INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 0000; **00**:1–21
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme

Constrained pseudo-transient continuation for discontinuous finite element discretizations

Marco Ceze* and Krzysztof J. Fidkowski†
Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA

SUMMARY

This paper presents a method for improving the robustness of implicit steady-state solvers for fluid flow problems discretized with the discontinuous Galerkin finite element method. This method does not help in situations where no steady-state solution exists, for example due to an under-resolved discretization. However, an often overlooked situation is one in which the steady-state solution exists but cannot be reached by the solver, which typically fails due to a non-physical state error during state iterations. This is a shortcoming that we address by incorporating physical realizability constraints (i.e. positive density and pressure) into pseudo-time marching steady-state solvers. While we focus on the discontinuous Galerkin method, the technique relies only on implicit time marching and hence can be extended to other spatial discretizations. We analyze the sensitivity of the method to a range of input parameters and results for compressible turbulent flows show that the constrained method is significantly more robust than a standard unconstrained method while on par in terms of cost. Copyright © 0000 John Wiley & Sons, Ltd.

Received . . .

1. INTRODUCTION

Much recent research in computational fluid dynamics has focused on high-order methods, which have been loosely defined as those methods that have a formal rate of convergence higher than second order with respect to mesh size [1]. A high-order convergence rate is useful for many problems of engineering interest, especially ones requiring a level of accuracy that is not easily attained using traditional second-order discretizations. Even in the presence of non-smooth solution features, high-order approximations can often maintain their advantage when combined with solution-based adaptivity that isolates those features [2, 3, 4].

Under this backdrop, one may ask why high-order adaptive methods have not yet taken over as the preferred discretization for practical, large-scale, engineering applications. The answer to this question is certainly multi-faceted and in this work we do not propose a panacea. Instead, we focus on one specific but pervasive problem: solver robustness. High-order discretizations yield systems of nonlinear equations that are generally much more difficult to solve compared to those encountered when using, for example, second-order finite volume methods. Moreover, when used in combination with mesh adaptation, solutions are required on coarse, under-resolved initial meshes. In such cases, even if a zero-residual solution exists, it may be very difficult to attain that converged solution using existing solvers.

Current "standard" solvers for nonlinear systems arising from high-order discretizations consist of some form of preconditioned Newton-Krylov [5]. Initial states for these solves are generally rudimentary, such as a constant free-stream state for external aerodynamics, and hence continuation strategies are crucial. Here, a popular choice is *pseudo-transient* continuation (PTC) [6, 7, 8], in which artificial time dependence is added to even steady-state discretizations in order to make

the state follow a hopefully physically-valid trajectory from initial condition to final state. The hope is founded in the argument that an exact unsteady solution to any physically-valid initial condition should remain physical, so that capturing these transients is a safeguard against straying into possibly non-physical states – a common failure mechanism for highly-nonlinear problems. However, this argument assumes that the discretization is a good approximation to the exact solution, which may not be the case, especially for under-resolved coarse meshes in the initial stages of adaptation.

The conservation equations, once discretized, do not guarantee positivity for physical quantities such as pressure and density in a gas dynamics simulation. Hence, a time-accurate but spatially under-resolved solution can violate these constraints. To exemplify this phenomenon, consider a one-dimensional Euler shock-tube in the domain $x = [-1, 1]$, where the boundary conditions on both ends of the tube are flow in the positive x-direction at a Mach number of $M = 0.5$ (Figure 1). We specify the full boundary state in convenient units by setting the density and velocity to unity and by computing momentum and total energy accordingly. Riemann solves at the outer boundaries ensure that the problem is well-posed. Upon initializing the flow at $M = 0.747$ in the negative x-direction, a shock occurs on the left end of the domain and an expansion occurs on the right end. Eventually, if all goes well, the flow settles to a steady state equal to the boundary condition. In order to verify this discretely, we divide the tube into 10 equal elements and discretize the Euler equations spatially with the discontinuous Galerkin method. We then evolve the solution in time with a first-order backward difference scheme. In order to resolve the transients, we chose the time-step adaptively such that at each step the maximum value for the CFL number is 0.1. We employ a dual time-step approach to solve the nonlinear systems for each physical time-step. These solves include primitive variable checks at every nonlinear update. However, these checks can only verify physics constraint satisfaction at selected points in each element and the only mechanism to avoid them is limiting the magnitude of the state updates.

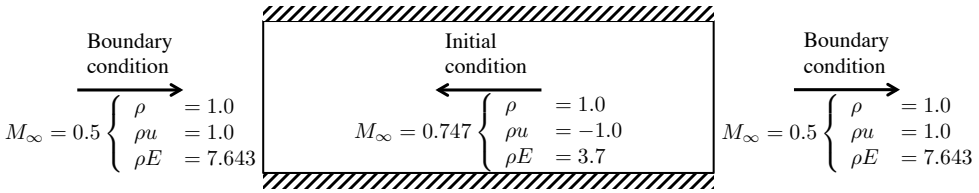
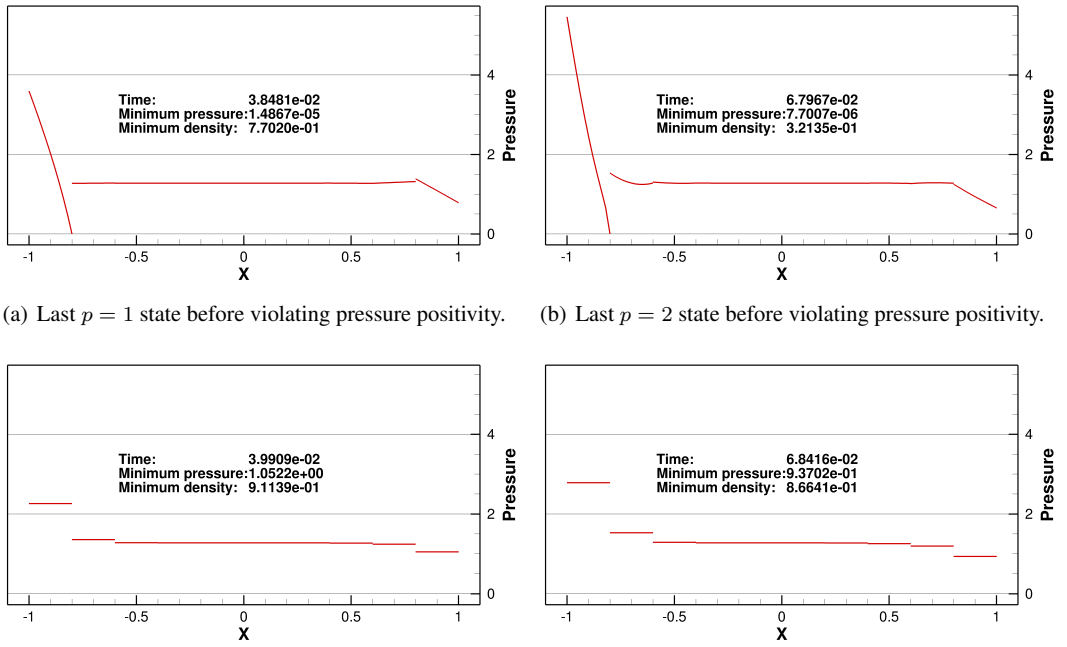


Figure 1. Setup for a shock-tube problem testing solver robustness.

For element-wise constant solution approximation ($p = 0$), the discrete solution reaches the expected steady state while for linear and quadratic ($p = 1, 2$) approximations, the state violates pressure positivity as the shock moves into the domain. Figures 2(a) and 2(b) show the pressure distributions just before violating pressure positivity and figures 2(c) and 2(d) show the pressure distributions for $p = 0$ at the approximate times of failure.

Under-resolved flow features such as shocks and boundary layers can cause oscillations in the numerical solution whose amplitudes may lead to non-physical values [9]. The use of well-established limiters in second-order finite-volume schemes substantially increases the robustness of these flow solvers. However, the application of limiters to high-order discretizations such as DG has had mixed success, with particular difficulties on unstructured adapted meshes. Problems include extension of the computational stencil, robustness for high-order, and lack of a zero-residual solution. An alternative remedy for oscillations is the introduction of artificial viscosity [10, 11], which addresses many shortcomings of limiting, but which is not without its own challenges, such as those concerning the amount of appropriate viscosity. Another alternative for problems without steady-state discontinuous features is a more sophisticated Newton continuation strategy, such as

DOI: 10.1002/nme



(a) Last $p = 1$ state before violating pressure positivity.

(b) Last $p = 2$ state before violating pressure positivity.

(c) $p = 0$ solution at approximately the time of failure for $p = 1$.

(d) $p = 0$ solution at approximately the time of failure for $p = 2$.

Figure 2. Pressure distributions for under-resolved, time-accurate shock-tube simulations.

parameter/order/boundary-condition sequencing [12, 13, 14, 15]. A challenge here is that for under-resolved simulations, runs with different parameters or lower orders may not be any easier to solve than the original problem [16].

In the present work we propose an alternative approach to improving solver robustness, one that is often less invasive relative to limiting or even artificial viscosity and that targets the root cause of robustness breakdown – the violation of physics constraints. Keyes *et al.* [17] point out that methods for handling these constraints are generally *ad hoc*. Here, we embed the constraints in the PTC solution path. The crux of the method is the incorporation of physics constraints in the form of residual penalties in the nonlinear system. These penalties are introduced multiplicatively so as to not modify the final steady-state solution. Instead, the penalties serve to "steer" the solution away from non-physical states during the pseudo-transient integration, so as to prevent the solver from stagnating at or near these problematic states. This constrained version of PTC, CPTC, thus artificially augments the inherent robustness of time-accurate integration that under-resolved discrete approximations do not necessarily inherit. The penalties provide a natural mechanism for communicating to the solver simple physical constraints that can have a significant impact on the solution trajectory. We show through practical examples that these penalties can be formulated in a general manner that minimizes tuning and user involvement, so that the solvers can be used in a "hands-off" adaptive framework.

