High-Order Resolution of Multidimensional Compressible Reactive Flow Using Implicit Shock Tracking

Matthew J. Zahr* and Joseph M. Powers†
University of Notre Dame, Notre Dame, Indiana 46556

A recently developed high-order implicit shock tracking method is novelly applied to a benchmark problem in two-dimensional compressible reactive flow, and results of remarkably high accuracy are achieved relative to competing shock capturing schemes. High-order implicit shock tracking is a discontinuous Galerkin discretization of conservation laws that simultaneously computes an approximate flow solution and aligns faces of the computational mesh with discontinuities in the flow to provide nonlinear stabilization and an improved approximation to the solution. The model problem is chosen such that its exact solution is available in analytic form to facilitate a detailed study of the truncation error of the tracking method relative to a nominally fifth-order weighted essentially nonoscillatory method. We show the implicit tracking method is able to robustly align the mesh with the shock and, particularly for polynomial bases of degree greater than 1, provide a high-quality approximation to the exact solution on meshes far coarser than required by standard methods. Finally, we demonstrate the tracking method obtains near optimal convergence rates in several error metrics for the problem under consideration.

Nomenclature

A	=	Jacobian matrix of flux function evaluated at Roe state
a_1, a_2, a_3, a_4	=	intermediate constants
B	=	Jacobian matrix of flux function
c	=	frozen sound speed
d	=	number of spatial dimensions
ds	=	differential surface element
dV	=	differential volume element
E	=	total energy
$\mathcal{E}_{h,q}$	=	mesh
e	=	specific internal energy
$e_H \, L^2(\Omega)$	=	error in total enthalpy
$e_\lambda \, L^1(\Omega)$	=	error in reaction progress
$e_\rho \, L^1(\Omega)$	=	error in density
$e^{(1)}$	=	first canonical unit vector in $\mathbb{R}^2$: $e^{(1)} = (1,0)$
F	=	flux function of conservation law
$\mathcal{F}$	=	numerical flux function
f	=	general function to be minimized
H	=	total enthalpy
$\mathcal{H}$	=	Heaviside function
h	=	mesh size parameter
k	=	smoothness parameter for smoothed absolute value function
κ	=	mesh regularization parameter
$\mathcal{L}$	=	Lagrangian of constrained optimization problem
M	=	Mach number
$m(\cdot)$	=	achieved order of convergence of error metric
$\boldsymbol{n}$	=	outward normal to physical domain
P	=	pressure
p	=	polynomial degree for solution approximation
Q	=	rotation matrix
q	=	polynomial degree of geometry approximation
$\hat{q}$	=	heat release per unit mass

R	=	gas constant
$\boldsymbol{R}$	=	discontinuous Galerkin residual, enriched trial space
$\boldsymbol{R}_{\mathrm{msh}}$	=	mesh quality residual
$\boldsymbol{r}$	=	discontinuous Galerkin residual
s	=	entropy
T	=	temperature
T^{ig}	=	ignition temperature
U	=	conservation law solution
$\boldsymbol{u}$	=	discontinuous Galerkin approximation to conservation law
V	=	right eigenvectors of flux Jacobian
$\mathcal{V}_{h,p}$	=	approximation space for solution U
V_1, V_2	=	velocity in rotated coordinate system
v_i	=	velocity in x_i direction
X_1, X_2	=	rotated coordinates
$X_{2,w}$	=	wall streamline in rotated coordinates
$\boldsymbol{x}$	=	nodal coordinates of mesh
$x_1, \ldots, x_d$	=	coordinate directions
α	=	reaction kinetic rate constant
β	=	shock angle
Γ_D, U_D^∂	=	supersonic inflow boundary and state
Γ_N, U_N^∂	=	supersonic outflow boundary and state
Γ_w, U_w^∂	=	wall boundary and state
γ	=	ratio of specific heats
δ_{ij}	=	Kronecker delta
Λ	=	eigenvalue matrix of flux Jacobian
λ	=	reaction progress
$\boldsymbol{\lambda}$	=	Lagrange multipliers
ρ	=	density
ψ	=	test function
$\Omega, \partial\Omega$	=	physical domain and boundary
ω_3	=	vorticity

Subscript

∞	=	freestream quantity

Superscript

$+$	=	interior trace of quantity

*Assistant Professor, Department of Aerospace and Mechanical Engineering, 300B Cushing Hall; mzahr@nd.edu. Member AIAA.

†Professor, Department of Aerospace and Mechanical Engineering, 366 Fitzpatrick Hall; powers@nd.edu. Associate Fellow AIAA.

I. Introduction

SHOCK-INDUCED combustion and detonation is a pervasive feature in high Mach number aeropropulsion, for example,

scramjets, rotating detonation engines, and others. It is necessary to capture the details of the reaction zone to understand ignition, stability, unstart, and many other important physical phenomena. For realistic hydrogen systems, the smallest reaction zone thickness is at the micron scale [1] and must be resolved to capture proper dynamics, especially near stability boundaries [2]. Numerical viscosity introduced by coarse grids, or intentionally introduced for stabilization, can interfere with detonation dynamics, for example, introducing artificial dissipative heating that can falsely trigger reaction [3,4]. For these problems, it is essential to have numerical methods that capture shocks while introducing minimal dissipation; however, many of the most widely used numerical methods to approximate high Mach flows either explicitly introduce viscosity for stabilization or inherently add dissipation at the shock front where the solution is underresolved.

The standard approaches for nonlinear stabilization of numerical methods for flows with discontinuous solutions or sharp gradients are shock capturing methods such as artificial viscosity [5–8], weighted essentially nonoscillatory (WENO) methods [9–13], limiting [14–18], and low-order finite volume methods [19]. These methods use the resolution in the numerical scheme, usually a piecewise polynomial basis, to capture the discontinuity, or a smeared approximation of it. Because polynomials are inherently ill suited to represent discontinuous functions, these techniques must be combined with extreme local mesh refinement [20–22] in the vicinity of the shock. Inviscid shock capturing methods lead to numerical schemes with significant numerical dissipation that can lead to the aforementioned issues, which places stringent requirements on mesh adaptation. This leads to expensive computations that require many degrees of freedom, even for relatively simple problems in two dimensions. There exists a class of numerical methods [4] that leverages the multiscale nature of wavelets as the underlying basis and allow for a highly adapted discretization with few degrees of freedom; however, they require continuity of the solution, necessitating the resolution of thin physically viscous structures. In contrast to the previous methods, shock fitting offers the opportunity to achieve high accuracy and high-order convergence rates [23–29]. However, because of the geometric complexity of generating a mesh of an unknown discontinuity surface, fitting methods have proven difficult to use as the foundation of a general-purpose method and tend to find the most success for specific applications in one and two dimensions [27,30]. While both low-order [24,28,29] and high-order/spectral [25,27] methods have been used in the context of shock fitting, Zahr and Persson [31] showed the benefits of shock fitting relative to a shock capturing method with adaptive mesh refinement are modest when used with low-order schemes and increasingly exaggerated as the order of the underlying discretization increases. For more information, Ref. [32] provides a comprehensive review of numerical methods for detonation problems.

We consider a new class of methods, high-order implicit shock tracking, for inviscid detonation with finite-rate kinetics that overcomes the limitations of shock capturing and traditional shock fitting methods. Implicit shock tracking [31,33–36] is a new approach to shock fitting that, instead of attempting to explicitly generate a mesh that conforms to the (unknown) discontinuities in the flow, poses and solves an optimization problem whose solution is a discontinuity-aligned mesh and the corresponding flow solution. The implicit tracking solver simultaneously updates the mesh and flow solution to iteratively converge to the solution of the optimization problem; this ensures the converged flow solution is never required on a nonaligned mesh, thereby avoiding the nonlinear stability issues that would otherwise arise. The advantage of the implicit tracking approach over traditional shock fitting methods is that the geometric complexity of generating a mesh that conforms to an unknown surface has been replaced with a nonlinear optimization problem, which can be solved using techniques from numerical optimization [37]. This enables high-order accurate solutions with more complicated geometrical features including multiple embedded discontinuity surfaces [33,34].

Combining this implicit tracking approach with high-order methods, in particular, a discontinuous Galerkin (DG) discretization, leads to a scheme that is nonlinearly stable and extremely accurate per degree of freedom and introduces minimal numerical dissipation. It is inherently an r-adaptive method that achieves high-order convergence to

the exact solution for inert flows [31,35] and reactive flows as we demonstrate in this paper. Furthermore, because we used a high-order finite element method as the underlying discretization, it naturally works with high-order unstructured meshes and is capable of handling complex geometries.

This paper is the first to apply high-order implicit shock tracking to reacting flow. The method is verified against an exact solution for planar two-dimensional inviscid irrotational detonation with finite-rate, one-step kinetics with a simple reactant depletion model [38]. We use this well-understood benchmark to perform a detailed numerical investigation into the tracking method and demonstrate the method is remarkably accurate compared to a fifth-order WENO method [38], a competing shock capturing scheme. In particular, we show that the error associated with the implicit tracking method is two orders of magnitude smaller and uses nearly three orders of magnitude fewer degrees of freedom than the WENO method. We also show the implicit tracking method converges at near-optimal rates under h refinement, in other words, $\mathcal{O}(h^{p+1})$, where h is the mesh size parameter and p is the polynomial degree of the basis, while the WENO method is limited to sub-first-order convergence as reported in Ref. [38]. Even on an extremely coarse mesh with 100 elements, the implicit tracking method produces a solution that, to high accuracy, preserves structure of inviscid flow, for example, close agreement with the exact solution field, near constant total enthalpy, and near zero vorticity.

The remainder of the paper is organized as follows. Section II introduces the two-dimensional benchmark problem, the governing equations (reactive Euler equations), and a brief review of the exact solution. Section III reviews the discontinuous Galerkin discretization and implicit shock tracking formulation; a complete description of the tracking method and the corresponding solvers can be found in Ref. [34]. Section IV presents an extensive numerical investigation into the robustness, accuracy, and convergence of the high-order implicit shock tracking method applied to the model problem and compares its performance to the fifth-order WENO scheme in Ref. [38]. Finally, Sec. V offers conclusions and an outlook for future work.

II.   Analytical Modeling

A.   Geometry and Flow Configuration

We model the flow configuration shown in Fig. 1. Here, a curved two-dimensional wedge has its apex at the origin $(x_1, x_2) = (0, 0)$. The wedge is immersed in a supersonic flowfield in which an inviscid unreacted fluid flows from the left boundary. When the fluid encounters
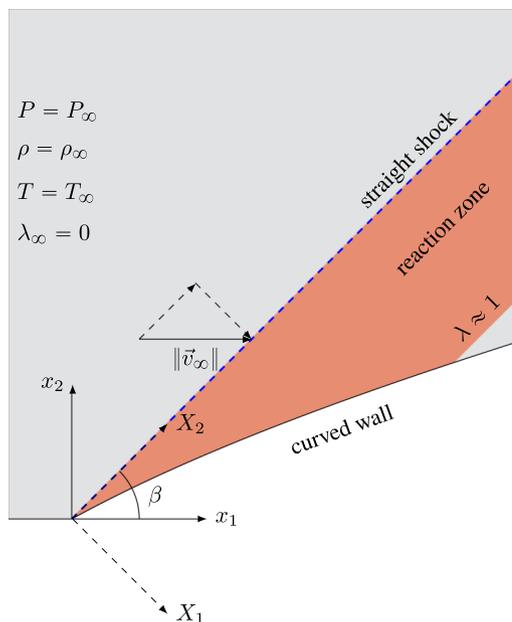

Fig. 1   Schematic of straight wall, curved shock flow setup.

the wedge, a straight oblique shock, inclined at angle β to the horizontal, is induced. The shock triggers one-step irreversible exothermic chemical reaction through a finite length reaction zone. The reaction induces streamline curvature, and the wedge curvature is selected so that it is parallel to flow streamlines. The freestream conditions prescribe the density ρ_∞, temperature T_∞, and pressure P_∞; the freestream velocity is aligned with the x_1 axis

$$\boldsymbol{v}_\infty = \begin{bmatrix} v_{1\infty} \\ 0 \end{bmatrix} \tag{1}$$

and no reaction has occurred at the inlet, $\lambda_\infty = 0$.

