AN ADAPTIVE DISCONTINUOUS GALERKIN TECHNIQUE WITH AN ORTHOGONAL BASIS APPLIED TO COMPRESSIBLE FLOW PROBLEMS

JEAN-FRANÇOIS REMACLE [†‡], JOSEPH E. FLAHERTY [‡], AND MARK S. SHEPHARD [‡]

Abstract. We present a high-order formulation for solving hyperbolic conservation laws using the Discontinuous Galerkin Method (DGM). We introduce an orthogonal basis for the spatial discretization and use explicit Runge-Kutta time discretization. Some results of higher-order adaptive refinement calculations are presented for inviscid Rayleigh Taylor flow instability and shock reflexion problems. The adaptive procedure uses an error indicator that concentrates the computational effort near discontinuities.

Key words. Discontinuous Galerkin, adaptive meshing , Orthogonal Basis.

1. Introduction.

The Discontinuous Galerkin Method (DGM) was initially introduced by Reed and Hill in 1973 [16] as a technique to solve neutron transport problems. Lesaint [13] 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 or electromagnetic problems [4]. The DGM is somewhere between a finite element and a finite volume method and has many good features of both.

Finite element methods (FEMs), for example, involve a double discretization. First, the physical domain Ω is discretized into a collection of $\mathcal{N}_e$ elements

$$\mathcal{T}_e = \bigcup_{e=1}^{\mathcal{N}_e} e \tag{1.1}$$

called a mesh. Then, the continuous function space $V(\Omega)$ containing the solution of the problem is approximated on each element e of the mesh, defining a finite-dimensional space $V_e(\mathcal{T}_e)$. The DGM is a finite element method in the sense that both geometrical and functional discretizations define the finite-dimensional approximation space V_e.

The accuracy of a finite element discretization depends both on geometrical and functional discretizations. Adaptivity seeks an optimal combination of these two ingredients: p-refinement is the expression used for functional enrichment and h-refinement for mesh enrichment.

Classical continuous FEMs typically use conforming meshes where elements share only complete boundary segments. Thus, spatial discretizations like those shown in Figure 1.1 would, normally, not be allowed. Since the approximation space V_e is also constrained to be a subspace of a continuous function space, e.g., H^1, the basis (shape functions) for V_e are typically associated with element vertices, edges, faces, or interiors. These simplify the imposition of continuity requirements but limit choices. The DGM allows more general mesh configurations and discontinuous bases (see Figure 1.1) which simplify both h- and p-refinement. For example, non-conforming meshes and arbitrary bases for functional approximation [20] may be used. In particular, we use a L^2-orthogonal basis that yields a diagonal mass matrix.

The DGM can also be regarded as an extension of Finite Volume Methods (FVMs) to arbitrary orders of accuracy without the need to construct complex stencils for high-order reconstruction. Indeed, the DGM stencil remains invariant for all polynomial degrees. This greatly simplifies parallel implementation for methods of all polynomial orders. Finally, the

[†] This work was supported ASCI Flash Center at the University of Chicago, under contract B341495, by the U.S. Army Research Office through grant DAAG55-98-1-0200, and by the National Science Foundation through grant DMS-0074174

[‡] Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, New York, USA

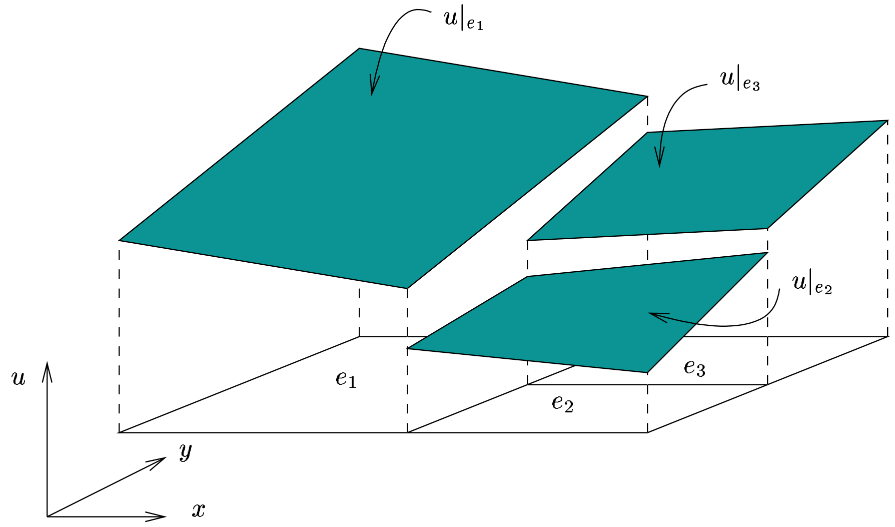

FIG. 1.1. *Non-conforming mesh with discontinuous approximations* $u|_{e_i}$ *on elements* e_i.

DGM has aspects in common with finite difference schemes in that it may use fluxes associated with Riemann problems [3].

Herein, we concentrate on DGM formulations for hyperbolic conservation laws (§2). For the spatial discretization, we choose an orthogonal basis that diagonalizes the mass matrix and, thus, simplifies its evaluation (§2.1). Time discretization is performed by an explicit total variation diminishing Runge-Kutta scheme [3]. To improve the performance of the explicit integration, we use a new local time stepping procedure similar to one used by Flaherty *et al.* [7] and which will be explained in a forthcoming paper [18].

We present procedures to perform adaptive computations where the discretization space V_e changes in time. Because of the flexibility of the DGM, we are able to change both mesh and elementary polynomial orders often, e.g., several thousand times with very little computational overhead.

Transient computation of unstable flows provide an application where adaptivity in time is crucial. The instability of an interface separating miscible fluids of different densities subject to gravity is known as a *Rayleigh-Taylor Instability* (RTI). Bubbles (spikes) of lighter (heavier) fluid penetrate into the heavier (lighter) fluid, leaving behind a region where the two fluids are mixed. This mixing region quickly becomes irregular and may provide an understanding of turbulence since the flow there has chaotic features [11, 22].

Young *et al.* [22] solved an incompressible RTI problem governed by the Boussinesq equations using spectral methods. We likewise believe that the complex structure of the mixing zone could be efficiently represented by high-order polynomials. Fryxell [9] used a piecewise parabolic method [21] with adaptive h-refinement to solve compressible RTI problems in two and three dimensions. Without explicit interface tracking [10], h-adaptivity will certainly be necessary to accurately represent the complex evolution of bubbles and spikes [11].

We present solutions of a standard two-dimenstional RTI problem using h- and p-refinement. Increasing the polynomial degree p improves the quality of the solution. However, p-refinement alone is not effective for capturing the fine scale structures near discontinuities. Using an error indicator based on solution jumps, we present results for the same problem using adaptive h-refinement and compare computations with those using adaptive p-refinement. Finally, an adaptive hp-refinement computation is performed which is shown to be the best of these RTI calculations.

2. Discontinuous Finite Element Formulation for Conservation Laws.

Consider an open set $\Omega \subset \mathbb{R}^3$ whose boundary $\partial\Omega$ is Lipschitz continuous with a normal $\vec{n}$ that is defined everywhere. We seek to determine $\mathbf{u}(\Omega, t) : \mathbb{R}^3 \times \mathbb{R} \rightarrow \mathsf{L}^2(\Omega)^m = V(\Omega)$ as the solution of a *system of conservation laws*

$$\partial_t \mathbf{u} \;+\; \mathbf{div}\, \vec{\mathbf{F}}(\mathbf{u}) \;=\; \mathbf{r}. \tag{2.1}$$

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

$$\vec{\mathbf{F}}(\mathbf{u}) = (\vec{F}_1(\mathbf{u}), \ldots, \vec{F}_m(\mathbf{u}))$$

is the flux vector with the ith component $\vec{F}_i(\mathbf{u}) : \left(\mathsf{H}^1(\Omega)\right)^m \rightarrow \mathsf{H}(\mathrm{div}\,, \Omega)$. Function space $\mathsf{H}(\mathrm{div}\,, \Omega)$ consists of square integrable vector valued functions whose divergence is also square integrable i.e.,

$$\mathsf{H}(\mathrm{div}\,, \Omega) = \left\{ \vec{v} \mid \vec{v} \;\in\; \mathsf{L}^2(\Omega)^3 \;\;,\; \mathrm{div}\, \vec{v} \in \mathsf{L}^2(\Omega) \;\right\}.$$

With the aim of constructing a Galerkin form of (2.1), let $(\cdot, \cdot)_\Omega$ and $\langle \cdot, \cdot \rangle_{\partial\Omega}$ respectively denote the standard $\mathsf{L}^2(\Omega)$ and $\mathsf{L}^2(\partial\Omega)$ scalar products. Multiply equation (2.1) by a test function $\mathbf{w} \in V(\Omega)$, integrate over Ω and use the divergence theorem to obtain the following variational formulation

$$(\partial_t \mathbf{u}, \mathbf{w})_\Omega - (\vec{\mathbf{F}}(\mathbf{u}), \mathbf{grad}\, \mathbf{w})_\Omega + \langle \vec{\mathbf{F}}(\mathbf{u}) \cdot \vec{n}, \mathbf{w} \rangle_{\partial\Omega} = (\mathbf{r}, \mathbf{w})_\Omega \;, \quad \forall \mathbf{w} \in V(\Omega). \tag{2.2}$$

