INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2006; **65**:295–331
Published online 6 October 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.1442

High-order finite volume schemes on unstructured grids using moving least-squares reconstruction. Application to shallow water dynamics

L. Cueto-Felgueroso, I. Colominas*, †, J. Fe, F. Navarrina and M. Casteleiro

Group of Numerical Methods in Engineering, GMNI, Department of Applied Mathematics, Civil Engineering School, Universidad de La Coruña, Campus de Elviña, 15071 La Coruña, Spain

SUMMARY

This paper introduces the use of moving least-squares (MLS) approximations for the development of high-order finite volume discretizations on unstructured grids. The field variables and their successive derivatives can be accurately reconstructed using this mesh-free technique in a general nodal arrangement. The methodology proposed is used in the construction of two numerical schemes for the shallow water equations on unstructured grids: a centred Lax–Wendroff method with added shock-capturing dissipation, and a Godunov-type upwind scheme, with linear and quadratic reconstructions. This class of mesh-free techniques provides a robust and general approximation framework which represents an interesting alternative to the existing procedures, allowing, in addition, an accurate computation of the viscous fluxes. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: shallow water dynamics; finite volume method; high-resolution schemes; high-order methods; moving least-squares approximation; unstructured grids

1. INTRODUCTION

The development of a general algorithm capable of achieving optimal performance in all flow problems is one of the most important and challenging areas of research in computational

*Correspondence to: I. Colominas, E.T.S. de Ingenieros de Caminos, Canales y Puertos, Universidad de La Coruña, Campus de Elviña, 15071 La Coruña, Spain.
†E-mail: icolominas@udc.es

Contract/grant sponsor: Ministerio de Educación y Ciencia; contract/grant number: DPI#2004-05156
Contract/grant sponsor: FEDER; contract/grant numbers: PGDIT01PXI11802PR, PGIDIT03PXIC118002PN
Contract/grant sponsor: University of La Coruña
Contract/grant sponsor: Fundación de la Ingeniería Civil de Galicia
Contract/grant sponsor: Colegio de Ingenieros de Caminos, Canales y Puertos
Contract/grant sponsor: Caixanova

Received 15 October 2004
Revised 18 March 2005
Accepted 8 June 2005

mechanics. In the context of shallow water dynamics, finite element and finite volume discretizations have become very popular in recent years on unstructured grids.

Continuous finite element formulations for fluid dynamics are usually elegant and applicable to a wide variety of flow conditions. Unfortunately, their frequent centred character hinders their suitability for problems involving shock waves and transcritical flow, thus requiring the development and tuning of more or less effective artificial viscosity models. One of the most successful of these finite element schemes applied to the shallow water equations is the Taylor–Galerkin FEM algorithm proposed by Peraire [1, 2], which has been further developed by Quecedo and Pastor [3, 4]. Sheu and Fang [5] have recently proposed a generalized Taylor–Galerkin finite element method to obtain high resolution of discontinuous flows.

Most finite volume formulations for the set of shallow water equations are reflections of high-resolution schemes originally devised to solve high-speed compressible flows, and have been successfully employed in the simulation of flows including the presence of shock waves, such as breaking dams or hydraulic jumps, almost invariably neglecting viscous and turbulent effects. Alcrudo and García-Navarro [6] developed a Godunov-type MUSCL high-resolution scheme based on Roe's Riemann solver. A similar formulation was used by Chippada *et al.* [7], whereas Zhao *et al.* [8] construct their numerical flux using Osher's method. Anastasiou and Chan [9] solved the full set of shallow water equations on unstructured meshes using a second-order Roe scheme and reported results for viscous flows at low Reynolds numbers. Other upwind schemes with shock-capturing capabilities have been proposed by Hu and Mingham [10] and Tseng [11]. Liszka and Wendroff [12] introduced composite methods, which combine Lax–Wendroff and Lax–Friedrichs schemes into a multistage algorithm, and Wang and Liu [13] have recently extended the methodology to unstructured triangular meshes. The concept of *high-order method* is most frequently used in the literature in reference to formally *second-order* methods (i.e. *linear reconstruction* in those schemes based on the generalized Godunov method).

The development of higher- (than second-) order methods is of great importance in practice. Various researchers have reported that first- and even second-order upwind schemes, despite providing excellent results in the case of discontinuous flows, exhibit excessive numerical dissipation when applied to more general flows (not necessarily including shock wave propagation) where turbulent effects are of interest [14–16], and several corrections to the original algorithms have been proposed in order to reduce the unnecessary artificial dissipation introduced in the computations. Unfortunately, these corrections are somewhat 'heuristic', and yet remains a compromise between accuracy and stability: the lesser the dissipation added the more accurate the results, whereas some amount of artificial viscosity is unavoidably necessary to yield stable algorithms. A suitable numerical method to solve such problems on unstructured grids should therefore not introduce excessive numerical dissipation, in order to capture fine viscous features of the flow and to avoid interactions with the turbulence model. Furthermore, when shock wave–turbulence interactions are present in the flow, the numerical method should possess the low dissipation of high-order methods and the shock-capturing capabilities of Godunov-type schemes [17].

The endeavour to solve increasingly complex flows has promoted the advent of unstructured grids as the most efficient approach to discretize highly irregular domains, perform adaptive refinements and capture small-scale features of the flow. As far as the development of high-order Godunov-type [18–20] finite volume schemes for unstructured meshes is concerned, the absence of an underlying spatial approximation framework, which stems from the inherent piecewise

constant representation, is certainly a most challenging algorithmic issue, since the construction of high-order polynomials for reconstruction requires the evaluation of high-order derivatives of the field variables from scattered, pointwise information. As a result, most schemes are at best second-order and even the required reconstruction of fluxes and gradients is addressed by using somewhat 'heuristic' techniques, which frequently lead to quite complex data processing when proper accuracy and low grid sensitivity are pursued.

In this paper, the authors would like to propose a mesh-free technique, the so-called moving least-squares (MLS) approximation, as an accurate and efficient technique to obtain high-order finite volume algorithms on unstructured grids. This class of approximation methods is particularly well suited for such purpose, providing a robust and general approximation framework which represents an interesting alternative to the existing techniques, and allowing, in addition, an accurate computation of the viscous fluxes. Originally devised for data processing and surface generation [21], the MLS approximation has become very popular among those researchers working in the class of the so-called *meshless* or *mesh-free* methods, being widely used both in eulerian and lagrangian formulations. In particular, the authors have recently proposed an algorithm for lagrangian particle hydrodynamics, where the MLS technique plays a key role to provide the spatial approximation within an arbitrary cloud of nodes [22].

The spatial approximation framework provided by the MLS approximants will be used as the basis to construct two different finite volume formulations for the set of shallow water equations. The first of them corresponds to a Lax–Wendroff-type centred scheme with added artificial dissipation. The spatial finite volume discretization uses the MLS approximation as a kind of 'shape functions' for unstructured grids. This class of centred schemes requires the introduction of an artificial dissipation model, in order to deal with shocks and/or steep gradients in the flow variables. The resulting scheme possesses accuracy and stability properties very similar to its finite element counterpart, the Taylor–Galerkin FEM.

The second formulation follows the ideas of the generalized Godunov method [18, 19, 23], with piecewise polynomial reconstructions inside each cell. Both the computation of the successive derivates of the flow variables, and the evaluation of the diffusive fluxes, will be addressed by means of MLS approximations. Second and third-order-reconstruction upwind schemes, based on Roe's approximate Riemann solver [24], will be presented and tested for inviscid and viscous flow applications. In the latter, the approximation framework provided by these mesh-free techniques is specially interesting in the accurate evaluation of the viscous fluxes at the cell edges.

The outline of the paper is as follows. Section 2 presents a brief introduction to some meshless approximation techniques, with special emphasis on MLS and reproducing kernel methods. The model equations and numerical formulations are discussed in Section 3. Finally, Section 4 is devoted to present and discuss several numerical examples.

2. MESHLESS APPROXIMATION: MOVING LEAST-SQUARES

2.1. The idea of a mesh-free interpolation

The endeavour to solve the continuum equations in a *particle* (as opposed to *cell* or *element*) framework, i.e. simply using the information stored at certain *nodes* or *particles* without reference to any underlying mesh, has given rise to a very active area of research: the class of so-called meshless, mesh-free or particle methods.

If this particle approach is to be used in combination with classical discretization procedures (e.g. the weighted residuals method), then a spatial approximation is required (some kind of 'shape functions', as in the finite element method). Such an interpolation scheme should accurately *reproduce* or *reconstruct* a certain function and its successive derivatives, using the *nodal* (particle) *values* and some 'low-level' geometrical information about the grid, such as the distance between particles. Furthermore, and in order to achieve some algorithms computationally efficient, the interpolation should have a *local* character, i.e. the reconstruction process should involve only a few 'neighbour' nodes.

Even though a 'perfect' meshless approximation scheme, capable of achieving high accuracy for any randomly distributed set of points, is still not available, several powerful interpolation techniques have been recently proposed, thus enabling the development of increasingly efficient and accurate meshless formulations. What follows is a brief introduction to a certain class of such interpolation schemes, namely those based on reproducing kernel and MLS approximations. Further emphasis is placed on the particular technique used in this study, although the reader is referred to the classical mesh-free literature to find in-depth descriptions of these algorithms.