B. Reactive Euler Equations

Following Powers and Aslam [38], the reactive Euler equations are chosen to model the steady flow of an inviscid, compressible fluid through a domain $\Omega \subset \mathbb{R}^d$ and the evolution of chemical kinetics:

$$\begin{aligned} (\rho v_j)_{,j} &= 0, \\ (\rho v_i v_j + P\delta_{ij})_{,j} &= 0, \\ \left(\rho v_j \left(e + \frac{v_i v_i}{2} + \frac{P}{\rho}\right)\right)_{,j} &= 0, \qquad \text{in } \Omega \\ (\rho v_j \lambda)_{,j} &= \alpha\rho(1-\lambda)\mathcal{H}(T - T^{\text{ig}}) \end{aligned} \tag{2}$$

where $\rho: \Omega \to \mathbb{R}_+$ is the density of the fluid, $v_i: \Omega \to \mathbb{R}$ for $i = 1, \ldots, d$ is the velocity of the fluid in the ith coordinate direction and $\boldsymbol{v} = (v_1, \ldots, v_d)$, $e: \Omega \to \mathbb{R}_+$ is the specific internal energy of the fluid, $P: \Omega \to \mathbb{R}_+$ is the pressure, $T: \Omega \to \mathbb{R}$ is the temperature, $\lambda: \Omega \to (0, 1)$ is the reaction progress, and $\mathcal{H}: \mathbb{R} \to \{0, 1\}$ is the Heaviside function. The term d is the number of spatial dimensions, which can be 1, 2, or 3. Here, we will be restricted to $d = 2$. The source term in the reaction equation models a single irreversible reaction with simple reactant depletion kinetics and no Arrhenius term. The reactant and product are both taken to be calorically perfect ideal gases with identical gas constants R and ratio of specific heats γ. For the mixture, the thermal and caloric state equations are

$$P = \rho R T, \quad e = \frac{1}{\gamma - 1}\frac{P}{\rho} - \lambda\hat{q} \tag{3}$$

with the frozen speed of sound $c: \Omega \to \mathbb{R}_+$ and Mach number $M: \Omega \to \mathbb{R}_+$,

$$c^2 = \left.\frac{\partial P}{\partial \rho}\right|_{s,\lambda} = \gamma\frac{P}{\rho}, \quad M = \frac{\sqrt{v_i v_i}}{c} \tag{4}$$

where $s: \Omega \to \mathbb{R}$ is the entropy. The relevant constants are the reaction kinetic rate constant α, ratio of specific heats γ, heat release per unit mass $\hat{q}$, gas constant R, and ignition temperature T^{ig}.

For convenience, we introduce the total energy $E: \Omega \to \mathbb{R}_+$ and total enthalpy $H: \Omega \to \mathbb{R}_+$ of the flow

$$E = e + \frac{v_i v_i}{2}, \quad H = E + \frac{P}{\rho} \tag{5}$$

that allows us to succinctly rewrite the energy equation as

$$(\rho v_j H)_{,j} = 0, \quad \text{in } \Omega \tag{6}$$

Expanding the energy equation and using mass conservation allows us to deduce that the total enthalpy of a fluid particle remains constant on a streamline,

$$\frac{dH}{dt} = 0 \tag{7}$$

where d/dt is the material derivative, where the material derivative of H for the steady flow has the standard definition $dH/dt = v_j H_{,j}$. Moreover, standard Rankine–Hugoniot jump analysis shows that H remains constant across a shock. Because the freestream has a spatially homogeneous value of $H = H_\infty$, and H must remain constant even during chemical reaction, it can be concluded that throughout the entire flowfield

$$H = \frac{\gamma}{\gamma - 1}\frac{P_\infty}{\rho_\infty} + \frac{1}{2}\boldsymbol{v}_\infty \cdot \boldsymbol{v}_\infty \tag{8}$$

The vorticity has its standard definition ($d = 2$):

$$\omega_3 = \frac{\partial v_2}{\partial x_1} - \frac{\partial v_1}{\partial x_2} \tag{9}$$

As discussed by Powers and Stewart [39], the relevant Helmholtz vorticity transport equation for continuous regions of the flow is

$$\frac{d\omega_3}{dt} = 0 \tag{10}$$

showing that vorticity does not change along streamlines. Also discussed in Ref. [39], shock curvature can generate vorticity at a discontinuity. Because we consider straight shocks only, there are no vorticity-generating mechanisms. Thus, a fluid particle that is initially irrotational will always be irrotational, and we have

$$\omega_3 = 0 \tag{11}$$

throughout the domain Ω.

C. Exact Solution

The exact solution of the problem in Sec. II.A was derived analytically in Ref. [38]; in this section, we summarize the solution with full details found in Ref. [38]. One first considers the rotation sketched in Fig. 1. The rotation transformation is

$$X_1 = x_1 \sin\beta - x_2 \cos\beta \tag{12}$$

$$X_2 = x_1 \cos\beta + x_2 \sin\beta \tag{13}$$

$$V_1 = v_1 \sin\beta - v_2 \cos\beta \tag{14}$$

$$V_2 = v_1 \cos\beta + v_2 \sin\beta \tag{15}$$

Here, X_1 and X_2 are the distance components in the rotated coordinate system, and V_1 and V_2 are the corresponding velocity components.

One can do a detailed analysis examining solutions that vary in X_1 only. A set of Rankine–Hugoniot jump equations can be enforced to give postshock conditions. Consistent with classical inert oblique shock theory, attention is restricted to solutions where the velocity tangent to the shock V_2 is the same constant both before and after the shock. Additionally, it is required to be the same constant throughout the reaction zone. Detailed algebraic analysis then allows one to determine explicit algebraic expressions for $\rho(\lambda)$, $V_1(\lambda)$, and $P(\lambda)$. For example, one gets for the shocked branch

$$\rho(\lambda) = \frac{\rho_\infty(\gamma + 1)M_\infty^2\sin^2\beta}{1 + \gamma M_\infty^2\sin^2\beta - \sqrt{(1 + \gamma M_\infty^2\sin^2\beta)^2 - (\gamma + 1)M_\infty^2\sin^2\beta[2 + ((\gamma - 1)/\gamma)(2\lambda\hat{q}/RT_\infty) + (\gamma - 1)M_\infty^2\sin^2\beta]}} \tag{16}$$

The remaining ordinary differential equation for the reaction kinetics can be rewritten as

$$\frac{d\lambda}{dX_1} = \frac{\alpha}{\rho_\infty v_{1\infty} \sin\beta} \rho(\lambda)(1-\lambda), \quad \lambda(0) = 0 \tag{17}$$

Equation (17) can be integrated exactly via separation of variables to yield $X_1(\lambda)$:

$$X_1(\lambda) = a_1\left(2a_3(\sqrt{1-a_4\lambda}-1) + \ell_n\left(\left(\frac{1}{1-\lambda}\right)^{a_2}\left(\frac{[1-\sqrt{(1-a_4\lambda)/(1-a_4)}][1+\sqrt{1/(1-a_4)}]}{[1+\sqrt{(1-a_4\lambda)/(1-a_4)}][1-\sqrt{1/(1-a_4)}]}\right)^{a_3\sqrt{1-a_4}}\right)\right) \tag{18}$$

Here, parameters $a_1, \ldots, a_4$ are defined as

$$a_1 = \frac{1}{(\gamma+1)M_\infty \sin\beta} \frac{\sqrt{\gamma R T_\infty}}{\alpha} \tag{19}$$

$$a_2 = 1 + \gamma M_\infty^2 \sin^2\beta \tag{20}$$

$$a_3 = M_\infty^2 \sin^2\beta - 1 \tag{21}$$

$$a_4 = 2\frac{M_\infty^2 \sin^2\beta}{(M_\infty^2 \sin^2\beta - 1)^2}\frac{\gamma^2-1}{\gamma}\frac{\hat{q}}{RT_\infty} \tag{22}$$

We see that a_1 has units of length and that a_2, a_3, and a_4 are dimensionless. Thus, one has X_1 as an explicit function of λ. The inverse $\lambda(X_1)$ may be obtained via numerical iteration. It is easily shown that the wall shape is given by

$$X_{2,w}(X_1) = \frac{v_{1\infty}\cos\beta}{\alpha}\ell_n\left(\frac{1}{1-\lambda(X_1)}\right) \tag{23}$$

III.   High-Order Implicit Shock Tracking for Inviscid Conservation Laws

In this section, we write the reactive Euler equations (2) as a general system of nonlinear conservation laws and introduce the high-order implicit shock tracking method based on a discontinuous Galerkin discretization [31,34].

A.   Conservation Law

The density, velocity, total energy, and reaction are combined into a vector of conservative variables $U: \Omega \to \mathbb{R}^{d+3}$, and the reactive Euler equations take the form of an inviscid conservation law,

$$\nabla \cdot F(U) = S(U) \text{ in } \Omega \Leftrightarrow F_{ij,j} = S_i \text{ in } \Omega \tag{24}$$

where the state vector $U: \Omega \to \mathbb{R}^{d+3}$, flux function $F: \mathbb{R}^{d+3} \to \mathbb{R}^{(d+3)\times d}$, and source term $S: \mathbb{R}^{d+3} \to \mathbb{R}^{d+3}$ are

$$U = \begin{bmatrix} \rho \\ \rho\boldsymbol{v} \\ \rho E \\ \rho\lambda \end{bmatrix}, \quad F(U) = \begin{bmatrix} \rho\boldsymbol{v}^T \\ \rho\boldsymbol{v}\boldsymbol{v}^T + PI_{d\times d} \\ \rho H\boldsymbol{v}^T \\ \rho\lambda\boldsymbol{v}^T \end{bmatrix},$$

$$S(U) = \begin{bmatrix} 0 \\ 0_d \\ 0 \\ \alpha\rho(1-\lambda)\mathcal{H}(T - T^{\text{ig}}) \end{bmatrix} \tag{25}$$

where $I_{d\times d} \in \mathbb{R}^{d\times d}$ is the identity matrix and $0_d \in \mathbb{R}^d$ is the zero vector. Equations (24) and (25) use a form commonly found in the computational literature, and its meaning is generally well understood to be consistent with Eqs. (2) and (6). However, in that the operator ∇ here is the spatial divergence operator, $\nabla = (\partial/\partial x_1, \partial/\partial x_2)^T$, it does not formally operate on the $d+3$ rows of F. Instead, it operates on each of the column elements of F. Consequently, care must be used in extending notions from vector calculus and differential geometry to Eq. (24).

The exact solution for the problem introduced in Sec. II.C was derived for flow in a semi-infinite domain. To reproduce this problem in a computational setting, we truncate the domain a finite distance from the wall and introduce suitable boundary conditions. We consider three types of boundary conditions: slip wall Γ_w, supersonic inflow Γ_D, and supersonic outflow Γ_N. The supersonic inflow is also known as a far-field or Dirichlet condition, and the supersonic outflow is a Neumann condition. For a slip wall $(\boldsymbol{v} \cdot \boldsymbol{n} = 0)$, the boundary state is defined as

$$U_w^\partial(U, \boldsymbol{n}) := \begin{bmatrix} \rho \\ \rho\boldsymbol{v}_-(\boldsymbol{v}, \boldsymbol{n}) \\ \rho E \\ \rho\lambda, \end{bmatrix} \tag{26}$$

where $\boldsymbol{v}_-(\boldsymbol{v}, \boldsymbol{n}) = (\boldsymbol{v} - 2\boldsymbol{v} \cdot \boldsymbol{n})\boldsymbol{n}$ is the velocity reflected about the normal. For a supersonic inflow, all characteristics are coming into the domain, and the boundary state depends solely on the prescribed density ρ_∞, velocity $\boldsymbol{v}_\infty$, pressure P_∞, and reaction λ_∞,

$$U_D^\partial(U) := \begin{bmatrix} \rho_\infty \\ \rho_\infty\boldsymbol{v}_\infty \\ \frac{P_\infty}{\gamma-1} + \frac{\rho_\infty}{2}\boldsymbol{v}_\infty \cdot \boldsymbol{v}_\infty \\ \rho_\infty\lambda_\infty \end{bmatrix} \tag{27}$$

Finally, at a supersonic outflow, all characteristics are leaving the domain, and the boundary state is taken from the interior,

$$U_N^\partial(U) := U \tag{28}$$

Then, the boundary condition on $\partial\Omega := \bar{\Gamma}_w \cup \bar{\Gamma}_D \cup \bar{\Gamma}_N$ is

$$U^\partial(U, \boldsymbol{n}) = \begin{cases} U_w^\partial(U, \boldsymbol{n}) & \text{on } \Gamma_w \\ U_D^\partial(U) & \text{on } \Gamma_D \\ U_N^\partial(U) & \text{on } \Gamma_N \end{cases} \tag{29}$$

where $\Gamma_w, \Gamma_D, \Gamma_N \subset \partial\Omega$ are the surfaces on the domain boundary where the wall, supersonic inflow, and supersonic outflow conditions, respectively, are applied.

B.   Discontinuous Galerkin Discretization

We use a standard nodal DG method [40,41] to discretize the conservation law (24). Let $\mathcal{E}_{h,q}$ be a collection of nonoverlapping, potentially curved, computational elements that discretize the domain Ω; h is a mesh element size parameter, q is the polynomial order associated with the curved elements, and we use $|\mathcal{E}_{h,q}|$ to denote the number of elements in the mesh. The DG construction begins with the elementwise weak form of the conservation law (24) that results from multiplying each equation by a test function ψ, integrating over a single element $K \in \mathcal{E}_{h,q}$, and applying the divergence theorem

$$\int_{\partial K} \psi^+ \cdot F(U)\boldsymbol{n}\,\mathrm{d}s - \int_K F(U) : \nabla\psi\,\mathrm{d}V = \int_K \psi \cdot S(U)\,\mathrm{d}V \quad (30)$$

where $\boldsymbol{n}: \Omega \to \mathbb{R}^d$ is the outward normal to the surface ∂K and ψ^+ denotes the trace of ψ interior to element K. To ensure the face integrals are single valued, we replace $F(U)\boldsymbol{n}$ in the first term with a numerical flux function $\mathcal{F}(U^+, U^-, \boldsymbol{n})$,

$$\int_{\partial K} \psi^+ \cdot \mathcal{F}(U^+, U^-, \boldsymbol{n})\,\mathrm{d}s - \int_K F(U) : \nabla\psi\,\mathrm{d}V = \int_K \psi \cdot S(U)\,\mathrm{d}V \quad (31)$$

where U^+ denotes the interior trace of U and U^- denotes the exterior trace of U if ∂K is an interior face, in other words, $\partial K \cap \partial\Omega = \varnothing$. Otherwise, U^- is a boundary state $U^\partial(U, \boldsymbol{n})$ constructed to enforce the appropriate boundary condition defined in the previous section. In this paper, we use a smoothed version of the Roe flux [42] (Appendix A).