Finite element methods (FEMs) involve a double discretization. First, the physical domain Ω is discretized into a collection of $\mathcal{N}_e$ elements like in (1.1) The continuous function space $V(\Omega)$ containing the solution of (2.2) is approximated on each element e of the mesh to define a finite-dimensional space $V_e(\mathcal{T}_e)$. With discontinuous finite elements, V_e is a "broken" function space that consists in the direct sum of elementary approximations $\mathbf{u}_e$ (we use here a polynomial basis $\mathbb{P}^q(e)$ of order q):

$$V_e(\mathcal{T}_e) = \{\mathbf{u} \mid \mathbf{u} \in \mathsf{L}^2(\Omega)^m, \mathbf{u}_e \in \mathbb{P}^q(e)^m = V_e(e), \forall e \in \mathcal{T}_e\}. \tag{2.3}$$

Because all approximation are disconnected, we can solve the conservation laws on each element to obtain

$$(\partial_t \mathbf{u}_e, \mathbf{w})_e - (\vec{\mathbf{F}}(\mathbf{u}_e), \mathbf{grad}\, \mathbf{w})_e + \langle \mathbf{F}_n, \mathbf{w} \rangle_{\partial e} = (\mathbf{r}, \mathbf{w})_e \;, \quad \forall \mathbf{w} \in V_e(e). \tag{2.4}$$

Now, a discontinuous basis implies that the normal trace $\mathbf{F}_n = \vec{\mathbf{F}}(\mathbf{u}) \cdot \vec{n}$ is not defined on ∂e. In this situation, a *n*umerical flux $\mathbf{F}_n(\mathbf{u}_e, \mathbf{u}_{e_k})$ is usually used on each portion ∂_{e_k} of ∂e shared by element e and neighboring element e_k. Here, $\mathbf{u}_e$ and $\mathbf{u}_{e_k}$ are the restrictions of solution $\mathbf{u}$, respectively, to element e and element e_k. This numerical flux must be continuous, so $\vec{\mathbf{F}} \in \mathsf{H}(\mathrm{div}\,, \Omega)^m$, and be consistent, so $\mathbf{F}_n(\mathbf{u}, \mathbf{u}) = \vec{\mathbf{F}}(\mathbf{u}) \cdot \vec{n}$. With such a numerical flux, equation (2.4) becomes

$$(\partial_t \mathbf{u}_e, \mathbf{w})_e - (\vec{\mathbf{F}}(\mathbf{u}_e), \mathbf{grad}\, \mathbf{w})_e + \sum_{k=1}^{n_e} \langle \mathbf{F}_n(\mathbf{u}_e, \mathbf{u}_{e_k}), \mathbf{w} \rangle_{\partial e_k} = (\mathbf{r}, \mathbf{w})_e \;, \quad \forall \mathbf{w} \in V_e(e), \tag{2.5}$$

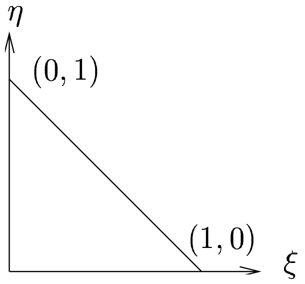

FIG. 2.1. *Reference triangular element*

where n_e is the number of faces of element e. Only the normal traces have to be defined on ∂e_k and several operators are possible [12, 21]. 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 [6,21], an exact Riemann solver is used to compute the numerical fluxes and a slope limiter [2] is used to produce monotonic solutions when polynomial degrees $q > 0$. For Rayleigh-Taylor instabilities, Roe's flux linearization [19] is used with a physical limiter that we describe in §4.

2.1. Spatial discretization. Even if DGM solutions do not depend on the choice of basis (because they all span $\mathbb{P}^q(e)$), some of them are more convenient and computationally efficient than others. We construct an orthogonal basis of $\mathbb{P}^q(e)$ with respect to the $\mathsf{L}^2(e)$ scalar product. As a result, an explicit time integration scheme will neither necessitate "lumping" nor inversion of the mass matrix. Another advantage, which is, perhaps, more important, is that the orthogonal basis makes p-refinement trivial.

In two dimensions, consider a right-triangular reference element as shown in Figure 2.1. Without a need to maintain inter-element continuity, consider a basis of p^{th} degree monomials in (ξ, η), i.e.,

$$B = \{b_i, i = 1, \dots, N_q\} = \{1, \xi, \eta, \xi^2, \xi\eta, \eta^2, \xi^3, \dots, \eta^q\}, \tag{2.6}$$

with $N_q = (q+1)(q+2)/2$. This basis is said to be hierarchical in the sense that, if P^i is the space of i^{th}-degree polynomials in (ξ, η), then

$$B = \mathbb{P}^0 \oplus \left(\mathbb{P}^1 \setminus \mathbb{P}^0\right) \oplus \left(\mathbb{P}^2 \setminus \mathbb{P}^1\right) \oplus \cdots \oplus \left(\mathbb{P}^{q-1} \setminus \mathbb{P}^q\right). \tag{2.7}$$

Any field f is approximated in $\mathbb{P}^q$ as

$$f(\xi, \eta) = \sum_{i=1}^{N_q} f_i b_i(\xi, \eta). \tag{2.8}$$

Let us define the scalar product on the reference element as

$$(b_i, b_j)_r = \int_0^1 \int_0^{1-\xi} b_i b_j \, d\xi d\eta \tag{2.9}$$

and the induced norm

$$||f||_r^2 = (f, f)_r. \tag{2.10}$$

$\mathbf{P}^{B\to G}{}_{11}$	1.414213562373095E+00
$\mathbf{P}^{B\to G}{}_{21}$	-2.000000000000000E+00
$\mathbf{P}^{B\to G}{}_{22}$	6.000000000000000E+00
$\mathbf{P}^{B\to G}{}_{31}$	-3.464101615137754E+00
$\mathbf{P}^{B\to G}{}_{32}$	3.464101615137750E+00
$\mathbf{P}^{B\to G}{}_{33}$	6.928203230275512E+00
$\mathbf{P}^{B\to G}{}_{41}$	2.449489742783153E+00
$\mathbf{P}^{B\to G}{}_{42}$	-1.959591794226528E+01
$\mathbf{P}^{B\to G}{}_{43}$	0.000000000000000E+00
$\mathbf{P}^{B\to G}{}_{44}$	2.449489742783160E+01
$\mathbf{P}^{B\to G}{}_{51}$	4.242640687119131E+00
$\mathbf{P}^{B\to G}{}_{52}$	-2.545584412271482E+01
$\mathbf{P}^{B\to G}{}_{53}$	-8.485281374238392E+00
$\mathbf{P}^{B\to G}{}_{54}$	2.121320343559552E+01
$\mathbf{P}^{B\to G}{}_{55}$	4.242640687119219E+01
$\mathbf{P}^{B\to G}{}_{61}$	5.477225575051629E+00
$\mathbf{P}^{B\to G}{}_{62}$	-1.095445115010309E+01
$\mathbf{P}^{B\to G}{}_{63}$	-3.286335345030997E+01
$\mathbf{P}^{B\to G}{}_{64}$	5.477225575051381E+00
$\mathbf{P}^{B\to G}{}_{65}$	3.286335345031001E+01
$\mathbf{P}^{B\to G}{}_{66}$	3.286335345030994E+01

TABLE 2.1

Second-order basis coefficients on the reference triangle of Figure 2.1.

We seek an alternate basis $G = \{g_i, i = 1, \dots, N_q\}$ of $\mathbb{P}^q$ which is orthonormal, i.e., $(g_i, g_j)_e = \delta_{ij}$. For this purpose, we apply Gram-Schmidt orthogonalization to basis B and construct

$$g_i = \sum_{j=1}^{i} \mathbf{P}_{ij}^{B\to G} b_j \tag{2.11}$$

with $\mathbf{P}^{B\to G} = \{g_1, \dots, g_{N_p}\}$ a triangular matrix representing the change of basis from B to G. The j^{th} column of $\mathbf{P}^{B\to G}$ is the coordinates of g_j in basis B. Any shape function b_k may be expressed as $b_k = \xi^{\alpha(k)}\eta^{\beta(k)}$ with exponents $\alpha(k)$ and $\beta(k)$ depending on k. Scalar products $(b_i, b_j)_e$ are calculated as

$$\begin{aligned}(b_i, b_j)_e &= \int_0^1 \int_0^{1-\xi} \xi^m \eta^n \, d\xi d\eta = \frac{1}{n+1} \int_0^1 \xi^m (1-\xi)^{n+1} d\xi \\ &= \frac{1}{n+1} \sum_{l=0}^{n+1} C_n^l \int_0^1 \xi^{m+l} d\xi = \frac{1}{n+1} \sum_{l=0}^{n+1} \frac{C_n^l}{m+l+1}\end{aligned} \tag{2.12}$$

with $m = \alpha(i) + \alpha(j)$ and $n = \beta(i) + \beta(j)$. This simple result (2.12) avoids the need for numerical integration in the Gram-Schmidt process so that any order shape functions can be computed without a loss of precision. In Table 2.1, we give the transformation $\mathbf{P}^{B\to G}$ for a complete second-order basis (q=2).

