Modeling Evolving Discontinuities with Spacetime Discontinuous Galerkin Methods

Reza Abedi[1], Shuo-Heng Chung[2], Morgan A. Hawker[3], Jayandran Palaniappan[1], and Robert B. Haber[1]

[1] Department of Mechanical Science & Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA; {r-haber, rabedi, palanppn}@uiuc.edu
[2] Department Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA; schung6@uiuc.edu
[3] C&I Engineering, Louisville, KY 40218, USA (formerly, Department of Theoretical & Applied Mechanics, University of Illinois at Urbana-Champaign); mahawker@gmail.com

Summary. We review recent progress in applying spacetime discontinuous Galerkin (SDG) finite element methods to problems whose solutions exhibit various types of moving discontinuities. SDG models and related solution methods offer a number of attractive features, including element-wise satisfaction of the governing balance laws, linear computational complexity in the number of spacetime elements, and a computational structure that readily supports parallel implementations. We describe the use of new unstructured spacetime meshing procedures ind discretizing evolving discontinuities. Specifically, we show how h-adaptive spacetime meshing can be used to capture weak shocks in linear elastodynamics, how the SDG framework provides a convenient setting for implementing cohesive models for dynamic fracture, and how more advanced spacetime meshing procedures can deliver sharp representations of discontinuous solution features by tracking the trajectories of contact discontinuities in compressible gas dynamics.

Key words: spacetime, discontinuous Galerkin, adaptive analysis, cohesive fracture, interface tracking

1 Introduction

1.1 Numerical Representations of Evolving Discontinuities

Discontinuous fields are common in continuum physics, yet their accurate representation remains one of the most challenging problems in computational mechanics, especially when the loci of singular surfaces evolve during the course of a simulation and are unknown *a priori*. Examples of evolving

discontinuities include shocks in solid and fluid dynamics, growing cracks and moving phase boundaries arising in heterogeneous flows or the microstructures of solid materials.

Standard conforming finite element models do not admit discontinuous functions. However, even for various nonconforming models that admit discontinuities, singular surfaces are generally restricted to discrete, mesh-related manifolds whose trajectories must be aligned with the discontinuous solution features to obtain an accurate result. This alignment poses special challenges, especially in problems where singular surfaces nucleate spontaneously or alter their connectivity through self-intersection. Beyond simply admitting discontinuous solution features in the discrete model, it is critical to enforce the correct jump conditions arising from the relevant balance laws and to respect the relevant interface kinetics to obtain a physically meaningful model.

Discretizing evolving discontinuities remains a ubiquitous and challenging problem in computational science and engineering and an active area in numerical methods research. We do not attempt a comprehensive review here. Instead, the following subsections identify some of the primary classes of available methods for resolving discontinuous fields, particularly those related to the methods presented in the later sections of this work. Our focus is primarily on finite element methods that address this issue.

Discontinuity/Shock Capturing with h-Refinement

We refer to methods that attempt to approximate discontinuous solution fields in continuous solution spaces as *capturing methods*; these are by far the most popular approach. The main advantage of this approach is that no special meshing procedures are needed to make the grid conform to evolving discontinuities, especially when the connectivities of singular surfaces change over time. The mesh might be Lagrangian or Eulerian and structured or unstructured, depending on the problem at hand. The solution space must be sufficiently enriched so that the continuous approximations of jumps are sufficiently sharp. Satisfying this requirement for problems with evolving discontinuities on a static grid can be prohibitively expensive, so some form of adaptive analysis is desirable, even though the mesh is not required to track the discontinuity.

Conforming finite element methods based on Bubnov-Galerkin projections suffer severe numerical artifacts and might even fail, especially for hyperbolic and hyperbolic-parabolic problems, when applied to problems with discontinuous soltuions. Non-local Gibbs oscillations as well as overshoot and undershoot at shocks are common problems. There is an extensive literature on stabilized finite element methods and shock-capturing operators to address these problems. The streamline upwind Petrov-Galerkin (SUPG) method, the Galerkin/least squares (GLS) method and the use of bubble functions exemplify successful finite element techniques for this class of problems (see for example, [8, 9, 23, 24, 26, 42]). The formulation of new methods with improved

shock-capturing properties remains an active research area; recent contributions include spectral vanishing viscosity methods [27, 44] and sub-cell shock capturing [39].

Cohesive Damage Models for Fracture

Cohesive damage models, first introduced by Dugdale [15] and Barenblatt [5], simulate crack initiation and growth by modeling the macroscopic effects of various nonlinear damage processes in the neighborhood of the crack tip. A constitutive *traction–separation law* (TSL) describes the traction acting on a cohesive interface as a nonlinear, bounded function of the separation across the interface [20, 34, 35, 45, 49, 51]. The TSL eliminates the crack-tip stress singularities that arise in classical fracture mechanics and introduces a microscopic length scale that is essential to the fracture model [19].

Numerical implementations of cohesive damage models present three special requirements: the model must admit jumps in the displacement field across cohesive surfaces, there must be a mechanism for enforcing the traction–separation law, and there must be sufficient enrichment to capture the evolution of mechanical fields associated with growing cracks. Special *cohesive elements* embedded in a mesh of conforming finite elements provide a common means to address the first two requirements [10, 13, 49]. The cohesive elements discretize the cohesive interface and the TSL is incorporated into their constitutive model. Alternatively, the cohesive law can also be enforced as a boundary condition on the cohesive interface [34, 45]. More recently, partition of unity, extended finite element or generalized finite element methods have emerged as alternative means to model cracks and cohesive interfaces within finite elements (as opposed to exclusively at boundaries) [6, 14, 31–33, 48]. These methods relax the coupling between the mesh geometry and the crack path.

Sufficient grid refinement is required to capture the detailed response in the vicinity of the active process zone and to enforce accurately the nonlinear traction–separation law. Numerical instabilities can arise if a minimum level of mesh refinement is not realized along the cohesive process zone [20]. The pattern of refinement must evolve to track the moving process zones as a crack propagates. Schrefler et al. periodically remesh the structure to model this phenomenon in quasi-static fracture analysis [41]. In the case of dynamic fracture, mesh refinement must also track the trajectories of sharp wavefronts that are either emitted by moving crack tips or generated by shock loading. Adaptive refinement algorithms can guarantee accuracy and stability, while avoiding the prohibitive expense of uniform mesh refinement. Pandolfi and Ortiz adaptively insert cohesive elements to follow arbitrary crack paths in fragmentation studies [38]. A recent adaptive analysis procedure by Krysl and Belytschko [28] employs an element-free Galerkin method to simplify adaptive refinement and track the crack in an arbitrary direction.

Tracking Methods

We use the term *discontinuity tracking* to describe methods that attempt to model solution discontinuities with true jumps in the discrete solution space by aligning certain mesh features that accommodate jumps with evolving discontinuous features. The most common choice is to associate jumps with element boundaries, but recent applications of extended finite element methods [53] and related partition of unity methods [4] allow greater flexibility by allowing discontinuities to pass through element interiors. However, it is still necessary to align integration cells with intra-element discontinuities and to respect the usual geometric quality constraints on the subdivided element domains.

The mesh must track discontinuities continuously through a time step in dynamics applications, rather than simply update the mesh between time steps. Thus, some form of moving-grid method is required. The Arbitrary Lagrangian–Eulerian (ALE) methods were among the first finite element procedures to implement this strategy [22]; there have been numerous applications since (see for example, [30]). In contrast to capturing methods, tracking methods' support for true jumps in the discrete solution also presents an opportunity to address the associated jump conditions in the formulation.

On the other hand, continuous tracking methods introduce a number of complications. Moving meshes distort element geometries and eventually trigger re-meshing operations to (if constraints on minimum element quality are enforced). Unfortunately, re-meshing itself generates significant errors as the solution is projected from the old mesh to the new. One can limit the mesh distortion by moving free nodes in concert with the motions of nodes constrained to follow a moving singular surface, but this usually adds cost and complexity to the algorithm. In Lagrangian formulations, moving mesh methods generate convective terms that might require special treatment in the finite element formulation. Finally, tracking methods can become intractable when the connectivities of singular surfaces evolve during the course of the simulation. Nonetheless, for problems where they are feasible, their ability to render true discontinuities without strong mesh refinement while enforcing the proper jump conditions often make tracking methods the most accurate and least expensive option for discretizing a discontinuity.

Implicit geometry models, such as level set methods [36], provide a powerful approach for modeling evolving interfaces with complex geometry and varying connectivity (see for example, [17]). These methods combine aspects of capturing and tracking methods; although the level-set contour tracks the evolving interface, a capturing method is typically used (often on a fixed Eulerian grid) to model the discontinuous response.

1.2 Spacetime Discontinuous Galerkin Methods

This contribution reviews recent progress in the use of adaptive spacetime discontinuous Galerkin (SDG) finite element methods for tracking evolving

discontinuities. See [12] for a survey of discontinuous Galerkin finite element methods. In this work, our focus is on spacetime discontinuous Galerkin (SDG) methods that involve direct discretization of spacetime domains and that use basis functions that are fully discontinuous across all spacetime element boundaries. When applied to hyperbolic problems and implemented on suitably constructed spacetime grids, SDG methods support efficient element-by-element or patch-by-patch solution procedures [2, 11, 25, 29, 40, 50]. SDG solution methods share a number of attractive features:

- element-wise satisfaction of the governing balance/conservation laws
- linear computational complexity in the number of spacetime elements (for fixed polynomial order)
- support for high-order approximations on a fixed, compact stencil
- support for spacetime adaptive meshing operations with zero projection error
- a computational structure that lends itself to parallel implementations

The discontinuous nature of the SDG basis functions might appear to address directly the problem of discretizing evolving discontinuities. Indeed, we illustrate the use of this property for discretizing growing cracks and moving contact discontinuities in Subsections 5.2 and 5.3 below. However, these techniques require a careful alignment between the spacetime grid and the trajectories of singular surfaces that might not be easy to achieve. Especially in cases where the connectivity of the discontinuous features changes over time, the flexibility of spacetime adaptive meshing and the accuracy delivered by the element-wise balance properties are of equal or greater importance in capturing moving discontinuities. We demonstrate both approaches below.

