INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **52**:903–923
Published online 16 March 2006 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1204

An adaptive discretization of shallow-water equations based on discontinuous Galerkin methods

Jean-François Remacle[1,*,†], Sandra Soares Frazão[1,2,‡], Xiangrong Li[3,§] and Mark S. Shephard[3,¶]

[1]*Department of Civil Engineering, Place du Levant 1, 1348 Louvain-la-Neuve, Belgium*
[2]*Fonds National de la Recherche Scientifique, rue dEgmont 5 B—1000 Bruxelles, Belgium*
[3]*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 © 2006 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 [1] 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].

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

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

*Correspondence to: J.-F. Remacle, Department of Civil Engineering, Place du Levant 1, 1348 Louvain-la-Neuve, Belgium.
†E-mail: remacle@gce.ucl.ac.be
‡E-mail: soares@gce.ucl.ac.be
§E-mail: xli@scorec.rpi.edu
¶E-mail: shephard@scorec.rpi.edu

Contract/grant sponsor: European commission

Received 17 April 2004
Revised 30 December 2005
Accepted 7 January 2006

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.

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 modelling 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 SWEs. Schwanenberg and Kongeter [3] have developed a local DGM for SWEs where they use the Harten and Lax numerical flux [4]. In their paper, Schwanenberg *et al.* use isotropic template-based mesh refinement that do not allow neither anisotropic meshes nor to unrefine more than the initial mesh.

Giraldo *et al.* [5] use a fast quadrature free DGM for solving the spherical SWEs.

Aizinger and Dawson [6], solve geophysical flow problems using the DGM. They introduce the Coriolis force together with some tidal forcings.

In this, our aim is to show how *h*-adaptivity is able to provide highly accurate solutions of the SWEs. For the spatial discretization of the unknowns, we choose an orthogonal basis that diagonalizes the mass matrix and, thus, simplifies its evaluation. The free surface allows 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 [7].

Transient computation of flows including moving features like abrupt wave fronts are applications where adaptivity in time is crucial. Without explicit interface tracking [8], *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 adaptation schemes are presented and compared with known results and experiments.

2. DISCONTINUOUS GALERKIN FORMULATION OF SWEs

2.1. Continuous formulation

Consider an open set $\Omega \subset \mathbb{R}^2$ whose boundary $\partial\Omega$ is Lipschitz continuous with a normal $\boldsymbol{n}$ that is defined almost everywhere. We seek to determine $\mathbf{u}(\Omega, t) : \mathbb{R}^2 \times \mathbb{R} \to \mathsf{L}^2(\Omega)^m = V(\Omega)$ 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

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

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

integrable i.e.

$$\mathrm{H}(\mathrm{div},\Omega)=\{\boldsymbol{v}\,|\,\boldsymbol{v}\in \mathrm{L}^2(\Omega)^2, \nabla\cdot\ \boldsymbol{v}\in \mathrm{L}^2(\Omega)\}$$

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

$$\int_\Omega \partial_t \mathbf{uw}\,\mathrm{d}v+\int_\Omega \mathbf{div}\,\vec{\mathbf{F}}(\mathbf{u})\cdot\mathbf{w}\,\mathrm{d}v=\int_\Omega \mathbf{rw}\,\mathrm{d}v \quad \forall \mathbf{w}\in V(\Omega) \tag{3}$$

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

$$\int_\Omega \partial_t \mathbf{uw}\,\mathrm{d}v-\int_\Omega \vec{\mathbf{F}}(\mathbf{u})\cdot\nabla\mathbf{w}\,\mathrm{d}v+\int_{\partial\Omega}\vec{\mathbf{F}}(\mathbf{u})\cdot\boldsymbol{n}\mathbf{w}\,\mathrm{d}s=\int_\Omega \mathbf{rw}\,\mathrm{d}v \quad \forall \mathbf{w}\in V(\Omega) \tag{4}$$

2.2. *Discrete formulation*

The physical domain Ω is discretized into a collection of $\mathscr{N}_e$ elements

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

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 for the DGM: 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.

In each element, we have m field components u_i, $i=1,\ldots,m$. In the case of SWEs, we have $m=3$ unknown fields: The water height h and the two components of the fluid velocity v_x and v_y. Here, each field is approximated with the same discrete space.

We note

$$\mathbb{P}^k=\mathrm{span}\{x^l y^m, 0\leqslant l, ml+m\leqslant k\}$$

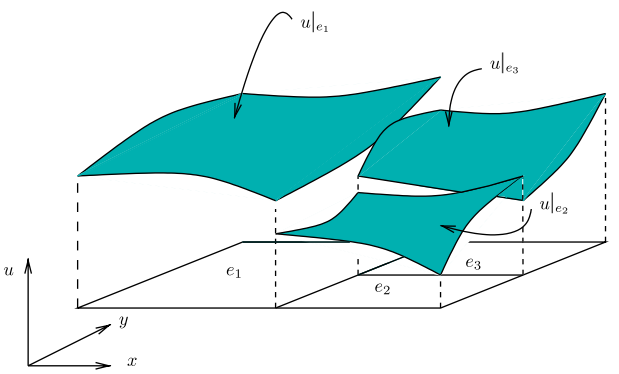

Figure 1. Three elements e_1, e_2 and e_3 and the piecewise discontinuous solution u.