Integration of shape functions are usually not done in the parametric coordinates (ξ, η) of the element but in the actual coordinates (x, y). $\mathsf{L}^2(e)$ orthogonality of g_i's will only be preserved if the mapping from the actual to the parametric coordinates is linear, i.e., the Jacobian of the mapping is constant. Curved elements, which are essential for higher-order analysis on curved domains [1], will require some modifications, e.g., we can use Gram-Schmidt orthogonalization with a different scalar product and induced norm. Shape functions would become element dependent and the matrix $\mathbf{P}^{B\to G}$ would have to be computed and stored for every curved element of the mesh. This is not excessive because the total memory never exceeds that required for a global mass matrix and the number of curved elements is typically $O(\sqrt{\mathcal{N}_e})$ for a problem with $\mathcal{N}_e$ elements.

3. Adaptive h- and p-Refinement. Adaptive analysis techniques have been shown to be highly effective for use in fluid mechanics problems (e.g., [14, 15]). H-refinement consists of modifying element sizes while p-refinement consists of modifying polynomial orders. We seek methods where it is possible to do either or both on any element of the mesh at any time step. We describe a procedure to alter element sizes or polynomial orders using only local operations to change the approximation space. We also give a heuristic rule as to when and where to enrich the solution space.

3.1. P-Refinement. Increasing the degree of an approximation is straightforward: higher-order coefficients are initially set to zero with the new approximation being an identity projection of the existing one. Reduction of polynomial degree is more complex because of the associated loss of precision. Reduction of order is not only used when higher-order coefficients are small and, hence, do not contribute to the accuracy of the solution, but to eliminate spurious oscillations in the solution when non smooth fields are present.

All shape functions are orthogonal to the constant one which ensures conservation of mass. More generally, consider a function $f \in \mathbb{P}^q(e)$ defined on element e as

$$f = \sum_{i=1}^{N_q} f_i g_i. \tag{3.1}$$

The L^2 projection $\Pi_{\mathsf{L}^2}(f) : \mathbb{P}^q(e) \to \mathbb{P}^r(e)$, $r < q$, is defined as

$$((\Pi_{\mathsf{L}^2}(f) - f), g_i)_e = 0 \ \ , \ \ i = 1, \dots, N_r. \tag{3.2}$$

Because of the $\mathsf{L}^2(e)$-orthogonality of g_i, $i = 1, \dots, N_p$, (3.2) is satisfied by

$$\Pi_{\mathsf{L}^2}(f) = \sum_{i=1}^{N_r} f_i g_i. \tag{3.3}$$

Thus, the L^2 projection from any order to a lower order is performed by dropping all higher-order coefficients.

3.2. H-Refinement. Modifying element sizes is also straightforward. The DGM does not impose inter-element continuity of the approximated fields. As with p-refinement, we are able to divide elements in a non-conforming way as shown in Figure 3.1. Element e may be split into four sub-elements e_i , $i = 1, \dots, 4$, or elements e_i, $i = 1, \dots, 4$, may be coarsened to recover element e. For both refinement and coarsening operations, a L^2 projection is performed to define the new solutions. With refinement, the identity projection is used. A loss of precision is associated with coarsening, as well as with the order reduction for p-refinement case.

3.3. Refinement Strategy. Both h- and p-refinement and coarsening procedures are strictly local to these elements being enriched. The h- and p-refinement procedures use identity projections with no loss of precision or conservation. Adaptivity with the DGM is, thus,

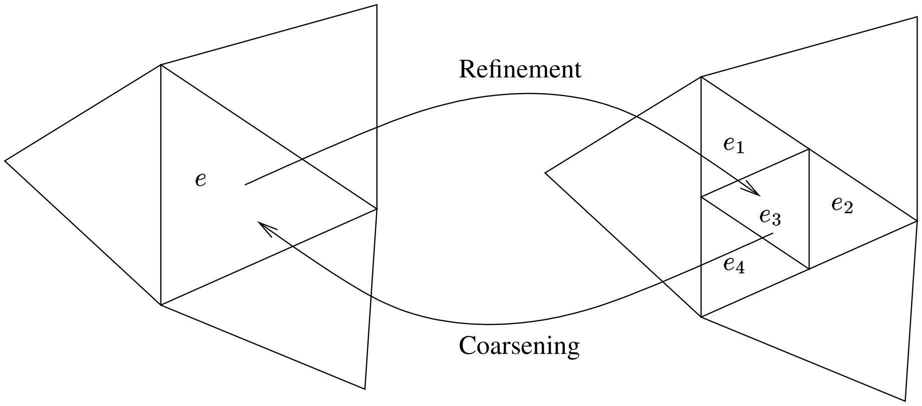

FIG. 3.1. *Non-conforming element refinement and coarsening of a triangle*

fast and accurate. This is important, since our intended applications involve tens of thousands of time steps with several thousand steps of adaptive refinement.

The level of refinement l_e of an element e is the number of h- or p-refinements that have been necessary to reach the current size or polynomial order of e. Let us consider an error indicator ϵ_e that gives an indication of the discretization error on element e. If l_{max} is the maximum level of refinement allowed (a user input) and if ϵ_{max} is the maximum value of ϵ_e, $e = 1, \ldots, \mathcal{N}_e$, then we determine the appropriate refinement level for element e by finding i such that $\epsilon_{max}/d^{i+1} \leq \epsilon_e < \epsilon_{max}/d^i$ and setting $l_e = \max(l_{max} - i, 1)$. The constant d is prescribed by the user. If, e.g., $d = 10$, then all elements where $\epsilon_e > \epsilon_{max}/10$ will be refined at the maximum level of refinement l_{max}. All elements where $\epsilon_{max}/100 \leq \epsilon_e < \epsilon_{max}/10$ will be refined to level $l_{max} - 1$, etc.

3.4. Double Mach Reflection. In order to demonstrate our refinement strategy, we have computed the classical problem of the reflection of a planar shock moving at Mach 10 by an oblique surface canted at 30^o to the free-stream flow [6, 21] (see Figure 3.2).

The flow is governed by the Euler equations of an inviscid compressible fluid. In two space dimensions, this system has the form (2.1) with

$$\mathbf{u} = [\rho, \rho v_x, \rho v_y, E]^T \tag{3.4}$$

where ρ is the density of the fluid, $\vec{v}$ is the velocity vector and E is the internal energy. Fluxes are

$$\vec{\mathbf{F}}(\mathbf{u}) = \left[\vec{F}_1, \vec{F}_2, \vec{F}_3, \vec{F}_4\right] = \left[\rho\vec{v}, \rho v_x\vec{v} + P\vec{e}_x, \rho v_y\vec{v} + P\vec{e}_y, \rho(E+P)\right]. \tag{3.5}$$

where P is the pressure and $\vec{e}_x$ and $\vec{e}_y$ are unit vectors in the x and y directions, respectively.

An equation of state (EOS) $P = P(\rho, E)$ is necessary to close the system. The DGM can be used for any physically consistent EOS. The only change involves the computation of numerical fluxes. We choose the perfect gas EOS

$$P = (\gamma - 1)\rho\left[E - \frac{\|\vec{v}\|^2}{2}\right] \tag{3.6}$$

with gas constant $\gamma = 1.4$.

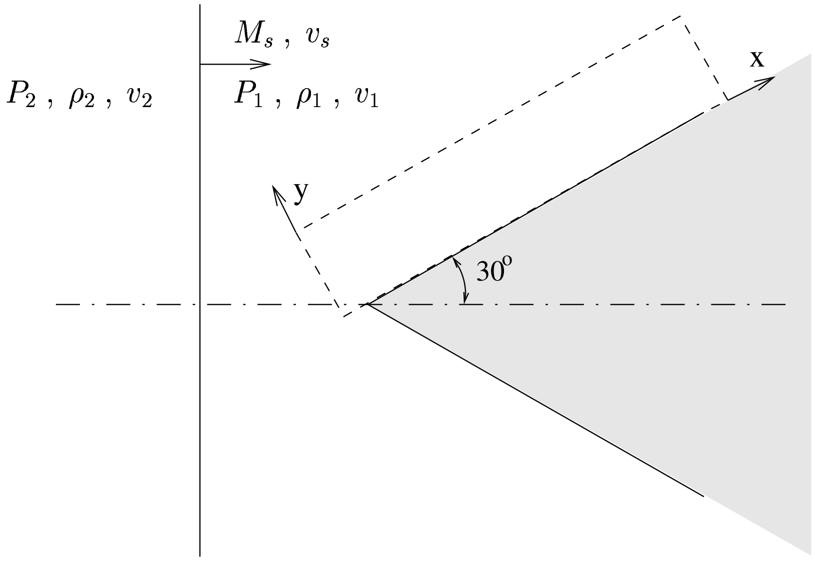

FIG. 3.2. *Double mach reflection of a strong shock travelling from left to right at $M_s = 10$. Conditions ahead of the shock are identified with a subscript 1 and those behind the shock with a subscript 2.*

The computational domain (shown with dashed lines in Figure 3.2) is a 4×1 unit rectangle oriented along the oblique surface. The reflecting wall lies on the bottom of the computational domain, starting at $x = 1/6, y = 0$. Boundary conditions at the top ($y = 1$) are set to those corresponding to the exact motion of a Mach 10 shock. Physical parameters for the gas ahead of the shock are $P_1 = 1$ and $\rho_1 = 1.4$. The Rankine-Hugoniot relations

$$v_s = M_s\sqrt{\gamma P_1/\rho_1} = 10,$$

$$p2/p1 = (2\gamma M_s^2 - (\gamma - 1))/(\gamma + 1),$$

$$\rho_2/\rho_1 = (\gamma + 1)M_s^2/((\gamma - 1)M_s^2 + 2),$$

and

$$\rho_1 v_s = \rho_2(v_s - v_2)$$

are used to compute post shock conditions.

