

 Open access • Journal Article • DOI:10.1002/NME.1196

Anisotropic adaptive simulation of transient flows using discontinuous Galerkin methods — [Source link](#)

Jean-François Remacle, Xiangrong Li, Mark S. Shephard, Joseph E. Flaherty

Institutions: Université catholique de Louvain, Rensselaer Polytechnic Institute

Published on: 21 Feb 2005 - International Journal for Numerical Methods in Engineering (Wiley)

Topics: Discontinuous Galerkin method, Finite element method, Hessian matrix, Discretization and Euler equations

Related papers:

- [Anisotropic mesh adaptation for CFD computations](#)
- [Anisotropic unstructured mesh adaption for flow simulations](#)
- [Adaptive remeshing for compressible flow computations](#)
- [Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part I: general principles](#)
- [3D anisotropic mesh adaptation by mesh modification](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/anisotropic-adaptive-simulation-of-transient-flows-using-3a6yach710>

Anisotropic Adaptive Simulation of Transient Flows using Discontinuous Galerkin Methods

Jean-François Remacle^{1*}, Xiangrong Li², Mark S. Shephard² and Joseph E. Flaherty²

¹ *Département de Génie Civil et Environnemental, Université Catholique de Louvain, Bâtiment Vinci, Place du Levant 1, B-1348 Louvain-la-Neuve, Belgium*

² *Scientific Computation Research Center, Rensselaer Polytechnic Institute, Troy, NY, 12180-3590, U.S.A*

SUMMARY

An anisotropic adaptive analysis procedure based on a discontinuous Galerkin finite element discretization and local mesh modification of simplex elements is presented. The procedure is applied to transient 2- and 3-dimensional problems governed by Euler's equation. A smoothness indicator is used to isolate jump features where an aligned mesh metric field is specified. The mesh metric field in smooth portions of the domain is controlled by a Hessian matrix constructed using a variational procedure to calculate the second derivatives. The transient examples included demonstrate the ability of the mesh modification procedures to effectively track evolving interacting features of general shape as they move through a domain. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: anisotropic adaptive, discontinuous Galerkin, mesh modification

1. INTRODUCTION

The appropriate means to ensure that a mesh-based numerical analysis procedure produces the most effective solution results is to apply an adaptive solution strategy. Efforts on the development of these techniques have been underway for over twenty years and have provided a number of important theoretical and practical results. However, these methods have not yet found their way into common practice for a number of reasons. Among the reasons for the slow acceptance is the lack of clear evidence that their implementations to be able to deal with entirely general domains and solution fields in a computationally effective manner. In cases where the solution field is characterized by strong directional gradients, the effective solution requires the adaptive creation of anisotropic mesh configurations. The paper presents a set of procedures to create adaptively defined anisotropic meshes over general 2- and 3-dimensional domains and demonstrates its application in transient flow simulations.

The three ingredients of an anisotropic adaptive procedure are:

*Correspondence to: J.F. Remacle, Département de Génie Civil et Environnemental, Université Catholique de Louvain, Bâtiment Vinci, Place du Levant 1, B-1348 Louvain-la-Neuve, Belgium.

†Email: remacle@gce.ucl.ac.be

- the equation discretization technology
- the anisotropic mesh correction indication procedures that use the analysis results to determine where and how modify the mesh to reach the desired level of accuracy, and
- the anisotropic mesh adaptation procedure to create a mesh configuration consistent with the mesh distribution the correction indication procedures have defined.

A number of finite element and finite volume discretization technologies are amenable to use with anisotropic meshes. In the present paper the applications considered are flow problems modeled using conservation laws and characterized by having moving features such as shocks. Therefore, the discontinuous Galerkin (DG) [15] finite element formulation given in Section 2 was selected for equation discretization. In addition to being well suited to the resolution of solution fields with discontinuities, the DG formulations provide flexibility in the selection of basis function leading to more effective numerical solution and can be effectively parallelized due to the order independent nearest neighbor only interactions [39]. One complication of the application of DG methods is their discontinuous nature does complicate the effective calculation of the second order derivative quantities used by most anisotropic adaptive procedure. The approach used in the current paper to address the evaluation of these derivatives is discussed in Section 3.2.

Recently a number of investigators have begun to consider the various components of the construction of anisotropic adaptive analysis procedures [3,8,14,20,21,22,33,35,38,41,45,46]. Ideally an adaptive analysis procedure would employ a bounded estimate [2,4] of the discretization error. Since such estimates are based on elemental level contributions, they have typically been used to determine where and how to improve the mesh when isotropic mesh refinement is used. For many classes of equations of interest bounded error estimates are not yet available. However, this does not preclude the use of simple error indicators based on various gradient measurements from providing useful adaptive procedures [10,19,29,36]. A second complexity that arises in anisotropic adaptive procedures is that even when available, the bounded error estimates are typically scalar norms that do not provide the directional information needed to define the desired mesh anisotropy. Therefore, anisotropic adaptive procedures employ the full set of second order derivatives (Hessian matrix) [8,20,41] or examine derivatives in the direction of specific mesh entities (typically edges) [33,46] to obtain directional information on the desired mesh layout. For purposes of this discussion the term mesh correction indicator is used to describe this information after it had been scaled to define the actual anisotropic element sizes desired over the domain. Section 3 discusses the procedures that constitute the mesh correction indicator used in this paper to define the anisotropic adaptive mesh size field.

Given the new mesh size field, there are the two means to construct a mesh that satisfies it. They are to regenerate the mesh against that mesh size field [3,8,20,41], or to perform appropriate local mesh modifications to match the desired mesh size field [33,35,38,46]. The remeshing based techniques have the advantage of not being constrained by the existence of the previous mesh entities in the construction of the new anisotropic mesh configuration. However, these methods do incur the cost of a complete mesh generation step and, in many applications, require the application of a solution field transfer process between meshes which is both expensive and subject to accuracy loss. Mesh modification procedures can be executed quickly with more controlled solution transfer procedures. However, with a only limited set of mesh modification operations allowed, the mesh configurations are not optimal. Of course, with the inclusion of a "full set" of mesh modification operations (e.g., like the procedure in reference [35]) the differences in the final mesh configuration between remeshing and mesh modification can essentially be eliminated. The procedure used in this work (see Section 4) applies a "full set" of mesh modification operators employing a set of intelligent heuristics to

effectively determine the appropriate mesh modifications to obtain the desired mesh configurations. Section 5 presents a set of 2- and 3-dimensional transient flow simulations to demonstrate the power of the method to solve flow problems with complex evolving features.

2. DISCONTINUOUS GALERKIN FORMULATION

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 $\mathbf{u}(\Omega, t) : \mathbb{R}^3 \times \mathbb{R} \rightarrow L^2(\Omega)^m = V(\Omega)$ as the solution of a *system of conservation laws*

$$\partial_t \mathbf{u} + \mathbf{div} \vec{\mathbf{F}}(\mathbf{u}) = \mathbf{r}. \quad (1)$$

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

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

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

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

With the aim of constructing a Galerkin form of (1), let

$$(x, y)_\Omega = \int_\Omega x y \, dv$$

and

$$\langle x, y \rangle_{\partial\Omega} = \int_{\partial\Omega} x y \, ds$$

denote the standard $L^2(\Omega)$ and $L^2(\partial\Omega)$ scalar products respectively. Multiply equation (1) by a test function $\mathbf{w} \in V(\Omega)$, integrate over Ω and use the divergence theorem to obtain the following variational formulation

$$(\partial_t \mathbf{u}, \mathbf{w})_\Omega - (\vec{\mathbf{F}}(\mathbf{u}), \nabla \mathbf{w})_\Omega + \langle \vec{\mathbf{F}}(\mathbf{u}) \cdot \vec{n}, \mathbf{w} \rangle_{\partial\Omega} = (\mathbf{r}, \mathbf{w})_\Omega, \quad \forall \mathbf{w} \in V(\Omega). \quad (2)$$

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

$$\mathcal{T}_e = \bigcup_{e=1}^{\mathcal{N}_e} e \quad (3)$$

called a mesh. The function space $V(\Omega)$ containing the solution of (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 $\mathbb{P}^p(e)$ of order p):

$$V_e(\mathcal{T}_e) = \{ \mathbf{u} \mid \mathbf{u} \in L^2(\Omega)^m, \mathbf{u}_e \in \mathbb{P}^p(e)^m = V_e(e) \}. \quad (4)$$

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

$$(\partial_t \mathbf{u}_e, \mathbf{w})_e - (\vec{\mathbf{F}}(\mathbf{u}_e), \nabla \mathbf{w})_e + \langle \vec{\mathbf{F}}_n, \mathbf{w} \rangle_{\partial e} = (\mathbf{r}, \mathbf{w})_e, \quad \forall \mathbf{w} \in V_e(e). \quad (5)$$

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 *numerical 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 $\vec{\mathbf{F}} \in \mathbf{H}(\text{div}, \Omega)^m$, 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 (5) becomes

$$(\partial_t \mathbf{u}_e, \mathbf{w})_e - (\vec{\mathbf{F}}(\mathbf{u}_e), \nabla \mathbf{w})_e + \sum_{k=1}^{n_e} \langle \mathbf{F}_n(\mathbf{u}_e, \mathbf{u}_{e_k}), \mathbf{w} \rangle_{\partial e_k} = (\mathbf{r}, \mathbf{w})_e, \quad \forall \mathbf{w} \in V_e(e), \quad (6)$$

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 [23, 47]. It is usual to define the trace as the solution \mathbf{u}_R of a Riemann problem across ∂e_k . We have then $\mathbf{F}_n(\mathbf{u}_e, \mathbf{u}_{e_k}) = \vec{\mathbf{F}}(\mathbf{u}_R) \cdot \vec{n}$. Herein, we consider problems with strong shocks [47, 17]. An exact Riemann solver is used to compute the numerical fluxes and a slope limiter [6] is used to produce monotonic solutions when polynomial degrees $p > 0$ are used.

The choice of a basis for $V_e(e)$ is an important issue in constructing an efficient method. Because the field is discontinuous, there is substantial freedom in the selection of the elemental basis. Here, we chose the L^2 -orthogonal basis described in [39] as a basis of $P(e)$:

$$P(e) = \{b_1, \dots, b_k\} \quad (7)$$

where

$$(b_i, b_j)_e = \delta_{i,j}.$$

For the time discretization, we use the local time stepping procedure described in [40] that allows to use overall time steps more than 20 times bigger than the classical stability limit of explicit schemes.

We will present the results of some compressible inviscid flow problems involving the solution of the Euler equations [24] by a DG method. The three-dimensional Euler equations have the form (1) with

$$\mathbf{u} = \{\rho, \rho v_x, \rho v_y, \rho v_z, E\}^t \quad (8)$$

$$\vec{\mathbf{F}}(\mathbf{u}) = \{\rho \vec{v}, \rho v_x \vec{v} + P \vec{e}_x, \rho v_y \vec{v} + P \vec{e}_y, \rho v_z \vec{v} + P \vec{e}_z, (\rho E + P) \vec{v}\}^t \quad (9)$$

$$\mathbf{r} = \mathbf{0}. \quad (10)$$

Here ρ is the fluid density, \vec{v} the velocity, E the internal energy, P the pressure and \vec{e}_x , \vec{e}_y and \vec{e}_z are the unit vectors in the x , y and z directions, respectively. An equation of state of the form $P = P(\rho, E)$ is also necessary to close the system. The DG method and the associated software [39] may be used for any equation of state which only enters the numerical method through the calculation of the numerical flux. Here, we have chosen the perfect gas equation of state

$$P = (\gamma - 1) \rho \left[E - \frac{\|\vec{v}\|^2}{2} \right] \quad (11)$$

with the gas constant $\gamma = 1.4$.

3. ANISOTROPIC MESH CORRECTION INDICATION

3.1. Approach Taken

The goal of the mesh correction indication process is to determine the anisotropic mesh configuration that will most effectively provide the level of accuracy required for the parameters of interest.

