INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2003; **00**:1–20 *Prepared using* fldauth.cls *[Version: 2002/09/18 v1.01]*

Adaptive Discontinuous Galerkin Method for the Shallow Water Equations

Jean-François Remacle, Sandra Soares Frazão [1,*]
Xiagrong Li [2] and Mark S. Shephard [2]

[1] *Department of Civil Engineering, Place du Levant 1, 1348 Louvain-la-Neuve, Belgium*
[2] *Scientific Computation Research Center, CII-7011, 110 8th Street, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, U.S.A.*

SUMMARY

In this paper, we present a Discontinuous Galerkin formulation of the shallow water equations. An orthogonal basis is used for the spatial discretization and an explicit Runge-Kutta scheme is used for time discretization. Some results of second order anisotropic adaptive calculations are presented for dam breaking problems. The adaptive procedure uses an error indicator that concentrates the computational effort near discontinuities like hydraulic jumps. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: Shallow Water Equations, Anisotropic Meshes, Discontinuous Galerkin Method

1. Introduction

The Discontinuous Galerkin Method (DGM) was initially introduced by Reed and Hill in 1973 [6] as a technique to solve neutron transport problems. Recently, the DGM has become popular and it has been used for solving a wide range of problems [2].

The DGM is a finite element method in the sense taht it involve a double discretization. First, the physical domain Ω is discretized into a collection of $\mathcal{N}_e$ elements

$$\mathcal{T}_e = \bigcup_{i=1}^{\mathcal{N}_e} \Omega_i \tag{1}$$

called a mesh. Then, the continuous function space $V(\Omega)$ containing the solution u of a given PDE is approximated using a finite expansion into polynomials in space and finite differences in time. 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. In this paper, we will only focus on h-refinement.

*Correspondence to: remacle@gce.ucl.ac.be

The shallow water equations (SWEs), which describe the inviscid flow of a thin layer of fluid in two dimensions, have been used for many years by both the atmospheric modeling and by the hydraulic communities as a vehicle for testing promising numerical methods for solving atmospheric, oceanic, dam breaking and river flow problems.

It is only recently that the DGM has been applied to shallow water equations. Schwanenberg et al. [10] have developped a local DGM for shallow water equations where they use the Harten and Lax numerical flux [**?**]. In [4], Giraldo et al. use a fast quadrature free DGM for solving the spherical SWEs. In this, our aim is to show how h-adaptivity is able to provide highly accurate solutions of ths SWEs. For the spatial discretization of the unknowns, we choose an orthogonal basis that diagonalizes the mass matrix and, thus, simplifies its evaluation (§**??**). Free surface allow gravity wave (sound waves) propagation at speed $c_g = \sqrt{gh}$ where h is the fluid depth and g is the acceleration of gravity. In our examples, water depth is sufficiently small so that $c_g = \mathcal{O}||\mathbf{v}||$ where $\mathbf{v}$ is the fluid velocity. Typically, when the sound waves have the same propagation speed as the material waves, an explicit time integration is well adapted for computing. We use here a second order explicit TVD Runge-Kutta time integration scheme [1].

Transient computation of flows including moving features like abrupt wave fronts are applications where adaptivity in time is crucial. Without explicit interface tracking [3], h-adaptivity will certainly be necessary to accurately represent the complex evolution of waves. We present procedures to perform adaptive computations where the discretization space V_h changes in time. Both conforming and non-conforming adaptatation schemes are presented and compared with known results and experiments.

2. Discontinuous Galerkin Formulation of Shallow Water Equations

2.1. Continuous Formulation

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

$$\partial_t \mathbf{u} \; + \; \mathbf{div}\, \vec{\mathbf{F}}(\mathbf{u}) \; = \; \mathbf{r}. \tag{2}$$

Here $\mathbf{div} = (\nabla \cdot , \ldots , \nabla \cdot)$ 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 $\vec{F}_i(\mathbf{u}) : \left(\mathsf{H}^1(\Omega)\right)^m \rightarrow \mathsf{H}(\mathrm{d}iv, \Omega)$. Function space $\mathsf{H}(\mathrm{d}iv, \Omega)$ consists of square integrable vector valued functions whose divergence is also square integrable i.e.,

$$\mathsf{H}(\mathrm{d}iv, \Omega) = \left\{ \, \vec{v} \mid \vec{v} \; \in \; \mathsf{L}^2(\Omega)^2 \;\; , \;\; \nabla \cdot \vec{v} \in \mathsf{L}^2(\Omega) \, \right\}.$$

With the aim of constructing a Galerkin form of (2), multiply equation (2) by a test function $\mathbf{w} \in V(\Omega)$, integrate over Ω to obtain

$$\int_\Omega \partial_t \mathbf{u} \; \mathbf{w} \; dv + \int_\Omega \mathbf{div}\, \vec{\mathbf{F}}(\mathbf{u}) \; \cdot \mathbf{w} \; dv = \int_\Omega \mathbf{r} \; \mathbf{w} \; dv \; , \quad \forall \mathbf{w} \in V(\Omega). \tag{3}$$

We then use the divergence theorem to obtain the following variational formulation

$$\int_\Omega \partial_t \mathbf{u}\ \mathbf{w}\ dv - \int_\Omega \vec{\mathbf{F}}(\mathbf{u}) \cdot \nabla \mathbf{w}\ dv + \int_{\partial\Omega} \vec{\mathbf{F}}(\mathbf{u}) \cdot \vec{n}\ \mathbf{w}\ ds = \int_\Omega \mathbf{r}\ \mathbf{w}\ dv\ , \quad \forall \mathbf{w} \in V(\Omega). \tag{4}$$

2.2. Discrete Formulation

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

$$\mathcal{T}_e = \bigcup_{e=1}^{\mathcal{N}_e} e \tag{5}$$

called a mesh. This first step is the one of *geometrical discretization*.

Then, the continuous function spaces (infinite dimensional) are replaced by finite dimensional expansions. The difference between the DGM and classical FEMs is that the solution is approximated in each element separately: no *a priori* continuity requirements are needed. The discrete solution may then be discontinuous at inter-element boundaries. Figure 1 shows a typical situation of three elements e_1, e_2 and e_3. The approximated field u is smooth in each element but may be discontinuous at inter-element boundaries.