Quadrilateral elements are used to discretize the domain with tensor products of Legendre polynomials used for the functional discretization [2]. These shape functions have the same orthogonality properties as those presented in §2.1. The DGM does not produce monotonic solutions when $p > 0$ and discontinuities are present. To reduce spurious oscillations produced near discontinuities with higher-order methods, the moment limiting scheme of Biswas *et al.* [2] is effective with rectangular elements.

We have used a level of refinement $l_{max} = 4$ with $q = 1$ and a local time stepping procedure of [18] for increased computational efficiency. The initial mesh consists of 44×11 rectangular elements. The effective mesh, i.e., the mesh we would have used without refinement and with the same resolution would have 704×176 elements. Figure 3.3 shows the evolution of mesh refinement in time. More than 400 mesh refinement steps were performed.

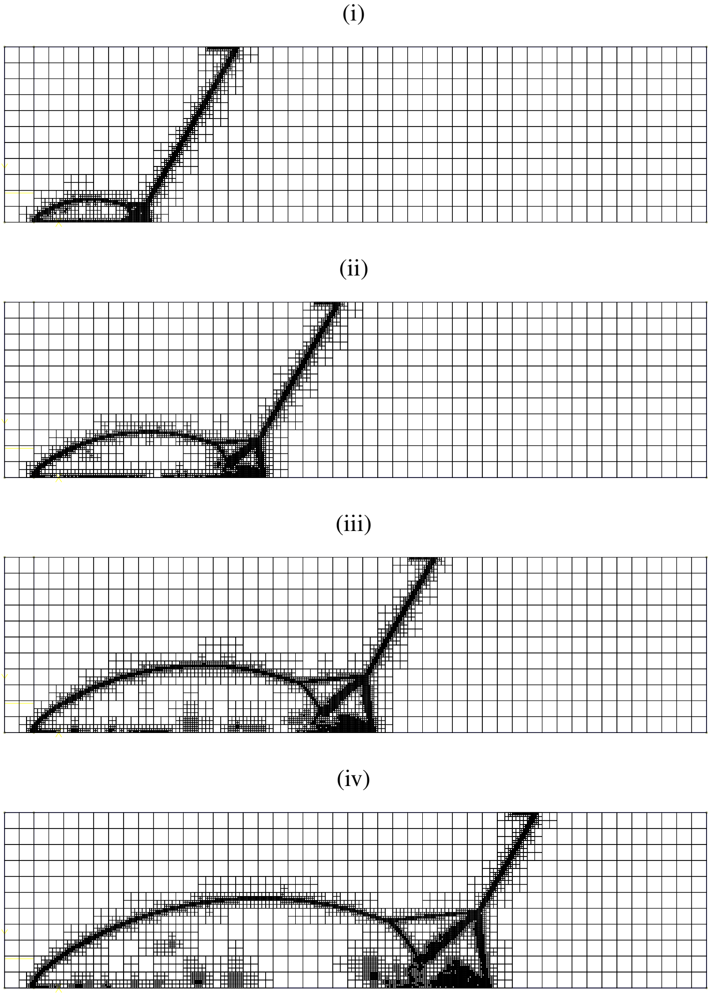

FIG. 3.3. *Refined grid for a double Mach reflection at times* 0.05 *(i),* 0.10 *(ii),*0.15 *(iii) and* 0.20 *(iv) with* $l_{max} = 4$ *and* $q = 1$.

Figure 3.4 shows the number of degrees of freedom involved in the computation as a function of time. The increase is linear, which is optimal since the solution of this problem is self similar with the total length of the discontinuities (shocks and contact surface) growing linearly in time. Figure 3.5 shows density contours at $t = 0.2$. All shocks are well resolved as well as the jet produced by the double Mach reflexion. The contact discontinuity, which turns to form the jet, is widened. This is essentially due to the limiting , which is a present research issue.

This computation was performed on one Intel PIII Xeon processor at 700 MHz. The total computation time, including adaptive refinement, was 37 minutes. As a comparison, we solved the same problem on a fixed mesh of 400×100, i.e., with much less effective resolution, in 2 hours and 27 minutes.

4. The Rayleigh-Taylor Instability. The Rayleigh-Taylor instability problem involves a heavy (cold) fluid overlying a light (warm) fluid. We consider two inviscid fluids initially

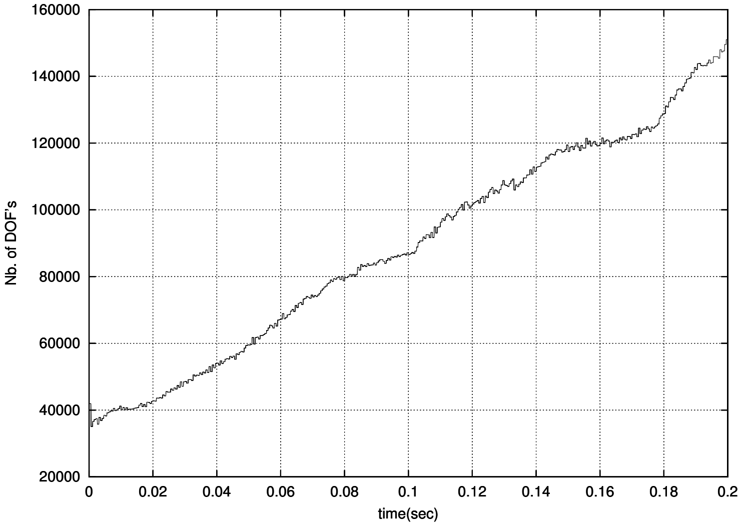

FIG. 3.4. *Number of degrees of freedom vs. time for the double Mach reflection.*

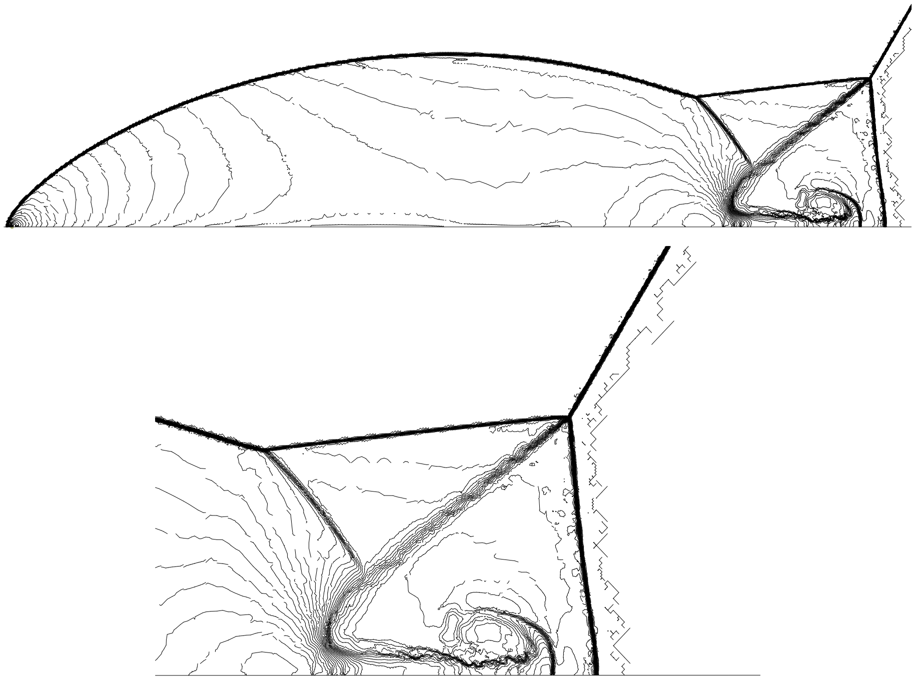

FIG. 3.5. *Density contours levels for a double Mach reflection at time* 0.2 *with* $l_{max} = 4$ *and* $q = 1$. *The bottom figure shows is a blow-up near the double Mach stems.*

in hydrostatic (unstable) equilibrium in a cavity (Figure 4.1). The upper half of the cavity is filled with a fluid of density two while the lower part is filled with a fluid of unit density. The initial pressure corresponds to hydrostatic equilibrium. An initial perturbation of the velocity initiates the instability.