The literature on error estimation techniques (e.g., [2, 4]) does provide the mathematical tools and techniques to reliably approach this goal for specific classes of equations under specific limitations on the relationship of the methods to analyze the discretization errors and to improve the mesh. However, these procedures do not yet provide all the ingredients needed for more complex set of equations such as the hyperbolic conservation equations considered here. In particular, the ability to bound the discretization error estimates in appropriate norms and to prove optimal anisotropic mesh configurations is not yet available. On the other hand, it is well recognized that the application of adaptive analysis procedures for these problems yields far superior results to non-adaptive methods. Therefore, the strategy adopted in the present paper is to construct the anisotropic mesh correction indicator in terms of a complete mesh metric field defined over the domain of the analysis and to construct this mesh metric field using a combination of best available methods.

The most direct definition of an anisotropic mesh metric field is one that defines the mapping of an ellipsoid into a unit sphere in terms of a diagonal distortion matrix, where the diagonal terms correspond to the lengths of the principal axes of the ellipsoid, times a rotation matrix that accounts for the orientation of the ellipsoid. When used for constructing the anisotropic mesh size field, lengths of the principal axes are interpreted as the desired mesh edge lengths at that location.

$$Q(x, y, z) = \underbrace{\begin{bmatrix} 1/h_1 & 0 & 0 \\ 0 & 1/h_2 & 0 \\ 0 & 0 & 1/h_3 \end{bmatrix}}_{\text{distortion}} \cdot \underbrace{\begin{bmatrix} \vec{e}_1 \\ \vec{e}_2 \\ \vec{e}_3 \end{bmatrix}}_{\text{rotation}} \quad (12)$$

where $\vec{e}_1, \vec{e}_2, \vec{e}_3$ are orthogonal unit vectors associated with the principal axes of the ellipsoid at point (x, y, z) , and h_1, h_2, h_3 are the desired mesh edge lengths along these three axes.

To date the most common approach to the definition of the mesh metric field for adaptive mesh construction is to relate it to the Hessian of an appropriate solution variable u [8, 20, 41]:

$$H(u) = \begin{bmatrix} \frac{\partial^2 u}{\partial x^2} & \frac{\partial^2 u}{\partial x \partial y} & \frac{\partial^2 u}{\partial x \partial z} \\ \frac{\partial^2 u}{\partial y \partial x} & \frac{\partial^2 u}{\partial y^2} & \frac{\partial^2 u}{\partial y \partial z} \\ \frac{\partial^2 u}{\partial z \partial x} & \frac{\partial^2 u}{\partial z \partial y} & \frac{\partial^2 u}{\partial z^2} \end{bmatrix} \quad (13)$$

and construct $Q(x, y, z)$ by decomposing, scaling $H(u)$ and bounding the maximum desired mesh edge lengths:

$$Q(x, y, z) = \begin{bmatrix} \sqrt{\lambda'_1} & 0 & 0 \\ 0 & \sqrt{\lambda'_2} & 0 \\ 0 & 0 & \sqrt{\lambda'_3} \end{bmatrix} \cdot \begin{bmatrix} \vec{e}_1 \\ \vec{e}_2 \\ \vec{e}_3 \end{bmatrix} \quad (14)$$

with

$$\lambda'_i = \max \left(\phi(x, y, z) |\lambda_i|, \frac{1}{h_{max}^2} \right) \quad \forall i \in (1, 2, 3) \quad (15)$$

where

- $|\lambda_i|$ is the i^{th} absolute eigenvalue of the Hessian matrix H ;
- \vec{e}_i is the i^{th} unit eigenvector of H ;
- $\phi(x, y, z)$ is a scale factor at point (x, y, z) , determined in terms of an error estimate/indicator (e.g. leading element interpolation error) to equilibrate the distribution of the error;
- h_{max} is user defined maximal allowable mesh edge length in the mesh. Since Hessian $H(u)$ can be singular, it is needed to apply h_{max} in case λ_i is zero or close to zero.

A variety of arguments have been given as to the rationale for using the second derivative information of the Hessian matrix in the construction of the anisotropic mesh metric field. The most compelling is to consider basic interpolation theory coupled with an equivalence of norm argument to show the error in the interpolant is equivalent to a norm of interest for the finite element methods. In the simplest possible terms, the error in a polynomial interpolant is proportional to the derivatives of order equal to the first order polynomial the interpolant can not exactly represent. In the case where piecewise linear finite elements are used, the interpolation error is proportional to second derivatives. Kunert [22] provides some degree of analysis of the use of the Hessian matrix anisotropic in adaptive analysis including pointing some of the critical limitations of its use. The analysis by Rachowicz [38] focuses on the L^2 error norm for interpolation on an anisotropic mesh showing that the error is associated with the $p + 1$ derivatives for a p th order interpolant. He further relates this error to an H^1 -seminorm of the finite element solution for the specific case of parallelogram elements [37] in which case the dominant error term is associated with error is associated with the $p + 1$ derivatives when the solution is of sufficient smoothness.

Since the examples presented in this paper are based on piecewise linear L^2 discontinuous finite elements, the Hessian matrix will be employed as a key ingredient in the construction of the mesh metric field in the regions where the solution is smooth. Specific care must be exercised in the definition of this mesh metric field. The most obvious concern is the ability to calculate values to the second derivatives of a discontinuous field. One approach used with C^0 finite element basis is the construction of a "recovered" Hessian [3] using patchwise projection procedures in a manner similar to that used to define the popular Zienkiewicz-Zhu error estimators [48]. Although it may be possible to use a similar approach here, the discontinuous nature of the DG basis makes it questionable. Therefore, the present work employs the variationally based construction procedure of Section §3.2 to evaluate the Hessian matrix in the portions of the domains where the exact solution is assumed to be smooth.

Since the procedure will be applied to the adaptive solution of transient flow problems that contain solution discontinuities (shocks, contact discontinuities and expansion waves), care must be taken in the construction of the mesh metric field. Clearly, it is inappropriate to construct and employ the Hessian matrix in the immediate vicinity of the discontinuities, the locations of which are not known a priori and which move as the transient solution evolves. Therefore, a two step procedure is used to construct the mesh metric field around discontinuities. We first determine the location of the elements crossing discontinuities using the solution smoothness indicator presented in Section §3.3. Then, we define the mesh metric field along the discontinuities using the procedure given in Section §3.4.

Mesh metric fields are constructed here using solutions of compressible flows. The structure of such flows is usually formed of very smooth regions separated by discontinuities (waves). Hessians based on such solutions will generate metric fields with brutal variations of mesh sizes. In Section §3.5, we will indicate how the mesh metric fields over the various portions of the domain are smoothed to produce the final mesh metric field used by the mesh adaptation procedures. The development of an efficient smoothing procedure of the anisotropic metric was a crucial step in the whole process of adaptation.

3.2. Calculation of Hessian Matrix from Discontinuous Fields

It is common in Computational Fluid Dynamics discretized by piecewise linears to use second order derivatives $H(u)$ of a flow variable u in order to compute error indications.

In the classical sense, $\partial^2 u / \partial x^2$ exists only if u is twice derivable. In a more general sense, i.e. in the sense of distributions, functions with less regularity may have second order derivatives. In the context of finite elements, discontinuities of a discrete field u and/or of its derivatives occur only at

element boundaries. The gradient ∇u of a field u discretized with the DGM has a contribution due to inter-element jumps. In the case of classical C^0 finite elements, ∇u can be computed directly i.e. by computing derivatives inside the elements. The hessian $H(u)$ will have a contribution due to inter-element jumps of the normal derivative of u .

In the DGM, the field u itself and all its derivatives are discontinuous between elements. The computation of derivatives requires then careful consideration. To recover some control on the gradient ∇u , it is possible to use the DGM method itself. This method is used when the DGM is used for solving elliptic problems (see e.g. [12]) i.e. when the fluxes $\vec{F}(u, \nabla u)$ are function of ∇u .

Using u , we construct a new field w in a space V_e which is sufficiently regular for computing ∇w in the classical sense (V_e will be defined later). We impose that u and w have the same gradients in a weak sense. The problem consists in finding, in each element e , $w \in V_e(e)$ such that

$$((\nabla u - \nabla w), \nabla w')_e = 0 \quad \forall w' \in V_e(e). \quad (16)$$

If w is the solution of (16), $w + C$ is also a solution if C is a constant. In order to have a unique solution to the problem, we discretize w using the same orthogonal basis $P(e)$ as we used for u (see Equation (7)). We choose *a priori* the value of the first coefficient $c_1 = 1$ of the interpolation

$$w = b_1 + \sum_{i=2}^{\dim P(e)} c_i b_i$$

because the constant part of a L^2 orthogonal space is contained into its first function.

We then integrate (16) by parts to obtain the jump of u and w on ∂e

$$((w - u), \nabla \cdot (\nabla w'))_e + \sum_{k=1}^{n_e} \langle \{u\} - \{w\}, \nabla w' \rangle_{\partial e_k} = 0 \quad \forall w \in V_e(e). \quad (17)$$

Fields u and w on ∂e are both multi-valued. A numerical value for $u|_{\partial e}$ and $w|_{\partial e}$ has to be chosen. Here, we use the average fluxes

$$\{x\} = \vec{n} \frac{x + x_{e_k}}{2}$$

in evaluating (17). Other choices have been proposed [12]. A second integration by parts yields jumps of u and w across ∂e

$$((\nabla u - \nabla w), \nabla w')_e + \sum_{k=1}^{n_e} \langle \llbracket u \rrbracket - \llbracket w \rrbracket, \nabla w' \rangle_{\partial e_k} = 0 \quad \forall w' \in V_e(w) \quad (18)$$

where

$$\llbracket x \rrbracket = \vec{n} \frac{x - x_{e_k}}{2}$$

is the half flux jump. Without any other assumption on the regularity of w , the solution of (18) is $u = w$. By choosing w verifying

$$(\llbracket w \rrbracket, \nabla w')_e = 0 \quad \forall w', \quad (19)$$

i.e. choosing w so that there is no flux jump (in a weak sense) of the field through faces, equation (18) transforms to

$$(\nabla w, \nabla w')_e = (\nabla u, \nabla w')_e + \sum_{k=1}^{n_e} \langle \llbracket u \rrbracket, \nabla w' \rangle_{\partial e_k} \quad \forall w' \in V_e(e). \quad (20)$$

The hypothesis in (19) requires further consideration: assumption (19) is similar to the one used for building the Crouzeix-Raviart finite element family [11]. Such non conforming element space provides interpolations that are continuous at edges mid-points only. Those finite elements generate stable gradients.

For second order derivatives, we could proceed exactly the same way and recover second order derivatives by computing jumps of stable first order derivatives ∇w along x , y and z directions and build up the Hessian matrix. This technique, although general and working properly on regular grids, does not give satisfactory results on highly distorted grids, like the ones we intend to generate and use. Consequently, we have used a more classical technique for reconstructing Hessians at mesh vertices. At each vertex n of coordinates x_n, y_n, z_n , we reconstruct three continuous linear fields

$$\begin{aligned}\left. \frac{\partial u}{\partial x} \right|_n &= a_1^n + a_2^n(x - x_n) + a_3^n(y - y_n) + a_4^n(z - z_n) \\ \left. \frac{\partial u}{\partial y} \right|_n &= b_1^n + b_2^n(x - x_n) + b_3^n(y - y_n) + b_4^n(z - z_n) \\ \left. \frac{\partial u}{\partial z} \right|_n &= c_1^n + c_2^n(x - x_n) + c_3^n(y - y_n) + c_4^n(z - z_n)\end{aligned}$$

from the discontinuous gradients ∇w . We apply this scheme on every element $e_i, i = 1, \dots, N$ of centroids x_i, y_i, z_i surrounding vertex n to obtain the 3 systems. For computing derivatives along x , we build

$$\underbrace{\begin{bmatrix} 1 & x_1 - x_n & y_1 - y_n & z_1 - z_n \\ 1 & x_1 - x_n & y_1 - y_n & z_1 - z_n \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_N - x_n & y_N - y_n & z_N - z_n \end{bmatrix}}_{[B]} \underbrace{\begin{pmatrix} a_1^n \\ a_2^n \\ a_3^n \\ a_4^n \end{pmatrix}}_{(a)} = \underbrace{\begin{pmatrix} \partial_x w(x_1, y_1, z_1) \\ \partial_x w(x_1, y_1, z_1) \\ \vdots \\ \partial_x w(x_N, y_N, z_N) \end{pmatrix}}_{(r_x)} \quad (21)$$

We can solve the associated normal equation

$$[B]^T [B](a) = [B]^T (r_x)$$

and find coefficients (a) . We can proceed the same way for finding out coefficients (b) and (c) by solving the two other normal equations

$$[B]^T [B](b) = [B]^T (r_y)$$

and

$$[B]^T [B](c) = [B]^T (r_z)$$

From these results, Hessian computation is straightforward. We have, for example then that

$$\frac{\partial^2 u}{\partial x^2} = a_2^n, \quad \frac{\partial^2 u}{\partial y^2} = b_3^n$$

or

$$\frac{\partial^2 u}{\partial x \partial y} = \frac{1}{2}(a_3^n + b_2^n).$$

3.3. Isolation of Discontinuities using a Smoothness Indicator

The main challenge of solving hyperbolic problems such as compressible gas dynamics is that the solutions is able to develop discontinuities in finite time even for smooth initial data. It has been shown [24] that only schemes that are of first order of accuracy are able to produce monotonic solutions when discontinuities are present. First order schemes produce too much numerical dissipation and do not exhibit required resolution for convection dominated problems (i.e. problems with small physical dissipation).