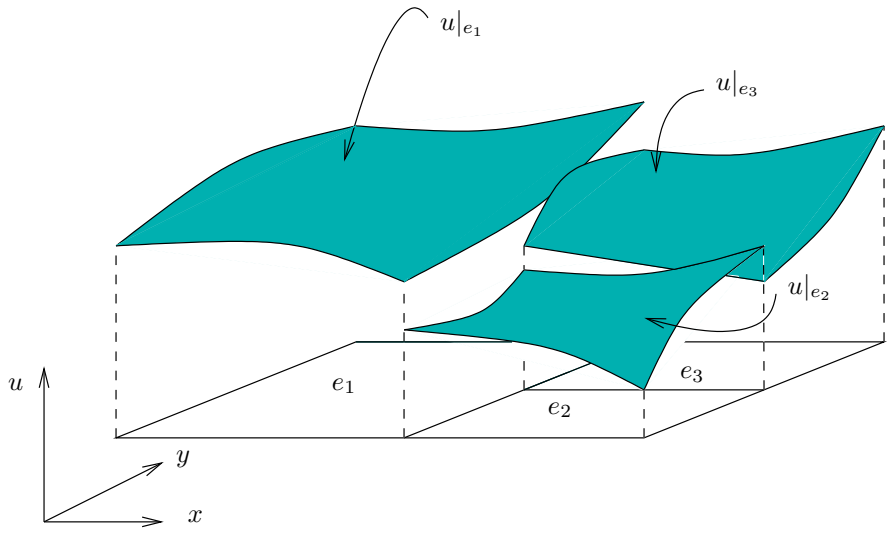


Figure 1. Three elements e_1, e_2 and e_3 and the piecewise discontinuous solution u.

In each element, we have m field components u_i, $i = 1, \ldots, m$. In the case of shallow water equations, we have $m = 3$ unknown fields: the water height h and the 2 components of the fluid velocity v_x and v_y. Here, each field is approximated with the same discrete space. This simplification is done without any loss of generality.

In each element, it is usually polynomial spaces that are chosen for approximating the u_i's. We note

$$\mathbb{P}^k = \text{span}\left\{x^l y^m, 0 \leq l, ml + m \leq k\right\}$$

the space of complete polynomials of total degree at most k. The dimension $d = \dim \mathbb{P}^k$ is $d = (k+1)(k+2)/2$. Again, we note

$$\mathbb{Q}^k = \text{span}\left\{x^l y^m, 0 \leq l, m \leq k\right\}$$

the space of complete polynomials of degree at most k. The dimension $d = \dim \mathbb{Q}^k$ is $d = (k+1)^2$. The approximation of component u_i over element e, noted u_i^e is written

$$u_i^e \simeq \sum_{j=1}^{d} \phi_j(x,y) U(e,i,j)$$

where the $U(e,i,j)$ are the coefficients of the approximation or degrees of freedom. In each element, we have $m \times d$ coefficients, and, because we consider that all approximations are disconnected, there are $\mathcal{N}_e \times m \times d$ coefficients for the whole mesh. We also have

$$B = \{\phi_1, \ldots, \phi_d\}$$

that is a basis of $\mathbb{P}^k$ if e is a triangle or of $\mathbb{Q}^k$ if e is a quadrangle. Usual FEM have limited choices for B due to the continuity requirements of the approximation. Nodal, hierarchical, Serendip or non-conforming basis are among the usual basis for classical FEMs. In case of the DGM, there are no limitations for the choice of the ϕ_i's. In fact, any basis B can be used and, in terms of the numerical solution, all basis for the same polynomial degree k are strictly equivalent i.e. leads to the same numerical soluion. For example, $\mathbb{P}^1$ can be spanned with $B = \{1, x, y\}$, $B = \{1 - x - y, x, y\}$ or any other combination. Even if the numerical solution does not depend on the choice of the basis B, it is advantageous to use bases [7] that have the following L^2-orthogonality property

$$\int_e \phi_i \phi_j \ dv = \delta_{ij}.$$

For quadrangles, orthogonal bases are constructed as tensor products of Legendre polynomials. For triangles, Dubiner [**?**] or Remacle [7] have developped orthogonal basis through Gram-Schmidt orthogonalization. For example, one of the many L^2-orthogonal expansion of $\mathbb{P}^1$ can be written in the cannonical triangle as

$$B = \left\{\sqrt{2}, -2 + 6x, -2\sqrt{3}(1 - x - 2y)\right\}.$$

For each field component u_i^e in element e , we have to solve the following d equations (one equation per ϕ_j)

$$\int_e \partial_t u_i^e \ \phi_j \ dv - \int_e \left[\vec{F}_i(\mathbf{u}^e) \ \nabla\phi_j + r_i \ \phi_j\right] \ dv + \int_{\partial e} \vec{F}_i(\mathbf{u}) \cdot \vec{n} \ \phi_j \ ds = 0 \ \ \forall \phi_j. \tag{6}$$

If the ϕ_j's are orthonormal over the reference element and if V_e is the volume of element e, (6) simplifies into

$$V_e \partial_t U(e,i,j) = \int_e \left[\vec{F}_i(\mathbf{u}^e) \ \nabla\phi_j + r_i \ \phi_j\right] \ dv - \int_{\partial e} \vec{F}_i(\mathbf{u}) \cdot \vec{n} \ \phi_j \ ds \ \ \forall \phi_j. \tag{7}$$

Note that equation (7) is correct only if elements have constant jacobians (straight sided triangles e.g.). Now, a discontinuous basis implies that $\mathbf{u}$ is not unique on ∂e and, consequently,

that the normal trace $\vec{F}_i \cdot \vec{n}$ is not defined on ∂e. In this situation, a *numerical flux* f_i 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. With a numerical flux, equations (7) becomes

$$V_e \partial_t U(e,i,j) = \int_e \left[\vec{F}_i(\mathbf{u}^e) \ \nabla \phi_j + r_i \ \phi_j \right] \ dv - \sum_{k=1}^{n_e} \int_{\partial e_k} f_i(\mathbf{u}^e, \mathbf{u}^{e_k}) \ \phi_j \ ds \ \ \forall \phi_j. \tag{8}$$

where n_e is the number of faces of element e.