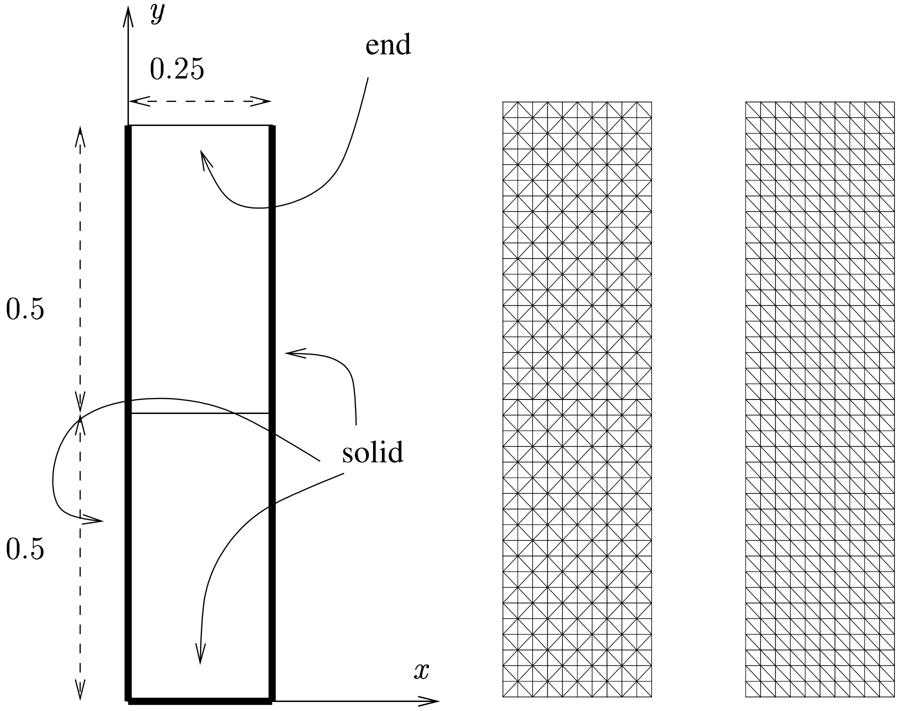

FIG. 4.1. *Geometry of the cavity (left) and meshes M^s (center) and M^{ns} (right).*

The flow is again governed by the Euler equations (3.4,3.5). with an additional body force corresponding to the gravity $\vec{g} = \{0, -1\}$

$$\mathbf{r} = [0, \rho\vec{g} \cdot \vec{e}_x, \rho\vec{g} \cdot \vec{e}_y, \rho\vec{g} \cdot \vec{v}]^T \,. \tag{4.1}$$

The initial data is summarized in Table 4.1 with $\epsilon_y = M_0\sqrt{\gamma/2}$ and $\epsilon_x = -\epsilon_y\tau/16$. We chose $M_0 = 0.1$, $\tau = 6$ and $\gamma = 1.4$. The particular form of the perturbation is defined to ensure a single mode instability, i.e., the heavy fluid falls in only one column. We impose no normal flow on three walls and parallel flow on the open upper surface (Figure 4.1). On the walls, the fluxes are $\mathbf{F}_n = \{0, p\vec{n} \cdot \vec{e}_x, p\vec{n} \cdot \vec{e}_y, 0\}$. Pressure remains as initially specified

	upper part	lower part
ρ	2	1
P	$2 - 2y$	$2 - y$
v_x	$\epsilon_x \sin(8\pi x)\cos(\pi y)\sin^{\tau-1}(\pi y)$	same as upper part
v_y	$-\epsilon_y \cos(8\pi x)\sin^{\tau}(\pi y)$	same as upper part

TABLE 4.1
Initial conditions for the RTI problem.

on the top. This problem is considered as compressible but no shocks occur and the only discontinuity is the contact surface separating the two fluids of different densities.

Cockburn and Shu [3, 5] describe a limiting procedure that prevents the approximate solution on an element from taking values outside of the range spanned by the neighboring solution averages. When applied to the RTI with $p > 0$, this limiter heavily diffuses the interface. The moment limiter of Biswas *et al* [2] does not apply to triangular elements. We observe that the amplitude of oscillations near contact surfaces does not increase with p. It is necessary, however, to avoid *catastrophic oscillations* that produce negative temperature or density. For this application, we found a "physical limiter" that essentially eliminates all spurious oscillations. After each time step (or sub-time step), we evaluate the solution at each integration point on element edges and faces. When a negative temperature or density is found, we reduce the order of the polynomial approximation by one. This procedure is applied several times until all oscillations are removed. To give an indication, in the second-order case ($q = 1$), the physical limiter was only activated 24 times for 10000 time steps and for all integration points of every edge and triangle. When the limiter is not applied, the solution becomes unbounded in all computations with $p > 0$.

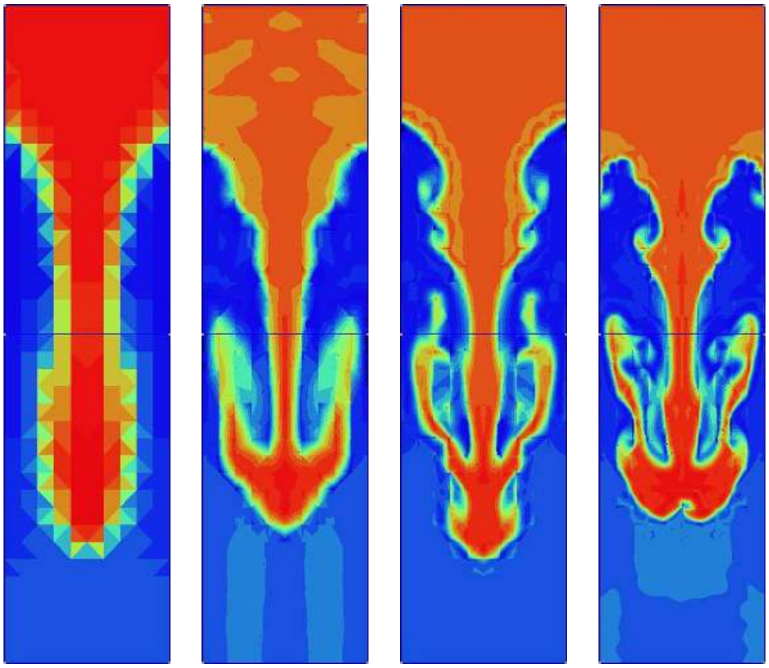

FIG. 4.2. *Density contours at* $t = 1.8$ *for RTI computations with* $q = 0, 1, 2, 3$ *(left to right) using the symmetric mesh* M^s*. The total number of equations is* $4\mathcal{N}_e N_p = 3040$ *for degree* 0, 9120 *for degree* 1, 18240 *for degree* 2 *and* 30400 *for degree* 3.

4.1. Calculations using Equal Order Approximations. We consider the two meshes M^s and M^{ns} of Figure 4.1 which are both composed of 760 triangles. M^s is symmetric while M^{ns} is not. We show density contours at $t = 1.8$ for computations performed with uniform polynomial degrees $q = 0, 1, 2$ and 3 in Figures 4.2 and 4.3. The structure of the flow is completely diffused using degree zero polynomials. For $q = 0$, Helmoltz instabilities are not apparent, the fluid column falls without generating any eddies. Increasing the polynomial degree reveals additional structure in the flow. With $q = 1$, the classical "mushroom cap"

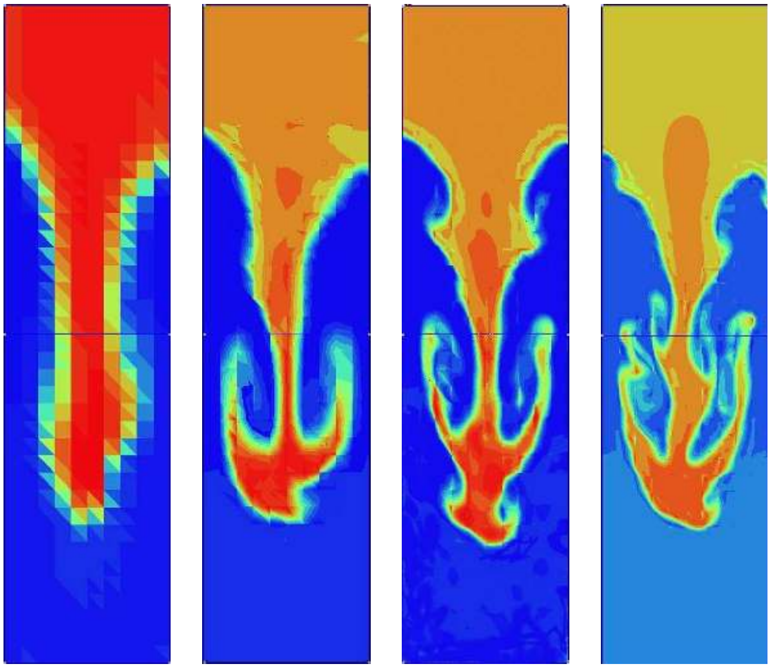

FIG. 4.3. *Density contourss at $t = 1.8$ for Rayleigh-Taylor computations with $q = 0, 1, 2, 3$ using the asymmetric mesh M^{ns}.*

structure of the RTI appears, with $q = 2$, secondary caps are created on the main fluid column. With $q = 3$, we see plumes separating from the main fluid column, generating a rough mixing zone.