The spurious oscillations produced near discontinuities by a higher-order method such as the DGM may amplify in time (especially near shocks) and cause the solution to become unbounded. It is crucial to be able to control and eliminate the spurious oscillations introduced by higher order schemes.

Procedures to suppress oscillations near discontinuities are called limiters [43, 44, 16, 5]. Limiters tends to reduce the accuracy of solutions to first order where they are applied. With an adaptive strategy, discontinuities are captured by reducing element sizes at their vicinity accounting for alignment with the discontinuity and directional variation differences in local solution information. The limiter is only applied in the one or two layers of elements crossing discontinuities.

We introduce here a procedure that allows us to detect discontinuities. Consider element e of boundary ∂e . Solving the DGM implies the computation of a numerical flux $\mathbf{F}_n(\mathbf{u}_e, \mathbf{u}_{e_k}) = \vec{\mathbf{F}}(\mathbf{u}_R) \cdot \vec{n}$ (cfr. Section §2) where \mathbf{u}_R is the solution of the Riemann problem at the boundary of the element ∂e . If $\mathbf{u}_{e,x}$ is the exact solution of (1) and h_e is the size of element e (e.g. the radius of the circumsphere of a tetrahedron) it has been proven in [1] that the following result holds ¹

$$\frac{1}{|\partial e|} \int_{\partial e} (\mathbf{u}_R - \mathbf{u}_{e,x}) ds = \mathcal{O}(h_e^{2p+1}). \quad (22)$$

The super-convergence result (22) implies that the solution \mathbf{u}_R of the Riemann problem at element interfaces is, in average, much closer to the exact solution than elementary solution \mathbf{u}_e . For linear problems, the solution of the Riemann problem is the upwind value of \mathbf{u}_e at boundary ∂e . On one element e , the downwind values (i.e. the upwind values of the next element) are the ones which are super-convergent. ²

We consider the following elemental quantity

$$I_e = \int_{\partial e} (\mathbf{u}_e - \mathbf{u}_R) ds = \int_{\partial e} (\mathbf{u}_e - \mathbf{u}_{e,x}) ds + \int_{\partial e} (\mathbf{u}_{e,x} - \mathbf{u}_R) ds. \quad (23)$$

Due to the superconvergence result (22), the second integral is $\mathcal{O}(h^{2(p+1)})$ while the first is $\mathcal{O}(h^{p+2})$. Thus, $I_e = \mathcal{O}(h^{p+2})$ across edges (2D case) or faces (3D case) where the solution is smooth. If \mathbf{u} is discontinuous in the immediate vicinity of ∂e , then either or both of $\mathbf{u}_{e,x} - \mathbf{u}_e$ and $\mathbf{u}_{e,x} - \mathbf{u}_R$ are $\mathcal{O}(1)$; hence,

$$I_e = C \begin{cases} h^{p+2}, & \text{if } \mathbf{u}|_{\partial e} \text{ is smooth} \\ h, & \text{if } \mathbf{u}|_{\partial e} \text{ is discontinuous} \end{cases}. \quad (24)$$

¹This result was proven in the case of linear problems and was tested successfully on a non-linear Burgers equation.

²We claim here the following conjecture about this superconvergence result : the convergence of the solution at downwind is necessary for the DGM to produce convergent result. If this result was not holding, truncation errors would propagate along characteristics of the flow.

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

$$\mathcal{I}_e = \frac{|\int (\mathbf{u} - \mathbf{u}_R) ds|}{h^{(p+1)/2} |\partial e| \|\mathbf{u}\|}. \quad (25)$$

In examples, we choose h_e 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. We have then that

$$\mathcal{I}_e = C \begin{cases} h^{(p+1)/2}, & \text{if } \mathbf{u}|_{\partial e} \text{ is smooth} \\ h^{-(p+1)/2}, & \text{if } \mathbf{u}|_{\partial e} \text{ is discontinuous} \end{cases}. \quad (26)$$

Consequently, $\mathcal{I}_e \rightarrow 0$ as either $h \rightarrow 0$ or $p \rightarrow \infty$ in smooth solution regions, whereas $\mathcal{I}_e \rightarrow \infty$ near a discontinuity. Thus, the discontinuity detection scheme is

$$\begin{cases} \text{if } \mathcal{I}_e > 1, & \mathbf{u} \text{ is discontinuous} \\ \text{if } \mathcal{I}_e < 1, & \mathbf{u} \text{ is smooth} \end{cases}. \quad (27)$$

In order to illustrate the efficiency of the smoothness indicator, we consider the following problem. The geometry of the domain is a square of size 1×1 centered at $x = 0$ and $y = 0$. The domain is initially divided into four quadrants. Quadrant 1 is the upper right, 2 the upper left, 3 the lower left and 4 the lower right. All boundary conditions are transmitting (we copy the interior data perpendicular to the boundary). We initialize each quadrant with the quantities given in Table 2. In this problem

	1	2	3	4
ρ	1.0	2.0	1.0	3.0
v_x	0.75	0.75	-0.75	-0.75
v_y	-0.5	0.5	0.5	-0.5
P	1.0	1.0	1.0	1.0

Table I. Initial conditions for the four-contact Riemann problem.

four contact discontinuities are rotating around the center of the square creating shear instabilities. The cascade to small scales will not be moderated by viscosity effects because our model is inviscid. For that reason, the more refinement we will allow, the smaller scales we will get. The structure of the problem is simple. The only feature of the flow is an interface (i.e. a discontinuity) between fluids of different densities. The interface rotates; and, while it gets unstable, grows in size in a fractal manner. Our aim here is then to track this moving and growing interface using h-refinement. The discontinuity detector is used to find those elements crossing the interface. Those elements are refined in a non-conforming way (i.e., quads were recursively split into 4) with a maximum level of refinement of 6. The maximum size reduction is $\frac{1}{2^6} = \frac{1}{64}$. With an initial mesh of 40×40 , the maximal mesh would contain 2560×2560 quadrilaterals. In our computation, we have used piecewise bilinear elements. There are 4 shape functions per unknown and 4 unknown fields for 2D Euler equations. There are then 16 unknowns per quadrilateral. The maximum mesh would then require $2560 \times 2560 \times 16 = 104,857,600$ unknowns.

Figure 2 shows density contours at $t = 0.3$. The associated mesh shown on Figure 1 was obtained after having performed 300 mesh adaptations i.e. one adaptation every 0.001 seconds. There are around 250,000 quadrilaterals in the pictured adapted mesh which makes around four million degrees of freedom.

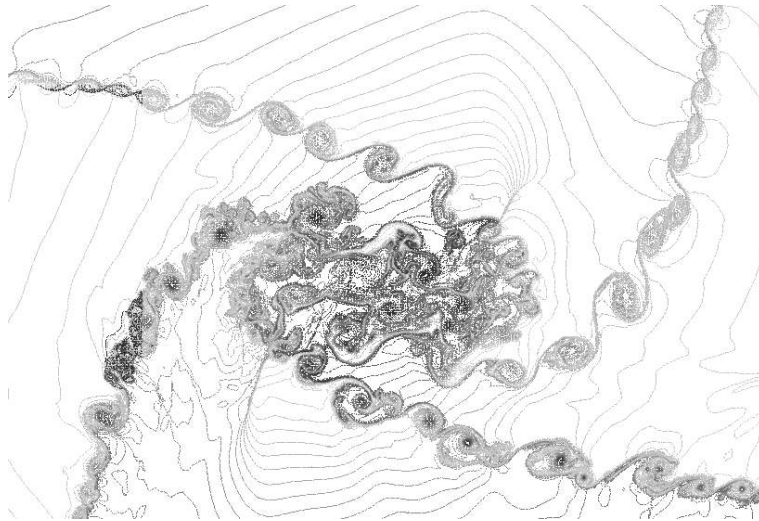


Figure 1. Density contours at $t = 0.3$ for the four-contact problem. View shows a zoom of the central area for the mostly refined computation.

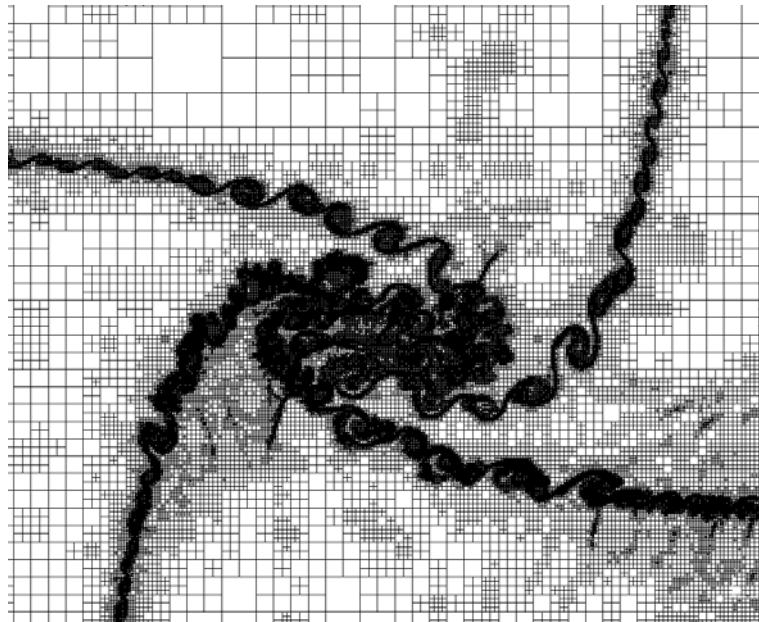


Figure 2. Refined mesh at $t = 0.3$ for the four-contact problem. View shows a zoom of the central area for the mostly refined computation.

3.4. Anisotropic Mesh Construction Across Jump Features

In portions of domain across discontinuities, we have shown in Section §3.3 that the discontinuities can be detected to provide useful information for an adaptive process. Due to the discontinuities, the element discretization error cannot be controlled (i.e. bounded) in elements crossing a discontinuity in the classic sense.

Let us consider the following function

$$\mathcal{H}(x, \alpha) = \left(\frac{1}{2} + \frac{\tan^{-1}(\alpha x)}{\pi} \right) \tag{28}$$

that models a discontinuous function with a jump of 1 at $x = 0$. At the limit $\lim_{\alpha \rightarrow \infty}$, \mathcal{H} tends to the Heaviside function. The second order derivative of (28) gives

$$\frac{\partial^2 \mathcal{H}(x, \alpha)}{\partial x^2} = \frac{2\alpha^2 x}{(1 + (\alpha x)^2)^2 \pi} \tag{29}$$

If we take α bounded in (28), we obtain a function that approximates the discontinuity with the same kind of behavior as a DGM numerical solution. The second order derivative (29) is then equal to 0 at $x = 0$, going from large positive values for $x < 0$ to large negative values at $x > 0$ (see Figure 3).