Define the finite element space of piecewise polynomial functions associated with the mesh $\mathcal{E}_{h,q}$,

$$\mathcal{V}_{h,p} = \left\{ v \in [L^2(\Omega_0)]^{d+3} \,|\, v|_K = w_K \circ \mathcal{T}_K^{-1}, w_K \in [\mathcal{P}_p(K_0)]^{d+3}, \right.$$
$$\left. \forall K \in \mathcal{E}_{h,q} \right\}$$

where $\mathcal{P}_p(K_0)$ is the space of polynomial functions of degree at most $p \geq 1$ on the master element K_0 and $K = \mathcal{T}_K(K_0)$ defines a mapping from the master element to element $K \in \mathcal{E}_{h,q}$. For simplicity, we assume all elements map from a single master element. The finite-dimensional residual of the weak form in (31) corresponding to the trial space $\mathcal{V}_{h,p}$ and test space $\mathcal{V}_{h',p'}$ is

$$r_{h',p'}^K(U_{h,p}) := \int_{\partial K} \psi_{h',p'}^+ \cdot \mathcal{F}(U_{h,p}^+, U_{h,p}^-, \boldsymbol{n})\,\mathrm{d}s$$
$$- \int_K F(U_{h,p}) : \nabla\psi_{h',p'}\,\mathrm{d}V - \int_K \psi_{h',p'} \cdot S(U_{h,p})\,\mathrm{d}V \quad (32)$$

After summing over all elements $K \in \mathcal{E}_{h,q}$, the finite-dimensional Galerkin weak form is as follows: find $U_{h,p} \in \mathcal{V}_{h,p}$ such that

$$\sum_{K \in \mathcal{E}_{h,q}} r_{h,p}^K(U_{h,p}) = 0 \quad (33)$$

for all $\psi_{h,p} \in \mathcal{V}_{h,p}$.

To establish the discrete (algebraic) form of Eq. (33), we introduce a (nodal) basis over each element and expand the finite-dimensional test functions $\psi_{h,p}$ and solution $U_{h,p}$ in terms of these basis functions and coefficients. Assembling the algebraic system corresponding to Eq. (33), we have

$$\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{x}) = \boldsymbol{0} \quad (34)$$

where $\boldsymbol{u} \in \mathbb{R}^{N_u}$ are the (assembled) coefficients of the solution $U_{h,p} \in \mathcal{V}_{h,p}$ and $\boldsymbol{x} \in \mathbb{R}^{N_x}$ are the nodal coordinates of the mesh $\mathcal{E}_{h,q}$, including high-order nodes for $q > 1$. It is convenient to transform the conservation law to a fixed reference domain as it leads to transformed fluxes that depend on the mapping from the reference to physical domain. As a result, the dependence of the residual on the state $\boldsymbol{u}$ and mesh coordinates $\boldsymbol{x}$ is similar, which facilitates implementation of required partial derivatives. We omit the details for brevity; see Refs. [31,34] for details.

Finally, define the enriched discrete residual $\boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x})$ as the algebraic version of

$$\sum_{K \in \mathcal{E}_{h,q}} r_{h,p+1}^K(U_{h,p}) \quad (35)$$

which will be used to define the implicit tracking objective function. The enriched residual is defined by the same trial space $\mathcal{V}_{h,p}$ as the residual in Eq. (34) but uses an enriched test space $\mathcal{V}_{h,p+1}$.

C. Implicit Shock Tracking

The main idea behind implicit shock tracking is that we do not explicitly generate a mesh of the discontinuity surface; rather, we pose an optimization problem over the discretized solution vector $\boldsymbol{u}$ and nodal mesh coordinates $\boldsymbol{x}$ such that its solution is a mesh aligned with the discontinuity surface $\boldsymbol{x}^\star$ and the corresponding discretized solution $\boldsymbol{u}^\star$, in other words, $\boldsymbol{r}(\boldsymbol{u}^\star, \boldsymbol{x}^\star) = \boldsymbol{0}$,

$$\underset{\boldsymbol{u} \in \mathbb{R}^{N_u}, \boldsymbol{x} \in \mathbb{R}^{N_x}}{\text{minimize}} \quad f(\boldsymbol{u}, \boldsymbol{x})$$
$$\text{subject to} \quad \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{x}) = \boldsymbol{0} \quad (36)$$

The objective function is constructed such that the solution of the optimization problem is a high-quality mesh that aligns with discontinuities in the solution

$$f(\boldsymbol{u}, \boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{R}(\boldsymbol{u}, \boldsymbol{x})\|_2^2 + \frac{\kappa^2}{2}\|\boldsymbol{R}_{\text{msh}}(\boldsymbol{x})\|_2^2 \quad (37)$$

where $\kappa \in \mathbb{R}_+$ is a parameter that weights the contribution of the two terms and $\boldsymbol{R}_{\text{msh}}: \mathbb{R}^{N_x} \to \mathbb{R}^{|\mathcal{E}_{h,q}|}$ is an elementwise mesh distortion metric; see Ref. [34] for details. The first term penalizes a measure of the DG solution error and promotes alignment of the mesh with discontinuities because a piecewise polynomial solution on an aligned mesh will have much lower error than on a nonaligned one. The second term actively promotes mesh smoothing and is a safeguard that prevents the mesh from entangling or becoming unacceptably skewed. Furthermore, the implicit tracking method directly inherits the benefits of standard DG methods, in other words, high-order accuracy and conservation, due to the constraint that exactly enforces the DG discretization.

The Lagrangian of the optimization problem in Eq. (36), $\mathcal{L}: \mathbb{R}^{N_u} \times \mathbb{R}^{N_x} \times \mathbb{R}^{N_u} \to \mathbb{R}$, takes the form

$$\mathcal{L}(\boldsymbol{u}, \boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{u}, \boldsymbol{x}) - \boldsymbol{\lambda}^T \boldsymbol{r}(\boldsymbol{u}, \boldsymbol{x}) \quad (38)$$

where $\boldsymbol{\lambda} \in \mathbb{R}^{N_u}$ is a vector of Lagrange multipliers associated with the DG constraint in Eq. (36). The first-order optimality, or Karush–Kuhn–Tucker, conditions state that $(\boldsymbol{u}^\star, \boldsymbol{x}^\star)$ is a first-order solution of the optimization problem if there exists $\boldsymbol{\lambda}^\star$ such that

$$\frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}^\star, \boldsymbol{x}^\star)^T - \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}^\star, \boldsymbol{x}^\star)^T \boldsymbol{\lambda}^\star = \boldsymbol{0},$$
$$\frac{\partial f}{\partial \boldsymbol{x}}(\boldsymbol{u}^\star, \boldsymbol{x}^\star)^T - \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}(\boldsymbol{u}^\star, \boldsymbol{x}^\star)^T \boldsymbol{\lambda}^\star = \boldsymbol{0},$$
$$\boldsymbol{r}(\boldsymbol{u}^\star, \boldsymbol{x}^\star) = \boldsymbol{0} \quad (39)$$

Following Ref. [34], we define the estimate of the optimal Lagrange multiplier $\hat{\boldsymbol{\lambda}}: \mathbb{R}^{N_u} \times \mathbb{R}^{N_x} \to \mathbb{R}^{N_u}$ that guarantees the first condition in Eq. (39) is always satisfied,

$$\hat{\boldsymbol{\lambda}}(\boldsymbol{u}, \boldsymbol{x}) = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{x})^{-T} \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{x})^T \quad (40)$$

which reduces the optimality criteria to

$$\boldsymbol{c}(\boldsymbol{u}^\star, \boldsymbol{x}^\star) := \frac{\partial f}{\partial \boldsymbol{x}}(\boldsymbol{u}^\star, \boldsymbol{x}^\star)^T - \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{x}}(\boldsymbol{u}^\star, \boldsymbol{x}^\star)^T \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{u}}(\boldsymbol{u}^\star, \boldsymbol{x}^\star)^{-T} \frac{\partial f}{\partial \boldsymbol{u}}(\boldsymbol{u}^\star, \boldsymbol{x}^\star)^T$$
$$= \boldsymbol{0},$$
$$\boldsymbol{r}(\boldsymbol{u}^\star, \boldsymbol{x}^\star) = \boldsymbol{0} \quad (41)$$

In this paper, we use the sequential quadratic programming (SQP) method proposed in Ref. [34]. A pair $(\boldsymbol{u}, \boldsymbol{x})$ is considered a solution of Eq. (36) if $\|\boldsymbol{c}(\boldsymbol{u}, \boldsymbol{x})\| < \epsilon_1$ and $\|\boldsymbol{r}(\boldsymbol{u}, \boldsymbol{x})\| < \epsilon_2$ for $\epsilon_1, \epsilon_2 > 0$.

IV. Numerical Experiment

For a one-to-one comparison, we consider the exact problem from Ref. [38], albeit nondimensionalized (Table 1). We use $(\tilde{\cdot})$ to denote the dimensional version of the nondimensional quantity $(\cdot)$ and nondimensionalize following the procedure in Ref. [43]:

$$t = \tilde{\alpha}\tilde{t}, \quad x_i = \frac{\tilde{\alpha}\tilde{x}_i}{M_\infty \sqrt{\tilde{P}_\infty/\tilde{\rho}_\infty}}, \quad v_i = \frac{\tilde{v}_i}{M_\infty \sqrt{\tilde{P}_\infty/\tilde{\rho}_\infty}},$$

$$\rho = \frac{\tilde{\rho}}{\tilde{\rho}_\infty}, \quad P = \frac{\tilde{P}}{M_\infty^2 \tilde{P}_\infty}, \quad T = \frac{\tilde{T}}{M_\infty^2 \tilde{T}_\infty} \tag{42}$$

This leads to the nondimensional constants

$$\alpha = 1, \quad R = 1, \quad \hat{q} = \frac{\tilde{q}\rho_\infty}{M_\infty^2 \tilde{P}_\infty} \tag{43}$$

and far-field conditions

$$\rho_\infty = 1, \quad \boldsymbol{v}_\infty = (\sqrt{\gamma}, 0), \quad P_\infty = \frac{1}{M_\infty^2}, \quad T_\infty = \frac{1}{M_\infty^2}, \quad \lambda_\infty = 0 \tag{44}$$

where we assume $\tilde{v}$ aligned with the x_1 direction and used $c = \sqrt{\gamma P/\rho}$. We adopt the nonstandard approach of nondimensionalizing constants as well as flow quantities to ensure the equations introduced in Sec. II maintain their current form in the nondimensional setting, for example, without requiring each equation to be reformulated once the constants cancel due to nondimensionalization. This also provides a direct connection to our implementation.

We use three error metrics to quantify the approximation error associated with the implicit tracking method applied to the reacting Euler equations. The first error metric we consider is the L^2 norm of the total enthalpy (8):

$$e_H(U_{h,p}) \sqrt{\int_\Omega |H(U_{h,p}) - H|^2} \tag{45}$$

Because this problem has an exact solution [38], we will also use the L^1 error of the numerical flow density $\rho(U_{h,p})$ and reaction progress $\lambda(U_{h,p})$,

Table 1   Dimensional and corresponding nondimensional quantities used for the oblique detonation simulation, consistent with Ref. [38]

Quantity	Dimensional	Nondimensional
Freestream Mach	— —	$M_\infty = 3$
Shock angle	— —	$\beta = 45$ deg
Ratio of specific heats	— —	$\gamma = 6/5$
Gas constant	$\tilde{R} = 287$ J/(kg·K^{-1})	$R = 1$
Heat release	$\tilde{q} = 3.0 \times 10^5$ J/kg	$\hat{q} = 0.387$
Reaction kinetic rate	$\tilde{\alpha} = 1000$ Hz	$\alpha = 1$
Ignition temperature	$\tilde{T}^{\text{ig}} = 361.58$ K	$T^{\text{ig}} = 0.134$
Freestream density	$\tilde{\rho}_\infty = 1$ kg/m^3	$\rho_\infty = 1$
Freestream temperature	$\tilde{T}_\infty = 300$ K	$T_\infty = 0.111$
Freestream velocity	$\tilde{\boldsymbol{v}}_\infty = (18\sqrt{2870}, 0)$ m/s	$v_\infty = (1.095, 0)$
Freestream pressure	$\tilde{P}_\infty = 8.61 \times 10^4$ Pa	$P_\infty = 0.111$
Freestream reaction	— —	$\lambda_\infty = 0$
Domain x_1 limits	$(-0.25, 0.75)$ m	$(-0.284, 1.99)$
Domain x_2 limits	$(0, 2)$ m	$(0, 2.27)$

$$e_\rho(U_{h,p}) \coloneqq \int_\Omega |\rho(U_{h,p}) - \rho^\star| \, dV,$$

$$e_\lambda(U_{h,p}) \coloneqq \int_\Omega |\lambda(U_{h,p}) - \lambda^\star| \, dV \tag{46}$$

where $\rho^\star$ and $\lambda^\star$ are the exact solutions for the density and reaction progress, respectively. Because of the nondimensionalization here relative to Ref. [38], we must scale the L^1 error in Ref. [38] by the square of the domain scaling (because we are in $d = 2$ dimensions), in other words,

$$\left(\frac{\tilde{\alpha}}{M_\infty \sqrt{\tilde{P}_\infty/\tilde{\rho}_\infty}}\right)^2 \approx 1.29 \tag{47}$$

for a one-to-one quantitative comparison between the errors. Because this scaling is $\mathcal{O}(1)$, it has little impact.