Another feature of these calculations is the asymmetry that results with the asymmetric mesh M^{ns}. Rayleigh-Taylor problems are unstable; this asymmetry is, thus, not surprising. In fact, without viscosity, the solution is chaotic. The mesh asymmetry induces a small perturbation that leads to significant modification to the flow. We do not conclude that symmetric meshes are preferable, they are just unable to break the initial symmetry of the flow. However, any other perturbation will break the symmetry which is not a physical feature of this flow.

4.2. Error Indicator. The selection of an error indicator for the RTI is a problem. Error estimators suppose the existence of an exact solution of the problem and convergence of a method to it. Since the RTI has no exact solution, this is not possible. Our current aim is to capture the rich structure of the interface between the two fluids. Thus, refinement might reasonably be assumed to be proportioned to the inter-element jump of density. Hence, we select

$$\epsilon_e = \sum_{i=1}^{n_k} \int_{\partial e_k} |\rho_e - \rho_{e_k}| \, de \tag{4.2}$$

where n_k is the number of edges of element e. This error indicator is convenient for a practical implementation because it requires the same "jump" information $(\mathbf{u}_e, \mathbf{u}_{e_k})$ as needed for computation of the numerical flux $\mathbf{F}_n(\mathbf{u}_e, \mathbf{u}_{e_k})$. The error indicator will "diffuse" on one layer of elements which makes it well suited for transient calculations that feature velocities which do not move more than a cell per time step. In fact, it is always too late to adapt at time

t and our indicator will simply propagate adaptation in all directions so that a certain number of time steps can be performed using the same discretization with good precision.

4.3. Adaptive Refinement. Figures 4.4 and 4.5 show an adaptive h-refinement computation using the non-conforming h-refinement procedure described in (§3.2) with $l_{max} = 3$, $q = 1$ and $d = 50$. The h-refinement procedure was performed every 10 time steps. More than $17,000$ time steps were necessary to reach $t = 2$; thus, the mesh was refined approximatively 1700 times. Since the error estimator "diffuses" on at least one layer of elements, there was always a maximal refinement at the interface between the two fluids. The parameter $d = 50$ controls the size of the region where refinement is performed. The initial symmetric mesh M^s preserves the flow symmetry during refinement, even as the fine-scale structure develops.

Similar computations are presented in Figure 4.5 using the asymmetric mesh M^{ns}. The symmetry of the flow is approximatively preserved for $t < 1.5$, which correspond to the linear RTI regime with this resolution [22]. After that, the fully nonlinear RTI develops and the mesh asymmetry introduces small asymmetric perturbations to the flow which grow in time. The whole flow is clearly asymmetric at $t = 2$.

In either case, the effort required for mesh adaptation is less than 5% of the total computation time.

On the top of Figure 4.6, we show density contours at $t = 1$ for different levels of refinement using the symmetric mesh M^s. For all refinement levels, the main columns are similar and smaller features appear when increasing the refinement level. This flow has the property that small wavelength perturbations propagate faster than large ones. That means that the instability arrives earlier with finer meshes, which is clear in figure 4.6.

On the bottom of Figure 4.6, we apply adaptive p-refinement to the same problem. A level of refinement of $l_{max} = 4$ corresponds to a maximum polynomial degree of five since the initial degree is $q = 1$. Table 4.2 shows both the CPU time for one time step and the time step for p- and h-refinement at $t = 1.0$. For example, p-refinement with $l_{max} = 4$ and h-refinement with $l_{max} = 3$ yield comparable results in terms of computational effort.

TABLE 4.2
CPU time per time step and time step for h- and p- refinement at $t = 1.0$.

l_{max}	1	2	3	4
p-refinement	$0.23, 6.0e^{-4}$	$0.33, 4.2e^{-4}$	$1.10, 3.33e^{-4}$	$2.27, 2.7e^{-4}$
h-refinement	$0.45, 5.0e^{-4}$	$0.80, 2.5e^{-4}$	$2.55, 1.25e^{-4}$	$7.50, 6.2e^{-5}$

As with h-refinement, we see a continuous enhancement of the results but it is clear that p-refinement is not able, at least for $p \leq 5$, to capture the fine scales that h-refinement can at this refinement level. There are possible reasons for this:

- for non-smooth fields, optimal convergence of DGM's cannot be obtained through p-refienement;
- p-refinement is not as local as h-refinement since the entire element is refined when only a part of it need be; and
- the "physical limiter" might be reducing the order of accuracy in crucial regions (near but not at a discontinuity).

On the other hand, for relatively small polynomial orders $(2, 3)$, p-refinement is clearly advantageous for this calculation. As a heuristic for hp-refinement, we propose to use h-refinement as described, with a given l_{max}, and subsequently increase p to l_{max} in all refined layers. Figure 4.7 shows densities at different times using hp-refinement with $l_{max} = 3$. The mixing zone is becoming much more complex than in previous calculations. Even at $t = 1.0$,

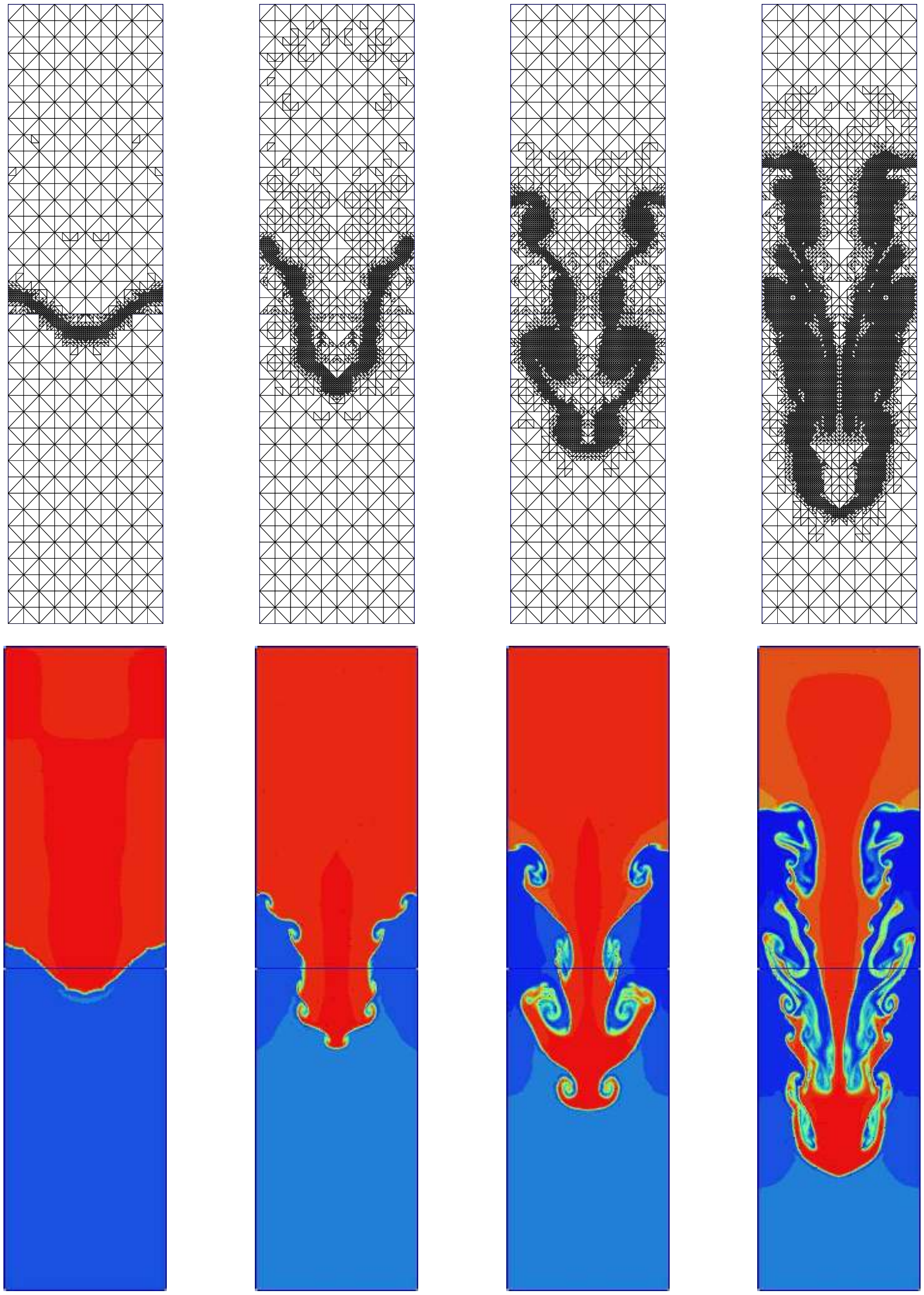

FIG. 4.4. *Meshes (top) and densities (bottom, in logarithmic scale) at* $t = 0.5$*,* $t = 1.0$*,* $t = 1.5$ *and* $t = 1.9$ *(left to right), with, respectively,* 2266*,* 4354*,* 9484 *and* 12816 *triangles and* 27192*,* 52248*,* 113808 *and* 201792 *unknowns. The initial symmetric mesh* M^s *(Figure 4.1) was refined.*

a large number of Helmoltz instabilities have started on the boundaries of the main column, initiating numerous small mixing zones that combine as the flow continues to develop. For $t > 1.5.$, the flow has a chaotic appearence. The symmetric mesh M^s and symmetric en-

