INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **57**:1115–1135
Published online 13 June 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1850

Sub-cell shock capturing and spacetime discontinuity tracking for nonlinear conservation laws

J. Palaniappan[1], S. T. Miller[2] and R. B. Haber[2,*,†]

[1]*Belcan Corporation, Caterpillar Champaign Simulation Center, Champaign, IL 61820, U.S.A.*
[2]*Department of Mechanical Science & Engineering, The University of Illinois at Urbana-Champaign, Urbana, IL 61801, U.S.A.*

SUMMARY

We describe two approaches to resolving shocks and other discontinuities in spacetime discontinuous Galerkin (SDG) methods for nonlinear conservation laws. The first is an adaptation of the sub-cell shock-capturing technique, recently introduced by Persson and Peraire, to the special circumstances of SDG solutions constructed on causal spacetime grids. We restrict the stabilization operator to spacetime element interiors, thereby preserving the $\mathcal{O}(N)$ computational complexity and the element-wise conservation properties of the basic SDG method, and use a special discontinuity indicator to limit the stabilization to elements traversed by discontinuous solution features. The method resolves discontinuities within individual spacetime elements having a sufficiently high-order basis. Numerical studies demonstrate the combination of sub-cell shock capturing with h-adaptive spacetime meshing that circumvents the projection errors inherent to purely spatial remeshing procedures.

In a second method, we use adaptive spacetime meshing operations to track the trajectories of singular surfaces while maintaining the quality of the surrounding mesh. We present a series of feasibility studies where we track shocks and contact discontinuities in solutions to the inviscid Euler equations, including an example where the trajectories are not known *a priori*. The SDG-tracking method sharply resolves discontinuities without mesh refinement and requires very little stabilization. Copyright © 2008 John Wiley & Sons, Ltd.

Received 14 November 2007; Revised 9 April 2008; Accepted 21 April 2008

KEY WORDS: discontinuous Galerkin; finite element methods; stabilized methods; mesh adaptation; Euler flow

*Correspondence to: R. B. Haber, Department of Mechanical Science & Engineering, 1206 West Green Street, Urbana, IL 61801, U.S.A.
†E-mail: r-haber@uiuc.edu

Contract/grant sponsor: The U.S. National Science Foundation; contract/grant numbers: ITR/AP DMR 01-21695, ITR/AP DMR 03-25939

1. INTRODUCTION

Wave propagation problems involving shocks and other moving singular features present formidable challenges in numerical methods for partial differential equations. At a fundamental level, difficulties arise when the discrete solution space used in a given numerical method is too far removed from the solution space of the underlying continuum problem. For example, nonlinear conservation laws have solutions in the space of functions of bounded variation (BV space), a space that admits solutions with jumps and nonsmooth features that are generally in motion. Conforming finite element methods, on the other hand, are usually constructed in discrete sub-spaces of Sobolev spaces, where Galerkin projections of discontinuous BV functions may exhibit Gibbs phenomena in the form of overshoot, undershoot and spurious oscillations.

In this paper, we explore two distinct approaches to numerical modeling of discontinuous solution features in nonlinear conservation laws: *capturing methods* and *tracking schemes*. Capturing methods generate 'smeared' models of discontinuous solution features, typically over several computational cells, and require strong mesh refinement for accuracy. We classify a scheme as a *capturing method*, whether the discrete solution space is conforming or not, if no special alignment between the numerical grid and discontinuous solution features is enforced. An extensive literature describes methods for stabilizing hyperbolic problems and suppressing numerical artifacts at shocks in the capturing approach; see, for example [1–4]. An effective stabilization scheme must be sufficiently powerful to control spurious artifacts without being overly diffusive. Shock-capturing operators are one means to address this requirement [5, 6]. Adaptive grid refinement improves the performance of capturing methods and may be essential when uniform mesh refinement is too costly.

Tracking schemes use grids aligned with discontinuous solution features and nonconforming bases to represent jumps explicitly. When successful, this approach eliminates the need for intense mesh refinement around shocks and can reduce the need for extra stabilization to control Gibbs effects. However, the requirement for mesh alignment with evolving solution features is a significant complication of the mesh-generation problem. De-Kang proposed a shock-tracking scheme based on a Godunov solver [7]; LeVeque and Shyue implemented shock tracking in two spatial dimensions within a finite volume scheme [8]. Finite element methods based on discrete spaces that admit discontinuities, including extended finite element, partition of unity and discontinuous spectral element schemes [9–11], have been used to implement tracking, usually in only one spatial dimension and with relatively simple shock systems.

The methods described in this paper are based on spacetime discontinuous Galerkin (SDG) finite element methods. Discontinuous Galerkin methods for hyperbolic problems offer element-wise conservation and support high-order discretizations on compact stencils. They also gain some degree of stabilization from upwind enforcement of jump conditions across inter-element boundaries, although this intrinsic stabilization is generally insufficient for problems with shocks. Cockburn and Shu introduce Runge–Kutta discontinuous Galerkin (RKDG) methods for nonlinear conservation laws that use slope limiters to guarantee monotone solutions [12]. Qiu and Shu propose a WENO limiter to stabilize the RKDG method [13].

Van der Vegt and van der Ven apply SDG methods with moving meshes to solve the Euler equations on three-dimensional domains partitioned into uniform-duration slabs [14]; they use residual-based indicators to modulate nonlinear stabilization for shocks. Lowrie *et al.* propose an SDG scheme for nonlinear conservation laws with a compact limiter that enforces monotonicity [15]. Palaniappan and coworkers propose a similar SDG scheme for scalar conservation laws on

DOI: 10.1002/fld

unstructured spacetime grids in which a different limiter provides the necessary stabilization [16]; the SDG method is extended to systems of conservation laws. In these methods, the unstructured spacetime mesh must satisfy a causality constraint to enable an advancing front, patch-by-patch solution scheme with linear complexity in the number of patches (a patch is a small collection of contiguous elements, cf. Section 3.3). See [17, 18] for analyses of SDG methods.

In this paper we use the SDG formulation to develop our shock-capturing and shock-tracking methods. In the capturing method, we stabilize the basic SDG formulation with a variant of the 'sub-cell' artificial viscosity operator introduced in [19]; this preserves the element-wise and patch-wise conservation properties of the SDG method. The sub-cell nomenclature refers to the ability of this approach to resolve shocks within a single element of sufficiently high polynomial order when a shock traverses the element's interior. Our pilot study of an SDG-tracking method uses recent advances in adaptive spacetime meshing [20, 21] to cover the trajectories of discontinuous solution features with inter-element boundaries. This approach exploits the discontinuous features of SDG solution spaces to deliver sharp realizations of shocks without extensive mesh refinement.

The remainder of this paper is organized as follows. Section 2 summarizes the basic SDG formulation for systems of conservation laws, including specializations to Burgers equation and the inviscid Euler equations. In Section 3, we formulate the SDG sub-cell shock capturing scheme and, in combination with h-adaptive spacetime meshing, demonstrate its effectiveness and convergence properties. Section 4 presents our pilot study of discontinuity tracking in spacetime. Conclusions and recommendations for future development appear in Section 5.

2. FORMULATION

This section reviews the formulation of the basic SDG method for systems of conservation laws; the addition of necessary stabilization is deferred until Section 3. We first review special notations for differential forms and for n-tuples of forms to facilitate our spacetime formulation. A generic spacetime formulation for systems of conservation laws, including governing equations, jump conditions and initial/boundary conditions, is presented in Section 2.2; the discrete SDG formulation is summarized in Section 2.3.

2.1. Differential forms and n-tuple notations

We use the notation of differential forms on spacetime manifolds to develop the SDG formulation. This approach delivers a direct, coordinate-free notation that greatly simplifies the formulation on unstructured spacetime grids. In contrast to alternative notations, the Stokes Theorem expressed in forms notation does not require unit vectors 'normal' to spacetime d-manifolds. Such objects are not well defined, given the absence of an inner product for general spacetime vectors in classical mechanics. We summarize the notation here; see [22, 23] for a more complete exposition of differential forms and the exterior calculus on manifolds and [16] for a more detailed exposition of their use in formulations of SDG methods.

Consider a flat spacetime manifold $\mathscr{D} \subset \mathscr{M} := \mathbb{E}^d \times \mathbb{R}$ in which d is the spatial dimension of the manifold. We use the basis $\{\mathbf{e}_i, \mathbf{e}_t\}_{i=1}^d$ to represent *vectors* in the tangent space $\mathscr{T}$, which is the same for all points $P \in \mathscr{M}$. The dual basis for *covectors* in $\mathscr{T}^*$ is $\{\mathbf{e}^i, \mathbf{e}^t\}_{i=1}^d$ where $\mathbf{e}^i(\mathbf{e}_j) = \delta^i_j$, $\mathbf{e}^i(\mathbf{e}_t) = 0$, $\mathbf{e}^t(\mathbf{e}_i) = 0$ and $\mathbf{e}^t(\mathbf{e}_t) = 1$. Thus, the component representations of any vector $\mathbf{a} \in \mathscr{T}$ and any covector $\mathbf{b} \in \mathscr{T}^*$ are $\mathbf{a} = a^i \mathbf{e}_i + a^t \mathbf{e}_t$ and $\mathbf{b} = b_i \mathbf{e}^i + b_t \mathbf{e}^t$; summation from 1 to d is implied for

DOI: 10.1002/fld

indices repeated between subscripts and superscripts, excepting the reserved index t for which no summation is implied.

The standard basis for r-covectors is denoted by $\{\mathbf{e}^\lambda\}$, in which $\lambda = i_1 \ldots i_r$ is a strictly increasing r-index. Any r-covector $\boldsymbol{\omega}$ has a unique component representation with respect to the standard basis, $\boldsymbol{\omega} = \omega_\lambda \mathbf{e}^\lambda$, in which summation over strictly increasing r-indices is implied. We use '$\wedge$' to denote the usual exterior product operator and $\mathbf{d}$ to denote the exterior derivative (cf. Spivak [23]).