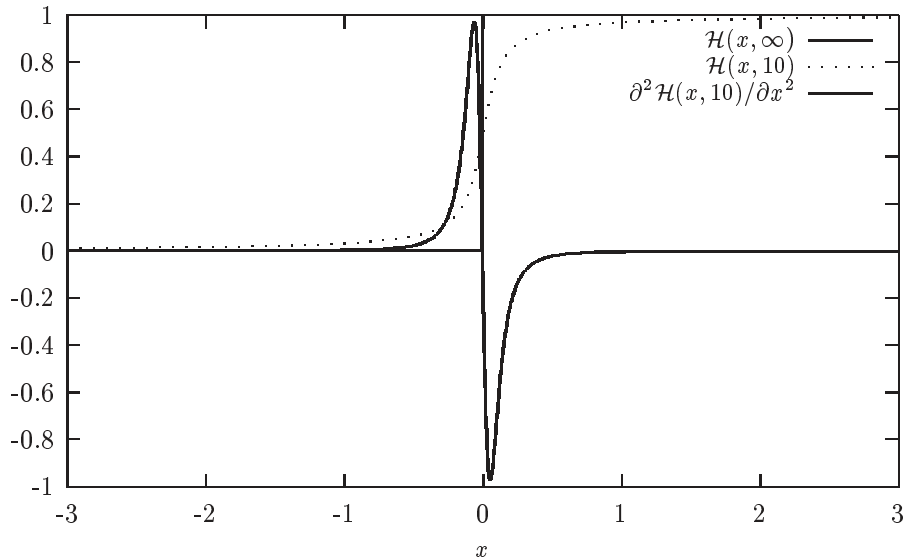


Figure 3. Illustration of the behavior of numerical second derivatives through discontinuities.

Numerical Hessians in elements crossing the discontinuity are highly ill conditioned and can not be used: across a shock of direction \vec{n} , $\partial^2 u / \partial n^2 = \vec{n} \cdot H \cdot \vec{n}$ changes of sign (see Figure 3). Its value is numerically underdetermined in terms of sign and amplitude. In elements where we have detected a discontinuity, we found it better to use the gradient $\vec{n} = \nabla w / |\nabla w|$ for determining the direction of the shock. The gradient of u is high everywhere through the shock and

has a constant sign. It is then much better conditioned. The resulting adaptive strategy can be described as follows:

- Determine the elements that cross a discontinuity using the algorithm (27);
- In elements where the solution is smooth, use (14) and (15) to compute an anisotropic metric field;
- In elements where the solution is discontinuous, use the reconstructed gradients ∇w solution of (20) to compute the normal direction \vec{n} to the discontinuity. Then, build up an anisotropic metric using $\lambda'_1 = \frac{1}{h_{min}^2}$ in the direction $\vec{e}_1 = \vec{n}$ and $\lambda'_2 = \lambda'_3 = \frac{1}{h_{nbr}^2}$ in the other directions using a normal equations for a smooth solution (see equations (14) and (15)). Since the solution is continuous in directions aligned to discontinuity, we choose

$$h_{nbr} = \frac{h^{\vec{n}+} + h^{\vec{n}-}}{2} \quad (30)$$

where $h^{\vec{n}+}$ and $h^{\vec{n}-}$ is the mesh size along directions aligned to discontinuity in nearby smooth regions on both sides of the discontinuity;

- When a metric field has been computed both in smooth and discontinuous regions, smooth this metric field to reconnect the anisotropic mesh metric field to be used by the mesh adaptation procedure. The step of metric smoothing is described in forthcoming Section §3.5.

Figure 4 show results of an anisotropic adaptive computation applied to the four contact problem that we described in Section §3.3. We have chosen here $h_{min} = 1/2560$ in order to resolve the discontinuities with the same accuracy as we did in the non-conforming refinement case. The number of triangles at $t = 0.3$ was around 82,000. There are 12 unknowns per linear triangle in case of Euler equations. So, the total number of equations was around 1,000,000.

3.5. Smoothing of Mesh Metric Field

The mesh metric field obtained from the scaled Hessian matrix (14) and jump features can have some artificial roughness due to the nature of the numerical operations required. To reduce this roughness, a mesh size smoothing procedure is used.

For a mesh size field specified over a mesh, methods to smooth mesh size variation have been described by Lohner [30, 31] and Borouchaki [9]. To maintain a desired mesh size growth ratio, Lohner adjusted the size specification by applying a geometric growth formula. Borouchaki defined measures to mesh metric variation over a mesh edge, and corrected the original mesh metric field so that mesh metric variation over all edges is less than a prescribed value. However, this approach is only effective in case isotropic. Its anisotropic extension may convert an anisotropic mesh metric field into isotropic one, thus lead to heavy over refinement since it reduces mesh metric in all directions based on information from one direction only.

In order to smooth a piecewise linear interpolation mesh metric field respecting existing anisotropy, let us consider two adjacent points where arbitrary mesh metrics are specified (Figure 6). We are interested in adjusting the two mesh metrics so that physical anisotropy is respected with direction and size both change smoothly from P to Q. The rest part of this section first introduces two useful definitions, anisotropy respect factor and directional H-shock, in adjusting mesh metrics, then presents anisotropic mesh size smoothing procedure.

Definition: The anisotropy respect factor related to point P and Q is the value:

$$\alpha = \frac{(R_q - 1) R_p}{(R_p - 1) R_q} \quad (R_p \geq R_q) \quad (31)$$

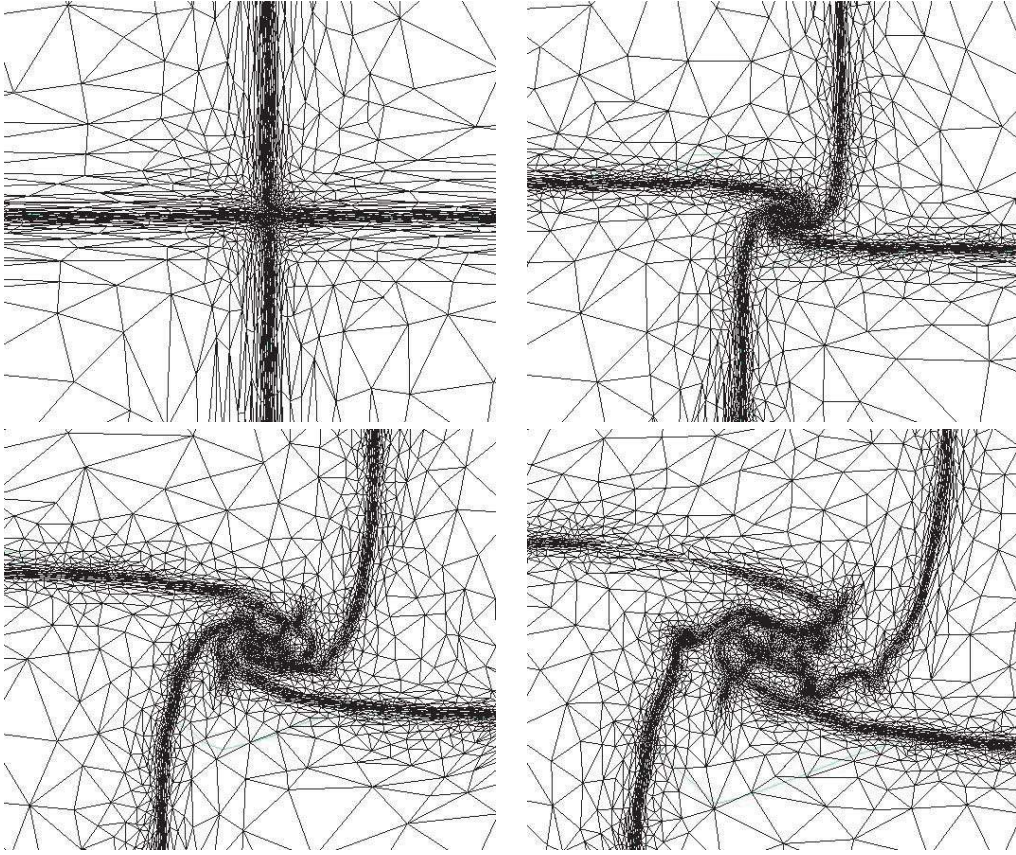


Figure 4. Anisotropic adaptive meshes. Zoom on the central zone of the square. On top left, the initial mesh. On top right, the mesh after 10 adaptations. On bottom left, the mesh after 20 adaptations. On bottom right, the mesh after 30 adaptations.

with R_p, R_q be the aspect ratio³ of the mesh metric at point P and Q, respectively.

In terms of definition, $\alpha \in [0, 1]$ with $\alpha = 0$ in case one of the mesh metrics is isotropic and $\alpha = 1$ in case the two mesh metrics have the same aspect ratio. This property is ideal for us to smooth direction variation between the two points. Equation (32) gives the method we use to correct eigenvectors of the less anisotropic mesh metric based on α . It ensures to respect the mesh metric with strong anisotropy and respect both in case $R_p = R_q$. Note that $\alpha = \frac{0}{0}$ is possible when both mesh metrics are isotropic, in which case the computation of α is not needed since eigenvector correction is unnecessary.

$$\vec{e}_i^{\bar{q}}|_{new} = (1 - \alpha) \vec{e}_i^{\bar{p}} + \alpha \vec{e}_i^{\bar{q}} \quad (i = 1, 2, 3) \quad (32)$$

where $\vec{e}_i^{\bar{p}}$ and $\vec{e}_i^{\bar{q}}$ are the eigenvector of mesh metric at point P and Q with $R_p \geq R_q$, respectively, and

³Given a mesh metric, the directional desired length distribution follows a ellipsoidal surface. Its aspect ratio R is defined as the maximum desired length to the minimum length. Clearly, $R \geq 1$.

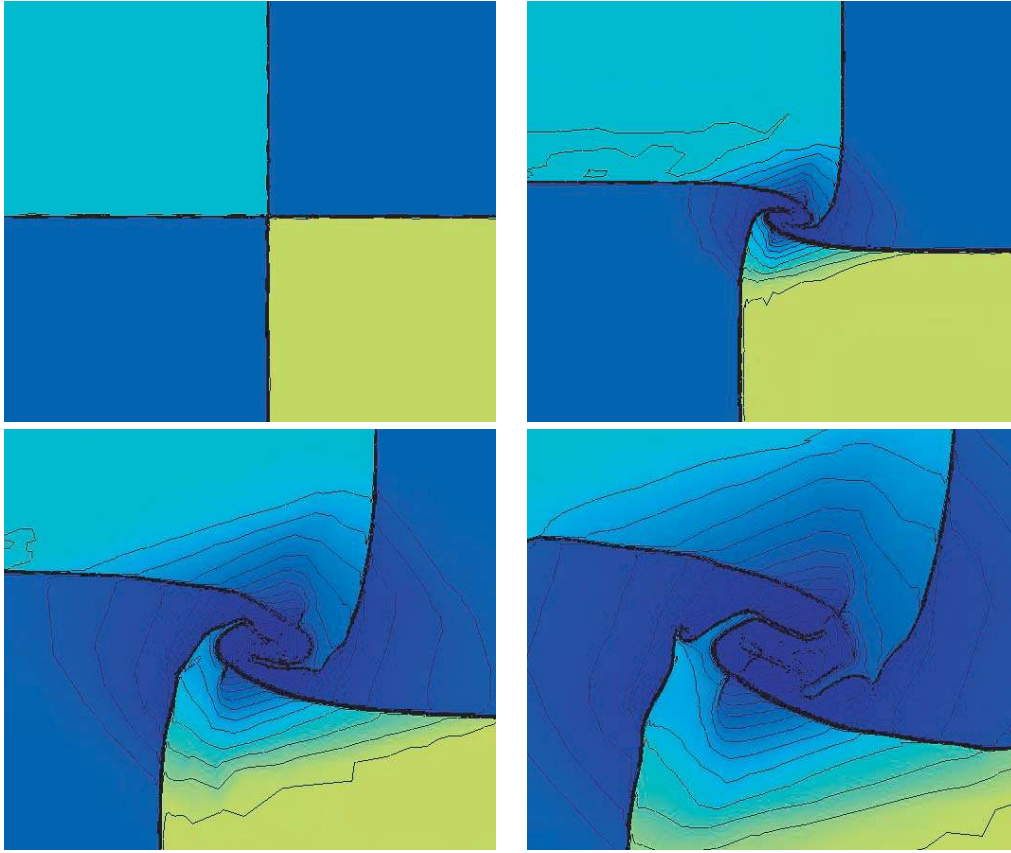


Figure 5. Density contours in log scale. Zoom on the central zone of the square. On top left, the initial solution. On top right, the solution after 10 adaptations. On bottom left, the solution after 20 adaptations. On bottom right, the solution after 30 adaptations.

