INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2015; **102**:1683–1703
Published online 17 March 2015 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nme.4858

Constrained pseudo-transient continuation

Marco Ceze*,† and Krzysztof J. Fidkowski

Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA

SUMMARY

This paper addresses the problem of finding a stationary point of a nonlinear dynamical system whose state variables are under inequality constraints. Systems of this type often arise from the discretization of PDEs that model physical phenomena (e.g., fluid dynamics) in which the state variables are under realizability constraints (e.g., positive pressure and density). We start from the popular pseudo-transient continuation method and augment it with nonlinear inequality constraints. The constraint handling technique does not help in situations where no steady-state solution exists, for example, because of an under-resolved discretization of PDEs. However, an often overlooked situation is one in which the steady-state solution exists but cannot be reached by the solver, which typically fails because of the violation of constraints, that is, a non-physical state error during state iterations. This is the shortcoming that we address by incorporating physical realizability constraints into the solution path from the initial condition to steady state. Although we focus on the DG method applied to fluid dynamics, our technique relies only on implicit time marching and hence can be extended to other spatial discretizations and other physics problems. We analyze the sensitivity of the method to a range of input parameters and present results for compressible turbulent flows that show that the constrained method is significantly more robust than a standard unconstrained method while on par in terms of cost. Copyright © 2015 John Wiley & Sons, Ltd.

Received 4 June 2014; Revised 29 October 2014; Accepted 27 November 2014

KEY WORDS: nonlinear solvers; Newton methods; discontinuous Galerkin; aerodynamics; transonic; aerodynamics; subsonic

1. INTRODUCTION

Nonlinear dynamical systems of the type $\dot{\mathbf{U}} = \mathbf{F}(\mathbf{U})$, $\mathbf{U}(0) = \mathbf{U}_0$ often occur in computational physics when one employs the method of lines to discretize a system of PDEs. When a steady solution is desired, one may remove the time dependence and use Newton's method to solve $\mathbf{F}(\mathbf{U}) = 0$. However, for highly nonlinear problems, direct application of Newton's method often fails by one of the two modes: (i) it converges to a local minimum of $|\mathbf{F}(\mathbf{U})|_2$ or (ii) it violates physical bounds on the state, for example, ones that state that pressure, density, or species concentrations must remain positive. A popular choice for globalizing Newton's method is pseudo-transient continuation (PTC) [1–3], in which the time dependence is retained even when seeking steady state in order to make the state follow a hopefully physically valid trajectory from initial condition to the 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. However, this argument assumes that the discretization is a good approximation to the exact differential operator, and this may not be known *a priori*. Furthermore, certain solution features such as shocks, expansions, and shear layers can occur during

*Correspondence to: Marco Ceze, Department of Aerospace Engineering, University of Michigan, Ann Arbor, MI 48109, USA.
†E-mail: mceze@umich.edu

the pseudo-transient but not be present in the final converged solution. Depending on the discretization scheme, those features can cause numerical oscillations, and resolving them temporally can lead to the violation of physical constraints.

Dynamical systems of nonlinear equations resulting from high-order discretizations are generally much more difficult to solve compared with those obtained from, for example, second-order finite volume methods. Moreover, when high-order methods are used in combination with mesh adaptation [4–6], 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.

Remediation procedures for keeping the state iterates within their physical bounds often rely on reducing the state update size. In addition to slowing down convergence, these procedures do not avoid attractive regions of state space—valleys of $|\mathbf{F}(\mathbf{U})|_2$—that violate the physics constraints. Other procedures such as locally modifying the state or the discrete residual operators are also used in practice, but these methods are often specific to a single discretization scheme. One such example is the use of well-established limiters in second-order finite volume methods, which substantially increase the robustness of those 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, lack of robustness for high order, and lack of a zero-residual solution. An alternative remedy is the introduction of artificial viscosity [7, 8], 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 parameter/order/boundary-condition sequencing [9–12]. The 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 [13]. Other approaches for mitigating the effects of under-resolved features, specific to finite element discretizations, include Petrov–Galerkin formulations, in which the test space can be tailored to improve stability in the presence of discontinuous features [14–16].

In the present work, we propose an alternative approach to improve solver robustness, one that is often less invasive relative to limiting or even artificial viscosity and 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 solution framework.

The setting for our work is the 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. Although our presentation assumes DG and RANS, the ideas presented in the paper can be extended to other 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 PTC with constraints. Section 5 discusses the 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

Consider the convection–diffusion source equation in compact, conservative form

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

where $\mathcal{C}_{is}$ and $\mathcal{D}_{is}$ are the convective and diffusive fluxes, respectively, $\mathcal{S}_s$ is a source term, $i \in [1, .., \dim]$ indexes the spatial dimensions, and s indexes the conservation equations. Here, we will focus on the RANS equations with the Spalart–Allmaras (SA) turbulence model. We represent the state 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 DG spatial discretization of the physics equations approximates the solution in a space $\mathcal{V}^{H,p}$ of piecewise polynomials of degree p with local support on each element $\kappa^H \in K^H$, where K^H is the set of elements resulting from a subdivision of the spatial domain. The resulting weak form reads as follows:

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

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

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

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.

Discontinuous Galerkin 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 number of faces on the two adjacent elements. However, for increased robustness, we scale this number by a 'stabilization augmentation factor', $\kappa_{\mathrm{BR2}} > 1$.

2.2. *Physics constraints*

The state, $\mathbf{u}^H$, 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 as follows:

$$c_i\left(\mathbf{u}^H\left(\mathbf{x}'_q\right)\right) = \begin{cases} \frac{p\left(\mathbf{u}^H(t,\mathbf{x})\right)}{p_\infty} > 0, \\ \frac{\rho\left(\mathbf{u}^H(t,\mathbf{x})\right)}{\rho_\infty} > 0, \end{cases} \tag{5}$$

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, and only for a linear variation of states can one guarantee that the pressure constraint will be violated at one of the end points as the interior extremum in this case can only be a maximum [22]. In this absence of a closed-form constraint condition for arbitrary state distributions on the discrete state vector, Equation 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, that intuition dictates that the eddy viscosity should be constrained similarly to pressure and density, that is, $\nu_t > 0$ and the modifications in [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

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

$$\mathbf{T}^n(\mathbf{U}^{n+1} - \mathbf{U}^n) + \mathbf{R}(\mathbf{U}^{n+1}) = 0, \tag{6}$$

where n indexes the time steps. $\mathbf{T}^n$ is a block diagonal matrix with elemental blocks given by

$$\mathbf{T}^n_{\kappa^H} = \mathbf{M}_{\kappa^H}\frac{1}{\Delta t^n_{\kappa^H}}, \tag{7}$$

where $\mathbf{M}_{\kappa^H}$ is an element's mass matrix and $\Delta t^n_{\kappa^H}$ is the nth time step, defined later in this section.