2.3. Shallow water equations

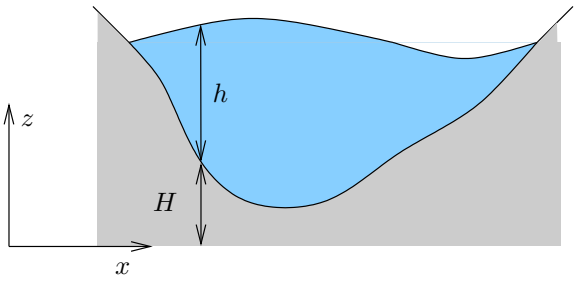


Figure 2. The water bed depth H and the water height h.

The movement of an incompressible fluid with constant density under the influence of a gravitational body force is considered. The description is basically inviscid except for the possible inclusion of a viscous bottom friction term. Vertical accelerations of the fluid are neglected, which allows to integrate the remaining part of the vertical momentum equation and to obtain an expression for the pressure which in turn can then be eliminated from the system. The error associated with this approximation is of the order of h^2/l^2 (h undisturbed water height, l characteristic length scale of the waves in x-direction). This estimate is equivalent to the so-called "long-wave limit" of wave motion, i.e. we are dealing with either very long waves or with shallow water. Physically, the horizontal velocity that is retained can be interpreted as a vertical average of the fluid velocity. The shallow water equations have the form (2) with $m = 3$,

$$\mathbf{u} = \left\{ \begin{array}{c} h \\ hv_x \\ hv_y \end{array} \right\} \quad , \quad \vec{\mathbf{F}}(\mathbf{u}) = \left\{ \begin{array}{c} h\vec{v} \\ hv_x\vec{v} + \frac{1}{2}gh^2\vec{e}_x \\ hv_y\vec{v} + \frac{1}{2}gh^2\vec{e}_y \end{array} \right\}$$

and

$$\mathbf{r} = \left\{ \begin{array}{c} 0 \\ -gh(\partial_x H + S_{fx}) \\ -gh(\partial_y H + S_{fy}) \end{array} \right\}.$$

Here h is the water depth above the water bed, H is the water bed depth (see Figure 2), $\vec{v}$ the water velocity, g the acceleration of gravity, $\vec{e}_x$ and $\vec{e}_y$ are the unit vectors in the x and

y directions, respectively and $v_x = \vec{v} \cdot \vec{e}_x$ and $v_y = \vec{v} \cdot \vec{e}_y$. The two components of the bottom friction S_{fx} and S_{fy} that can be expressed by the Manning formula

$$S_{fx} = \frac{n^2 v_x \sqrt{v_x^2 + v_y^2}}{h^{4/3}} \quad , \quad S_{fy} = \frac{n^2 v_y \sqrt{v_x^2 + v_y^2}}{h^{4/3}}.$$

where n is an empirical roughness coefficient.

2.4. Numerical Flux

We concentrate on the time evolution of our flow model from an initial state that consists of two semi-infinite uniform zones which are separated by a discontinuity: this setup is usually referred as a *Riemann problem* [**?**, 5]. One can imagine a realization of this situation by positioning a diaphragm ("an infinitely thin dam") between the two fluid states and somehow rupture it at time $t = 0$. Our objective is to determine the resulting induced wave motion as a function of the initial state. This problem is geometrically one- dimensional in that the solution only depends on one space coordinate normal to the diaphragm. The Riemann problem related to non linear hyperbolic systems is usually hard to compute because of the non linearity of fluxes. Usually, it is only available numerically through a Newton-Raphson iteration.

The idea of using solution of Riemann problems for solving non linear conservation laws numerically is from Godunov [**?**]. At the time, it was of course in the context of finite differences. Following the idea of Godounov, we consider that each edge ∂e of the mesh is a diaphragm separating two states: the solutions $\mathbf{u}$ in the two elements neighboring ∂e. The numerical flux $\mathbf{f}$ is then computed using the solution $\mathbf{u}^R$ of the associated Riemann problem:

$$\mathbf{f} = \vec{\mathbf{F}}(\mathbf{u}^R) \cdot \vec{n}.$$

For linear problems, this technique consist in taking the "fully upwind flux" and therefore the resulting scheme is stable. For non linear problems, it has been proven that the choice of Godunov fluxes are ensuring that the numerical solution satisfy the entropy condition [5].

Practically, solving exactly Riemann problems at each mesh edge is complex and computationally prohibitive. In most of the cases (except perhaps in the context of high speed compressible flows with very strong shocks [11]), an approximate solution to the Riemann problem is sufficient. Approximate Riemann solvers are proved to produce always more numerical dissipation than Godunov fluxes. Hence, numerical experience suggests that the choice of a given numerical flux (that respect a discrete entropy condition) does not have a significant impact on the accuracy of the solution, especially when polynomial degree k increases.