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

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

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 $\mathbb{P}^k$. Usual FEM have limited choices for B due to the continuity requirements of the approximation. Nodal, hierarchical 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. lead to the same numerical solution. 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 [9] that have the following L^2-orthogonality property

$$\int_e \phi_i \phi_j \, \mathrm{d}v = \delta_{ij}$$

For quadrangles, orthogonal bases are constructed as tensor products of Legendre polynomials. For triangles, Dubiner [10] or Remacle *et al.* [9] have developed 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

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

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

$$\int_e \partial_t u_i^e \phi_j \, \mathrm{d}v - \int_e [\boldsymbol{F}_i(\mathbf{u}^e) \nabla \phi_j + r_i \phi_j] \, \mathrm{d}v + \int_{\partial e} \boldsymbol{F}_i(\mathbf{u}) \cdot \boldsymbol{n} \phi_j \, \mathrm{d}s = 0 \quad \forall \phi_j \tag{6}$$

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

$$V_e \partial_t U(e,i,j) = \int_e [\boldsymbol{F}_i(\mathbf{u}^e) \nabla \phi_j + r_i \phi_j] \, \mathrm{d}v - \int_{\partial e} \boldsymbol{F}_i(\mathbf{u}) \cdot \boldsymbol{n} \phi_j \, \mathrm{d}s \quad \forall \phi_j \tag{7}$$

Note that Equation (7) is correct only if elements have constant Jacobians (straight-sided triangles, for example). Now, a discontinuous basis implies that $\mathbf{u}$ is not unique on ∂e and, consequently, that the normal trace $\boldsymbol{F}_i \cdot \boldsymbol{n}$ 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 neighbouring 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, Equation (7) becomes

$$V_e \partial_t U(e,i,j) = \int_e [\boldsymbol{F}_i(\mathbf{u}^e)\nabla\phi_j + r_i\phi_j]\,\mathrm{d}v - \sum_{k=1}^{n_e} \int_{\partial e_k} f_i(\mathbf{u}^e, \mathbf{u}^{e_k})\phi_j\,\mathrm{d}s \quad \forall\phi_j \tag{8}$$

where n_e is the number of faces of element e.

2.3. Shallow-water equations

The movement of an incompressible fluid with constant density under the influence of a gravitational body force is considered. The behaviour is basically inviscid except for the possible inclusion of a viscous bottom friction term. Vertical accelerations of the fluid are neglected, which allows integrating 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, the undisturbed water height, l, the 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 SWEs have the form (2) with $m = 3$,

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

and

$$\mathbf{r} = \begin{Bmatrix} 0 \\ -gh(\partial_x H + S_{fx}) \\ -gh(\partial_y H + S_{fy}) \end{Bmatrix}$$

Here h is the water depth above the bed, H the bed elevation (see Figure 2), $\boldsymbol{v}$ the water velocity, g the acceleration of gravity, $\boldsymbol{e}_x$ and $\boldsymbol{e}_y$ are the unit vectors in the x and y directions, respectively, and $v_x = \boldsymbol{v}\cdot\boldsymbol{e}_x$ and $v_y = \boldsymbol{v}\cdot\boldsymbol{e}_y$. The two components of the bottom friction S_{fx} and

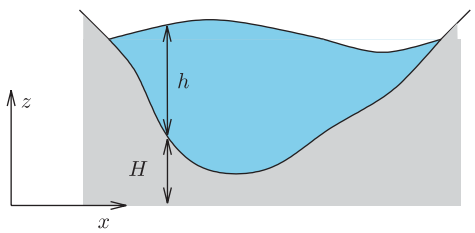

Figure 2. The water bed depth H and the water height h.

S_{fy} can be expressed by the Manning formula [11]

$$S_{fx} = \frac{n^2 v_x \sqrt{v_x^2 + v_y^2}}{h^{4/3}}, \quad S_{fy} = \frac{n^2 v_y \sqrt{v_x^2 + v_y^2}}{h^{4/3}} \tag{9}$$

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 set-up is usually referred as a *Riemann problem* [12, 13]. One can imagine a realization of this situation by positioning a diaphragm ('an infinitely thin dam') between the two fluid states and somehow rupturing 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 the solution of Riemann problems for solving non linear conservation laws numerically is from Godunov [14]. At the time, it was of course in the context of finite differences. Following the idea of Godunov, we consider that each edge ∂e of the mesh is a diaphragm separating two states: The solutions $\mathbf{u}$ in the two elements neighbouring ∂e. The numerical flux $\mathbf{f}$ is then computed using the solution $\mathbf{u}^{\mathrm{R}}$ of the associated Riemann problem:

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

For linear problems, this technique consists 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 satisfies the entropy condition [13].

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 [15]), an approximate solution to the Riemann problem is sufficient. Approximate Riemann solvers produce more numerical dissipation than Godunov fluxes. Hence, numerical experience suggests that the choice of a given numerical flux (that respects a discrete entropy condition) does not have a significant impact on the accuracy of the solution, especially when polynomial degree k increases.

Among the approximate Riemann solvers designed for computing the numerical flux $\mathbf{f}$, a very pragmatic and successful approach has been taken by Roe [16] and later extended to the SWEs [17, 18]. The approach consists in constructing the exact solution of a linearized Riemann problem at each mesh edge. In the framework of SWEs applied to water–wave propagation problems, an important parameter is the Froude number Fr, defined as the ratio between the water velocity v and the celerity $c = \sqrt{gh}$. The Froude number determines the subcritical ($Fr < 1$) or supercritical ($Fr > 1$) character of the flow. Following Soares Frazão and Zech [19], the Roe numerical flux for SWEs can be written as a function of Fr as

$$\mathbf{f}(\mathbf{u}_e, \mathbf{u}_{e_k}) = \tfrac{1}{2}[(1 + Fr_A)\vec{\mathbf{F}}(\mathbf{u}_e) + (1 - Fr_A)\vec{\mathbf{F}}(\mathbf{u}_{e_k})] \cdot \boldsymbol{n} + \tfrac{1}{2} c_A (1 - Fr_A^2)[\mathbf{u}_e - \mathbf{u}_{e_k}] \tag{10}$$

where the subscript A denotes average quantities. In (10), the average Froude number

$$Fr_A = \frac{\boldsymbol{v}_A \cdot \boldsymbol{n}}{c_A}$$

is computed using Roe averages i.e.