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

$$\left(\mathbf{T}^k + \left.\frac{\partial\mathbf{R}}{\partial\mathbf{U}}\right|_{\mathbf{U}^k}\right)\Delta\mathbf{U}^k = -\mathbf{R}(\mathbf{U}^k), \tag{8}$$

where k is used for the nonlinear iteration number to distinguish the method from the backward Euler case. Note that when $\Delta t_{\kappa^H} \to \infty$ for all κ^H, the iterative procedure of Equation 8 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 [12], solves the linear system at each step to a relative tolerance, η_l. Here, $10^{-8} \leqslant \eta_l \leqslant 10^{-2}$ is the ratio between the final and the initial linear residual norms

$$\eta_{l,i} = \frac{\left|\left(\mathbf{T}^k + \frac{\partial\mathbf{R}}{\partial\mathbf{U}}|_{\mathbf{U}^k}\right)\Delta\mathbf{U}^k_i + \mathbf{R}(\mathbf{U}^k)\right|_2}{|\mathbf{R}(\mathbf{U}^k)|_2}, \tag{9}$$

where $\Delta\mathbf{U}_i^k$ is the i-th GMRES iteration for Equation 8 at the k-th nonlinear iteration. Note that we initialize the linear solves with $\Delta\mathbf{U}_0^k = 0$.

The DG discretization described in Section 2 produces a residual Jacobian that is block-sparse, which means that DOFs in an element are coupled only to DOFs in the 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 because of the propagation of boundary information into the domain. To alleviate those transients and aid the linear solves, small time steps are used. This causes a diagonal dominance in the coefficient matrix in Equation 8 and makes the calculation closer to time-accurate if Δt_{κ^H} is the same for all elements. As an alternative to global time stepping, element-wise local time steps can be used by setting a global CFL number and then calculating different time steps on each element according to

$$\Delta t_{\kappa^H} = \mathrm{CFL}\,\frac{L_{\kappa^H}}{\lambda_{\kappa^H}^{\max}}, \tag{10}$$

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

$$\mathbf{U}^{k+1} = \mathbf{U}^k + \omega^k \Delta\mathbf{U}^k. \tag{11}$$

The value of the under-relaxation parameter is typically set based on a user-prescribed maximum allowable variation of the constrained physical quantities such as pressure and density. We discuss how we set ω^k in Section 5.

Alternatively, PTC can be interpreted as a globalization strategy for Newton's method [25] where a series of problems defined by Equations 8 and 11 are solved for $k = 1, 2, \ldots,$ 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 PTC 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 Equation 8 becomes Newton's method, and the state approaches the steady solution.

Many strategies for evolving the CFL are available [1, 26]. 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,

$$\mathrm{CFL}^{k+1} = \begin{cases} \beta\cdot\mathrm{CFL}^k \quad \text{for } \beta > 1 & \text{if} \quad \omega^k = 1 \\ \mathrm{CFL}^k & \text{if} \quad \omega_{\min} < \omega^k < 1 \\ \kappa\cdot\mathrm{CFL}^k \quad \text{for } \kappa < 1 & \text{if} \quad \omega^k < \omega_{\min} \end{cases}. \tag{12}$$

Here, we set the parameters to $\omega_{\min} = 0.01$, $1.05 \leqslant \beta \leqslant 2.0$, 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 because 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 that the matrix in front of $\Delta\mathbf{U}^k$ in Equation 8 (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 Equation 8 by its own transpose gives

$$\Delta\mathbf{U}^{k^T}\mathbf{A}^T \underbrace{\left(\mathbf{T}^k + \left.\frac{\partial\mathbf{R}}{\partial\mathbf{U}}\right|_{\mathbf{U}^k}\right)}_{\mathbf{A}} \Delta\mathbf{U}^k = -\Delta\mathbf{U}^{k^T} \underbrace{\mathbf{A}^T\mathbf{R}(\mathbf{U}^k)}_{\left.\frac{\partial f}{\partial\mathbf{U}}\right|_{\mathbf{U}^k}} > 0, \tag{13}$$

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 Equation 13. This function is

$$f(\tilde{\mathbf{U}}) = \frac{1}{2}|\mathbf{R}_t(\tilde{\mathbf{U}})|^2_{L_2} = \frac{1}{2}\mathbf{R}_t(\tilde{\mathbf{U}})^T\mathbf{R}_t(\tilde{\mathbf{U}}). \tag{14}$$

where the unsteady residual is defined by

$$\mathbf{R}_t(\tilde{\mathbf{U}}) \equiv \mathbf{T}^k(\tilde{\mathbf{U}} - \mathbf{U}^k) + \mathbf{R}(\tilde{\mathbf{U}}), \tag{15}$$

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 because 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.* [3]. Their approach involves a step that projects the state into the feasible domain after each nonlinear 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 [13]. 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 [13]. The augmented residual is

$$\mathbf{R}_p(\mathbf{U}) \equiv \mathbf{R}(\mathbf{U}) + \mathbf{P}(\mathbf{U}, \mu_{\mathrm{P}}), \tag{16}$$

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

$$\mathbf{P}(\mathbf{U}, \mu_{\mathrm{P}}) \equiv \mathbf{\Phi}(\mathbf{U}, \mu_{\mathrm{P}})\, \mathbf{R}(\mathbf{U}), \tag{17}$$

where $\mathbf{\Phi}$ is a diagonal matrix of the same size as the residual Jacobian. Each sub matrix of $\mathbf{\Phi}$ corresponding to an element κ^H is given by

$$\mathbf{\Phi}_{\kappa^H}(\mathbf{U}_{\kappa^H}, \mu_{\mathrm{P}}) = \mu_{\mathrm{P}}\, \mathbb{P}_{\kappa^H}(\mathbf{U}_{\kappa^H}) \cdot \mathbf{I}, \tag{18}$$

where $\mathbf{I}$ is the identity matrix of the same size as the element's mass matrix, and $\mathbf{U}_{\kappa^H}$ is the piece of the global discrete state vector corresponding to κ^H. The elemental penalty, $\mathbb{P}_{\kappa^H}$, is a barrier function that imposes the constraints similar to interior penalty optimization methods. Here, we consider an inverse barrier where the penalty function is the sum of inverses of the constraints, c_i, from Equation 5. Because the constraints are applied to a functional representation of the 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

$$\mathbb{P}_{\kappa^H}(\mathbf{U}_{\kappa^H}) = \sum_{i}^{N_c} \sum_{q}^{N'_q} \frac{w'_q}{c_i(\mathbf{u}^H(\mathbf{x}'_q))}, \tag{19}$$

where N'_q is the number of quadrature points $\mathbf{x}'_q$ with weights w'_q, and N_c is the number of constraints indexed by i. Note that $\mathbb{P}_{\kappa^H}$ tends to infinity as the constraints approach zero from the positive side.