1.3 Organization of This Paper

The remaining content of this paper is organized as follows. Section 2 introduces the special notation based on the exterior calculus and differential forms used to formulate our spacetime methods. Section 3 presents summaries of the SDG formulations for linear elastodynamics and for the inviscid Euler equations for gas dynamics that support the numerical examples in this paper. Explanations of our spacetime meshing procedures and the patch-by-patch SDG solution procedure they support appear in Section 4. Section 5 presents three applications of the SDG method to modeling evolving discontinuities. These cover capturing weak shocks in elastodynamics *via* h-adaptive spacetime meshing, implementation of a cohesive damage model for elastodynamic crack growth within the SDG framework, and a study of the use of interface tracking in spacetime for modeling a moving contact discontinuity in a shock tube simulation. Section 6 presents conclusions and prospects for further development.

2 Differential Forms and Notation

This section introduces special notation used in our spacetime formulations that accesses the mathematical machinery of differential forms and the exterior calculus on manifolds. See [7, 18, 43] for a more complete development of these subjects and [2] for more detail on our particular usage. While this notation is non-standard in mechanics it provides a direct coordinate-free notation for the spacetime setting that delivers similar advantages to those offered by direct tensor notation in the purely spatial setting. In particular, we obtain a simple and intuitive means for expressing balance and conservation laws over arbitrary spacetime subdomains that automatically include the relevant jump conditions on the spacetime trajectories of singular surfaces.

Let d be the spatial dimension, and let the reference spacetime analysis domain D be an open $(d+1)$-manifold in $\mathbb{E}^d \times \mathbb{R}$ with a regular boundary. The spacetime coordinates $\left(x^1, \ldots, x^d, t\right)$ are the spatial coordinates followed by the time coordinate and are defined with respect to the ordered basis $(\mathbf{e}_1, \ldots, \mathbf{e}_d, \mathbf{e}_t)$. The dual basis is $\left(\mathbf{e}^1, \ldots, \mathbf{e}^d, \mathbf{e}^t\right)$. We follow the standard summation convention; latin indices range from 1 through d, except the index 't' which denotes time and does not imply summation when repeated.

The top form on D is $\boldsymbol{\Omega} := \mathbf{d}x^1 \wedge \ldots \wedge \mathbf{d}x^d \wedge \mathbf{d}t$, where we make use of the standard ordered basis for 1-forms: $\left(\mathbf{d}x^1, \ldots, \mathbf{d}x^d, \mathbf{d}t\right)$. The standard ordered basis for d-forms is $(\mathbf{d}\hat{x}_1, \ldots, \mathbf{d}\hat{x}_d, \star\mathbf{d}t)$, in which $\mathbf{d}\hat{x}_j := \star\mathbf{d}x^j$ where $\star$ is the Hodge star operator. Thus, $\mathbf{d}x^j \wedge \mathbf{d}\hat{x}_k = \delta^j_{\ k}\boldsymbol{\Omega}$; $\mathbf{d}t \wedge \mathbf{d}\hat{x}_k = \mathbf{0}$; $\mathbf{d}t \wedge \star\mathbf{d}t = \boldsymbol{\Omega}$; and $\mathbf{d}x^j \wedge \star\mathbf{d}t = \mathbf{0}$. We identify $\mathbf{d}\hat{x}_j$ and $\star\mathbf{d}t$ with, respectively, the differential spacetime volume element that is the geometric dual of $\mathbf{e}^j$ and the purely spatial differential volume element. We use $\boldsymbol{i} := \boldsymbol{i}_{\mathbf{e}_t}$ to denote the temporal insertion operator.

We use forms with scalar and tensor coefficients to develop our theory. Bold Italic symbols denote forms (with either scalar or tensor coefficients), while bold non-Italic symbols denote vector and tensor fields on D. We introduce a special 1-form with vector coefficients, $\boldsymbol{d}\mathbf{x} := \mathbf{e}_i \boldsymbol{d}x^i$, and a special d-form with vector coefficients, $\star\boldsymbol{d}\mathbf{x} := \mathbf{e}^i \boldsymbol{d}\hat{x}_i$. We define the exterior product of two forms with tensor coefficients as

$$\mathbf{a}\boldsymbol{\psi} \wedge \mathbf{b}\boldsymbol{\omega} := \mathbf{a}(\mathbf{b})(\boldsymbol{\psi} \wedge \boldsymbol{\omega}) \tag{1}$$

in which $\mathbf{a}$ and $\mathbf{b}$ are tensor fields on D of orders m and n respectively $(m \geq n)$, and $\boldsymbol{\psi}$ and $\boldsymbol{\omega}$ are p and q-forms on D such that $p + q \leq d + 1$.

3 Formulations

This section presents brief reviews of the SDG formulations for linearized elastodynamics and the Euler equations for inviscid gas dynamics. Component expansions are provided to illuminate the relationship between the differential forms expression. More detailed formulations can be found in the references cited below.

3.1 Elastodynamics

This subsection presents a summary of a single-field SDG formulation of linearized elastodynamics [2]. The displacement field $\mathbf{u} = u_i \mathbf{e}^i$, the primary solution variable on D, is related to the strain and velocity 1-forms according to

$$\boldsymbol{E} = \mathbf{E} \wedge \boldsymbol{dx} = E_{ij} \mathbf{e}^i \otimes \mathbf{e}^j \wedge \mathbf{e}_k \boldsymbol{dx}^k = E_{ij} \mathbf{e}^i \boldsymbol{dx}^j \tag{2}$$

$$\boldsymbol{v} = \dot{\mathbf{u}} \boldsymbol{dt} = \dot{u}_i \mathbf{e}^i \boldsymbol{dt} \tag{3}$$

where $\mathbf{E} = \text{sym}\, \nabla \mathbf{u}$. These two kinematic quantities sum to form a single spacetime 1-form $\boldsymbol{\varepsilon}$ called the *strain-velocity*:

$$\boldsymbol{\varepsilon} := \boldsymbol{E} + \boldsymbol{v} \tag{4}$$

The fundamental force-like quantities are the *stress* and *linear momentum density* d-forms given by

$$\boldsymbol{\sigma} := \bar{\boldsymbol{\sigma}} \wedge \star \boldsymbol{dx} = \sigma^{ij} \mathbf{e}_i \otimes \mathbf{e}_j \wedge \mathbf{e}^k \boldsymbol{d\hat{x}}_k = \sigma^{ij} \mathbf{e}_i \boldsymbol{d\hat{x}}_j \tag{5}$$

$$\boldsymbol{p} := \mathbf{p} \star \boldsymbol{dt} = \rho \dot{u}^i \mathbf{e}_i \star \boldsymbol{dt} \tag{6}$$

where $\bar{\boldsymbol{\sigma}}$ is the symmetric Cauchy stress tensor, $\mathbf{p}$ is the linear momentum density vector and ρ is the mass density. The standard relationship of linear elasticity between $\boldsymbol{\varepsilon}$ and $\bar{\boldsymbol{\sigma}}$ is strongly enforced. The restriction of $\boldsymbol{\sigma}$ to a spacetime d-manifold Γ with arbitrary orientation delivers the surface traction $\mathbf{t}_\Gamma$ acting on Γ: $\boldsymbol{\sigma}|_\Gamma = \mathbf{t}_\Gamma$.

The *spacetime momentum flux* $\boldsymbol{M}$ is defined as

$$\boldsymbol{M} := \boldsymbol{\sigma} - \boldsymbol{p} \tag{7}$$

such that the restriction of $\boldsymbol{M}$ to Γ delivers the flux of linear momentum across Γ. The exterior derivative of $\boldsymbol{M}$ is the $(d+1)$-form whose vector coefficient is the residual of the homogeneous Equation of Motion:

$$\boldsymbol{dM} = (\boldsymbol{\nabla} \cdot \bar{\boldsymbol{\sigma}} - \dot{\mathbf{p}})\, \boldsymbol{\Omega} = \left(\bar{\sigma}^{ij}{}_{,j} - \rho \ddot{u}^i \right) \mathbf{e}_i \boldsymbol{\Omega} \tag{8}$$

The *body force* is given by the $(d+1)$-form, $\boldsymbol{b} = \mathbf{b}\boldsymbol{\Omega} = b^i \mathbf{e}_i \boldsymbol{\Omega}$, where $\mathbf{b}$ is the body force vector per unit mass. Thus, the vector coefficient of the $(d+1)$-form $\boldsymbol{dM} + \rho \boldsymbol{b}$ is the residual of the Equation of Motion.

Consider the integral form of Balance of Linear Momentum,

$$\int_{\partial Q} \boldsymbol{M} + \int_Q \rho \boldsymbol{b} = \mathbf{0} \quad \forall Q \subset D \Rightarrow \int_Q (\boldsymbol{dM} + \rho \boldsymbol{b}) = \mathbf{0} \quad \forall Q \subset D \tag{9}$$

in which subdomains Q are assumed to have regular boundaries, and we use the Stokes Theorem to eliminate the boundary flux integral. Let Γ^J be the union of jump manifolds on D where $\boldsymbol{M}$ and $\boldsymbol{\varepsilon}$ are possibly discontinuous.

Apply the Localization Theorem to (9) to obtain the Equation of Motion, including the jump part of $\boldsymbol{dM}$, denoted $[\boldsymbol{M}]$, on $\Gamma^J \cup \partial D$:

$$(\boldsymbol{dM} + \rho\boldsymbol{b})|_{D\setminus\Gamma^{\mathrm{J}}} = \mathbf{0} \tag{10a}$$

$$[\boldsymbol{M}]|_{\Gamma^{\mathrm{J}}\cup\partial D} = \mathbf{0} \tag{10b}$$

We replace (10b) with a stronger condition,

$$(\boldsymbol{M}^* - \boldsymbol{M})|_{\Gamma^{\mathrm{J}}\cup\partial D} = \mathbf{0} \tag{11}$$

in which $\boldsymbol{M}^*$ is either a Godunov or prescribed value of the momentum flux.