To begin, we use DistMesh [44] to create a mesh of the rectangular domain with a curved wall (Fig. 1, Sec. II.C) consisting of 109 straight-sided ($q = 1$) triangular elements that do not conform to the discontinuity surface. Then, we include high-order nodes to create $q = 2, 3$ meshes; all nodes on the (faceted) wall boundary were projected onto the true (curved) wall to provide high-order approximations to the wall boundary.

A. Initialization with Low-Order Implicit Shock Tracking

To initialize the high-order tracking method, we apply the implicit tracking method to the $q = 1$ mesh with a $p = 0$ (piecewise constant) solution space due to the inherent robustness of low-order discretizations. The nodal coordinates of the unstructured mesh (no knowledge of the shock location) are used to initialize $\boldsymbol{x}$ in the implicit tracking problem (36). The DG solution $\boldsymbol{u}$ is initialized from the $p = 0$ DG solution on the nonaligned mesh. We allow the tracking algorithm to take 100 SQP iterations (Fig. 2), which results in nine element collapses and ultimately converges to a mesh that aligns closely with the true shock position (straight shock of angle $\beta = 45$ deg that intersects the origin); however, the solution is underresolved due to the coarse, first-order approximation.

B. High-Order Implicit Shock Tracking

To improve the approximation, use the first-order tracking solution to initialize high-order ($p = q = 1, 2, 3$) implicit tracking simulations (keeping the number of elements fixed) (Fig. 3 for ρ and Fig. 4 for λ). In all cases, the tracking procedure successfully tracks the shock; the high-order approximations ($p = q > 1$) yield accurate approximations to the flow on the coarse mesh with $\mathcal{O}(100)$ elements, while the low-order ($p = q = 1$) solution is underresolved. Table 2 shows the three error metrics for the $p = q = 1, 2, 3$ implicit tracking simulations along with the number of DG and mesh degrees of freedom. Even on this coarse mesh with 109 triangular elements, the implicit tracking solutions have small errors.

The nominally fifth-order finite difference WENO scheme with Lax–Friedrichs scheme used in Ref. [38] was unable to drive the L^1 error of the density (after scaling) below 10^{-2} even on their finest grid of 1024×1024, which corresponds to 5.24×10^6 degrees of freedom (DOF). Implicit tracking offers a substantial improvement because errors below 10^{-2} are achieved with less than 4000 total DOF (including mesh and DG DOF), a reduction in three orders of magnitude.

The vorticity and enthalpy error $|H(U_{h,p}) - H_\infty|$ of the flow for the $p = q = 1, 2, 3$ implicit tracking simulations are provided in Figs. 5 and 6, respectively. These fields provide insight into the spatial distribution of the error because the true flow is irrotational and constant total enthalpy. As expected, the largest error occurs in the vicinity of the discontinuity and domain boundaries (the wall and outflow in particular). Even though the implicit tracking algorithm is intended to align with the discontinuity, in practice, it will always be slightly misaligned due to the truncation error in the solution, which prevents the flow from being perfectly irrotational and constant
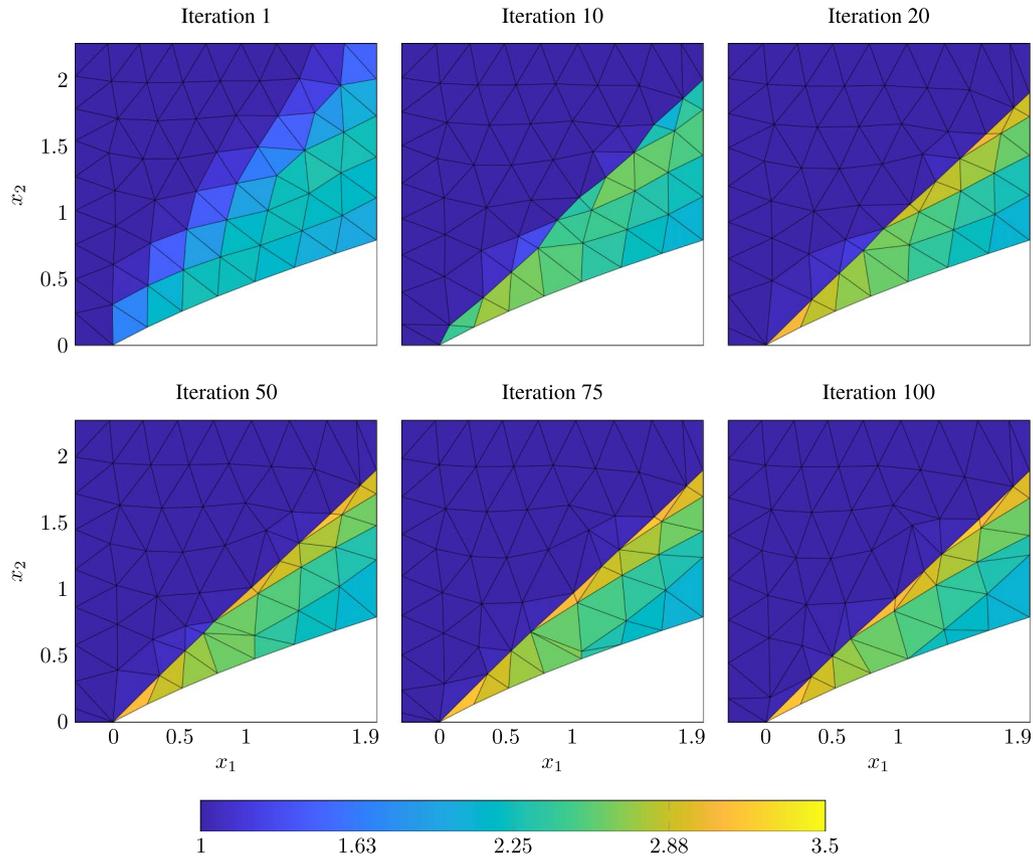
Fig. 2   Iterations of implicit tracking algorithm (colored by density) using a $p = 0$ basis for the solution and $q = 1$ basis for the mesh.

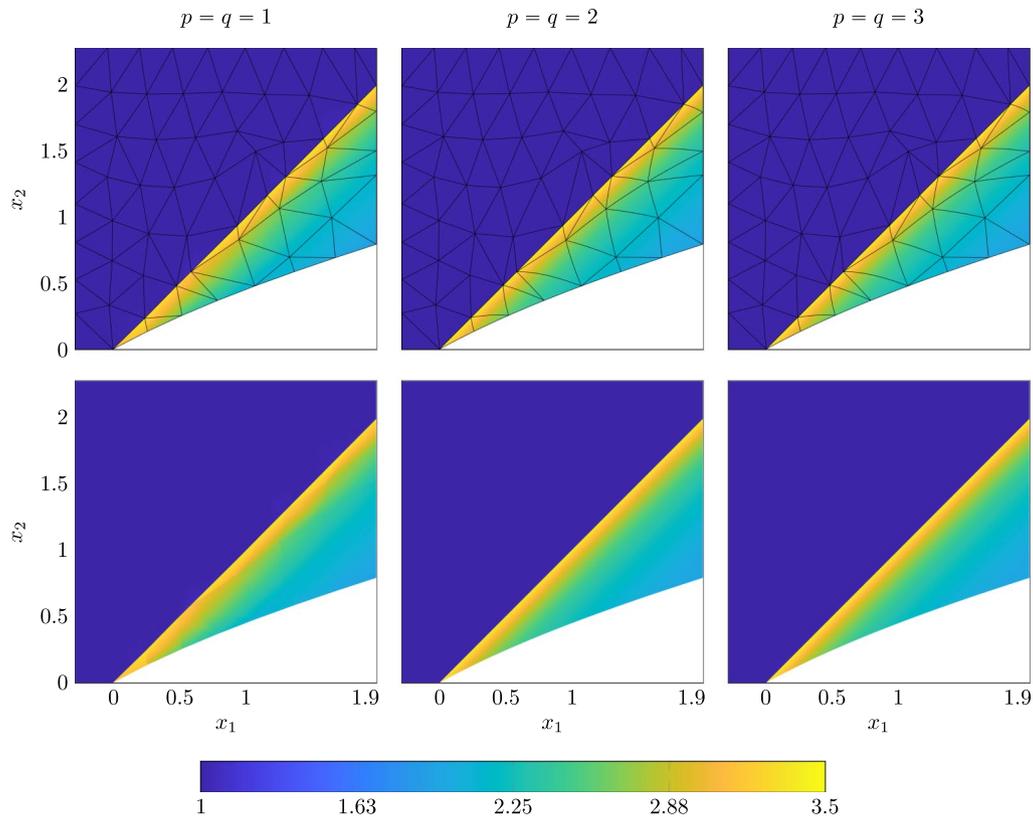Fig. 3   Solution (density ρ) of Eq. (2) using implicit tracking with a $p = q = 1$ (left), $p = q = 2$ (middle), and $p = q = 3$ (right) basis for the solution and mesh with (top) and without (bottom) element boundaries.
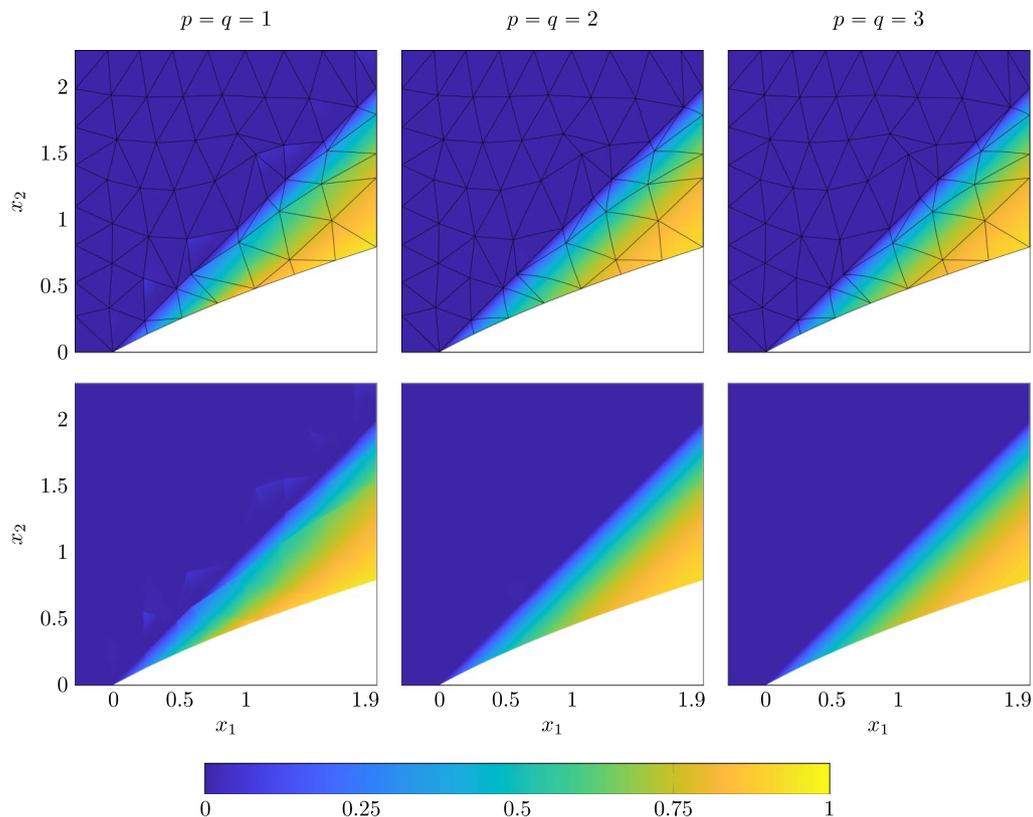
Fig. 4   Solution (reaction λ) of Eq. (2) using implicit tracking with a $p = q = 1$ (left), $p = q = 2$ (middle), and $p = q = 3$ (right) basis for the solution and mesh with (top) and without (bottom) element boundaries.