2.2. Meshless approximants

The origin of modern meshless methods could be dated back to the 1970s with the pioneering works in generalized finite differences and vortex particle methods [25, 26]. However, the strongest influence upon the present trends is commonly attributed to early smoothed particle hydrodynamics (SPH) formulations [27–29], where a lagrangian particle tracking is used to describe the motion of a fluid. Although this general feature is shared with vortex particle methods, SPH includes a spatial approximation framework (some kind of 'mesh-free shape functions'), developed using the concept of *kernel estimate*, which is inspired by the following property of the Dirac δ function

$$u(\mathbf{x}) = \int_{\mathbf{y}\in\Omega} u(\mathbf{y})\delta(\mathbf{x}-\mathbf{y})\,\mathrm{d}\Omega \tag{1}$$

The kernel estimate $\langle u(\mathbf{x})\rangle$ of a given function $u(\mathbf{x})$ is defined as

$$\langle u(\mathbf{x})\rangle = \int_{\mathbf{y}\in\Omega} u(\mathbf{y})W(\mathbf{x}-\mathbf{y},\rho)\,\mathrm{d}\Omega \tag{2}$$

and its discrete SPH counterpart $\hat{u}(\mathbf{x})$ is

$$\hat{u}(\mathbf{x}) = \sum_{j=1}^{n} u_j W(\mathbf{x}-\mathbf{x}_j,\rho)V_j \tag{3}$$

where Ω is the problem domain, which is assumed to be represented by a set of n *nodes* or *particles* (used as quadrature points in (2)), $W(\mathbf{x}-\mathbf{x}_j,\rho)$ is a *kernel* (smoothing) function with compact support centred at particle j, and V_j is the tributary or statistical 'volume' associated to particle j. The parameter ρ, usually called *smoothing length* or *dilation parameter*, is a certain characteristic measure of the size of the support of W_j (e.g. kernels with circular supports of radius 2ρ). Exponential and spline functions are most frequent kernels. In analogy with the finite

element method, the approximation (3) could be cast in terms of 'SPH shape functions', as

$$\hat{u}(\mathbf{x}) = \sum_{j=1}^{n} u_j N_j(\mathbf{x}), \quad N_j(\mathbf{x}) = W(\mathbf{x} - \mathbf{x}_j, \rho) V_j \tag{4}$$

Using standard kernels, the approximation given by (4) is poor near boundaries, and lacks even zeroth order completeness, i.e.

$$\sum_{j=1}^{n} N_j(\mathbf{x}) \neq 1 \tag{5}$$

The gradient of $\hat{u}(\mathbf{x})$ is evaluated as

$$\nabla_{\mathbf{x}} \hat{u}(\mathbf{x}) = \sum_{j=1}^{n} u_j \nabla_{\mathbf{x}} N_j(\mathbf{x}) = \sum_{j=1}^{n} u_j \nabla_{\mathbf{x}} W_j(\mathbf{x}) V_j \tag{6}$$

In practice, alternative expressions for $\nabla_{\mathbf{x}} \hat{u}(\mathbf{x})$ are frequent in the SPH literature to enforce conservation properties in the discrete equations. Higher-order derivatives could be computed in a similar way. Note that the reconstructed values of $u(\mathbf{x})$ and its derivatives at a certain location are obtained using the information from neighbouring nodes and weightings that are functions of distances between nodes, with no reference to any grid-based data structure (Figure 1).

This basic approximation structure is retained in other improved interpolation schemes. In this study only MLS and reproducing kernel particle methods (RKPM) are considered. Although different in their formulation, the resulting numerics are almost identical for both methods, and they can be presented within a common framework.

Let us consider a function $u(\mathbf{x})$ defined in a domain Ω. The basic idea of the MLS approach is to approximate $u(\mathbf{x})$, at a given point $\mathbf{x}$, through a weighted least-squares fitting of $u(\mathbf{x})$ in a neighbourhood of $\mathbf{x}$ as

$$u(\mathbf{x}) \approx \hat{u}(\mathbf{x}) = \sum_{i=1}^{m} p_i(\mathbf{x}) \alpha_i(\mathbf{z}) \Big|_{\mathbf{z}=\mathbf{x}} = \mathbf{p}^{\mathrm{T}}(\mathbf{x}) \boldsymbol{\alpha}(\mathbf{z}) \big|_{\mathbf{z}=\mathbf{x}} \tag{7}$$

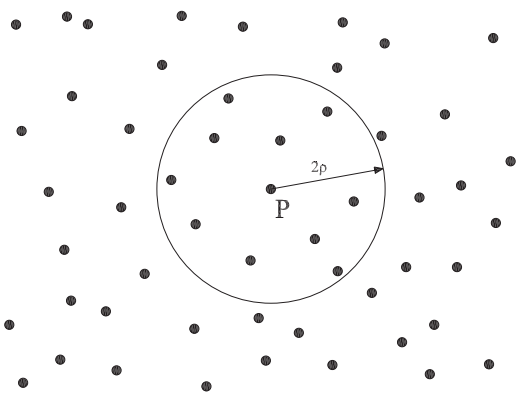

Figure 1. Mesh-free approximation: general scheme. Support for reconstruction at P.

where $\mathbf{p}^{\mathrm{T}}(\mathbf{x})$ is an m-dimensional polynomial basis and $\boldsymbol{\alpha}(\mathbf{z})|_{\mathbf{z}=\mathbf{x}}$ is a set of parameters to be determined, such that they minimize the following error functional

$$J(\boldsymbol{\alpha}(\mathbf{z})|_{\mathbf{z}=\mathbf{x}}) = \int_{\mathbf{y}\in\Omega_{\mathbf{x}}} W(\mathbf{z}-\mathbf{y},\rho)\Big|_{\mathbf{z}=\mathbf{x}} \left[u(\mathbf{y}) - \mathbf{p}^{\mathrm{T}}(\mathbf{y})\boldsymbol{\alpha}(\mathbf{z})\Big|_{\mathbf{z}=\mathbf{x}} \right]^2 \mathrm{d}\Omega_{\mathbf{x}} \tag{8}$$

being $W(\mathbf{z}-\mathbf{y},\rho)|_{\mathbf{z}=\mathbf{x}}$ a kernel with compact support (denoted by $\Omega_{\mathbf{x}}$) centred at $\mathbf{z}=\mathbf{x}$, frequently chosen among the kernels used in standard SPH. As mentioned before, ρ is the smoothing length, which measures the size of $\Omega_{\mathbf{x}}$. The stationary conditions of J with respect to $\boldsymbol{\alpha}$ lead to

$$\int_{\mathbf{y}\in\Omega_{\mathbf{x}}} \mathbf{p}(\mathbf{y}) W(\mathbf{z}-\mathbf{y},\rho)\Big|_{\mathbf{z}=\mathbf{x}} u(\mathbf{y})\,\mathrm{d}\Omega_{\mathbf{x}} = \mathbf{M}(\mathbf{x})\boldsymbol{\alpha}(\mathbf{z})|_{\mathbf{z}=\mathbf{x}} \tag{9}$$

where the moment matrix $\mathbf{M}(\mathbf{x})$ is

$$\mathbf{M}(\mathbf{x}) = \int_{\mathbf{y}\in\Omega_{\mathbf{x}}} \mathbf{p}(\mathbf{y}) W(\mathbf{z}-\mathbf{y},\rho)\Big|_{\mathbf{z}=\mathbf{x}} \mathbf{p}^{\mathrm{T}}(\mathbf{y})\,\mathrm{d}\Omega_{\mathbf{x}} \tag{10}$$

In numerical computations, the global domain Ω is represented by a set of n particles. We can then evaluate the integrals in (9) and (10) using those particles inside $\Omega_{\mathbf{x}}$ as quadrature points (nodal integration) to obtain, after rearranging,

$$\boldsymbol{\alpha}(\mathbf{z})|_{\mathbf{z}=\mathbf{x}} = \mathbf{M}^{-1}(\mathbf{x})\mathbf{P}_{\Omega_{\mathbf{x}}}\mathbf{W}_V(\mathbf{x})\mathbf{u}_{\Omega_{\mathbf{x}}} \tag{11}$$

where the vector $\mathbf{u}_{\Omega_{\mathbf{x}}}$ contains the pointwise *nodal* values of $u(\mathbf{x})$ at the $n_{\mathbf{x}}$ nodes within $\Omega_{\mathbf{x}}$ (Figure 1)

$$\mathbf{u}_{\Omega_{\mathbf{x}}} = (u(\mathbf{x}_1)\ u(\mathbf{x}_2)\ \cdots\ u(\mathbf{x}_{n_{\mathbf{x}}}))^{\mathrm{T}} \tag{12}$$