In the SDG setting, we introduce a mesh partition $\mathcal{P}$ of D into finite elements, and introduce a discrete bases so that $\boldsymbol{M}$ and $\boldsymbol{\varepsilon}$ are continuous within each element but are possibly discontinuous across element boundaries. Then the strong form of balance of linear momentum augmented by a kinematic compatibility condition can be expressed as follows. Let V^Q be the solution space for the displacement field on element $Q \in \mathcal{P}$. We then have the following strong form and weighted residual statement:

Problem 1 (Strong Form). For each $Q \in \mathcal{P}$, find $\mathbf{u} \in V^Q$ such that

$$\boldsymbol{dM} + \rho\boldsymbol{b} = \mathbf{0} \quad \text{in } Q \tag{12a}$$

$$(\boldsymbol{M}^* - \boldsymbol{M})|_{\partial Q} = \mathbf{0} \tag{12b}$$

$$(\boldsymbol{\varepsilon}^* - \boldsymbol{\varepsilon})|_{\partial Q} = \mathbf{0} \tag{12c}$$

$$(\mathbf{u}_0^* - \mathbf{u}_0)\star\boldsymbol{dt}|_{\partial Q^{\mathrm{ti}}} = \mathbf{0} \tag{12d}$$

in which $\mathbf{u}_0$ is a projection of the displacement solution $\mathbf{u}$ into a subspace with vanishing total energy, ∂Q^{ti} is the time-inflow boundary of Q and the values $\boldsymbol{M}^*$, $\boldsymbol{\varepsilon}^*$ and $\mathbf{u}_0^*$ are either Godunov or prescribed values, as described below. The jump conditions (12c,d) enforce kinematic boundary conditions and compatibility across element boundaries.

Problem 2 (Weighted Residual Statement). For each $Q \in \mathcal{P}$, find $\mathbf{u} \in V^Q$ such that $\forall\hat{\mathbf{u}} \in V^Q$,

$$\int_Q \dot{\hat{\mathbf{u}}} \wedge (\boldsymbol{dM} + \rho\boldsymbol{b}) + \int_{\partial Q} \left\{\dot{\hat{\mathbf{u}}} \wedge (\boldsymbol{M}^* - \boldsymbol{M}) + (\boldsymbol{\varepsilon}^* - \boldsymbol{\varepsilon}) \wedge \boldsymbol{i}\hat{\boldsymbol{\sigma}}\right\} + \int_{\partial Q^{\mathrm{ti}}} k\hat{\mathbf{u}} \wedge (\mathbf{u}_0^* - \mathbf{u}_0)\star\boldsymbol{dt} = 0 \tag{13}$$

in which $\hat{\mathbf{u}}$ is the weighting function, and k is a constant introduced for dimensional consistency.

An application of the Stokes Theorem leads to the weak problem statement:

Problem 3 (Weak Form). For each $Q \in \mathcal{P}$, find $\mathbf{u} \in V^Q$ such that $\forall\hat{\mathbf{u}} \in V^Q$,

$$\int_Q (-\boldsymbol{d}\dot{\hat{\mathbf{u}}} \wedge \boldsymbol{M} + \dot{\hat{\mathbf{u}}} \wedge \rho \boldsymbol{b}) + \int_{\partial Q} \left\{ \dot{\hat{\mathbf{u}}} \wedge \boldsymbol{M}^* + (\boldsymbol{\varepsilon}^* - \boldsymbol{\varepsilon}) \wedge \boldsymbol{i}\hat{\boldsymbol{\sigma}} \right\} + \int_{\partial Q^{\mathrm{ti}}} k\hat{\mathbf{u}} \wedge (\mathbf{u}_0^* - \mathbf{u}_0) \star \boldsymbol{d}t = 0 \tag{14}$$

Replacing V^Q in Problem 3 with a discrete subspace, $V_h^Q \subset V_Q$, generates the SDG finite element formulation. It is easy to show that the discrete form of Problem 3 balances both linear and angular momentum over every spacetime element [2].

We define the target boundary values of $\boldsymbol{M}$ and $\boldsymbol{\varepsilon}$ for each $Q \subset D$:

$$\boldsymbol{M}^* = \begin{cases} \boldsymbol{M} & \text{on } \partial Q^{\mathrm{co}} \cup (\partial Q \cap \partial D^{\boldsymbol{\varepsilon}}) \\ \boldsymbol{M}^+ & \text{on } \partial Q^{\mathrm{ci}} \setminus \partial D^{\mathrm{ci}} \\ \boldsymbol{M}^{\mathrm{G}}(\boldsymbol{M}, \boldsymbol{M}^+, \boldsymbol{a}_{\partial Q}) & \text{on } \partial Q^{\mathrm{nc}} \setminus \partial D^{\mathrm{nc}} \\ \underline{\boldsymbol{M}} & \text{on } \partial Q \cap \left(\partial D^{\mathrm{ci}} \cup \partial D^{\boldsymbol{M}}\right) \end{cases} \tag{15a}$$

$$\boldsymbol{\varepsilon}^* = \begin{cases} \boldsymbol{\varepsilon} & \text{on } \partial Q^{\mathrm{co}} \cup \left(\partial Q \cap \partial D^{\mathbf{M}}\right) \\ \boldsymbol{\varepsilon}^+ & \text{on } \partial Q^{\mathrm{ci}} \setminus \partial D^{\mathrm{ci}} \\ \boldsymbol{\varepsilon}^{\mathrm{G}}(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}^+, \mathbf{a}_{\partial Q}) & \text{on } \partial Q^{\mathrm{nc}} \setminus \partial D^{\mathrm{nc}} \\ \underline{\boldsymbol{\varepsilon}} & \text{on } \partial Q \cap \left(\partial D^{\mathrm{ci}} \cup \partial D^{\boldsymbol{\varepsilon}}\right) \end{cases} \tag{15b}$$

in which $\boldsymbol{M}$ and $\boldsymbol{\varepsilon}$ are traces of interior fields on Q, $\boldsymbol{M}^+$ and $\boldsymbol{\varepsilon}^+$ are traces of exterior fields on Q, ∂Q^{co} is the causal (space-like) outflow boundary of Q, ∂Q^{ci} (∂D^{ci}) is the causal inflow boundary of Q (D), and ∂Q^{nc} (∂D^{nc}) is the non-causal (time-like) boundary of Q (D). The non-causal domain boundary ∂D^{nc} is partitioned into disjoint parts, $\partial D^{\boldsymbol{M}}$ and $\partial D^{\boldsymbol{\varepsilon}}$, where $\boldsymbol{M}$ and $\boldsymbol{\varepsilon}$ are determined by the prescribed functions $\underline{\boldsymbol{M}}$ and $\underline{\boldsymbol{\varepsilon}}$. The Godunov values, $\boldsymbol{M}^{\mathrm{G}}$ and $\boldsymbol{\varepsilon}^{\mathrm{G}}$, are obtained from the solutions to local Riemann problems [2].

3.2 Euler Equations

We define the conservation fields: mass density ρ, linear momentum density $\mathbf{p} = \rho\mathbf{v}$, and total energy denstiy $\mathcal{E} = e + \frac{1}{2}|\mathbf{v}|^2$ where e is the specific internal energy. Let $p = \frac{\rho e}{\gamma - 1}$ be the pressure with adiabatic index γ and $\bar{\boldsymbol{\sigma}} = p\delta^{ij}\mathbf{e}_i \otimes \mathbf{e}_j$ be the stress. The *spacetime conservation fluxes* and their exterior derivatives are

$$\boldsymbol{F}_\rho = \rho \star \boldsymbol{d}t + \rho\mathbf{v} \wedge \star \boldsymbol{d}\mathbf{x} \quad \boldsymbol{d}\boldsymbol{F}_\rho = (\dot{\rho} + \boldsymbol{\nabla} \cdot \rho\mathbf{v})\, \boldsymbol{\Omega} \tag{16a}$$

$$\boldsymbol{F}_{\mathbf{p}} = \mathbf{p} \star \boldsymbol{d}t + (\mathbf{p} \otimes \mathbf{v} - \bar{\boldsymbol{\sigma}}) \wedge \star \boldsymbol{d}\mathbf{x} \quad \boldsymbol{d}\boldsymbol{F}_{\mathbf{p}} = [\dot{\mathbf{p}} + \boldsymbol{\nabla} \cdot (\mathbf{p} \otimes \mathbf{v} - \bar{\boldsymbol{\sigma}})]\, \boldsymbol{\Omega} \tag{16b}$$

$$\boldsymbol{F}_{\mathcal{E}} = \mathcal{E} \star \boldsymbol{d}t + [\mathcal{E}\mathbf{v} + \bar{\boldsymbol{\sigma}}(\mathbf{v})] \wedge \star \boldsymbol{d}\mathbf{x} \quad \boldsymbol{d}\boldsymbol{F}_{\mathcal{E}} = \left\{ \dot{\mathcal{E}} + \boldsymbol{\nabla} \cdot [\mathcal{E}\mathbf{v} + \bar{\boldsymbol{\sigma}}(\mathbf{v})] \right\} \boldsymbol{\Omega} \tag{16c}$$

Note that the exterior derivatives render local residual forms of conservation of mass, linear momentum and energy. We rewrite the fluxes in (16) in vector format as

$$\boldsymbol{F} := \left\{ \begin{array}{c} \boldsymbol{F}_\rho(\mathbf{q}) \\ \boldsymbol{F}_{\mathbf{p}}(\mathbf{q}) \\ \boldsymbol{F}_{\mathcal{E}}(\mathbf{q}) \end{array} \right\} \qquad \mathbf{q} := \left\{ \begin{array}{c} \rho \\ \mathbf{p} \\ \mathcal{E} \end{array} \right\}. \tag{17}$$

Following arguments similar to those in the previous subsection, we write the integral and local forms of the system conservation statement $\forall Q \subset D$:

$$\int_{\partial Q} \boldsymbol{F} = \mathbf{0} \Leftrightarrow \left\{ \begin{array}{l} \boldsymbol{dF}|_Q = \mathbf{0} \\ [\boldsymbol{F}]|_{\partial Q} = (\boldsymbol{F}^* - \boldsymbol{F})|_{\partial Q} = \mathbf{0}. \end{array} \right. \tag{18}$$

We use Godunov values for the fluxes $\boldsymbol{F}^*$; this ensures satisfaction of the entropy inequality over each subdomain Q.[4] The jump conditions in (18) are the *Rankine-Hugoniot* conditions that govern the motions of shocks.

We introduce a discrete subspace V_h^Q and apply a simple Bubnov-Galerkin projection to develop the SDG weighted residual statement and the weak form used for numerical implementation.