Table 2   Degrees of freedom and error metrics for high-order implicit tracking simulations applied to the reacting flow problem with a straight shock. Small (sub-1%) errors are achieved on coarse, high-order discretizations with $\mathcal{O}(10^3)$ DOF using implicit tracking, while the WENO scheme [38] requires far more DOF and results in larger errors

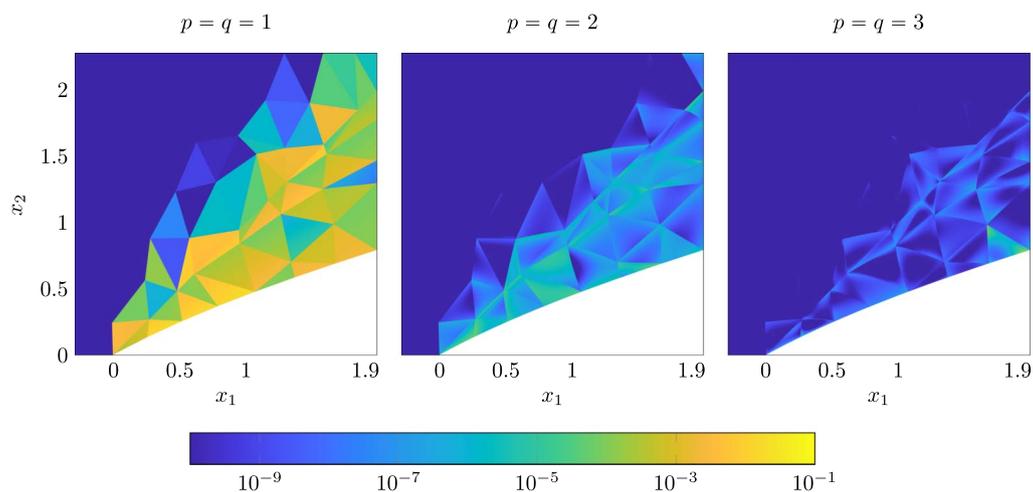| Quantity	$p = q = 1$	$p = q = 2$	$p = q = 3$	WENO (1024×1024) [38]
Mesh DOF	142	500	1076	0
Solution DOF	1635	3270	5450	5,242,880
e_H	1.50×10^{-3}	2.24×10^{-5}	2.63×10^{-6}	— —
e_ρ	5.07×10^{-2}	1.38×10^{-3}	1.97×10^{-4}	3.0×10^{-2}
e_λ	3.44×10^{-2}	1.51×10^{-3}	5.54×10^{-5}	— —

Fig. 5   Vorticity magnitude of Eq. (2) using implicit tracking with a $p = q = 1$ (left), $p = q = 2$ (middle), and $p = q = 3$ (right) basis for the solution and mesh.
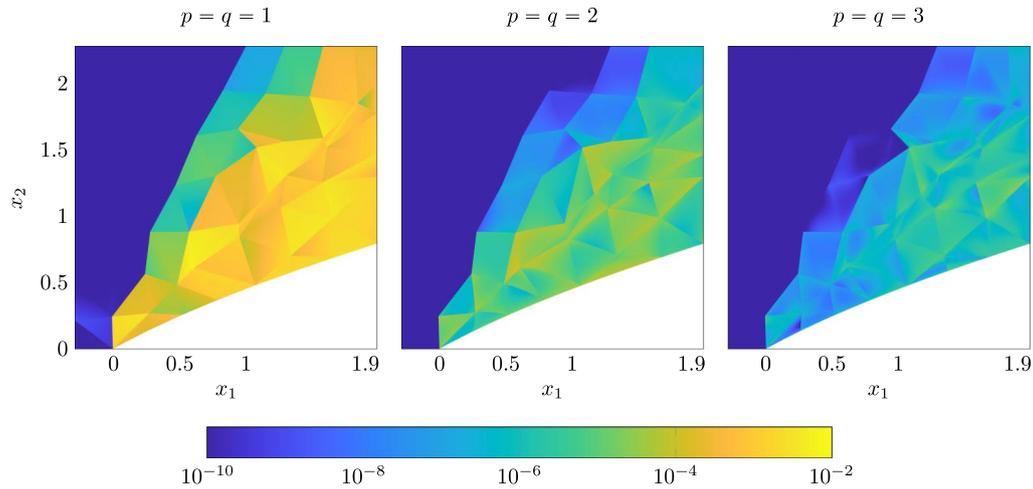
Fig. 6   Magnitude of enthalpy error ($|H - H_\infty|$) of Eq. (2) using implicit tracking with a $p = q = 1$ (left), $p = q = 2$ (middle), and $p = q = 3$ (right) basis for the solution and mesh.

enthalpy. Another source of error comes from the fact that the wall in the numerical simulation is only a piecewise polynomial approximation to the true wall that induces a straight shock (the condition under which the exact solution was derived). Despite these two sources and the inherent truncation error associated with using piecewise polynomial functions to approximate the solution, the deviation from irrotational and constant enthalpy is small, particularly for the $p = q = 3$ simulation. It is interesting to note that, despite the supersonic inlet flow, there are small errors upstream of the oblique shock, which we attribute to the globally coupled nature of the implicit shock tracking solver and smoothed numerical flux function. That is, the solver modifies all DG degrees of freedom to minimize the enriched residual, and because of the smoothed numerical flux, even degrees

of freedom upstream of the shock can impact the residual in the elements adjacent to the shock surface.

To further emphasize the improved accuracy of implicit shock tracking, for both the exact and implicit tracking solutions, we extract a slice (orthogonal to the shock) of the density ρ and reaction progress λ (Fig. 7) and plot the location of the discontinuity along with contours of the density (Fig. 8). These plots confirm that the $p = q = 1$ tracking simulation is underresolved, but the $p = q > 1$ simulations nearly perfectly capture the exact solution and do not exhibit any nonphysical oscillations (Gibbs' phenomena). Figure 8 exactly mimics Fig. 3 of Ref. [38] for a one-to-one comparison. The finite difference WENO method in Ref. [38] on a 256×256 grid (3.27×10^5 DOF) provided an acceptable approximation to the
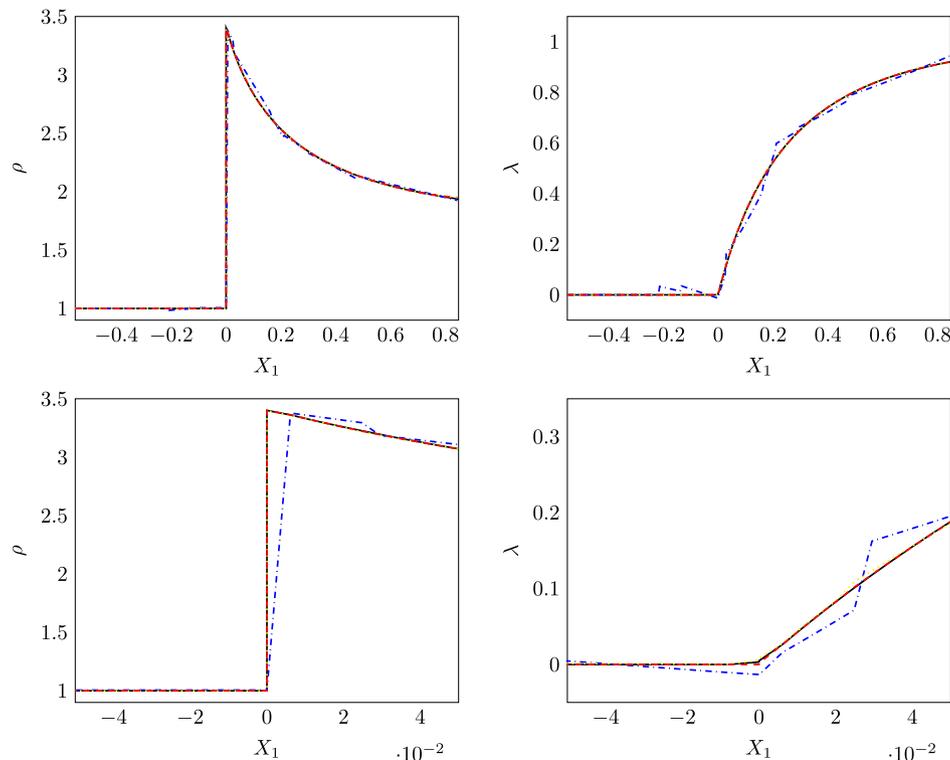

Fig. 7   Slices of density and reaction progress of the implicit tracking solutions along the line $\Gamma := \{(1.988 - t, 0.7898 + t)|t \in (0, 0.988)\}$ (t is a dummy variable that facilitates the parametrization of Γ): exact solution (——), $p = q = 1$ (– · – ·), $p = q = 2$ (· · · · ·), and $p = q = 3$ (– – –). The bottom row is a zoom of the top row at the shock location. The $p = q = 1$ solution is underresolved, while the $p = q > 1$ solutions provide accurate approximations of the solution. The $p = q = 2$ and $p = q = 3$ solutions nearly lie on top one another.
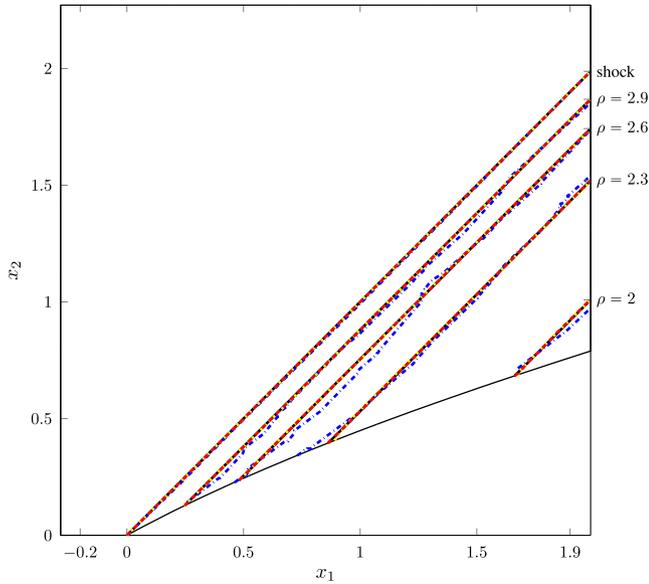
Fig. 8 Location of the discontinuity and contours of the density: exact solution (——), $p = q = 1$ (– · – ·), $p = q = 2$ (· · · · · ·), and $p = q = 3$ (– – –).

density contours; however, it was far from exact, while the implicit tracking method for $p = q > 1$ matches the true density contours almost perfectly.

C. Solver Performance

The convergence of the tracking method to a solution of the optimization problem in Eq. (36) is provided in Fig. 9. The DG residual $\|r(u, x)\|$ converges to tighter tolerances as the polynomial degree is increased; however, it proves difficult to converge the optimality condition $\|c(u, x)\|$ to a tight tolerance. This difficulty,

as discussed in Ref. [34], comes from the fact that traditional numerical flux functions are not smooth with respect to mesh deformations on shock-aligned meshes, and therefore the first-order optimality conditions are not defined. The smoothed numerical flux function improves solver performance by restoring smoothness but is not sufficient to lead to deep convergence of the optimality conditions. As the polynomial degree is increased, the enriched residual converges to a smaller value, suggesting we are approaching a solution of the weak form. The mesh distortion also improves as the polynomial degree is increased, which is expected because there are more DOF to manipulate to simultaneously track the discontinuity and smooth the mesh.

D. Grid Convergence

Finally, we investigate the convergence of the method to the exact solution under h refinement, where h is the length of the longest edge in the mesh (usually proportional to $|\mathcal{E}_{h,p}|^{-1/2}$ in $d = 2$ dimensions). This is a rigorous test for the implicit shock tracking method because even minute error, for example in the boundary conditions or numerical flux, will hinder convergence. Furthermore, it is critical that the computational setup precisely reproduces the assumptions under which the exact solution was derived [38]. The two critical assumptions used in the derivation of the exact solution were as follows:

1) The wall is curved in such a way that the shock surface is a straight line.

2) The domain is semi-infinite with prescribed far field conditions.

We easily reproduce the first condition by generating $q = 3$ (cubic) meshes that provide fourth-order-accurate approximations to the curved wall boundary derived in Ref. [38]. The semi-infinite domain cannot be reproduced in a computational setting, so we restrict the domain to the box $[-0.284, 1.99] \times [0, 2.27]$ (nondimensionalization of the domain used in Ref. [38]) and apply suitable boundary conditions on the artificial boundaries that have been introduced. Unfortunately, characteristic boundary conditions as introduced in Sec. II are most suitable when the domain boundaries are far from the region of interest. In the present setting, we consider
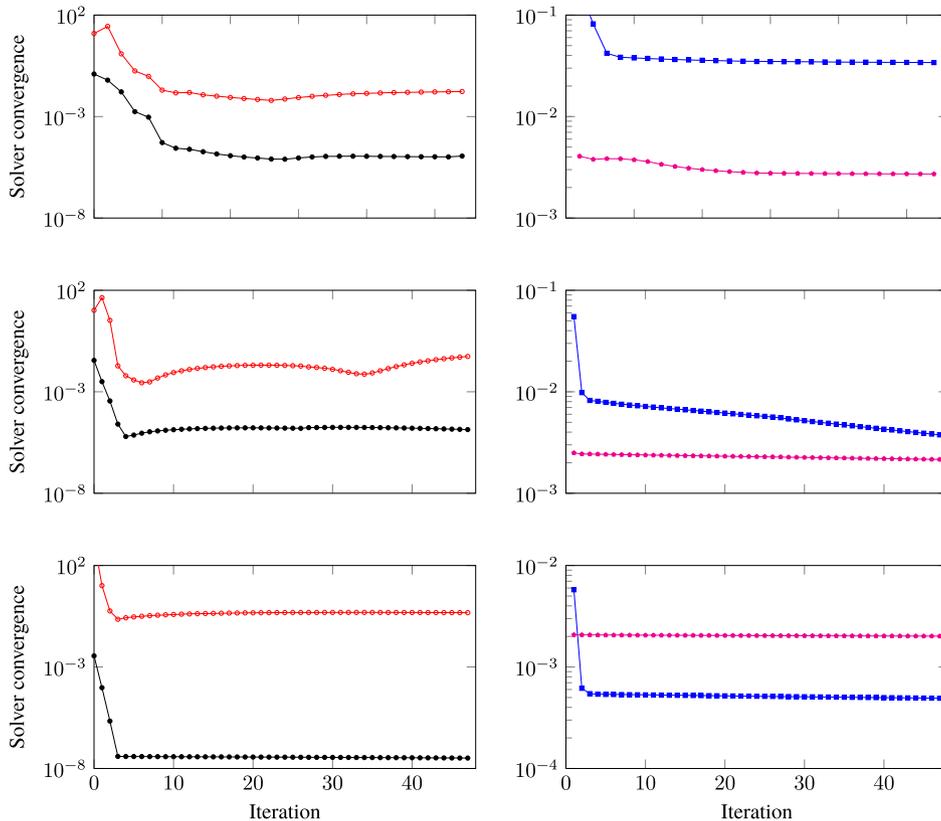