$\vec{e}_i^q|_{new}$ is the corrected eigenvector at point Q .

Definition: The H-shock related to point P and Q along eigenvector \vec{e}_i is the value [9]:

$$c(\vec{e}_i) = \max\left(\frac{h_p}{h_q}, \frac{h_q}{h_p}\right)^{\frac{1}{L'(PQ)}} \quad (33)$$

with h_p, h_q the desired length along direction \vec{e}_i at point P and Q respectively. $L'(PQ)$ is the length of segment PQ with respect to the mesh metric variation over PQ (refer to (34) in Section §4.1).

This value measures the direction-related desired edge length variation between two mesh metrics. Specifically, it represents the progression ratio when fitting the mesh metric variation over PQ with a sequence of edges in geometric progression.

Given the allowed maximum value of H-shock β , the anisotropic smoothing procedure for piecewise linear mesh metric field is as follows:

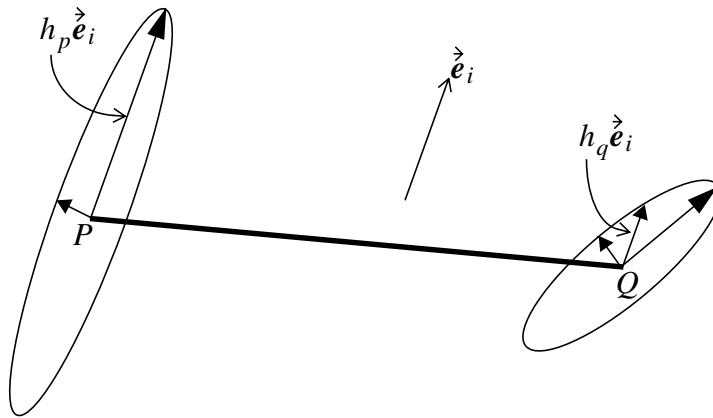


Figure 6. Definition of anisotropy respect factor and H-shock. P and Q are two adjacent points in the domain with mesh metric specified. Geometrically, mesh metric defines the desired edge length distribution as indicated by the two ellipses. \vec{e}_i shows one of the eigenvectors of the mesh metric at P . h_p and h_q indicate the desired edge length along direction \vec{e}_i at point P and Q respectively.

```

Initialize a dynamic list of edges
Loop over edges of the mesh that defines mesh metric field
  let PQ the current edge
  compute  $R_p, R_q$ 
  if both metrics at  $P$  and  $Q$  are isotropic
    adjust mesh size at  $P$  or  $Q$  using algorithm on page 1150 of ref. [9] if necessary
  else
    compute  $\alpha$  (suppose  $R_p \geq R_q$ )
    get all eigenvectors of the two mesh metrics at  $P$  and  $Q$ 
    compute new directions at  $Q$  based on equation (32)
    loop over eigenvectors at  $P$  and new directions at  $Q$ 
      let  $\vec{e}$  current eigenvector and  $M$  its associated point ( $M=P$  or  $Q$ )
      compute  $h_p, h_q$  along  $\vec{e}$  and  $c(\vec{e})$ 
      if  $c(\vec{e}) > \beta$  due to too big desired length at point  $M$ 
        compute a new desired length in  $\vec{e}$  so that  $c(\vec{e}) = \beta$ 
      end if
    end loop
    if any new desired length at point  $P$  or  $Q$  has been computed
      replace the mesh metric at  $P$  or  $Q$ 
      insert edges connected to point  $P$  or  $Q$  into the dynamic edge list
    end if
  end if
end loop
process the edges in the dynamic list in the same fashion until the list is empty

```

4. ANISOTROPIC MESH ADAPTATION VIA LOCAL MESH MODIFICATION

Given the mesh metric field defined over the domain, the goal is to apply local mesh modification operators to yield a mesh of the same quality as would be obtained by an anisotropic domain remeshing procedure but at less computational cost. Although quite effective on polygonal domains in the case of isotropic mesh adaptation, procedures that apply a small set of refinement and de-refinement templates would substantially limit the degree to which the modified mesh can reflect the required mesh metric field. In addition such templates cannot deal properly with adapting meshes in curved domains. Therefore, a more complete set of mesh modification operations are needed. Key to the application of these procedures is the ability to use the information about the current local mesh and the mesh metric field to quickly determine the appropriate mesh modification to apply without the cost of examining all the possible operations that could be applied. This section outlines how the intelligent mesh modification operations described in references [25, 26] that are used to convert the given mesh into one that satisfies the given mesh metric field.

A number of local mesh modification operators have been developed for simplex meshes to either change the local mesh resolution or to change the shape quality of the elements in the mesh. At the most basic level the operators developed for these purposes include entity (i) split, (ii) collapse, (iii) swap, and (iv) reposition operations. To meet the needs of a particular process these operations can be applied in various combinations as driven by criteria devised for their application. For purposes of anisotropic mesh adaptation, mesh modification is used to directionally refine and coarsen the mesh, and to realign the mesh to satisfy the given anisotropic mesh metric field.

4.1. Mesh modification criteria

Since the anisotropic mesh size field represents the transformation that map an ellipsoid into a unit sphere, the ideal tetrahedron that satisfies the mesh size field should be mapped into a unit equilateral tetrahedron in the transformed space. Figure 7 demonstrate this concept. The left figure depicts two desired anisotropic tetrahedra in physical space, while the transformation associated with the mesh metric field is indicated by the two matrices. As illustrated by the right figure, both tetrahedra are transformed into a unit equilateral tetrahedron by their corresponding transformation matrix.

To make any given mesh satisfying the given mesh size field by mesh modifications, we take philosophy as follows:

- identify those mesh entities not satisfying the mesh size field;
- perform appropriate mesh modifications so that local mesh will better satisfy the mesh size field;
- repeat above steps until the mesh size field is satisfied to an acceptable degree.

The degree of the satisfaction of a mesh to a mesh size field can be measured by mesh edge length in the transformed space. Consider mesh edge AB and the transformation representation of the mesh size field $Q(x)$ over AB (refer to figure 8). In general, the length of PQ in transformed space can be computed by [25, 42, 20]:

$$L'(AB) = \int_A^B \sqrt{\vec{e} \cdot Q(x)Q(x)^T \cdot \vec{e}} dx \quad (34)$$

with \vec{e} be the unit vector associated with edge AB in physical space.

Since it is not possible to ensure all mesh edges perfect through mesh modifications, the primary motivation of mesh modifications is to make the transformed length of all mesh edges fallen into a

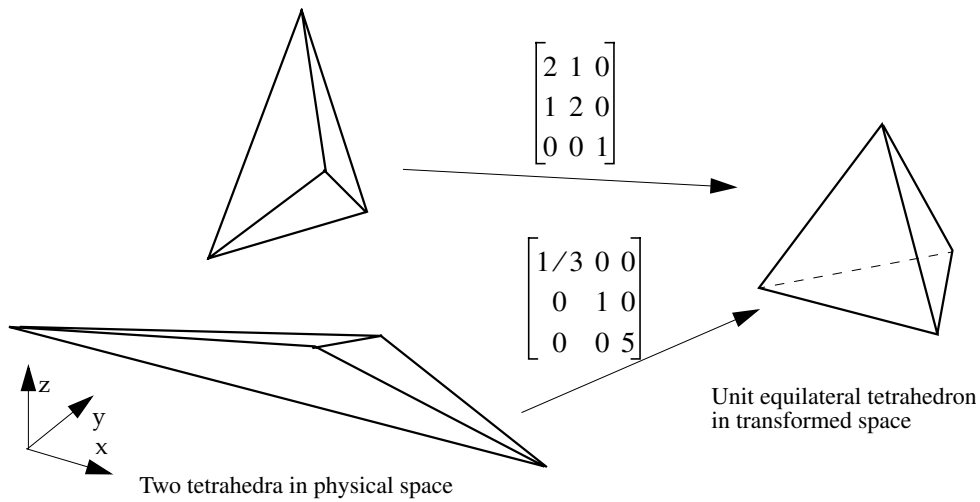


Figure 7. Desired tetrahedra are mapped into unit equilateral tetrahedra by the transformation mesh metric defines.

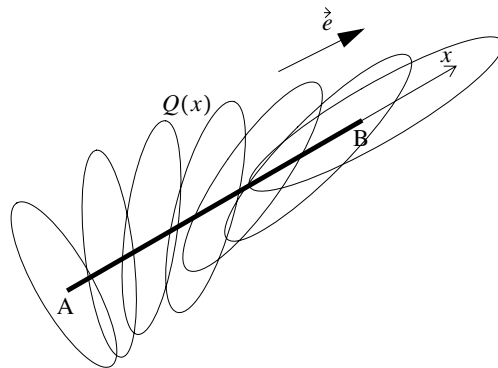


Figure 8. Illustration of Length computation in transformed space. Ellipses indicate $Q(x)$ defined over edge AB .

small interval close to one. Particularly, we choose interval $[0.5, 1.4]$ in the examples given in section §5. The reason for using such an interval is to ensure that the two new edges from a bisection will not be short edges so that oscillation between refining and coarsening are prevented. (Note that intervals other than $[0.5, 1.4]$ can also be used as long as they will not cause oscillation.) Hereafter, for convenience, a mesh edge is considered “short” if its transformed length is less than the lower bound of the interval and a mesh edge is considered “long” if its transformed length is greater than the upper bound.

Sliver tetrahedra (poorly-shaped tetrahedra not bounded by any short mesh edge in transformed space) may exist even if the edge length criteria is met, so an additional criteria is needed to determine and eliminate sliver tetrahedra. One of the standard non-dimensional shape measure, the cubic of mean ratio [27] in the transformed space, is used for this purpose. Let Q be the associated transformation

matrix of the tetrahedron ⁴, the cubic of mean ratio in transformed space, η' , is:

$$\eta' = \frac{15552 (|Q| V)^2}{\left(\sum_{i=1}^6 \vec{l}_i \cdot Q Q^T \cdot \vec{l}_i^T\right)^3} \quad (35)$$

where $|Q|V$ is the volume of the tetrahedron in the transformed space ($|Q|$ represents the determinant of the transformation, V is the volume of a tetrahedron in physical space), and $\vec{l}_i (i = 1..6)$ are vectors associated with the six edges of the tetrahedron. Note that η' has been normalized to interval $[0, 1]$ with 0 for flat tetrahedron and 1 for equilateral tetrahedron in transformed space. In section §4.4, we consider a tetrahedron as sliver that must be corrected if its η' is less than a given threshold.

4.2. Refinement

Splits are base operators used in refinement procedures. In three dimensions the split operators are edge, face, and region split. There are a wide variety of algorithms that have been developed for the application of these operations for use in adaptive refinement [18]. Refinement procedures based on edge splits are most common, either matching them to predefined patterns [7, 18] or applying specific criteria [28] selected to maintain control of element degradation. The methods that apply templates with no consideration of element shape can lead to element shape degradation while those based on a longest edge, or alternating edge, criteria lead to refinement propagation that can produce a substantial increase in the number of elements. Some procedures employ a Delaunay point insertion process to refine the mesh [32] which can be shown to be equivalent to an entity split followed by appropriate swaps to satisfy the Delaunay criteria.

In the presented mesh adaptation procedure, a full set of predefined patterns described in [18] are used to refine the mesh since it is efficient (linear complexity), prevents over-refinement and provide possibilities to maintain or even improve mesh quality. The result mesh quality after refinement has been controlled from the following three aspects:

- Whenever there are multiple options to split a tetrahedron, always create the shortest diagonal edge in the transformed space;
- To achieve the refinement patterns as demonstrated in figure 9, perform refinement in multiple iterations, mark a set of longest edges in transformed space to be split for each iteration, and terminate the iteration when the longest edge of the mesh in the transformed space is less than the upper bound of the interval of allowed edge length in transformed space;
- Eliminate short edges in the transformed space after each refinement iteration.

As figure 10 illustrates, whenever two edges of a triangle is marked to be split, there will be two options to triangulate the triangle domain and the short diagonal in transformed produce a better configuration. Similarly in case of 3-D when there are two or three options to introduce the interior edge.