Problem 4 (SDG Weighted Residual Statement). For each $Q \in \mathcal{P}(D)$, find $\mathbf{q} \in V_h^Q$ such that

$$\int_Q \hat{\mathbf{q}} \wedge \boldsymbol{dF} + \int_{\partial Q} \hat{\mathbf{q}} \wedge (\boldsymbol{F}^* - \boldsymbol{F}) = \mathbf{0} \; \forall \hat{\mathbf{q}} \in V_h^Q. \tag{19}$$

Problem 5 (SDG Weak Statement). For each $Q \in \mathcal{P}(D)$, find $\mathbf{q} \in V_h^Q$ such that

$$-\int_Q \boldsymbol{d}\hat{\mathbf{q}} \wedge \boldsymbol{F} + \int_{\partial Q} \hat{\mathbf{q}} \wedge \boldsymbol{F}^* = \mathbf{0} \; \forall \hat{\mathbf{q}} \in V_h^Q \tag{20}$$

A piece-wise constant choice for $\hat{\mathbf{q}}$ in (20) proves element-wise conservation:

$$\int_{\partial Q} \boldsymbol{F}^* = \mathbf{0} \; \forall Q \in \mathcal{P}(D) \tag{21}$$

In general, stabilization is required to control local overshoot and to ensure a robust numerical method. However, in problems with convex flux functions and where a limited amount of overshoot is acceptable, the basic method presented here can be applied.

4 Causal Spacetime Meshing and Patch-wise Solution Procedure

The construction of causal spacetime meshes and a patch-by-patch solution procedure with linear computational complexity in the number of patches, as described in [1, 2, 37], are key aspects of our implementation of the SDG

[4] This has been proven only for scalar case in [37].

method. We briefly summarize these concepts here; the reader is referred to [1, 2, 37] for a more detailed treatment.

We use an advancing-front mesh generation procedure called *Tent Pitcher* [16, 46] to construct spacetime meshes in 2D $\times$ time that satisfy a patch-wise *causality condition*. A patch is a collection of contiguous tetrahedral spacetime elements whose boundary is, therefore, a collection of spacetime triangles. A triangle is *causal* if all the characteristics of the governing hyperbolic equations have the same orientation relative to the triangle at all points on the triangle. A patch is causal if all its boundary faces are either causal or part of the domain boundary. This implies that each boundary face of a causal patch can be classified as either an inflow, outflow or boundary face depending on the face's location and whether the characteristic flow across the face is inward or outward relative to the patch. Since no information can enter a patch through its outflow faces, the solution in each patch depends only on characteristic data flowing through its inflow and boundary faces. This implies a dependency graph that defines a partial ordering of patches wherein the solution on each patch depends only on boundary data and solutions on earlier patches in the partial ordering. Thus, patches can be solved locally, without approximation, using only boundary data and outflow data from previously-solved patches. Inter-element boundaries within a patch can be noncausal, so the elements within a patch must, in general, be solved simultaneously.

Tent Pitcher begins with a constant-time triangulation of the spatial domain at the initial time of the analysis interval. It then visits each vertex in the triangulation in turn to construct a new patch. The basic tent pitching procedure involves advancing a vertex in time to construct a new "tent pole" and then constructing a set of tetrahedra surrounding the tent pole to form a new patch. The height (duration) of the tent pole is limited to ensure that the outflow faces of the new patch are all causal. The tent-pole height might be further limited by a *progress constraint* to ensures that it is always possible to continue to pitch new patches. The product of the tent-pitching algorithm is an unstructured, patch-wise causal spacetime mesh. The local nature of the causality condition only enforces a local restriction on the duration of individual patches, similar to a CFL condition; there is no global restriction on the time step size, as in conventional explicit time-marching schemes. Thus, patches containing elements with larger spatial diameters and slower wave speeds can have longer durations than patches with smaller spatial diameters and higher wave speeds.

Our solution procedure interleaves mesh generation with the finite element solution; we compute the finite element solution on each patch as soon as it is generated by Tent Pitcher. Since the number of tetrahedral elements within a patch is limited by the highest vertex degree in the space triangulation, and because the highest polynomial order in each element is bounded, we can show that this patch-by-patch solution procedure has linear complexity in the number of spacetime patches. There is no need to assemble and store a global system of equations; the finite element routines are written to solve a single

patch at a time in a serial implementation. In a parallel computation, on the other hand, multiple tents can be pitched and solved simultaneously on separate processors, subject only to the partial ordering constraint for patches. The local character of the algorithm reduces the need for interprocessor communication.

5 Applications

This section presents three example applications that demonstrate different ways SDG finite element models can be used to model solutions with evolving discontinuities.

5.1 Discontinuity Capturing Using Adaptive Spacetime Meshing

We next present a brief review of how h-adaptive spacetime meshing can be used to capture weak shocks in elastodynamics, as reported in [3].[5] We seek local refinement along the trajectories of shocks, so that refinement in one location does not impose a global restriction on the durations of elements elsewhere in the spacetime analysis domain. Although the SDG formulation for elastodynamics presented in Subsection 3.1 balances linear and angular momentum over every spacetime element to within machine precision, the method is dissipative, and energy is generally not balanced. Experience shows that an error measure based on element-wise dissipation is an effective indicator for adaptive refinement, both for controlling the overall solution accuracy and for concentrating refinement along the trajectories of shocks. To achieve an efficient solution, we attempt to distribute a limited amount of numerical dissipation evenly over the spacetime elements.

Error Indicator for Element-wise Numerical Dissipation

The numerical energy dissipation for spacetime element Q is given by:

$$\varphi^Q = \frac{1}{2}\int_{\partial Q} (\dot{\mathbf{u}}^* \wedge \boldsymbol{M}^* + \boldsymbol{\varepsilon}^* \wedge \mathbf{i}\boldsymbol{\sigma}^*) + \int_Q \dot{\mathbf{u}} \wedge \rho \boldsymbol{b}. \tag{22}$$

Let φ^* be the user-specified target dissipation per element. The dissipation on element Q is considered acceptable when $\underline{\varphi} \leq \varphi^Q \leq \overline{\varphi}$, where $\underline{\varphi} = (1-\eta)\varphi^*$ and $\overline{\varphi} = (1+\eta)\varphi^*$ in which η is a user-specified parameter subject to $0 < \eta < 1$. Element Q is marked for refinement when $\varphi^Q > \overline{\varphi}$, and element Q is coarsenable when $\varphi^Q < \underline{\varphi}$. Otherwise, element Q is marked acceptable. The parameter η must be chosen sufficiently large to minimize undesirable cycling between coarsening and refinement. We use $\eta = 0.2$ in our current implementation.

[5] More precisely, the features in this example are sharp wavefronts that approximate weak shocks. In the interest of conciseness, we nonetheless refer to these solution features as shocks.

Adaptive Tent Pitcher Algorithm

We use an adaptive extension of the Tent Pitcher algorithm to implement adaptive refinement and coarsening within our patch-by-patch, advancing-front solution algorithm. Rather than adapting patches of spacetime elements directly, Tent Pitcher implements adaptive refinement by managing the triangulation of the current front. Each time a patch is solved, the solver checks whether any elements in the patch have been marked for refinement. If the result is positive, then the dissipation error is deemed unacceptable, and the solver *rejects* the patch when it is returned to Tent Pitcher. Tent Pitcher, in turn, discards the rejected patch and, using a newest-vertex-bisection algorithm [1], refines the triangles on the current mesh front that correspond to the elements marked for refinement. This effectively refines the subsequent spacetime mesh in both space and time when tent pitching is resumed, because the causality constraint dictates shorter tent-pole heights (local time steps) at vertices associated with refined triangles. Note that Tent Pitcher discards only the solution on the rejected patch. The solutions on all previously-solved patches are unaffected due to the patch-wise causal structure of the spacetime grid, so the amount of redundant calculation due to refinement is small.

Tent Pitcher accepts the solution on the current patch if all elements in the patch are either acceptable or coarsenable. In this case, Tent Pitcher stores the patch solution, advances the mesh front, and copies the status (acceptable or coarsenable) from the patch elements to the corresponding facets of the new mesh front. Requests for coarsening need not be acted on immediately, since they do not involve unacceptable error. Tent Pitcher's coarsening operation involves deleting a degree-4 vertex so as to merge two pairs of adjacent, coarsenable triangles into two two triangles in the active mesh front. A vertex can be deleted when all of the triangles surrounding it are marked coarsenable. In order to maintain the integrity of the spacetime grid, each pair of triangles must be coplanar before the coarsening operation can be executed. Typically, this requirement is not satisfied immediately, so Tent Pitcher postpones coarsening until it has pitched new tent poles with heights adjusted to meet the coplanarity constraint.

Numerical Example: Crack-Tip Wave Scattering

We next review results reported in greater detail in [2] that demonstrate the ability of the SDG formulation with adaptive spacetime meshing to capture shock-like features in elastodynamics. Figure 1 shows a center-cracked plate that we model using plane-stress assumptions, Young's modulus $E = 10$, Poisson ratio $\nu = 0.3$, and density $\rho = 2$. A spatially uniform tensile traction of magnitude T acts along the top and bottom edges of the plate. The magnitude T ramps rapidly from zero at time $t = 0.0$ to a maximum value of 10 at time $t = 0.002$ to approximate a weak stress-velocity shock; T holds constant at the maximum value until the simulation interval terminates at time $t =$

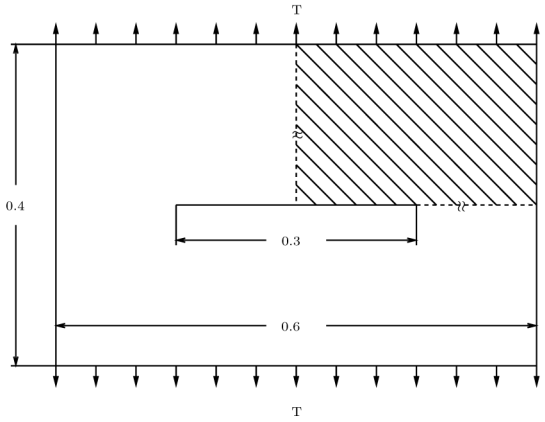


Fig. 1. Spatial domain and boundary conditions for the crack scattering problem

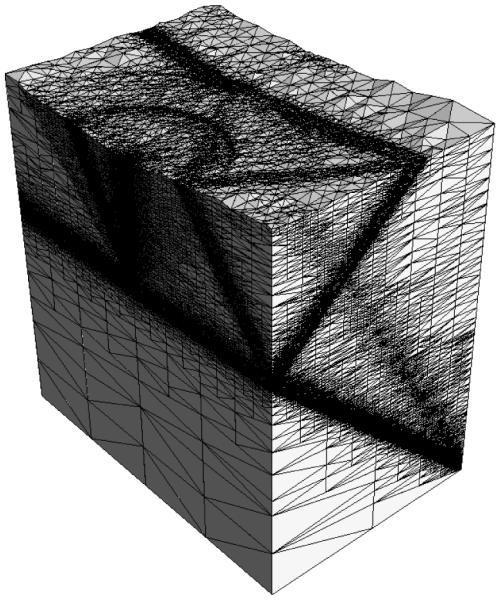


Fig. 2. Adaptive spacetime mesh for crack-tip wave scattering problem with 11 million tetrahedra (intermediate stage, front is at roughly $t = 0.100$)

0.300. We enforce symmetry boundary conditions to model only the shaded region shown in the figure, and we use complete cubic polynomials to model the displacement field within each spacetime tetrahedron. A uniform 2×4 rectangular grid defines the initial space mesh, with each rectangle subdivided into two triangles.