For computing the numerical flux $\mathbf{f}$, a very pragmatic and successful approach has been taken by Roe [9]. The exact solution to a linearized Riemann problem is constructed. The Roe numerical flux for shallow water equations is written as

$$\mathbf{f}(\mathbf{u}_e, \mathbf{u}_{e_k}) = \frac{1}{2}\left[(1 + \mathrm{Fr})\ \vec{\mathbf{F}}(\mathbf{u}_e) + (1 - \mathrm{Fr})\ \vec{\mathbf{F}}(\mathbf{u}_{e_k})\right] \cdot\ \vec{n} + c_A(1 + \mathrm{Fr}^2)\left[\mathbf{u}_e - \mathbf{u}_{e_k}\right] \tag{9}$$

In (9), the average Froude number

$$\mathrm{Fr} = \frac{\vec{v}_A \cdot \vec{n}}{c_A}$$

is computed using Roe averages i.e.

$$(v_x)_A = \frac{(v_x)_e\sqrt{h_e} + (v_x)_{e_k}\sqrt{h_{e_k}}}{\sqrt{h_e}+\sqrt{h_{e_k}}} \quad , \quad (v_y)_A = \frac{(v_y)_e\sqrt{h_e} + (v_y)_{e_k}\sqrt{h_{e_k}}}{\sqrt{h_e}+\sqrt{h_{e_k}}} \quad ,$$

$$\vec{v}_A = \{(v_x)_A, (v_y)_A\}^T \text{ and } c_A = \sqrt{g\frac{1}{2}(h_e + h_{e_k})}.$$

The absolute value of the Froude number Fr is bounded by 1, we correct its value in order to fullfill this physical constraint: if $\mathrm{Fr} > 1$, $\mathrm{Fr} = 1$ and $\mathrm{Fr} < 1$, $\mathrm{Fr} = -1$. The situation is similar in the case of the Euler equations, as noted in the original paper by Roe [**?**]. The flux is the sum of a high order centered term plus a dissipation term of order zero:

$$\begin{aligned} \mathbf{f} \quad = \quad & \underbrace{\frac{1}{2}\left[\vec{\mathbf{F}}(\mathbf{u}_e) + \vec{\mathbf{F}}(\mathbf{u}_{e_k})\right]\cdot\ \vec{n}}_{\text{Centered differences}} + \\ & \underbrace{\frac{1}{2}\mathrm{Fr}\left[\vec{\mathbf{F}}(\mathbf{u}_e) - \vec{\mathbf{F}}(\mathbf{u}_{e_k})\right]\cdot\ \vec{n} + c_A(1+\mathrm{Fr}^2)\left[\mathbf{u}_e - \mathbf{u}_{e_k}\right]}_{\text{Dissipation}}. \end{aligned}$$

It has been proven in [**?**] that, in smooth regions, $\mathbf{u}_e - \mathbf{u}_{e_k} = \mathcal{O}\left(h^{k+1}\right)$ so that the dissipation introduced by the upwinding does not exceed the truncature error of the scheme. Near discontinuities or in badly resolved zones where the mesh is too coarse, $\mathbf{u}_e - \mathbf{u}_{e_k} = \mathcal{O}(h)$ and the scheme is first order, as expected.

3. Anisotropic Mesh Adaptation

3.1. Definition of the Metric Field $\mathcal{M}$

The goal of our mesh adaptation process is to determine the anisotropic mesh configuration that will most effectively provide the level of accuracy required for the parameters of interest. The strategy adopted in the present paper is to construct an optimal anisotropic mesh through a metric field $\mathcal{M}(\vec{x})$, $\vec{x} \in \Omega$ defined over the domain of the analysis Ω. The goal of a mesh adaptation is to build a mesh where every edge is of size 1, using the non uniform measure of distance defined by the metric $\mathcal{M}$. If $\vec{y}$ is the vector representing an edge with its Euclidian coordinates y_1, y_2, y_3, the optimum mesh is characterized by:

$$\vec{y}^T\mathcal{M}\vec{y} = 1 \quad \forall \vec{y}.$$

The metric $\mathcal{M}$ is a symmetric covariant tensor with all its eigenvalues $\lambda_1, \lambda_2, \lambda_3$ positive. If $\vec{E}_1$, $\vec{E}_2$, $\vec{E}_3$ are the orthogonal unit eigenvectors associated with the eigenvalues $\lambda_1, \lambda_2, \lambda_3$, the metric tensor can be written as

$$\mathcal{M} = \mathcal{R}^T\Lambda\mathcal{R}$$

with

$$\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \lambda_3)$$

and

$$\mathcal{R} = \left[\begin{array}{c} \vec{E}_1 \\ \vec{E}_2 \\ \vec{E}_3 \end{array} \right].$$

The interpretation of mesh optimality in terms of unit edges is then equivalent to

$$\vec{y}^T \mathcal{R}^T \Lambda \mathcal{R} \vec{y} = 1 \quad \forall \vec{y}.$$

where $\mathcal{R}\vec{y} = \{Y_1, Y_2, Y_3\}^T$ are the coordinates of $\vec{y}$ in the principal axis of the metric $\mathcal{M}$ i.e. (see Figure 3)

$$\vec{y} = Y_1 \vec{E}_1 + Y_2 \vec{E}_2 + Y_3 \vec{E}_3.$$

The optimality is then re-written as

$$Y_1^2 \lambda_1 + Y_2^2 \lambda_2 + Y_3^2 \lambda_3 = 1$$

which means that, at each point of the domain, an edge is of optimal size if it extremity sits on an ellipsoid of equation $Y_1^2 \lambda_1 + Y_2^2 \lambda_2 + Y_3^2 \lambda_3 = 1$ (see Figure 3). The eigenvalues of the

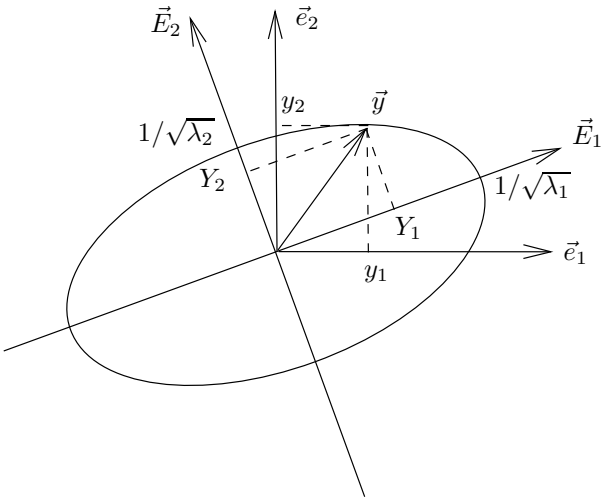


Figure 3. Interpretation in 2D of the different parameters in the definition of the metric $\mathcal{M}$.

metric are then direcly related to desired mesh sizes in the principal directions of the metric $\mathcal{M}$.

