

Discontinuous Galerkin Methods Applied to Shock and Blast Problems

N. Chevaugnon,¹ J. Xin,¹ P. Hu,¹ X. Li,¹ D. Cler,² J. E. Flaherty,¹
and M. S. Shephard¹

Received January 12, 2004; accepted (in revised form) May 5, 2004

We describe procedures to model transient shock interaction problems using discontinuous Galerkin methods to solve the compressible Euler equations. The problems are motivated by blast flows surrounding cannons with perforated muzzle brakes. The goal is to predict shock strengths and blast over pressure. This application illustrates several computational difficulties. The software must handle complex geometries. The problems feature strong interacting shocks, with pressure ratios on the order of 1000 as well as weaker precursor shocks traveling rearward that also must be accurately captured. These aspects are addressed using anisotropic mesh adaptation. A shock detector is used to control the adaptation and limiting. We also describe procedures to track projectile motion in the flow by a level-set procedure.

KEY WORDS: Discontinuous Galerkin methods; hyperbolic conservation problems; blast; shock detection; anisotropic mesh refinement; adaptive methods.

1. INTRODUCTION

We propose to solve blast problems motivated by flows surrounding cannons with perforated muzzle brakes using a discontinuous Galerkin method (DGM), as described in Sec. 2. Detection of shock structures appears crucial to drive both adaptive mesh refinement and limiting and we describe such detection scheme (Sec. 3) and present some one-dimensional examples. We also describe an anisotropic adaptive mesh refinement procedure (Sec. 4), that allows us to solve large problems, in an efficient

¹ Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY 12180, USA. E-mail: flahej@rpi.edu

² Benet Laboratories, TACOM/ARDEC, U. S. Army, Watervliet, NY 12189, USA.

manner. We apply this to two examples involving blast simulation. In Sec. 5, we describe progress to include the presence of solid objects moving in the flow field by using a level-set/ghost-fluid approach, and present some applications.

2. DISCONTINUOUS GALERKIN METHODS FOR EULER EQUATIONS

The discontinuous Galerkin method was introduced by Reed and Hill in 1973 [23] as a technique to solve neutron transport problems. Lesaint and Raviart [18] presented the first numerical analysis of the method for a linear advection equation. However, the technique lay dormant for several years and has only recently become popular as a method for solving fluid dynamics and electromagnetic problems [6]. As with all mesh based procedures, DGM uses a double discretization. First, the physical domain Ω is discretized into a collection of \mathcal{N}_e elements

$$\mathcal{T}_e = \bigcup_{e=1}^{\mathcal{N}_e} e \quad (2.1)$$

called a mesh. Then, the function space $V(\Omega)$ containing the solution of the problem is approximated on each element e of the mesh by a finite-dimensional space $V_e(\mathcal{T}_e)$.

The accuracy of the double discretization depends on geometrical and functional discretization. Classical Galerkin finite element approximation uses conforming meshes where elements share only complete boundary segments. The space V_e is also constrained to be a subspace of a continuous function space, for example, H^1 , with a basis that is typically associated with element vertices, edges, faces, or interiors. These simplify the imposition of the C^0 continuity requirements of H^1 but limit choices. The DGM allows more general mesh configurations and discontinuous bases that simplify both h- and p-refinement. For example, non-conforming meshes and arbitrary bases for functional approximation [32] may be used. Herein, we use a L^2 -orthogonal basis as a product of Joacobi polynomials [8] that yields a diagonal mass matrix [24].

The DGM can easily support arbitrary orders of spatial discretization accuracy without the need to construct complex stencils for high-order reconstruction [24]. Indeed, the DGM stencil remains invariant for all polynomial degrees. This greatly simplifies parallel implementation. The DGM has additional flexibility in specifying fluxes across elements faces and permits the use of fluxes based on exact or approximate solutions of Riemann problems [5].

We apply the DGM to hyperbolic conservation laws (Sec. 2.1) using a spatial discretization of p -degree orthogonal polynomials. Time discretization is performed by an explicit total variation diminishing Runge–Kutta method [14]. To improve the performance of the explicit integration on irregular and unstructured grid, we use a new local time stepping procedure [11,26].

2.1. Discontinuous Finite Element Formulation for Conservation Laws

Consider an open set $\Omega \subset \mathbb{R}^3$ with boundary $\partial\Omega$. We seek to determine $\mathbf{u}(\Omega, t): \mathbb{R}^3 \times \mathbb{R} \rightarrow L^2(\Omega, t)^m = V(\Omega, t)$ as the solution of a system of m conservation laws

$$\partial_t \mathbf{u} + \mathbf{div} \vec{\mathbf{F}}(\mathbf{u}) = \mathbf{r}. \quad (2.2)$$

Here $\mathbf{div} = (\text{div}, \dots, \text{div})$ is the vector valued divergence operator and

$$\vec{\mathbf{F}}(\mathbf{u}) = (\vec{F}_1(\mathbf{u}), \dots, \vec{F}_m(\mathbf{u})) \quad (2.3)$$

is the flux vector with the i th component $\vec{F}_i(\mathbf{u}): (\mathbf{H}^1(\Omega))^m \rightarrow \mathbf{H}(\text{div}, \Omega)$, where $\mathbf{H}(\text{div}, \Omega)$ consists of square integrable vector valued functions whose divergence is also square integrable, that is,

$$\mathbf{H}(\text{div}, \Omega) = \left\{ \vec{v} \mid \vec{v} \in L^2(\Omega^3), \text{div} \vec{v} \in L^2(\Omega) \right\}. \quad (2.4)$$