Equation 19 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 Equation 17—onto the residual vector is always positive for nonzero $\mathbf{R}$ because the elemental penalties are strictly positive in the feasible domain, that is, 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 of the solution, not the solution itself. To derive the update equation for CPTC, we apply Equation 8 to $\mathbf{R}_p$ and, for clarity, we switch to tensor notation, where the subscripts index the tensor entries and the superscripts denote the iteration number

$$\left(T_{ij} + \left.\frac{\partial((\delta_{il} + \Phi_{il})R_l)}{\partial U_j}\right|_{\mathbf{U}^k} \right) \Delta U_j^k = -(\delta_{il} + \Phi_{il}^k) R_l(\mathbf{U}^k), \tag{20}$$

where δ_{il} denotes the Kronecker's delta. Multiplying Equation 20 by $(\delta_{il} + \Phi_{il}^k)^{-1}$ and expanding the derivatives give

$$\left(\underbrace{(\delta_{il}+\Phi^k_{il})^{-1}T_{ij}}_{\text{globalization matrix}}+\frac{\partial R_l}{\partial U_j}\Big|_{\mathbf{U}^k}+\underbrace{(\delta_{il}+\Phi^k_{il})^{-1}\left(\frac{\partial \Phi_{im}}{\partial U_j}\Big|_{\mathbf{U}^k}R_m(\mathbf{U}^k)\right)}_{\text{penalization matrix}}\right)\Delta U^k_j=-R_l(\mathbf{U}^k). \tag{21}$$

The term $\frac{\partial \Phi_{im}}{\partial U_j}$ is a 3rd-order tensor operator on the state vector $\mathbf{U}$—see Appendix A for derivation. At each step, this tensor is evaluated with the current state, $\mathbf{U}^k$, and contracted by $R_m(\mathbf{U}^k)$ resulting in a matrix the same size as the residual Jacobian. Separating the terms such that the unpenalized residual, $\mathbf{R}$, is on the right-hand side 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 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 $\mathbf{I}+\mathbf{\Phi}^k$ 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, whereas the penalization term does not vanish because the function value of inverse barrier penalties (Equation 19) 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 Equation 21 as 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 that the state is initialized by uniform free-stream conditions, we can equate the coefficients multiplying the globalization and penalization matrices

$$\frac{1}{(1+\mu^0_{\mathrm{P}}\mathbb{P}_0)\cdot\mathrm{CFL}^0}=\frac{\mu^0_{\mathrm{P}}}{(1+\mu^0_{\mathrm{P}}\mathbb{P}_0)}\Rightarrow\mu^0_{\mathrm{P}}=\frac{1}{\mathrm{CFL}^0}. \tag{22}$$

μ_{P} in the numerator of the right-hand side of Equation 22 comes from $\partial\Phi_{im}/\partial U_j$ in Equation 21, and CFL^0 in the left-hand side is factored out of the elemental time step. As for PTC, in Equation 22, we assume that the residuals are properly scaled so that a single-CFL time continuation globalizes all of the equations.

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 SER for μ_{P}

$$\mu^{k+1}_{\mathrm{P}}=\mu^k_{\mathrm{P}}\frac{1+\mu^k_{\mathrm{P}}\langle\mathbb{P}_{\kappa H}(\mathbf{U}^k)\rangle}{1+\mu^{k-1}_{\mathrm{P}}\langle\mathbb{P}_{\kappa H}(\mathbf{U}^{k-1})\rangle}, \tag{23}$$

where $\langle\cdot\rangle$ indicates an average over all the elements. The evolution strategy in Equation 23 makes the solver acknowledge the presence of a feasibility constraint by increasing its repelling effect as the solution path goes toward 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 lead 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 method 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 Equation 8. 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, $\varepsilon_{\text{res}}$
 2: Choose initial CFL$^0$ and its increase factor $\beta$ (Equation 12)
 3: Initialize $\mu_{\text{P}}^0$ according to Equation 22
 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 Equation 21 using GMRES
 7:     Compute under-relaxation parameter $\omega^k$ (Section 5)
 8:     if $\omega^k \geqslant \omega_{\min}$ 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 $\mu_{\text{P}}^{k+1}$ with Equation 23
17:     Update CFL$^{k+1}$ with Equation 12
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, $\mathbf{x}_m$. This involves evaluating the basis functions at $\mathbf{x}_m$ and using the discrete vectors $\mathbf{U}$ and $\Delta\mathbf{U}$ to yield the field representations, $\mathbf{u}^H(\mathbf{x})$ and $\Delta\mathbf{u}^H(\mathbf{x})$. 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.

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 $\mathbf{x}_m$, 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 13 of Algorithm 2 because pressure is a nonlinear function of the state.

Note that Algorithm 2 is a limiting procedure for compressible flow, but the same idea can be applied to other problems with either linear or nonlinear constraints—density and pressure, respectively.

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 PTC (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 $|\mathbf{R}_t|_{L_2}^2$. Here, we relax Armijo's rule by a factor $\kappa_{\text{LS}} > 1$ and we select the initial step size as the minimum $\omega_{\kappa H}$ over all the elements. The effect of κ_{LS} is discussed in Section 6.2. Algorithm 3 summarizes the line-search procedure.

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 Equation 16, and hence, $\mathbf{R}$ is replaced by $\mathbf{R}_p$.

Algorithm 2 Limit physical update

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

Algorithm 3 Line search

```
1: $\omega_{\text{phys}} \leftarrow 1$                              ▷ Initial guess for physical update
2: for all $\kappa^H$ do
3:     Select limit points, $\mathbf{x}_m$
4:     Evaluate $\mathbf{u}^{H,p}(\mathbf{x}_m)|_{\kappa^H}$ and $\Delta\mathbf{u}^{H,p}(\mathbf{x}_m)|_{\kappa^H}$
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:     $\omega^k \leftarrow \frac{\omega^k}{2}$
12:     $\tilde{\mathbf{U}} \leftarrow \mathbf{U}^k + \omega^k\Delta\mathbf{U}^k$
13: end while
14: return $\omega^k$
```

In references [22, 31, 33], 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 SA 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,

Algorithm 4 Greedy algorithm

```
1: if $\omega^k = \omega_{\text{phys}}$ then
2:     while $\omega^k \leqslant 1$ do
3:         $\omega^k \leftarrow \beta_\omega \cdot \omega^k$                    $\triangleright$ 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:             $\omega^k \leftarrow \frac{\omega^k}{2}$
7:             return $\omega^k$
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:            $\omega^k \leftarrow \frac{\omega^k}{\beta_\omega}$
11:            return $\omega^k$
12:        end if
13:    end while
14: end if
15: return $\omega^k$
```

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 earlier 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 as follows.