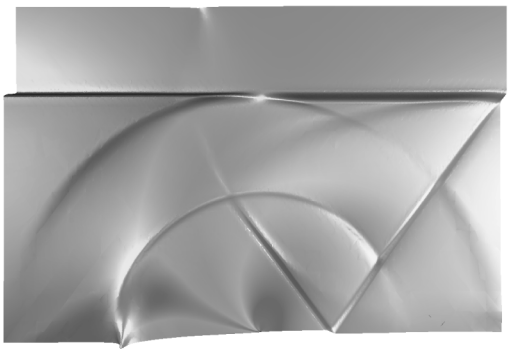


Fig. 3. Visualization of elastic wave scattering by a stationary crack-tip at $t = 0.105$)

Fig. 2 shows the state of the spacetime mesh constructed by the adaptive Tent Pitcher algorithm at an intermediate stage of the simulation. The spatial directions are aligned with the horizontal axes, and time increases upward in the vertical direction. The fine details of the elastodynamic solution are clearly evident in the pattern of mesh refinement. The dark diagonal bands along the right face of the spacetime volume are traces of plane-wave shocks generated by the sudden traction loading, while the lighter diagonal band ascending at a steeper angle traces the trajectory of a Rayleigh wave moving along the free edge of the plate. The spacetime trajectory of the crack tip runs along the vertical center-line of the left face of the spacetime volume. The apex of the cone-shaped region of mesh refinement indicates the initial scattering event. The separation of the scattered wave into dilatational and shear components can be seen in the pattern of refinement. The outer perimeter of the cone (tangent to the dark band on the top surface due to the plane wave) indicates the progress of the faster-moving dilatational wave, while the dark circular band within the cone traces the trajectory of the slower shear wave. In this example, the ratio of the largest to smallest element diameter is 1024. The ability of the adaptive SDG method to limit refinement to the trajectories of moving shocks and the absence of global time-step size constraints yield significant computational savings.

Fig. 3 shows a visualization of the adaptive solution at roughly the same time as the top of the spacetime volume in Fig. 2. The images were generated by a pixel-exact visualization system [52]. The strain energy density field is mapped to color (rendered as gray scale here); the velocity magnitude is mapped to the height field, which is then shaded by a lighting model to reveal its form. The visualization reveals fine features of the solution such as a Rayleigh wave moving along the crack surface and the scattered shear and pressure waves emanating from the crack-tip. Small-scale features, such as the shock fronts, are well resolved. The solution is free of spurious oscillations,

although no extra stabilization was added to the SDG formulation for this linear problem.

5.2 Cohesive Fracture Model

Cohesive damage models are a popular means for modeling discontinuous deformation fields in computational solid mechanics. First introduced by Dugdale [15] and Barenblatt [5], these relatively simple models describe fracture and delamination processes, including an initiation criterion and a means to calculate the rate of crack extension. They require supplementary criteria to determine the direction of crack growth. The works of Needleman and Xu [34,49] led to a strong resurgence of interest in cohesive damage models; the recent literature is extensive (see for example, [6, 10, 13, 14, 28, 31–33, 35, 48]). We do not attempt a comprehensive literature review here. Rather, we describe an implementation of the Xu and Needleman traction separation law for elastodynamic fracture [49] within the adaptive SDG framework described in the previous subsection. In this context, we note the related work of Huang and Costanzo [21].

By using cohesive damage models, one attempts to simulate crack initiation and extension by modeling the macroscopic effects of various nonlinear damage processes in the neighborhood of the crack tip. Specifically, a constitutive *traction–separation law* (TSL) describes the tractions acting on a cohesive interface as a nonlinear, bounded function of the interface separation. The TSL eliminates the crack-tip stress singularities that arise in classical fracture mechanics and introduces a microscopic length scale that is essential to the fracture model [19]. We restrict our attention to two spatial dimensions ($d = 2$) and to the history-independent, exponential relationship developed by Xu and Needleman [49], although the computational framework we describe could be adapted to most other TSLs.

Incorporation of Cohesive Damage Model in SDG Framework

In general, numerical implementations of cohesive damage models present three special requirements: the numerical model must admit jumps in the displacement field across cohesive interfaces; there must be a mechanism for enforcing the traction–separation law; and there must be some form of adaptive enrichment to resolve the active cohesive process zone and to capture the evolution of mechanical fields associated with growing cracks, including the sharp wavefronts generated by shock loads and by sudden crack initiation or arrest. Recalling the discontinuous format of the SDG basis, we address the first requirement by aligning element boundaries with cohesive interfaces. We weakly enforce the traction–separation law by introducing a special definition of the target momentum flux $\boldsymbol{M}^*$ on cohesive interfaces, as described below. The adaptive refinement strategy described in the preceding subsection effectively addresses the third requirement. However, we describe below the use

of an additional error indicator to ensure that the SDG solution renders the TSL accurately.

Let Γ^{C} denote the union of all the cohesive-interface trajectories in $\overline{D}$, and modify the partition of the non-causal domain boundary according to $\partial D^{\mathrm{nc}} = \overline{\partial D^{\mathbf{M}} \cup \partial D^{\boldsymbol{\varepsilon}} \cup \partial D^{\mathrm{C}}}$, where $\partial D^{\mathrm{C}} = \partial D \cap \Gamma^{\mathrm{C}}$. The only modification to the elastodynamic formulation required to implement the cohesive model is to replace (15) with

$$\boldsymbol{M}^* = \begin{cases} \boldsymbol{M} & \text{on } \partial Q^{\mathrm{co}} \cup (\partial Q \cap \partial D^{\boldsymbol{\varepsilon}}) \\ \boldsymbol{M}^+ & \text{on } \partial Q^{\mathrm{ci}} \setminus \partial D^{\mathrm{ci}} \\ \boldsymbol{M}^{\mathrm{G}}(\boldsymbol{M}, \boldsymbol{M}^+, \boldsymbol{a}_{\partial Q}) & \text{on } \partial Q^{\mathrm{nc}} \setminus \left(\partial D^{\mathrm{nc}} \cup \Gamma^{\mathbf{C}}\right) \\ \underline{\boldsymbol{M}} & \text{on } \partial Q \cap \left(\partial D^{\mathrm{ci}} \cup \partial D^{\boldsymbol{M}}\right) \\ \boldsymbol{M}^{\mathrm{C}} & \text{on } \partial Q \cap \Gamma^{\mathrm{C}} \end{cases} \tag{23a}$$

$$\boldsymbol{\varepsilon}^* = \begin{cases} \boldsymbol{\varepsilon} & \text{on } \partial Q^{\mathrm{co}} \cup \left[\partial Q \cap \left(\partial D^{\mathbf{M}} \cup \Gamma^{\mathbf{C}}\right)\right] \\ \boldsymbol{\varepsilon}^+ & \text{on } \partial Q^{\mathrm{ci}} \setminus \partial D^{\mathrm{ci}} \\ \boldsymbol{\varepsilon}^{\mathrm{G}}(\boldsymbol{\varepsilon}, \boldsymbol{\varepsilon}^+, \mathbf{a}_{\partial Q}) & \text{on } \partial Q^{\mathrm{nc}} \setminus \left(\partial D^{\mathrm{nc}} \cup \Gamma^{\mathbf{C}}\right) \\ \underline{\boldsymbol{\varepsilon}} & \text{on } \partial Q \cap \left(\partial D^{\mathrm{ci}} \cup \partial D^{\boldsymbol{\varepsilon}}\right) \end{cases} \tag{23b}$$

in which $\boldsymbol{M}^{\mathrm{C}} := \tilde{\mathbf{s}} \boldsymbol{dx}^{\mathrm{C}}$, where $\boldsymbol{dx}^{\mathrm{C}}$ is the spacetime d-volume element on Γ^{C} and $\tilde{\mathbf{s}}$ is the cohesive traction vector generated by the traction–separation relation.

We use two adaptive error indicators to ensure accurate solutions for both the bulk and cohesive responses. The first error indicator is the element-wise numerical dissipation, as described above. The second indicator uses the L-2 norm of the traction error along the cohesive interface. We use this indicator to ensure that the finite element tractions match the cohesive traction-separation law to within one percent. The combination of these adaptive criteria captures sharp wave-fronts in the bulk and strong gradients in the crack-tip fields.

Adaptive Enforcement of the Traction–Separation Relation

Cohesive models can produce numerical instability when too few elements are included in the active cohesive process zone [20]. Beyond stability problems, under-resolved finite element grids can generate significant errors in the work of separation and the history of crack-tip motion. Adaptive control of the numerical dissipation does not directly address these problems, so we introduce a second adaptive error indicator to limit the mismatch between the tractions generated by the finite element stress solution and the target cohesive tractions.