With the aim of constructing a Galerkin form of Eq. (2.2), let $(\cdot, \cdot)_\Omega$ and $\langle \cdot, \cdot \rangle_{\partial\Omega}$, respectively, denote the standard $L^2(\Omega)$ and $L^2(\partial\Omega)$ scalar products

$$(\mathbf{u}, \mathbf{w})_\Omega = \int_\Omega \mathbf{u} \mathbf{w} \, d\omega \quad (2.5)$$

and

$$\langle \mathbf{u}, \mathbf{w} \rangle_{\partial\Omega} = \int_{\partial\Omega} \mathbf{u} \mathbf{w} \, d\sigma. \quad (2.6)$$

Multiply Eq. (2.2) by a test function $\mathbf{w} \in V(\Omega)$, integrate over Ω and use the divergence theorem to obtain the variational formulation

$$(\partial_t \mathbf{u}, \mathbf{w})_\Omega - (\vec{\mathbf{F}}(\mathbf{u}), \mathbf{grad} \mathbf{w})_\Omega + (\vec{\mathbf{F}}(\mathbf{u}) \cdot \vec{n}, \mathbf{w})_{\partial\Omega} = (\mathbf{r}, \mathbf{w})_\Omega, \quad \forall \mathbf{w} \in V(\Omega). \quad (2.7)$$

With the DGM, V_e is a ‘‘broken’’ function space that consists of the direct sum of elementary approximations \mathbf{u}_e

$$\begin{aligned} V_e(\mathcal{T}_e) &= \left\{ \mathbf{u} \mid \mathbf{u} \in L^2(\Omega)^m, \text{restriction}(\mathbf{u})|_e \right. \\ &= \left. \mathbf{u}_e \in \mathbb{P}^p(e)^m = V_e(e), \forall e \in \mathcal{T}_e \right\}, \end{aligned} \quad (2.8)$$

where $\mathbb{P}^p(e)^m$ is a polynomial of degree p on element e . Because all approximations are disconnected, we can solve on each element to obtain

$$(\partial_t \mathbf{u}_e, \mathbf{w})_e - (\vec{\mathbf{F}}(\mathbf{u}_e), \mathbf{grad} \mathbf{w})_e + \langle \mathbf{F}_n, \mathbf{w} \rangle_{\partial e} = (\mathbf{r}, \mathbf{w})_e, \quad \forall \mathbf{w} \in V_e(e). \quad (2.9)$$

A discontinuous basis implies that the normal flux $\mathbf{F}_n = \vec{\mathbf{F}}(\mathbf{u}) \cdot \vec{n}$, where \vec{n} is the unit outer normal to ∂e , is not defined on ∂e . In this situation, a ‘‘numerical flux’’ $\mathbf{F}_n(\mathbf{u}_e, \mathbf{u}_{e_k})$ is used on each portion ∂e_k of ∂e shared by element e and neighboring element e_k . The numerical flux must be continuous, so $\mathbf{F}_n \in \mathbf{H}(\text{div}, \Omega)^m$, and consistent, so $\mathbf{F}_n(\mathbf{u}, \mathbf{u}) = \vec{\mathbf{F}}(\mathbf{u}) \cdot \vec{n}$. With such a numerical flux, Eq. (2.9) becomes

$$\begin{aligned} (\partial_t \mathbf{u}_e, \mathbf{w})_e - (\vec{\mathbf{F}}(\mathbf{u}_e), \mathbf{grad} \mathbf{w})_e \\ + \sum_{k=1}^{n_e} \langle \mathbf{F}_n(\mathbf{u}_e, \mathbf{u}_{e_k}), \mathbf{w} \rangle_{\partial e_k} = (\mathbf{r}, \mathbf{w})_e, \quad \forall \mathbf{w} \in V_e(e), \end{aligned} \quad (2.10)$$

where n_e is the number of faces of element e . Only the normal traces have to be defined on ∂e_k and several options are possible (see, e.g., Refs. [31, 33]). It is usual to define the trace as the solution of a Riemann problem across ∂e_k . Herein, when we consider problems with strong shocks [7, 33], an exact Riemann solver is used to compute the numerical fluxes and a moment limiter [2] is used to restrict spurious oscillations when polynomial degrees $p > 0$ are used. The moment limiter is applied on each element after a new solution field is computed, so that the minimal and maximal value of the solution moments in one element are bounded by their mean values on neighboring elements.

2.2. Euler’s Equations

Flows satisfy the compressible Euler equations, which, in two dimensions, have the form (2.2) with

$$\mathbf{u} = [\rho, \rho u, \rho v, E]^T \quad (2.11)$$

where ρ is the density of the fluid, \vec{v} is the velocity with Cartesian components u and v and E is the internal energy. The flux is

$$\vec{\mathbf{F}}(\mathbf{u}) = \left[\vec{F}_1, \vec{F}_2, \vec{F}_3, \vec{F}_4 \right] = \left[\rho \vec{v}, \rho u \vec{v} + P \vec{e}_x, \rho v \vec{v} + P \vec{e}_y, (\rho E + P) \vec{v} \right] \quad (2.12)$$

where P is the pressure and \vec{e}_x and \vec{e}_y are unit vectors in the x and y coordinate directions. An equation of state (EOS) $P = P(\rho, E)$ is necessary to close the system. The DGM software can be used with any physically consistent EOS and we choose the perfect gas law.

$$P = (\gamma - 1) \rho \left[E - \frac{\|\vec{v}\|^2}{2} \right] \quad (2.13)$$

with gas constant $\gamma = 1.4$.