The discrete version of the moment matrix is $\mathbf{M}(\mathbf{x}) = \mathbf{P}_{\Omega_{\mathbf{x}}}\mathbf{W}_{\mathbf{V}}(\mathbf{x})\mathbf{P}^{\mathrm{T}}_{\Omega_{\mathbf{x}}}$, where matrices $\mathbf{P}_{\Omega_{\mathbf{x}}}$ and $\mathbf{W}_{\mathbf{V}}(\mathbf{x})$, whose dimensions are, respectively, $(m \times n_{\mathbf{x}})$ and $(n_{\mathbf{x}} \times n_{\mathbf{x}})$, can be obtained as

$$\mathbf{P}_{\Omega_{\mathbf{x}}} = (\mathbf{p}(\mathbf{x}_1)\ \mathbf{p}(\mathbf{x}_2)\ \cdots\ \mathbf{p}(\mathbf{x}_{n_{\mathbf{x}}})) \tag{13}$$

$$\mathbf{W}_{\mathbf{V}}(\mathbf{x}) = \mathrm{diag}\{W_i(\mathbf{x}-\mathbf{x}_i)V_i\}, \quad i=1,\ldots,n_{\mathbf{x}} \tag{14}$$

Complete details can be found in Referred to [30–32]. In the above equations, V_i and $\mathbf{x}_i$ are, respectively, the tributary volume (used as quadrature weight) and co-ordinates associated to particle i. Note that the tributary volumes of neighbouring particles are included in matrix $\mathbf{W}_{\mathbf{V}}$, thus obtaining an MLS version of the RKPM (the so-called MLSRKPM) [30]. Otherwise, we can use $\mathbf{W}$ instead of $\mathbf{W}_{\mathbf{V}}$

$$\mathbf{W}(\mathbf{x}) = \mathrm{diag}\{W_i(\mathbf{x}-\mathbf{x}_i)\}, \quad i=1,\ldots,n_{\mathbf{x}} \tag{15}$$

which corresponds to the classical MLS approximation (in the nodal integration of the functional (8), the same quadrature weight is associated to all particles). Introducing (11) into (7) the interpolation structure can be identified as

$$\hat{u}(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{M}^{-1}(\mathbf{x})\mathbf{P}_{\Omega_{\mathbf{x}}}\mathbf{W}(\mathbf{x})\mathbf{u}_{\Omega_{\mathbf{x}}} = \mathbf{N}^{\mathrm{T}}(\mathbf{x})\mathbf{u}_{\Omega_{\mathbf{x}}} = \sum_{j=1}^{n_{\mathbf{x}}} N_j(\mathbf{x})u_j \tag{16}$$

and, therefore, the MLS shape functions can be written as

$$\mathbf{N}^{\mathrm{T}}(\mathbf{x}) = \mathbf{p}^{\mathrm{T}}(\mathbf{x})\mathbf{M}^{-1}(\mathbf{x})\mathbf{P}_{\Omega_{\mathbf{x}}}\mathbf{W}_V(\mathbf{x}) \tag{17}$$

The functional basis $\mathbf{p}(\mathbf{x})$ is strongly related to the accuracy of the MLS fitting. Numerical tests [32] have shown that, for a rth order MLS fitting (rth order complete polynomial basis) and general irregularly spaced points, the nominal order of accuracy for the approximation of a sth order gradient is roughly $(r-s+1)$. In general, any linear combination of the functions included in the basis is exactly reproduced by the MLS approximation.

In this study, the following cubic polynomial basis was used

$$\mathbf{p}(\mathbf{x}) = (1 \;\; x_1 \;\; x_2 \;\; x_1x_2 \;\; x_1^2 \;\; x_2^2 \;\; x_1^2x_2 \;\; x_1x_2^2 \;\; x_1^3 \;\; x_2^3)^{\mathrm{T}} \tag{18}$$

which provides cubic completeness. In the above expressions, (x_1, x_2) denotes the Cartesian co-ordinates of $\mathbf{x}$. To improve the conditioning of the moment matrix, it is most frequent to use *scaled* and *locally defined* monomials in the basis. Thus, if the shape functions were to be evaluated at a certain point $\mathbf{x}_I$, the basis would be of the form $\mathbf{p}((\mathbf{x}-\mathbf{x}_I)/\rho)$, instead of $\mathbf{p}(\mathbf{x})$. With this transformation, the MLS shape functions read

$$\mathbf{N}^{\mathrm{T}}(\mathbf{x}_I) = \mathbf{p}^{\mathrm{T}}(\mathbf{0})\mathbf{C}(\mathbf{x}_I) = \mathbf{p}^{\mathrm{T}}(\mathbf{0})\mathbf{M}^{-1}(\mathbf{x}_I)\mathbf{P}_{\Omega_{\mathbf{x}_I}}\mathbf{W}(\mathbf{x}_I) \tag{19}$$

where $\mathbf{C}(\mathbf{x})$ was defined as

$$\mathbf{C}(\mathbf{x}) = \mathbf{M}^{-1}(\mathbf{x})\mathbf{P}_{\Omega_{\mathbf{x}}}\mathbf{W}(\mathbf{x}) \tag{20}$$

The approximate derivatives of $u(\mathbf{x})$ can be expressed in terms of the derivatives of the MLS shape functions. For instance, the first and second derivatives of $u(\mathbf{x})$, evaluated at $\mathbf{x}_I$, read

$$\left.\frac{\partial u(\mathbf{x})}{\partial x_\alpha}\right|_{\mathbf{x}=\mathbf{x}_I} \approx \sum_{j=1}^{n_{\mathbf{x}_I}} u_j \left.\frac{\partial N_j(\mathbf{x})}{\partial x_\alpha}\right|_{\mathbf{x}=\mathbf{x}_I}, \qquad \left.\frac{\partial^2 u(\mathbf{x})}{\partial x_\alpha \partial x_\beta}\right|_{\mathbf{x}=\mathbf{x}_I} \approx \sum_{j=1}^{n_{\mathbf{x}_I}} u_j \left.\frac{\partial^2 N_j(\mathbf{x})}{\partial x_\alpha \partial x_\beta}\right|_{\mathbf{x}=\mathbf{x}_I} \tag{21}$$

The successive derivatives of the shape functions are obtained in terms of the derivatives of the polynomial basis $\mathbf{p}((\mathbf{x}-\mathbf{x}_I)/\rho)$ and the derivatives of $\mathbf{C}(\mathbf{x})$ [33].

A wide variety of kernel functions appears in the literature, most of them being spline or exponential functions. In this study we use a very popular cubic spline

$$W_j(\mathbf{x}) = W(\mathbf{x}-\mathbf{x}_j, \rho) = \frac{\alpha}{\rho^v}\begin{cases} 1 - \frac{3}{2}s^2 + \frac{3}{4}s^3 & s \leqslant 1 \\ \frac{1}{4}(2-s)^3 & 1 < s \leqslant 2 \\ 0 & s > 2 \end{cases} \tag{22}$$

where $s = \|\mathbf{x}-\mathbf{x}_j\|/\rho$, v is the number of dimensions and α takes the value $\frac{2}{3}$, $10/7\pi$ or $1/\pi$ in one, two or three dimensions, respectively. The coefficient α/ρ^v is a scale factor necessary only if non-corrected SPH interpolation is being used, to assure the normality property $\int W \,\mathrm{d}V = 1$. We do not use it in our MLS computations. Anisotropic weightings for 2D/3D computations can be constructed, for instance, as tensor product of 1D kernels, as

$$W_j(\mathbf{x}-\mathbf{x}_j) = \prod_{n=1}^{v} W_j^n(x^n - x_j^n, \rho^n) \tag{23}$$

where x^n is the nth Cartesian co-ordinate of particle $\mathbf{x}$. In the above expression we let W_j^n and ρ^n (the 1D kernel function and its characteristic smoothing length) be different for each spatial dimension.

2.2.1. Diffuse derivatives. The concept of diffuse derivative is very interesting from a computational point of view in MLS approximations. In the diffuse approach, the derivatives of $\mathbf{C}(\mathbf{x})$ are always neglected. Thus, the derivatives of the shape functions can only be written in terms of the derivatives of the basis functions $\mathbf{p}((\mathbf{x}-\mathbf{x}_I)/\rho)$ as

$$\frac{\partial^k \mathbf{N}^{\mathrm{T}}(\mathbf{x})}{\partial x^{\alpha}\partial x^{(k-\alpha)}} \approx \frac{\partial^k \mathbf{p}^{\mathrm{T}}(\mathbf{0})}{\partial x^{\alpha}\partial x^{(k-\alpha)}}\mathbf{C}(\mathbf{x}) \tag{24}$$

It has been shown (see Reference [34] and references therein) that the diffuse derivatives of a function $u(\mathbf{x})$ converge at optimal rate to the exact derivatives.

2.3. Application to finite volume procedures on unstructured grids

The MLS technique outlined above will be used to provide a general and accurate approximation framework for finite volume schemes on unstructured grids (some kind of FV shape functions). The computation of the MLS shape functions and their derivatives involves two major steps:

- Determination of the 'neighbourhood' (cloud of nodes) of the evaluation point; i.e. the set of nodes (centroids) that contribute to the fit.
- Evaluation of the MLS shape functions and their required full/diffuse derivatives, as exposed above.