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

$$\boldsymbol{v}_A = \{(v_x)_A, (v_y)_A\}^{\mathrm{T}} \quad \text{and} \quad c_A = \sqrt{g\tfrac{1}{2}(h_e + h_{e_k})}$$

In order to use (10) for both the sub- and supercritical case, the absolute value of the Froude number Fr is bounded by 1. We correct its value in order to fulfill this physical constraint: If $Fr > 1$, $Fr = 1$ and if $Fr < -1$, $Fr = -1$. The situation is similar in the case of the Euler equations, as noted in the original paper by Roe [16]. The flux is the sum of a high-order centred term plus a dissipation term of order zero:

$$\mathbf{f} = \underbrace{\tfrac{1}{2}[\vec{\mathbf{F}}(\mathbf{u}_e) + \vec{\mathbf{F}}(\mathbf{u}_{e_k})] \cdot \boldsymbol{n}}_{\text{Centred differences}} + \underbrace{\tfrac{1}{2}Fr[\vec{\mathbf{F}}(\mathbf{u}_e) - \vec{\mathbf{F}}(\mathbf{u}_{e_k})] \cdot \boldsymbol{n} + \tfrac{1}{2}c_A(1 - Fr^2)[\mathbf{u}_e - \mathbf{u}_{e_k}]}_{\text{Dissipation}}$$

It has been proven in Reference [20] that, in smooth regions, $\mathbf{u}_e - \mathbf{u}_{e_k} = \mathcal{O}(h^{k+1})$ 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 $\mathcal{M}(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega$. 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 $\boldsymbol{y}$ is the vector representing an edge with its Euclidian coordinates y_1, y_2, y_3, the optimum mesh is characterized by

$$\boldsymbol{y}^{\mathrm{T}}\mathcal{M}\boldsymbol{y} = 1 \quad \forall \boldsymbol{y}$$

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

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

with

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

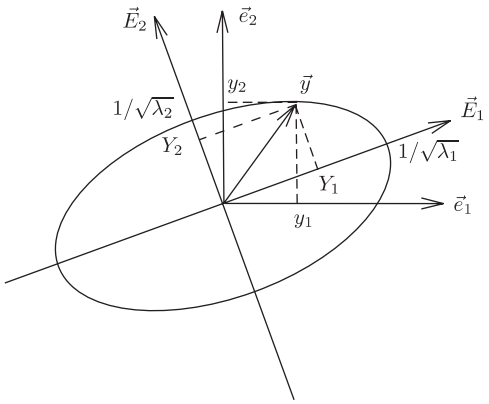

Figure 3. Interpretation in 2D of the different parameters in the definition of the metric $\mathscr{M}$.

and

$$\mathscr{R} = \begin{bmatrix} \boldsymbol{E}_1 \\ \boldsymbol{E}_2 \\ \boldsymbol{E}_3 \end{bmatrix}$$

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

$$\boldsymbol{y}^{\mathrm{T}} \mathscr{R}^{\mathrm{T}} \Lambda \mathscr{R} \boldsymbol{y} = 1 \quad \forall \boldsymbol{y}$$

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

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

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 its 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 metric are then directly related to desired mesh sizes in the principal directions of the metric $\mathscr{M}$.

3.2. *Computing mesh sizes*

Those three 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 [21]. The Hessian is symmetric and we have then the orthogonal decomposition

$$H = \mathscr{R}^{\mathrm{T}} \bar{\Lambda} \mathscr{R}$$

with

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

In terms of Taylor expansion theorem, the discretization error e in the direction of the ith eigenvector of H is proportional to $\bar{s_i}^2 \bar{\lambda_i}$, i.e. $e = c\bar{s_i}^2 \bar{\lambda_i}$, where $\bar{s_i}$ is the local mesh edge length in the ith direction for previous computation. We can compute a mesh with equidistributed error by choosing

$$s_i^2 = \frac{\varepsilon}{c\bar{s_i}^2 |\bar{\lambda_i}|}$$

where ε is a targeted error that is defined by the user, c is the coefficient of proportionality and s_i is desired local mesh edge length in the ith direction. It is usual to define maximal and minimal element sizes $s_{\max}$ and $s_{\min}$ that prevent to build up nonrealistic metric fields. The metric is then corrected by

$$\lambda_i = \max\left(\min\left(\frac{1}{s_i^2}, \frac{1}{s_{\min}^2}\right), \frac{1}{s_{\max}^2}\right)$$

In Reference [22], we have shown how to detect efficiently nonsmooth regions in DGM computations. Our discontinuity detector has the following form:

$$\mathscr{I}_e = \frac{\sum_{j=1}^{n_e} \left| \int_{\partial e_j} (h^e - h^{e_j})\, \mathrm{d}s \right|}{s_e^{(k+1)/2} |\partial e| \|h^e\|} \tag{11}$$

In all forthcoming 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.

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

$$\begin{aligned} &\text{if } \mathscr{I}_e > 1, \quad h \text{ is discontinuous} \\ &\text{if } \mathscr{I}_e < 1, \quad h \text{ is smooth} \end{aligned} \tag{12}$$

Near discontinuities, Hessians are highly ill conditioned. When 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 $s_{\max}$ on the other direction.

A metric smoother is crucial at this point in order to reconnect smooth and discontinuous regions. The algorithm is described in Reference [21].

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

$$\mathscr{T}^{n+1} = \mathscr{T}^n + \mathscr{C}' - \mathscr{C} \tag{13}$$

where $\mathscr{T}^n$ denotes the mesh before the local mesh modification and $\mathscr{T}^{n+1}$ 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 (13) as a two-step procedure:

$$\mathscr{T}' = \mathscr{T}^n + \mathscr{C}' \tag{14}$$

$$\mathscr{T}^{n+1} = \mathscr{T}' - \mathscr{C} \tag{15}$$

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

$$\mathscr{T}' = \mathscr{T}^n + \mathscr{C}' \tag{16}$$

$$\Pi_{\mathsf{L}^2}(\mathbf{u}, \mathscr{C}, \mathscr{C}') \tag{17}$$

$$\mathscr{T}^{n+1} = \mathscr{T}' - \mathscr{C} \tag{18}$$

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

In two dimensions, we only use the three mesh modification operators that are described below.