Fig. 9 Convergence of the DG residual $\|r(u, x)\|$ (——•——), enriched DG residual $\|R(u, x)\|$ (——■——), optimality condition $\|c(u, x)\|$ (——○——), and mesh distortion $\|\kappa R_{\mathrm{msh}}(x)\|$ (——+——) for the implicit tracking method with $p = q = 1$ (top), $p = q = 2$ (middle), and $p = q = 3$ (bottom).
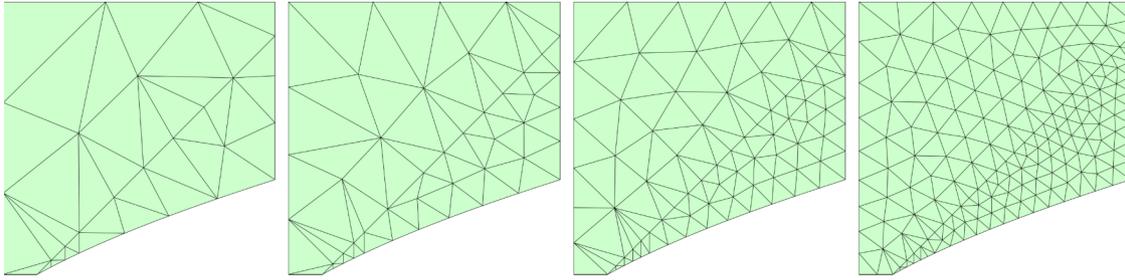
Fig. 10   Sequence of meshes used for h-refinement study. The meshes have been aligned with the exact shock surface during the mesh generation phase and refinement is targeted in the postshock region.
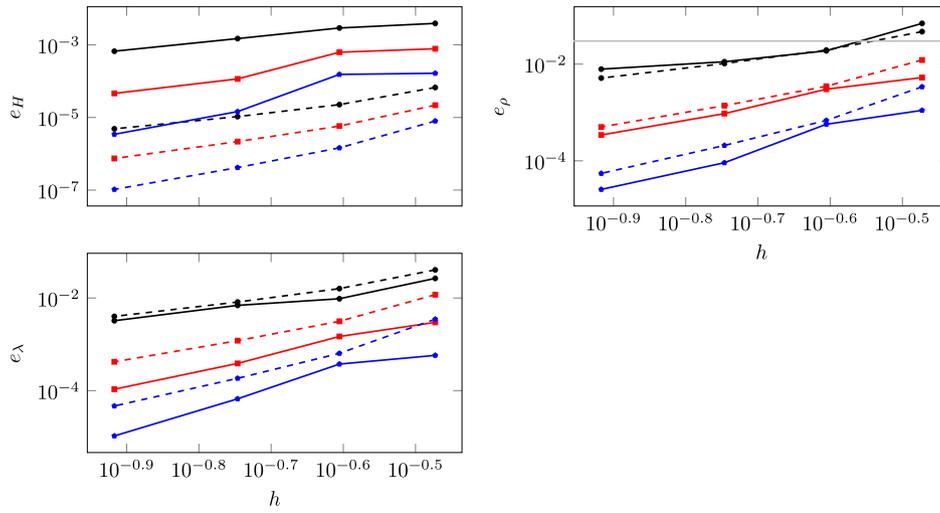

Fig. 11   Convergence of the implicit shock tracking method in the three error metrics considered under h refinement using a $q = 3$ polynomial approximation for the mesh and $p = 1$ (——•——), $p = 2$ (——■——), and $p = 3$ (——•——) solution approximation. For reference, the L^2 projection error is included using the $q = 3$ mesh approximation and $p = 1$ (– • –), $p = 2$ (– ■ –), $p = 3$ (– • –) solution approximation. The horizontal gray line corresponds to the error obtained in Ref. [38] using the fifth-order WENO scheme on a 1024×1024 grid.

the error over the entire computational domain so this will never be the case. To completely avoid this issue, we use the exact solution to prescribe a Dirichlet boundary condition on all boundaries; in other words, all boundaries are supersonic inflow boundaries with the inflow state coming from the exact solution.

To efficiently use computing resources, the sequences of meshes used in the convergence study were not generated through uniform refinement of an initial mesh. Rather, each mesh in the sequence (four in total) was generated using DistMesh with refinement targeted in the postshock region. Furthermore, since we have already demonstrated

the tracking abilities of the implicit tracking method, we use DistMesh to ensure each mesh in our sequence perfectly conforms to the shock surface in the exact solution (Fig. 10). The meshes shown in Fig. 10 are the original meshes used as the initial guess for the implicit tracking method, which moves the nodes of the mesh to minimize the enriched residual. The optimized meshes are visually indistinguishable from the original meshes and therefore not shown. Under this setup, the implicit tracking method is applied on the sequence of meshes (Fig. 10) for solution bases of degree $p = 1, 2, 3$. For means of comparison, we also compute the L^2 projection of the exact solution on each mesh in

Table 3   Summary of h-refinement convergence results for the implicit tracking method using sequence of meshes in Fig. 10

p	q	$|\mathcal{E}_{h,p}|$	h	e_H	$m(e_H)$	e_ρ	$m(e_\rho)$	e_λ	$m(e_\lambda)$
1	3	34	3.37e-01	3.91e-03	——	6.96e-02	——	2.65e-02	——
1	3	62	2.48e-01	2.92e-03	0.95	1.89e-02	4.28	9.64e-03	3.31
1	3	119	1.79e-01	1.49e-03	2.07	1.12e-02	1.60	6.93e-03	1.01
1	3	263	1.21e-01	6.70e-04	2.04	7.87e-03	0.90	3.23e-03	1.94
2	3	34	3.37e-01	7.83e-04	——	5.27e-03	——	2.98e-03	——
2	3	62	2.48e-01	6.27e-04	0.72	3.03e-03	1.82	1.47e-03	2.30
2	3	119	1.79e-01	1.16e-04	5.21	9.48e-04	3.58	3.87e-04	4.12
2	3	263	1.21e-01	4.61e-05	2.34	3.42e-04	2.59	1.07e-04	3.27
3	3	34	3.37e-01	1.65e-04	——	1.11e-03	——	5.77e-04	——
3	3	62	2.48e-01	1.54e-04	0.24	5.70e-04	2.17	3.74e-04	1.42
3	3	119	1.79e-01	1.45e-05	7.26	9.14e-05	5.64	6.65e-05	5.32
3	3	263	1.21e-01	3.44e-06	3.67	2.56e-05	3.24	1.05e-05	4.70

The slope of the curves [$m(\cdot)$] as a function of resolution is included and trends near the optimal rate $\mathcal{O}(h^{p+1})$.

Table 4 Summary of *h*-refinement convergence results for the L^2 projection of the exact solution onto the DG basis associated with each mesh in Fig. 10

p	q	$\|\mathcal{E}_{h,p}\|$	h	e_H	$m(e_H)$	e_ρ	$m(e_\rho)$	e_λ	$m(e_\lambda)$
1	3	34	3.37e-01	6.70e-05	— —	4.70e-02	— —	4.04e-02	— —
1	3	62	2.48e-01	2.26e-05	3.56	1.95e-02	2.88	1.59e-02	3.04
1	3	119	1.79e-01	1.06e-05	2.34	1.03e-02	1.97	8.17e-03	2.06
1	3	263	1.21e-01	4.87e-06	1.97	5.14e-03	1.77	4.00e-03	1.82
2	3	34	3.37e-01	2.18e-05	— —	1.22e-02	— —	1.18e-02	— —
2	3	62	2.48e-01	5.80e-06	4.34	3.48e-03	4.10	3.14e-03	4.33
2	3	119	1.79e-01	2.17e-06	3.03	1.39e-03	2.84	1.20e-03	2.97
2	3	263	1.21e-01	7.43e-07	2.73	5.01e-04	2.59	4.19e-04	2.67
3	3	34	3.37e-01	7.99e-06	— —	3.40e-03	— —	3.48e-03	— —
3	3	62	2.48e-01	1.46e-06	5.57	6.84e-04	5.25	6.38e-04	5.56
3	3	119	1.79e-01	4.17e-07	3.85	2.08e-04	3.66	1.84e-04	3.82
3	3	263	1.21e-01	1.04e-07	3.53	5.49e-05	3.39	4.66e-05	3.50

The slope of the curves $[m(\cdot)]$ as a function of resolution is included and tends to the optimal rate $\mathcal{O}(h^{p+1})$.

the sequence and the associated error metrics. The L^2 projection is included because it is, in the L^2 sense, the best we can do with the given solution basis and gives us a means to assess the additional error introduced by the DG tracking framework. The convergence results are shown in Fig. 11 for each error metric and tabulated in Table 3 (implicit tracking) and Table 4 (L^2 projection). From the figure, we see the enthalpy error of the L^2 projection of the exact solution is several orders of magnitude lower than for the implicit tracking solution; however, for the other error metrics, the two approximations are very similar with the implicit tracking solution being more accurate in several cases ($p > 1$). From Table 4, we see the L^2 projection of the exact solution obtains optimal convergence rates $\mathcal{O}(h^{p+1})$ for all three error metrics. For the implicit tracking method, the convergence is less clean, but we observe near-optimal rates (within half an order) (Table 3). The density error reported in Ref. [38] from the fifth-order WENO scheme on the 1024×1024 grid (5242880 DOF) is also included in Fig. 11; the implicit tracking framework provides a more accurate approximation on much coarser grids.

V. Conclusions

This is the first work that uses recently developed high-order implicit tracking methods [31,33,34] to solve chemically reacting flows and among the first methods to demonstrate high-order convergence for multidimensional shocked reacting flow problems. The optimization-based method is able to to robustly approximate the solution and track the discontinuity while maintaining a high-quality mesh by using a continuation strategy that initializes the high-order method with a low-order approximation. This paper shows that a mesh with only 100 elements (after collapse) is sufficient to accurately capture the simple reaction zone structure as well as accurately portray the topology of the shock wave using the tracking method; this translates to fewer than 10^4 degrees of freedom, even for the fourth-order method (cubic elements, $p = q = 3$). Furthermore, the implicit tracking method converges at near optimal rates $\mathcal{O}(h^{p+1})$ under mesh refinement. Therefore, in addition to being accurate on coarse meshes, the error decreases rapidly, particularly for polynomial degrees greater than 1.

The results represent a significant advantage over those obtained by popular WENO methods. Direct comparison of this paper's method on the same problem studied in Ref. [38] reveals the tracking method required less than 7000 total degrees of freedom to obtain an L^1 error of 10^{-4} in the density field near, while the error of the WENO method exceeded 10^{-2} using 5.2×10^6 DOF, for example, the $p = q = 3$ tracking method vs WENO in Table 2, and converges slower than first-order under h refinement.

Having demonstrated the potential of the high-order implicit shock tracking in this paper for compressible, reactive flows using the simple two-dimensional benchmark problem with an analytical exact solution, future work will turn to solving relevant problems in two and three dimensions with richer kinetic and diffusion models. Introduction of Arrhenius temperature dependency with one-step kinetics will introduce a new thin zone that must be captured. Simple two- and three-step kinetic models can better display induction and explosion zone dynamics at the expense of additional thin zones. And ultimately, the authors would like to extend to detailed chemical kinetics, which introduces challenges of simultaneously capturing thin and thick zones spanning at least four orders of magnitude. Additionally, the authors would like to introduce mass, momentum, and energy diffusion, which will require the capture of additional thin boundary layers and diffusion zones. The authors believe implicit shock tracking is very well suited for these problems. Given the generality of the implicit tracking formulation, the extension to steady, viscous problems with more complicated kinetics will be straightforward by extending the DG method. Other critically important avenues of research are extension of the tracking method to be time dependent and improvement of the underlying optimization solver. The extension to time-dependent problems will be essential to capture ignition and extinction events as well as instabilities in a hypersonic environment. To this point, time-dependent problems have mostly been considered in a monolithic space-time setting [33,34] and slab-based space-time setting [45]; however, significant improvements with regard to iterative linear solvers and convergence strategies are required for these methods to be practical. Another promising strategy to extend to time-dependent problems that directly leverages the infrastructure of the tracking method is a method of lines approach that would lead to an implicit tracking problem at each time step.