Figure 11 depicts a 2D example to demonstrate the needs of multiple refinement iterations and coupling with short edge elimination. In this example, three iterations are applied to refine the two initial triangles to desired size level and a set of longest edges is marked for refinement in each iteration.

⁴In case the transformation is not constant over the tetrahedron, the one with maximum aspect ratio is considered as associated transformation.

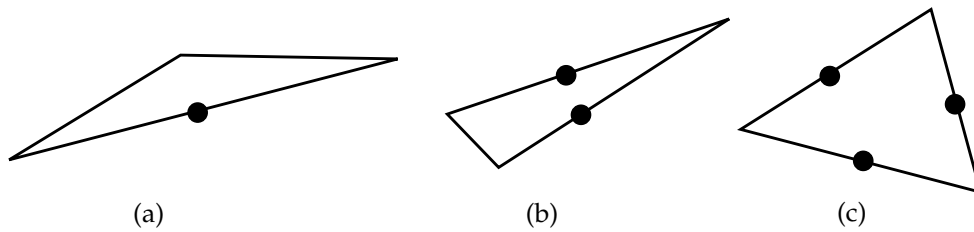


Figure 9. Examples of desired refinement patterns. Refinement edges are indicated by black bullets. In (a) only the longest edge can be split; in (b) the shortest edge can not be split; while in (c) all edges can be split.

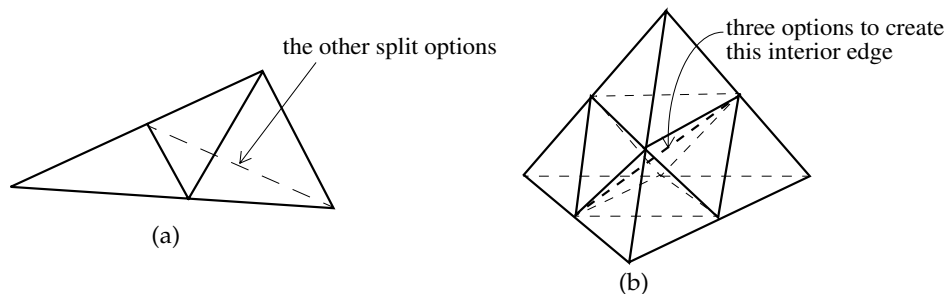


Figure 10. Multiple options to create diagonal edges. (a) two ways to split a triangle when two edges are marked for refinement; (b) three ways to split a tetrahedron into eight child tetrahedra.

In the first iteration, only two edges (indicated by bullets) are split, which creates a short edge since one of the initial triangle has poor quality. Collapsing the short edge is needed, and it can be seen that the combination of split and collapse improves mesh quality. The second iteration only split the longest edge since it is much longer than others. The third iteration splits six long edges that are in close length.

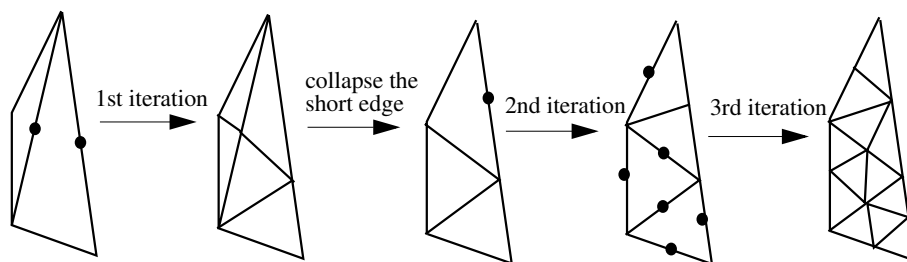


Figure 11. 2D example of the refinement algorithm.

4.3. Coarsening

Edge, face, and region collapse operations can be defined in an analogous way to the split operations and can be used for mesh coarsening. The edge collapse tends to be the most useful approach, however the other operations have been found of use in specific cases. Since the application of these must consider complexities of element shape, the most common approach is to limit coarsening to the reversing of a refinement step. However, this does not allow coarsening past the initial mesh and does require maintaining history information. In case a short edge is adjacent to a long edge, repositioning the common vertex of both edges can also be a useful approach.

The coarsening algorithm first determines a list of vertices that bound short edges, then eliminates them in the order of topologically every other vertex. Short edges are identified based on length computation in transformed space. Three local mesh modification operators are used when processing a vertex in the list:

- edge collapsing;
- vertex repositioning;
- compound operator.

Consider a mesh vertex that bounds at least a short mesh edge, the coarsening process first get its shortest adjacent mesh edge, and evaluate the removal of this vertex by collapsing it onto the vertex at the other end of the shortest edge. If this collapsing will create long mesh edges in transformed space, repositioning this vertex will be evaluated. Compound operations are investigated if the edge collapsing is geometrically not acceptable. The purpose of the compound operation is still to collapse the shortest edge. It first determines flat elements collapsing the shortest edge would yield, then determines the swap or collapse operation(s) to eliminate those flat elements. In particular, the collapse operation is attempted in case the flat tetrahedron has a short edge, and swap operation is attempted otherwise.

To prevent the possible oscillation between refining and collapsing, any of above local mesh modifications is considered unacceptable if it creates a long mesh edge in transformed space.

4.4. Re-Alignment

Local mesh modifications, particularly edge and face swap operators, can be used to improve the quality of the mesh by replacing poorly shaped elements with higher quality elements. In addition to the classic 2-to-3 and 3-to-2 swap operations, more extensive swap operations have proven to be quite useful [13, 18, 35]. Swap operations are commonly used in evolving geometry problems where meshes undergo severe distortions and can be effectively used with adaptive refinement procedures to maintain mesh quality while limiting the amount of refinement propagation [18, 34].

The re-alignment algorithm aims at eliminating existing sliver tetrahedra in the transformed space. It first visits each element of the mesh in turn to identify sliver tetrahedra in transformed space (see the criteria discussed in Section §4.1), then process each identified sliver tetrahedron by analyzing the shape of the sliver tetrahedron in transformed space, evaluate a small set of promising operations to eliminate the sliver, and apply the one that improves alignment most if possible.

To support the intelligent determination of local mesh modifications, it is useful to classify sliver tetrahedra into two types (refer to figure 12). A tetrahedron is classified as type I sliver if two opposite edges of the tetrahedron almost intersect; A tetrahedron is classified as type II if one vertex of the tetrahedron is close to the centroid of its opposite face.

Key mesh entities to eliminate the sliver tetrahedron can be identified for these two types: in case of type I, it is a pair of mesh edges (indicated by circles); In case of type II, it is a mesh face (indicated

by the three squares) and a mesh vertex (indicated by the circle). Table II lists the promising local mesh modification operation(s) to be evaluated for each type. To be effective, mesh modifications are evaluated in three priority levels and local mesh modifications in the next priority level are evaluated until all mesh modification in previous level can not improve local mesh configuration. In case more than one operation is possible, the one producing best local mesh configuration is applied.

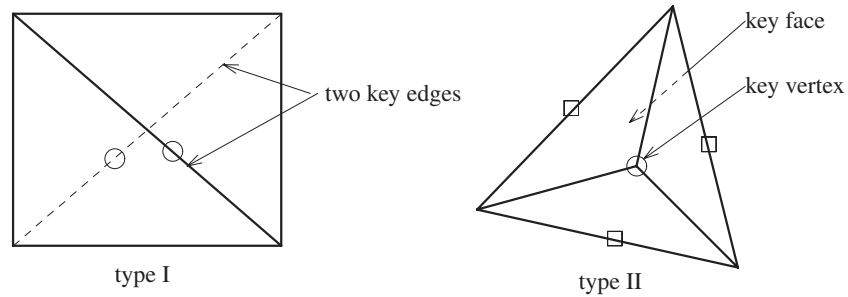


Figure 12. Sliver tetrahedron types and associated key entities.

Priority	Mesh modifications for type I	Mesh modifications for type II
1	swap either key mesh edge	swap the key face or relocate the key vertex
2	split either key edge and relocate the new vertex, split both edges and collapse the interior edge	split the face then split/relocate the new vertex, swap the three edges that bound the face
3	relocate vertices of the tetrahedron	relocate the three vertices that bound the face

Table II. Determination of local mesh modifications.

5. RESULTS

5.1. Cannon Blast Problem

Consider the problem of a 2D perforated tube of diameter 155 mm (a section of a cannon). The diameter of the perforated holes inside the barrel (the muzzle break) are $d = 28.6$ mm.

The initial conditions for the problem are the one of a shock tube. We consider a virtual interface inside the barrel (see mesh refinement for $t = 0$ at Figure 13). The initial pressure for the gas inside of the tube are $P = 57,273,627.96$ Pa i.e. more than 500 times the external atmospheric pressure of $P_{atm} = 101,300$ Pa. The initial temperature of the air inside the tube is $T = 2,111.5$ K and its initial velocity along x direction is 0.

The final time of the computation was $t_{end} = 5 \times 10^{-4}$ sec. A second order Runge-Kutta time integration scheme was used. The time steps were computed adaptively with a CFL limit of 1.0. Starting time steps were about 5×10^{-8} seconds and the final time steps were about 1.5×10^{-8} seconds. The mesh was refined every 10^{-6} seconds so that the total number of mesh refinements is 501, including the initial refinement that enables the correct capture of the initial discontinuous state

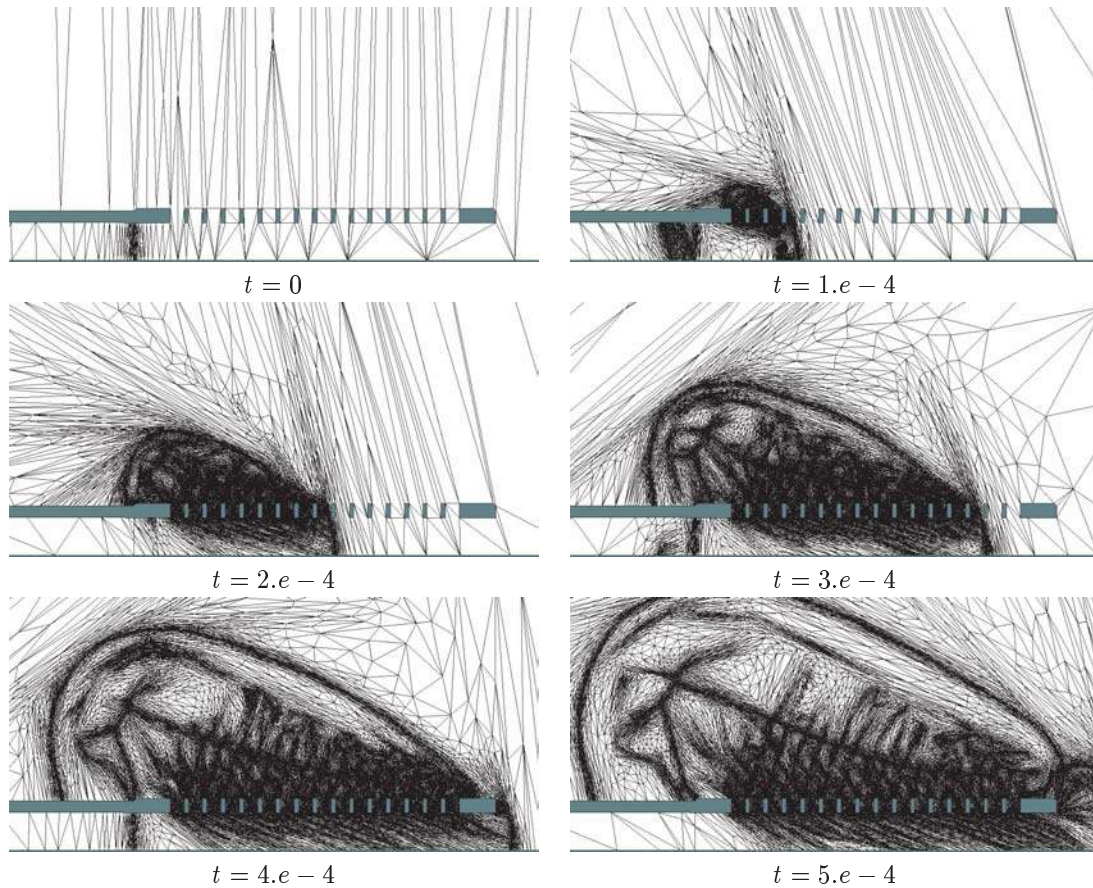


Figure 13. Evolution of the adapted meshes for the cannon blast problem.