3.2. Computing mesh sizes

Those 3 sizes $s_1 = 1/\sqrt{\lambda_1}$, $s_2 = 1/\sqrt{\lambda_2}$ and $s_3 = 1/\sqrt{\lambda_3}$ are computed using an error indication procedure. In smooth regions, i.e. regions far from hydraulic jumps, the error is computed using the tensor of second order derivatives of the water height h:

$$H_{ij} = \frac{\partial^2 h}{\partial x_i \partial x_j}.$$

The Hessian H is computed at each vertex using a patchwise linear reconstruction of the gradients ∇h [8]. The Hessian is symmetric and we have then the orthogonal decomposition

$$H = \mathcal{R}^T \bar{\Lambda} \mathcal{R}$$

with

$$\bar{\Lambda} = \mathrm{diag}(\bar{\lambda_1}, \bar{\lambda_2}, \bar{\lambda_3})$$

If we suppose that the discretization error e in the direction of the ith eigenvector of H is proportional to $\bar{\lambda_i}$, i.e. $e = c\lambda i$, we can compute a mesh with equidistributed error by choosing

$$\lambda_i = \frac{c\,|\bar{\lambda_i}|}{\epsilon}$$

where ϵ is a targetted error that is defined by the user and where c is the coefficient of proportinality. It is usual to defined maximal and minimal element sizes s_{max} and s_{min} that prevent to build up non realistic metric fields. The metric is then corrected by

$$\lambda_i = \max\left(\min\left(\frac{|\bar{\lambda_i}|}{\epsilon}, \frac{1}{s_{min}^2}\right), \frac{1}{s_{max}^2}\right).$$

In [], we have shown how to detect efficiently non smooth regions in DGM computations. Our discontinuity detector has the following form:

$$\mathcal{I}_e = \frac{\sum_{k=1}^{n_e} |\int\limits_{\partial e_k} (h^e - h^{e_k})\, ds|}{s_e^{(p+1)/2}\, |\partial e|\, \|h^e\|}. \tag{10}$$

In examples, we choose s_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 can show that $\mathcal{I}_e \to 0$ as either $s_e \to 0$ or $p \to \infty$ in smooth solution regions, whereas $\mathcal{I}_e \to \infty$ near a discontinuity. Thus, the discontinuity detection scheme is

$$\begin{cases} \text{if } \mathcal{I}_e > 1, & h \text{ is discontinuous} \\ \text{if } \mathcal{I}_e < 1, & h \text{ is smooth} \end{cases}. \tag{11}$$

Near discontinuities, Hessians are highly ill conditionned. Whe a discontinuity is detected, we use the gradients ∇h in order to compute the metric field. Because we know that, without diffusion, a hydraulic jump has an infinitely small thickness, we use the minimal allowable mesh size s_{min} in the direction of the gradient and the maximal size on the other direction.

A metric smoother is crucial at this point in orer to reconnect smooth and discontinuous regions. The algorithm is described in [8].

3.3. Mesh Adaptation using Local Mesh modifications

In our approach, the mesh is adapted using local mesh modification operators that enable fast and accurate solution transfers. Given the mesh metric field $\mathcal{M}$ defined over the domain, the goal is to apply some local mesh modification operators to yield a mesh of the same quality as would be obtained by an anisotropic domain remeshing procedure. In every mesh modification

operator, one cavity triangulation $\mathcal{C}$, i.e. a set of mesh entities that form a connected volume, is replaced by another cavity triangulation $\mathcal{C}'$ with the same closure. Formally, we write

$$\mathcal{T}^{n+1} = \mathcal{T}^n + \mathcal{C}' - \mathcal{C}. \tag{12}$$

where $\mathcal{T}^n$ denotes the mesh before the local mesh modification and $\mathcal{T}^{n+1}$ is the mesh after the local mesh modification. In the kernel of each mesh modification, we ensure that there is a moment when both cavity triangulations are present. We are able then to rewrite (12) as a two step procedure:

$$\mathcal{T}' = \mathcal{T}^n + \mathcal{C}'. \tag{13}$$
$$\mathcal{T}^{n+1} = \mathcal{T}' - \mathcal{C}. \tag{14}$$

$\mathcal{T}'$ represents a mesh that is topologically incorrect but both cavity triangulations $\mathcal{C}$ and $\mathcal{C}'$ are present so that we can insert a callback to the solver that aim is to transfer the solution from $\mathcal{C}$ to $\mathcal{C}'$. One mesh modification plus solution transfer operation is done in 3 steps:

$$\mathcal{T}' = \mathcal{T}^n + \mathcal{C}'. \tag{15}$$
$$\Pi_{\mathsf{L}^2}(\mathbf{u}, \mathcal{C}, \mathcal{C}') \tag{16}$$
$$\mathcal{T}^{n+1} = \mathcal{T}' - \mathcal{C}. \tag{17}$$

where $\Pi_{\mathsf{L}^2}(\mathbf{u}, \mathcal{C}, \mathcal{C}')$ is, in our case, the L^2 projection of the solution $\mathbf{u}$ form the cavity $\mathcal{C}$ to $\mathcal{C}'$. The L^2 projection of the the solution from $\mathcal{C}$ to $\mathcal{C}'$ ensures that conservative quantities like mass or momentum's are conserved through the solution transfer process.

In two dimensions, we only use three mesh modification operators.

3.3.1. Edge splitting The first mesh modification operator is the edge splitting. It consists in modifying a cavity $\mathcal{C}$ composed of 2 neighboring elements e_1 end e_2 by splitting the common edge edge and replacing e_1 and e_2 by e_3, e_4, e_5 and e_6 as represented in Figure4.