Because the greedy algorithm is an extension of the line search, the same remarks made earlier is applied 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

The first test case is a flow problem that has a physical steady solution but in which a time-accurate integration results in the violation of the physics constraints. It consists of a one-dimensional Euler shock tube in the domain $x = [-1, 1]$, where the boundary conditions on both ends of the tube are flows 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 the outer boundaries ensuring 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 go well, the flow settles to a steady state equal to the boundary condition.

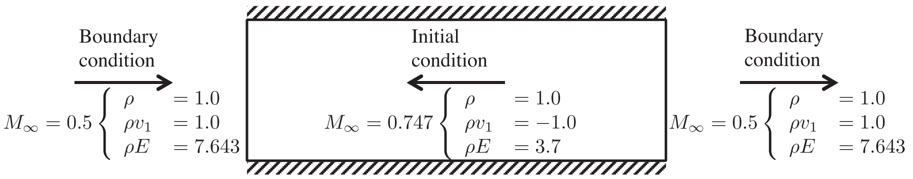
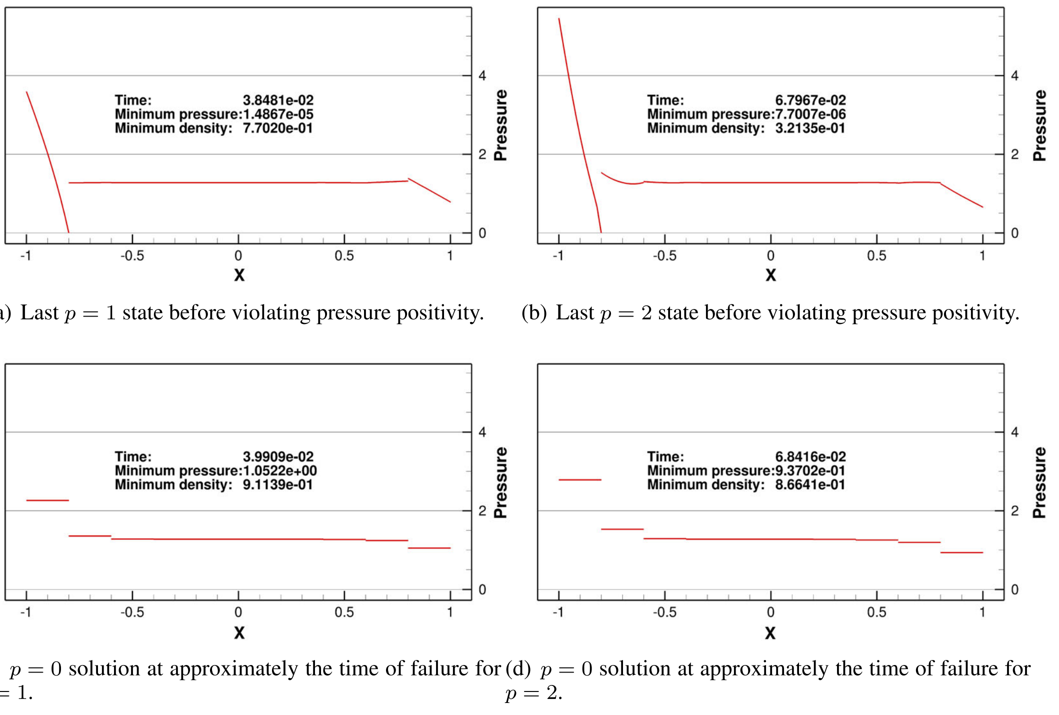

Boundary condition
$M_\infty = 0.5 \begin{cases} \rho &= 1.0 \\ \rho v_1 &= 1.0 \\ \rho E &= 7.643 \end{cases}$
Initial condition
$M_\infty = 0.747 \begin{cases} \rho &= 1.0 \\ \rho v_1 &= -1.0 \\ \rho E &= 3.7 \end{cases}$
Boundary condition
$M_\infty = 0.5 \begin{cases} \rho &= 1.0 \\ \rho v_1 &= 1.0 \\ \rho E &= 7.643 \end{cases}$

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



(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: (a) Last $p = 1$ state before violating pressure positivity; (b) Last $p = 2$ state before violating pressure positivity; (c) $p = 0$ solution at approximatety the time of failure for $p = 1$; and (d) $p = 0$ solution at approximatety the time of failure for $p = 2$;

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	$\beta = 1.05,\ 1.5,\ 2$

To draw a parallel to PTC, we advance the discrete solution in time with the backward Euler scheme with a domain-wide constant time step for which the maximum CFL is 0.1. Although the piecewise constant ($p = 0$) spatial solution approximation has no problem reaching the expected steady state, piecewise linear and quadratic ($p = 1, 2$) approximations violate the pressure constraint in their transients (Figure 2). These violations are caused by oscillations triggered by the shock moving into the domain from the left side. The schemes' intrinsic dissipation levels are not enough to dampen those oscillations, and some form of explicit artificial dissipation should be considered.

The purpose here is to assess the ability of PTC and CPTC to skip the non-physical transients and to reach the steady state. The assessment considers a range of mesh resolutions, approximation orders, initial CFL, and CFL growth factors (β in Equation 12). 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 $\kappa_{\mathrm{LS}} = 1.05$ for all runs. The nonlinear residual convergence tolerance is 10^{-8}.

Because the solution transient undergoes a shock, we compare the PTC 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 a method similar to Persson and Peraire's [8] shock-capturing scheme for which solution regularity is sensed by density changes between the current p^{th}-order solution and its projection onto $\mathcal{V}^{H,p-1}$. The artificial viscosity is computed from the regularity indicator via a nonlinear but smoothly varying switch and the diffusion term discretized with BR2 for which $\kappa_{\text{BR2}} = 1$. Note that $p = 0$ runs do not include any shock-capturing term.

We consider CPTC in two modes. One where the penalty factor, μ_{P}, varies according to Equation 23 and another where we keep its value constant. In both modes, we initialize μ_{P} according to Equation 22. 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 ($\leqslant 3\%$) on the method's success rate.

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, from 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.

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

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} (%)
Without artificial diffusion	66.33	91.67	88.67
With artificial diffusion	89.67	96.00	94.67

PTC, pseudo-transient continuation; CPTC, constrained pseudo-transient continuation.

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

PTC, pseudo-transient continuation; CPTC, constrained pseudo-transient continuation; GMRES, generalized minimal residual.