(see Figure 13). The total number of solution time steps is 45,438. The total number of degrees of freedom for this problem starts at 5,556 which corresponds to 463 triangles. After the 501 adaptations, the number of degrees of freedom reaches 778,488 which corresponds to 64,874 triangles. Figure 13 show the evolution of the mesh for the cannon blast problem. The minimum mesh size allowed for this problem was $h_{min} = 1mm$ and the smoothing factor was $\beta = 3$. Figure 14 plots the density contours corresponding to the adapted meshes of Figure 13. One can clearly see that the density contours do not have any pre and post shock noise due to the alignment of anisotropic elements with shock waves, and the simultaneous development between anisotropic elements and the density contours. Figure 15 and figure 16 gives two close-up views to further demonstrate the captured solution by aligned anisotropic elements. In figure 15, the complex shock-shock interactions happening above the perforated holes are captured by anisotropic elements distributed in the direction and position the density contours indicate. In figure 16, the zoom near the front shock shows the alignment between the anisotropic elements and the front shock.

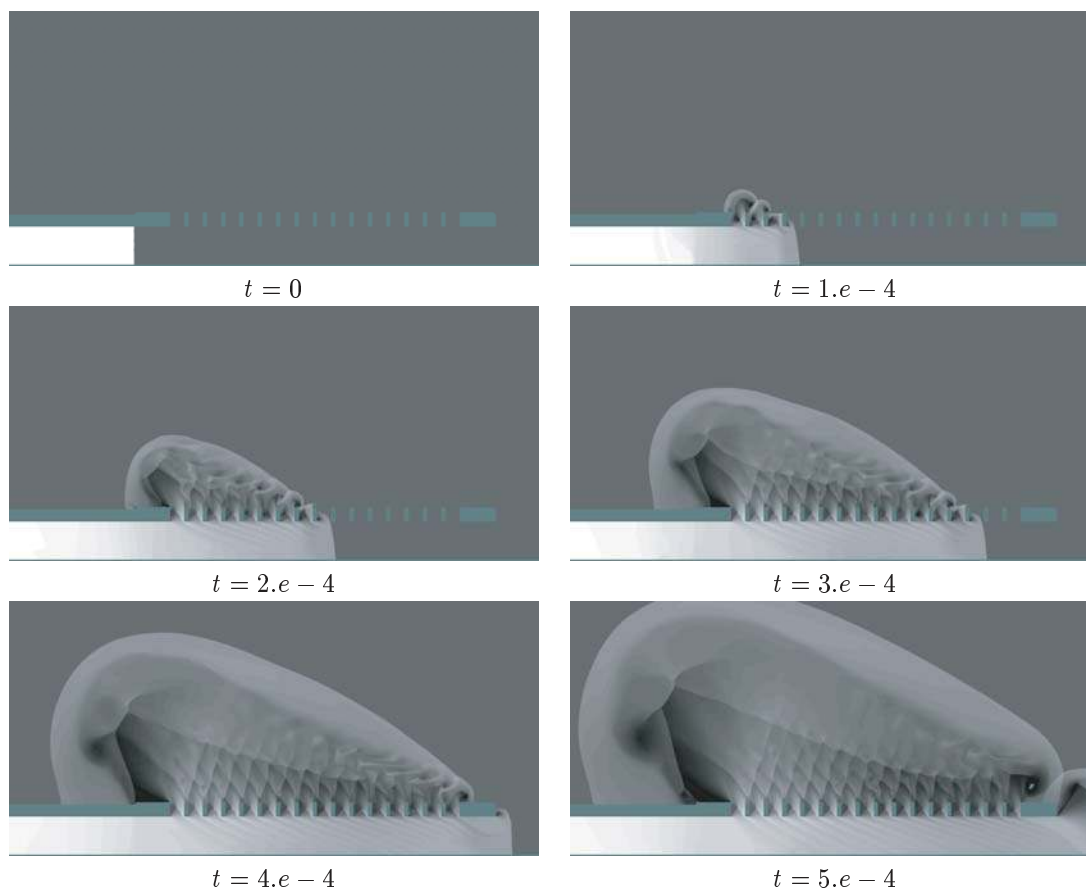


Figure 14. Evolution of the density contours in log scale for the cannon blast problem.

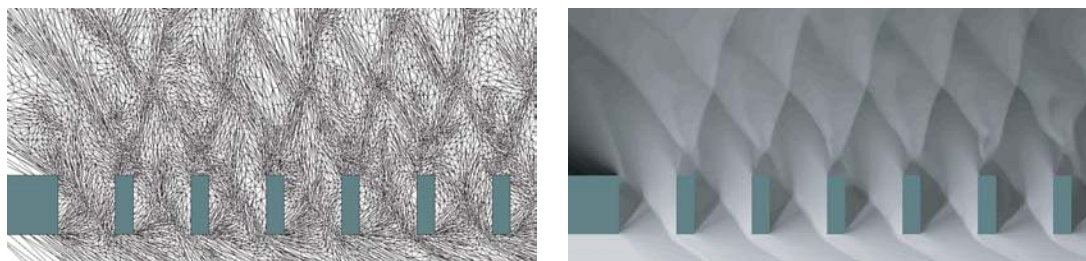


Figure 15. Complex shock-shock interaction structure near the muzzle at $t = 5.e - 4$.

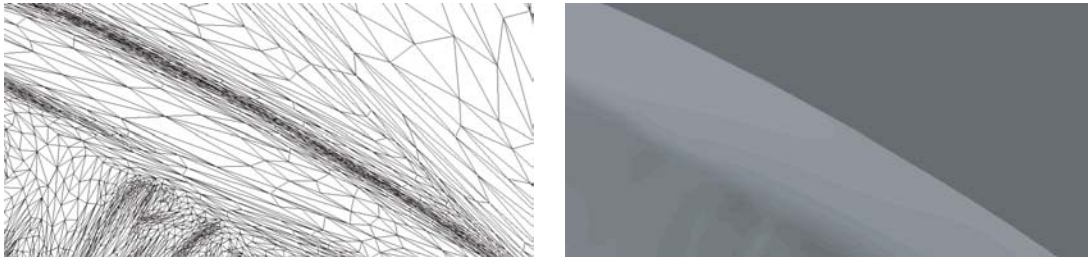


Figure 16. Zoom near to the front shock at $t = 5.e - 4$.

5.2. Three-Dimensional backward facing step

This example simulates the shock development when a backward facing step is impulsively inserted into a Mach 3 gas flow in a straight pipe. Figure 17 shows the analysis domain. Since axisymmetric, only a small section (15 degree) of a cylinder is used. The cylinder is of length 7.62 and of radius 1.52, and the step is situated at $x = 1.524$ and of height 0.508. The initial condition is a constant Mach 3 flow field in the x -axis, in particular,

$$\begin{aligned} p &= 1 \\ \rho &= 1 \\ u_1 &= M_s \sqrt{\gamma} = 3\sqrt{1.4} = 3.55 \end{aligned}$$

where p denotes pressure, ρ denotes density, u_1 is the velocity in x direction, M_s is Mach number and γ is gas parameter. The boundary conditions are as follows:

- At inlet and outlet, the velocity, density and pressure are that of the initial Mach 3 flow;
- At the two cut planes parallel to the center line of the cylinder, symmetry boundary condition is applied;
- For all other surfaces, slip wall boundary condition is applied.

Starting from a uniform isotropic initial mesh of size 0.5, a steady flow pattern with shock is reached in about 4 seconds. The mesh is updated every 5×10^{-3} seconds, therefore, a total of 800 mesh adaptations are performed. The total number of degrees of freedom in the initial mesh is 14,960 which corresponds to 748 tetrahedra. After 800 mesh adaptations, the number of degrees of freedom reaches 96,020 which corresponds to 5081 tetrahedra. Figure 18 shows the evolution of the mesh and the corresponding density contour surface for the backward step problem. It can be seen that the mesh aligns to the discontinuity of density with anisotropic tetrahedra and develops as the discontinuity develops. Figure 19 shows a close-up view of the mesh and density contour near the top surface where the shock reflects. It can be seen that elements become needle-like where the shock strikes the top surface.

6. CLOSING REMARKS

A general procedure for the adaptive construction of anisotropic meshes over general 2- and 3-dimensional domains has been presented. Its application has been demonstrated on the transient flow

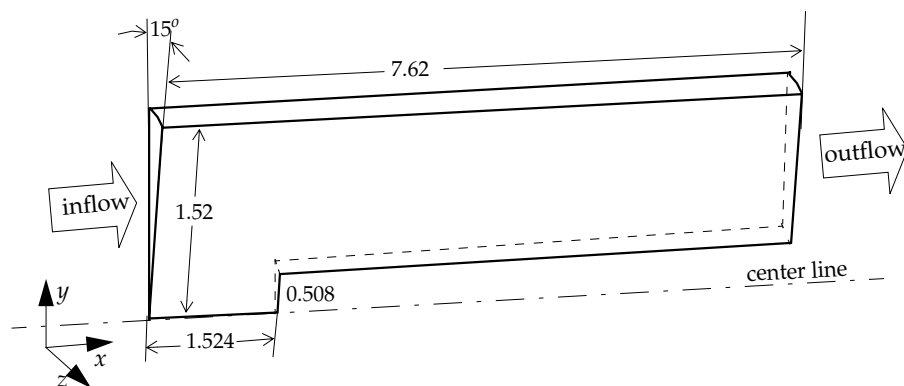


Figure 17. Simulation domain of backward step.

simulations that have complex evolving features. Key features of the procedures presented include:

- a general approach to the construction of an anisotropic mesh metric field capable of continued improvement as new error estimation and correction indication procedures are developed,
- a variationally-based procedure to calculate higher derivatives applicable for use with discontinuous Galerkin methods,
- a procedure to detect solution discontinuities and isolate them for the generation of an appropriate anisotropic mesh at those locations,
- a set of intelligent mesh modification procedures that can modify a given mesh to match any given mesh metric field.

ACKNOWLEDGEMENTS

The various components of this work were supported by Simmetrix Inc., the ASCI Flash Center at the University of Chicago under contract B341495, and the DOE SciDAC program through agreement DE-FC02-01-ER25460.

REFERENCES

1. S. Adjerid, K.D. Devine, J.E. Flaherty, and L. Krivodonova. A posteriori error estimation for discontinuous galerkin solutions of hyperbolic problems. *Computer methods in applied mechanics and engineering*, 191:1097–1112, 2002.
2. M. Ainsworth and J.T. Oden. *A Posteriori Error Estimation in Finite Element Analysis*. Wiley-Interscience, 2000.
3. R.C. Almeida, P.A. Feijoo, A.C. Galeao, C. Padra, and R.S. Silva. Adaptive finite element computational fluid dynamics using an anisotropic error estimator. *Computer Methods in Applied Mechanics and Engineering*, 182(3-4):379–400, 2000.
4. I. Babuska and T. Strouboulis. *The Finite Element Method and its Reliability*. Oxford University Press, 2001.
5. R. Biswas, K. Devine, and J.E. Flaherty. Parallel adaptive finite element methods for conservation laws. *Applied Numerical Mathematics*, 14:255–284, 1994.
6. R. Biswas, K. D. Devine, and J. E. Flaherty. Parallel adaptive finite element method for conservation laws. *Applied Numerical Mathematics*, 14:255–283, 1984.
7. F. Bornemann, B. Erdmann, and R. Kornhuber. Adaptive multilevel methods in three space dimensions. *International Journal for Numerical Methods in Engineering*, 36:3187–3203, 1993.
8. H. Borouchaki, P.L. George, F. Hecht, P. Laug, and Saltel. Delaunay mesh generation governed by metric specifications - part i: Algorithms and part ii: Applications. *Finite Elements in Analysis and Design*, 25:61–83, 85–109, 1997.