Once the shape functions and their derivatives have been evaluated at a certain location $\mathbf{x}$, the flow variables and their successive derivatives can be approximated using (21), or similar expressions for higher-order derivatives. Note that, using fixed clouds, the MLS shape functions do not change in time and, therefore, they need to be computed only once at the preprocessing phase.

As mentioned before, a key issue concerning MLS approximations in the context of finite volume schemes on unstructured grids is the definition of the cloud of nodes (somewhat the MLS *stencil*) for each evaluation point. The selection process must be suitable for general unstructured grids, and the stencil should be as compact as possible for the sake of computational efficiency and physical meaning.

Figures 2 and 3 present the stencils used in this study to compute the MLS shape functions at the centroids and edge quadrature points, respectively. A stronger enforcement of the boundary conditions was achieved through the introduction of a set of 'zero area' cells attached to the boundary (an approach analogous to the use of so-called *ghost cells* [35]). Note that the centroids of these boundary cells, i.e. the midpoints of those edges lying on the boundary, have been included in the above stencils.

Once the cloud of neighbour centroids has been determined, the smoothing length ρ is set to be proportional to the maximum distance between the evaluation point $\mathbf{x}_I$ and its neighbours, as

$$\rho = k\ \max(\|\mathbf{x}_j - \mathbf{x}_I\|) \tag{25}$$

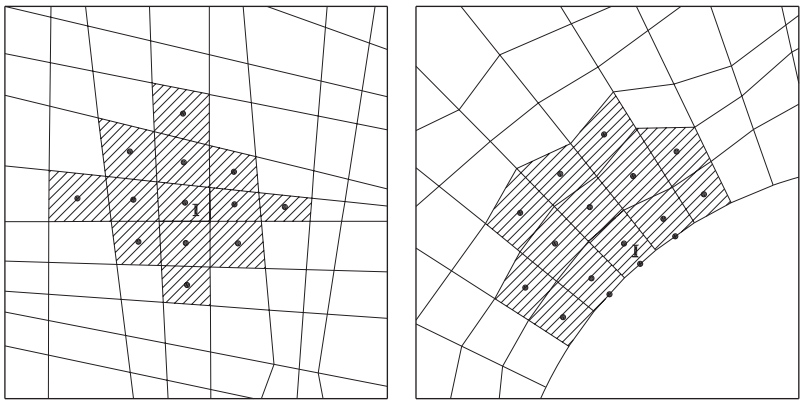

Figure 2. Typical MLS stencil: centroids.

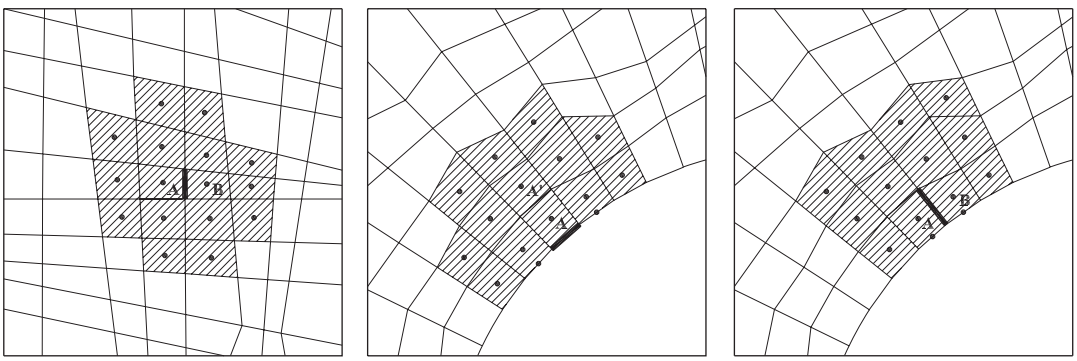

Figure 3. Typical MLS stencil: quadrature points on edges.

Values of k around 0.6–0.7 seem to be adequate (recall that, using radial weighting, the support of the kernel expands over a circle of radius 2ρ). Note that a *cell-centred* approach with quadrilateral volumes has been adopted in this study. However, the exposed methodology could be extended to any other type of control volumes, provided that suitable stencils are defined.

3. TWO NUMERICAL SCHEMES FOR THE SHALLOW WATER EQUATIONS

The spatial approximation described above was used as the basis to construct two different high-order finite volume formulations for the set of shallow water equations.

The first formulation corresponds to a Lax–Wendroff-type centred scheme with added artificial dissipation. The spatial finite volume discretization uses the MLS approximation as a kind of 'shape functions' for unstructured grids, which provide the general approximation framework.

This class of centred schemes requires the introduction of an artificial dissipation model, in order to deal with shocks and/or steep gradients in the flow variables.

The second formulation follows the ideas of the generalized Godunov method [18, 19, 23], with piecewise polynomial reconstructions inside each cell. In this study, both the computation of the successive derivatives of the flow variables and the evaluation of the diffusive fluxes have been addressed by means of MLS approximations.

3.1. Governing equations

The shallow water mathematical model is obtained by depth-integrating the Navier–Stokes equations, neglecting the vertical component of the acceleration and assuming that the fluid is incompressible and isothermal. In conservative form, the resulting system of equations can be written as [2]

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} = \mathbf{R}_s + \frac{\partial \mathbf{R}_{dx}}{\partial x} + \frac{\partial \mathbf{R}_{dy}}{\partial y} \tag{26}$$

being

$$\mathbf{U} = \begin{pmatrix} h \\ hu_x \\ hu_y \end{pmatrix} \tag{27}$$

$$\mathbf{F}_x = \begin{pmatrix} hu_x \\ hu_x^2 + \frac{1}{2}g(h^2 - H^2) \\ hu_x u_y \end{pmatrix} \quad \mathbf{F}_y = \begin{pmatrix} hu_y \\ hu_x u_y \\ hu_y^2 + \frac{1}{2}g(h^2 - H^2) \end{pmatrix} \tag{28}$$

$$\mathbf{R}_{dx} = \begin{pmatrix} 0 \\ 2\nu h \dfrac{\partial u_x}{\partial x} \\ \nu h \left(\dfrac{\partial u_y}{\partial x} + \dfrac{\partial u_x}{\partial y} \right) \end{pmatrix} \quad \mathbf{R}_{dy} = \begin{pmatrix} 0 \\ \nu h \left(\dfrac{\partial u_y}{\partial x} + \dfrac{\partial u_x}{\partial y} \right) \\ 2\nu h \dfrac{\partial u_y}{\partial y} \end{pmatrix} \tag{29}$$

$$\mathbf{R}_S = \begin{pmatrix} 0 \\ g(h - H)\dfrac{\partial H}{\partial x} - \dfrac{gn^2 |\mathbf{u}| u_x}{h^{1/3}} \\ g(h - H)\dfrac{\partial H}{\partial y} - \dfrac{gn^2 |\mathbf{u}| u_y}{h^{1/3}} \end{pmatrix} \tag{30}$$

In the above, $\mathbf{u} = (u_x, u_y)$ is the depth-averaged velocity, h is the total height of fluid, H is a certain reference level (mean water level), g is the gravity acceleration and ν is the eddy viscosity coefficient. The Chezy–Manning formula has been used to model the bottom friction,

where n represents the Manning friction coefficient. Coriolis acceleration, surface traction and variable atmospheric pressure effects have been neglected.

3.2. A one-step Lax–Wendroff scheme

The Lax–Wendroff time marching algorithm is obtained by performing a second-order Taylor series expansion in time about $t = t^n$, as

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t \left(\frac{\partial \mathbf{U}}{\partial t}\right)^n + \frac{\Delta t^2}{2}\left(\frac{\partial^2 \mathbf{U}}{\partial t^2}\right)^n \tag{31}$$

The time derivatives are expressed in terms of spatial derivatives using the original Equation (26), to yield [2]

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t \left(\mathbf{R}_s + \frac{\partial \mathbf{R}_{di}}{\partial x_i} - \frac{\partial \mathbf{F}_i}{\partial x_i}\right)^n + \frac{\Delta t^2}{2}\left\{\mathbf{G}\left(\mathbf{R}_s - \frac{\partial \mathbf{F}_i}{\partial x_i}\right) - \frac{\partial}{\partial x_i}\left[\mathbf{A}_i\left(\mathbf{R}_s - \frac{\partial \mathbf{F}_j}{\partial x_j}\right)\right]\right\}^n \tag{32}$$

where all derivatives of order higher than second have been dropped. The notation

$$\frac{\partial \mathbf{F}_i}{\partial x_i} = \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y}, \quad \frac{\partial \mathbf{R}_{di}}{\partial x_i} = \frac{\partial \mathbf{R}_{dx}}{\partial x} + \frac{\partial \mathbf{R}_{dy}}{\partial y} \tag{33}$$

$$\frac{\partial}{\partial x_i}\left[\mathbf{A}_i\left(\mathbf{R}_s - \frac{\partial \mathbf{F}_j}{\partial x_j}\right)\right] = \frac{\partial}{\partial x}\left[\mathbf{A}_x\left(\mathbf{R}_s - \frac{\partial \mathbf{F}_j}{\partial x_j}\right)\right] + \frac{\partial}{\partial y}\left[\mathbf{A}_y\left(\mathbf{R}_s - \frac{\partial \mathbf{F}_j}{\partial x_j}\right)\right] \tag{34}$$

has been used for simplicity, and

$$\mathbf{A}_x = \frac{\partial \mathbf{F}_x}{\partial \mathbf{U}}, \quad \mathbf{A}_y = \frac{\partial \mathbf{F}_y}{\partial \mathbf{U}}, \quad \mathbf{G} = \frac{\partial \mathbf{R}_s}{\partial \mathbf{U}} \tag{35}$$

are the Jacobian matrices of the convective fluxes and source term, respectively. The particular expression for $\mathbf{G}$ depends on the source terms considered. The Jacobians $\mathbf{A}_x$ and $\mathbf{A}_y$ are

$$\mathbf{A}_x = \begin{pmatrix} 0 & 1 & 0 \\ -u_x^2 + gh & 2u_x & 0 \\ -u_x u_y & u_y & u_x \end{pmatrix}, \quad \mathbf{A}_y = \begin{pmatrix} 0 & 0 & 1 \\ -u_x u_y & u_y & u_x \\ -u_y^2 + gh & 0 & 2u_y \end{pmatrix} \tag{36}$$