3. DISCONTINUITY DETECTION

A discontinuity detection procedure automatically detects regions of the flow field where the solution is discontinuous. For the Euler equations, discontinuities may be contact discontinuities, where the density is discontinuous; shock waves, where the density, velocity, and pressure are discontinuous; and expansion wave where flow gradients are discontinuous. A knowledge of discontinuity locations allows us to restrict the limiting procedure to the vicinity of discontinuities, which both saves time and improves solution accuracy in smooth flow regions. Discontinuity detection is also used to drive the anisotropic adaptation procedure (Sec. 4).

To define a discontinuity detection strategy, we partition an element boundary ∂e into portions ∂e^- and ∂e^+ where, respectively, the flow is into ($\vec{v} \cdot \vec{n} < 0$) and out of ($\vec{v} \cdot \vec{n} > 0$) element e .

Let q be a flow variable or a derived quantity from the flow field, for example, density or entropy, for the Euler equations, and Q_e be the DG value of q on element e . Consider the jump in Q_e across the inflow edges (faces) of e and examine

$$I_e = \int_{\partial e^-} (Q_e - Q_{e_k}) d\sigma = \int_{\partial e^-} (Q_e - q) d\sigma + \sum_{k=1}^{n_e} \int_{\partial e_k^+} (q - Q_{e_k}) d\sigma. \quad (3.1)$$

In smooth flow regions, the first integral on the right is $\mathcal{O}(h^{p+2})$, while the second might be $\mathcal{O}(h^{2(p+1)})$ [1, 15, 16] due to super-convergence. Here, h is a measure of an element size, as defined below. Thus, in smooth regions,

$I_e = \mathcal{O}(h^{p+2})$. If q is discontinuous in the immediate vicinity of ∂e , then either or both of $q - Q_e$ and $q - Q_{e_k}$ are $\mathcal{O}(1)$; hence,

$$I_e = \begin{cases} \mathcal{O}(h^{p+2}), & \text{if } q|_{\partial e} \text{ is smooth,} \\ \mathcal{O}(h), & \text{if } q|_{\partial e} \text{ is discontinuous.} \end{cases} \tag{3.2}$$

We construct the detector by normalizing I_e relative to an ‘‘average’’ $\mathcal{O}(h^{(p+1)/2})$ convergence rate and the solution on e to obtain

$$\mathcal{I}_e = \frac{|\int_{\partial e^-} (Q_e - Q_{e_k}) d\sigma|}{h^{(p+1)/2} |\partial e^-| \|Q_e\|}. \tag{3.3}$$

In examples, we choose h as the radius of the circumscribed circle in element e , and use a maximum norm based on local solution maxima at integration points in two dimensions and an element average in one dimension.

Using (3.2), $\mathcal{I}_e \rightarrow 0$ as either $h \rightarrow 0$ or $p \rightarrow \infty$ in smooth solution regions, whereas $\mathcal{I}_e \rightarrow \infty$ near discontinuities. Thus, a detection scheme is

$$\begin{cases} \text{if } \mathcal{I}_e > 1, & q \text{ is discontinuous,} \\ \text{if } \mathcal{I}_e < 1, & q \text{ is smooth.} \end{cases} \tag{3.4}$$

Example 3.1. We apply the discontinuity detection scheme [Eqs. (3.3) and (3.4)] to a blast problem [33] for the one-dimensional Euler Eq. [(2.2), (2.11)–(2.13)] with $v=0$] subject to the initial data

$$(\rho, u, P)(x, 0) = \begin{cases} (1, 0, 1000), & 0 \leq x < 0.1, \\ (1, 0, 0.01), & 0.1 \leq x < 0.9, \\ (1, 0, 100), & 0.9 \leq x \leq 1. \end{cases} \tag{3.5}$$

The solution involves multiple interactions of strong shocks, expansions, and contact discontinuities. The most important interactions take place in a short period prior to $t=0.04$. An exact solution of this problem is not known, so, following Woodward and Colella [33], comparisons were done relative to a numerical solution computed with $\mathcal{N}_e=3072$ uniform elements and piecewise polynomials of degree $p=2$. We present solutions for the density (Fig. 1) with $\mathcal{N}_e=800$, $p=2$, using entropy as the quantity q for the shock detection scheme and moment limiting [2] in regions where discontinuities were detected.

We see that the shocks have been identified and limiting is confined to these regions. Comparing the numerical and ‘‘exact’’ solutions, we see that all the important features of the various interactions have been sharply captured.