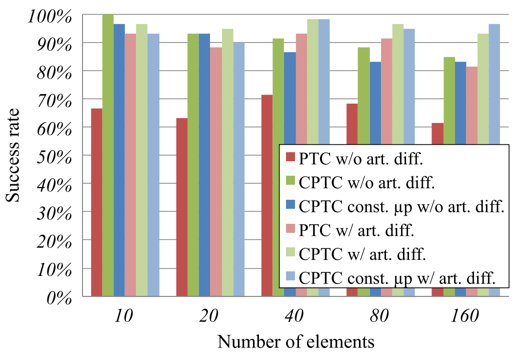
class while marginalizing the other classes—for example, success of all runs with $\mathrm{CFL}^0 = 1.0$. Note that the average of the success rates over all parameter values within a class recovers the global success rate.

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 Figure 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 $\mathrm{CFL}^0 > 1$. Increasing the CFL, however, is not a selective mechanism as it washes all transients and it can affect the globalization character of PTC.

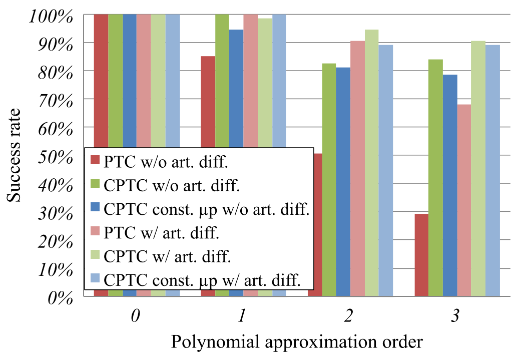
Constrained PTC's success without artificial diffusion decreases with increasing mesh resolution for this flow problem (Figure 3(a)). The reason for this behavior is that the shock becomes steeper as the mesh gets finer, and in the absence of the shock-capturing term, the pressure undershoots to lower 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}*

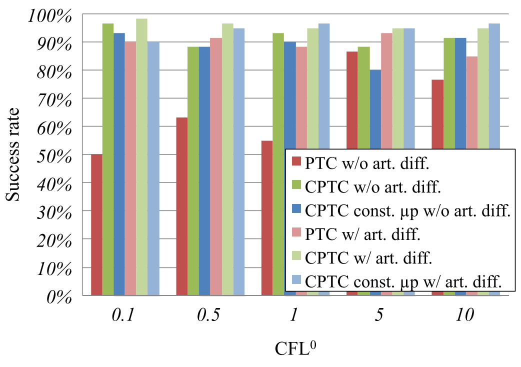
We now analyze the effect of relaxing Armijo's rule on the success rate of the solver. In order to properly exercise both PTC and CPTC methods, we choose two turbulent flows in which DG



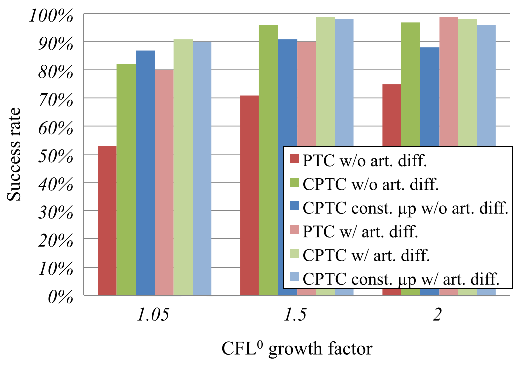
(a) Runs divided in different mesh resolutions.



(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: (a) Run divided in different mesh resolution; (b) Run divided in different p-orders; (c) Run divided in different CFL^0; (d) Run divided in different CFL growth factors;

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 RAE 2822 airfoil. The steady solution for 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 [8]. 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 $M_\infty = 0.2, \alpha = 16°, Re = 9 \times 10^6$ 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 $|\mathbf{R}| < 10^{-8}$.

The mesh chosen for the transonic case (Figure 4) is publicly provided by the high-order workshop committee [36]. Here, our purpose in choosing this mesh 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 are 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

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

p-order	κ_{BR2}	CFL^0	β	η_l	GMRES vectors
RAE 2822, $M_\infty = 0.734, \alpha = 2.79°, Re = 6.5 \times 10^6$					
$p = 1$	15.0	1.0	2.0	10^{-3}	80
$p = 2$	10.0	1.0	2.0	10^{-3}	80
MDA 30p30n, $M_\infty = 0.2, \alpha = 16°, Re = 9 \times 10^6$					
$p = 1$	2.0	1.0	1.5	10^{-3}	80
$p = 2$	2.0	1.0	1.5	10^{-8}	100

GMRES, generalized minimal residual.

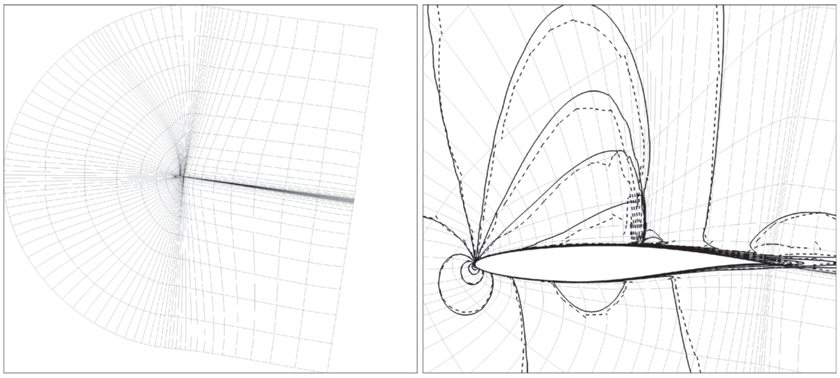
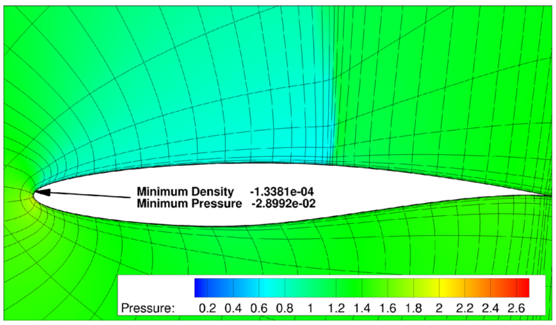

(a) Global view of the mesh mesh (2024 elements). (b) Mach contours. Thick dashed lines: $p = 1$; thick continuous lines: $p = 2$.