A *differential r-form on* $\mathscr{D}$ (*with scalar coefficients*) is an r-covector field on $\mathscr{D}$. The standard basis for 1-forms is $\{\mathrm{d}x^i, \mathrm{d}t\}_{i=1}^d$, where for our flat manifold the $\mathrm{d}x^i$ are 1-forms with uniform values $\mathbf{e}^i$, and $\mathrm{d}t$ is the 1-form with uniform value $\mathbf{e}^t$. Thus, any 1-form with scalar coefficients has the unique component representation, $\boldsymbol{\omega} = \omega_i \, \mathrm{d}x^i + \omega_t \, \mathrm{d}t$, in which ω_i and ω_t are scalar fields on $\mathscr{D}$. Top forms in spacetime are $(d+1)$-forms, for which the standard basis is the singleton set $\{\boldsymbol{\Omega}\}$, where $\boldsymbol{\Omega} = \mathrm{d}x^1 \wedge \cdots \wedge \mathrm{d}x^d \wedge \mathrm{d}t$. A top-form $\boldsymbol{\alpha}$ with scalar coefficients is expressed as $\boldsymbol{\alpha} = \alpha\boldsymbol{\Omega}$ in which α is a scalar function on $\mathscr{D}$.

We use forms with scalar, vector and tensor coefficients. The usual definition of the exterior product operator addresses forms with scalar coefficients; here we extend the definition to address forms with tensor coefficients. Let $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ be r- and s-forms on $\mathscr{D}$, respectively, with $r, s \in \{x \in \mathbb{N} : x \leqslant d+1\}$, and let $\mathbf{a}$ and $\mathbf{b}$ be tensor fields on $\mathscr{D}$ of order m and n, respectively, where the $m \geqslant n \geqslant 0$. We write $\mathbf{a}\boldsymbol{\alpha}$ and $\mathbf{b}\boldsymbol{\beta}$ to describe an r-form and an s-form with tensor coefficients. The *exterior product of* $\mathbf{a}\boldsymbol{\alpha}$ *and* $\mathbf{b}\boldsymbol{\beta}$ is the $(r+s)$-form with tensor coefficients of order $m-n$ given by $\mathbf{a}\boldsymbol{\alpha} \wedge \mathbf{b}\boldsymbol{\beta} := \mathbf{a}(\mathbf{b})(\boldsymbol{\alpha} \wedge \boldsymbol{\beta})$. A *magnitude* operator for forms with scalar or vector coefficients is defined by $|\mathbf{a}\boldsymbol{\alpha}| := |\mathbf{a}|\boldsymbol{\alpha}$.

The *Hodge star operator*, denoted as $\star$, is defined by $\boldsymbol{\alpha} \wedge \star\boldsymbol{\alpha} = \boldsymbol{\Omega}$, in which $\star\boldsymbol{\alpha}$ is a $(d+1-r)$-form. We shall have use of d-forms, for which we define a preferred basis $\{\star\mathrm{d}x^i, \star\mathrm{d}t\}_{i=1}^d$. This implies that $\mathrm{d}x^j \wedge \star\mathrm{d}x^k = \delta^{jk}\boldsymbol{\Omega}$, $\mathrm{d}t \wedge \star\mathrm{d}x^j = \mathbf{0}$ and $\mathrm{d}x^j \wedge \star\mathrm{d}t = \mathbf{0}$ for $j, k = 1, \ldots, d$. We define the action of the Hodge star operator on a form with tensor coefficients as $\star(\mathbf{a}\boldsymbol{\alpha}) = \mathbf{a}\star\boldsymbol{\alpha}$.

A useful 1-form with vector coefficients and an associated d-form with vector coefficients are

$$\mathbf{dx} := \mathbf{e}_i \, \mathrm{d}x^i, \quad \star\mathbf{dx} := \mathbf{e}_i \star \mathrm{d}x^i \tag{1}$$

Given any differentiable scalar field w on $\mathscr{D}$, we find

$$\mathbf{d}(w \star \mathbf{dx}) = (\nabla w)\boldsymbol{\Omega}, \quad \mathbf{d}(w \star \mathrm{d}t) = \dot{w}\boldsymbol{\Omega} \tag{2}$$

for any differentiable tensor field $\mathbf{a}$ on $\mathscr{D}$ of order one or greater, we have

$$\mathbf{d}(\mathbf{a} \wedge \star\mathbf{dx}) = (\boldsymbol{\nabla} \cdot \mathbf{a})\boldsymbol{\Omega}, \quad \mathbf{d}(\mathbf{a} \wedge \star\mathrm{d}t) = \dot{\mathbf{a}}\boldsymbol{\Omega} \tag{3}$$

For any open region $\mathscr{Q} \subset D$ with a regular boundary $\partial\mathscr{Q}$ and any d-form $\boldsymbol{\omega}$ with either scalar or tensor coefficients, we have *Stokes*' *Theorem* [23]:

$$\int_{\mathscr{Q}} \mathbf{d}\boldsymbol{\omega} = \int_{\partial\mathscr{Q}} \boldsymbol{\omega} \tag{4}$$

Let $\{\boldsymbol{\alpha}_i\}_{i=1}^n$ and $\{\boldsymbol{\beta}_i\}_{i=1}^n$ be n-tuples of differential forms. To facilitate our treatment of systems, we extend the exterior derivative, exterior product, Hodge star and magnitude operators to act on

DOI: 10.1002/fld

n-tuples of forms as follows:

$$\mathbf{d}\begin{Bmatrix}\boldsymbol{\alpha}_1\\ \vdots\\ \boldsymbol{\alpha}_n\end{Bmatrix}=\begin{Bmatrix}\mathbf{d}\boldsymbol{\alpha}_1\\ \vdots\\ \mathbf{d}\boldsymbol{\alpha}_n\end{Bmatrix},\qquad \begin{Bmatrix}\boldsymbol{\alpha}_1\\ \vdots\\ \boldsymbol{\alpha}_n\end{Bmatrix}\wedge\begin{Bmatrix}\boldsymbol{\beta}_1\\ \vdots\\ \boldsymbol{\beta}_n\end{Bmatrix}=\begin{Bmatrix}\boldsymbol{\alpha}_1\wedge\boldsymbol{\beta}_1\\ \vdots\\ \boldsymbol{\alpha}_n\wedge\boldsymbol{\beta}_n\end{Bmatrix}$$
$$\star\begin{Bmatrix}\boldsymbol{\alpha}_1\\ \vdots\\ \boldsymbol{\alpha}_n\end{Bmatrix}=\begin{Bmatrix}\star\boldsymbol{\alpha}_1\\ \vdots\\ \star\boldsymbol{\alpha}_n\end{Bmatrix},\qquad \left|\begin{Bmatrix}\boldsymbol{\alpha}_1\\ \vdots\\ \boldsymbol{\alpha}_n\end{Bmatrix}\right|=\begin{Bmatrix}|\boldsymbol{\alpha}_1|\\ \vdots\\ |\boldsymbol{\alpha}_n|\end{Bmatrix} \tag{5}$$

2.2. Governing equations and initial/boundary conditions

This section presents the governing equations for a generic system of conservation laws with source terms using differential forms notation; we use the n-tuple notation to extend the formulation for scalar conservation laws in [16] to systems. Since the hyperbolic system supports discontinuous solutions, we address both the diffuse and jump parts of the localized equations.

Let $\mathscr{D}\subset\mathscr{M}$ be the $(d+1)$-dimensional spacetime analysis domain, where $\mathscr{D}$ has a suitably regular boundary $\partial\mathscr{D}$. We introduce n-tuples of *conservation fields*, *spacetime fluxes* and *source terms*

$$\mathbf{u}:=\begin{Bmatrix}\mathbf{u}_1\\ \vdots\\ \mathbf{u}_n\end{Bmatrix},\quad \mathbf{F}:=\begin{Bmatrix}\mathbf{F}_1\\ \vdots\\ \mathbf{F}_n\end{Bmatrix},\quad \mathbf{S}:=\begin{Bmatrix}\mathbf{S}_1\\ \vdots\\ \mathbf{S}_n\end{Bmatrix} \tag{6}$$

in which the component $\mathbf{u}_\gamma$ is a tensor field on $\mathscr{D}$ of order $m_\gamma\in\mathbb{N}$, i.e. a scalar, vector or tensor field. The spacetime flux components $\mathbf{F}_\gamma$ are d-forms with tensor coefficients of order m_γ whose restrictions to any spacetime d-manifold embedded in $\mathscr{D}$ deliver the fluxes of the associated conservation quantities (i.e. $\mathbf{u}_\gamma$) across the d-manifold; the source terms $\mathbf{S}_\gamma$ are $(d+1)$-forms given by $\mathbf{S}_\gamma=\mathbf{S}_\gamma\boldsymbol{\Omega}$ in which $\mathbf{S}_\gamma$ are tensor fields of order m_γ. The integral form of the system of conservation laws is then written as

$$\int_{\partial\mathscr{Q}}\mathbf{F}+\int_{\mathscr{Q}}\mathbf{S}=\mathbf{0}\quad\forall\mathscr{Q}\subset\mathscr{D} \tag{7}$$

in which only open subdomains $\mathscr{Q}$ with suitably regular boundaries are considered.