The *cohesive traction error on* Γ^{C} is given by $\tilde{\mathbf{s}} - \mathbf{s}$, where $\mathbf{s}$ is the vector traction coefficient of $\boldsymbol{\sigma}|_{\Gamma^{\mathrm{C}}}$. We define the *cohesive traction error indicator on element* Q as

$$\tau^Q := \frac{\| |\tilde{\mathbf{s}} - \mathbf{s}| \|_{L_2(\partial Q \cap \Gamma^{\mathrm{C}})}}{\| \bar{s} \|_{L_2(\partial Q \cap \Gamma^{\mathrm{C}})}} \tag{24}$$

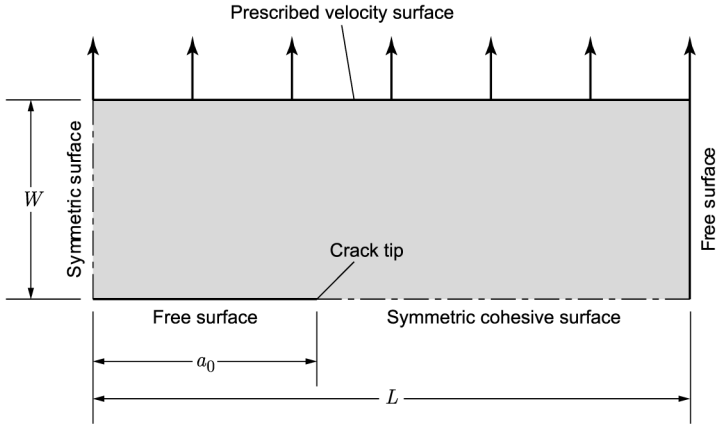


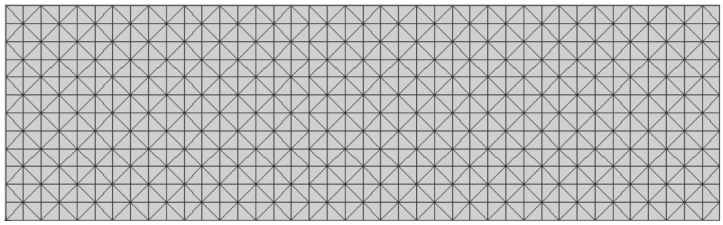
Fig. 4. Model for middle-crack tension specimen.

in which $\bar{s}$ is the cohesive strength parameter in the Xu and Needleman model. Let τ^* be the target value of the cohesive traction error indicator. The cohesive traction error on element Q is acceptable when $\underline{\tau} \leq \tau^{\mathrm{Q}} \leq \overline{\tau}$, where $\underline{\tau} = (1-\eta)\tau^*$ and $\overline{\tau} = (1+\eta)\tau^*$ in which $0 < \eta < 1$. Refinement is required when $\tau^{\mathrm{Q}} > \overline{\tau}$, and element Q might be coarsenable (depending on whether it is also coarsenable with respect to the dissipation error) when $\tau^{\mathrm{Q}} < \underline{\tau}$.

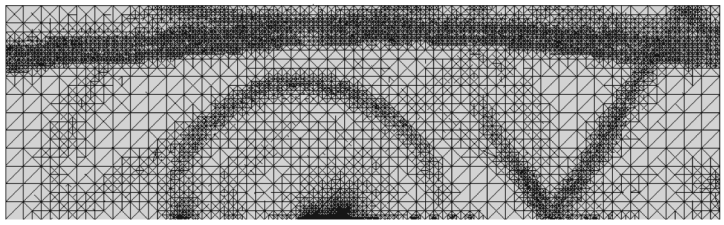
Numerical Example: Cohesive Fracture Under Shock Loading Conditions

We consider cohesive crack growth in a middle-crack tension specimen loaded by a uniform, prescribed velocity along two opposite edges. We use symmetry conditions to model a single quadrant with dimensions, $L = 10$ mm by $W = 3$ mm, and an initial crack length, $a_0 = 4.25$ mm, as shown in Figure 4. The bulk material properties and the parameters for the traction–separation law are identical to those used in [49]. A uniform, prescribed velocity, applied along the top edge of the specimen, ramps from zero to a sustained velocity of 15 m/s over an interval of 0.1 μs. This approximates an elastodynamic shock that reaches the crack tip at $t = 1.44$ μs. We use tetrahedral spacetime elements with complete cubic polynomial bases.

The adaptive SDG model's ability to resolve shocks accurately and to maintain the fidelity of the numerical approximation of the TSL is key to the integrity of this study. Figure 5a shows the spatial discretization that initiates the adaptive solution process. Both adaptive error indicators are active; the target per-element energy dissipation is $\varphi^* = 5 \times 10^{-17}$ J, and the target normalized cohesive traction error is $\tau^* = 1\%$. Figure 5b shows the pattern of adaptive mesh refinement at roughly $t = 3.5$ μs, well after the initial, stationary tip scatters the main shock wave and the crack begins



(a) Initial space mesh



(b) Refinement pattern on mesh front as crack starts to propagate

Fig. 5. Progression of mesh refinement during solution.

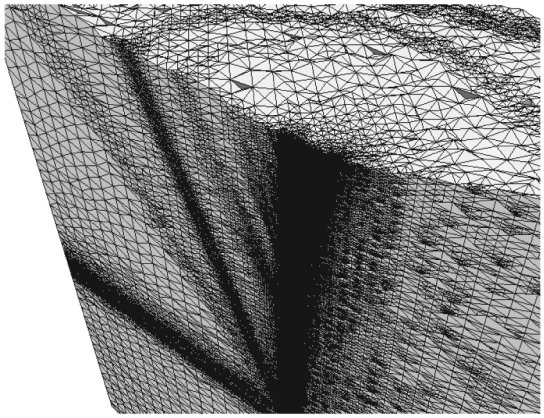


Fig. 6. Detail of spacetime mesh in vicinity of crack-growth-initiation event

to propagate. The pattern of mesh refinement reflects the pattern of shock fronts in the bulk material. The apparent distortion of the wavefronts is due to the fact that the spacetime mesh does not advance uniformly in time; it is not due to solution error. There is a region of refinement along the cohesive interface near the initial tip position where the adaptive solution tracks the moving cohesive process zone. There are strong gradients in the stress field surrounding the crack tip, but the TSL ensures that the cohesive tractions remain bounded.

Figure 6 shows a detail of the portion of the spacetime mesh (the vertical axis represents time) that captures the transtion from a stationary to a running crack. The diagonal bands visible on the front face of the mesh are associated with, from left to right, the dilatational and shear waves scattered by the stationary tip followed by a series of waves emitted by the moving cohesive process zone as it accelerates to the right.

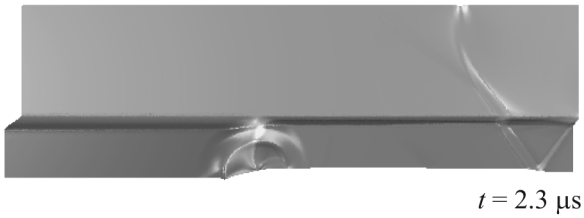
Figure 7 presents three still images from an animation of the SDG solution generated by the per-pixel-accurate spacetime rendering procedure described in [52]; the sequence depicts the transition from a stationary crack to a propagating cohesive failure. The color field (rendered in gray scale here) represents the log of the strain-energy density; the height field depicts the modulus of the material velocity.

Figure 7a shows wave scattering shortly after the shock reaches the crack plane; a sharp gradient in the energy density is evident at the crack tip. The cohesive process zone is stationary, and the response is similar to that of a stationary, mathematically sharp crack. However, as expected, the cohesive model eliminates the singular stress field at the tip. Figure 7b shows the transition to crack propagation. A spike begins to develop in the velocity field near the initial crack-tip position. Post-simulation analysis shows that the velocity field eventually develops the same $r^{-\frac{1}{2}}$ singular structure predicted by linear elastodynamic solutions for running cracks [19], where r is the radial distance to the core of the singularity. The singularity grows in strength and moves with the accelerating crack tip, as shown in Figure 7c. The emergence of the velocity singularity was unexpected, and to our knowledge, has not been predicted by other numerical models for this problem. Our solutions show that the velocity singularity persists through two decades of the radius r. However, in view of the discrete nature of our solutions, it is not clear whether the singular structure persists to arbitrarily small radii. A future publication will present a more detailed investigation of the velocity singularity for running cracks.

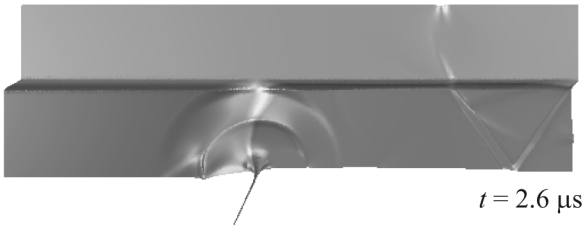
There are also qualitative differences in the crack-growth kinetics predicted by the SDG solution and the Xu and Needleman model. Initiation at 1.6 μs is nearly instantaneous upon the arrival of the primary shock front in [49], and the crack-tip undergoes rapid acceleration to approach the Rayleigh wave speed. The SDG solution, on the other hand, shows a lag of roughly 1 μs between the arrival of the primary shock front and crack-growth initiation, and the subsequent crack-tip acceleration is more gradual. The slower response in the SDG model might be explained by the finite time required to transition from the non-singular velocity field surrounding the stationary crack to the singular form generated by the running crack.

5.3 Front Tracking and Combined Tracking/Capturing Methods

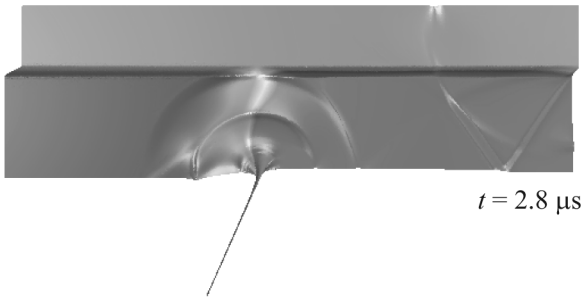
This section presents an example in which we combine two distinct methods for resolving discontinuities. After introducing an expanded set of adaptive



(a) Wave scattering by stationary crack; growth is delayed



(b) Singular velocity field develops as crack growth initiates



(c) Singularity strengthens as crack tip accelerates

Fig. 7. Visualization of crack propagation over time.

meshing operations, we describe a shock-capturing technique for nonlinear conservation laws as well as an interface tracking method that delivers sharp resolution of a moving contact discontinuity by aligning the spacetime mesh with the trajectory of the singular surface.

