies (T^3) project and Sandia National Laboratory contract 1852733. 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).

References

[1]Diosady, L. T. and Darmofal, D. L., "Preconditioning methods for discontinuous Galerkin solutions of the Navier–Stokes equations," *Journal of Computational Physics*, Vol. 228, No. 11, 2009, pp. 3917–3935.

[2]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.

[3]Burgess, N. K., Glasby, R. S., Erwin, J. T., Stefanski, D. L., and Allmaras, S. R., "Finite-element solutions to the Reynolds Averaged Navier-Stokes equations using a Spalart-Allmaras Turbulence Model," *AIAA Paper 2017-1224, Presented at the 56th AIAA Aerospace Sciences Meeting, Grapevine TX*, Jan. 2017.

[4]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.

[5]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.

[6]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.

[7]Kamenetskiy, D., Bussoletti, J., Hilmes, C., Johnson, F., Venkatakrishnan, V., and Wigton, L., "Numerical Evidence of

Multiple Solutions for the Reynolds-Averaged Navier-Stokes Equations for High-Lift Configurations," AIAA Paper 2013-663 presented at the 51st AIAA Aerospace Sciences Meeting, Grapevine, TX.

[8]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.

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

[10]Lanzkron, P. J., Rose, D. J., and Wilkes, J. T., "An analysis of approximate nonlinear elimination," *SIAM Journal on Scientific Computing*, Vol. 17, No. 2, 1996, pp. 538–559.

[11]Cai, X.-C. and Li, X., "Inexact Newton methods with restricted additive Schwarz based nonlinear elimination for problems with high local nonlinearity," *Siam journal on scientific computing*, Vol. 33, No. 2, 2011, pp. 746–762.

[12]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.

[13]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.

[14]Skogestad, J. O., Keilegavlen, E., and Nordbotten, J. M., "Domain decomposition strategies for nonlinear flow problems in porous media," *Journal of Computational Physics*, Vol. 234, 2013, pp. 439–451.

[15]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.

[16]Jameson, A., Schmidt, W., and Turkel, E., "Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes," AIAA Paper 81-1259.

[17]Jameson, A., "Solution of the Euler Equations by a Multigrid Method," *Applied Mathematics and Computation*, Vol. 13, 1983, pp. 327–356.

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

[19]Trottenberg, U., Schuller, A., and Oosterlee, C., *Multigrid*, Academic Press, London, UK, 2000.

[20]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.

[21]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.

[22]Spalart, P. R. and Allmaras, S. R., "A One-equation Turbulence Model for Aerodynamic Flows," *La Recherche Aérospatiale*, Vol. 1, 1994, pp. 5–21.

[23]Allmaras, S. R., Johnson, F. T., and Spalart, P. R., "Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence," *Proceeding of the 7th ICCFD International Conference on Computational Fluid Dynamics, Hawaii, HW*, July 2012.

[24]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.

[25]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.

[26]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.

[27]Lee-Rausch, E. M., Frink, N. T., Mavriplis, D. J., Rausch, R. D., and Milholen, W. E., "Transonic Drag Prediction on a DLR-F6 Transport Configuration using Unstructured Grid Solvers," *Computers and Fluids*, Vol. 38, No. 3, March 2009, pp. 511–532.

[28]Ahrabi, B. R. and Mavriplis, D. J., "An Implicit Block ILU Smoother for Preconditioning of Newton-Krylov Solvers with Application to Finite-Element Discretizations," *AIAA Paper Presented at the 58th AIAA Aerospace Sciences Meeting, San Diego,CA*, Jan. 2019.

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

[30]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," *54th AIAA Aerospace Sciences Meeting*, Jan. 2016, AIAA paper 2016-0308.

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