Appendix A: Smoothed Roe Flux for Reactive Euler Equations

For completeness, we detail the derivation of the Roe flux for the reactive Euler equations and introduce a smoothed version that is continuously differentiable with respect to the normal $\boldsymbol{n}$. First, define the Jacobian of the flux function $F(U)$ in Eq. (25) in the normal direction $\boldsymbol{n}$, $B(U, \boldsymbol{n}) \in \mathbb{R}^{(d+3)\times(d+3)}$,

$$B(U, \boldsymbol{n}) := \frac{\partial[F(U)\boldsymbol{n}]}{\partial U} \qquad (A1)$$

The Roe flux uses the eigenvalue decomposition of the Jacobian $B(U, \boldsymbol{n})$ to upwind based on characteristics. Even in two spatial dimensions, the eigenvalue decomposition of $B(U, \boldsymbol{n})$ for a general normal direction is complicated, even with symbolic tools. To simplify derivation of the decomposition, we transform to a coordinate system aligned with the normal direction; for example, the first transformed coordinate direction is aligned with the normal $\boldsymbol{n} = Q(\boldsymbol{n})\boldsymbol{e}_1$. The the velocity in the standard coordinate system v_i, $i = 1, 2$, transforms to the velocity in the rotated coordinate system $\tilde{v}_i$, $i = 1, 2$, as $\boldsymbol{v} = Q(\boldsymbol{n})\tilde{\boldsymbol{v}}$, where

$$
\boldsymbol{n} = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}, \quad \boldsymbol{e}^{(1)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \boldsymbol{v} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, \quad \tilde{\boldsymbol{v}} = \begin{bmatrix} \tilde{v}_1 \\ \tilde{v}_2 \end{bmatrix},
$$

$$
Q(\theta) = \begin{bmatrix} n_1 & -n_2 \\ n_2 & n_1 \end{bmatrix} \tag{A2}
$$

Because the density, energy, and reaction progress are scalars, they are invariant with respect to coordinate transformations. Therefore,

the reactive Euler state transforms as $U = \hat{Q}\,\tilde{U}$, where U is defined in Eq. (25), $\tilde{U}$ is the representation of U in the transformed coordinate system, and $\hat{Q}(\boldsymbol{n}) \in \mathbb{R}^{5 \times 5}$ is the matrix that transforms the reactive Euler state into the new coordinate system

$$
\hat{Q}(\boldsymbol{n}) := \begin{bmatrix} 1 & & & \\ & Q(\boldsymbol{n}) & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \tag{A3}
$$

From these definitions, we have

$$
F(U)\boldsymbol{n} = F(U)Q\boldsymbol{e}^{(1)} = \hat{Q}F(\tilde{U})\boldsymbol{e}^{(1)} = \hat{Q}F(\hat{Q}^T U)\boldsymbol{e}^{(1)} \tag{A4}
$$

where the first and last equalities follow from the coordinate transformation and the second equality is derived as

$$
F(U)Q = \begin{bmatrix} \rho \boldsymbol{v}^T \\ \rho \boldsymbol{v}\boldsymbol{v}^T + PI_{d \times d} \\ \rho H \boldsymbol{v}^T \\ \rho \lambda \boldsymbol{v}^T \end{bmatrix} Q = \begin{bmatrix} \rho (Q^T \boldsymbol{v})^T \\ \rho \boldsymbol{v}(Q^T \boldsymbol{v})^T + PQ \\ \rho H (Q^T \boldsymbol{v})^T \\ \rho \lambda (Q\boldsymbol{v})^T \end{bmatrix}
$$

$$
= \begin{bmatrix} \rho \tilde{\boldsymbol{v}}^T \\ \rho Q\tilde{\boldsymbol{v}}\tilde{\boldsymbol{v}}^T + PQ \\ \rho H \tilde{\boldsymbol{v}} \\ \rho \lambda \tilde{\boldsymbol{v}} \end{bmatrix} = \hat{Q}F(\tilde{U}) \tag{A5}
$$

From the flux transformation in Eq. (A4), the Jacobian transforms as

$$
B(U, \boldsymbol{n}) = \hat{Q}\,\tilde{B}(\hat{Q}^T U)\hat{Q}^T \tag{A6}
$$

where $\tilde{B}(\tilde{U}) := B(\tilde{U}, \boldsymbol{e}^{(1)})$. Let

$$
\tilde{B}(\tilde{U}) = \tilde{V}(\tilde{U})\tilde{\Lambda}(\tilde{U})[\tilde{V}(\tilde{U})]^{-1} \tag{A7}
$$

be the eigenvalue decomposition of $\tilde{B}(\tilde{U})$; then, Eq. (A6) implies

$$
B(U, \boldsymbol{n}) = V(U, \boldsymbol{n})\Lambda(U, \boldsymbol{n})[V(U, \boldsymbol{n})]^{-1} \tag{A8}
$$

is the eigenvalue decomposition of $B(U, \boldsymbol{n})$, where

$$
V(U, \boldsymbol{n}) = \hat{Q}(\boldsymbol{n})\tilde{V}(\hat{Q}(\boldsymbol{n})^T U), \quad \Lambda(U, \boldsymbol{n}) = \tilde{\Lambda}(\hat{Q}(\boldsymbol{n})^T U),
$$

$$
[V(U, \boldsymbol{n})]^{-1} = \left[\tilde{V}(\hat{Q}(\boldsymbol{n})^T U)\right]^{-1} \hat{Q}(\boldsymbol{n})^T \tag{A9}
$$

By examining these equations, it is clear the we need only derive the eigenvalue decomposition of $\tilde{B}(\tilde{U})$ to completely specify the eigenvalue decomposition of $B(U, \boldsymbol{n})$.

The transformed Jacobian matrix is

$$
\tilde{B}(\tilde{\boldsymbol{v}}, \tilde{H}, \tilde{\lambda}) := \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ \frac{(\gamma-1)}{2}\tilde{s}^2 - \tilde{v}_1^2 & (3-\gamma)\tilde{v}_1 & 0 & (\gamma-1) & (\gamma-1)\hat{q} \\ -\tilde{v}_1\tilde{v}_2 & \tilde{v}_2 & \tilde{v}_1 & 0 & 0 \\ \left[\frac{(\gamma-1)}{2}\tilde{s}^2 - \tilde{H}\right]\tilde{v}_1 & \tilde{H} - (\gamma-1)\tilde{v}_1^2 & -(\gamma-1)\tilde{v}_1\tilde{v}_2 & \gamma\tilde{v}_1 & (\gamma-1)\hat{q}\tilde{v}_1 \\ -\tilde{\lambda}\tilde{v}_1 & \tilde{\lambda} & 0 & 0 & \tilde{v}_1 \end{bmatrix} \tag{A10}
$$

where $\tilde{s} := \sqrt{\tilde{v}_i\tilde{v}_i}$ and we have used the fact that it can be written solely in terms of the velocity, enthalpy, and reaction progress to reparameterize the terms. Using the symbolic software MAPLE, we found the eigenvalue decomposition of $\tilde{B}(\tilde{\boldsymbol{v}}, \tilde{H}, \tilde{\lambda})$ to be

$$
\tilde{V}(\tilde{\boldsymbol{v}}, \tilde{H}, \tilde{\lambda}) = \begin{bmatrix} 1 & 1 & 0 & 0 & 1 \\ \tilde{v}_1 - \tilde{c} & \tilde{v}_1 & 0 & 0 & \tilde{v}_1 + \tilde{c} \\ \tilde{v}_2 & \tilde{v}_2 & 1 & 0 & \tilde{v}_2 \\ \tilde{H} - \tilde{v}_1\tilde{c} & \frac{\tilde{s}^2}{2} & \tilde{v}_2 & \hat{q} & \tilde{H} + \tilde{v}_1\tilde{c} \\ \tilde{\lambda} & 0 & 0 & -1 & \tilde{\lambda} \end{bmatrix},
$$

$$
\tilde{\Lambda}(\tilde{\boldsymbol{v}}, \tilde{H}, \tilde{\lambda}) = \begin{bmatrix} \tilde{v}_1 - \tilde{c} & & & & \\ & \tilde{v}_1 & & & \\ & & \tilde{v}_1 & & \\ & & & \tilde{v}_1 & \\ & & & & \tilde{v}_1 + \tilde{c} \end{bmatrix} \tag{A11}
$$

where the left eigenvectors are

$$
\left[\tilde{V}(\tilde{\boldsymbol{v}}, \tilde{H}, \tilde{\lambda})\right]^{-1}
$$

$$
= \frac{\gamma-1}{2\tilde{c}^2} \begin{bmatrix} \frac{\tilde{s}^2}{2} + \frac{\tilde{v}_1\tilde{c}}{\gamma-1} & -\tilde{v}_1 - \frac{\tilde{c}}{\gamma-1} & -\tilde{v}_2 & 1 & \hat{q} \\ -\tilde{s}^2 + \frac{2\tilde{c}^2}{\gamma-1} & 2\tilde{v}_1 & 2\tilde{v}_2 & -2 & -2\hat{q} \\ -\frac{2\tilde{c}^2\tilde{v}_2}{\gamma-1} & 0 & \frac{2\tilde{c}^2}{\gamma-1} & 0 & 0 \\ \tilde{\lambda}\tilde{s}^2 & -2\tilde{\lambda}\tilde{v}_1 & -2\tilde{\lambda}\tilde{v}_2 & 2\tilde{\lambda} & 2\hat{q}\tilde{\lambda} - \frac{2\tilde{c}^2}{\gamma-1} \\ \frac{\tilde{s}^2}{2} - \frac{\tilde{v}_1\tilde{c}}{\gamma-1} & -\tilde{v}_1 + \frac{\tilde{c}}{\gamma-1} & -\tilde{v}_2 & 1 & \hat{q} \end{bmatrix} \tag{A12}
$$

and the speed of sound is given by

$$
\tilde{c} := \sqrt{(\gamma-1)(\tilde{H} - \tilde{s}^2/2 + \tilde{\lambda}\hat{q})} \tag{A13}
$$

Straightforward manipulation recovers the classical result for an ideal gas $\tilde{c}^2 = \gamma\tilde{P}/\tilde{\rho}$, where $\tilde{\rho}$ and $\tilde{P}$ are the corresponding density and pressure, respectively. This leads to the final form of that decomposition that expresses the eigenvectors and eigenvalues of $B(U, \boldsymbol{n})$

in terms of the velocity, enthalpy, and reaction progress in the physical coordinates

$$V(\boldsymbol{v}, H, \lambda, \boldsymbol{n}) = \hat{Q}(\boldsymbol{n})\tilde{V}(Q(\boldsymbol{n})^T\boldsymbol{v}, H, \lambda),$$

$$\Lambda(\boldsymbol{v}, H, \lambda, \boldsymbol{n}) = \tilde{\Lambda}(Q(\boldsymbol{n})^T\boldsymbol{v}, H, \lambda),$$

$$[V(\boldsymbol{v}, H, \lambda, \boldsymbol{n})]^{-1} = \left[\tilde{V}(Q(\boldsymbol{n})^T\boldsymbol{v}, H, \lambda)\right]^{-1}\hat{Q}(\boldsymbol{n})^T \quad (A14)$$

Finally, the Roe flux is

$$\mathcal{F}^{\text{roe}}(U^+, U^-, \boldsymbol{n}) := \frac{1}{2}(F(U^+)\boldsymbol{n} + F(U^-)\boldsymbol{n})$$
$$+ \frac{1}{2}|A(U^+, U^-, \boldsymbol{n})|(U^+ - U^-) \quad (A15)$$

where the matrix

$$A(U^+, U^-, \boldsymbol{n}) := B(\boldsymbol{v}^\star(U^+, U^-), H^\star(U^+, U^-), \lambda^\star(U^+, U^-), \boldsymbol{n})$$
$$(A16)$$

and the Roe averages are defined as

$$\boldsymbol{v}^\star(U^+, U^-) := \frac{\sqrt{\rho^+}\boldsymbol{v}^+ + \sqrt{\rho^-}\boldsymbol{v}^-}{\sqrt{\rho^+} + \sqrt{\rho^-}},$$

$$H^\star(U^+, U^-) := \frac{\sqrt{\rho^+}H^+ + \sqrt{\rho^-}H^-}{\sqrt{\rho^+} + \sqrt{\rho^-}}, \quad (A17)$$

$$\lambda^\star(U^+, U^-) := \frac{\sqrt{\rho^+}\lambda^+ + \sqrt{\rho^-}\lambda^-}{\sqrt{\rho^+} + \sqrt{\rho^-}}$$

where $\boldsymbol{v}^\pm$, $H^\pm$, and $\lambda^\pm$ are the velocity, enthalpy, and reaction progress computed from the $U^\pm$ state, respectively. The matrix absolute value is defined in terms of the eigenvalue decomposition of the matrix as

$$|A(U^+, U^-, \boldsymbol{n})| = V(\boldsymbol{v}^\star, H^\star, \lambda^\star, \boldsymbol{n})|\Lambda(\boldsymbol{v}^\star, H^\star, \lambda^\star, \boldsymbol{n})|$$
$$\times [V(\boldsymbol{v}^\star, H^\star, \lambda^\star, \boldsymbol{n})]^{-1} \quad (A18)$$

where the absolute value is applied entrywise to a diagonal matrix. Finally, we define the smoothed Roe flux as

$$\mathcal{F}^{\text{roe,sm}}(U^+, U^-, \boldsymbol{n}) := \frac{1}{2}(F(U^+)\boldsymbol{n} + F(U^-)\boldsymbol{n})$$
$$+ \frac{1}{2}|A(U^+, U^-, \boldsymbol{n})|_s(U^+ - U^-) \quad (A19)$$

where

$$|A(U^+, U^-, \boldsymbol{n})|_s = V(\boldsymbol{v}^\star, H^\star, \lambda^\star, \boldsymbol{n})|\Lambda(\boldsymbol{v}^\star, H^\star, \lambda^\star, \boldsymbol{n})|_s$$
$$\times [V(\boldsymbol{v}^\star, H^\star, \lambda^\star, \boldsymbol{n})]^{-1} \quad (A20)$$

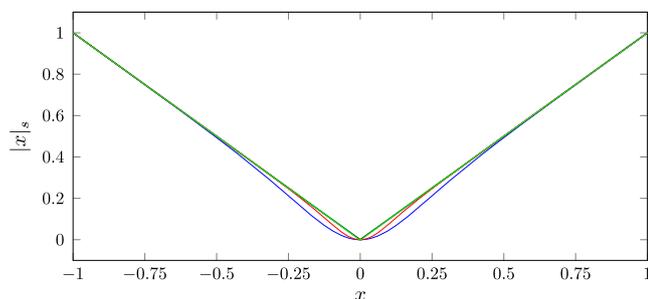**Fig. A1 Smoothed absolute value function for $k = 5$ (blue line), $k = 10$ (red line), and $k = 100$ (green line).**

and $|\cdot|_s : \mathbb{R} \to \mathbb{R}$ is a smooth approximation to the absolute value given by

$$|x|_s := x \tanh(kx) \quad (A21)$$

where $k = 100$ is a smoothness parameter (Fig. A1), applied entrywise to a diagonal matrix.