The setting for our work is the discontinuous Galerkin (DG) discretization of the Reynolds-averaged compressible Navier-Stokes (RANS) equations. DG is a finite element discretization that uses element-wise discontinuous high-order trial and test functions. While our presentation assumes DG and RANS, many of the ideas presented in the paper can be extended to other high-order discretizations and other equations. The outline for the remainder of the paper is as follows. In Section 2, we review the DG spatial discretization of the RANS equations and present the relevant physics constraints. In Section 3, we review PTC, and in Section 4 we present the augmentation of

DOI: 10.1002/nme

PTC with constraints. Section 5 discusses implementation considerations related to treatment of the solution update. We show results in Section 6, and we finish with conclusions in Section 7.

2. SPATIAL DISCRETIZATION

The Reynolds-averaged Navier-Stokes (RANS) equations with the Spalart-Allmaras (SA) turbulence model are written in their compact, conservative form as

$$\partial_t \mathbf{u}_s + \partial_i \mathcal{C}_{is}(\mathbf{u}) - \partial_i \mathcal{D}_{is}(\mathbf{u}, \nabla \mathbf{u}) = \mathcal{S}_s(\mathbf{u}), \tag{1}$$

where $\mathcal{C}_{is}$ and $\mathcal{D}_{is}$ are the convective and diffusive fluxes respectively, $\mathcal{S}_s$ is the SA source term, $i \in [1, .., \dim]$ indexes the spatial dimensions, and s indexes the equations of conservation of mass, momentum, energy, and turbulent viscosity. Accordingly, the state vector is denoted by $\mathbf{u} = [\rho, \rho v_i, \rho E, \rho\tilde{\nu}]^T$, where ρ is the density, v_i are the spatial components of the velocity, E is the specific total energy, and $\tilde{\nu}$ is the working variable for the SA model.

The discontinuous Galerkin (DG) spatial discretization of the flow equations approximates the solution in a space $\mathcal{V}^{H,p}$ of piecewise polynomials of degree p with local support on each element $\kappa^H \in T^H$, where T^H is the set of elements resulting from a subdivision of the spatial domain. The resulting weak form reads:

$$(\partial_t \mathbf{u}^H, \mathbf{w}^H) + \mathbb{R}(\mathbf{u}^H, \mathbf{w}^H) = 0 \qquad \forall\, \mathbf{w}^H \in \mathcal{V}^{H,p}, \tag{2}$$

where $(\cdot, \cdot)$ denotes an inner product and $\mathbb{R}(\mathbf{u}^H, \mathbf{w}^H)$ is a weighted residual statement that includes source, convective, and diffusive terms.

The Riemann flux involved in the convective term is approximated with Roe's [18] solver in which the SA working variable, $\tilde{\nu}$, is transported as a conserved scalar, $\rho\tilde{\nu}$. The diffusion term is discretized using the second form of Bassi & Rebay [19] (BR2). We adopt modifications by Allmaras *et al.* [20] to the original SA model [21] as these modifications ensure stability of the model at negative $\tilde{\nu}$. Also, we discretize the SA equation in $(\rho\tilde{\nu})$ form by combining it with the mass conservation equation.

The discrete system is obtained by expanding the state $\mathbf{u}^H$ in terms of the basis functions that span $\mathcal{V}^{H,p}$ and by using these basis functions as the test functions $\mathbf{w}^H$. The resulting discrete system reads

$$\mathbf{M}\frac{d\mathbf{U}}{dt} = -\mathbf{R}(\mathbf{U}), \tag{3}$$

where $\mathbf{U}$ is the discrete state, $\mathbf{R}$ is the discrete residual operator and $\mathbf{M}$ is the block diagonal mass matrix that corresponds to the volume integral of basis function products on each element in the mesh.

2.1. BR2 stabilization

The diffusive flux, $\mathcal{D}_{is}$, can be written as

$$\mathcal{D}_{is}(\mathbf{u}) = \mathcal{A}_{isjk}(\mathbf{u})\partial_j \mathbf{u}_k, \tag{4}$$

where the tensor $\mathcal{A}_{isjk}$ is a nonlinear function of the state vector, i, j index the spatial dimensions, and s indexes the state vector components. For simplified notation, we omit the dependence of $\mathcal{A}_{isjk}$ on the state vector in the remainder of the text.

DG requires flux evaluations at element interfaces, where the state approximation is generally discontinuous. In the BR2 [19] treatment the diffusive flux is averaged across the interface and augmented by a stabilization term that spreads (lifts) the interface state jumps across the adjacent elements. A constant factor in front of this stabilization term dictates stability – a linear analysis indicates that the minimum value for this factor is the maximum of the number of faces on the two adjacent elements. However, for increased robustness, we scale this number by a "stabilization augmentation factor", $\kappa_{\text{BR2}} > 1$.

Prepared using nmeauth.cls
DOI: 10.1002/nme

2.2. Physics constraints

The flow field is subject to physics constraints that are not guaranteed to be satisfied as the discretized equations only enforce the conservation of state quantities and the entropy condition. Thermodynamic realizability constraints ensure that the equation of state is valid. In the case of fluid flow, these constraints are:

$$\begin{aligned} \frac{p(\mathbf{u}^H(t,\mathbf{x}))}{p_\infty} &> 0, \\ \frac{\rho(\mathbf{u}^H(t,\mathbf{x}))}{\rho_\infty} &> 0, \end{aligned} \tag{5}$$

where p_∞ and ρ_∞ refer to free-stream pressure and density, respectively. These denominators are included here only for non-dimensional convenience and they clearly do not alter the constraints. Note that ρ is a conserved variable and, therefore, its extrema match the extrema of the corresponding position in the conserved state $\mathbf{u}^H$. Pressure, as a nonlinear function of the state, does not have this property, with the exception that linear states one only needs to check end points [22]. In this absence of a closed-form constraint condition on the discrete state vector, Eqn. 5 is verified at a discrete set of points, which in this work are the quadrature points used for the element and face integrals involved in the residual calculation. Note, for Reynolds-averaged turbulent simulations, intuition dictates that the eddy viscosity should be constrained similarly to pressure and density, *i.e.* $\nu_t > 0$ and the modifications in Ref. [20] impose this constraint by modifying the definition of $\nu_t(\mathbf{u})$ from its original form in the baseline SA model – hence, no additional constraints are imposed on ν_t.

3. PSEUDO-TRANSIENT CONTINUATION

Since we are interested in the steady-state solution of the flow equations, high accuracy is not required for discretizing the unsteady term of Eqn. 3. Instead, stability is the main attribute which makes backward Euler an attractive choice. The fully discrete form of Eqn. 3 is then:

$$\mathbf{M}\frac{1}{\Delta t}(\mathbf{U}^{n+1} - \mathbf{U}^n) + \mathbf{R}(\mathbf{U}^{n+1}) = 0, \tag{6}$$

where $\mathbf{M}$ is the mass matrix and n indexes the time step.

For steady calculations, the residual at the future state in Eqn. 6 is expanded about the current state and the steps in the iterative procedure require linear solves for the update $\Delta\mathbf{U}^k$,

$$\left(\mathbf{M}\frac{1}{\Delta t} + \frac{\partial \mathbf{R}}{\partial \mathbf{U}}\Big|_{\mathbf{U}^k}\right)\Delta\mathbf{U}^k = -\mathbf{R}(\mathbf{U}^k), \tag{7}$$

where k is used for the nonlinear iteration number to distinguish the method from the backward Euler case. Note that for $\Delta t \to \infty$ the iterative procedure of Eqn. 7 reduces to Newton's root-finding method. In this work, a restarted Generalized Minimal Residual (GMRES) linear solver [23, 24], aided by an element line-Jacobi preconditioner [15], solves the linear system at each step to a relative tolerance, η_l. Here, $10^{-8} \leq \eta_l \leq 10^{-2}$, is the ratio between the final and the initial linear residual norms.

The DG discretization described in Section 2 produces a residual Jacobian that is block-sparse, which means that degrees of freedom in an element are coupled only to degrees of freedom in neighbor elements. Within each block, sparsity may exist for certain choices of basis functions, but we do not take advantage of such sparsity.

In the first stages of calculations initialized by states that do not satisfy all boundary conditions, strong transients occur due to the propagation of boundary information into the domain. To alleviate those transients and to avoid robustness problems, small time steps are used in an attempt to make the solution follow a physical path. This causes a diagonal dominance in the coefficient matrix

Prepared using nmeauth.cls
DOI: 10.1002/nme

in Eqn. 7 and makes the calculation closer to time-accurate if Δt does not vary spatially. As an alternative to global time stepping, element-wise local time steps can be used by setting a global Courant Friedrichs Lewy (CFL) number and then calculating different time steps on each element according to,

$$\Delta t_{\kappa^H} \quad = \quad \text{CFL}\, \frac{L_{\kappa^H}}{\lambda_{\kappa^H}^{\max}}, \tag{8}$$

where $\lambda_{\kappa^H}^{\max}$ is the maximum wave speed in element κ^H, and L_{κ^H} is a measure of the element's size, here taken as the hydraulic diameter.

At each iteration, k, the flow state vector $\mathbf{U}^k$ is updated with $\Delta\mathbf{U}^k$. For robustness purposes, an under-relaxation parameter, ω^k, is used to ensure a physical solution at the next iteration,

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \omega^k \Delta\mathbf{U}^k. \tag{9}$$

As discussed in Section 5, the value of the under-relaxation parameter is typically set based on a user-prescribed maximum allowable variation of physical quantities such as pressure and density.