3.3.1. Edge splitting. The first mesh modification operator is the edge splitting. It consists in modifying a cavity $\mathscr{C}$ composed of two neighbouring elements e_1 end e_2 by splitting the common edge and replacing e_1 and e_2 by e_3, e_4, e_5 and e_6 as represented in Figure 4.

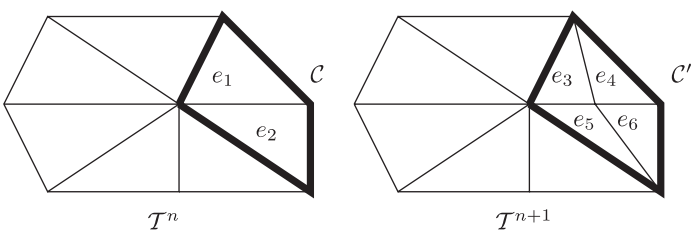

Figure 4. Edge splitting in 2D.

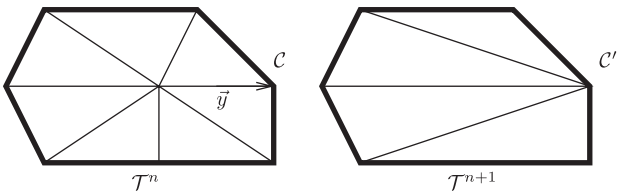

Figure 5. Edge collapsing in 2D.

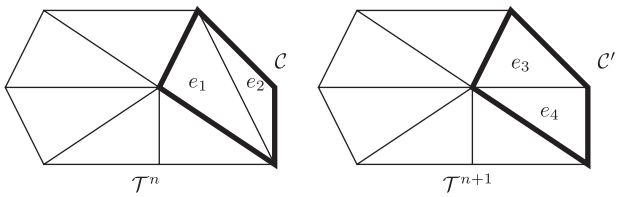

Figure 6. Edge swapping in 2D. An edge separating two triangles is replaced by the opposite edge.

3.3.2. Edge collapsing. The second mesh modification operator is the edge collapsing. An edge $\boldsymbol{y}$ is too short, so it is decided to remove it. The origin of the edge is moved to its end, modifying a cavity $\mathscr{C}$ composed of all the triangles neighbouring the origin of the edge (see Figure 5).

3.3.3. Edge swapping. The last mesh modification operator is the edge swapping. It consists in modifying a cavity $\mathscr{C}$ composed of two neighbouring 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 Figure 6.

The mesh adaptation algorithm for the three-dimensional case is described in Reference [23]. 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. The consequence of the numerous local mesh modifications on the quality of the numerical solution is of importance. 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 complexity.

While performing edge collapsing, some information is lost because the discrete space where the numerical solution lives is made smaller. Some numerical dissipation may then be introduced by the operation. The L^2 projection $\Pi_{\mathrm{L}^2}(\mathbf{u},\mathscr{C},\mathscr{C}')$ may also introduce Gibbs oscillations (overshoots) for polynomial order $k>0$. Because we use L^2 projections, the conservative quantities are transferred correctly from $\mathscr{C}$ to $\mathscr{C}'$. Note that, for the target cavity $\mathscr{C}'$, the solution over $\mathscr{C}$ is piecewise discontinuous and, consequently, we have to be careful and

compute the L^2 projection accurately. Despite that, we do not expect that edge collapsing's will cause any problem since they are performed in regions of low error only and, therefore, the resulting modifications of the solution due to the operation should not have any significant impact.

The edge swapping is certainly the most contradicting. Edge swappings are performed everywhere, including regions of high error. Similarly to edge collapsing, edge swapping is an operation that modifies the solution and therefore introduce errors. Conservation is not an issue because we do not consider curved geometries here. This issue will be considered in a forthcoming paper.

4. RESULTS

4.1. Radial dam break problem

Consider the SWEs with piecewise initial data

$$h(r) = \begin{cases} 2 & \text{if } r < 1 \\ 1 & \text{if } r > 1 \end{cases}$$

$r = \sqrt{x^2 + y^2}$ 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 a solution that consists in a rarefaction wave and a hydraulic jump (shock). The rarefaction wave moves in the direction of the highest water depth (i.e. radially, towards the origin) while the shock moves in 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 shock rapidly forms. This problem has an interesting structure and will be used to verify our refinement methodology:

- Examine the influence of multiple mesh refinement and coarsening steps on the preservation of 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 discretizations (i.e. verify that anisotropic refinement is cheaper than isotropic refinement or no refinement for a given level of solution resolution).

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

$$\begin{aligned} \frac{\partial h}{\partial t} + \frac{\partial}{\partial r}(hv_r) &= -\frac{hv_r}{r} \\ \frac{\partial v_r}{\partial t} + \frac{\partial}{\partial r}\left(hv_r^2 + \frac{1}{2}gh^2\right) &= -\frac{hv_r^2}{r} \end{aligned} \tag{19}$$

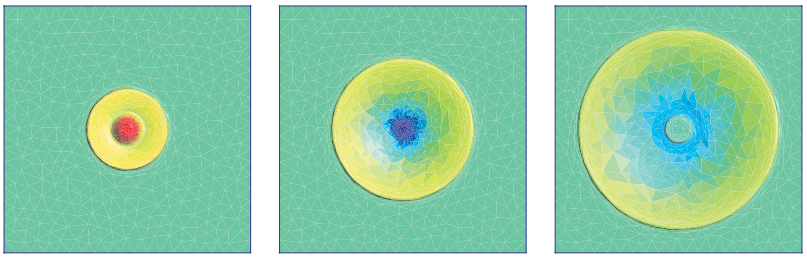

Figure 7. Plots of the water height at different time steps. Those are the solution of the anisotropic run: (a) $t = 0.5$; (b) $t = 1$; and (c) $t = 1.5$.

with a high resolution finite volume scheme on a very fine grid of 2000 points [24]. In what follows, we will call it the 'reference' solution.