Figure 4. RAE 2822, $M_\infty = 0.734, \alpha = 2.79°$, $Re = 6.5{\times}10^6$: quartic mesh used for κ_{LS} study: (a) Global view of the mesh mesh (2024 element); (b) Mach contours. Thick dashed line: $p = 1$; thick continuous lines: $p = 2$.;

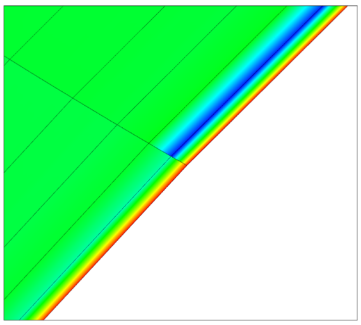
Table V. RAE 2822, $M_\infty = 0.734, \alpha = 2.79°$, $Re = 6.5 \times 10^6$: success of all runs for various κ_{LS}.

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

C, converged; LM, local minimum; NP, non-physical; GMRES, generalized minimal residual; PTC, pseudo-transient continuation; CPTC, constrained pseudo-transient continuation.



(a) Pressure distribution with location of constraint violation.



(b) Zoom on pressure constraint violation.

Figure 5. RAE 2822, $M_\infty = 0.734, \alpha = 2.79°, Re = 6.5 \times 10^6, p = 1, \kappa_{\mathrm{LS}} = 0.95$: PTC state iterate that violates physics constraints: (a) Pressure distribution with location of constraint violation; (b) Zoom on pressure constraint violation;

enforcing Armijo's rule ($\kappa_{\mathrm{LS}} < 1.0$) makes the solver resolve certain transients that lead to violating physical realizability (Figure 5). The exception here is the $p = 2$, $\kappa_{\mathrm{LS}} = 0.9$ 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 $\kappa_{\mathrm{LS}} = 0.95$. 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 two orders of magnitude (Figure 6(a)), which demonstrates that non-physical states can occur not only during the initial transients but also 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 [36].

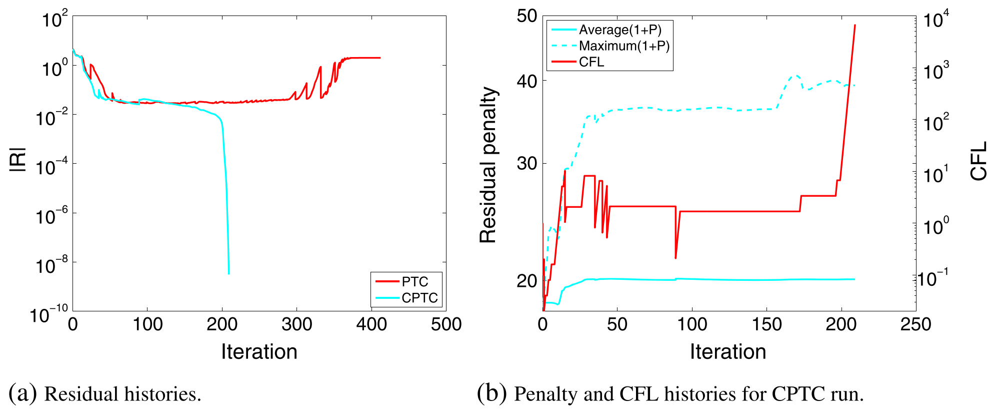


Figure 6. RAE 2822, $M_\infty = 0.734, \alpha = 2.79°$, $Re = 6.5 \times 10^6$, $p = 1$: PTC versus CPTC for $\kappa_{LS} = 0.95$: (a) Residual histories; (b) Penalty and CFL histories for CPTC run;

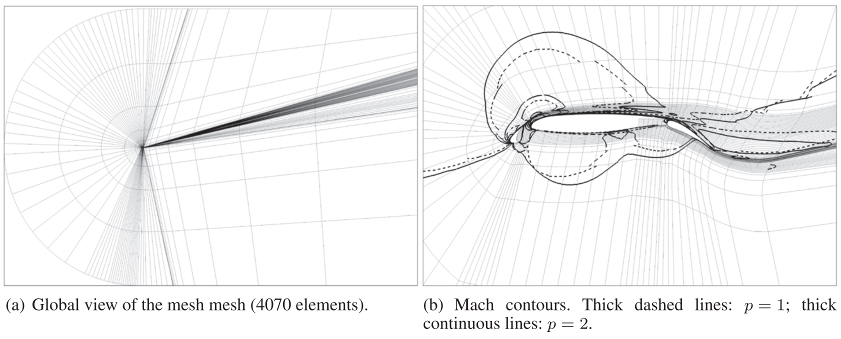

(a) Global view of the mesh mesh (4070 elements).
(b) Mach contours. Thick dashed lines: $p = 1$; thick continuous lines: $p = 2$.

Figure 7. MDA 30p30n, $M_\infty = 0.2, \alpha = 16°$, $Re = 9 \times 10^6$: quartic mesh used for κ_{LS} study: (a) Global view of the mesh mesh (4070 element); (b) Mach contours. Thick dashed line: $p = 1$; thick continuous lines: $p = 2$;

Table VI compares the success of PTC and CPTC for the high-lift case. First we remark that, similar to the transonic case, the added spatial resolution of $p = 2$ reduces the number of non-physical 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 $\kappa_{LS} = 0.9$ for $p = 2$, with which both methods fail to converge.

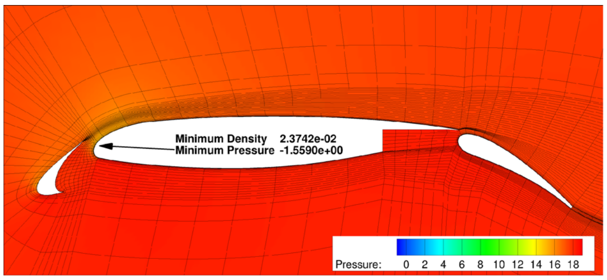
Given that both methods are successful with $\kappa_{LS} = 0.9$ and $p = 1$ but not with $p = 2$, we investigate if order continuation is successful in this condition. Figure 9 compares PTC and CPTC starting from free stream and solving directly for $p = 2$ against order continuation. In the latter case, 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 resolve 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 because Armijo's rule is strictly enforced for the norm of the unsteady residual (Equation 15), the spatial

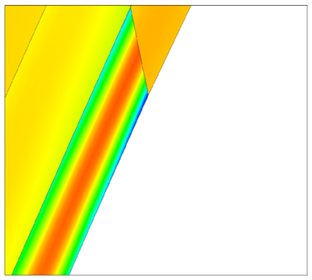
Table VI. MDA 30p30n, $M_\infty = 0.2, \alpha = 16°, Re = 9 \times 10^6$: success of all runs for various κ_{LS}.

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

C, converged; LM, local minimum; NP, non-physical; GMRES, generalized minimal residual; PTC, pseudo-transient continuation; CPTC, constrained pseudo-transient continuation.