Alternatively, pseudo-transient continuation (PTC) can be interpreted as a globalization strategy for Newton's method [25] where a series of problems defined by Eqns. 7 and 9 are solved for $k = 1, 2, ...,$ until $\mathbf{R}(\mathbf{U}^k) = 0$. Its globalization character comes from the fact that $|\mathbf{R}(\mathbf{U}^k)|_{L_2}$ is not required to decrease at each step, hence, it can escape from local minima.

3.1. CFL evolution strategy

In the pseudo-transient continuation method, the continuation parameter is the CFL number. Hence, a strategy must be chosen to evolve the CFL from its initial value to a large value such that Eqn. 7 becomes Newton's method and the state approaches the steady solution.

Many strategies for evolving the CFL are available [26, 6]. Among them, a widely used strategy is the *Switched Evolution Relaxation* (SER) method proposed by Mulder and van Leer [27]. The general idea of SER is to change the time step or the CFL number based on a measure of convergence which is inferred from the relative reduction in the residual L_2-norm between consecutive iterations. Specifically, SER attempts to resolve transients by reducing the CFL number whenever the residual increases and, conversely, increasing the CFL as the solution approaches the basin of attraction of $\mathbf{R}(\mathbf{U}) = 0$. Resolving transients, however, may require many iterations leading to slow convergence or, sometimes, impeding convergence [22].

Alternatively, the CFL can evolve based on the value of the under-relaxation parameter. In this strategy, the CFL increases by a factor $\beta > 1$ if a full update ($\omega = 1$) happened in the previous step of the solver. On the other hand, if the update had to be limited too much, $\omega < \omega_{\min}$, the CFL is reduced by multiplying it by $\kappa < 1$ and the solver step is repeated. In summary,

$$\text{CFL}^{k+1} = \begin{cases} \beta \cdot \text{CFL}^k & \text{for } \beta > 1 \quad \text{if} \quad \omega^k = 1 \\ \text{CFL}^k & \text{if} \quad \omega_{\min} < \omega^k < 1 \\ \kappa \cdot \text{CFL}^k & \text{for } \kappa < 1 \quad \text{if} \quad \omega^k < \omega_{\min} \end{cases}. \tag{10}$$

Here, we set the parameters to: $\omega_{\min} = 0.01$, $1.05 \leq \beta \leq 2.0$, and $\kappa = 0.1$.

This strategy accounts for the physical feasibility constraints for the state update. However, it is an indirect way of avoiding non-physical states in the flow field since the direction $\Delta\mathbf{U}^k$ may still produce states that are closer to becoming non-physical even at very small CFL. In particular, this is observed on highly under-resolved meshes.

3.2. Optimization aspect of PTC

Assume the matrix in front of $\Delta\mathbf{U}^k$ in Eqn. 7 (call it $\mathbf{A}$) is real and non-singular and that the update direction $\Delta\mathbf{U}^k$ is not zero. Multiplying the left-hand side of Eqn. 7 by its own transpose gives

$$\Delta\mathbf{U}^{k^T}\mathbf{A}^T \underbrace{\left(\mathbf{M}\frac{1}{\Delta t} + \frac{\partial\mathbf{R}}{\partial\mathbf{U}}\Big|_{\mathbf{U}^k}\right)}_{\mathbf{A}} \Delta\mathbf{U}^k = -\Delta\mathbf{U}^{k^T} \underbrace{\mathbf{A}^T\mathbf{R}(\mathbf{U}^k)}_{\frac{\partial f}{\partial\mathbf{U}}\Big|_{\mathbf{U}^k}} > 0, \tag{11}$$

DOI: 10.1002/nme

where the inequality arises from the fact that the left hand side is the dot product of a nonzero vector with itself, which is always positive. Therefore, $\Delta\mathbf{U}^k$ is a descent direction for the scalar function $f(\tilde{\mathbf{U}})$ defined by its gradient in the right-hand side of Eqn. 11. This function is

$$f(\tilde{\mathbf{U}}) = \frac{1}{2}|\mathbf{R}_t(\tilde{\mathbf{U}})|^2_{L_2} = \frac{1}{2}\mathbf{R}_t(\tilde{\mathbf{U}})^T\mathbf{R}_t(\tilde{\mathbf{U}}). \tag{12}$$

where the unsteady residual is defined by

$$\mathbf{R}_t(\tilde{\mathbf{U}}) \equiv \mathbf{M}\frac{1}{\Delta t}(\tilde{\mathbf{U}} - \mathbf{U}^k) + \mathbf{R}(\tilde{\mathbf{U}}), \tag{13}$$

Consequently, there is a trial state $\tilde{\mathbf{U}}$ along the direction $\Delta\mathbf{U}^k$ such that $f(\tilde{\mathbf{U}}) < f(\mathbf{U}^k)$.

4. INCORPORATING CONSTRAINTS

The minimization character of the PTC method motivates the use of constraint handling techniques from optimization to incorporate the physics constraints from Section 2.2 into the solution path since non-physical states (*e.g.* negative pressure) can lead to instability [28]. Interior penalty methods [29] are attractive because of their simplicity and efficiency in acknowledging feasibility constraints. These methods augment a scalar objective function with a term – the penalty – that tends to infinity as the solution path approaches a feasibility boundary, creating a repelling effect with respect to prohibited regions of the domain.

A different approach for incorporating constraints into pseudo-transient methods is proposed by Kelley *et al.* [8]. Their approach involves a step that projects the state into the feasible domain after each non-linear iteration and the fundamental difference between their method and the method we propose here is that we incorporate the constraints when computing the solution update.

A simple way of incorporating the realizability constraints in the solution path is to formulate an optimization problem that minimizes $|\mathbf{R}_t(\mathbf{U})|^2_{L_2}$ by varying $\mathbf{U}$, subject to the constraints. However, this least-squares minimization problem gives an ill-conditioned (approximate) Hessian matrix due to a squaring of the residual Jacobian matrix [16]. In addition, factorizing the Hessian would generally require its explicit construction, which would be computationally intensive even for small problems. For these reasons, the least-squares optimization approach is inadequate for any realistic problem.

As an alternative to constrained least-squares, we augment the residual with a penalty vector to account for the constraints [16]. The augmented residual is

$$\mathbf{R}_p(\mathbf{U}) \equiv \mathbf{R}(\mathbf{U}) + \mathbf{P}(\mathbf{U}, \mu_{\mathrm{P}}), \tag{14}$$

where μ_{P} is a penalty factor. In order to have a repelling effect with respect to non-feasible regions of the domain, the penalization vector $\mathbf{P}$ must have a positive projection on the direction of the residual vector $\mathbf{R}$. To satisfy this requirement, we define the penalization vector as

$$\mathbf{P}(\mathbf{U}, \mu_{\mathrm{P}}) \equiv \Phi(\mathbf{U}, \mu_{\mathrm{P}})\ \mathbf{R}(\mathbf{U}), \tag{15}$$

where Φ is a diagonal matrix of the same size as the residual Jacobian with the elemental penalties $\mathbb{P}_{\kappa^H}$ for each row corresponding to an element κ^H.

$$\Phi_{ij}(\mathbf{U}, \mu_{\mathrm{P}}) = \begin{cases} \mu_{\mathrm{P}}\, \mathbb{P}_{\kappa^H}(\mathbf{U}) & \text{if } i = j \in \mathrm{dof}(\kappa^H) \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

Note that $j \in \mathrm{dof}(\kappa^H)$ denotes the degrees of freedom, in global ordering, pertinent to element κ^H. $\mathbb{P}_{\kappa^H}$ is an element-specific barrier function that imposes the constraints similarly to interior penalty optimization methods. Here, we consider an inverse barrier where the penalty function is a sum of inverses of the constraints c_i. Since the constraints are applied to a functional representation of the

Prepared using nmeauth.cls
DOI: 10.1002/nme

state, an integral of the inverse barrier would have to be evaluated in order to enforce the constraints everywhere in the domain; we approximate this integral by using a quadrature rule and the penalty function is written as

$$\mathbb{P}_{\kappa^H}(\mathbf{U}) = \sum_{i}^{N_c} \sum_{q}^{N'_q} \frac{w'_q}{c_i(\mathbf{u}^H(\mathbf{x}'_q))}. \tag{17}$$

where N_q is the number of quadrature points x_q with weights w_q, and N_c is the number of constraints indexed by i. Note that $\mathbb{P}_{\kappa^H}$ in tends to infinity as the constraints approach zero from the positive side.

Eqn. 17 involves a summation over quadrature points, $\mathbf{x}'_q$, that lie inside κ^H, with weights w'_q. This summation corresponds to integrating the inverse barrier function in a reference element. The primed points and weights are determined by an enhanced quadrature rule used for integrating the barrier function. That is, if the quadrature rule for the residual calculation as a function of the polynomial order is QuadRule(q), the rule used for the barrier is QuadRule$(q + \Delta q)$, where $\Delta q = 4$ for all cases presented in this article.

Note that the projection of $\mathbf{P}$ – as defined in Eqn. 15 – onto the residual vector is always positive for non-zero $\mathbf{R}$ since the elemental penalties are strictly positive in the feasible domain, *i.e.* the physical states.

Furthermore, a root of the residual operator corresponds to a root of $\mathbf{R}_p$, so that the steady-state solution is independent of the values of the elemental penalties. We emphasize that the objective of this method is to change the path to the solution, not the solution itself. By applying the pseudo-transient continuation procedure (Eqn. 7) to $\mathbf{R}_p$ we are including physics constraints in the solution path from the initial condition to steady state. The update direction along that path at step k satisfies

$$\left(\underbrace{(\mathbf{I} + \Phi^k)^{-1} \frac{\mathbf{M}}{\Delta t}}_{a} + \frac{\partial \mathbf{R}}{\partial \mathbf{U}}\Big|_{\mathbf{U}^k} + \underbrace{(\mathbf{I} + \Phi^k)^{-1} \left(\frac{\partial \Phi}{\partial \mathbf{U}}\Big|_{\mathbf{U}^k} \mathbf{R}(\mathbf{U}^k) \right)}_{b} \right) \Delta \mathbf{U}^k = -\mathbf{R}(\mathbf{U}^k), \tag{18}$$

where $\mathbf{I}$ is the identity matrix and $\Phi^k = \Phi(\mathbf{U}^k, \mu_{\mathrm{P}}^k)$. The equation above is derived by substituting $\mathbf{R}_p$ into Eqn. 7 and by separating the terms such that the unpenalized residual, $\mathbf{R}$, is on the right-hand side. This adds the implementation convenience of simply adding entries to the coefficient matrix of the linear systems solved at each step k.