We have done three computations of the problem. The first one has been done using a uniform mesh with element sizes of $\frac{1}{200\text{th}}$ 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 in each direction as a function of position and time. The mesh was adapted every 0.01 s. 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. $\frac{1}{200\text{th}}$ of the size of the problem. The maximum mesh size $s_{\max} = 5/20$ i.e. $\frac{1}{20\text{th}}$ of the size of the problem. The third computation was done using anisotropic refinement. Adaptation parameters limits were the same as for the isotropic refinement. The three problems have the same effective discretization with respect to the smallest mesh size at any location in any direction.

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

Figure 8 shows 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 Section 3.3 that the kind of projections used do not cause problems 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 (this is of course a weak justification that will be refined).

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

$$\varepsilon = \frac{\int_0^5 |h - h_{\text{ex}}| \, \mathrm{d}r}{\int_0^5 |h_{\text{ex}}| \, \mathrm{d}r}$$

for both uniform and adapted meshes (h_{ex} is the 'reference' solution computed with the fine 1D grid). We have plotted h and h_{ex} along 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 anisotropically (see Figure 10) adapted grid than in the uniform grid and about four times (see Figure 9) less elements than

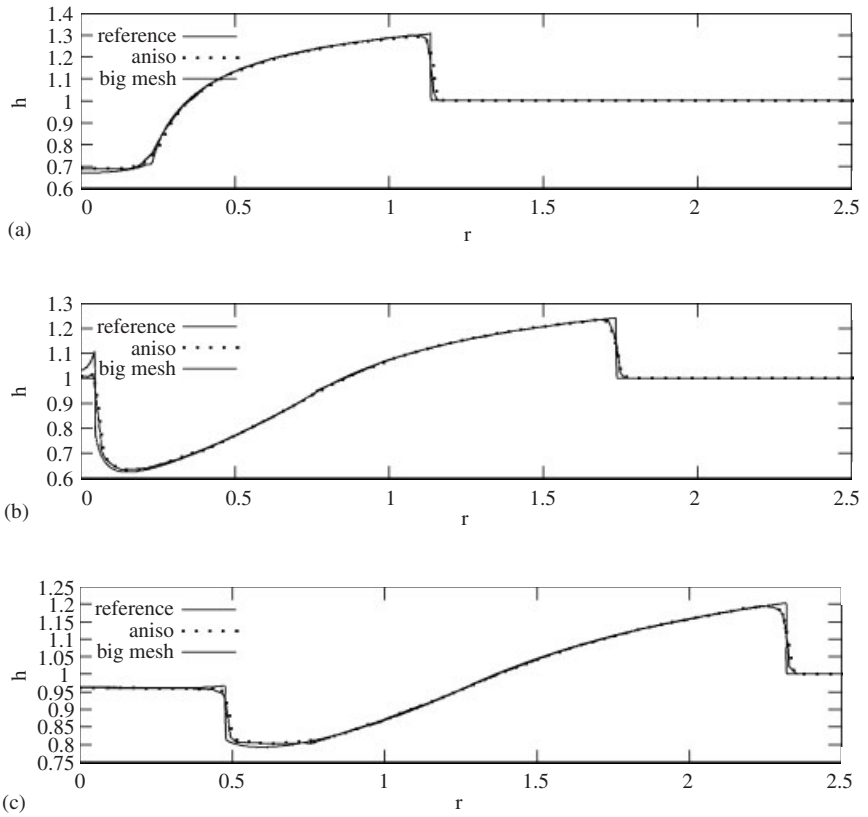

Figure 8. Comparison at different times between exact solution, solution on the uniform grid and solution on the anisotropically adapted grid: (a) $t = 0.5$; (b) $t = 1.0$; and (c) $t = 1.5$.

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

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

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

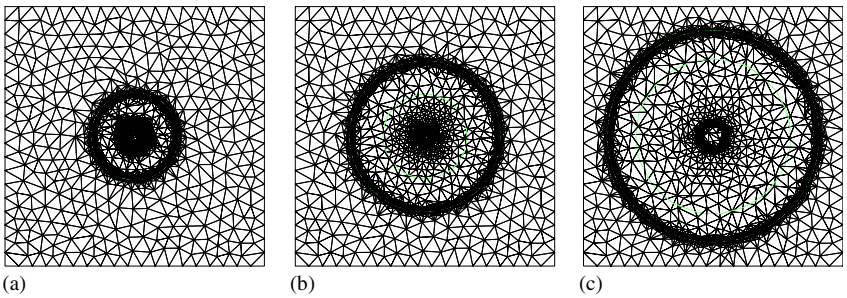

Figure 9. Adapted meshes for the isotropic refinement case: (a) $t = 0.5$, 5386 triangles; (b) $t = 1$, 7198 triangles; and (c) $t = 1.5$, 9510 triangles.

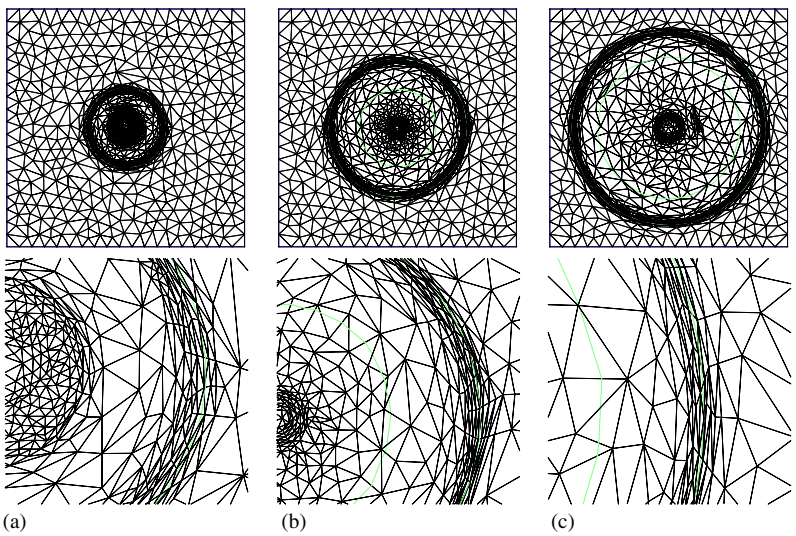

Figure 10. Adapted meshes for the anisotropic refinement case. The images on bottom show zooms of the mesh near the hydraulic jump: (a); $t = 0.5$, 2078 triangles; (b) $t = 1$, 1920 triangles; and (c) $t = 1.5$, 1932 triangles.