We apply Stokes' Theorem and the Localization Theorem to obtain the system of governing equations on each $\mathscr{Q}\subset\mathscr{D}$:

$$(\mathbf{dF}+\mathbf{S})|_{\mathscr{Q}}=\mathbf{0} \tag{8a}$$

$$(\mathbf{F}^*-\mathbf{F})|_{\partial\mathscr{Q}}=\mathbf{0} \tag{8b}$$

in which $\mathbf{F}^*$ is an n-tuple containing *target fluxes* $\mathbf{F}^*_\gamma$ on $\partial\mathscr{Q}$, and $\mathbf{F}|_{\partial\mathscr{Q}}$ (without decoration) denotes the interior trace of $\mathbf{F}$ on $\partial\mathscr{Q}$. Depending on the location and local orientation of $\partial\mathscr{Q}$, the target fluxes $\mathbf{F}^*$ are computed from initial and boundary data on $\partial\mathscr{D}$ or as Godunov values based on the interior and exterior traces of $\mathbf{u}$ on ∂Q; see [16] and the appendices therein for details. The term

DOI: 10.1002/fld

$\mathbf{dF}$ in the governing equations (8a) represents the diffuse part of the exterior derivative operator, while the Rankine–Hugoniot conditions in (8b) are the jump part that governs the behavior of shocks. The use of Godunov fluxes in the jump conditions maintains the correct characteristic structure across inter-element boundaries, reduces numerical dissipation relative to alternative flux definitions and guarantees (for scalar conservation laws) satisfaction of the entropy condition in the weak formulation [16]. Nonetheless, it is sometimes expedient to substitute less costly numerical fluxes for the Godunov values.

Each conservation field $\mathbf{u}_\gamma$ has an associated *spatial flux* $\mathbf{f}_\gamma$, a tensor field of order $m_\gamma+1$ on $\mathscr{D}$. In general, each spatial flux component at any point $P\in\mathscr{D}$ is expressed as a function of the full n-tuple of conservation fields evaluated at P: $\mathbf{f}_\gamma|_P=\hat{\mathbf{f}}_\gamma(\mathbf{u}|_P)$. We construct the *spacetime flux* for $\mathbf{u}_\gamma$, denoted by $\mathbf{F}_\gamma$, as the d-form:

$$\mathbf{F}_\gamma := \mathbf{u}_\gamma \star \mathrm{d}t + \mathbf{f}_\gamma \wedge \star \mathrm{d}x \tag{9}$$

Note that $\mathbf{F}_\gamma|_P=\hat{\mathbf{F}}_\gamma(\mathbf{u}|_P)$. Recalling (2) and (3), we identify the components of (8a) with the residual of the governing conservation equations:

$$\mathbf{dF}_\gamma+\mathbf{S}_\gamma=(\dot{\mathbf{u}}_\gamma+\nabla\cdot\mathbf{f}_\gamma+\mathbf{S}_\gamma)\boldsymbol{\Omega} \tag{10}$$

To exemplify the general formulation, we present two specific instances of conservation laws that we use for numerical examples later in this work. In both cases, we have $\mathbf{S}=\mathbf{0}$.

2.2.1. Inviscid Burgers equation. The inviscid Burgers equation is usually interpreted as a single scalar conservation law in one spatial dimension. Accordingly, we set $d=n=1$. Let u denote the scalar conservation field on $\mathscr{D}$. We set $\mathbf{u}_1=u$ and $m_1=0$. The spatial flux of u is

$$\hat{\mathbf{f}}_1=\tfrac{1}{2}u^2 \tag{11}$$

2.2.2. Euler equations. The Euler equations describe inviscid, compressible fluid flow in up to three spatial dimensions, so we set $d=1$, 2 or 3according to the problem at hand. We have three conservation fields ($n=3$), given by

$$\mathbf{u}=\begin{Bmatrix}\mathbf{u}_1\\ \mathbf{u}_2\\ \mathbf{u}_3\end{Bmatrix}=\begin{Bmatrix}\rho\\ \mathbf{m}\\ \mathscr{E}\end{Bmatrix} \tag{12}$$

in which ρ is the scalar ($m_1=0$) *mass density field*, $\mathbf{m}=\rho\mathbf{v}$ is the vector ($m_2=1$) *linear momentum density field* and $\mathscr{E}=\rho e+(1/2\rho)\mathbf{m}\cdot\mathbf{m}$ is the scalar ($m_3=0$) *total energy density field*; here $\mathbf{v}$ is the *velocity field* and e is the fluid's *internal energy density field*.

The spatial flux functions are given by

$$\hat{\mathbf{f}}_1=\hat{\mathbf{f}}_\rho:=\mathbf{m} \tag{13a}$$

$$\hat{\mathbf{f}}_2=\hat{\mathbf{f}}_{\mathbf{m}}:=\frac{1}{\rho}\mathbf{m}\otimes\mathbf{m}-\boldsymbol{\sigma} \tag{13b}$$

$$\hat{\mathbf{f}}_3=\hat{\mathbf{f}}_{\mathscr{E}}:=\frac{1}{\rho}[\mathscr{E}\mathbf{m}+\boldsymbol{\sigma}(\mathbf{m})] \tag{13c}$$

DOI: 10.1002/fld

in which the stress tensor $\boldsymbol{\sigma}$ for inviscid fluids is given by $\boldsymbol{\sigma}=-\tilde{p}\mathbf{I}$. For fluids governed by the ideal gas law, the fluid pressure $\tilde{p}$ is

$$\tilde{p}=\frac{\rho e}{\gamma-1}=\frac{1}{\gamma-1}\left(\mathscr{E}-\frac{1}{2\rho}\mathbf{m}\cdot\mathbf{m}\right) \tag{14}$$

in which γ denotes the adiabatic index.

2.3. SDG method

We construct a spacetime mesh as a disjoint partition $\mathscr{P}_h$ of the analysis domain $\mathscr{D}$ into N open *spacetime finite elements*, $\mathscr{Q}_\alpha$, such that $\mathscr{P}_h=\{\mathscr{Q}_\alpha\}_{\alpha=1}^{N}$. We use the mesh $\mathscr{P}_h$ to construct a discrete, discontinuous Galerkin, n-tuple solution space, $\mathscr{V}_h:=\{\mathbf{w}:\mathbf{w}_\gamma|_{\mathscr{Q}}\in\mathscr{P}^{\gamma}_{\mathscr{Q}};\ \forall\mathscr{Q}\in\mathscr{P}_h, i=1,\ldots,d, \gamma=1,\ldots,n\}$, where $\mathscr{P}^{\gamma}_{\mathscr{Q}}$ is the space of tensor fields of order m_γ on $\mathscr{Q}$ whose components are polynomial functions on $\mathscr{Q}$ of a given order, $p^{\mathscr{Q}}_\gamma\in\mathbb{N}$. In the numerical examples reported here, we find it convenient to use a uniform polynomial order for the components of the n-tuple ($p^{\mathscr{Q}}_\gamma=p^{\mathscr{Q}};\gamma=1,\ldots,n$) and the same polynomial order in all elements ($p^{\mathscr{Q}}=p\ \forall\mathscr{Q}\in\mathscr{P}_h$). However, either or both of these constraints could be relaxed, as in an hp-adaptive procedure. We use polynomial basis functions with support on individual elements to construct $\mathscr{V}_h$ and write $\mathscr{V}^{\mathscr{Q}}_h=\mathscr{V}_h|_{\mathscr{Q}}$ for each element $\mathscr{Q}\in\mathscr{P}_h$.

The discrete weighted residuals statement follows from the Bubnov-Galerkin version of the Method of Weighted Residuals as applied to (8).

2.3.1. Discrete weighted residuals statement. For each $\mathscr{Q}\in\mathscr{P}_h$, find $\mathbf{u}\in\mathscr{V}^{\mathscr{Q}}_h$ such that

$$\int_{\mathscr{Q}}\mathbf{w}\wedge(\mathbf{dF}+\mathbf{S})+\int_{\partial\mathscr{Q}}\mathbf{w}\wedge(\mathbf{F}^*-\mathbf{F})=\mathbf{0}\quad\forall\mathbf{w}\in\mathscr{V}^{\mathscr{Q}}_h \tag{15}$$

We apply Stokes' Theorem (4)–(15) to obtain the discrete, weak form of the problem.

2.3.2. Discrete weak form. For each $\mathscr{Q}\in\mathscr{P}_h$, find $\mathbf{u}\in\mathscr{V}^{\mathscr{Q}}_h$ such that

$$\int_{\mathscr{Q}}(-\mathbf{dw}\wedge\mathbf{F}+\mathbf{w}\wedge\mathbf{S})+\int_{\partial\mathscr{Q}}\mathbf{w}\wedge\mathbf{F}^*=\mathbf{0}\quad\forall\mathbf{w}\in\mathscr{V}^{\mathscr{Q}}_h \tag{16}$$

It is easy to show that restricting $\mathbf{w}$ to element-wise constant functions recovers exactly the integral form of the element-wise conservation law (7), written with respect to the mathematically and physically consistent fluxes, $\mathbf{F}=\mathbf{F}^*$ [16].

3. SDG SHOCK CAPTURING WITH ADAPTIVE SPACETIME MESHING

3.1. Stabilization

The basic SDG formulation (16) is stable when applied to systems of linear conservation laws and to scalar nonlinear conservation laws with purely convex or purely concave fluxes [16]. However, SDG solutions without stabilization exhibit overshoot and undershoot in the vicinity of discontinuous

DOI: 10.1002/fld

solution features, and this may cause the method to fail, even for these relatively benign problems, when solution fields take on nonphysical values. For general nonlinear conservation laws, including those with mixed convex–concave flux functions, the basic SDG method requires stabilization for polynomial orders greater than zero. In this section, we present a stabilization scheme, an adaptation of the sub-cell method of Persson and Peraire [19] to our SDG setting, which controls overshoot and undershoot while rendering the SDG method stable for general systems of nonlinear conservation laws. We add artificial diffusion to the governing equation (8) in those elements where nonsmooth response is detected by a special discontinuity sensor, as described below. The specific form of the diffusion operator is chosen to preserve the element-wise conservation property of the basic SDG formulation. For sufficiently high-order polynomials, the stabilized SDG method is able to resolve shocks within a single spacetime element. We also describe the use of h-adaptive spacetime meshing in resolving discontinuous solution features.