The globalization and penalization terms – respectively "a" and "b" in Eqn. 18 – are block diagonal for the DG method in this work. In addition, the elemental CFL number gets amplified by $(1 + \mu_{\mathrm{P}}\, \mathbb{P}_{\kappa^H})$ as $\mathbf{I} + \Phi^k$ is a diagonal matrix. In the limit of an infinite time step, the solution path seeks a minimum of $|\mathbf{R}_p|_{L_2}$. Similarly, the globalization term vanishes locally at elements where the solution approaches a non-physical region while the penalization term does not vanish because the function value of inverse barrier penalties (Eqn. 17) tends to infinity at a slower rate than the magnitude of its derivative. In the remainder of the text, we will refer to the method in Eqn. 18 as *Constrained Pseudo-transient Continuation* (CPTC).

The final value of μ_{P} is not specified *a priori* as it controls the effect of penalization with respect to the globalization term. The choice of initial value for μ_{P} balances the globalization and penalization terms for the first nonlinear iteration. Assuming the state is initialized by uniform free-stream conditions, we can equate the coefficients multiplying the globalization and penalization matrices,

$$\frac{1}{(1 + \mu_{\mathrm{P}}^0 \mathbb{P}_0) \cdot \mathrm{CFL}^0} = \frac{\mu_{\mathrm{P}}^0}{(1 + \mu_{\mathrm{P}}^0 \mathbb{P}_0)} \Rightarrow \mu_{\mathrm{P}}^0 = \frac{1}{\mathrm{CFL}^0}. \tag{19}$$

μ_{P} in the numerator of the right-hand side of Eqn. 19 comes from $\partial\Phi/\partial\mathbf{U}$ in Eqn. 18 and CFL^0 in left-hand side is factored out of the elemental time step. As for PTC, in Eqn. 19 we assume that the residuals are properly scaled so that a single-CFL time continuation globalizes all of the equations.

DOI: 10.1002/nme

As the solution evolves, the balance between penalization and globalization may change. This balance should shift depending on how close the current state iterate is from being non-physical. One possible strategy is a form of *Switched Evolution-Relaxation* (SER) for μ_{P}:

$$\mu_{\mathrm{P}}^{k+1} = \mu_{\mathrm{P}}^{k} \frac{1 + \mu_{\mathrm{P}}^{k} \langle \mathbb{P}_{\kappa^H}(\mathbf{U}^k) \rangle}{1 + \mu_{\mathrm{P}}^{k-1} \langle \mathbb{P}_{\kappa^H}(\mathbf{U}^{k-1}) \rangle}, \tag{20}$$

where $\langle \cdot \rangle$ indicates an average over all the elements. The evolution strategy in Eqn. 20 makes the solver acknowledge the presence of a feasibility constraint by increasing its repelling effect as the solution path goes towards a non-physical state. Conversely, if the solution path is moving away from a feasibility boundary the repelling effect decreases.

Note that in Reference [22] we evolve μ_{P} using SER based on the maximum elemental penalty. Although successful in avoiding non-physical states in many difficult flow problems, that strategy tends to produce ill-conditioned linear systems in the Newton steps that sometimes leads to GMRES failure. Also, we found that varying μ_{P} between nonlinear iterations is not strictly necessary and the method still attains satisfactory robustness with constant μ_{P}. Section 6.1 compares the method's performance for these two strategies.

The CPTC method is summarized in Algorithm 1. The unconstrained PTC follows a similar algorithm, where the steps related to the penalty factor (steps 3 and 16) are ignored and the update direction (step 6) is computed using Eqn. 7. For all the cases presented here, the CFL is reduced by a factor $\kappa = 0.1$ when the under-relaxation factor is below $\omega_{\min} = 0.01$. At that point the state is reverted to a safe state, $\mathbf{U}_{\text{safe}}$, stored when the last full update occurred.

Algorithm 1 Constrained PTC

1: Set a residual tolerance, ε_{res}
2: Choose initial CFL^0 and its increase factor β (Eqn. 10)
3: Initialize μ_{P}^0 according to Eqn. 19
4: Initialize $\mathbf{U}_{\text{safe}}$ to initial condition
5: **while** $|\mathbf{R}(\mathbf{U}^k)| > \varepsilon_{\text{res}}$, $k <$ maximum iterations **do**
6: Compute $\Delta\mathbf{U}^k$ by solving Eqn. 18 using GMRES
7: Compute under-relaxation parameter ω^k (Section 5)
8: **if** $\omega^k \geq \omega_{\min}$ **then**
9: $\mathbf{U}^{k+1} \leftarrow \mathbf{U}^k + \omega^k \Delta U^k$.
10: **if** $\omega^k = 1$ **then**
11: $\mathbf{U}_{\text{safe}} \leftarrow \mathbf{U}^{k+1}$ ▷ Store a safe state
12: **end if**
13: **else**
14: $\mathbf{U}^{k+1} \leftarrow \mathbf{U}_{\text{safe}}$ ▷ Revert to last safe state
15: **end if**
16: Update μ_{P}^{k+1} with Eqn. 20
17: Update CFL^{k+1} with Eqn. 10
18: $k \leftarrow k + 1$
19: **end while**

5. SOLUTION UPDATE

In optimization problems, line searches are used to find a step size along a descent direction that sufficiently reduces the value of the objective function and its gradient. These conditions are known as the *Wolfe conditions*. When solving systems of nonlinear equations, line searches improve the global convergence properties of Newton-based methods [30].

The line search described here uses two main ingredients. First, it requires interpolating the state and its update at certain points, $\vec{x}_m$. This involves evaluating the basis functions at $\vec{x}_m$ and using

DOI: 10.1002/nme

the discrete vectors $\mathbf{U}$ and $\Delta\mathbf{U}$ to yield the field representations, $\mathbf{u}^H(\vec{x})$ and $\Delta\mathbf{u}^H(\vec{x})$. The second ingredient is an update limiter that restricts the unsafe changes in the constrained variables (pressure and density) to a maximum fraction, $\eta_{\max}$, of the current values. This procedure is described in Algorithm 2.

Algorithm 2 Limit physical update

1: Given $\mathbf{u}^H(\vec{x}_m)|_{\kappa^H}$, $\Delta\mathbf{u}^H(\vec{x}_m)|_{\kappa^H}$, and a fraction $\eta_{\max} < 1$
2: $\omega_{\kappa^H} \leftarrow 1$
3: **for all** $\vec{x}_m \in \kappa^H$ **do**
4: $\quad\omega_\rho \leftarrow 1$ ▷ ω_ρ is the step size for density
5: $\quad\rho_m = \rho(\mathbf{u}^H(\vec{x}_m)|_{\kappa^H})$ ▷ Current density at $\vec{x}_m$
6: $\quad\Delta\rho_m = \rho(\Delta\mathbf{u}^H(\vec{x}_m)|_{\kappa^H}$ ▷ Change in density at $\vec{x}_m$
7: $\quad\omega_\rho = -\dfrac{\eta_{\max}\rho_m}{\Delta\rho_m}$
8: $\quad$**if** $\omega_\rho < 0$ **OR** $\omega_\rho > 1$ **then**
9: $\quad\quad\omega_\rho \leftarrow 1$
10: $\quad$**end if**
11: $\quad\omega_p \leftarrow \omega_\rho$ ▷ ω_p is the step size for pressure
12: $\quad p_m = p(\mathbf{u}^H(\vec{x}_m)|_{\kappa^H})$ ▷ Current pressure at $\vec{x}_m$
13: $\quad\tilde{p}_m = p(\mathbf{u}^H(\vec{x}_m)|_{\kappa^H} + \omega_p\Delta\mathbf{u}^H(\vec{x}_m)|_{\kappa^H})$ ▷ Trial pressure at $\vec{x}_m$
14: $\quad$**while** $\tilde{p}_m < (1-\eta_{\max})\cdot p_m$ **do**
15: $\quad\quad\omega_p \leftarrow \dfrac{\omega_p}{2}$
16: $\quad\quad\tilde{p}_m = p(\mathbf{u}^{H,p}(\vec{x}_m)|_{\kappa^H} + \omega_p\Delta\mathbf{u}^{H,p}(\vec{x}_m)|_{\kappa^H})$
17: $\quad$**end while**
18: $\quad\omega_{\kappa^H} \leftarrow \min(\omega_\rho, \omega_p, \omega_{\kappa^H})$
19: **end for**
20: **return** ω_{κ^H}

Some clarifications are in order. First, the maximum fractional change is fixed at $\eta_{\max} = 10\%$ – based on experimentation – for all cases presented in this work. Also, for the points $\vec{x}_m$, we reuse the quadrature points from computing the interior and boundary integrals involved in the residual calculation. Finally, the bisection method is used in step 14 of Algorithm 2 because pressure is a nonlinear function of the state.

5.1. Line search

The line-search algorithm presented in this work is based on the work of Modisette [31], and it relies on the optimization character of pseudo-transient continuation (Section 3.2). In short, both algorithms satisfy Armijo's rule [32] by back-tracking from an initial step size until an update leads to a reduction in $|\mathbf{R}_t|^2_{L_2}$. Here, we relax Armijo's rule by a factor κ_{LS} and we select the initial step-size as the minimum ω_{κ^H} over all the elements. The effect of κ_{LS} is discussed in Section 6.2. Algorithm 3 summarizes the line-search procedure.