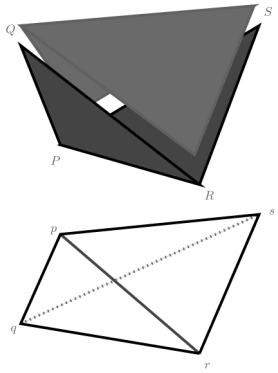


Fig. 8. Edge-flip operation implemented in spacetime (top) vs. instantaneously (bottom)

New Adaptive Meshing Operations

The adaptive meshing procedures used in the preceding examples, although effective, relied on a limited number of meshing operations. We performed mesh refinement instantaneously by subdividing triangles in the space-like mesh front according to a newest vertex scheme. Mesh coarsening required a preliminary step wherein a patch of triangles to be coarsened must first be pitched to a coplanar spacetime configuration prior to each vertex deletion. No provision was made for the spacetime analogues of edge-flips and vertex smoothing — operations that maintain and improve mesh quality in conventional adaptive algorithms. An expanded, more robust set of spacetime adaptive meshing procedures in the current implementation of Tent Pitcher improves solution accuracy and provides new capabilities for tracking discontinuous and non-smooth solution features.

In the previous implementation, the "tent poles" generated by advancing a vertex in time were constrained to strictly vertical orientations (parallel to the time coordinate axis). The current implementation accommodates inclined tent poles, such that the spatial position of a vertex can vary continuously as it advances in time, while ensuring that all patches satisfy the causality and progress contraints required by our $\mathcal{O}(N)$ solution scheme. We use this new capability to carry out smoothing operations on the spatial projection of the mesh front to improve and maintain element quality. That is, we compute a smoothed spatial position for each unconstrained vertex in the space mesh as an area-weighted average of the centroids of the surrounding triangles. An inclined tent pole then moves the vertex from its previous position to the smoothed location. Smoothing is applied every time an unconstrained vertex is pitched, so good mesh quality is maintained throughout the spacetime meshing process. Inclined tent poles are also important in the interface-tracking method described below.

All adaptive operations other than mesh refinement,[6] including edge flips, smoothing and vertex deletions, are now implemented over spacetime patches in a manner that eliminates projection errors associated with remeshing. For example, we use the spacetime edge-flip operation, depicted in Fig. 8, to maintain the Delaunay property in the spatial projection of the mesh front. In the spacetime version, the dark gray inflow triangles, PQR and PRS, of the spacetime tetrahedron $PQRS$ coincide with outflow faces of previously solved elements, so no projection error is incurred. Thus, the edge flip from PR to QS, can be executed with zero projection error. If this operation is carried out instantaneously, as in conventional remeshing procedures, there is unavoidable projection error incurred by switching from piecewise polynomials defined over triangles pqr and prs to polynomials defined on triangles qsp and qrs. Overall, the new set of spacetime meshing operations increases solution accuracy while reducing the number of elements in an adapted spacetime mesh.

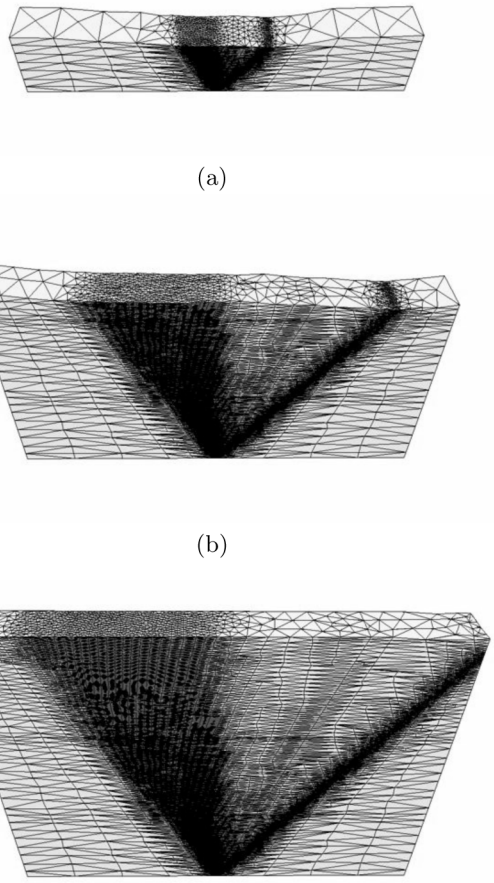
Tracking Moving Interfaces

Discontinuous Galerkin methods in which the mesh explicitly tracks evolving discontinuities present an attractive alternative, where feasible, to discontinuity-capturing methods. If element boundaries are aligned with the trajectories of singular surfaces, then the SDG basis functions can model discontinuous solutions directly, including explicit treatment of the relevant jump conditions, without the expensive mesh refinement required in capturing methods. Further, the need for extra stabilization in the vicinity of a discontinuity can be reduced or eliminated. The expanded set of spacetime meshing operations provides a framework for a new approach to interface tracking. We use inclined tent poles to generate patches with internal element boundaries that track the singular surface. For example, we can use a zero-mass-flux condition to determine the proper tent-pole inclination to track a material interface in a heterogeneous flow, or we can use a constutive kinetics equation to track the motion of a phase boundary in a solid. Adaptive smoothing, edge-flip, refinement and vertex-deletion operations maintain the integrity and resolution of the mesh surrounding the moving interface. All these operations are local and are carried out continuously as the simulation proceeds. We thus avoid the mesh distortion and the projection errors associated with global remeshing operations in many other interface tracking methods.

Numerical Example: Combined Interface Tracking and Shock Capturing

Figs. 9 and 10 illustrate a shock-tube simulation based on the inviscid Euler equations (cf. Subsection 3.2) in which we use a combination of techniques

[6] Mesh refinement is still carried out instantaneously on the mesh front. However, this is not problematic because refinement does not induce any projection error and does not require a preliminary synchronization step.



(a)

(b)

(c)

Fig. 9. Shock tube example. Adaptive spacetime mesh tracks trajectory of contact discontinuity moving to right at half the shock speed

to model the discontinuous solution features. Although the basic SDG formulation features some intrinsic upwinding, the nonlinear structure of this problem requires some additional stabilization to control overshoot and undershoot. We use uniform artificial viscosity and piecewise linear polynomials in this preliminary study. Better resolution of the shocks would be possible using a shock-capturing scheme and higher-order polynomials. However, since testing the SDG approach to interface tracking is the main objective of this study, we defer these improvements to future work. We use adaptive spacetime refinement to capture a shock moving from the domain center to the

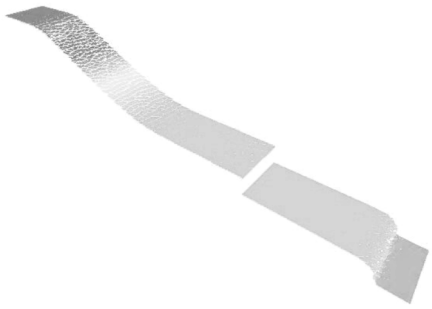


Fig. 10. SDG solution for shock tube example; height and color fields (rendered in gray scale here) indicate density and pressure. Interface tracking delivers sharp resolution of contact discontinuity.

right and an expansion wave opening to the left. Regions of higher density in the spacetime meshes shown in Fig. 9 identify each feature.

Three residual-based error indicators (one for each conservation equation) in each spacetime element Q drive this adaptive simulation. The error indicators share a common form, given by

$$e_i = \frac{\|\mathbf{dF}_i\|_{L_1(Q)} + \|\mathbf{F}_i - \mathbf{F}_i^*\|_{L_1(\partial Q)}}{\|\rho_i^0\|_{L_1(\Omega^0)}}; \quad i = 1, 2, 3 \tag{25}$$

where Ω^0 is the spatial analysis domain at the initial time and

$$\mathbf{F}_1 := \mathbf{F}_\rho \quad \rho_1^0 = \rho^0 \tag{26a}$$

$$\mathbf{F}_2 := \mathbf{F}_\mathbf{p} \quad \rho_2^0 = \sqrt{\rho^0 \mathcal{E}^0} \tag{26b}$$

$$\mathbf{F}_3 := \mathbf{F}_\mathcal{E} \quad \rho_3^0 = \mathcal{E}^0 \tag{26c}$$

in which ρ^0 and $\mathcal{E}^0$ are the mass density and total energy density fields on Ω^0. Adaptive refinement is triggered when any of the three error indicators exceeds its maximum allowable value in any element in a patch; adaptive coarsening is triggered when all three error indicators fall below a minimum threshold in all the elements in a patch. As seen in Fig. 9, this approach effectively limits spacetime mesh refinement to the trajectories of the shock and the expansion wave. However, as with any capturing method and especially due to the crude form of the stabilization used here, the shock appears smeared out in Fig. 10. Nonetheless, the conservation properties of the SDG model ensures that the shock moves at the correct velocity.

We use a different approach, *interface tracking*, to resolve the contact discontinuity in the density field. This feature appears as a coherent surface in the spacetime mesh midway between the center of the domain and the right-traveling shock; it is identified by a set of spacetime element faces that precisely cover the trajectory of the contact discontinuity, rather than by mesh

refinement. We used the new capability to mesh with inclined tent poles to align the spacetime mesh with the motion of the discontinuity. Since the singular surface aligns with element boundaries in the spacetime mesh, the SDG formulation weakly enforces the correct set of jump conditions at the contact discontinuity. Notably, the contact discontinuity is captured as a true jump in the SDG solution (Fig. 10) — without mesh refinement and with no extra stabilization active beyond the inherent stability of the SDG formulation. Thus, we obtain a more accurate resolution of the contact discontinuity at substantially lower cost than would be required if a capturing technique were applied to this feature.

We used *a priori* knowledge of the motion of the contact discontinuity to construct the spacetime mesh in this proof-of-concept study. However, the motion of discontinuities is often solution-dependent, so a more robust approach is generally required. We are currently testing a more robust approach, in which the inclination of the tent pole in each patch is iteratively adjusted to satisfy the zero-mass-flux condition that governs the motion of the contact discontinuity. We plan to use this method and related techniques in continuing work to track moving interfaces in materials microstructures and to follow crack growth along solution-dependent paths.