Let $\mathbf{p}$ denote an n-tuple of d-forms defined as exterior products of the spatial gradients of the components of $\mathbf{u}$ with $\star\mathbf{dx}$:

$$\mathbf{p} := \begin{Bmatrix} \nabla \mathbf{u}_1 \wedge \star \mathbf{dx} \\ \vdots \\ \nabla \mathbf{u}_n \wedge \star \mathbf{dx} \end{Bmatrix} \tag{17}$$

We augment (8) with an artificial diffusion term to obtain

$$[\mathbf{d}(\mathbf{F} - \alpha^{\mathcal{Q}}\mathbf{p}) + \mathbf{S}]|_{\mathcal{Q}} = \mathbf{0} \tag{18a}$$

$$[\mathbf{F}^* - \mathbf{F} - \alpha^{\mathcal{Q}}(\mathbf{p}^* - \mathbf{p})]|_{\partial \mathcal{Q}} = \mathbf{0} \tag{18b}$$

in which $\alpha^{\mathcal{Q}}$ is a uniform diffusivity for element $\mathcal{Q}$, $-\alpha^{\mathcal{Q}}\mathbf{p}$ represents the flux due to artificial diffusion and $\mathbf{p}^*$ is the target value for $\mathbf{p}$. Replacing the residuals in (15) with those given in (18), and applying Stokes' Theorem, we obtain the stabilized weak problem:

For each $\mathcal{Q} \in \mathscr{P}_h$, find $\mathbf{u} \in \mathscr{V}_h^{\mathcal{Q}}$ such that

$$\int_{\mathcal{Q}} [-\mathbf{dw} \wedge (\mathbf{F} - \alpha^{\mathcal{Q}}\mathbf{p}) + \mathbf{w} \wedge \mathbf{S}] + \int_{\partial \mathcal{Q}} \mathbf{w} \wedge (\mathbf{F}^* - \alpha^{\mathcal{Q}}\mathbf{p}^*) = \mathbf{0} \quad \forall \mathbf{w} \in \mathscr{V}_h^{\mathcal{Q}} \tag{19}$$

The choice of $\mathbf{p}^*$ determines certain key properties of the method. It is easy to show that setting $\mathbf{p}^* = \mathbf{0}$ preserves the element-wise conservation property with respect to the original Godunov fluxes, $\mathbf{F}^*$, and we enforce this choice from here on. This simple specification maintains the integrity of the patch-by-patch solution procedure described in Section 3.3, as it effectively restricts the artificial diffusion to the element interiors and thereby avoids artificial parabolic coupling between elements. The choice for $\mathbf{p}^*$ suggested in [19] destroys the purely hyperbolic structure of the system and would therefore break our causal solution procedure. It also modifies the element-wise conservation property to include artificial diffusive fluxes that do not reflect the original system. Thus, $\mathbf{p}^* = \mathbf{0}$ is preferred in the present SDG method.

3.2. Discontinuity indicator

The weak form (19) with $\mathbf{p}^* = \mathbf{0}$ defines the stabilized SDG method, but it remains to define a shock-capturing scheme for sensing discontinuities and adjusting the strength of the artificial

DOI: 10.1002/fld

diffusion. Although the intensity of the artificial diffusion can be adjusted on a per-element basis, our numerical experience indicates that it is best to adjust the diffusion on a per-patch basis, as described below.

Let $\{u_{\gamma j}\}_{j=1}^{d(m_\gamma)}$ be the set of scalar components of the conservation field $\mathbf{u}_\gamma$. We expand the components $u_{\gamma j}$ in each element $\mathcal{Q}$ using a hierarchical polynomial basis. For polynomial degree p, we have $N(p)$ basis functions $\{\psi_i\}_{i=1}^{N(p)}$, and the solution on $\mathcal{Q}$ is expressed as

$$u_{\gamma j} = \sum_{i=1}^{N(p)} u_{\gamma j}^i \psi_i \tag{20}$$

A truncation of the solution on $\mathcal{Q}$ to order $(p-1)$ in the hierarchical basis is given by

$$\hat{u}_{\gamma j} = \sum_{i=1}^{N(p-1)} u_{\gamma j}^i \psi_i \tag{21}$$

We construct $\mathbf{u}_\gamma$ and its truncation $\hat{\mathbf{u}}_\gamma$ using the scalar component fields defined in (20) and (21).

We introduce a discontinuity sensor for $\mathbf{u}_\gamma$ in element $\mathcal{Q}$ that measures the level of participation of the highest-order basis functions in the solution:

$$\Sigma_\gamma^{\mathcal{Q}} := \frac{\|\mathbf{u}_\gamma - \hat{\mathbf{u}}_\gamma\|_{L_2(\mathcal{Q})}^2}{\|\mathbf{u}_\gamma\|_{L_2(\mathcal{Q})}^2} \tag{22}$$

No stabilization is required for the overly diffusive piece-wise constant case, so we skip the computation in (22) and set $\Sigma_\gamma^{\mathcal{Q}} = 0$ when $p = 0$. To avoid a possible division by zero in (22) when $p > 0$, we perform a check for near-zero solutions before evaluating $\Sigma_\gamma^{\mathcal{Q}}$. If

$$\frac{\|\mathbf{u}_\gamma\|_{L_2(\mathcal{Q})}}{u_\gamma^0 |\mathcal{Q}|} < \varepsilon \tag{23}$$

we set $\Sigma_\gamma^{\mathcal{Q}} = 0$ and skip the computation in (22). Here, u_γ^0 is a strictly positive characteristic magnitude for $\mathbf{u}_\gamma$ that is computed using the problem data, and $\varepsilon \ll 1$ is a user-specified tolerance that can be selected as some multiple of the machine precision.

We expect $\Sigma_\gamma^{\mathcal{Q}}$ to decay with increasing polynomial order according to $1/(p+1)^4$ when the solution is smooth near element $\mathcal{Q}$ [19]; a slower rate of decay indicates the presence of a nonsmooth solution feature. Accordingly, we propose the following shock-capturing scheme for adjusting the strength of the artificial diffusion. Let $s_\gamma^{\mathcal{Q}} := \log(\Sigma_\gamma^{\mathcal{Q}})$. The intensity of artificial diffusion required by conservation field $\mathbf{u}_\gamma$ in element $\mathcal{Q}$ is

$$\alpha_\gamma^{\mathcal{Q}} = \begin{cases} 0 & \text{if } s_\gamma^{\mathcal{Q}} < \tilde{s} - \beta \\ \dfrac{\tilde{\alpha}}{2}\left[1 + \dfrac{s_\gamma^{\mathcal{Q}} - \tilde{s}}{\beta}\right] & \text{if } \tilde{s} - \beta \leqslant s_\gamma^{\mathcal{Q}} \leqslant \tilde{s} + \beta \\ \tilde{\alpha} & \text{if } s_\gamma^{\mathcal{Q}} > \tilde{s} + \beta \end{cases} \tag{24}$$

in which $\tilde{\alpha} = C_\alpha c_0 h/p$ and $\tilde{s} = \log[C_s/(p+1)^4]$; here h denotes the spatial diameter of element $\mathcal{Q}$, c_0 is the maximum wave speed in the analysis domain, while C_α, C_s and β are user-selected

DOI: 10.1002/fld

parameters. The effect of (24) is to eliminate artificial diffusion in elements where the sensor decays somewhat faster than the expected rate, with a linear ramp to the maximum-strength artificial diffusion in elements where the indicator decays more slowly than expected.

In practice, we find it best to use the same diffusion value for all fields and for all elements within each patch. Thus, we determine the element diffusions in patch $\mathbb{P}$ according to

$$\alpha^{\mathcal{Q}} = \alpha^{\mathbb{P}} := \max_{\bar{\mathcal{Q}} \in \mathbb{P}} \left(\max_{\gamma=1,\ldots,n} \alpha_{\gamma}^{\bar{\mathcal{Q}}} \right) \quad \forall \mathcal{Q} \in \mathbb{P} \tag{25}$$

The discontinuity sensor in (22) is unreliable when applied to very low-order polynomial bases. We therefore recommend that a minimum polynomial order be imposed for this method, say $p_{\min} = 2$. Alternatively, one could use the hierarchical basis to approximate the discrete solution in the next higher-order polynomial space, $p+1$, and modify the discontinuity sensor to measure the relative participation of the basis functions of polynomial order $p+1$. This approach, although untested, should support any polynomial order $p \geqslant 0$.

The form of $\tilde{\alpha}$ limits the convergence rate in regions where shock capturing is active to $\mathcal{O}(h/p)$. Although increasing p delivers some error reduction, there are diminishing returns as p increases. Meanwhile, the computational complexity of a patch solution grows rapidly and nonlinearly as p increases. Thus, the improvement in accuracy delivered by higher-order elements in which shock capturing is active generally does not justify their greater computational expense.

We use the following heuristic to solve a patch of elements in the shock-capturing scheme:

1. Generate an estimate of the solution in each element using available inflow data, and carry out one iteration of the nonlinear patch solution procedure with $\alpha^{\mathcal{Q}} = 0$ in all elements.
2. Use (22) and (24) to compute the sensors and the artificial diffusion for each component of the conservation fields in each element of the patch, subject to the special conditions for near-zero solutions and for $p = 0$. Compute the patch-wise maximum diffusion $\alpha^{\mathbb{P}}$ according to (25).
3. If $\alpha^{\mathbb{P}} = 0$, continue the nonlinear iterations to solve the patch equations with zero diffusion. Otherwise, discard the solution from step 1, reset the initial estimate of the solution, assign the diffusivity $\alpha^{\mathbb{P}}$ to all the elements in the patch according to (25) and solve the nonlinear patch equations while holding the artificial diffusion fixed.
4. Store the converged solution, and proceed to the next patch.