small so that elements with large aspect ratio are not constructed, 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 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 efficiency of the procedure grows. In fact, it looks 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 simulations, e.g. for Euler equations of gas dynamics. There, the number of

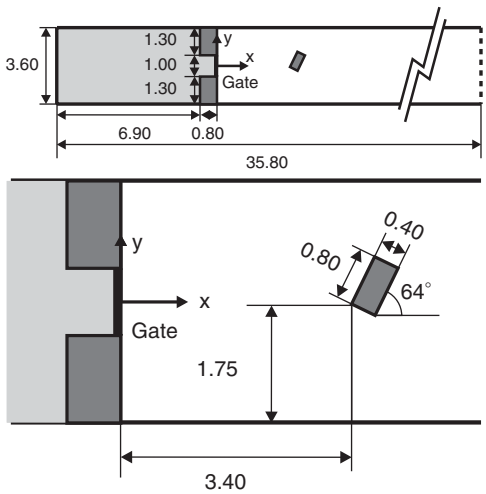

Figure 11. Experimental set-up for the dam break problem in presence of a building.

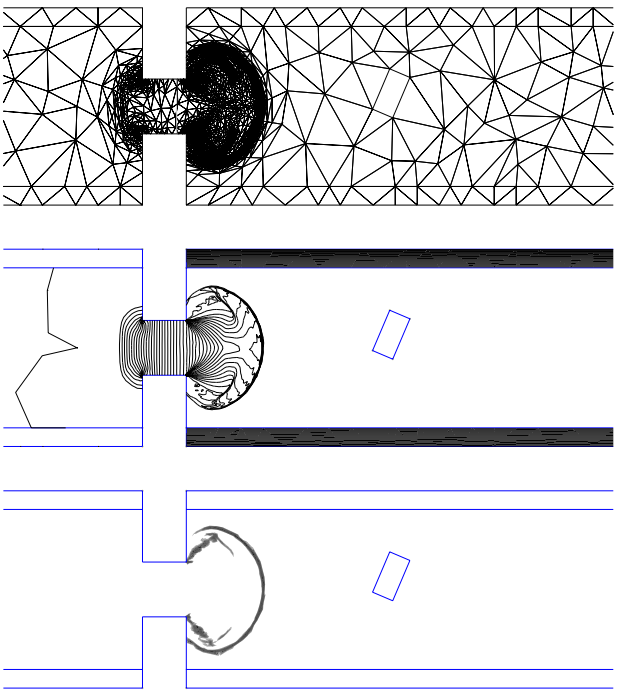

Figure 12. Adapted mesh (top), water level contours (middle) and discontinuity detector (bottom) at $t = 0.66$ s. At this time, the mesh is composed of 4413 triangles which correspond to 39 717 unknowns.

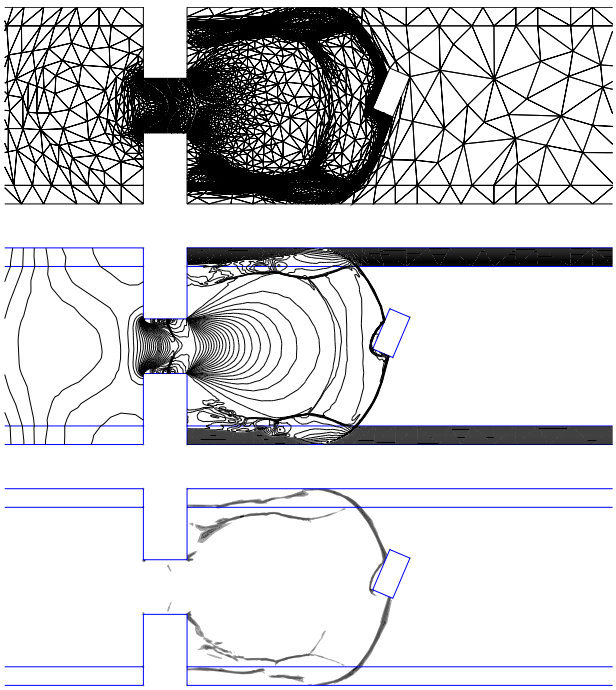

Figure 13. Adapted mesh (top), water level contours (middle) and discontinuity detector (bottom) at $t = 1.97$ s. At this time, the mesh is composed of 10 258 triangles which correspond to 92 322 unknowns.

elements in a shock will grow quadratically in the isotropic case while remaining constant in the anisotropic case.

4.2. Dam breaking in presence of a building

The second example is the one of a dam break wave against a rectangular building [25]. The experiments were performed at the Civil Engineering Department of the Université catholique de Louvain (Belgium), on the set-up sketched in Figure 11. The channel has a total length of 35.80 m and is 3.60 m wide. The cross-section near the bed is trapezoidal. The building is located 3.40 m downstream from the dam. The initial conditions consist in a 0.40 m water level in the upstream reservoir and a 0.02 m thin water layer in the downstream channel. The friction coefficient n of (9) was chosen as $n = 0.01$.

Two types of flow measurements were performed: Water level and velocity at 5 gauging points, and field data using a Voronoï imaging technique [26, 27]. This latter technique, initially developed for dense granular flow measurements, was used to obtain the experimental pictures that are compared to the computations in some of the following figures. A complete description of the test case and of the available experimental data can be found in Reference [25].