The integration of (32) over a cell (control volume) Ω yields to

$$\int_\Omega \Delta \mathbf{U}\, \mathrm{d}\Omega = \Delta t \int_\Omega \left(\mathbf{R}_s + \frac{\partial \mathbf{R}_{di}}{\partial x_i} - \frac{\partial \mathbf{F}_i}{\partial x_i}\right)^{(n)} \mathrm{d}\Omega + \frac{\Delta t^2}{2}\int_\Omega \left\{\mathbf{G}\left(\mathbf{R}_s - \frac{\partial \mathbf{F}_i}{\partial x_i}\right) - \frac{\partial}{\partial x_i}\left[\mathbf{A}_i\left(\mathbf{R}_s - \frac{\partial \mathbf{F}_j}{\partial x_j}\right)\right]\right\}^n \mathrm{d}\Omega \tag{37}$$

Making use of the divergence theorem and rearranging the different terms, it is obtained

$$\int_\Omega \Delta\mathbf{U}\,\mathrm{d}\Omega = \Delta t \int_\Gamma (\mathscr{R}_d - \mathscr{F})^{(n)} \cdot \mathbf{n}\,\mathrm{d}\Gamma - \frac{\Delta t^2}{2}\int_\Gamma \mathscr{S}^n \cdot \mathbf{n}\,\mathrm{d}\Gamma + \Delta t \int_\Omega \mathbf{R}_s^n \mathrm{d}\Omega + \frac{\Delta t^2}{2}\int_\Omega \left[\mathbf{G}\left(\mathbf{R}_s - \frac{\partial \mathbf{F}_i}{\partial x_i}\right)\right]^n \mathrm{d}\Omega \tag{38}$$

where $\mathbf{n}$ is the outward pointing unit normal to the boundary Γ and

$$\mathscr{F} = (\mathbf{F}_x, \mathbf{F}_y), \quad \mathscr{R}_d = (\mathbf{R}_{dx}, \mathbf{R}_{dy}), \quad \mathscr{S} = \left(\mathbf{A}_x\left(\mathbf{R}_s - \frac{\partial \mathbf{F}_j}{\partial x_j}\right), \quad \mathbf{A}_y\left(\mathbf{R}_s - \frac{\partial \mathbf{F}_j}{\partial x_j}\right)\right) \tag{39}$$

In the absence of source terms, $\mathbf{R}_s = \mathbf{0}$ and Equation (38) reduces to

$$\int_\Omega \Delta\mathbf{U}\,\mathrm{d}\Omega = \Delta t \int_\Gamma (\mathscr{R}_d - \mathscr{F})^{(n)} \cdot \mathbf{n}\,\mathrm{d}\Gamma + \frac{\Delta t^2}{2}\int_\Gamma \left(\mathbf{A}_x \frac{\partial \mathbf{F}_j}{\partial x_j} n_x + \mathbf{A}_y \frac{\partial \mathbf{F}_j}{\partial x_j} n_y\right)^{(n)} \mathrm{d}\Gamma \tag{40}$$

Adopting a standard finite volume discretization for (38), surface integrals are computed using the centroid of each cell, where the conserved variables are stored, and boundary integrals are evaluated at certain representative points (e.g. at the centre of each edge). Thus, the discrete equation for each cell I results

$$\Delta\mathbf{U}_I A_I = \Delta t \sum_{i\mathrm{edge}}^{n\mathrm{edge}} (\mathscr{R}_d - \mathscr{F})^n_{i\mathrm{edge}} \cdot \mathbf{n}_{i\mathrm{edge}} L_{i\mathrm{edge}} - \frac{\Delta t^2}{2}\sum_{i\mathrm{edge}}^{n\mathrm{edge}} \mathscr{S}^n_{i\mathrm{edge}} \cdot \mathbf{n}_{i\mathrm{edge}} L_{i\mathrm{edge}} + \Delta t \mathbf{R}^n_{sI} A_I + \frac{\Delta t^2}{2}\left[\mathbf{G}\left(\mathbf{R}_s - \frac{\partial \mathbf{F}_j}{\partial x_j}\right)\right]^n_I A_I \tag{41}$$

where A_I is the area of cell I, $n\mathrm{edge}_I$ the number of edges, $L_{i\mathrm{edge}}$ the length of edge $i\mathrm{edge}$ and $\mathbf{U}_I$ the average value of $\mathbf{U}$ over the cell I (associated to the cell centroid).

3.2.1. Spatial approximation. The final numerical algorithm is obtained after introducing the spatial approximation presented in Section 2 into the above general formulation. Recall the MLS approximation $\hat{\phi}(\mathbf{x})$ of a function $\phi(\mathbf{x})$, given by

$$\hat{\phi}(\mathbf{x}) = \sum_{j=1}^{n_\mathbf{x}} \phi_j N_j(\mathbf{x}) \tag{42}$$

in terms of the values of the variables $\{\phi_j, j = 1, \ldots, n_\mathbf{x}\}$ at $n_\mathbf{x}$ neighbouring cell centres. The approximate gradient $\nabla\hat{\phi}$ is computed as

$$\nabla\hat{\phi}(\mathbf{x}) = \sum_{j=1}^{n_\mathbf{x}} \phi_j \nabla N_j(\mathbf{x}) \tag{43}$$

This interpolation scheme provides the basis to reconstruct the necessary information at the cell edges. Assuming a group representation, convective fluxes are first computed at the centroids using the cell-average information, and then interpolated at the edge quadrature points as

$$\mathbf{F}_x(\mathbf{x}_{i\text{edge}}) = \sum_{j=1}^{n_i} \mathbf{F}_{xj} N_j(\mathbf{x}_{i\text{edge}}), \quad \mathbf{F}_y(\mathbf{x}_{i\text{edge}}) = \sum_{j=1}^{n_i} \mathbf{F}_{yj} N_j(\mathbf{x}_{i\text{edge}}) \tag{44}$$

where, for simplicity, $n_i = n_{\mathbf{x}_{i\text{edge}}}$ denotes the number of neighbour centroids taken into account in the reconstruction process. Similarly, other required entities can be interpolated as

$$\mathbf{R}_s(\mathbf{x}_{i\text{edge}}) = \sum_{j=1}^{n_i} \mathbf{R}_{sj} N_j(\mathbf{x}_{i\text{edge}}), \quad \left.\frac{\partial \mathbf{F}_k}{\partial x_k}\right|_{\mathbf{x}_{i\text{edge}}} = \sum_{j=1}^{n_i} (\mathbf{F}_{xj}, \mathbf{F}_{yj}) \cdot \nabla N_j(\mathbf{x}_{i\text{edge}}) \tag{45}$$

$$\mathbf{A}_x(\mathbf{x}_{i\text{edge}}) = \sum_{j=1}^{n_i} \mathbf{A}_{xj} N_j(\mathbf{x}_{i\text{edge}}), \quad \mathbf{A}_y(\mathbf{x}_{i\text{edge}}) = \sum_{j=1}^{n_i} \mathbf{A}_{yj} N_j(\mathbf{x}_{i\text{edge}}) \tag{46}$$

Diffusive fluxes are not computed following this scheme, but directly at the edges. For such purpose, the velocity gradient is required at the quadrature points, which is computed as

$$\nabla u_x(\mathbf{x}_{i\text{edge}}) = \sum_{j=1}^{n_i} u_{xj} \nabla N_j(\mathbf{x}_{i\text{edge}}), \quad \nabla u_y(\mathbf{x}_{i\text{edge}}) = \sum_{j=1}^{n_i} u_{yj} \nabla N_j(\mathbf{x}_{i\text{edge}}) \tag{47}$$

or, in compact form,

$$\nabla \mathbf{u}(\mathbf{x}_{i\text{edge}}) = \sum_{j=1}^{n_i} \mathbf{u}_j \otimes \nabla N_j(\mathbf{x}_{i\text{edge}}) \tag{48}$$

In general, any variable and its gradient can be computed using Equations (42) and (43) and the information stored at the centroids.