We found no advantage in treating the artificial diffusion as a nonlinear operator to be recomputed after each iteration in the nonlinear solution process. This would, at best, increase the solution cost without improving the solution. At worst, the nonlinear character of the shock-capturing operator could lead to nonconvergence in certain iterative solution schemes. We therefore use the above heuristic in our implementation. This approach allows us to specify a simple linear ramp for the middle branch of (24), as opposed to the smooth function proposed in [19] that better accommodates many nonlinear solution schemes.

3.3. Adaptive spacetime meshing

An adaptive solution procedure can significantly improve the accuracy of shock-capturing methods; h-adaptive meshing can generate highly refined meshes in the vicinity of shocks and other discontinuous solution features, while permitting coarser meshes in regions where the solution is smooth. In this subsection, we briefly review our advancing-front spacetime meshing procedure, an interleaved

DOI: 10.1002/fld

patch-by-patch solution scheme and extensions of the basic algorithm for h-adaptive analysis. Although our discussion focuses on meshing domains in $2D \times$ time, we use the same concepts to address domains in $1D \times$ time. In principle, the techniques we describe are extensible to domains in $3D \times$ time, but we have not yet implemented this capability. A more detailed treatment of causal meshing can be found in [20, 21, 24].

We use an advancing-front mesh-generation procedure called *Tent Pitcher* [20] to construct simplicial spacetime meshes that satisfy a patch-wise *causality constraint*. A 'patch' is a collection of tetrahedral elements whose boundary is a collection of oriented, triangular *facets* in spacetime. The causality constraint requires that all facets are causal, except for those that lie on $\partial\mathscr{D}^{\mathrm{nc}}$; any patch that satisfies this requirement is called a *causal patch*. Inter-element boundaries within a patch are permitted to be noncausal, so the elements within a patch must be solved simultaneously. However, the solution within a patch depends only on data from earlier patches and on prescribed initial and boundary data. This asymmetric dependency implies a partial ordering of patches that enables an $\mathcal{O}(N)$ patch-by-patch solution procedure. Tent Pitcher generates a sequence of causal patches that respects this partial ordering.

Tent Pitcher begins with a triangulation of the spatial domain that can be generated by any available simplicial mesh generator We append a temporal coordinate to the spatial coordinates for each node in the mesh; to begin, all time coordinates are set to the starting time of the analysis interval. We refer to this mesh as the *space mesh*. However, as the solution evolves, distinct time coordinates become associated with different nodes in the mesh. Thus, the space mesh generally describes a spacetime *terrain* on which time is a nonuniform function of spatial position. The space mesh controls the construction of a *spacetime mesh*, composed of $(d+1)$-simplices, as described below.

The basic tent-pitching operation begins with the selection of a node in the space mesh whose time coordinate is a local minimum with respect to its neighbors; Tent Pitcher advances the selected node in time to create a new node in the spacetime mesh. The spacetime segment connecting the node at its previous and new times is called a *tent pole*. Tent Pitcher constructs a new patch (or *tent*) of spacetime elements as the set of tetrahedra that connect the top and bottom of the tent pole to the neighboring nodes in the space mesh. Updating the time coordinate of the selected space-mesh node to the temporal value at the top of the tent pole completes the tent-pitching operation.

Two criteria control the temporal extent (duration) of the tent pole: the outflow facets of the new patch must all be causal, and a *progress constraint* must be met to ensure that Tent Pitcher is always able to continue creating new patches. The causality constraint imposes a local condition on the tent-pole duration that is similar to a Courant–Friedrichs–Lewy condition. However, there is no global time-step restriction as in most explicit time-stepping methods and the causality constraint derives from the requirements of our patch-by-patch solution procedure, rather than from a stability condition. Tent Pitcher repeats this basic tent-pitching operation to generate a fully unstructured spacetime mesh as a sequence of causal spacetime patches while advancing the space-mesh nodes in time. The solution procedure terminates when all nodes in the space mesh have reached or exceeded the final analysis time.

Our implementation interleaves the mesh generation and solution procedures; we immediately compute the patch-wise SDG solution every time Tent Pitcher creates a new patch of spacetime elements. Since the number of elements per patch is governed by the highest degree vertex in the spatial triangulation, and because the highest polynomial order in each element is bounded, we can show that our patch-by-patch solution procedure has $\mathcal{O}(N)$ complexity, where N is the number of spacetime patches (elements).

DOI: 10.1002/fld

The goal of our h-adaptive scheme is to divide nearly evenly the finite element error between all of the spacetime elements. We use a residual-based, *a posteriori* error indicator derived from (16), and define an n-tuple of error indicators for the discrete solution on element $\mathcal{Q}$ as sums of L_1 norms of the volume and boundary residuals:

$$\boldsymbol{\varepsilon}^{\mathcal{Q}} = \begin{Bmatrix} \varepsilon_1^{\mathcal{Q}} \\ \vdots \\ \varepsilon_n^{\mathcal{Q}} \end{Bmatrix} := \int_{\mathcal{Q}} |\mathbf{dF} + \mathbf{S}| + \int_{\partial \mathcal{Q}} |\mathbf{F}^* - \mathbf{F}| \tag{26}$$

Based on the definitions of the target fluxes given in [16], the outflow faces do not contribute to the error and can be dropped from the boundary integral in (26). Every time a spacetime patch is solved, we compute the n-tuple of error indicators $\boldsymbol{\varepsilon}^{\mathcal{Q}}$ for each element $\mathcal{Q}$ in the patch.

The error indicator (26) drives an extended, adaptive version of the Tent Pitcher meshing algorithm. User-specified parameters $\{\underline{\varepsilon}_\gamma, \overline{\varepsilon}_\gamma\}_{\gamma=1,n}$ define a target range for the element-wise error, $\underline{\varepsilon}_\gamma \leqslant \varepsilon_\gamma^{\mathcal{Q}} \leqslant \overline{\varepsilon}_\gamma$. Before accepting a new patch solution, we perform checks on each spacetime element $\mathcal{Q}$ in the patch to ensure that the upper inequality is strictly enforced, while taking steps to loosely enforce the lower inequality. If $\varepsilon_\gamma^{\mathcal{Q}} > \overline{\varepsilon}_\gamma$ for any value of γ, we flag element $\mathcal{Q}$ as needing refinement; if $\varepsilon_\gamma^{\mathcal{Q}} < \underline{\varepsilon}_\gamma$ for all values of γ, we flag element $\mathcal{Q}$ as coarsenable; otherwise, we flag $\mathcal{Q}$ as needing neither refinement nor coarsening. We copy these spacetime element flags to the corresponding elements in the space mesh.

If any spacetime element in the patch is flagged as needing refinement, then the allowable error has been exceeded and we reject the entire patch. That is, we discard the patch solution, delete all the spacetime elements in the patch and immediately subdivide the corresponding elements in the space mesh using a newest-vertex-bisection algorithm [20]. This spatial refinement induces a simultaneous temporal refinement in the subsequent spacetime mesh since the causality constraint limits the duration of spacetime elements pitched over the subdivided space elements. Although the spacetime mesh produced by refinement is nonconforming, our discontinuous Galerkin formulation accommodates this situation without modification and does not require a projection of the solution onto the refined mesh.

Coarsening operations are postponed until all the elements surrounding a node in the space mesh are flagged as coarsenable. Before pitching a new tent over a node of degree n in the space mesh, Tent Pitcher checks the status of the space elements surrounding the node. If any of the surrounding elements are not coarsenable, then a normal tent-pitching operation is executed without coarsening. If all the surrounding elements are coarsenable, Tent Pitcher constructs a special spacetime patch that effectively deletes the node from the space mesh, as illustrated in Figure 1(a). The inflow facets of coarsening patches match the outflow facets of previously solved patches, so the spacetime mesh is locally conforming and no solution projection is required. Thus, our spacetime coarsening method eliminates the projection errors associated with coarsening operations in most conventional remeshing procedures.

Maintenance of element quality is an important consideration in any adaptive solution procedure; edge-flip and node-smoothing operations are commonly used to address this requirement in spatial discretizations. Each of these operations requires a solution projection with associated projection errors that limit the convergence rates of conventional adaptive remeshing procedures to $\mathcal{O}(h)$, independent of polynomial order. Similarly to our approach to coarsening, we implement edge-flip

DOI: 10.1002/fld

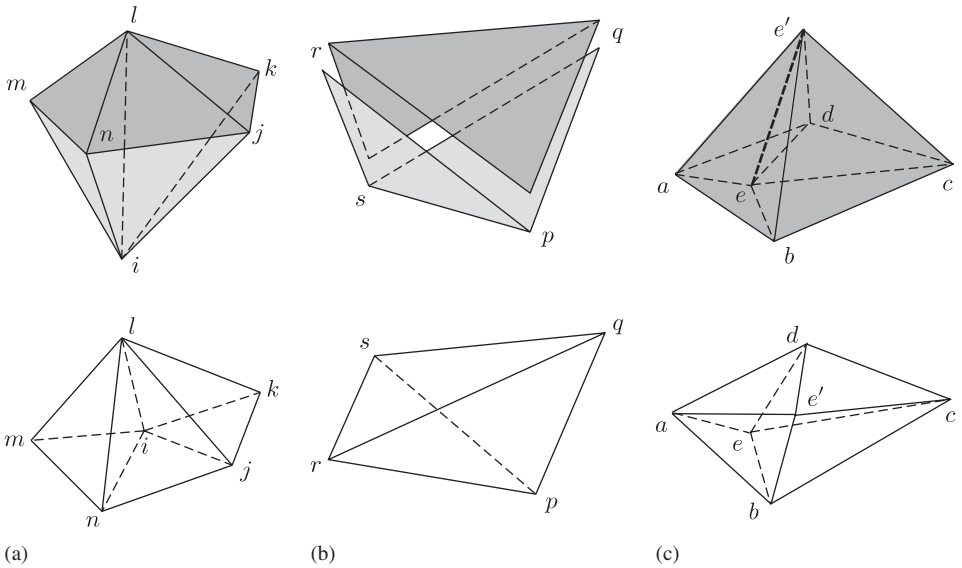

Figure 1. Adaptive operations executed as spacetime patches (above) and as instantaneous remeshing operations in space only (below): (a) vertex deletion (degree 5); (b) edge flip; and (c) inclined tent pole.

and node-repositioning operations as special spacetime patches that generate locally conforming spacetime meshes, as depicted in Figures 1(b) and (c). This approach eliminates the need for solution projections, so our adaptive solution procedure preserves the full convergence rates of higher-order elements.