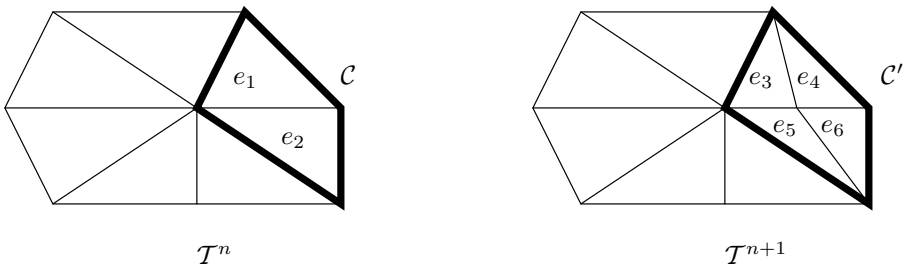


Figure 4. Edge splitting in 2D.

3.3.2. Edge collapsing The second mesh modification operator is the edge collapsing. An edge $\vec{y}$ is too long so it is decided to remove it. The origin of the edge is moved to its end, modifying a cavity $\mathcal{C}$ composed of all the triangles neighboring the origin of the edge (see Figure5).

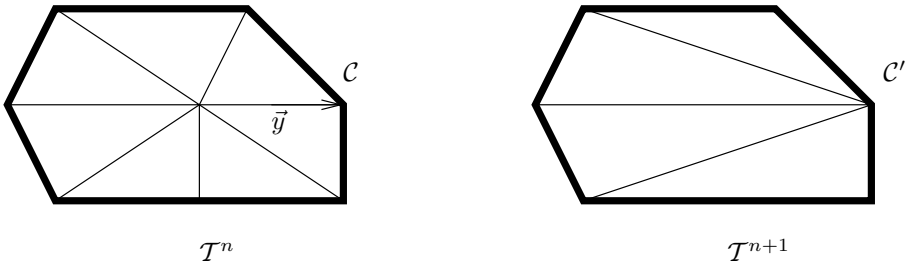


Figure 5. Edge collapsing in 2D.

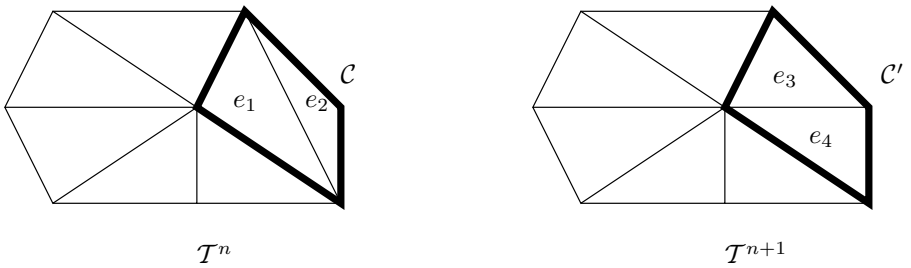


Figure 6. Edge swapping in 2D. An edge separating 2 triangles is replaced by the opposite edge.

3.3.3. Edge swapping The last mesh modification operator is the edge swapping. It consists in modifying a cavity $\mathcal{C}$ composed of 2 neighboring elements e_1 end e_2 by swapping the edge and replacing e_1 and e_2 by e_3 and e_4 as represented in Figure6.

The mesh adaptation algorithm, who is indeed quite complex, is described in [**?**]. Edge collapsing's are used to coarsen the mesh in regions of low error while edge splitting's are used to refine the mesh in regions of high error. Edge swapping's are used to align the mesh to the metric axes and to enhance mesh quality. Note that we do not consider here vertex repositioning in our mesh modification procedure. What is interresing us here is the consequense of the numerous local mesh modifications on the quality of the numerical solution. Questions are

- Does the adaptive procedure introduce excessive numerical dissipation?
- Does the adaptive procedure introduce oscillations?
- Does the adaptive procedure introduce some loss of conservativity?

The edge splitting procedure does not change the numerical solution Consequently it does not introduce any sort of problem.

While performing an edge collapsing, some information is lost because the discrete space where lives the numerical solution is made smaller. Some numerical dissipation may then be introduced by the operation. The L^2 projection $\Pi_{\mathsf{L}^2}(\mathbf{u}, \mathcal{C}, \mathcal{C}')$ may also introduce Gibbs osciallations (overshoots) if $p > 0$. Because we use L^2 projections, the conservative quantities are transferred correctly from $\mathcal{C}$ to $\mathcal{C}'$). Note that, for target cavity $\mathcal{C}'$, the solution over $\mathcal{C}$ is piecewise discontinuous and, consequently, we have to be careful and compute the L^2 projection accurately. Despite of that, we do not excpect that edge collapsing's will cause any problem: edge collapsing's are performed in regions of low error only and, therefore, the

resulting modifications of the solution dued to the operation should not have some significant impact.

The edge swapping is certainly the most controversary. Edge swappings are performed everywhere, and then also in regions of high error. Similarly to edge collapsing, edge swapping is an operation that modifies the solution and therefore introduce errors. Conservation should not be an issue.

4. Results

4.1. Radial dam break problem.

Consider the SWEs with piecewise initial data

$$h(r) = \left\{ \begin{array}{lll} 2 & \text{if} & r < 1 \\ 1 & \text{if} & r > 1 \end{array} \right. ,$$

$r = \sqrt{x^2 + y^2}$ being the radial coordinate. The fluid is initially at rest i.e. $v_x = v_y = 0$. This is the problem of a dam break i.e. two zones at rest (null velocity) that are separated by an interface (the dam), the water height being different in these two zones. At $t = 0$, the interface is impulsively removed (the dam breaks). The dam break Riemann problem for SWEs has solution that consists in a rarefaction wave and a hydraulic jump (shock). The rarefaction wave moves into the direction of the highest water depth (i.e. radially, towards the origin) while the shock moves into the direction of low water depth. Once the rarefaction wave hits the origin $r = 0$, it is reflected and the fluid flows back outwards. A second second shock rapidely forms. This problem has an interresting structure and will be used to verify our refinement methodology:

- Verify if multiple mesh refinements and coarsenings preserve conservation i.e. verify if the adaptive scheme is able to compute the right speed for the waves;
- Show that the anisotropic refinement produces optimal discrteisations i.e. verify that anisotropic refinement is cheaper than isotropic refinement or no refinement.