The only computation that we have done for this problem was anisotropically adapted. Parameters of the simulation were $s_{\min} = 0.01$ m and $\varepsilon = 0.05$. For sake of comparison, we

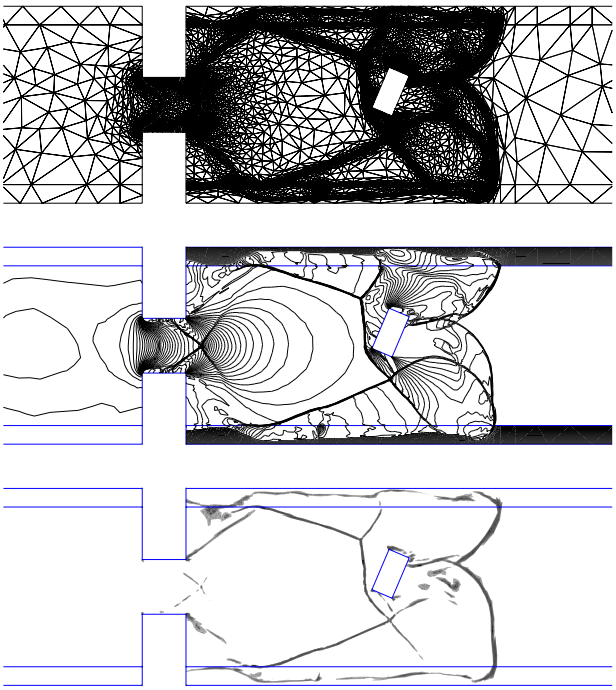

Figure 14. Adapted mesh (top), water level contours (middle) and discontinuity detector (bottom) at $t = 3.42$ s. At this time, the mesh is composed of 15 606 triangles which correspond to 140 454 unknowns.

have generated a uniform triangular mesh with a uniform mesh size of 0.01 m. This 'equivalent' mesh was containing 2 968 944 triangles (i.e. about three million elements and about 27 million unknowns!). In our anisotropic simulation, the initial mesh was containing about 2000 triangles. After 10 s of computation, this number was about 16 000 (i.e. 2 orders of magnitude less than the equivalent uniform mesh). Figures 12–14 show adapted meshes, water level contours and discontinuity indicator at different times. The indicator $\mathscr{I}_e$ of Equation (11) was only shown when its value was greater than 1. The threshold value $\mathscr{I}_e = 1$ was convenient to detect the principal hydraulic jumps of the flow. Note that the indicator did not detect the slope discontinuity of the water bed at $y = \pm 1.8$ m. On the other side, Figure 14 show that, even if the indicator $\mathscr{I}_e$ did not predict any discontinuity along this line, the second-order derivative of h being large there, the mesh is refined along that line because of the large value of $\partial^2 h/\partial y^2$. It is an example among many others that show that the use of second-order derivatives may be an incorrect strategy for predicting the discretization error.

Another useful application of the discontinuity indicator $\mathscr{I}_e$ is related to limiting. With limiting only used near discontinuities, i.e. when $\mathscr{I}_e > 1$, we need not be concerned with maintaining a high order of accuracy; thus, we focus on the slope limiting procedure introduced by Cockburn and Shu [28]. Slope limiting compares solution gradients on e with average solution gradients on neighbouring elements e_k. The computed and average gradients are

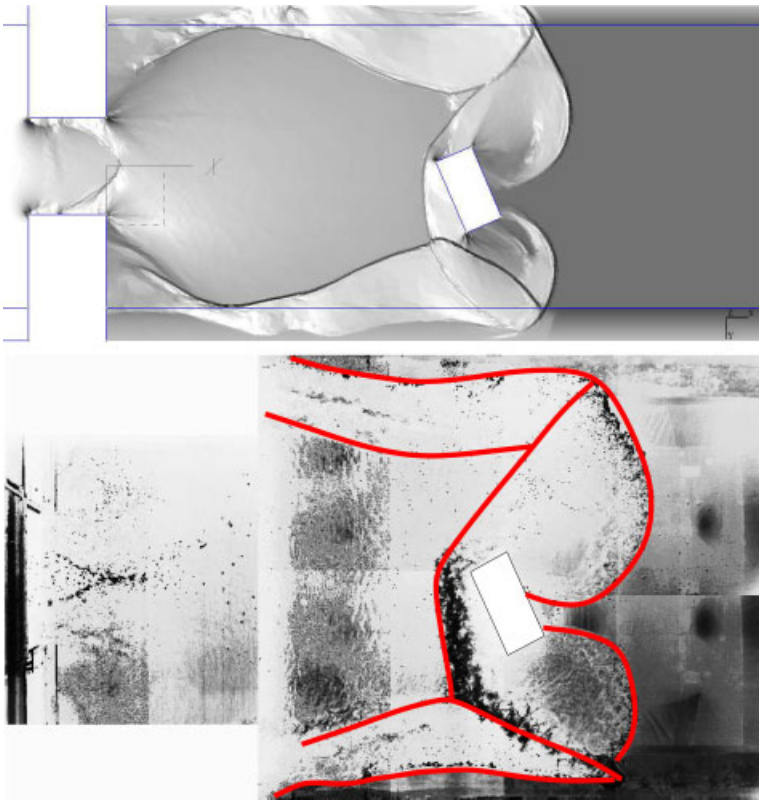

Figure 15. Water level at $t = 2.63$ s. for the isolated building problem. On top, the anisotropic computation and on bottom, the experiment. Thick lines were drawn on top of the picture in order to enhance the hydraulic jumps.

compared and elemental slopes are restricted to the range spanned by neighbouring averages when all have the same sign. Slopes are set to zero should signs disagree [28–30].

The adaptive mesh procedure was able to capture the complex features of the flow with a high level of accuracy. The comparison with our experiment (Figure 15) shows that the adaptive DGM was able to predict complex wave interactions with accuracy.