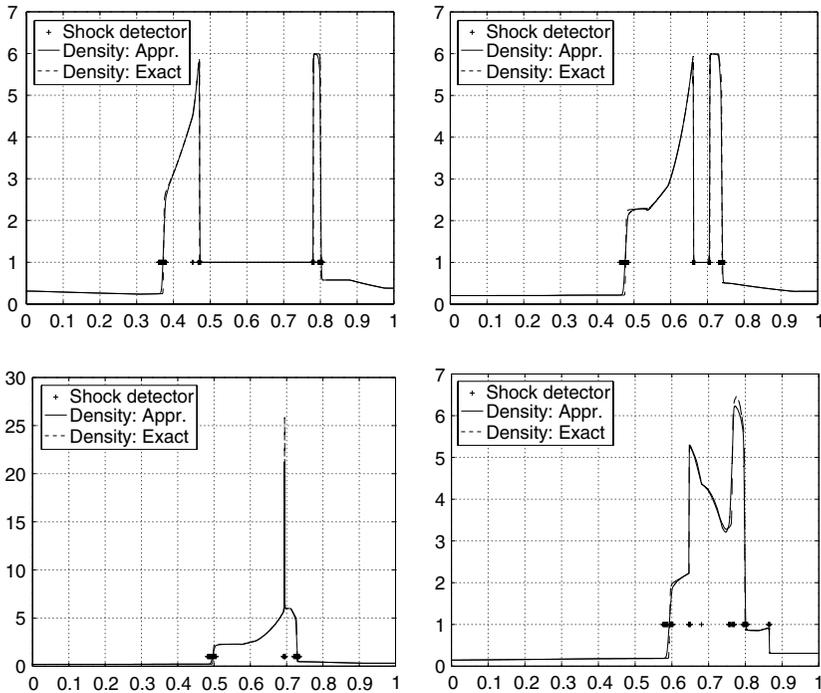


Fig. 1. Density for Example 3.1 at $t=0.016$ (top left), 0.026 (top right), 0.028 (bottom left), and 0.038 (bottom right) for computations performed with $p=2$, $\mathcal{N}_e=800$. Elements where discontinuity is detected and moment limiting is applied are marked with a + sign at unity.

4. MESH ADAPTATION

Our software is able to perform hierarchical mesh refinement where elements are refined using an isotropic refinement template. Since interelement continuity constraints are not needed with the DGM, element refinement need not consider neighboring cells. The level of mesh refinement of a cell is computed using an error indicator [24] based on the value of the jump in solutions between neighboring cells. Coarsening is equally simple, by reversing refinement after projecting the solution from a finer mesh to the coarser parent mesh.

Example 4.1. To illustrate this process, consider a simple scalar advection equation in two dimensions on a unit square. The square is partitioned by an unstructured mesh of triangles with edge length of the order of approximately 0.1. The initial density is

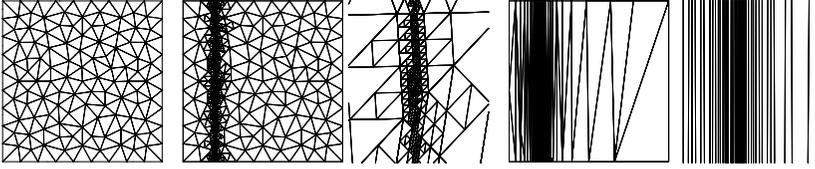


Fig. 2. Meshes for Example 4.1. From left to right: initial mesh, non-conforming isotropic mesh at $t=0.6$, enlargement of this mesh near the discontinuity, anisotropic conforming mesh at $t=0.6$, and enlargement of this mesh near the discontinuity.

$$\rho = \begin{cases} 1, & \text{if } 0 \leq x < 0.8, \\ 2, & \text{if } 0.8 \leq x \leq 1. \end{cases} \quad (4.1)$$

This initial field is advected by a unit velocity in the $-x$ direction. The boundary conditions are reflective at $y=0, 1$, and prescribed at $x=1$. With six levels of adaptive mesh refinement, the minimum edge has a size of order $h_0=0.1/(2^6)=0.0015625$. The solution has the same accuracy as one on a uniform mesh of 819200 elements in two dimensions. Figure 2 displays the initial mesh and the adaptive mesh at $t=0.6$ with six levels of mesh adaptation. The adapted mesh has 6097 elements, which is 134 times less than needed by a uniform mesh with the same resolution. This gain in degrees of freedom has to be weighted against the cost of adapting the mesh with the evolving solution.

Isotropic mesh refinement does not account for situations, such as the one at hand, where refinement is only needed in one direction. Thus, an anisotropic mesh refinement scheme has the potential to greatly improve performance [12].

Our anisotropic mesh refinement procedure adapts a conforming mesh using local mesh modification operations such as element subdivision, edge collapsing, and edge and face swapping. A complete description of this mesh adaptation procedure appears in Refs. [19,20].

The anisotropic mesh adaptation procedure requires the definition of an anisotropic mesh size field. The natural way to represent an anisotropic size field is to define a symmetric positive definite tensor field or a metric. Given any point x in the size field domain, and a direction \vec{a} , the size field describes the size l of a mesh edge in the given direction. The mesh adaptation procedure uses a definition of the metric at each vertex and assumes a linear variation of the size field along element edges.

Building an anisotropic size field requires directional error information. For piecewise linear ($p=1$) approximations, this may be based on the Hessian of the field. The principal directions of the Hessian are then the principal direction of the metric field. The eigenvalues of the Hessian

field are used to define the eigenvalues of the metric field by using a normalization, obtained by selecting a small mesh size in directions where the Hessian is large.

We compute an approximation of the Hessian of the field using a procedure of Remacle *et al.* [25], where a stabilized gradient of the field is first computed by solving a DG problem on each element. A stabilized Hessian is obtained from this gradient by solving a second local DG problem. Near discontinuities, the Hessian of the exact solution does not exist. Thus, using an approximate Hessian near discontinuities will provide unreliable results. Our procedure uses the shock detector (Sec. 3) to decide if the Hessian should be used or not. If the Hessian cannot be used, we use the gradient of the field on each side of the shock to determine principal directions of the size field. A user defined minimal size field and a user defined maximum aspect ratio are used to specify mesh spacing in the principal direction of the size field.