DOI: 10.1002/nme

Algorithm 3 Line search

1: $\omega_{\text{phys}} \leftarrow 1$ ▷ Initial guess for physical update
2: **for all** κ^H **do**
3: Select limit points, $\vec{x}_m$
4: Evaluate $\mathbf{u}^{H,p}(\vec{x}_m)|_{\kappa^H}$ and $\Delta\mathbf{u}^{H,p}(\vec{x}_m)|_{\kappa^H}$
5: Call Algorithm 2 ▷ Limit physical update
6: $\omega_{\text{phys}} \leftarrow \min(\omega_{\kappa^H}, \omega_{\text{phys}})$
7: **end for**
8: $\omega^k \leftarrow \omega_{\text{phys}}$ ▷ Set initial step size
9: $\tilde{\mathbf{U}} \leftarrow \mathbf{U}^k + \omega^k \Delta\mathbf{U}^k$ ▷ Trial state vector
10: **while** $|\mathbf{R}_t(\tilde{\mathbf{U}})|_{L_2} > \kappa_{\text{LS}} |\mathbf{R}(\mathbf{U}^k)|_{L_2}$ **OR** $\tilde{\mathbf{U}}$ is not physical **do**
11: $\omega^k \leftarrow \dfrac{\omega^k}{2}$
12: $\tilde{\mathbf{U}} \leftarrow \mathbf{U}^k + \omega^k \Delta\mathbf{U}^k$
13: **end while**
14: **return** ω^k

Note that step 10 in Algorithm 3 checks if the trial state, $\tilde{\mathbf{U}}$, is physical. This check involves verifying if the physics constraints are satisfied at the limit points. Also, when the line search is used with CPTC, the residual operator is penalized according to Eqn. 14, and hence $\mathbf{R}$ is replaced by $\mathbf{R}_p$.

In references [31, 33, 22] the L_2-norm in step 10 of Algorithm 3 is separated into residual norms for each of the conservation equations and a drop is required in each of those norms. This improves robustness with respect to badly-scaled discrete systems that cause the residual norm to be dominated by the worst residual component. The poor scaling is frequently present in flow problems involving turbulence models. Specifically, in the case of the Spalart-Allmaras model, a simple scalar scaling [34, 35] of SA's discrete equation is very effective in bringing the equation-specific residuals to similar magnitudes. In such a case, requiring the reduction of individual residual norms restricts the step-sizes to small values thus requiring many iterations in the globalization phase. For this reason, we do not separate the residual norms in this work and, instead, we rescale the additional discrete equation corresponding to the SA turbulence model.

5.1.1. Greedy algorithm The physical update limiter in Algorithm 2 is heuristic and the line search described above can prematurely exit with $\omega^k = \omega_{\text{phys}}$ while $\omega_{\text{phys}} < 1$. This can slow down the convergence and increase the susceptibility to limit cycles. To address this possibility, a greedy algorithm is introduced. This algorithm amplifies ω^k while Armijo's rule is satisfied or until a full update is obtained, $\omega^k = 1$. The algorithm is summarized below.

DOI: 10.1002/nme

Algorithm 4 Greedy algorithm

1: **if** $\omega^k = \omega_{\text{phys}}$ **then**
2: **while** $\omega^k \leq 1$ **do**
3: $\omega^k \leftarrow \beta_\omega \cdot \omega^k$ ▷ For all cases, we use $\beta_\omega = 1.1$
4: $\tilde{\mathbf{U}} \leftarrow \mathbf{U}^k + \omega^k \Delta \mathbf{U}^k$
5: **if** $\tilde{\mathbf{U}}$ is not physical **then**
6: $\omega^k \leftarrow \dfrac{\omega^k}{2}$
7: **return** ω^k
8: **end if**
9: **if** $|\mathbf{R}_t(\tilde{\mathbf{U}})|_{L_2} > \kappa_{\text{LS}} |\mathbf{R}(\mathbf{U}^k)|_{L_2}$ **then**
10: $\omega^k \leftarrow \dfrac{\omega^k}{\beta_\omega}$
11: **return** ω^k
12: **end if**
13: **end while**
14: **end if**
15: **return** ω^k

Since the greedy algorithm is an extension of the line search, the same remarks made above apply here. Specifically, the residual operator is penalized in step 9 when this algorithm is applied to CPTC.

6. RESULTS

6.1. One-dimensional shock tube

We now revisit the shock-tube problem presented in Section 1 for which time-accurate integration results in non-physical states. The purpose here is to assess the ability of PTC and CPTC to skip the non-physical transients and to reach steady-state. The assessment considers a range of mesh resolutions, approximation orders, initial CFL, and CFL growth factors (β in Eqn. 10). Table I shows the values for the parametric study which consists of a total of 300 parameter combinations for each method. The linear systems at each nonlinear step are solved to a relative tolerance of $\eta_l = 10^{-2}$ and Armijo's rule relaxation factor is $\kappa_{\text{LS}} = 1.05$ for all runs. The nonlinear residual convergence tolerance is 10^{-8}.

Table I. One-dimensional shock tube: variable parameters.

Parameter	Values
Number of elements	10, 20, 40, 80, 160
Approximation order	$p = 0,\ 1,\ 2,\ 3$
CFL^0	$0.1,\ 0.5,\ 1,\ 5,\ 10$
CFL growth factor	$\beta = 1.05,\ 1.5,\ 2$

Since the solution transient undergoes a shock, one should consider including some form of shock-capturing scheme in the residual operator in order to improve robustness. Here, we compare the pseudo-transient continuation methods for two forms of the residual operator. One where the residual includes only the convection term from Euler's equation and another where the residual consists of both convection and artificial diffusion. We use Persson and Peraire's [11] shock-capturing scheme for which solution regularity is sensed by density jumps between the current

DOI: 10.1002/nme

p^{th}-order solution and its projection onto $\mathcal{V}^{H,p-1}$. The added diffusion is discretized with BR2 for which $\kappa_{\text{BR2}} = 1$. Note, the $p = 0$ runs do not include any shock-capturing term.

We consider CPTC in two modes. One where the penalty factor, μ_{P}, varies according to Eqn. 20 and another where we keep its value constant. In both modes, we initialize μ_{P} according to Eqn. 19. Table II compares the success rate – percentage of runs that reach steady-state – of CPTC and PTC. Note that the inclusion of physics constraints in the solution path significantly improves the robustness in converging to steady state. Also, the simplification of holding μ_{P} constant has a small impact ($\leq 3\%$) on the method's success rate.

Table II. One-dimensional shock tube: success rate for PTC and CPTC over the 300 parameter combinations in Table I.

Description	PTC	CPTC, variable μ_{P}	CPTC, constant μ_{P}
w/o Artificial diffusion	66.33%	91.67%	88.67%
w/ Artificial diffusion	89.67%	96.00%	94.67%

From a robustness perspective, CPTC with variable penalty factor produces a better improvement than the use of the artificial diffusion term. This observation, however, is reserved to cases where a shock occurs only during the solution transient and is not present in the steady solution as CPTC does not change the final steady solution. Furthermore, the inclusion of a diffusion term governed by a regularity sensor in the residual operator produces nonlinear algebraic systems that are generally more difficult to solve. The latter point is supported by Table III which shows that converged runs with PTC take, on average, approximately three times more nonlinear iterations when the residual includes the artificial diffusion term. Conversely, CPTC's negative impact is on the average cost of the linear systems at each nonlinear step. This is measured by the average number of GMRES iterations per nonlinear iteration. CPTC takes, on average, between 9% to 14% more GMRES iterations than PTC at each nonlinear step for this shock-tube problem. Note that this negative impact is compensated by fewer nonlinear iterations such that the total number of GMRES iterations is generally smaller than the same metric for PTC with the exception of CPTC with variable μ_{P} without artificial diffusion.

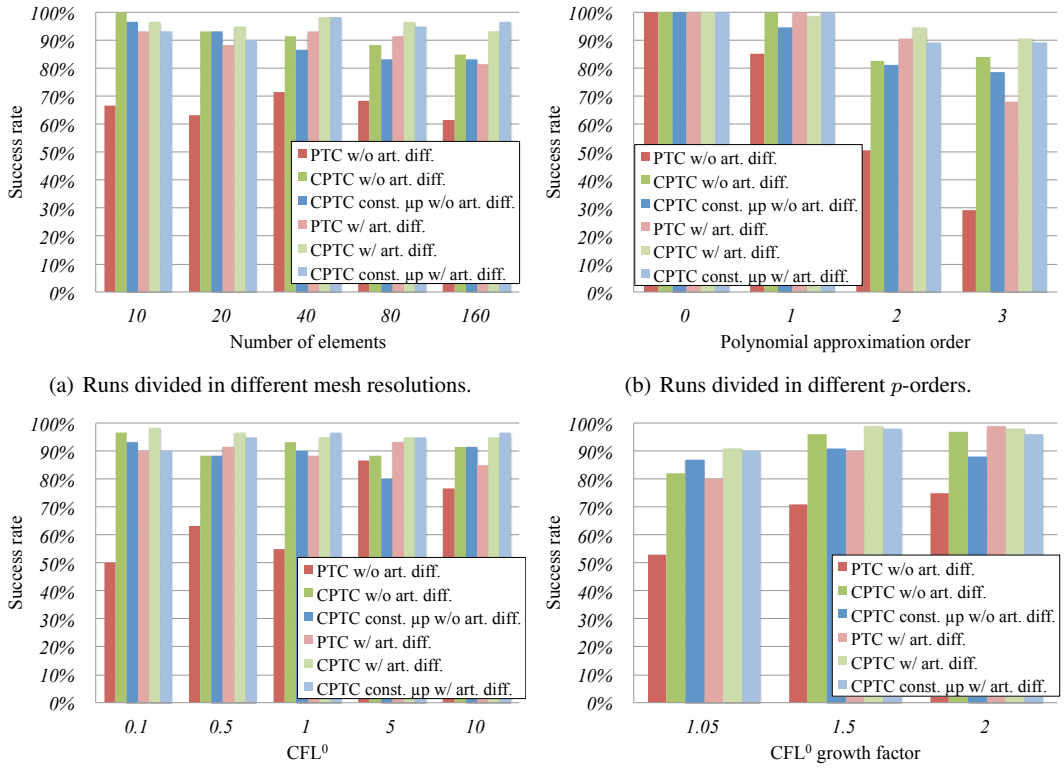
Table III. One-dimensional shock tube: cost metrics for all converged runs normalized by PTC's performance (absolute values in parentheses).