Conventional adaptive remeshing operations are expensive and require a global synchronization that can degrade the efficiency of parallel solution schemes. If the solution error is too large within a time step, expensive global solution data must be discarded in preparation for refinement. For these reasons, adaptive remeshing operations are usually scheduled as infrequent events. All adaptive operations are local in our spacetime implementation, and only local data are discarded when a refinement operation is required. This keeps reanalysis costs low. Our adaptive remeshing procedure is asynchronous, so it has relatively small impact on the efficiency of parallel solvers. Coarsening operations are also executed as soon as possible to further improve the efficiency of our solver.

The local character of our remeshing operations supports continuous remeshing to improve element quality and to coarsen as soon as possible. Before pitching a new tent, Tent Pitcher identifies and executes all admissible edge-flip operations that would improve the quality of the space mesh. When pitching over any node whose spatial position is unconstrained, we use an area-weighted average of surrounding element centroid positions to determine the spatial position of the top of an inclined tent pole. Thus, we continually smooth the spatial node positions throughout the generation of the spacetime mesh. Improving the quality of the space mesh implicitly enhances the quality of the spacetime elements they generate while increasing allowable patch durations under the causality constraint. Beyond their role in mesh-smoothing operations, inclined tent poles can track material interfaces and the trajectories of discontinuous solution features, as described in Section 4.

DOI: 10.1002/fld

3.4. Numerical studies of SDG shock capturing

3.4.1. Inviscid Burgers equation. We first consider the inviscid Burgers equation, a scalar conservation law whose fluxes are given in Section 2.2.1, on the one-dimensional spatial domain $x \in [0, 1]$ with terminal time $t_f = 0.3333333$ and initial data $u|_{t=0} = \frac{1}{2} + \sin[2\pi(x - \frac{1}{12})]$. The initial data steepens to form a shock at $t = \frac{1}{4}$; thereafter, the shock's trajectory is given by $x_{\text{shock}} = \frac{17}{24} + \frac{1}{2}(t - \frac{1}{4})$. We use a nonadaptive space mesh consisting of $N_x = 20$ elements and impose a uniform duration on the spacetime elements, $\Delta t = 0.0166667$; this generates a spacetime mesh with a total of 1200 elements. Figure 2 shows a time sequence of the solution in the vicinity of the shock obtained with various choices for the polynomial order p and with shock-capturing parameters, $C_\alpha = \frac{1}{4}$, $C_s = 1$ and $\beta = 1$.

At $t = 0.23333$, the shock reaches the common edge between two elements at $x = 0.70$, and is resolved over the adjacent elements without disturbing the solution elsewhere in the domain. The benefits of using higher-order polynomials with the SDG shock-capturing method are evident as the shock moves into an element interior, near the left edge, at $t = 0.25$. For $p = 2$, the shock is still resolved over two elements. However, the disturbance in the left element decreases as p increases and is barely visible for $p = 10$.

The shock is at the center of an element at $t = 0.283333$ and is fairly well resolved for all polynomial orders. For $p = 7, 10$ the shock-capturing method delivers true sub-cell capturing. As the shock approaches the right element edge and eventually coincides with it, the results are similar to the corresponding cases near the left edge. The bottom-right plot, at $t = 0.283333$, shows the solution over the entire spatial domain; the shock-capturing solution is accurate away from the shock for all polynomial orders.

3.4.2. Shu–Osher problem. This and subsequent examples in this paper involve the Euler equations, as described in Section 2.2.2. The present example demonstrates the convergence of the SDG shock-capturing method with h-refinement as well as the ability of the discontinuity indicator described in Section 3.2 to distinguish true jumps from smooth solution features with steep gradients. We solve the one-dimensional shock–turbulence interaction problem introduced in [25], where the

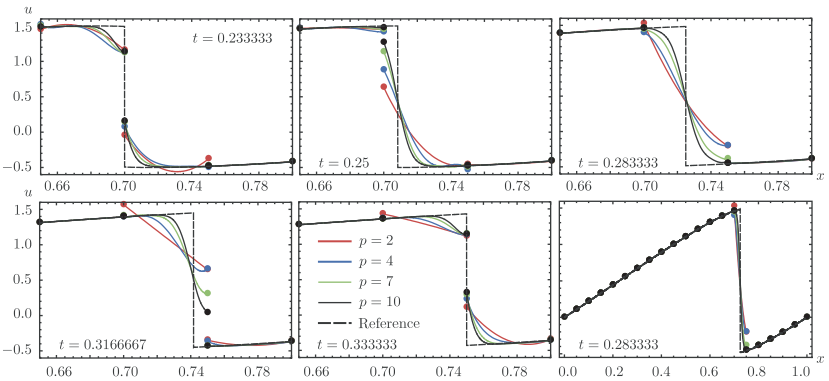

Figure 2. Inviscid Burgers equation: resolution of a shock by the SDG-capturing scheme as the shock traverses a space element from left to right. The first five plots show details of the shock region; the bottom-right plot shows the entire solution domain. Dots indicate element boundaries.

DOI: 10.1002/fld

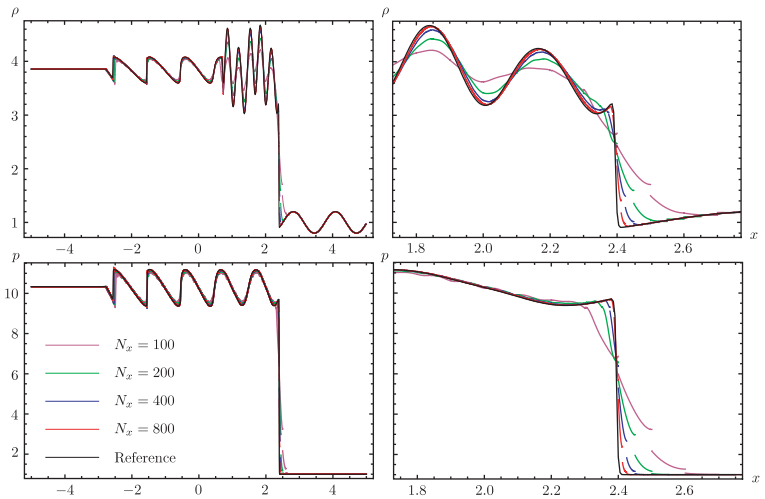

Figure 3. Shu–Osher problem: mass density (top), pressure (bottom) for various mesh sizes; full domain (left), detail of shock region (right).

spatial domain is $x \in [-5, 5]$, the initial data are given by

$$(\rho, v, p) = \begin{cases} (3.857143, 2.629369, 10.333333) & \text{for } x < -4 \\ (1 + 0.2\sin(5x), 0.0, 1.0) & \text{for } x > -4 \end{cases} \tag{27}$$

and the final time of the analysis interval is $t = 1.8$.

We compute SDG solutions for $p = 5$ on space meshes consisting of $N_x = 100, 200, 400$ and 800 elements; these generate spacetime meshes consisting of 21 800; 86 800; 346 800 and 1 386 400 elements, respectively. The shock-capturing parameters are $C_\alpha = 1, C_s = 1$ and $\beta = 2$. Figure 3 shows plots of the density and pressure fields at $t = 1.8$. The reference solution is generated by a fifth-order WENO scheme with 2000 cells across the spatial domain. The plots on the right show details of the solution over the spatial interval $[1.75, 2.75]$, which contains a smooth, high-gradient region to the left connected by a shock to a lower-gradient region to the right. The discontinuity indicator successfully identifies the shock, adding diffusion only to those elements involved in resolving the shock. As the mesh is refined, the solution converges only linearly in h. The optimal convergence rates of $\mathcal{O}(h^{p+1})$ are no longer obtained after use of the diffusion operator. Further, we found that $\mathcal{O}(h)$ errors generated at the shock propagate downstream to pollute the solution even in regions where the solution is smooth. The cause and possible remedies for this undesirable behavior merit further investigation. This problem might be similar in nature to the downstream pollution described in [26, 27].

4. DISCONTINUITY TRACKING IN SPACETIME

4.1. Motivation and method

On first examination, SDG finite element spaces are a more favorable choice than conforming C^0 finite element spaces for discrete approximations of the BV spaces where continuum solutions to

DOI: 10.1002/fld

conservation laws reside. In some respects, this evaluation holds true; the SDG jump conditions provide a natural mechanism for including the Rankine–Hugoniot conditions in the finite element formulation and the discontinuous format of the basis leads directly to element-wise conservation properties that are not available with C^0 bases. One might hope that discontinuous SDG bases would also enable improved resolution of discontinuous solution features, such as shocks and contact discontinuities. However, this prospect is not realized in general. Singular surfaces in continuum BV solutions can arise on any collection of spacetime d-manifolds in $\mathscr{D}$, while they are restricted to inter-element boundaries in SDG approximations. Unless special steps are taken to align the mesh, the trajectories of singular surfaces still traverse the interiors of SDG elements where the discrete solution space is locally continuous. As a result, Galerkin projections into SDG finite element spaces suffer artifacts similar to those obtained with conforming discretizations. Although inter-element jumps can play a beneficial role, SDG projections of discontinuous solution features still exhibit local overshoot and undershoot. While global oscillations are atypical in SDG solutions, pollution from overshoot and undershoot is sometimes observed in the wake of an advancing shock front. Thus, costly mesh refinement combined with stabilization, such as the shock-capturing method described in Section 3, is still required to resolve accurately discontinuous solution features unless the SDG mesh is specially aligned.