6 Conclusions

We have demonstrated three distinct approaches to discretizing evolving discontinuous solution features using SDG solution methods. The first option, h-adaptive procedures for capturing discontinuities, mitigates the cost of the necessary grid refinement by adapting the mesh simultaneously in space and time, thereby avoiding the expense imposed by a global uniform time step dictated by the smallest elements in a spatially adapted discretization. The second approach, the incorporation of a cohesive damage model in the SDG framework, is facilitated by the discontinuous SDG solution spaces and the treatment of jump conditions that naturally accommodate the enforcement of the traction–separation law. We included an adaptive error indicator that directly ensures the accuracy of the finite element approximation of the TSL. The high resolution obtained with this model led to the discovery that the material velocity exhibits a strong singular response for running cracks with the same $r^{-\frac{1}{2}}$ structure observed in classical elastodynamic fracture. We also presented a preliminary study of how the use of inclined tent poles and a more robust set of adaptive meshing procedures can be used to track moving discontinuities. This approach uses the discontinuous nature of SDG solution spaces to eliminate the need for strong mesh refinement near the discontinuity, to enforce the correct jump conditions dictated by the governing balance laws, and to render a sharp resolution of the discontinuity. It's worth noting that our software framework allows us to combine more than one of these

techniques to address different classes of evolving discontinuities in a single computation.

Execution times for the numerical examples reported here ranged from several hours to several days of serial processing on desk-top processors; the longest were for the highly refined multi-scale model that resolves the singular velocity field at a moving cohesive crack tip. These relatively long execution times are for a non-optimized C++ research code and are not representative of the true cost of the SDG method. Recent improvements to our serial code yielded substantial speed-ups in the range of $30 - 40$ times faster. The optimizations included use of the Blitz++ library [47] to mitigate some of the inefficiencies of array storage in C++, more effective use of standard math libraries, use of more aggressive compiler optimizations, and straightforward eliminations of redundant calculations. Additional serial speed-ups are expected as we continue to optimize the code. Parallel implementation is another promising direction to improve performance. We obtain high efficiency and nearly linear speed-ups in the number of parallel processors for non-adaptive computations. The main challenge for an effective adaptive parallel implementation is maintaining load balance in the face of intense and evolving adaptive refinement. We believe that the local nature of our patch-by-patch solution scheme will prove useful in on-going research that addresses load balancing in parallel–adaptive SDG computations.

We continue to develop the underlying SDG formulation and to extend the technologies for discontinuity capturing reported in this work. One area of continuing effort involves improved methods for SDG shock capturing to control overshoot in nonlinear conservation laws. We seek methods that are suitable for spacetime implementation, that do not interfere with the element-wise conservation property, that preserve the compact form of the SDG stencil, and that introduce as little artificial viscosity as possible. SDG variants of sub-cell shock capturing strategies [39] appear particularly promising at this time. We continue to develop the SDG technology for tracking moving interfaces and discontinuities. One such improvement involves the use of interface kinetics models to track solution-dependent motions of interfaces and discontinuities. Another application is the use of the interface tracking technology to track cohesive crack growth along arbitrary, solution-dependent paths (vs. the predetermined paths in our current studies). Extending our h-adaptive technology to hp enrichments is another promising direction for development.

Acknowledgements

The authors gratefully acknowledge the contributions of Jeff Erickson, Michael Garland, Shripad Thite and Yuan Zhou to this work. Support from the Center for Process Simulation and Design (CPSD) and the Center for Simulation of Advanced Rockets (CSAR) at the University of Illinois at Urbana-Champaign is gratefully acknowledged. The U.S. National Science Foundation supports

research in CPSD *via* grant NSF DMR 01-21695. The CSAR research program is supported by the U.S. Department of Energy through the University of California under subcontract B341494.

References

1. Abedi R, Chung S-H, Erickson J, Fan Y, Garland M, Guoy D, Haber R, Sullivan J, Thite S, and Zhou Y (2004) Proc 20th Ann ACM Symp on Comp Geometry 300–309
2. Abedi R, Haber RB, Petracovici B (2006) Comp Methods Appl Mechs Engrg 195: 3247–3273.
3. Abedi R, Haber RB, Thite S, Erickson J (2006) Rev Europ Mécanique Num 15:619–642
4. Babuška I, Melenk I (1997) Int J Num Methods Engrg 40:727–758
5. Barenblatt GI (1962) Adv Appl Mechs 7:55–129.
6. Belytschko T, and Black T (1999) Int J Num Methods Engrg 45:601–620
7. Bishop RL, Goldberg SI (1980) Tensor analysis on manifolds. Prentice Hall (reprinted by Dover), Englewood Cliffs, New Jersey
8. Brezzi F, Bristeau M-O, Franca L, Mallet M, Rogé G (1992) Comp Methods Appl Mechs Engrg 96:117–129
9. Brooks AN, Hughes TJR (1982) Comp Methods Appl Mechs Engrg 32:199–259
10. Camacho GT, Ortiz M (1996) Int J Solids Structs 33:2899–2938
11. Cockburn B, Gremaud P-A (1996) SIAM J Num Analysis 33:522–554
12. Cockburn B, Karniadakis GE, Shu CW (eds.) (2000) Discontinuous Galerkin methods -theory, computation and applications, Lecture Notes in Computational Science and Engineering 11, Springer-Verlag, New York
13. Cornec A, Scheider I, Schwalbe K-H (2003) Engrg Fracture Mechs 70:1963–1987
14. de Borst R, Remmers JJC, Needleman A (2006) Engrg Fracture Mechs 73:160–177
15. Dugdale DS (1960) J Mechs Phys Solids 8:100–104
16. Erickson J, Guoy D, Sullivan J, Üngör A (2002) Proc 11th Int Meshing Roundtable (Sandia National Laboratories) 391–402
17. Fedkiw RP, Aslam T, Merriman B, Osher S (1999) J. Comp. Physics 152:457–492
18. Fleming WH (1964) Functions of several variables, Addison-Wesley, Reading,Massachusetts
19. Freund LB (1990) Dynamic Fracture Mechanics, Cambridge University Press, Cambridge, UK
20. Geubelle PH, Baylor J, (1998) Composites 29B:589–602
21. Huang H, Costanzo F (2004) Int J Fracture 127:119–146
22. Hughes TJR, Liu WK, Zimmermann TK (1981) Comp Methods Appl Mechs Engnrg 29:329–349
23. Hughes TJR, Franca LP, Hulbert GM (1989) Comp Methods Appl Mechs Engnrg 73:173–189
24. Hughes TJR, Mallet M (1986) Comp Methods Appl Mechs Engnrg 58:305–328
25. Jaffré J, Johnson C, Szepessy A (1995) Math Models Methods Appl Sci 5:367–386

26. Johnson C, Szepessy A (1987) Math Comp 49:427–444
27. Karamanos G-S, Karniadakis GE (2000) J Comp Physics 163:22–50
28. Krysl P, Belytschko T (1999) Int J Num Methods Engrg 44:767–800
29. Lowrie RB, Roe PL, van Leer B (1998) in Barriers and Challenges in Computational Fluid Dynamics, ICASE/LaRC Interdisciplinary Series in Science and Engineering, Natl Inst Aerospace 6:79–98
30. Luo H, Baum JD, Löhner R (2004) J Comp Physics 194:304–328
31. Mergheim J, Kuhl E, Steinmann P (2005) Int J Num Methods Engrg 63:276–289
32. Moës N, Belytschko T (2002) Eng Fracture Mechs 69:813–833
33. Moës N, Dolbow J, Belytschko T (1999) Int J Num Methods Engrg 46:131–150
34. Needleman A (1987) J Appl Mechs 54:525–532
35. Ortiz M, Pandolfi A (1999) Int J Num Methods Engrg 44:1267–1282
36. Osher S, Sethian JA (1988) J Comp Physics 79:12–49
37. Palaniappan J, Haber RB, Jerrard RL (2004) Comp Methods Appl Mechs Engnrg 193:3607–3631
38. Pandolfi A, Ortiz M (2002) Engnrg Computers 18:148–159
39. Persson P-P, Peraire J (2006) 44th AIAA Aerospace Sci Mtg Exhibit, AIAA, AIAA–2006–0012
40. Richter GR (1994) Appl Num Math 16:65–80
41. Schrefler BA, Secchi S, Simoni L (2006) Comp Methods Appl Mechs Engrg 195:444–461
42. Shakib F, Hughes TJR, Johan A (1991) Comp Methods Appl Mechs Engrg 89:141–219
43. Spivak M (1965) Calculus on manifolds, W. A. Benjamin, New York
44. Tadmor E (1989) SIAM J Num Analysis 26:30–44
45. Tvergaard V, Hutchinson JW (1992) J Mechs Phys Solids 40:1377–1397
46. Üngör A, Sheffer A (2002) Int J Foundations Comp Sci 13: 201–221
47. Veldhuizen TL (1998) Arrays in Blitz++. In: Caromel D, Oldehoeft RR, Tholburn M (eds) Proc 2nd Int Symp on Scientific Computing in Object-Oriented Parallel Environments (ISCOPE'98), Lecture Notes in Computer Science 1505, Springer-Verlag, London UK
48. Wells GN, Sluys LJ (2001) Int J Num Methods Engrg 50:2667–2682
49. Xu X-P, Needleman A (1994) J Mechs Phys Solids 42:1397–1434
50. Yin L, Acharya A, Sobh N, Haber RB, and Tortorelli DA (2000) A spacetime discontinuous Galerkin method for elastodynamics analysis. In: Cockburn B, Karniadakis G, Shu CW (eds.) Discontinuous Galerkin methods: Theory, computation and applications, Lecture Notes in Computational Science and Engineering 11, Springer-Verlag, Berlin
51. Zhou F, Molinari J-F, Shioya T (2005) Engrg Fracture Mechs 72:1383–1410
52. Zhou Y, Garland M, Haber RB (2004) Pixel-exact rendering of spacetime finite element solutions, In: Proc IEEE Visualization 2004, IEEE 425–432
53. Zi G, Belytschko T (2003) Int J Num Methods Engrg 57:2221–2240.