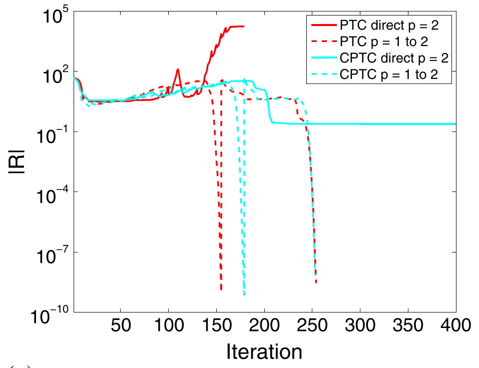

Minimum Density 2.3742e-02
Minimum Pressure -1.5590e+00
Pressure: 0 2 4 6 8 10 12 14 16 18

(a) Pressure distribution with location of constraint violation.

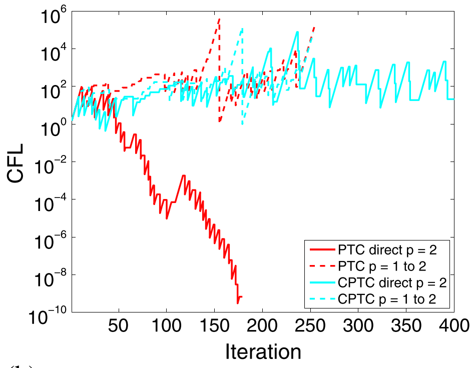


(b) Zoom on pressure constraint violation.

Figure 8. MDA 30p30n, $M_\infty = 0.2, \alpha = 16°, Re = 9 \times 10^6, p = 1, \kappa_{\mathrm{LS}} = 1.05$: PTC state iterate that violates physics constraints: (a) Pressure distribution with location of constraint violation; (b) Zoom on pressure constraint violation;



(a) Residual histories.



(b) CFL histories.

Figure 9. MDA 30p30n, $M_\infty = 0.2, \alpha = 16°, Re = 9 \times 10^6, \kappa_{\mathrm{LS}} = 0.90$: order continuation for PTC and CPTC: (a) Residual histories; (b) CFL histories;

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 two orders of magnitude.

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 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 PTC.

The results show that CPTC's success in reaching steady state is significantly less sensitive to input parameters in comparison with its unconstrained counterpart. This property makes CPTC a good candidate for 'hands-off' adaptive frameworks. The caveat is that the linear systems at each nonlinear step are generally more expensive to solve for CPTC than for PTC. In some cases, this is compensated by fewer nonlinear 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 nonlinear problems, and using gradient information in the line search would involve updating the residual Jacobian which is computationally expensive. Our choice here leans toward simplicity while maintaining reasonable robustness.

APPENDIX

A. DERIVATION OF THE PENALTY GRADIENT TENSOR

This appendix presents the derivation of the term $\frac{\partial \Phi_{ij}}{\partial U_k}$ in Equation 21. For clarity, we use tensor notation with Einstein's convention in which the components of the discrete state vector are represented as

$$U_k = \sum_{\kappa^H \in K^H} \mathbb{U}_{sb} V^{\kappa^H}_{sbk}, \tag{A.1}$$

where s and b index the conserved state components and basis functions, respectively. $V^{\kappa^H}_{sbk}$ is a bookkeeping tensor that converts the $N_s \times N_b$ unknowns in element local numbering to the global indexed by k. The element's state is spatially represented via the basis functions $\phi^H(\mathbf{x})$ as follows

$$\mathrm{u}^H_s(\mathbf{x}) = \mathbb{U}_{sb} \phi^H_b(\mathbf{x}). \tag{A.2}$$

The penalty matrix $\mathbf{\Phi}$ in tensor notation is written as

$$\Phi_{ij} = \sum_{\kappa^H \in K^H} \mu_{\mathrm{P}} \mathbb{P}_{\kappa^H} \delta_{fg} W^{\kappa^H}_{fgij}, \tag{A.3}$$

where $W^{\kappa^H}_{fgij}$ is another bookkeeping tensor that converts the element local indices $f, g = 1 \rightarrow N_s \cdot N_b$ into global numbering indices, i and j. The derivative of Φ_{ij} with respect to the discrete state vector entries is expanded via chain rule using the definitions in Equations A.1 and A.2

$$\frac{\partial \Phi_{ij}}{\partial U_k} = \sum_{\kappa^H \in K^H} \mu_{\mathrm{P}} \delta_{fg} W^{\kappa^H}_{fgij} V^{\kappa^H}_{sbk} \frac{\partial \mathbb{P}_{\kappa^H}}{\partial \mathrm{u}^H_s} \phi^H_b(\mathbf{x}), \tag{A.4}$$

with the elemental penalty expressed as

$$\frac{\partial \mathbb{P}_{\kappa^H}}{\partial \mathrm{u}_s^H} = \sum_l^{N_l} \sum_q^{N_q'} - \frac{w_q'}{c_l(\mathbf{u}^H(\mathbf{x}_q'))^2} \frac{\partial c_l}{\partial \mathrm{u}_s^H}\bigg|_{\mathbf{x}_q'}, \tag{A.5}$$

where l indexes the constraints and q indexes the enhanced quadrature points.

Now, we present the gradient of the constraints for the compressible flow of an ideal gas. For the pressure constraint, the gradient is written as (assuming the conservative state vector presented in Section 2)

$$\frac{\partial c_1}{\partial \mathrm{u}_s^H} = \frac{\gamma - 1}{p_\infty} \cdot \begin{cases} \dfrac{\mathrm{u}_z^H \mathrm{u}_z^H}{2\mathrm{u}_1^{H^2}} & \text{for} \quad s = 1, \;\; z = 2 \rightarrow 1 + \dim \\ \dfrac{\mathrm{u}_s^H}{\mathrm{u}_1^H} & \text{for} \quad s = 2 \rightarrow 1 + \dim \\ 1 & \text{for} \quad s = 2 + \dim \end{cases}, \tag{A.6}$$

where γ is the ratio of specific heats. The gradient of the density constraint is simply

$$\frac{\partial c_2}{\partial \mathrm{u}_s^H} = \frac{\delta_{1s}}{\rho_\infty}. \tag{A.7}$$

The elemental penalties and their gradients are updated at each nonlinear iteration and stored as a vector similar to the discrete state. The bookkeeping tensors are implicitly defined in the routines that add the penalty terms into the coefficient matrix in Equation 21.