3.2.2. Artificial viscosity. As a consequence of the centred character of the spatial approximation, the above Lax–Wendroff scheme is not free from spurious oscillations in the presence of shocks. Some artificial dissipation model is therefore required to preclude the onset of instabilities near discontinuities. Making use again of the approximation framework provided by the MLS shape functions, we propose a rather simple shock-capturing technique which is completely analogous to those used in finite element general purpose algorithms when applied to high-speed flows. Although early methods used the gradient of velocity to locate discontinuities, later studies have shown that pressure-based algorithms perform better in shock capturing [36]. Following this idea, and implemented straightforwardly as an 'added viscosity' rather than a 'smoothing' of the variables (as is commonly employed in high-speed flow computations), we add the shock-capturing viscous fluxes, $\mathbf{R}_{dx}^{\text{SC}}$ and $\mathbf{R}_{dy}^{\text{SC}}$, to the right-hand side of (26) as

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} = \mathbf{R}_s + \frac{\partial (\mathbf{R}_{dx} + \mathbf{R}_{dx}^{\text{SC}})}{\partial x} + \frac{\partial (\mathbf{R}_{dy} + \mathbf{R}_{dy}^{\text{SC}})}{\partial y} \tag{49}$$

where

$$\mathbf{R}_{dx}^{\mathrm{SC}} = \begin{pmatrix} v_h^{\mathrm{SC}} \dfrac{\partial h}{\partial x} \\ 2 v_V^{\mathrm{SC}} h \dfrac{\partial u_x}{\partial x} \\ v_V^{\mathrm{SC}} h \left(\dfrac{\partial u_y}{\partial x} + \dfrac{\partial u_x}{\partial y} \right) \end{pmatrix} \quad \mathbf{R}_{dy}^{\mathrm{SC}} = \begin{pmatrix} v_h^{\mathrm{SC}} \dfrac{\partial h}{\partial y} \\ v_V^{\mathrm{SC}} h \left(\dfrac{\partial u_y}{\partial x} + \dfrac{\partial u_x}{\partial y} \right) \\ 2 v_V^{\mathrm{SC}} h \dfrac{\partial u_y}{\partial y} \end{pmatrix} \tag{50}$$

and the shock-capturing viscosities

$$v_h^{\mathrm{SC}} = C_h \varepsilon^2 \frac{|\mathbf{u}| + c}{h} |\nabla h|, \quad v_V^{\mathrm{SC}} = C_V \varepsilon^2 \frac{|\mathbf{u}| + c}{h} |\nabla h| \tag{51}$$

In these expressions, ε is a characteristic length (e.g. the typical mesh spacing), c is the gravity wave celerity and C_h and C_V are parameters that control the amount of artificial dissipation. The required flow information, h, $\mathbf{u}$ and ∇h, is computed using the MLS shape functions. In the case of transcritical flows, an entropy fix scheme should also be included in this formulation [37].

3.3. An upwind scheme based on the generalized Godunov method

High-resolution schemes based on Riemann solvers are widely recognized as powerful computational tools to handle highly convective flows, including shock wave propagation. Recent studies have shown their superior performance, compared to artificial viscosity schemes [38]. Unfortunately, upwind schemes are frequently associated to an excessive numerical dissipation in more general applications [14–16], being rather widely regarded as 'specialized' methods, and less competitive for smooth flows.

An effective approach to reduce the amount of numerical dissipation of the upwind scheme is the development of a continuous (usually polynomial) reconstruction of the field variables inside each cell, requiring the evaluation of gradients and, eventually, higher-order derivatives. On unstructured meshes, it is difficult to obtain reconstructions of order higher than second using existing procedures, and even the development of second-order algorithms with low grid sensitivity is not straightforward [35]. It is in this context that the interesting features of mesh-free interpolation schemes such as MLS, particularly well suited to provide accurate derivatives on irregularly spaced points [32], can be exploited. This section presents a high-resolution scheme, based on Roe's flux difference splitting [24], applied to the set of shallow water equations on unstructured meshes. Second- and third-order polynomial reconstructions are developed, using MLS approximation to compute first- and second-order derivatives of the flow variables.

Recall the shallow water equations written in conservative form (26)

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} = \mathbf{R}_s + \frac{\partial \mathbf{R}_{dx}}{\partial x} + \frac{\partial \mathbf{R}_{dy}}{\partial y} \tag{52}$$

Integrating over a control volume Ω, and using the divergence theorem,

$$\int_\Omega \frac{\partial \mathbf{U}}{\partial t}\,\mathrm{d}\Omega = \int_\Gamma (\mathscr{R}_d - \mathscr{F})\cdot \mathbf{n}\,\mathrm{d}\Gamma + \int_\Omega \mathbf{R}_s\,\mathrm{d}\Omega \tag{53}$$

where $\mathbf{n}$ is the outward pointing unit normal to the boundary Γ and

$$\mathscr{F} = (\mathbf{F}_x, \mathbf{F}_y), \quad \mathscr{R}_d = (\mathbf{R}_{dx}, \mathbf{R}_{dy}) \tag{54}$$

A finite volume discretization leads to a system of ordinary differential equations

$$\frac{\partial \mathbf{U}_I}{\partial t} = \frac{1}{A_I}\sum_{i\mathrm{edge}=1}^{n\mathrm{edge}_I} [(\mathscr{R}_d - \mathscr{F})\cdot \mathbf{n}]_{i\mathrm{edge}} L_{i\mathrm{edge}} + \mathbf{R}_{sI} \tag{55}$$

where A_I is the area of cell I, $n\mathrm{edge}_I$ the number of cell edges, $L_{i\mathrm{edge}}$ the longitude of edge $i\mathrm{edge}$ and $\mathbf{U}_I$ the average value of $\mathbf{U}$ over the cell I (associated to the cell centroid). Standard ODE solvers can be applied to (55). We have used the third-order TVD–Runge–Kutta algorithm proposed by Shu and Osher [39]

$$\begin{aligned} U^1 &= U^n + \Delta t L(U^n) \\ U^2 &= \tfrac{3}{4}U^n + \tfrac{1}{4}U^1 + \tfrac{1}{4}\Delta t L(U^1) \\ U^{n+1} &= \tfrac{1}{3}U^n + \tfrac{2}{3}U^2 + \tfrac{2}{3}\Delta t L(U^2) \end{aligned} \tag{56}$$

In the above equations, the operator $L(\cdot)$ represents the right-hand side of (55). The diffusive fluxes are evaluated using the same procedure as in the Lax–Wendroff scheme, computing velocity gradients at the edge quadrature points by means of the MLS approximation. The numerical inviscid fluxes are obtained using Roe's flux difference splitting [24]. For this purpose, left ($\mathbf{U}^+$) and right ($\mathbf{U}^-$) states are defined on each interface (Figure 4). The numerical flux is then computed as [40]

$$(\mathbf{F}_x, \mathbf{F}_y)\cdot \mathbf{n} = \tfrac{1}{2}[(\mathbf{F}_x(\mathbf{U}^+), \mathbf{F}_y(\mathbf{U}^+)) + (\mathbf{F}_x(\mathbf{U}^-), \mathbf{F}_y(\mathbf{U}^-))]\cdot \mathbf{n} - \tfrac{1}{2}|\tilde{\mathbf{J}}|(\mathbf{U}^- - \mathbf{U}^+) \tag{57}$$

where $\tilde{\mathbf{J}}(\mathbf{U}^+, \mathbf{U}^-)$ is an approximate flux Jacobian, satisfying certain matrix properties [40]. Equation (57) can also be written as [41]

$$(\mathbf{F}_x, \mathbf{F}_y)\cdot \mathbf{n} = \frac{1}{2}[(\mathbf{F}_x(\mathbf{U}^+), \mathbf{F}_y(\mathbf{U}^+)) + (\mathbf{F}_x(\mathbf{U}^-), \mathbf{F}_y(\mathbf{U}^-))]\cdot \mathbf{n} - \frac{1}{2}\sum_{k=1}^{3} \tilde{\alpha}_k |\tilde{\lambda}_k| \tilde{\mathbf{r}}_k \tag{58}$$

where $\{\tilde{\lambda}_k, k=1,3\}$ and $\{\tilde{\mathbf{r}}_k, k=1,3\}$ are, respectively, the eigenvalues and eigenvectors of the approximate Jacobian $\tilde{\mathbf{J}}(\mathbf{U}^+, \mathbf{U}^-)$

$$\tilde{\lambda}_1 = \tilde{u}_x n_x + \tilde{u}_y n_y + \tilde{c}, \quad \tilde{\lambda}_2 = \tilde{u}_x n_x + \tilde{u}_y n_y, \quad \tilde{\lambda}_3 = \tilde{u}_x n_x + \tilde{u}_y n_y - \tilde{c} \tag{59}$$

$$\tilde{\mathbf{r}}_1 = \begin{pmatrix} 1 \\ \tilde{u}_x + \tilde{c}n_x \\ \tilde{u}_y + \tilde{c}n_y \end{pmatrix}, \quad \tilde{\mathbf{r}}_2 = \begin{pmatrix} 0 \\ -\tilde{c}n_y \\ \tilde{c}n_x \end{pmatrix}, \quad \tilde{\mathbf{r}}_2 = \begin{pmatrix} 1 \\ \tilde{u}_x - \tilde{c}n_x \\ \tilde{u}_y - \tilde{c}n_y \end{pmatrix} \tag{60}$$