In principle, an adaptive SDG discretization could provide an improved discrete approximation space for BV solutions if the mesh is contrived to cover the trajectories of singular surfaces in a tracking solution scheme. The discrete solution space would then be able to represent whatever discontinuities are present as true jumps, thereby eliminating overshoot and undershoot from the Galerkin projection and reducing the need for extra stabilization. The resulting solutions would be far more accurate, and the cost of simulation would be dramatically reduced as there would be no need for intense mesh refinement around discontinuities. The problem of tracking multiple interacting shocks is difficult at best, and is not possible unless the discrete solution space supports singular surfaces with arbitrary orientation in spacetime. Our SDG discretizations satisfy this requirement, and our advancing-front, adaptive solution procedure offers new flexibility for tracking features in spacetime with element edges and faces. This section presents a pilot study of a discontinuity-tracking scheme based on the SDG method for nonlinear conservation laws. Our purpose is to investigate the performance and behavior of SDG approximations when the mesh can be made to track singular surfaces. Therefore, we study relatively simple problems where the solutions contain a limited number of singular surfaces with known trajectories; we emphasize that our existing software and algorithms do not provide a robust tracking capability for more complex problems.

Consider first the problem of tracking in two spatial dimensions, $d=2$. Inclined tent poles, cf. Figure 1(c), are the primary tracking mechanism in our adaptive spacetime meshing procedures. When a tent pole is to be pitched over a vertex that lies on a singular surface, we use local information about the surface's trajectory to prescribe an inclined direction for the tent pole. Given that one or two neighboring vertices also lie on the singular surface, the resulting patch must contain a corresponding number of triangular inter-element boundaries with vertices on the surface's trajectory. If the singular surface is flat, the resulting mesh covers it with element faces. Otherwise, the element faces cover a faceted approximation of the singular surface.

Inclined tent poles tend to generate distorted element shapes, and we sometimes limit their durations to control this effect, However, we also apply adaptive operations, including smoothing of the node positions in the space mesh, edge flips, coarsening and refinement, in the vicinity of inclined tent poles to develop and maintain a well-graded, high-quality mesh. We use a similar

DOI: 10.1002/fld

adaptive meshing scheme to track discontinuities in one spatial dimension, $d=1$. With the exception of one example where multiple singular surfaces originate at the same spacetime location, we do not address situations where a singular surface intersects another singular surface or a domain boundary.

4.2. *Numerical studies of discontinuity tracking*

4.2.1. Sod problem in $1D\times time$. We study a one-dimensional Riemann problem introduced by Sod [28] to demonstrate the advantages of tracking discontinuities. The spatial domain is $[0,1]$, the temporal interval is $[0,0.25]$, and the piece-wise-constant initial conditions are

$$(\rho, v, p)=\begin{cases}(1,0,1), & x<\frac{1}{2}\\ (0.125,0,0.1), & x>\frac{1}{2}\end{cases} \tag{28}$$

The solution exhibits four nonsmooth or discontinuous features (in left to right order): (1) the leading edge of a rarefaction wave traveling to the left, (2) a stationary edge of the rarefaction wave at $x=0.5$, (3) a contact discontinuity and (4) a shock; the latter two features travel to the right. All four features nucleate simultaneously from a single discontinuity at $(x,t)=(0.5,0)$. We use a special initial patch composed of five triangular elements, with a common vertex at the nucleation point and four inter-element boundaries inclined to coincide with the trajectories of the solution features, to track the nucleation event. Once the special patch is solved, we use Tent Pitcher patches with inclined tent poles to track individually the trajectories of the four solution features. We use $p=3$ and employ h-adaptivity with $\underline{\varepsilon}=0.60*10^{-5}$ and $\overline{\varepsilon}=1.40*10^{-5}$, starting from an initial space mesh composed of four elements.

Figure 4 displays the solutions obtained with capturing and tracking. The capturing parameters, $C_\alpha=1$, $C_s=1$ and $\beta=2.5$, are selected by a numerical experiment to ensure that overshoot and undershoot are eliminated while avoiding excessive diffusion. In contrast to the results for the Shu–Osher problem in Section 3.4.2, no downstream pollution is evident in the shock-capturing solution. The solution values in the piece-wise uniform branches match the reference solution to within-machine precision. A similar adjustment for the tracking case reduces these parameters to $C_\alpha=0.025$, $C_s=4$ and $\beta=0.25$. Only the initial patch and the patches along the shock trajectory require stabilization, this suggests that the basic SDG formulation still requires a small amount of stabilization even when tracking is used to cover all the singular surfaces in the solution.

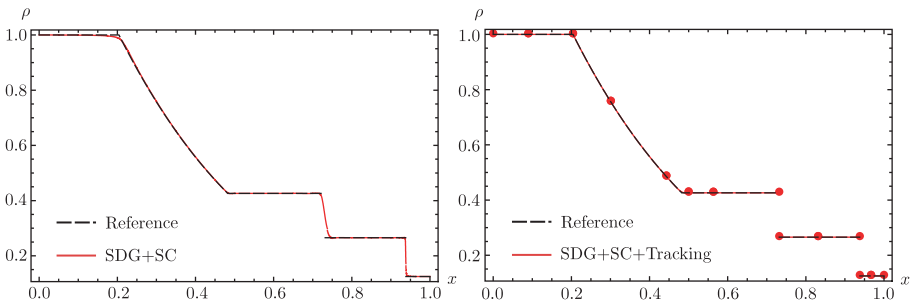

Figure 4. 1D × time Sod problem: capturing (left) and tracking (right) solutions for the mass density field at $t=0.25$. The dots indicate the locations of element boundaries (due to a bug in the plotting software, these are slightly out of position).

DOI: 10.1002/fld

At $t=0.25$, the adaptive tracking solution uses only 11 elements in the space mesh, compared with 38 space elements in the capturing solution. The differences are more dramatic for the number of spacetime elements, an accurate indicator of computational expense for our $\mathcal{O}(N)$ SDG solution procedure. The tracking solution uses a total of 698 spacetime elements, as shown in Figure 5, whereas capturing uses 11 525 elements. Thus, the cost of the tracking solution is only 6% of the cost of the capturing solution. Despite its low cost, the tracking solution is far more accurate, as the shock and the contact discontinuity are resolved perfectly as true jumps. These features, especially the contact discontinuity, are smeared in the capturing solution.

4.2.2. Sod problem in $2D\times time$. We consider a simple extension to 2D × time of the Sod problem presented in Section 4.2.1. The space domain is $[0,1]\times[0,0.10]$, the temporal interval is $[0,0.25]$, and the initial conditions are given by (28). The solution is uniform in the y-direction, and has the same structure in the x-direction as the solution in the previous example. In this example, we track only the contact discontinuity and capture the other solution features. This obviates the need for special patches that track multiple surfaces emanating from a common line; Tent Pitcher patches with inclined tent poles suffice to track the contact trajectory. We use a quadratic basis ($p=2$); the shock-capturing parameters are $C_\alpha=2.5, C_s=1$ and $\beta=2$; and the tolerances for adaptive mesh refinement are $\underline{\varepsilon}=3.0*10^{-7}$ and $\overline{\varepsilon}=7.0*10^{-7}$. We use the Harten–Lax–van Leer-Contact (HLLC) flux [29], a relatively inexpensive numerical flux that accurately models contact discontinuities, to approximate the Godunov fluxes on the noncausal faces within each patch.

Figure 6 shows results from the tracking simulation, which generates a spacetime mesh containing 328 204 tetrahedral elements. Tracking sharply resolves the contact discontinuity as a true jump on a coarse mesh, notably, without the intense mesh refinement that is still required to capture the shock. Moderate refinement suffices to capture the rarefaction wave.

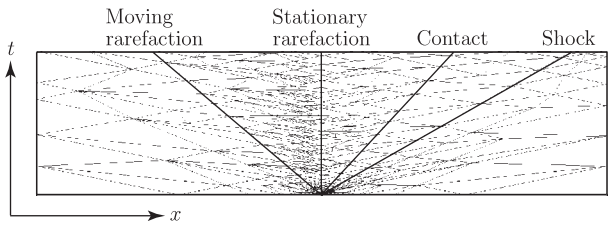

Figure 5. Spacetime mesh generated by h-adaptive tracking solution of the 1D × time Sod problem.

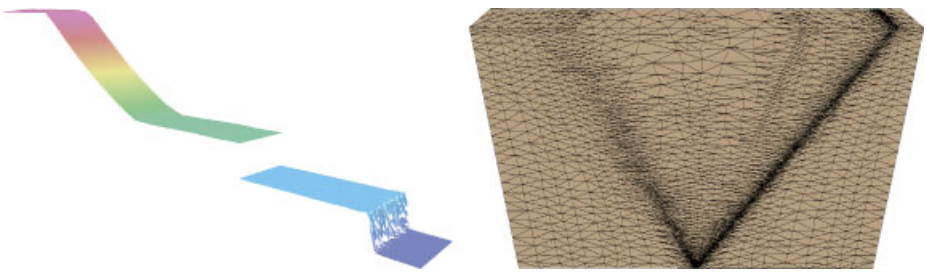

Figure 6. Sod problem in 2D × time: mass density at $t=0.25$ (left), spacetime mesh tracking only the contact discontinuity (right).