The problem of Example 4.1 is solved using the anisotropic procedure. For consistency with the prior results we set $h_{\min} = h_0$ and select an aspect ratio $\alpha = 100$. The mesh obtained at time $t = 0.6$ with this setting is displayed in Fig. 2. This mesh only has 100 elements, and the results are as accurate as the solution with isotropic refinement. Element edges are aligned with the discontinuity, which makes the computation of the numerical flux across edges more accurate. In realistic situations, discontinuity surfaces are not planar and the aspect ratio that the mesh adaptation procedure will be able to reach for a fixed minimum size depends on the curvature of the discontinuity surface. Nevertheless, the savings in the number of degrees of freedom makes simulation of problems involving complex shock structures less expensive.

Example 4.2. The mesh adaptation procedure is used to solve a simplified two-dimensional blast problem. The domain under consideration (Fig. 3) is a slice of the three-dimensional problem with a tube having a diameter of 7.65 mm, and a length of 30 mm. The left side of the domain represents an axis of symmetry. On the tube walls we prescribe slip boundary conditions. The edge inside the tube at the bottom left is the inlet, and other edges are subsonic outflow surfaces. At $t = 0$, the external pressure is atmospheric at 101325 Pa, the velocity is 0 m/s and temperature is 300 K. At the inflow boundary, we impose a pressure of 6×10^6 Pa, a temperature of 631.626 K, and a velocity of 631 m/s, which correspond to a typical precursor shock in a blast problem [4, 13]. Time steps are adaptively set to maintain a maximum CFL limit of 0.9, and correspond to a range of $1 \times 10^{-8} - 5 \times 10^{-8}$ s. Mesh Adaptation is done every 1×10^{-6} s and the problem is solved to for 1×10^{-4} s.

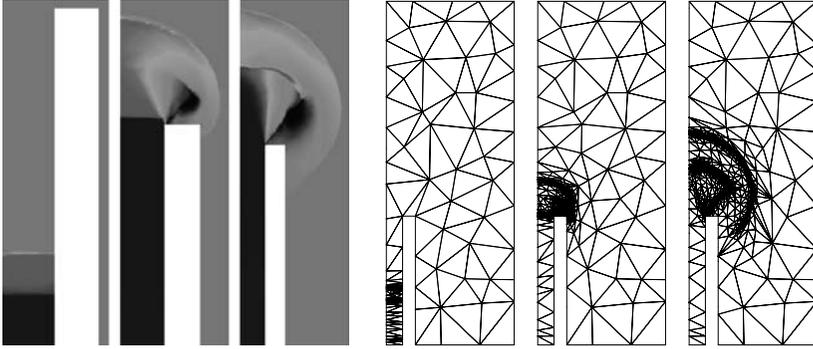


Fig. 3. Density profile (left) near the tube and adaptive mesh (right) for Example 4.2 at $t = 2 \times 10^{-5}$, 6×10^{-5} , and 8×10^{-5} s (left to right).

We set the minimum mesh size for adaptation to 0.1 mm and the maximum aspect ratio to 50. In the first phase of the simulation, the flow is mostly one-dimensional while the shock structure progresses in the tube. Thus, like a shock tube, a shock is followed by a contact discontinuity which is followed by an expansion wave traveling toward the open end of the tube. The adaptive mesh is concentrated at the contact discontinuity and the shock is aligned with the shock plane. Even when the two discontinuities are close, the mesh adaptation procedure is precise enough to separate the two structures with some coarsening between them. At time $t = 6 \times 10^{-5}$, the shock has exited the tube and forms a characteristic bubble shape. The contact is just at the edge of the tube. At time $t = 8 \times 10^{-5}$, the structure is fully developed, with a main “spherical” shock, followed by the contact, and a shear layer linking the contact with the exit of the tube. The final mesh contains 50,000 degrees of freedom.

Example 4.3. We consider a more complex tube with a muzzle brake. The tube has a diameter of 155 mm and a set of 16 perforations of 28.6 mm diameter near the muzzle. We model a slice of it with a cut plane through the center of the holes. A shock-tube type initial condition, with a shock placed 75 mm to the left of the perforations is used to initiate the flow. The upstream pressure is 565 times the downstream pressure and the upstream temperature is 2111.5 K. Fig. 4 shows the density and meshes at time $t = 1 \times 10^{-4}$, 5×10^{-4} and 1×10^{-3} s. At time $t = 1 \times 10^{-4}$ s, the initial shock has reached the fourth hole, and “mushroom-like” structures are forming outside the first and second holes. At time 5×10^{-4} s, the initial shock is outside the tube, forming its own plume. Inside the tube, shear layers from each hole are propagating