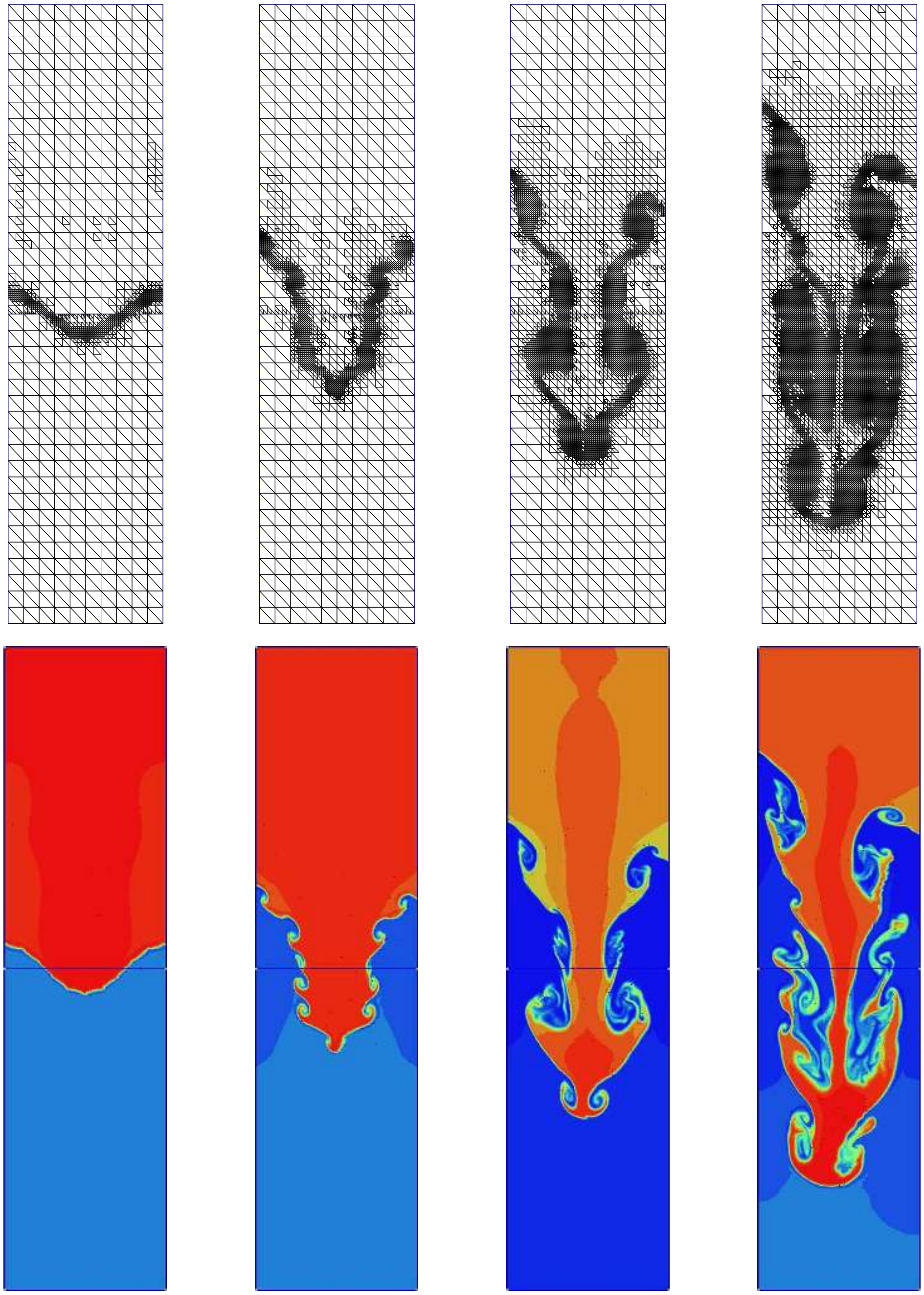

FIG. 4.5. *Meshes (top) and densities (bottom, in logarithmic scale) at* $t = 0.5$*,* $t = 1.0$*,* $t = 1.5$ *and* $t = 1.9$ *(left to right). The initial symmetric mesh* M^{ns} *(Figure 4.1) was refined.*

richement were not sufficient to preserve symmetry. The only cause of asymmetry in the flow are roundoff errors.

5. Conclusions. We have shown that the discontinuous Galerkin method is well suited to adaptive flow computations. The solution limiting and the use of an orthogonal basis allows us to use p-refinement efficiently. A simple error indicator was appropriate to capture the interface with precision and to show details of the flow structure. It is clear that our DGM has to be improved for shock calculations: general purpose limiters are not available for shock problems. Further work will focus on higher-order limiters that maintain accuracy in smooth regions and limit oscillations near discontinuities. One possibility is the adaptive moment limiter of Biswas *et al.* [2] that works well on structured meshes.

The final computation of Figure 4.7 was done on a six processor Linux cluster. Parallel computation with dynamic load balancing will be essential for three-dimensional computations. The DGM software is being combined with a parallel data management system [8, 17] for this purpose.

REFERENCES

[1] F. BASSI AND S. REBAY, *A high-order accurate discontinuous finite element solution of the 2D Euler equations*, Journal of Computational Physics, 138 (1997), pp. 251–285.

[2] R. BISWAS, K. DEVINE, AND J. FLAHERTY, *Parallel adaptive finite element method for conservation laws*, Applied Numerical Mathematics, 14 (1984), pp. 255–283.

[3] B. COCKBURN, S. HOU, AND C. SHU, *The Runge-Kutta local projection discontinuous Galerkin finite element method for the conservation laws IV: the multidimensional case*, Mathemathics Of Computations, 54 (1990), pp. 545–581.

[4] B. COCKBURN, G. KARNIADAKIS, AND C.-W. SHU, eds., *Discontinuous Galerkin Methods*, vol. 11 of Lecture Notes in Computational Science and Engineering, Berlin, 2000, Springer.

[5] B. COCKBURN AND C. SHU, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for the conservation laws II: General framework*, Mathematics of Computations, 52 (1989), pp. 411–435.

[6] P. COLELLA AND H. M. GLAZ, *Efficient solution algorithms for the Riemann problem for real gases*, Journal of Computational Physics, 59 (1985), pp. 264–289.

[7] J. FLAHERTY, R. LOY, M. SHEPHARD, B. SZYMANSKI, J. TERESCO, AND L. ZIANTZ, *Adaptive local refinement with octree load-balancing for the parallel solution of three-dimensional conservation laws*, Journal of Parallel and Distributed Computing, 47 (1997), pp. 139–152.

[8] J. FLAHERTY, R. LOY, M. SHEPHARD, AND J. TERESCO, *Software for the Parallel Adaptive Solution of Conservation Laws by Discontinuous Galerkin Methods*, Lecture Notes in Computational Science and Engineering, Springer Verlag, 2000, pp. 113–124.

[9] B. FRYXELL, K. OLSON, P. RICKER, F. TIMMES, M. ZINGALE, D. LAMB, R. R. P. MACNEICE, J. TRURAN, AND H. TUFO, *Flash: An adaptive mesh hydrodynamics code for modeling astrophysical thermonuclear flashes*, accepted to Journal ApJS, (2000).

[10] F. FURTADO, J. G. ANDJ. GROVE, X.-L. LI, W. LINDQUIST, R. MENIKOFF, D. SHARP, AND Q. ZHANG, *Front tracking and the interaction of nonlinear hyperbolic waves*, Lecture Notes in Engineering, 43 (1989), pp. 99–111.

[11] J. GLIMM, J. GROVE, X. LI, W. OH, AND D. C. TAN, *The dynamics of bubble growth for Rayleigh-Taylor unstabe interfaces*, Physics of Fluids, 31 (1988), pp. 447–465.

[12] B. V. LEER, *Flux vector splitting for the Euler equations*, tech. report, ICASE Report, NASA Langley Research Center, 1995.

[13] P. LESAINT, *Sur la résolution des systèmes hyperboliques du premier ordre par la méthode des éléments finis*, PhD thesis, Université Pierre et Marie Curie, 1975.