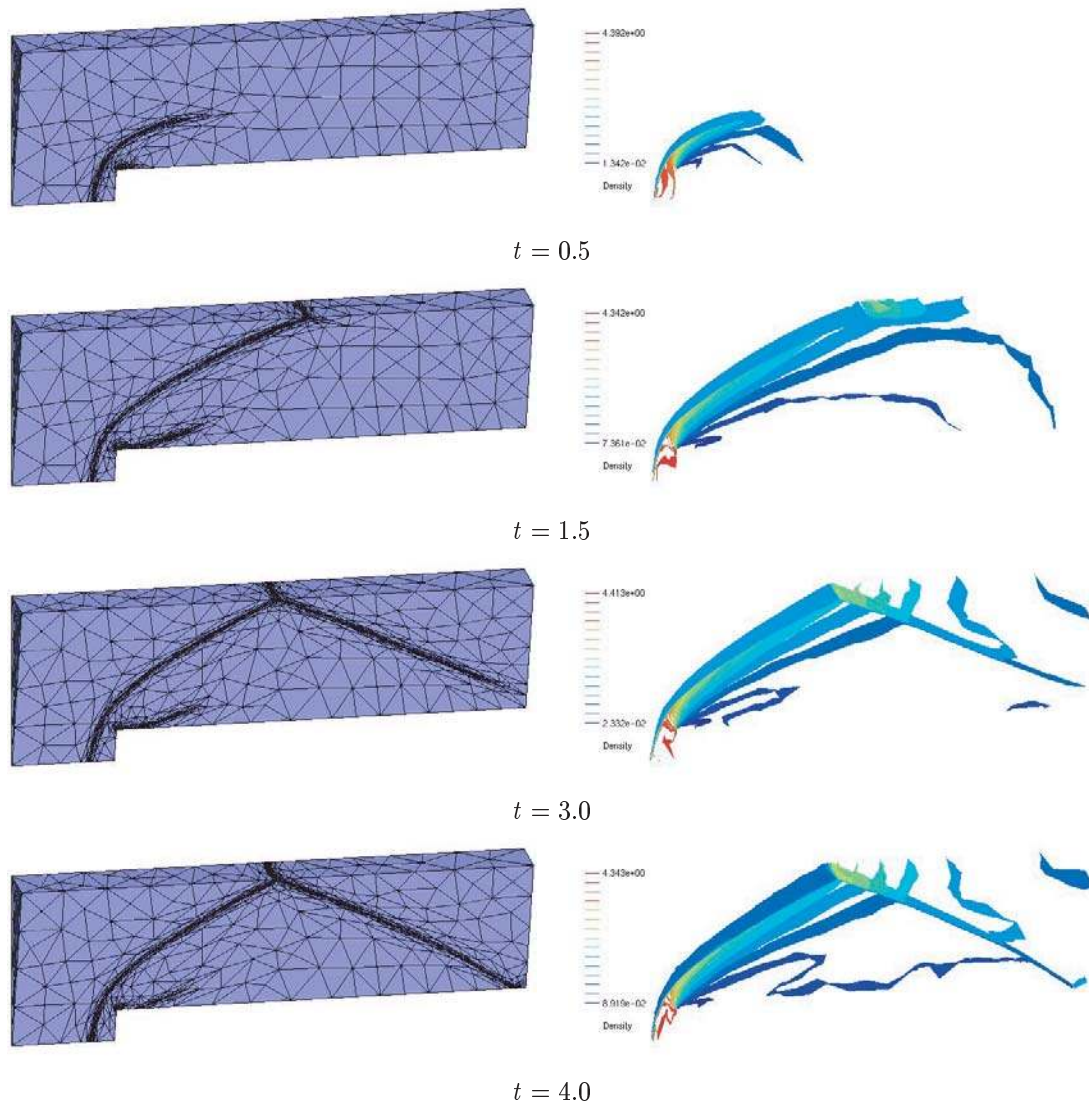


Figure 18. Evolution of mesh and density contour for backward facing step problem.

9. H. Borouchaki, F. Hecht, and P. J. Frey. Mesh gradation control. *International Journal for Numerical Methods in Engineering*, 43(6):1143–1165, 1998.
10. C.L. Bottasso and M.S. Shephard. A parallel adaptive finite element flow solver for rotary wing aerodynamics. *AIAA*, 35(6):1–8, 1997.
11. Dietrich Braess. *Finite Elements : Theory, Fast Solvers, and Applications in Solid Mechanics*. Cambridge Univ. Pr., 1997.
12. F. Brezzi, G. Manzini, D. Marini, P. Pietra, and A. Russo. Discontinuous finite elements for elliptic problems. *Numerical Methods For Partial Differential Equations*, 16:365–378, 2000.
13. E. Briere de l’Isle and P. L. George. Optimization of tetrahedral meshes. In I. Babuska, J. E. Flaherty, W. D. Henshaw

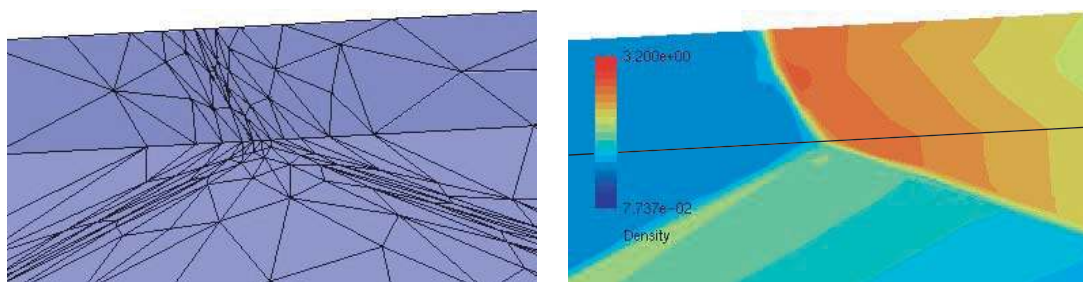


Figure 19. Zoom near the shock reflection at $t = 4$ seconds.

- J. E. Hopcroft, J. E. Olinger, and T. Tezduyar, editors, *Modeling, Mesh Generation, and Adaptive Numerical Methods for Partial Differential Equations*, pages 97–128. Springer-Verlag, 1993.
14. M.J. Castro-Diaz, F. Hecht, and B. Mohammadi. Anisotropic unstructured grid adaptation for flow simulations. *International Journal for Numerical Methods in fluids*, 25(4):475–491, 1997.
 15. B. Cockburn, G.E. Karniadakis, and C.-W. Shu, editors. *Discontinuous Galerkin Methods*, volume 11 of *Lecture Notes in Computational Science and Engineering*, Berlin, 2000. Springer.
 16. B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin methods for scalar conservation laws II: General framework. *Mathematics of Computation*, 52:411–435, 1989.
 17. P. Colella and H. M. Glaz. Efficient solution algorithms for the Riemann problem for real gases. *Journal of Computational Physics*, 59:264–289, 1985.
 18. H. L. de Cougny and M. S. Shephard. Parallel refinement and coarsening of tetrahedral meshes. *International Journal for Numerical Methods in Engineering*, 46:1101–1125, 1999.
 19. M. Dindar, M. S. Shephard, J.E. Flaherty, and K. Jansen. Adaptive cfd analysis for rotorcraft aerodynamics. *Computer Methods in Applied Mechanics and Engineering*, 189:1055–1076, 2000.
 20. P. L. George and F. Hecht. Non isotropic grids. In J. Thompson, B. K. Soni, and N. P. Weatherill, editors, *CRC Handbook of Grid Generation*, pages 20.1–20.29. Boca Raton, 1999. CRC Press, Inc.
 21. J. Goodman, K. Samuelsson, and Szepessy A. Anisotropic refinement algorithms for finite elements. Technical report, NADA KTH, Stockholm, March 1996.
 22. G. Kunert. Toward anisotropic mesh construction and error estimation in the finite element method. *Numerical Meth. in Partial Differential Equations*, 18:625–648, 2002.
 23. B. Van Leer. Flux vector splitting for the Euler equations. Technical report, ICASE Report, NASA Langley Research Center, 1995.
 24. R. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser-Verlag, 1992.
 25. X. Li. *Mesh Modification Procedures for General 3-D Non-manifold Domains*. PhD thesis, Rensselaer Polytechnic Institute, August, 2003.
 26. X. Li, M.S. Shephard, and M.W. Beall. 3-d anisotropic mesh adaptation by mesh modifications. *Computer Methods in Applied Mechanics and Engineering*, 2003. in preparation.
 27. A. Liu and B. Joe. On the shape of tetrahedra from bisection. *Mathematics of Computations*, 63:141–154, 1994.
 28. A. Liu and B. Joe. Quality local refinements of tetrahedral meshed based on bisection. *SIAM Journal on Scientific Computing*, 16:1269–1291, 1995.
 29. R. Löhner. Some useful data structures for the generation of unstructured grids. *Commun. Appl. Num. Meth.*, 4:123–135, 1988.
 30. R. Löhner. Adaptive remeshing for transient problems. *Computer Methods in Applied Mechanics and Engineering*, 75:195–214, 1989.
 31. R. Löhner. Extensions and improvements of the advancing front grid generation technique. *Communications in Numerical Methods in Engineering*, 12:683–702, 1996.
 32. D.J. Mavriplis. Adaptive mesh generation for viscous flows using delaunay triangulation. *Journal of Computational Physics*, 90:271–291, 1990.
 33. J.-D. Muller. Anisotropic adaptation and multigrid for hybrid grids. *International Journal for Numerical Methods in fluids*, 40:445–455, 2002.
 34. C. Ollivier-Gooch. An unstructured mesh improvement toolkit with application to mesh improvement, generation and (de-)refinement. In *AIAA Paper 98-0218, 36th Aerospace Sciences Meeting and Exhibit*, January 1998.
 35. C.C. Pain, A.P. Umpheby, C.R.E. de Oliveria, and A.J.H. Goddard. Tetrahedral mesh optimization and adaptivity for steady-

- state and transient finite element calculations. *Computer Methods in Applied Mechanics and Engineering*, 190:3771–3796, 2001.
36. S. Z. Pirzadeh. An adaptive unstructured grid method by grid subdivision, local remeshing and grid movement. In *AIAA Paper 99-3255, 14th AIAA Computational Fluid Dynamics Conf.*, July 1999.
 37. W. Rachowicz. An anisotropic h-type refinement strategy. *Computer Methods in Applied Mechanics and Engineering*, 109:169–181, 1993.
 38. W. Rachowicz. An anisotropic h-adaptive finite element method for compressible navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 147:231–252, 1997.
 39. J.-F. Remacle, J. E. Flaherty, and M. S. Shephard. An adaptive discontinuous galerkin technique with an orthogonal basis applied to compressible flow problems. *SIAM Journal on Scientific Computing*, in press, 2002.
 40. J.-F. Remacle, K. Pinchedez, J. E. Flaherty, and M. S. Shephard. An efficient local time stepping-discontinuous galerkin scheme for adaptive transient computations. *Computer Methods in Applied Mechanics and Engineering*, 2002. accepted.
 41. P. Saramito and N. Roquet. An adaptive finite element method for viscoplastic fluid flow in pipes. *International Journal for Numerical Methods in Engineering*, 190:5391–5412, 2001.
 42. Y. Thomas Tracy. *Concepts from tensor analysis and differential geometry*. Academic, New York, 1965.
 43. B. van Leer. Towards the ultimate conservation difference scheme, II. *Journal of Computational Physics*, 14:361–367, 1974.
 44. B. van Leer. Towards the ultimate conservation difference scheme, V. *Journal of Computational Physics*, 32:1–136, 1979.
 45. M. Walkley, P.K. Jimack, and M. Berzins. Anisotropic adaptivity for finite element solutions for three-dimensional convection-dominated problems. *International Journal for Numerical Methods in fluids*, 40:551–559, 2002.
 46. P.C. Walsh and D.W. Zingg. Solution adaptation of unstructured grids for two-dimensional aerodynamic computations. *AIAA Journal*, 39(5):831–837, 2001.
 47. P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54:115–173, 1984.
 48. O.C. Zienkiewicz and J.Z. Zhu. Superconvergent patch recovery and a posteriori error estimates, part 1: The recovery technique. *International Journal for Numerical Methods in Engineering*, 33(7):1331–1364, 1992.