For that purpose, we need the exact solution of the problem. There is no analytical solution for the radial dam break problem. Our reference solution was computed numerically by solving the one dimensional equations

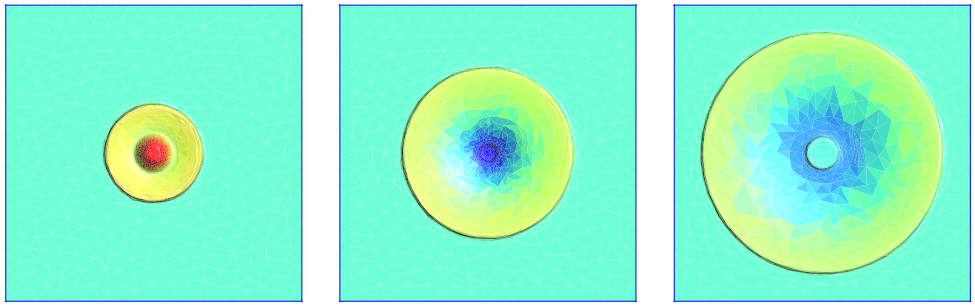
$$\begin{array}{rcl} \dfrac{\partial h}{\partial t} + \dfrac{\partial}{\partial r}\left(h v_r\right) & = & -\dfrac{h v_r}{r} \\ \dfrac{\partial v_r}{\partial t} + \dfrac{\partial}{\partial r}\left(h v_r^2 + \dfrac{1}{2} g h^2\right) & = & -\dfrac{h v_r^2}{r} \end{array} \tag{18}$$

with a high resolution finite volume scheme on a very fine grid of 2000 points [**?**].

We have done three computations of the problem. The first one has been done using an uniform mesh with element sizes of $1/200^{\text{th}}$ of the size of the domain i.e. 0.025. The uniform mesh has about 20.000 nodes and 40.000 elements. The second computation was done using isotropic refinement i.e. using the same mesh size (the smallest one) in all directions. The mesh was adapted every 0.01 second. Final time of the computation was $t = 1.25$ so that 125 mesh

adaptations were performed. The minimum mesh size $s_{min} = 5/200$ i.e. $1/200^{\text{th}}$ of the size of the problem. The maximum mesh size $s_{max} = 5/20$ i.e. $1/20^{\text{th}}$ of the size of the problem. The third computation was done using anisotropic refinement. Adaptation parameters were the same as for the isotropic refinement. We say that the three problems have the same effective discretisation i.e. have the same minimum mesh size.

Figures 7 and 10 show meshes and plots of water height h at different time steps for an anisotropic simulation.



$t = 0.5$ $t = 1$ $t = 1.5$

Figure 7. Plots of the water height at different time steps. Those are the solution of the anisotropic run.

Figure 8 show plots of the water height along radial direction. Those results show that the positions of the waves are not deteriorated by the multiple mesh adaptations. We have shown in §3.3 that the kind of projections we were using will not cause any sort of problem in term of conservation. Another important result is that the solution does not seem to be deteriorated by excessive numerical diffusion: solutions with and without refinement look similar.

In order to refine the comparison between solutions with mesh refinement and with the uniform mesh, we have computed (Table 4.1) the L^1 relative norm of the error

$$\epsilon = \frac{\int_0^5 |h - h_{ex}| dr}{\int_0^5 |h_{ex}| dr}$$

for both uniform and adapted meshes (h_{ex} is the "exact" solution computed with the finie 1D giid). We have plotted h and h_{ex} alongn the radial direction (Figure 8) We see that both uniform and adapted mesh have relative errors ϵ that are comparable.

Of course, there is about 40 times less elements in the anisotropicllay (see Figure 10) adapted grid than in the uniform grid and about 4 times (see Figure 9) less elements than in the isotropically adapted grid. One can remark that the gain in terms of element number grows with time. At $t = 0.5$, the isotropic mesh has 5386 triangles while the anisotropic one has 2078. At time $t = 1.5$, the isotropic mesh has now 9510 triangles while the anisotropic one only 1932. At early stages of the computation, the radius of curvature of the shock is small so that elements with large aspect ratio are impossible to construct, at least with the given mesh size parameter s_{min}. Small sizes are required in the radial direction in order to capture the shock and in the azimuthal direction in order to capture the curved shape of the shock. The use

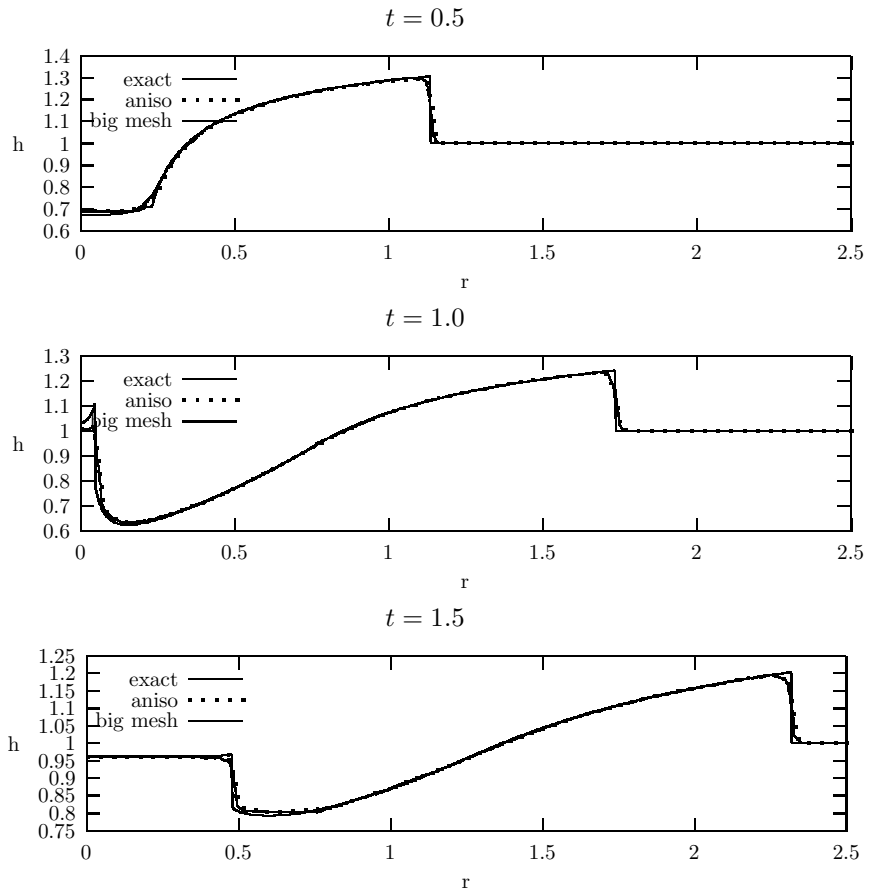
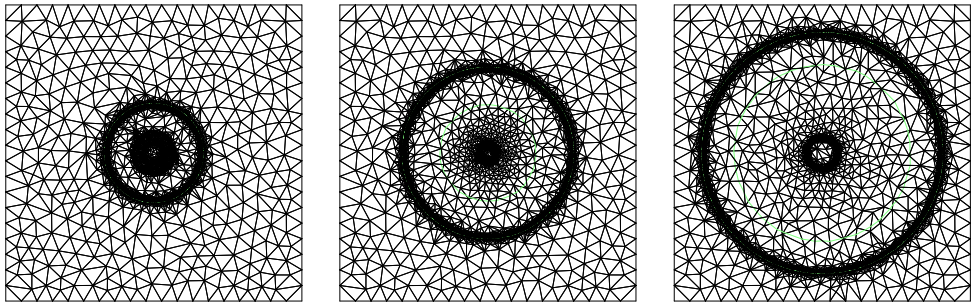