[14] R. LOHNER, *Mesh adaptation in fluid mechanics*, Engineering Fracture Mechanics, 50 (1995), pp. 819–847.

[15] P. NITHIARASU AND O. C. ZIENKIEWICZ, *Adaptive mesh generation for fluid mechanics problems*, International Journal for Numerical Methods in Engineering, 47 (2000), pp. 624–662.

[16] W. REED AND T. HILL, *Triangular mesh methods for the neutron transport equation*, Tech. Report LA-UR-73-479, Los Alamos Scientific Laboratory, (1973).

[17] J.-F. REMACLE, O. KLAAS, J. E. FLAHERTY, AND M. S. SHEPHARD, *A parallel algorithm oriented mesh database*, to appear in Engineering With Computers, (2002).

[18] J.-F. REMACLE, K. PINCHEDEZ, J. FLAHERTY, AND M. SHEPHARD, *An efficient local time stepping-discontinuous galerkin scheme for adaptive transient computations*, Submitted to Computer Methods in Applied Mechanics and Engineering, (2002).

[19] P. ROE, *Approximate Riemann solvers, parameter vectors and difference schemes*, Journal of Computational Physics, 43 (1981), pp. 357–372.

[20] T. Warburton, S. Sherwin, and G. Karniadakis, *Spectral basis functions for 2D hybrid hp elements*, SIAM Journal on Scientific Computing, 20 (1999), pp. 1671–1695.

[21] P. Woodward and P. Colella, *The numerical simulation of two-dimensional fluid flow with strong shocks*, Journal of Computational Physics, 54 (1984), pp. 115–173.

[22] Y.-N. Young, H. Tufo, A. Dubey, and R. Rosner, *On the miscible Rayleigh-Taylor instability*, Under consideration for publication in J. Fluid Mech., (2000).

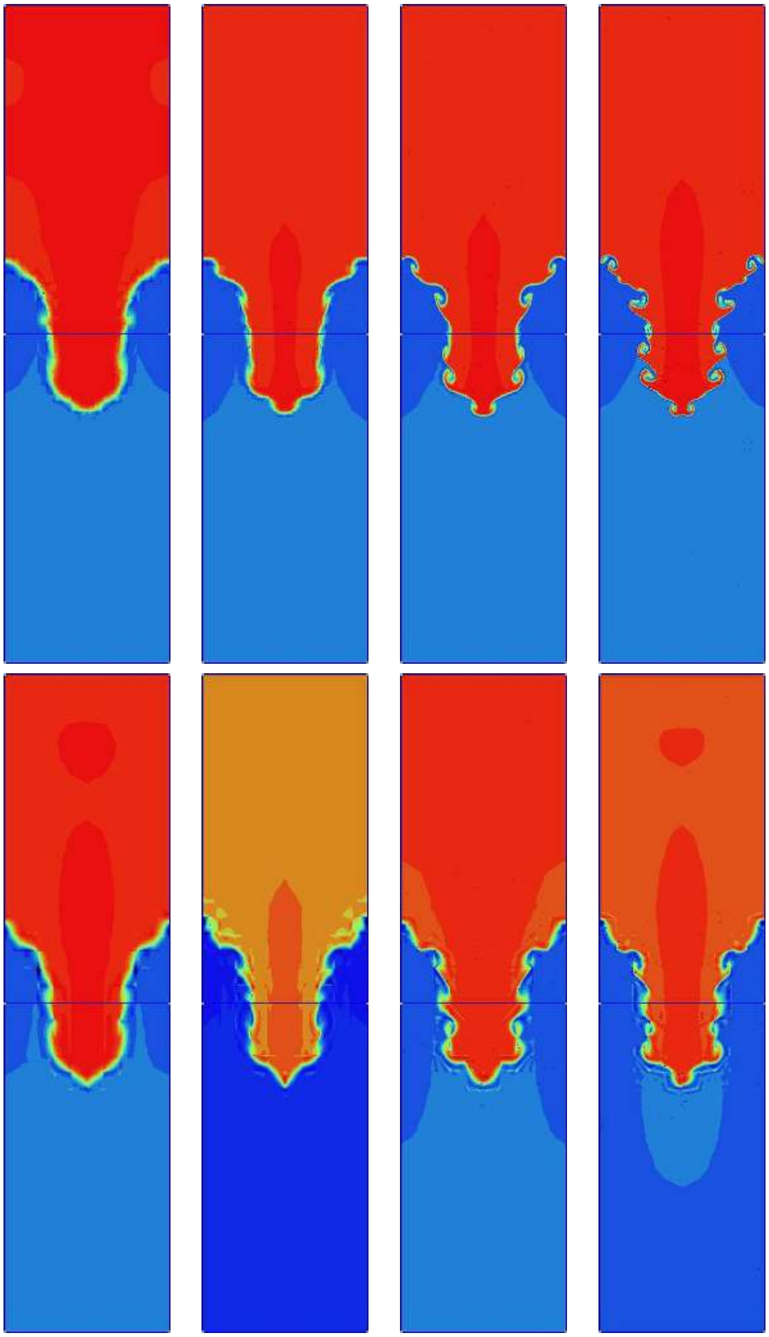

FIG. 4.6. *Densities (in logarithmic scale) at* $t = 1.0$ *for* $l_{max} = 1, 2, 3, 4$ *(left to right) using h-refinement (top) and p-refinement (bottom)*

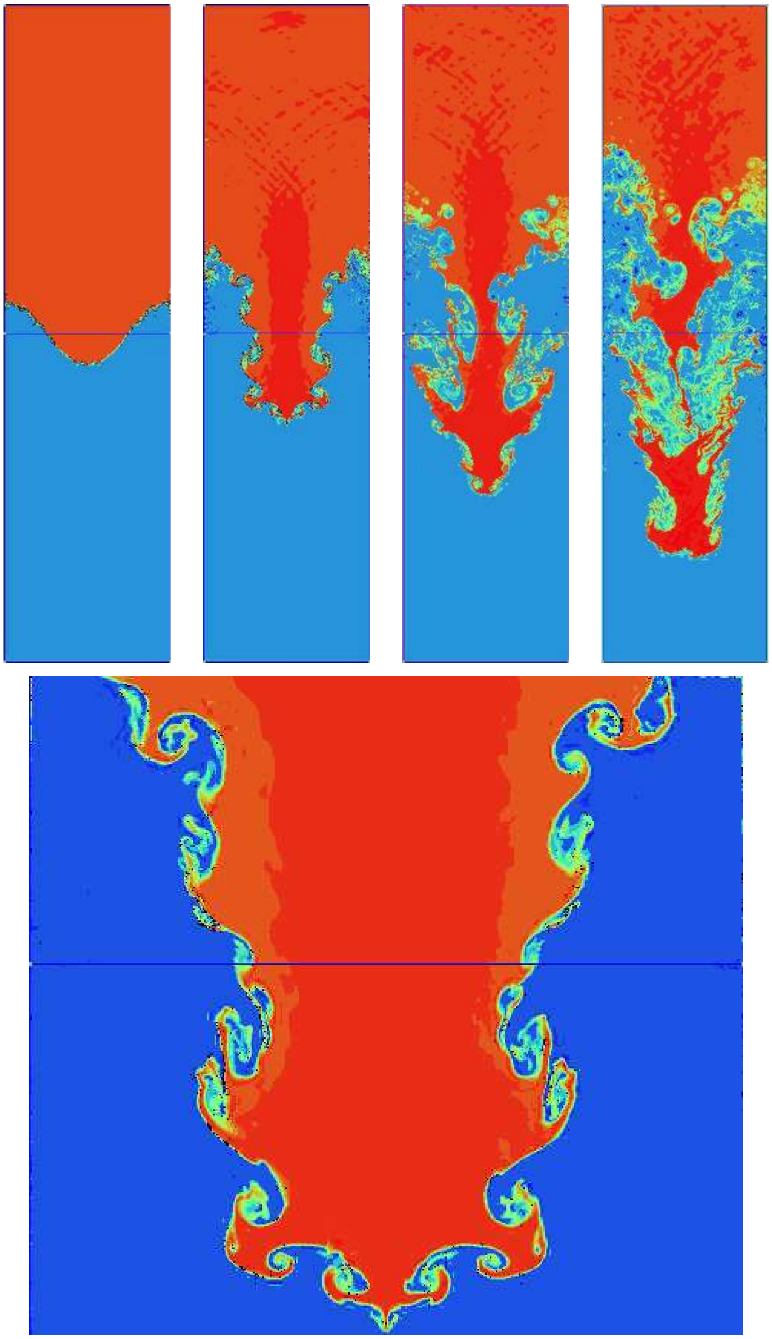

FIG. 4.7. *Densities (top, in logarithmic scale) at* $t = 0.5$*,* $t = 1.0$*,* $t = 1.5$ *and* $t = 1.9$ *(left to right) with respectively* 162848*,* 214176*,* 621312 *and* 793792 *unknowns and on bottom. Close up of the spike tip at* $t = 1$ *(bottom).*