Average cost	PTC	CPTC, variable μ_{P}	CPTC, const. μ_{P}
Without artificial diffusion			
Nonlinear iterations	1 (40.78)	0.95	0.84
GMRES iterations	1 (59.81)	1.02	0.92
GMRES iter. per nonlinear iter.	1 (1.67)	1.10	1.09
With artificial diffusion			
Nonlinear iterations per run	1 (120.75)	0.61	0.64
GMRES iterations per run	1 (146.28)	0.69	0.71
GMRES iter. per nonlinear iter.	1 (1.59)	1.12	1.14

We now analyze the effect of the parameters in Table I on the success rate of the continuation methods. For this, we compute marginal success rates, one for each parameter value within each class while marginalizing the other classes – *e.g.*: success of all runs with $\text{CFL}^0 = 1.0$. Note, the average of the success rates over all parameter values within a class recovers the global success rate.

DOI: 10.1002/nme

Figure 3 compares the marginal success rates for CPTC against PTC. Note that PTC suffers more than CPTC from increasing approximation order regardless of the form of the residual operator. The magnitude of the oscillations caused by the shock increases with the polynomial order and these oscillations are the root cause of violation of the physics constraints. Including these constraints in the solution path improves the ability of the pseudo-time procedure to skip the non-physical transients. Another mechanism that allows the pseudo-time procedure to skip transients is increasing the CFL number. This is supported by Figures 3(d) and 3(c), which show, respectively, that PTC's success rate increases with the CFL growth factor and PTC without artificial diffusion is more successful with $\text{CFL}^0 > 1$. Increasing the CFL, however, is not a selective mechanism as it washes all transients and it can affect the globalization character of pseudo-transient continuation.



(a) Runs divided in different mesh resolutions.

(b) Runs divided in different p-orders.

(c) Runs divided in different CFL^0.

(d) Runs divided in different CFL growth factors.

Figure 3. One-dimensional shock tube: success rates with varying parameters. Results are marginalized over all other parameters in each case.

CPTC's success without artificial diffusion decreases with increasing mesh resolution for this flow problem (Figure 3(a)). The reason for this behavior is that the shock becomes steeper as the mesh gets finer and, in the absence of the shock capturing term, the pressure undershoots to lower values making it harder to circumvent non-physical regions of the solution space. Note that teaming CPTC with artificial diffusion practically eliminates the dependence of the marginal success rates on mesh resolution.

6.2. *Effect of κ_{LS}*

We now analyze the effect of relaxing Armijo's rule on the success rate of the solver. In order to properly exercise both PTC and CPTC methods, we choose two turbulent flows in which DG methods typically use parameter continuation and order sequencing. The first case is transonic flow at $M_\infty = 0.734$, $\alpha = 2.79°$, and $Re = 6.5 \times 10^6$ over the RAE2822 airfoil. The steady solution for

DOI: 10.1002/nme

this problem presents a shock on the upper surface of the airfoil, and hence the residual operator includes the artificial diffusion term of Persson and Peraire [11]. The main difficulty of this case is the fast flow acceleration over the upper leading-edge region that causes the pressure to reach very low values in the solution transient. The second case is turbulent flow at $M_\infty = 0.2, \alpha = 16^\circ, Re = 9 \times 10^6$ over the MDA 30P30N high-lift configuration. Here, the residual operator does not include the artificial diffusion term and the main cause for difficulty is the high angle of attack which causes the flow to experience strong shear while contouring the airfoil shape.

In both cases, we rescale the discrete SA equation to bring the nonlinear residuals to similar magnitudes [35] and we keep μ_{P} constant as described in Section 6.1. We consider $\kappa_{\text{LS}} = \{0.9, 0.95, 1.0, 1.05, 1.1\}$ and we assign the remaining parameters to the values listed in Table IV. In each run, the flow is initialized with the free stream condition throughout the domain and the nonlinear residual is considered converged when $|\mathbf{R}| < 10^{-8}$.

Table IV. Fixed parameters for the κ_{LS} sensitivity study.

p-order	κ_{BR2}	CFL^0	β	η_l	GMRES Vectors
RAE2822, $M_\infty = 0.734, \alpha = 2.79^\circ, Re = 6.5 \times 10^6$					
$p = 1$	15.0	1.0	2.0	10^{-3}	80
$p = 2$	10.0	1.0	2.0	10^{-3}	80
MDA 30p30n, $M_\infty = 0.2, \alpha = 16^\circ, Re = 9 \times 10^6$					
$p = 1$	2.0	1.0	1.5	10^{-3}	80
$p = 2$	2.0	1.0	1.5	10^{-8}	100

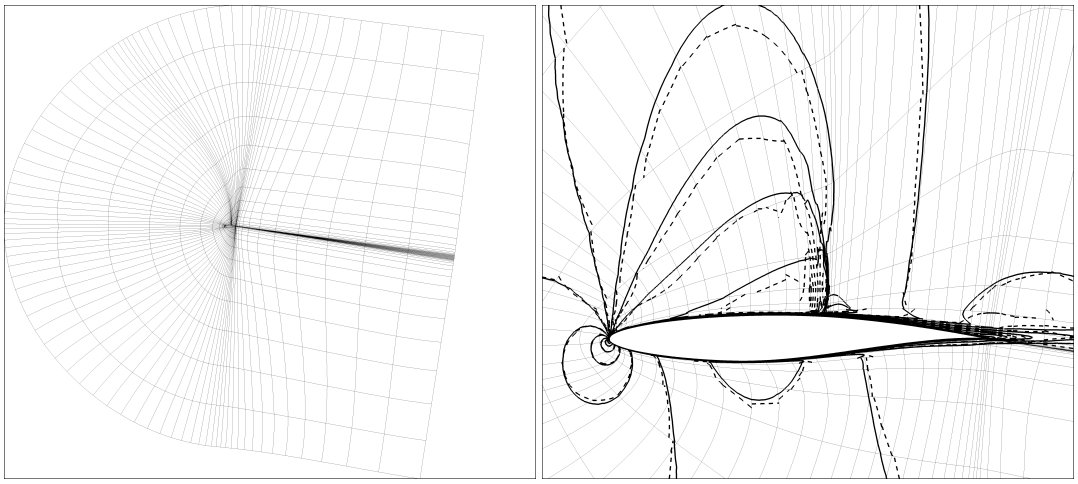
The mesh chosen for the transonic case (Figure 4) is publicly provided by the High-Order Workshop committee [1] and its purpose is to establish an appropriate reference for testing the accuracy and efficiency of high-order methods. Here, our purpose with choosing it is to test the robustness of PTC and CPTC on a reasonably – but not fully – resolved mesh instead of comparing the methods on an inappropriately coarse mesh such that the flow features and not even representable. Each edge of the mesh is a quartic polynomial and the off-wall spacing is such that $y^+_{\max} \approx 8$ for both $p = 1$ and $p = 2$. Note in Figure 4(b) the mesh clustering in the shock region.

Table V shows the success of both methods in reaching the steady solution for the transonic case. First, we note that CPTC converges all cases but that the solver's performance varies significantly with the value of κ_{LS} – larger values require fewer nonlinear iterations. Within the PTC runs, strictly enforcing Armijo's rule ($\kappa_{\text{LS}} < 1.0$) makes the solver resolve certain transients that lead to violating physical realizability (Figure 5). The exception here is the $p = 2$, $\kappa_{\text{LS}} = 0.9$ run that converges while its $p = 1$ counterpart does not. This an example where order continuation would fail with PTC because the supposedly easier $p = 1$ solution is effectively harder to obtain.

Atkins and Pampell [28] examine an instability that occurs for DG when the pressure goes negative while solving the Euler equations. Here, we note a similar instability manifesting in the residual norm (Figure 6(a)) when the minimum pressure and density become negative in the PTC run with $\kappa_{\text{LS}} = 0.95$. Note in Figure 6(b) that the maximum penalty peaks in the transition from time-continuation to the full Newton stage, when the CFL ramps from $\mathcal{O}(1)$ to $\mathcal{O}(10^4)$. The residual norm at that point has already dropped 2 orders of magnitude (Figure 6(a)) which demonstrates that non-physical states can occur not only during the initial transients, but at any point in the solution path.

We now analyze the performance of the two methods in the high-lift case. The mesh used for this case is shown in Figure 7 and it consists of 4070 quartic elements generated via structured agglomeration of linear cells. The off-wall spacing, excluding the flap cove region, is such that $y^+_{\max} \approx 50$ for $p = 1$ and $p = 2$. This mesh is publicly provided by the High-Order Workshop committee [1].

DOI: 10.1002/nme



(a) Global view of the mesh mesh (2024 elements).

(b) Mach contours. Thick dashed lines: $p = 1$; thick continuous lines: $p = 2$.

Figure 4. RAE2822, $M_\infty = 0.734, \alpha = 2.79^\circ, Re = 6.5 \times 10^6$: quartic mesh used for κ_{LS} study.