Figure 8. Comparison at different times between exact solution, solution on the uniform grid ans solution on the anisotropically adapted grid.

	Uniform mesh Uniform mesh	Adapted mesh Isotropic	Adapted mesh Anisotropic
$t = 0.00$	1.10758E-03	1.52250E-03	1.56324E-03
$t = 0.25$	3.84628E-03	3.26702E-03	4.12431E-03
$t = 0.50$	3.65177E-03	3.00082E-03	3.42923E-03
$t = 0.75$	4.09531E-03	3.40864E-03	4.07662E-03
$t = 1.00$	4.94453E-03	4.73405E-03	5.09021E-03
$t = 1.25$	3.94322E-03	3.65974E-03	3.85844E-03
$t = 1.50$	3.26890E-03	3.73175E-03	3.66172E-03

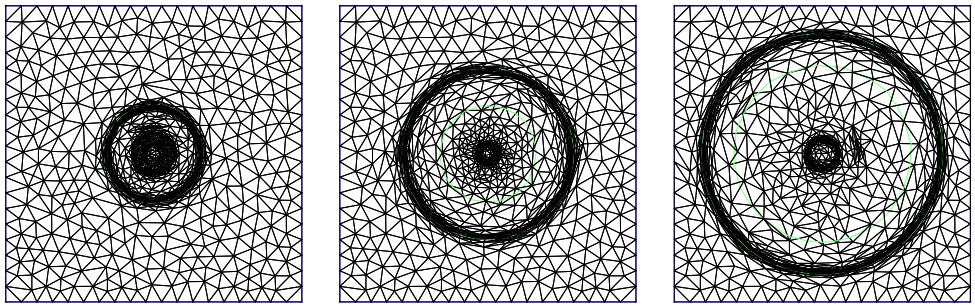
Table I. Relative error ϵ for the radial dam break.

of curved elements could certainly help here. When the shock evolves, it grows radially and its curvature decreases. More anisotropic elements can then be constructed and the efficienty of the procedure gets better. In fact, it seems that the number of elements in the shock remains almost constant in the process so that the total number of element does not grow in the anisotropic refinement case. In the isotropic case, the number of elements in the shock grows linearly with the distance of the shock to the origin. This effect will be more dramatic in 3D. There, the number of elements in a shock will grow quadratically in the isotropic case while remaining constant in the anisotropic case.



$t = 0.5$, 5386 triangles, $t = 1$, 7198 triangles, $t = 1.5$, 9510 triangles

Figure 9. Adapted meshes for the isotropic refinement case.



$t = 0.5$, 2078 triangles, $t = 1$, 1920 triangles, $t = 1.5$, 1932 triangles

Figure 10. Adapted meshes for the anisotropic refinement case.

4.2. Dam breaking in presence of a building

Comparison with experimental data...

5. Conclusions

REFERENCES

1. 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. *Mathematics of Computations*, 54:545–581, 1990.
2. B. Cockburn, G. Karniadakis, and C.-W. Shu, editors. *Discontinuous Galerkin Methods*, volume 11 of *Lecture Notes in Computational Science and Engineering*, Berlin, 2000. Springer.
3. F. Furtado, J. Glimm andJ. Grove, X.-L. Li, W.B. Lindquist, R. Menikoff, D.H. Sharp, and Q. Zhang. Front tracking and the interaction of nonlinear hyperbolic waves. *Lecture Notes in Engineering*, 43:99–111, 1989.
4. F. Giraldo, J. Hestheaven, and T. Wartburton. Nodal high-order discontinuous galerkin methods for the spherical shallow water equations. *Journal of Computational Physics*, 181:499–525, 2002.
5. R. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser-Verlag, 1992.
6. W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. *Tech. Report LA-UR-73-479, Los Alamos Scientific Laboratory*, 1973.
7. 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 Review*, 45(1):53–72, 2003.
8. J.-F. Remacle and M. S. Shephard. An algorithm oriented mesh database. *International Journal for Numerical Methods in Engineering*, 58(2):349–374, 2003.
9. P.L. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics*, 43:357–372, 1981.
10. D. SCHWANENBERG and J. KONGETER. A discontinuous galerkin method for the shallow water equations with source terms. In B. Cockburn, G. Karniadakis, and C.-W. Shu, editors, *Discontinuous Galerkin Methods*, volume 11 of *Lecture Notes in Computational Science and Engineering*, pages 419–424, Berlin, 2000.
11. P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54:115–173, 1984.