Acknowledgments

References

[1] Powers, J. M., and Paolucci, S., "Accurate Spatial Resolution Estimates for Reactive Supersonic Flow with Detailed Chemistry," *AIAA Journal*, Vol. 43, No. 5, 2005, pp. 1088–1099.
https://doi.org/10.2514/1.11641

[2] Romick, C. M., Aslam, T. D., and Powers, J. M., "Verified and Validated Calculation of Unsteady Dynamics of Viscous Hydrogen–Air Detonations," *Journal of Fluid Mechanics*, Vol. 769, April 2015, pp. 154–181.
https://doi.org/10.1017/jfm.2015.114

[3] Menikoff, R., "Errors When Shock Waves Interact due to Numerical Shock Width," *SIAM Journal on Scientific Computing*, Vol. 15, No. 5, 1994, pp. 1227–1242.
https://doi.org/10.1137/0915075

[4] Singh, S., Rastigejev, Y., Paolucci, S., and Powers, J. M., "Viscous Detonation in H2–O2–Ar Using Intrinsic Low-Dimensional Manifolds and Wavelet Adaptive Multilevel Representation," *Combustion Theory and Modelling*, Vol. 5, No. 2, 2001, pp. 163–184.
https://doi.org/10.1088/1364-7830/5/2/303

[5] Von Neumann, J., and Richtmyer, R. D., "A Method for the Numerical Calculation of Hydrodynamic Shocks," *Journal of Applied Physics*, Vol. 21, No. 3, 1950, pp. 232–237.
https://doi.org/10.1063/1.1699639

[6] Tadmor, E., "Shock Capturing by the Spectral Viscosity Method," *Computer Methods in Applied Mechanics and Engineering*, Vol. 80, Nos. 1–3, 1990, pp. 197–208.
https://doi.org/10.1016/0045-7825(90)90023-F

[7] Persson, P.-O., and Peraire, J., "Sub-Cell Shock Capturing for Discontinuous Galerkin Methods," AIAA Paper 2006-112, Jan. 2006.
https://doi.org/10.2514/6.2006-112

[8] Barter, G. E., and Darmofal, D. L., "Shock Capturing with PDE-Based Artificial Viscosity for DGFEM: Part I. Formulation," *Journal of Computational Physics*, Vol. 229, No. 5, 2010, pp. 1810–1827.
https://doi.org/10.1016/j.jcp.2009.11.010

[9] Shu, C.-W., and Osher, S., "Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes," *Journal of Computational Physics*, Vol. 77, No. 2, 1988, pp. 439–471.
https://doi.org/10.1016/0021-9991(88)90177-5

[10] Shu, C.-W., and Osher, S., "Efficient Implementation of Essentially Non-Oscillatory Shock-Capturing Schemes, II," *Journal of Computational Physics*, Vol. 83, No. 1, 1989, pp. 32–78.
https://doi.org/10.1016/0021-9991(89)90222-2

[11] Jiang, G.-S., and Shu, C.-W., "Efficient Implementation of Weighted ENO Schemes," *Journal of Computational Physics*, Vol. 126, No. 1, 1996, pp. 202–228.
https://doi.org/10.1006/jcph.1996.0130

[12] Xu, S., Aslam, T., and Stewart, D. S., "High Resolution Numerical Simulation of Ideal and Non-Ideal Compressible Reacting Flows with Embedded Internal Boundaries," *Combustion Theory and Modelling*, Vol. 1, No. 1, 1997, pp. 113–142.
https://doi.org/10.1080/713665233

[13] Henrick, A. K., Aslam, T. D., and Powers, J. M., "Mapped Weighted Essentially Non-Oscillatory Schemes: Achieving Optimal Order near Critical Points," *Journal of Computational Physics*, Vol. 207, No. 2, 2005, pp. 542–567.
https://doi.org/10.1016/j.jcp.2005.01.023

[14] van Leer, B., "Towards the Ultimate Conservative Difference Scheme. V. A Second-Order Sequel to Godunov's Method," *Journal of Computational Physics*, Vol. 32, No. 1, 1979, pp. 101–136.
https://doi.org/10.1016/0021-9991(79)90145-1

[15] Cambier, J.-L., Adelman, H. G., and Menees, G. P., "Numerical Simulations of Oblique Detonations in Supersonic Combustion Chambers," *Journal of Propulsion and Power*, Vol. 5, No. 4, 1989, pp. 482–491.
https://doi.org/10.2514/3.23180

[16] Cockburn, B., and Shu, C.-W., "TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws II: General Framework," *Mathematics of Computation*, Vol. 52, No. 186, 1989, pp. 411–435.
https://doi.org/10.2307/2008474

[17] Yee, H., Klopfer, G. H., and Montagné, J.-L., "High-Resolution Shock-Capturing Schemes for Inviscid and Viscous Hypersonic Flows," *Journal of Computational Physics*, Vol. 88, No. 1, 1990, pp. 31–61.
https://doi.org/10.1016/0021-9991(90)90241-R

[18] Burbeau, A., Sagaut, P., and Bruneau, C.-H., "A Problem-Independent Limiter for High-Order Runge–Kutta Discontinuous Galerkin Methods," *Journal of Computational Physics*, Vol. 169, No. 1, 2001, pp. 111–150.
https://doi.org/10.1006/jcph.2001.6718

[19] LeVeque, R. J., *Finite Volume Methods for Hyperbolic Problems*, Cambridge Univ. Press, Cambridge, New York, 2002.

[20] Berger, M. J., and Oliger, J., "Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations," *Journal of Computational Physics*, Vol. 53, No. 3, 1984, pp. 484–512.
https://doi.org/10.1016/0021-9991(84)90073-1

[21] Berger, M. J., and Colella, P., "Local Adaptive Mesh Refinement for Shock Hydrodynamics," *Journal of Computational Physics*, Vol. 82, No. 1, 1989, pp. 64–84.
https://doi.org/10.1016/0021-9991(89)90035-1

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

[23] Salas, M. D., "Shock Fitting Method for Complicated Two-Dimensional Supersonic Flows," *AIAA Journal*, Vol. 14, No. 5, 1976, pp. 583–588.
https://doi.org/10.2514/3.61399

[24] Moretti, G., "Computation of Flows with Shocks," *Annual Review of Fluid Mechanics*, Vol. 19, No. 1, 1987, pp. 313–337.
https://doi.org/10.1146/annurev.fl.19.010187.001525

[25] Kopriva, D. A., "Shock-Fitted Multidomain Solution of Supersonic Flows," *Computer Methods in Applied Mechanics and Engineering*, Vol. 175, Nos. 3–4, 1999, pp. 383–394.
https://doi.org/10.1016/S0045-7825(98)00362-4

[26] Dadone, A., and Fortunato, B., "Three-Dimensional Flow Computations with Shock Fitting," *Computers & Fluids*, Vol. 23, No. 3, 1994, pp. 539–550.
https://doi.org/10.1016/0045-7930(94)90018-3

[27] Brooks, G. P., and Powers, J. M., "Standardized Pseudospectral Formulation of the Inviscid Supersonic Blunt Body Problem," *Journal of Computational Physics*, Vol. 197, No. 1, 2004, pp. 58–85.
https://doi.org/10.1016/j.jcp.2003.11.017

[28] Paciorri, R., and Bonfiglioli, A., "A Shock-Fitting Technique for 2D Unstructured Grids," *Computers & Fluids*, Vol. 38, No. 3, 2009, pp. 715–726.
https://doi.org/10.1016/j.compfluid.2008.07.007

[29] Bonfiglioli, A., Grottadaurea, M., Paciorri, R., and Sabetta, F., "An Unstructured, Three-Dimensional, Shock-Fitting Solver for Hypersonic Flows," *Computers & Fluids*, Vol. 73, March 2013, pp. 162–174.
https://doi.org/10.1016/j.compfluid.2012.12.022

[30] Henrick, A. K., Aslam, T. D., and Powers, J. M., "Simulations of Pulsating One-Dimensional Detonations with True Fifth Order Accuracy," *Journal of Computational Physics*, Vol. 213, No. 1, 2006, pp. 311–329.
https://doi.org/10.1016/j.jcp.2005.08.013

[31] Zahr, M. J., and Persson, P.-O., "An Optimization-Based Approach for High-Order Accurate Discretization of Conservation Laws with Discontinuous Solutions," *Journal of Computational Physics*, Vol. 365, July 2018, pp. 105–134.
https://doi.org/10.1016/j.jcp.2018.03.029

[32] Powers, J. M., "Review of Multiscale Modeling of Detonation," *Journal of Propulsion and Power*, Vol. 22, No. 6, 2006, pp. 1217–1229.
https://doi.org/10.2514/1.17897

[33] Corrigan, A., Kercher, A. D., and Kessler, D. A., "A Moving Discontinuous Galerkin Finite Element Method for Flows with Interfaces," *International Journal for Numerical Methods in Fluids*, Vol. 89, No. 9, 2019, pp. 362–406.
https://doi.org/10.1002/fld.4697

[34] Zahr, M. J., Shi, A., and Persson, P.-O., "Implicit Shock Tracking Using an Optimization-Based High-Order Discontinuous Galerkin Method," *Journal of Computational Physics*, Vol. 410, March 2020, Paper 109385.
https://doi.org/10.1016/j.jcp.2020.109385

[35] Corrigan, A. T., Kercher, A. D., Kessler, D. A., and Wood-Thomas, D. A., "Convergence of the Moving Discontinuous Galerkin Method with Interface Condition Enforcement in the Presence of an Attached Curved Shock," AIAA Paper 2019-3207, June 2019.
https://doi.org/10.2514/6.2019-3207

[36] Kercher, A., Corrigan, A. T., and Kessler, D. A., "The Moving Discontinuous Galerkin Method with Interface Condition Enforcement for Viscous Flows," AIAA Paper 2020-1315, Jan. 2020.
https://doi.org/10.2514/6.2020-1315

[37] Nocedal, J., and Wright, S. J., *Numerical Optimization*, 2nd ed., Springer-Verlag, New York, 2006.

[38] Powers, J. M., and Aslam, T. D., "Exact Solution for Multidimensional Compressible Reactive Flow for Verifying Numerical Algorithms," *AIAA Journal*, Vol. 44, No. 2, 2006, pp. 337–344.
https://doi.org/10.2514/1.14404

[39] Powers, J. M., and Stewart, D. S., "Approximate Solutions for Oblique Detonations in the Hypersonic Limit," *AIAA Journal*, Vol. 30, No. 3, 1992, pp. 726–736.
https://doi.org/10.2514/3.10978

[40] Cockburn, B., and Shu, C.-W., "Runge–Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems," *Journal of Scientific Computing*, Vol. 16, No. 3, 2001, pp. 173–261.
https://doi.org/10.1023/A:1012873910884

[41] Hesthaven, J. S., and Warburton, T., *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, Springer-Verlag, New York, 2008.

[42] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, No. 2, 1981, pp. 357–372.
https://doi.org/10.1016/0021-9991(81)90128-5

[43] Grismer, M. J., "Numerical Simulations of Steady and Unsteady Oblique Detonation Phenomena with Application to Propulsion," Ph.D. Dissertation, Univ. of Notre Dame, Notre Dame, IN, 1996.

[44] Persson, P.-O., and Strang, G., "A Simple Mesh Generator in MATLAB," *SIAM Review*, Vol. 46, No. 2, 2004, pp. 329–345.
https://doi.org/10.1137/S0036144503429121

[45] Corrigan, A. T., Kercher, A., and Kessler, D. A., "The Moving Discontinuous Galerkin Method with Interface Condition Enforcement for Unsteady Three-Dimensional Flows," AIAA Paper 2019-0642, Jan. 2019.
https://doi.org/10.2514/6.2019-0642

P. J. Drummond
Associate Editor