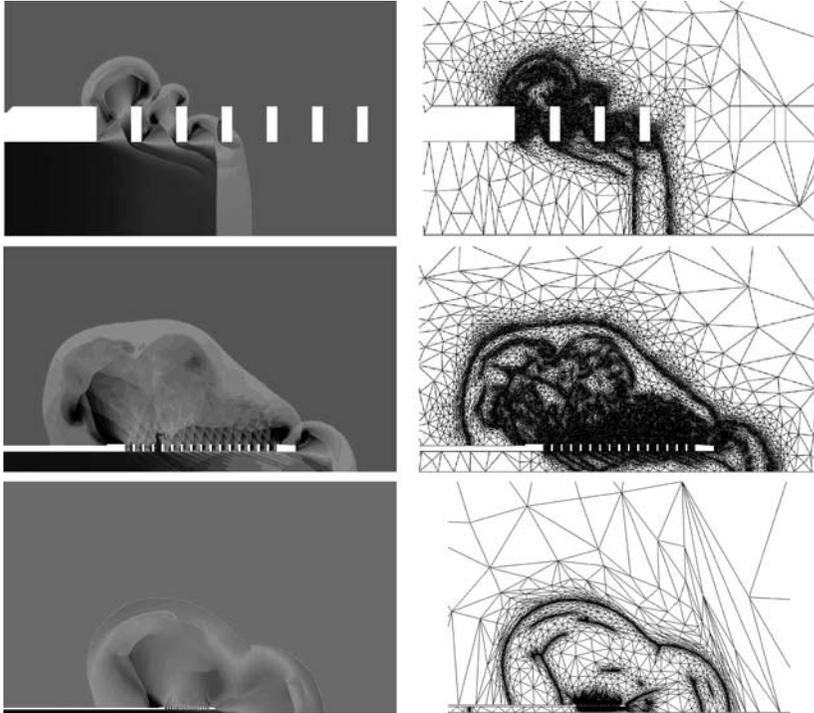


Fig. 4. Density (left) and adaptive mesh (right) for Example 4.3. at $t = 1 \times 10^{-4}$, 5×10^{-4} , and 1×10^{-3} s (top to bottom).

downstream. A main “pear-shaped” shock structure, enclosing the complex “diamond-shaped” structures resulting from the interacting shocks, is forming and moving away from the muzzle. At time $t = 2 \times 10^{-3}$ s, the initial shock has exited the muzzle and formed a “spherical” structure that merges with the pear-shaped one. The mesh is closely following the evolution of the shock structures and is aligned with them. Even though the shocks are of very different strengths, they are captured by the size field computation. The mesh at $t = 1 \times 10^{-3}$ s after 1001 mesh adaptations contains 900,000 degrees of freedom.

5. MOVING OBJECT

The next step is to include moving objects such as projectiles in the flow field. The basic approaches to handle such calculations involve (i) using deformable meshes that move with the interface [28,30] and (ii) using a fixed mesh and a technique to capture the interface, such as

a volume of fluid [17], front tracking [21], or level-set [3,29]. Herein, we describe a level-set technique [22,27] to track the interface between the rigid body and the compressible fluid. We adapt the Ghost Fluid Method (GFM) [10] to capture the correct boundary conditions at the interface and enable us to perform flow calculations over the whole domain, without explicit regard of the object.

The level-set function $\phi(\vec{x}, t)$ is defined as a signed distance function from a point \vec{x} to the boundary of the moving object at time t . If we denote the boundary of the object as B , the inside of the object as I , and the fluid part of the domain as O , then $\phi(x, t)$ satisfies

$$\phi(\vec{x}, t) < 0, \quad \text{if } \vec{x} \in I, \quad \phi(\vec{x}, t) = 0, \quad \text{if } \vec{x} \in B, \quad \phi(\vec{x}, t) > 0, \quad \text{if } \vec{x} \in O. \quad (5.1)$$

The normal vector to the moving object is

$$\vec{n}(\vec{x}, t) = \nabla\phi(\vec{x}, t), \quad \vec{x} \in F, \quad (5.2)$$

and is extended to the entire domain ($I \cup B \cup O$). This normal vector is a constant on any trajectory perpendicular to a surface with a constant value of ϕ . In the present work, we assume that $\phi(\vec{x}, t)$ can be explicitly defined, which is reasonable for the rigid objects under consideration. Treatment of deformable objects will need an actual discretization and computation of the change of the shape of ϕ . The development of the following “boundary” conditions implied by the moving object would, however, be the same.

Elements inside the rigid object ($\phi < 0$) are modeled as “ghost fluid” elements. The Euler equations are solved using the DG method just as they are in the actual fluid. The field variables in ghost elements are set to reproduce the behavior at a solid–fluid interface.

Mesh entities that lie completely in the fluid ($\phi > 0$) require no special attention. Entities that are either in the object ($\phi < 0$) or span the interface $\phi = 0$ require a proper setting of fluid variables in the “ghost” (interior) part of the object to capture the correct conditions at the moving interface. The moving interface is treated as a contact discontinuity since the fluid near the interface moves with the speed of the object. Thus, the pressure and normal velocity $\vec{v}^n = \vec{v} \cdot \vec{n}$ are continuous at the interface and we prescribe ghost fluid values as a reflection about $\phi = 0$ surface to enforce these conditions [9, 10]. Consider any ghost point \vec{x}_G in the object and its reflected image \vec{x}_R across the interface. The density ρ_G , energy E_G , and velocity \vec{v}_G at \vec{x}_G , in terms of similar quantities at \vec{x}_R are

$$\rho_G = \rho_R, \quad (5.3)$$

$$\vec{v}_G = \vec{v}_R^t - \vec{v}_R^n + 2\vec{U}_G^n, \quad (5.4)$$

$$E_G = E_R - \frac{1}{2}\rho_R|\vec{v}_R|^2 + \frac{1}{2}\rho_R|\vec{v}_G|^2, \quad (5.5)$$

where \vec{v}_R^t and \vec{v}_R^n are the tangential and normal velocity components and \vec{U}_G is the velocity of the moving rigid body at \vec{x}_G . For a given point \vec{x}_G , the closest point \vec{x}_I on the interface can be approximated as

$$\vec{x}_I = \vec{x}_G - \phi(\vec{x}_G)\nabla\phi(\vec{x}_G). \quad (5.6)$$

Thus, the reflected point is

$$\vec{x}_R = \vec{x}_G - 2\phi(\vec{x}_G)\nabla\phi(\vec{x}_G). \quad (5.7)$$

In the practice, only those ghost entities within a small distance to the interface need be considered. The bandwidth can be set to about three to four times the element cell size near the interface. Thus, the level-set function need only be defined in a small neighborhood of the fluid–solid interface. Thus, the computation is efficient even when the object has a large volume relative to the element size.