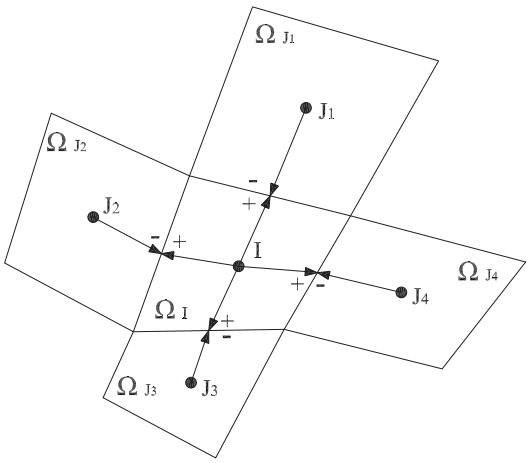

Figure 4. Extrapolated variables used to evaluate the inviscid fluxes across the interfaces of control volumes Ω_I and $\{\Omega_{J_k}, k=1,4\}$.

and the corresponding wave strengths $\{\tilde{\alpha}_k, k=1,3\}$

$$\tilde{\alpha}_1 = \frac{\Delta h}{2} + \frac{1}{2\tilde{c}}(\Delta(hu_x)n_x + \Delta(hu_y)n_y - (\tilde{u}_x n_x + \tilde{u}_y n_y)\Delta h)$$

$$\tilde{\alpha}_2 = \frac{1}{\tilde{c}}((\Delta(hu_y) - \tilde{u}_y\Delta(h))n_x - (\Delta(hu_x) - \tilde{u}_x\Delta(h))n_y) \tag{61}$$

$$\tilde{\alpha}_3 = \frac{\Delta h}{2} - \frac{1}{2\tilde{c}}(\Delta(hu_x)n_x + \Delta(hu_y)n_y - (\tilde{u}_x n_x + \tilde{u}_y n_y)\Delta h)$$

where $\Delta(\cdot) = (\cdot)^- - (\cdot)^+$, $\mathbf{n} = (n_x, n_y)$ is the outward pointing unit normal to the interface, and the Roe-averaged values (computed using $\mathbf{U}^+$ and $\mathbf{U}^-$) are defined as

$$\tilde{u}_x = \frac{u_x^+\sqrt{h^+} + u_x^-\sqrt{h^-}}{\sqrt{h^+} + \sqrt{h^-}}, \quad \tilde{u}_y = \frac{u_y^+\sqrt{h^+} + u_y^-\sqrt{h^-}}{\sqrt{h^+} + \sqrt{h^-}}, \quad \tilde{c} = \sqrt{g(h^+ + h^-)/2} \tag{62}$$

A first-order scheme is obtained by setting $\mathbf{U}^+$ and $\mathbf{U}^-$ to be the cell-average variables at the left and right control volumes. Although first-order schemes often provide valuable information for the engineering practice, their accuracy is severely undermined by an excess of numerical dissipation. More accurate methods (the so-called *high-order schemes*) can be devised by choosing 'better' values for the left and right states, through a suitable reconstruction procedure.

3.3.1. Reconstruction. This is probably the most complex issue concerning the development of accurate and robust high-order upwind schemes for unstructured grids. Reconstruction is usually addressed by substituting the piecewise constant representation of the basic first-order scheme by a piecewise polynomial reconstruction of the field variables inside each control volume,

obtained from cell-averaged data. The development of very-high-order schemes of this kind has been severely limited by the absence of robust approximation techniques, capable of providing accurate estimates of the successive derivatives of the field variables on unstructured grids. Thus, the concept of high-order scheme is almost invariably used in the literature in reference to formally second-order schemes (piecewise linear reconstruction). This section presents the linear and quadratic reconstructions used in this study. The first- and second-order derivatives of the field variables will be computed using MLS approximation.

Using a Taylor series expansion, the linear componentwise reconstruction of the field variables inside each cell I reads

$$U(\mathbf{x}) = U_I + \nabla U_I \cdot (\mathbf{x} - \mathbf{x}_I) \tag{63}$$

where U_I is the average value of U over I (associated to the centroid), $\mathbf{x}_I$ denotes the Cartesian co-ordinates of the centroid and ∇U_I is the gradient of the variable at the centroid. The aforementioned gradient is assumed to be constant inside each cell and, therefore, the reconstructed variable is still discontinuous across interfaces.

Analogously, the quadratic reconstruction reads

$$U(\mathbf{x}) = U_I + \nabla U_I \cdot (\mathbf{x} - \mathbf{x}_I) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_I)^{\mathrm{T}} \mathbf{H}_I (\mathbf{x} - \mathbf{x}_I) - \frac{1}{2}\left[I_{xx} \frac{\partial^2 U}{\partial x^2} + 2I_{xy} \frac{\partial^2 U}{\partial x \partial y} + I_{yy} \frac{\partial^2 U}{\partial y^2} \right] \tag{64}$$

where $\mathbf{H}_I$ is the centroid hessian matrix and

$$I_{xx} = \int_\Omega (x - x_I)^2 \, \mathrm{d}\Omega, \quad I_{xy} = \int_\Omega (x - x_I)(y - y_I) \, \mathrm{d}\Omega, \quad I_{yy} = \int_\Omega (y - y_I)^2 \, \mathrm{d}\Omega \tag{65}$$

The last term in (64) has been introduced to enforce conservation of the mean, i.e.

$$\frac{1}{A_I} \int_{\mathbf{x} \in \Omega_I} U(\mathbf{x}) \, \mathrm{d}\Omega = U_I \tag{66}$$

Note that the incorporation of this term does not reduce the order of the approximation given by (64). For steady-state computations we can simply use

$$U(\mathbf{x}) = U_I + \nabla U_I \cdot (\mathbf{x} - \mathbf{x}_I) + \tfrac{1}{2}(\mathbf{x} - \mathbf{x}_I)^{\mathrm{T}} \mathbf{H}_I (\mathbf{x} - \mathbf{x}_I) \tag{67}$$

In the case of unlimited reconstructions, the derivatives of the field variables are computed directly at centroids using MLS. Thus, the approximate gradients read

$$\nabla U_I = \sum_{j=1}^{n_{\mathbf{x}_I}} U_j \nabla N_j(\mathbf{x}_I) \tag{68}$$

whereas the second-order derivatives can be written as

$$\frac{\partial^2 U_I}{\partial x^2} = \sum_{j=1}^{n_{\mathbf{x}_I}} U_j \frac{\partial^2 N_j(\mathbf{x}_I)}{\partial x^2} \quad \frac{\partial^2 U_I}{\partial x \partial y} = \sum_{j=1}^{n_{\mathbf{x}_I}} U_j \frac{\partial^2 N_j(\mathbf{x}_I)}{\partial x \partial y} \quad \frac{\partial^2 U_I}{\partial y^2} = \sum_{j=1}^{n_{\mathbf{x}_I}} U_j \frac{\partial^2 N_j(\mathbf{x}_I)}{\partial y^2} \tag{69}$$

In this study, first-order derivatives are computed as full MLS derivatives, whereas second-order derivates are approximated by the diffuse ones.

The numerical integration of the inviscid fluxes was performed using one Gauss point per edge in the case of linear reconstruction. It is convenient to combine higher-order reconstructions with more accurate quadrature rules. Thus, two Gauss points were used for the quadratic reconstruction.

In practice, the use of unlimited reconstructions may lead to oscillatory solutions in the presence of shocks. What follows is a brief presentation of two families of limiting strategies for the above reconstructions.

3.3.2. Monotonicity enforcement. Barth and Jespersen [23] have proposed an extension of Van Leer's scheme [42] which is suitable for unstructured grids. The basic idea is to enforce '*monotonicity*' in the reconstructed solution. In this context, monotonicity implies that no new extrema are created by the reconstruction process [23]. The enforcement is local, in the sense that only certain neighbour cells are considered for the 'no new extrema' criterion.