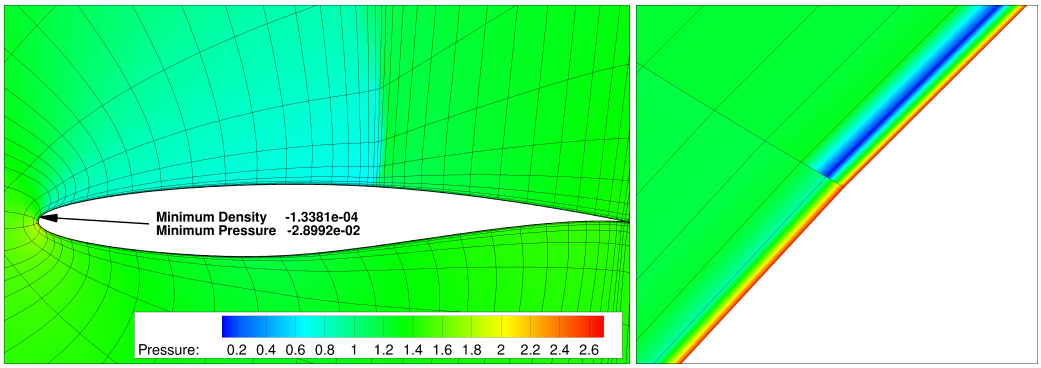
Table V. RAE2822, $M_\infty = 0.734, \alpha = 2.79^\circ, Re = 6.5 \times 10^6$: success of all runs for various κ_{LS}. C = converged; LM = local minimum; NP = nonphysical.

κ_{LS}	Success		Nonlinear iterations		GMRES iterations	
	$p = 1$	$p = 2$	$p = 1$	$p = 2$	$p = 1$	$p = 2$
PTC						
0.9	NP	C	279	499	1892	8128
0.95	NP	NP	412	125	3287	861
1.0	C	C	215	323	4734	8035
1.05	C	C	116	345	2893	8709
1.1	C	C	93	153	2153	3817
CPTC						
0.9	C	C	301	568	4231	9388
0.95	C	C	209	571	3311	9405
1.0	C	C	186	431	4692	10703
1.05	C	C	197	184	4852	4946
1.1	C	C	101	156	2399	4123

Table VI compares the success of PTC and CPTC for the high-lift case. First we remark that, similarly to the transonic case, the added spatial resolution of $p = 2$ reduces the number of nonphysical problems encountered by PTC. This is somewhat counter-intuitive as it opposes the idea of order continuation which assumes that at higher approximation orders the problem becomes more difficult. In contrast to the transonic case, however, relaxing Armijo's rule here makes PTC violate the physical constraints for $p = 1$ as seen in Table VI and exemplified in Figure 8. CPTC, on the other hand, is less sensitive to κ_{LS} as the constrained solver is successful with nearly all of the values of κ_{LS} with the exception of $\kappa_{\text{LS}} = 0.9$ for $p = 2$, with which both methods fail to converge.

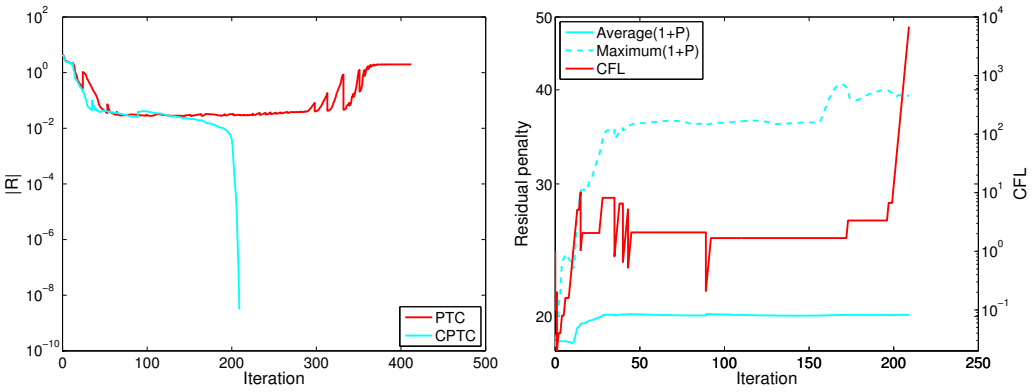
Given that both methods are successful with $\kappa_{\text{LS}} = 0.9$ and $p = 1$ but not with $p = 2$, we investigate if order continuation is successful in this condition. Figure 9 compares PTC and CPTC

Prepared using nmeauth.cls
DOI: 10.1002/nme



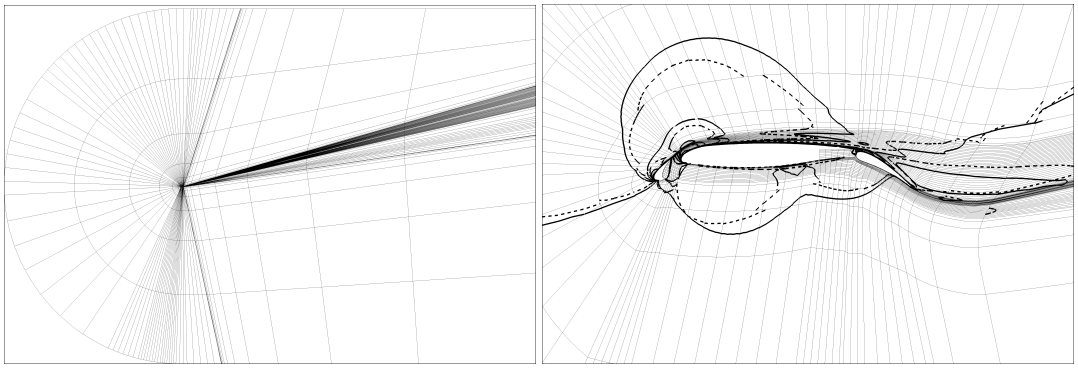
(a) Pressure distribution with location of constraint violation. (b) Zoom on pressure constraint violation.

Figure 5. RAE2822, $M_\infty = 0.734, \alpha = 2.79^\circ, Re = 6.5 \times 10^6, p = 1, \kappa_{\text{LS}} = 0.95$: PTC state iterate that violates physics constraints.



(a) Residual histories. (b) Penalty and CFL histories for CPTC run.

Figure 6. RAE2822, $M_\infty = 0.734, \alpha = 2.79^\circ, Re = 6.5 \times 10^6, p = 1$: PTC versus CPTC for $\kappa_{\text{LS}} = 0.95$.



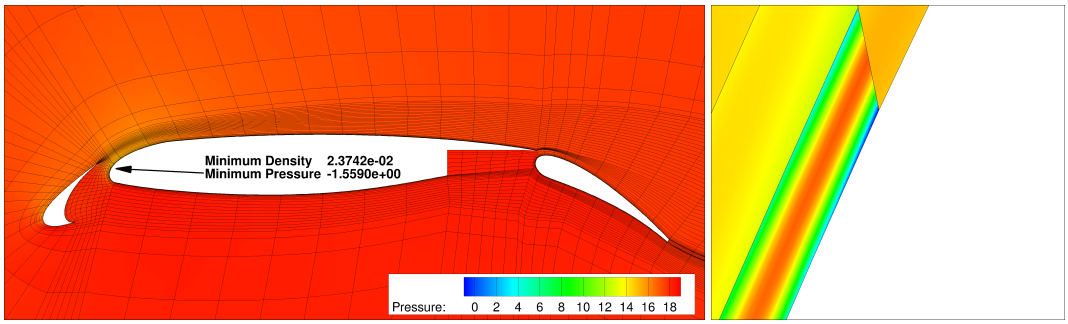
(a) Global view of the mesh mesh (4070 elements). (b) Mach contours. Thick dashed lines: $p = 1$; thick continuous lines: $p = 2$.

Figure 7. MDA 30p30n, $M_\infty = 0.2, \alpha = 16^\circ, Re = 9 \times 10^6$: quartic mesh used for κ_{LS} study.

DOI: 10.1002/nme

Table VI. MDA 30p30n, $M_\infty = 0.2, \alpha = 16^\circ, Re = 9 \times 10^6$: success of all runs for various κ_{LS}. C = converged; LM = local minimum; NP = nonphysical.

κ_{LS}	Success		Nonlinear iterations		GMRES iterations	
	$p=1$	$p=2$	$p=1$	$p=2$	$p=1$	$p=2$
			PTC			
0.9	C	NP	158	179	3913	2387
0.95	C	C	132	243	3453	40406
1.0	NP	C	142	268	411	46494
1.05	NP	C	198	287	1246	47205
1.1	NP	C	271	153	4512	31670
			CPTC			
0.9	C	LM	159	400	3748	59539
0.95	C	C	148	238	3360	37822
1.0	C	C	138	327	3640	53295
1.05	C	C	114	244	3487	43110
1.1	C	C	162	171	4353	35962



(a) Pressure distribution with location of constraint violation.

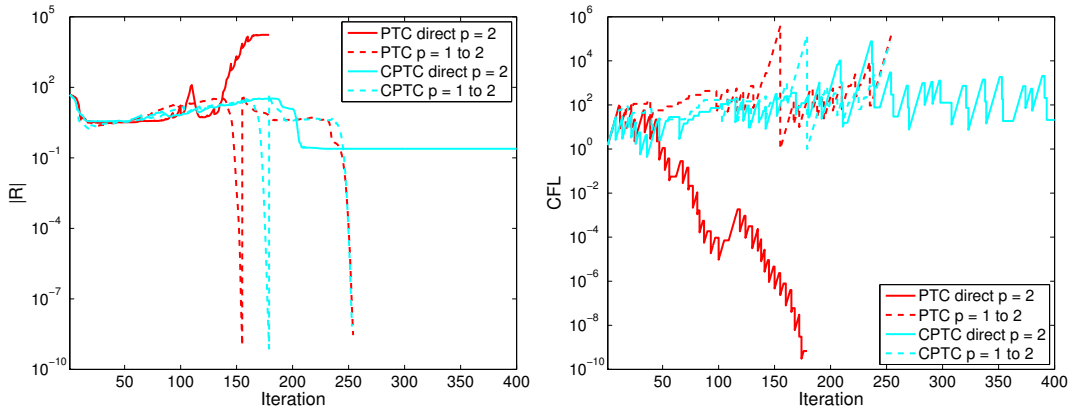
(b) Zoom on pressure constraint violation.