Example 5.1. Consider a two-dimensional Mach 3 flow in a 1×4 channel. Initially the channel contains a diatomic gas having a unit density and pressure. A gas having this density and pressure enters from the left boundary at Mach 3. A circular cylinder with radius 0.125 moves from right to left at Mach 3. Reflecting boundary conditions are applied to solid channel walls.

Density contours are shown at $t=0.1$, 0.25, and 1.6 s in Fig. 5. These results compare well with the flow about a fixed ball in a channel with the fluid having a speed corresponding to the sum of the speed of the cylinder and the speed of the fluid.

Example 5.2. We solve Example 4.2, with the flow driven by a square projectile having an initial speed of Mach 0.9. Preliminary results, presented in Fig. 6, show the pressure field at stages when the projectile is inside the tube and when it has exited. These preliminary results are encouraging. We reproduce the strong precursor shock. Had we applied a moving-mesh to explicitly track the projectile, difficulties arise when the projectile exits the tube. With the presented method, no special difficulties arise in this case.

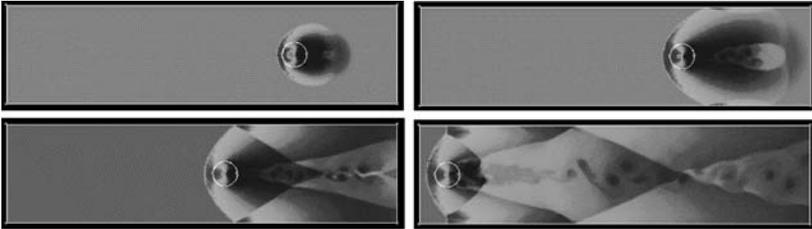


Fig. 5. Density field at $t = 0.1$ (upper left), 0.25 (upper right), 0.5 (lower left), and 1.6 s (lower right).

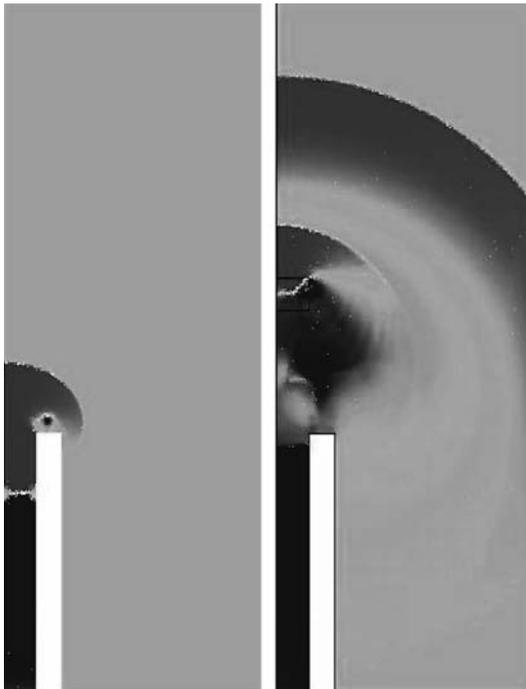


Fig. 6. Pressure field for Example 5.2 at $t = 6 \times 10^{-5}$ s when the projectile is inside the tube (left) and at $t = 2 \times 10^{-4}$ s when the projectile has left the tube (right).

6. CONCLUSION

We described procedures to model transient shock problems applied to blast flow using DGMs and mesh adaptation. The DGM enables us to sharply resolve shocks and other discontinuities. Error estimation results related to DG theory [1, 15] allow us to build robust discontinuity

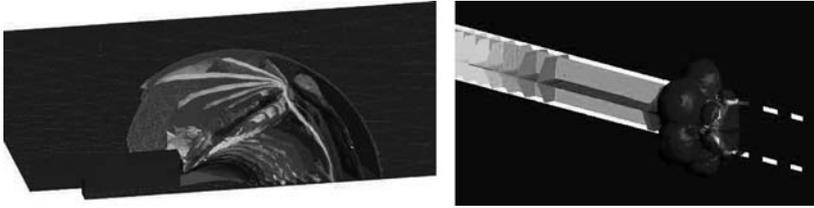


Fig. 7. Three-dimensional results: density iso-surface. Left idealized square section cannon, right, idealized hexagonal section cannon with muzzle break.

indicators that efficiently drive the limiting and the mesh adaptation procedures. The level-set method/GFM introduced to account for moving objects is an interesting alternative to the more complex procedure of following objects with a moving mesh. More work is necessary to assess the method's accuracy, stability, and efficiency. Future work includes comparisons with a moving-mesh approach and extensions to three-dimensional problems. Encouraging results in this direction are shown in Fig. 7. Two three-dimensional idealized blast problems were solved in a parallel computational environment. The results on the left of Fig. 7 were obtained on a four-processor system with a final mesh of about two million degrees of freedom. The idealized tube has a square section. The result on the right of Fig. 7 is an idealized tube with a hexagonal cross section and with holes in the tube wall. This computation was done on an eight-processor system with a final mesh of about eight million degrees of freedom.