ACKNOWLEDGEMENTS

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. Kelley CT, Keyes DE. Convergence analysis of pseudo-transient continuation. *SIAM Journal on Numerical Analysis* 1998; **35**(2):508–523. DOI: 10.1137/S0036142996304796.
2. Coffey TS, Kelley CT, Keyes DE. Pseudo-transient continuation and differential-algebraic equations. *Journal of Scientific Computing* 2003; **25**(2):553–569. DOI: 10.1137/S106482750241044X.
3. Kelley CT, Liao L-Z, Qi L, Chu MT, Reese JP, Winton C. Projected pseudo-transient continuation. *SIAM Journal on Numerical Analysis* 2008; **46**(6):3071–3083.
4. Wang L, Mavriplis D. Adjoint-based $\boldsymbol{h-p}$ adaptive discontinuous Galerkin methods for the 2D compressible Euler equations. *Journal of Computational Physics* 2009; **228**:7643–7661.
5. Yano M, Modisette JM, Darmofal DL. The importance of mesh adaptation for higher-order discretizations of aerodynamics flows. *Technical Report 2011–3852*, 2011.
6. Ceze M, Fidkowski KJ. Anisotropic *hp*-adaptation framework for functional prediction. *AIAA Journal* 2013; **51**(2):492–509.
7. Neumann JV, Richtmyer RD. A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics* 1950; **21**:232–237.
8. Persson P-O, Peraire J. Sub-cell shock capturing for discontinuous Galerkin methods. *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, 2006. DOI: 10.2514/MASM06.
9. Hicken JE, Zingg DW. Globalization strategies for inexact-Newton solvers. *19th AIAA Computational Fluid Dynamics*, San Antonio, Texas, 2009. DOI: 10.2514/CFD09.
10. Hicken JE, Buckley H, Osusky M, Zingg DW. Dissipation-based continuation: a globalization for inexact-Newton solvers. *20th AIAA Computaional Fluid Dynamics Conference*, Honolulu, Hawaii, 2011. DOI: 10.2514/MCFD11.
11. Fidkowski KJ, Oliver TA, Lu J, Darmofal DL. *p*-multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. *Journal of Computational Physics* 2005; **207**:92–113.

12. Fidkowski KJ. A high-order discontinuous Galerkin multigrid solver for aerodynamic applications. *MS Thesis*, Department of Aeronautics and Astronautics, 2004.
13. Ceze M, Fidkowski KJ. A robust adaptive solution strategy for high-order implicit CFD solvers. *20th AIAA Computaional Fluid Dynamics Conference*: AIAA, 2011.
14. Scovazzi G, Christon MA, Hughes TJR, Shadid JN. Stabilized shock hydrodynamics: I. A lagrangian method. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**(4–6):923–966. http://www.sciencedirect.com/\penalty-\@Mscience/article/pii/S0045782506002374.
15. Hughes TJR, Scovazzi G, Tezduyar TE. Stabilized methods for compressible flows. *Journal of Scientific Computing* 2010; **43**(3):343–368.
16. Moro D, Nguyen NC, Peraire J. A hybridized discontinuous Petrov–Galerkin scheme for scalar conservation laws. *International Journal for Numerical Methods in Engineering* 2012; **91**(9):950–970.
17. Keyes DE, Reynolds DR, Woodward CS. Implicit solvers for large-scale nonlinear problems. *Journal of Physics: Conference Series* 2006; **46**:433–442.
18. Roe PL. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics* 1981; **43**:357–372.
19. Bassi F, Rebay S. GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations. In *Discontinuous Galerkin methods: Theory, computation and applications*, Cockburn K, Shu (eds). Springer: Berlin, 2000.
20. Allmaras SR, Johnson FT, Spalart PR. Modifications and clarifications for the implementation of the Spalart-Allmaras turbulence model. *Seventh Intenational Conference on Computational Fluid Dynamics (ICCFD7)*, Big Island, Hawaii, 9-13 July 2012.
21. Spalart PR, Allmaras SR. A one-equation turbulence model for aerodynamic flows. *30th Aerospace Sciences Meeting and Exhibit*: AIAA, Reno, Nevada, 1992. DOI: 10.2514/MASM92.
22. Ceze Marco, Fidkowski Krzysztof J. Pseudo-transient Continuation, Solution Update Methods, and CFL Strategies for DG Discretizations of the RANS-SA Equations. *21th AIAA Computaional Fluid Dynamics Conference*, San Diego, California, 2013. DOI: 10.2514/MCFD13.
23. Saad Y, Schultz MH. Gmres: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific Computing* 1986; **7**(3):856–869.
24. Saad Y. A flexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing* 1993; **14**(2):461–469.
25. Knoll DA, Keyes DE. Jacobian-free Newton-Krylov methods: a survey of approaches and applications. *Journal of Computational Physics* 2004; **193**:357–397.
26. Bucker HM, Pollul B, Rasch A. On CFL evolution strategies for implicit upwind methods in linearized Euler equations. *International Journal for Numerical Methods in Fluids* 2009; **59**:1–18.
27. Mulder WA, van Leer B. Experiments with implicit upwind methods for the Euler equations. *Journal of Computational Physics* 1985; **59**(2):232–246. DOI: 10.1016/0021-9991(85)90144-5.
28. Atkins HL, Pampell A. Robust and accurate shock capturing method for high-order discontinuos Galerkin methods. *20th AIAA Computaional Fluid Dynamics Conference*, Honolulu, Hawaii, 2011. DOI: 10.2514/MCFD11.
29. Hartung J. A stable interior penalty method for convex extremal problems. *Numerische Mathematik* 1978; **29**(2): 149–158.
30. Kelley CT. *Iterative methods for linear and nonlinear equations*, Frontiers in Applied Mathematics 16. Society for Industrial and Applied Mathematics, 1995.
31. Modisette JM. An automated reliable method for two-dimensional Reynolds-averaged Navier-Stokes simulations. *PhD Dissertation*, 2011.
32. Armijo Larry. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics* 1966; **16**(1):1–3.
33. Ceze MA. A robust *hp*-adaptation method for discontinuous Galerkin discretizations applied to aerodynamic flows. *Ph.D. Thesis*, 2013.
34. Chisholm TT, Zingg DW. A Jacobian-free Newton-Krylov algorithm for compressible turbulent fluid flows. *Journal of Computational Physics* 2009; **228**:3490–3507.
35. Ceze M, Fidkowski KJ. Drag prediction using adaptive discontinuous finite elements. *AIAA Journal of Aircraft* August 2014; **51**(4):1284–1294.
36. Wang ZJ, Fidkowski K, Abgrall R, Bassi F, Caraeni D, Cary A, Deconinck H, Hartmann R, Hillewaert K, Huynh HT, Kroll N, May G, Persson P-O, van Leer B, Visbal M. High-order CFD methods: current status and perspective. *International Journal for Numerical Methods in Fluids* 2013; **72**(8):811–845. DOI: 10.1002/fld.3767.