Figure 8. MDA 30p30n, $M_\infty = 0.2, \alpha = 16^\circ, Re = 9 \times 10^6, p = 1, \kappa_{\text{LS}} = 1.05$: PTC state iterate that violates physics constraints.

starting from free stream and solving directly for $p = 2$ against order continuation. In the latter case, we initialize the solution with free stream conditions and solve for a $p = 1$ solution, then we use this solution as a starting point for a $p = 2$ calculation. For an appropriate comparison, we re-solve the initial $p = 1$ flow using the same parameters used for direct $p = 2$ listed in Table IV. Note that both PTC and CPTC are successful with order continuation for this case.

As noted in the transonic case, Figure 9(a) shows that violating the physics constraints with PTC and direct $p = 2$ leads to the residual norm climbing several orders of magnitude. Note that since Armijo's rule is strictly enforced for the norm of the unsteady residual (Equation 13), the spatial residual norm only climbs because the unsteady term dominates the unsteady residual as the CFL is reduced (Figure 9(b)). Figure 9(a) also shows CPTC stagnating for direct $p = 2$ after the residual norm drops approximately 2 orders of magnitude.

DOI: 10.1002/nme



(a) Residual histories.

(b) CFL histories.

Figure 9. MDA 30p30n, $M_\infty = 0.2, \alpha = 16^\circ, Re = 9 \times 10^6, \kappa_{\text{LS}} = 0.90$: order continuation for PTC and CPTC.

7. CONCLUSIONS

We augmented the PTC method with nonlinear inequality constraints that enforce physically valid thermodynamic states. This augmentation is possible because the solution update direction at each nonlinear step is a descent direction for the unsteady residual. The challenge, however, is to penalize the residual in a computationally efficient manner. To this end, we use a vector penalization approach that does not increase the memory footprint of the residual Jacobian. The latter point is due to the local nature of the physics constraints.

We presented an example in which following the time-accurate path incurs in negative pressure. The residual penalties, as formulated here, are a natural mechanism to locally amplify the CFL and skip the transients that lead to non-physical states. Because this mechanism is selective and local, it does not affect the global convergence property of pseudo-transient continuation.

The results show that CPTC's success in reaching steady-state is significantly less sensitive to input parameters in comparison to its unconstrained counterpart. This property makes CPTC a good candidate for "hands-off" adaptive frameworks. The caveat is the linear systems at each non-linear step are generally more expensive with CPTC than with PTC. In some cases, this is compensated by fewer non-linear iterations.

We anticipate further improvements in the line-search algorithm, especially with regard to eliminating the κ_{LS} factor. Defining a general rule for relaxing Armijo's condition is a difficult task in non-linear problems and using gradient information in the line search would involve updating the residual Jacobian which is computationally expensive. Our choice here leans towards simplicity while maintaining reasonable robustness.

ACKNOWLEDGMENTS

The authors acknowledge the support given by the University of Michigan to the development of this work and appreciate the computational resources provided by NASA. Funding for this research is provided by the Air Force Office of Scientific Research under contract number FA9550-10-C-0040.

REFERENCES

1. Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, M. Visbal, High-order CFD methods: Current status and perspective, International Journal for Numerical Methods in FluidsDOI: 10.1002/fld.3767.
2. L. Wang, D. Mavriplis, Adjoint-based $h - p$ adaptive discontinuous Galerkin methods for the 2D compressible Euler equations, Journal of Computational Physics 228 (2009) 7643–7661.
3. M. Yano, J. Modisette, D. Darmofal, The importance of mesh adaptation for higher-order discretizations of aerodynamics flows, AIAA Paper 2011-3852 (2011).
4. M. Ceze, K. J. Fidkowski, Anisotropic *hp*-adaptation framework for functional prediction, AIAA Journal 51 (2) (2013) 492–509. doi:10.2514/1.J051845.
5. P.-O. Persson, J. Peraire, Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-Stokes equations, SIAM Journal for Scientific Computing 30 (6) (2008) 2709–2733.
6. C. T. Kelley, D. E. Keyes, Convergence analysis of pseudo-transient continuation, SIAM Journal on Numerical Analysis.
7. T. S. Coffey, C. T. Kelley, D. E. Keyes, Pseudo-transient continuation and differential-algebraic equations, Journal of Scientific Computing 25 (2).
8. C. T. Kelley, L.-Z. Liao, L. Qi, M. T. Chu, J. Reese, C. Winton, Projected pseudo-transient continuation, SIAM Journal on Numerical Analysis 46 (6) (2008) 3071–3083.
9. G. Buttazzo, A. Frediani, S. R. Allmaras, J. E. Bussoletti, C. L. Hilmes, F. T. Johnson, R. G. Melvin, E. N. Tinoco, V. Venkatakrishnan, L. B. Wigton, D. P. Young, Algorithm issues and challenges associated with the development of robust CFD codes, in: Variational Analysis and Aerospace Engineering, Vol. 33 of Springer Optimization and Its Applications, Springer New York, 2009, pp. 1–19.
10. J. V. Neumann, R. D. Richtmyer, A method for the numerical calculation of hydrodynamic shocks, Journal of Applied Physics 21 (1950) 232–237.
11. P.-O. Persson, J. Peraire., Sub-cell shock capturing for discontinuous Galerkin methods, in: 44th AIAA Aerospace Sciences Meeting and Exhibit, no. 2006-112, 2006.
12. J. E. Hicken, D. W. Zingg, Globalization strategies for inexact-Newton solvers, in: 19th AIAA Computational Fluid Dynamics, no. AIAA 2009-4139, 2009.
13. J. E. Hicken, H. Buckley, M. Osusky, D. W. Zingg, Dissipation-based continuation: a globalization for inexact-Newton solvers, in: 20th AIAA Computaional Fluid Dynamics Conference, no. AIAA 2011-3237, 2011.
14. K. J. Fidkowski, T. A. Oliver, J. Lu, D. L. Darmofal, *p*-multigrid solution of high–order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations, Journal of Computational Physics 207 (2005) 92–113.
15. K. J. Fidkowski, A high–order discontinuous Galerkin multigrid solver for aerodynamic applications, MS thesis, M.I.T., Department of Aeronautics and Astronautics (June 2004).
16. M. Ceze, K. J. Fidkowski, A robust adaptive solution strategy for high-order implicit CFD solvers, in: 20th AIAA Computaional Fluid Dynamics Conference, no. AIAA 2011-3696, AIAA, 2011.
17. D. E. Keyes, D. R. Reynolds, C. S. Woodward, Implicit solvers for large-scale nonlinear problems, Journal of Physics: Conference Series 46 (2006) 433–442.
18. P. L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, Journal of Computational Physics 43 (1981) 357–372.
19. F. Bassi, S. Rebay, GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations, in: K. Cockburn, Shu (Eds.), Discontinuous Galerkin Methods: Theory, Computation and Applications, Springer, Berlin, 2000, pp. 197–208.
20. S. R. Allmaras, F. T. Johnson, P. R. Spalart, Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model, in: Seventh Intenational Conference on Computational Fluid Dynamics (ICCFD7), 2012.
21. P. R. Spalart, S. R. Allmaras, A one-equation turbulence model for aerodynamic flows, in: 30th Aerospace Sciences Meeting and Exhibit, no. AIAA-92-0439, AIAA, 1992.
22. M. Ceze, K. J. Fidkowski, Pseudo-transient Continuation, Solution Update Methods, and CFL Strategies for DG Discretizations of the RANS-SA Equations, in: 21th AIAA Computaional Fluid Dynamics Conference, 2013.
23. Y. Saad, M. H. Schultz, Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on Scientific Computing 7 (3) (1986) 856–869.
24. Y. Saad, A flexible inner-outer preconditioned gmres algorithm, SIAM Journal on Scientific Computing 14 (2) (1993) 461–469.
25. D. Knoll, D. Keyes, Jacobian-free Newton-Krylov methods: a survey of approaches and applications, Journal of Computational Physics 193 (2004) 357–397.
26. H. M. Bucker, B. Pollul, A. Rasch, On CFL evolution strategies for implicit upwind methods in linearized Euler equations, International Journal for Numerical Methods in Fluids 59 (2009) 1–18.
27. W. A. Mulder, B. van Leer, Experiments with implicit upwind methods for the Euler equations, Journal of Computational Physics.
28. H. L. Atkins, A. Pampell, Robust and accurate shock capturing method for high-order discontinuos Galerkin methods, in: 20th AIAA Computaional Fluid Dynamics Conference, 2011.
29. J. Hartung, A stable interior penalty method for convex extremal problems, Numerische Mathematik 29 (2).
30. C. T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, 1995.
31. J. M. Modisette, An automated reliable method for two-dimensional Reynolds-averaged Navier-Stokes simulations, PhD dissertation, Massachusetts Institute of Technology (2011).
32. L. Armijo, Minimization of functions having Lipschitz continuous first partial derivatives, Pacific Journal of Mathematics 16 (1).
33. M. d. Ceze, A robust *hp*-adaptation method for discontinuous galerkin discretizations applied to aerodynamic flows, Ph.D. thesis, The University of Michigan (2013).
34. T. T. Chisholm, D. W. Zingg, A Jacobian-free Newton-Krylov algorithm for compressible turbulent fluid flows, Journal of Computational Physics (228) (2009) 3490–3507.

35. M. Ceze, K. J. Fidkowski, Drag prediction using adaptive discontinuous finite elements, Journal of Aircraft Accepted.

Prepared using nmeauth.cls
DOI: 10.1002/nme