REFERENCES

1. Adjerid, S., Devine, K. D., Flaherty, J. E., and Krivodonova, L. (2002). A posteriori error estimation for discontinuous Galerkin solutions of hyperbolic problems. *Comput. Meth. Appl. Mech. Eng.* **191**, 1097–1112.
2. Biswas, R., Devine, K., and Flaherty, J. (1994). Parallel adaptive finite element method for conservation laws. *Appl. Numer. Math.* **14**, 255–283.
3. Chang, Y., Hou, T., Merriman, B., and Osher, S. (1996). A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *J. Comput. Phys.* **124**, 449–464.
4. Cler, D. L., Chevaugnon, N., Shephard, M. S., Flaherty, J. E., and Remacle, J.-F. (2003). CFD application to gun muzzle blast—a validation case study. In *41st AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada.
5. Cockburn, B., Hou, S., and Shu, C. (1990). The Runge–Kutta local projection discontinuous Galerkin finite element method for the conservation laws IV: the multidimensional case. *Math. Comput.* **54**, 545–581.
6. Cockburn, B., Karniadakis, G., and Shu, C.-W. (2000). *Discontinuous Galerkin Methods*. Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin.
7. Colella, P., and Glaz, H. M. (1985). Efficient solution algorithms for the Riemann problem for real gases. *J. Comput. Phys.* **59**, 264–289.

8. Dubiner, M. (1991). Spectral methods on triangles and other domains. *SIAM J. Scien. Comput.* **6**, 345–390.
9. Fedkiw, R. (2002). Coupling an Eulerian fluid calculation to a Lagrangian solid calculation with the ghost fluid method. *J. Comput. Phys.* **175**, 200–224.
10. Fedkiw, R., Merriman, B., Donat, R., and Osher, S. (1999). A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method). *J. Comput. Phys.* **152**, 457.
11. Flaherty, J. E., Loy, R., Shephard, M., Szymanski, B., Teresco, J., and Ziantz, L. (1997). Adaptive local refinement with octree load-balancing for the parallel solution of three-dimensional conservation laws. *J. Parallel Distr. Comput.* **47**, 139–152.
12. Frey, P. J., and Alauzet, F. (2003). Anisotropic mesh adaptation in 3D. In *12th International Meshing Roundtable*, Santa Fe, New Mexico.
13. Klingenberg, J. M. H. G. (1992). *Gun Muzzle Blast and Flash*. Progress in Astronautics and Aeronautics Series. AIAA Reston, VA.
14. Gottlieb, S., and Shu, C.-W. (1998). Total variation diminishing Runge-Kutta schemes. *Math. Comput.* **67**, 73–85.
15. Krivodonova, L., and Flaherty, J. E. (2003). Error estimation for discontinuous Galerkin solutions of two-dimensional hyperbolic problems. *Adv. Comput. Math.* **19**, 57–71.
16. Krivodonova, L., Xin, J., Remacle, J.-F., Chevaugéon, N., and Flaherty, J. E. (2004). Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws. *Appl. Numer. Math.* **48**, 323–338.
17. Lafauric, B., Nandone, C., Scardovelli, R., Zaleski, S., and Zanetti, G. (1994). Modelling merging and fragmentation in multiphase flows with SURFER. *J. Comput. Phys.* **113**, 134–147.
18. Lesaint, P., and Raviart, P. (1974). On a finite element method for solving the neutron transport equation. In de Boor, C. (ed.), *Mathematical Aspects of Finite Elements in Partial Differential Equations*, Academic Press, New York. pp. 89–145.
19. Li, X. (2003). *Mesh Modification Procedures for General 3-D Non-Manifold Domains*. PhD Thesis, Rensselaer Polytechnic Institute.
20. Li, X., Shephard, M. S., and Beall, M. W. (2003). 3D anisotropic mesh adaptation by mesh modifications. *Comput. Meth. Appl. Mech. Eng.*, submitted.
21. Mulder, W., Osher, S., and Sethian, J. (1992). Computing interface motion in compressible gas dynamics. *J. Comput. Phys.* **100**, 209–228.
22. Osher, S., and Fedkiw, R. (2002). *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, Berlin.
23. Reed, W., and Hill, T. (1973). *Triangular Mesh Methods for the Neutron Transport Equation*. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory.
24. Remacle, J.-F., Flaherty, J. E., and Shephard, M. S. (2003). An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems. *SIAM Rev.* **45**, 55–73.
25. Remacle, J.-F., Li, X., Chevaugéon, N., and Shephard, M. S. (2002). Transient mesh adaptation using conforming and non-conforming mesh modifications. In *11th International Meshing Roundtable*, Sandia National Laboratories. pp. 261–271.
26. Remacle, J.-F., Pinchedez, K., Flaherty, J. E., and Shephard, M. S. (2001). An efficient local time stepping-discontinuous Galerkin scheme for adaptive transient computations. Technical Report 2001-13, SCOREC, RPI, Troy, NY.
27. Sethian, J. (1999). *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, New York, NY.

28. Stein, K., Tezduyar, T., and Benney, R. (2004). Automatic mesh update with the solid-entention mesh moving technique. *Comput. Meth. Appl. Mech. Eng.* **193**, 2019–2032.
29. Sussman, M., Smereka, P., and Osher, S. (1994). A level set approach for computing solutions to incompressible two-phase flow. *J. Comput. Phys.* **114**, 146–159.
30. Tezduyar, T. (2001). Finite element methods for flow problems with moving boundaries and interfaces. *Arch. Comput. Methods Eng.* **8**, 83–130.
31. van Leer, B. (1995). *Flux Vector Splitting for the Euler Equations*, Technical report, ICASE Report, NASA Langley Research Center.
32. Warburton, T., Sherwin, S., and Karniadakis, G. (1999). Spectral basis functions for 2d hybrid hp elements. *SIAM J. Sci. Comput.* **20**, 1671–1695.
33. Woodward, P., and Colella, P. (1984). The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.* **54**, 115–173.