DOI: 10.1002/fld

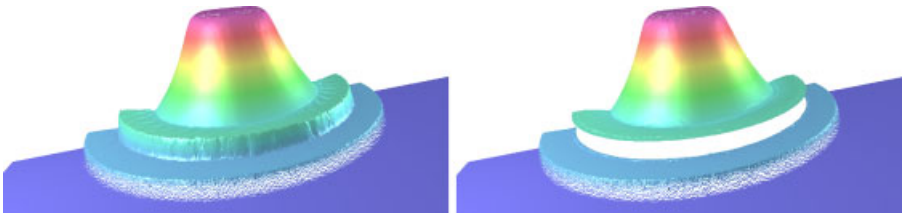

Figure 7. Cylindrical explosion problem: capturing (left), tracking only the contact discontinuity (right). Height and color fields show mass density at $t=1.50$.

4.2.3. Solution-dependent tracking. The preceding example demonstrates that tracking is effective in 2D $\times$ time when the spacetime mesh exactly covers the *a priori* specified trajectory of a singular surface. However, in general, trajectories of solution features are not known *a priori*. In addition, an exact covering with flat facets is generally not possible in higher spatial dimensions where singular surfaces may be curved in space and/or time. We now consider a cylindrical explosion problem [30] to study the effectiveness of tracking when the covering is only approximate and the trajectory is unknown. The spatial domain is $[0, 15]\times[-8, 15]$ with symmetry boundary conditions along the y-axis; the initial data are

$$(\rho, \mathbf{v}, p)=\begin{cases}(1, \mathbf{0}, 1), & x^2+y^2<\pi \\ (0.125, \mathbf{0}, 0.1) & \text{otherwise}\end{cases} \tag{29}$$

The first branch of the initial data holds within a semi-circle of radius $r=\pi$ centered at the origin. We approximate this semi-circle with 90 inscribed linear segments in the initial space mesh and apply the corresponding initial data on the interior of the resulting polygon. As in the previous example, we track only the contact discontinuity via inclined tent poles; this preserves the polygonal approximation of the expanding contact front. To determine the tent-pole directions, we calculate the average normal flow velocity at the bottom of each tent pole (via the solution on the preceding patch) and use this velocity to prescribe the tent-pole inclination. We use HLLC numerical fluxes, and the simulation parameters are $p=2, C_\alpha=5, C_s=0.25, \beta=2, \underline{\varepsilon}=0.6*10^{-5}$ and $\overline{\varepsilon}=1.4*10^{-5}$. For purposes of comparison, we analyze the same model without tracking so that all singular features must be captured.

Figure 7 shows results from the two simulations. The pure-capturing simulation resolves the shock and contact discontinuity reasonably well, while the tracking solution generates a true jump at the contact discontinuity. Minor stabilization is still required at the contact, but the polygonal approximation of the doubly curved singular surface is sufficiently accurate to allow for the use of a relatively coarse mesh.

5. CONCLUSIONS

The proposed shock-capturing scheme provides a robust and effective means for stabilizing SDG finite element models while preserving their element-wise conservation and causal properties; adding h-adaptive refinement substantially improves performance. However, regions of intense mesh refinement along singular surfaces dominate the cost of capturing, and higher-order elements

DOI: 10.1002/fld

in these regions are inefficient due to the combined effects of stabilization, the low regularity of the continuum solution and meshing constraints that restrict adaptive coarsening to ensure a well-graded grid. Thus, an hp-adaptive implementation that reduces the polynomial orders of elements affected by singular surfaces should significantly boost efficiency relative to the present h-adaptive scheme. We found that our shock-capturing scheme sometimes generates downstream pollution error of $\mathcal{O}(h)$. When this pollution is present, there is little point in using high-order elements in the SDG formulation. A stabilization scheme that avoids this problem would, therefore, be of considerable value. Automatic methods for determining the tuning parameters in the SDG shock-capturing method are also desirable.

The SDG-tracking method proved very effective in our limited pilot study; it resolves jumps sharply along inter-element boundaries while excluding them from element interiors. Our tracking solutions preserve the proper structure and velocities of rarefaction waves, shocks and contact discontinuities, including one example where all three wave types emerge at distinct velocities from a single source. Although nonvanishing stabilization is still needed, we found that tracking circumvents Gibbs phenomena and, therefore, substantially reduces the amount of stabilization required. Tracking dramatically reduces solution cost, relative to capturing methods, by eliminating the need for mesh refinement along singular surfaces. The discussion in [26] suggests that shock tracking might also provide a means to mitigate or eliminate the pollution. Overall, this study suggests that SDG discontinuity tracking has the potential to deliver break-through gains in accuracy and efficiency. However, nontrivial challenges in spacetime meshing technology must be met to realize these benefits within a robust implementation.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge Prof. C. W. Shu for graciously providing the reference solution to the Shu–Osher problem, the contributions of Shuo-Heng Chung, Shripad Thite and Yuan Zhou to the meshing and visualization software used in this work as well as support provided by the Center for Process Simulation and Design (CPSD) and the Materials Computation Center (MCC) at the University of Illinois at Urbana-Champaign.

REFERENCES

1. Brooks AN, Hughes TJR. Streamline-upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1982; **32**:199–259.
2. Hughes TJR, Franca LP, Hulbert GM. A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective–diffusive systems. *Computer Methods in Applied Mechanics and Engineering* 1989; **73**:173–189.
3. Liu XD, Osher S, Chan T. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics* 1994; **115**:200–212.
4. Jiang GS, Shu CW. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics* 2000; **126**:202–228.
5. Hughes TJR, Mallet M, Mizukami A. A new finite element formulation for computational fluid dynamics: II. Beyond SUPG. *Computer Methods in Applied Mechanics and Engineering* 1986; **54**:341–355.
6. Hughes TJR, Mallet M. A new finite element formulation for computational fluid dynamics: IV. A discontinuity-capturing operator for multidimensional advective–diffusive systems. *Computer Methods in Applied Mechanics and Engineering* 1986; **58**:329–336.
7. De-Kang M. Entropy satisfaction of a conservative shock-tracking method. *SIAM Journal on Numerical Analysis* 1999; **36**(2):529–550.

DOI: 10.1002/fld

8. LeVeque R, Shyue K. Two-dimensional front tracking based on high resolution wave propagation methods. *Journal of Computational Physics* 1996; **123**:354–368.
9. Chessa J, Belytschko T. A local space–time discontinuous finite element method. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:1325–1343.
10. Babuška I, Melenk J. The partition of unity method. *International Journal for Numerical Methods in Engineering* 1997; **40**(4):727–758.
11. Touil H, Hussaini MY, Sussman M. Tracking discontinuities in hyperbolic conservation laws with spectral accuracy. *Journal of Computational Physics* 2007; **225**:1810–1826.
12. Cockburn B, Shu CW. TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws. V: multidimensional systems. *Journal of Computational Physics* 1998; **141**:199–224.
13. Qiu J, Shu CW. Runge–Kutta discontinuous Galerkin method using WENO limiters. *SIAM Journal on Scientific Computing* 2005; **26**:907–929.
14. van der Vegt JJW, van der Ven H. Space–time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows. Part I. General formulation. *Journal of Computational Physics* 2004; **182**:546–585.
15. Lowrie RB, Roe PL, van Leer B. Space–time methods for hyperbolic conservation laws. *Barriers and Challenges in Computational Fluid Dynamics*. ICASE/LaRC Interdisciplinary Series in Science and Engineering, vol. 6. National Institute of Aerospace, 1998; 79–98.
16. Palaniappan J, Haber RB, Jerrard RL. A spacetime discontinuous Galerkin method for scalar conservation laws. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:3607–3631.
17. Jaffré J, Johnson C, Szepessy A. Convergence of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *Mathematical Models and Methods in Applied Sciences* 1995; **5**:367–386.
18. Cockburn B, Gremaud PA. Error estimates for finite element methods for hyperbolic conservation laws. *SIAM Journal on Numerical Analysis* 1996; **33**:522–554.
19. Persson PP, Peraire J. Sub-cell shock capturing for discontinuous Galerkin methods. *Forty-fourth AIAA Aerospace Sciences Meeting and Exhibit*, AIAA, Reno, NV, U.S.A., 2006; 5–18.
20. Abedi R, Chung SH, Erickson J, Fan Y, Garland M, Guoy D, Haber R, Sullivan J, Thite S, Zhou Y. Spacetime meshing with adaptive refinement and coarsening. *Twentieth Annual ACM Symposium on Computational Geometry*, Brooklyn, NY, U.S.A., 2004; 300–309.
21. Abedi R, Chung SH, Hawker MA, Palaniappan J, Haber RB. Modeling evolving discontinuities with spacetime discontinuous Galerkin methods. In *IUTAM Symposium on Discretization Methods for Evolving Discontinuities*, Combescure AR, de Borst R, Belytschko T (eds). IUTAM Bookseries, vol. 5. Springer: Berlin, 2007; 59–87.
22. Arnold VI. *Mathematical Methods of Classical Mechanics* (2nd edn). Springer: New York, 1989.
23. Spivak M. *Calculus on Manifolds*. W.A. Benjamin: New York, 1965.
24. Abedi R, Haber R, Thite S, Erickson J. An *h*-adaptive spacetime-discontinuous Galerkin method for linearized elastodynamics. *Revue Européenne de Mécanique Numérique* (*European Journal of Computational Mechanics*) 2006; **15**(6):619–642.
25. Shu CW, Osher S. Efficient implementation of essentially non-oscillatory shock-capturing schemes II. *Journal of Computational Physics* 1989; **83**:32–78.
26. Casper J, Carpenter MH. Computational considerations for the simulation of shock-induced sound. *SIAM Journal on Scientific Computing* 1998; **19**:813–828.
27. Efraimsson G, Kreiss G. A remark on numerical errors downstream of slightly viscous shocks. *SIAM Journal on Numerical Analysis* 1999; **36**:853–863.
28. Sod GA. A survey of finite-difference methods for systems of nonlinear conservation laws. *Journal of Computational Physics* 1978; **27**:1–31.
29. Harten A, Lax PD, van Leer B. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review* 1983; **25**:35–61.
30. Toro E. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer: Berlin, 1997.

DOI: 10.1002/fld