5. CONCLUSIONS

In this paper, we have shown the application of an adaptive meshing procedure on transient computation of shallow-water equations. The adaptive procedures were able to effectively predict the wave motions and locations through a large number of mesh adaption steps. The mesh adaptations did effectively control the discretization errors and were executed such that they did not introduce excessive numerical dissipation during the required projection operations.

The ability to perform a large number of adaptation (125 mesh adaptations for the radial dam break and more than 500 for the isolated building case) without degradating the solution is principally due to the underlying numerical schemes. The use of the discontinuous Galerkin method with orthogonal basis and controlled local mesh modification operations allows conservative and high-order mesh-to-mesh projections, avoiding there the usual drawbacks of multiple adaptations.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the financial support offered by the European commission for the IMPACT project under the fifth framework programme (1998–2002), environment and sustainable development thematic programme, for which Karen Fabbri was the EC project officer. The experimental data presented in this paper were produced within WP3 (Flood Propagation) of the project. The overall contribution made by the IMPACT project team is also recognized.

The authors also wish to thank Benoît Spinewine for making the pictures of the experimental flow available.

REFERENCES

1. Reed WH, Hill TR. Triangular mesh methods for the neutron transport equation. *Technical Report LA-UR-73-479*, Los Alamos Scientific Laboratory, 1973.
2. Cockburn B, Karniadakis G, Shu C-W (eds). *Discontinuous Galerkin Methods*, Lecture Notes in Computational Science and Engineering, vol. 11. Springer: Berlin, 2000.
3. Schwanenberg D, Kongeter J. A discontinuous Galerkin method for the shallow water equations with source terms. In *Discontinuous Galerkin Methods*, Cockburn B, Karniadakis G, Shu C-W (eds), Lecture Notes in Computational Science and Engineering, vol. 11. Springer: Berlin, 2000; 419–424.
4. Harten A, Lax PD, van Leer B. On upstream differencing and godunov-type schemes for hyperbolic conservation laws. *SIAM Review* 1983; **25**(1):35–61.
5. Giraldo F, Hesthaven J, Wartburton T. Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations. *Journal of Computational Physics* 2002; **181**:499–525.
6. Aizinger V, Dawson C. A discontinuous Galerkin method for two-dimensional flow and transport in shallow water. *Advances in Water Resources* 2002; **25**(1):67–84.
7. Cockburn B, Hou S, Shu C. The Runge–Kutta local projection discontinuous Galerkin finite element method for the conservation laws IV: the multidimensional case. *Mathematics of Computations* 1990; **54**:545–581.
8. Furtado F, Glimm J, Grove J. *Front Tracking and the Interaction of Nonlinear Hyperbolic Waves*, Li X-L, Lindquist WB, Menikoff R, Sharp DH, Zhang Q (eds). Lecture Notes in Engineering, 1989; **43**:99–111.
9. Remacle J-F, Flaherty JE, Shephard MS. An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems. *SIAM Review* 2003; **45**(1):53–72.
10. Dubiner M. Spectral methods on triangles and other domains. *Journal of Scientific Computation* 1991; **6**:345–390.
11. Yen BC. Dimensionally homogeneous manning's formula. *Journal of Hydraulic Engineering* 1992; **118**(9):1326–1332.
12. Toro EF. *Riemann Solvers and Numerical Methods for Fluid Dynamics. A Practical Introduction*. Springer: Berlin, Heidelberg, 1999.
13. LeVeque R. *Numerical Methods for Conservation Laws*. Birkhäuser: Basel, 1992.
14. Godunov SK. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Mathematics of the Sbornik* 1959; **47**:207–306.
15. Woodward P, Colella P. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics* 1984; **54**:115–173.
16. Roe PL. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics* 1981; **43**:357–372.
17. Glaister P. Approximate Riemann solutions of the shallow-water equations. *Journal of Hydraulic Research* 1988; **3**(26):293–306.
18. Alcrudo F, Garcia-Navarro P, Saviron JM. Flux difference splitting for 1d open channel flow equations. *International Journal for Numerical Methods in Fluids* 1992; **14**:1009–1018.
19. Soares Frazão S, Zech Y. Dam-break in channels with 90° bend. *Journal of Hydraulic Engineering*, *American Society of Civil Engineers* (ASCE) 2002; **128**(11):956–968.

20. Adjerid S, Devine KD, Flaherty JE, Krivodonova L. A posteriori error estimation for discontinuous Galerkin solutions of hyperbolic problems. *Computer Methods in Applied Mechanics and Engineering* 2002; **191**: 1097–1112.
21. Remacle J-F, Shephard MS. An algorithm oriented mesh database. *International Journal for Numerical Methods in Engineering* 2003; **58**(2):349–374.
22. Krivodonovna L, Xin J, Remacle J-F, Chevaugeon N, Flaherty JE. Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws. *Applied Numerical Mathematics* 2004; **48**(3–4):323–338.
23. Li X. Mesh Modification procedures for general 3-D non-manifold domains. *Ph.D. Thesis*, Rensselear Polytechnic Institute, August 2003.
24. LeVeque R. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press: Cambridge, 2002.
25. Soares Frazão S, Zech Y. Experimental study of dam-break flow against an isolated obstacle. *Journal of Hydraulic Research*, submitted.
26. Capart H, Young DL, Zech Y. Voronoï imaging methods for the measurements of granular flows. *Experiments in Fluids* 2002; **32**:121–135.
27. Spinewine B, Capart H, Larcher M, Zech Y. Three-dimensional voronoï imaging methods for the measurement of near-wall particulate flows. *Experiments in Fluids* 2003; **34**:227–241.
28. Cockburn B, Shu C-W. TVB Runge–Kutta local projection discontinuous Galerkin methods for scalar conservation laws II: general framework. *Mathematics of Computation* 1989; **52**:411–435.
29. van Leer B. Towards the ultimate conservation difference scheme, II. *Journal of Computational Physics* 1974; **14**:361–367.
30. van Leer B. Towards the ultimate conservation difference scheme, V. *Journal of Computational Physics* 1979; **30**:1–136.