Recall the piecewise linear reconstruction $U(\mathbf{x})_I$ of a variable U inside a certain cell I

$$U(\mathbf{x})_I = U_I + \nabla U_I \cdot (\mathbf{x} - \mathbf{x}_I) \tag{70}$$

and consider a *limited* version of this reconstruction, as

$$U(\mathbf{x})_I = U_I + \Phi_I \nabla U_I \cdot (\mathbf{x} - \mathbf{x}_I) \tag{71}$$

where Φ_I is a slope limiter ($0 \leqslant \Phi_I \leqslant 1$) such that reconstruction (71) satisfies

$$U^{\min} \leqslant U(\mathbf{x})_I \leqslant U^{\max} \tag{72}$$

being

$$U^{\min} = \min_{j \in \mathscr{A}_I} (U_j), \quad U^{\max} = \max_{j \in \mathscr{A}_I} (U_j) \tag{73}$$

where $\mathscr{A}_I$ is the set of 'neighbour' cells. In practice, restriction (72) is only enforced at the quadrature points on the edges of cell I; thus, for each quadrature point q, its associated slope limiter Φ_I^q is computed in terms of the unlimited extrapolated value U_I^q, as

$$\Phi_I^q = \begin{cases} \min\left(1, \dfrac{U^{\max} - U_I}{U_I^q - U_I}\right) & U_I^q - U_I > 0 \\ \min\left(1, \dfrac{U^{\min} - U_I}{U_I^q - U_I}\right) & U_I^q - U_I < 0 \\ 1 & U_I^q - U_I = 0 \end{cases} \tag{74}$$

and, finally,

$$\Phi_I = \min_q (\Phi_I^q) \tag{75}$$

In the case of the quadratic reconstruction (67), a similar limiting strategy can be written as

$$U(\mathbf{x}) = U_I + \Phi_I(\nabla U_I \cdot (\mathbf{x} - \mathbf{x}_I) + \tfrac{1}{2}(\mathbf{x} - \mathbf{x}_I)^{\mathrm{T}} \mathbf{H}_I (\mathbf{x} - \mathbf{x}_I)) \tag{76}$$

where the limiter Φ_I is obtained following the same procedure exposed above for the linear case.

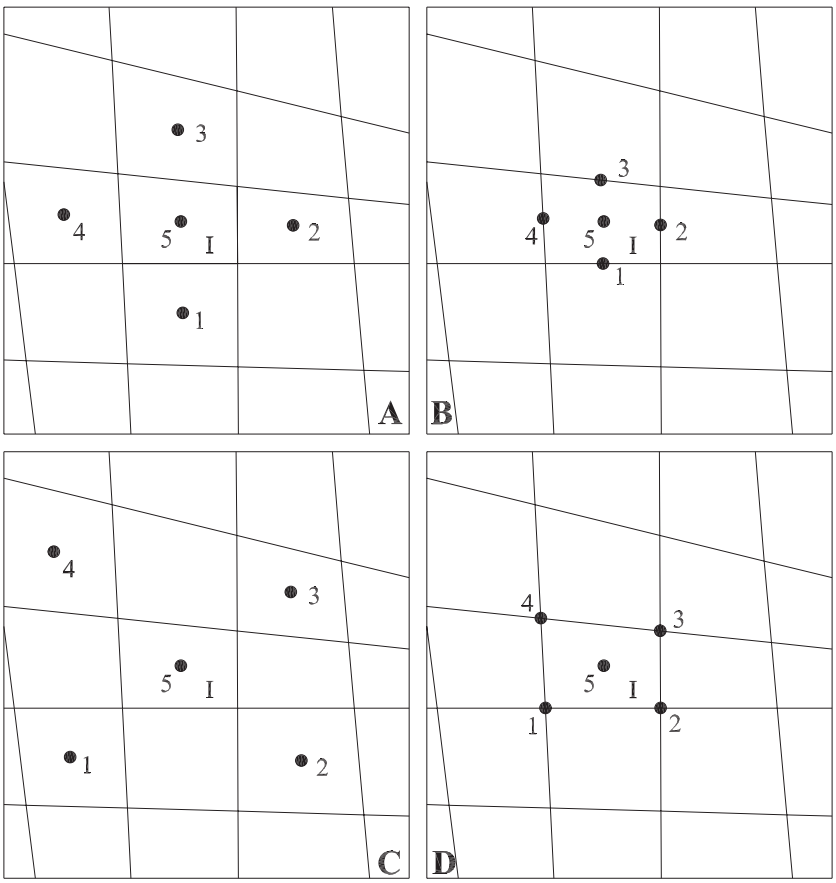

Figure 5. Neighbourhoods for the limiting of the reconstruction inside cell I.

In this study the neighbourhood to determine the extremum values $U^{\min}$ and $U^{\max}$ is comprised of the reconstruction cell I and its first-order neighbours (Figure 5(A)). In the following, the above limiter will be referred to as '*BJ limiter*'.

3.3.3. Averaged derivatives. This section presents a general strategy to obtain limited gradients and hessian matrices. Thus, the limited gradient associated to a certain cell I, ∇U_I is obtained as a weighted average of a series of *representative* gradients, as

$$\nabla U_I = \sum_{k=1}^{N} \omega_k \nabla U_k \tag{77}$$

where $\{\nabla U_k, k = 1, \ldots, N\}$ is a set of unlimited gradients, used as a basis to construct the limited one. In an approach similar to that exposed in Reference [35], the weights

$\{\omega_k, k=1,\ldots,N\}$ are given by

$$\omega_k(g_1, g_2, \ldots, g_N) = \frac{\prod_{i\neq k}^{N} g_i + \varepsilon^{N-1}}{\sum_{j=1}^{N}\left(\prod_{i\neq j}^{N} g_i\right) + N\varepsilon^{N-1}} \quad k=1,\ldots,N \tag{78}$$

where $\{g_i, i=1,\ldots,N\}$ are functions of the unlimited gradients (in this study, $g_i = \|\nabla U_i\|^2$) and ε is a small number, introduced to avoid division by zero. The hessian matrices will also be limited following these ideas, but in this case the functions g_i read

$$g_i = \left(\frac{\partial^2 U_i}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 U_i}{\partial x \partial y}\right)^2 + \left(\frac{\partial^2 U_i}{\partial y^2}\right)^2 \quad i=1,\ldots,N \tag{79}$$

For quadrilateral cells we propose a limiter based on (77)–(78) with $N=5$; i.e. the limited derivatives are obtained as a weighted average of five unlimited derivatives. Figure 5 presents four suitable configurations to determine such *representative derivatives*. In this study only the configuration given by Figure 5(A) will be considered. In the following, the above limiter will be referred to as '*PC5 limiter*'.

4. NUMERICAL EXAMPLES

This section intends to provide further insight into the behaviour of the proposed methodologies and presents additional information on computational and practical implementation issues.

It is known that high-resolution schemes are particularly well suited to yield accurate solutions of inviscid flows with shock waves. The ability to accurately capture such complex flows is also tested in the case of the Lax–Wendroff algorithm, combined with a shock-capturing viscosity model.

The dissipation properties of the proposed schemes are also analysed in the case of viscous flow at moderate to high Reynolds numbers. In the case of smooth viscous flow, the Lax–Wendroff scheme is expected to yield quite accurate solutions, thus representing a good opportunity to assess the quality of the results provided by the high-order upwind schemes.

Special attention is paid to the proposed third-order-reconstruction upwind scheme, both in inviscid and viscous flow applications. In the latter case, the low-dissipation properties of this scheme look particularly interesting, with substantial improvements with respect to the second-order scheme. The unstructured quadrilateral meshes were generated using the code GEN4U, based on the formulation proposed by Sarrate and Huerta [43].

4.1. Inviscid flows

4.1.1. Two-dimensional dam break problem. This first example is a rather classical benchmark test for discontinuous transient flow solvers. The problem set-up is depicted in Figure 6 (left), and corresponds to two reservoirs, with water levels $h_1 = 10\,\text{m}$ and $h_2 = 5\,\text{m}$, respectively, separated by an asymetrically located lockgate, which is 'instantaneously' removed at the beginning of the simulation. Viscosity and bottom friction effects are not considered. The solution at $t = 7.2\,\text{s}$ (90 time steps) was obtained using the Lax–Wendroff and Roe schemes.

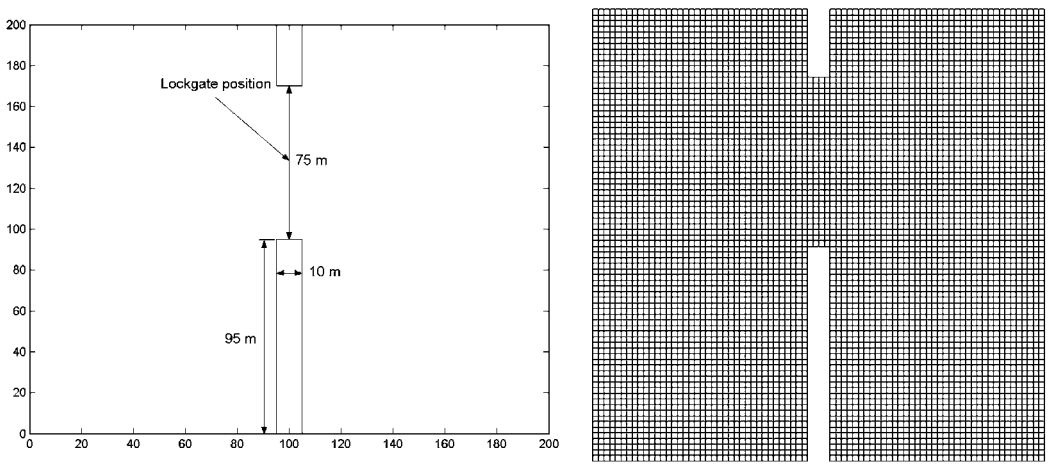

Figure 6. Two-dimensional breaking dam: problem set-up (left) and computational grid (right).

The uniform computational grid comprises of 6200 cells, with $\Delta x = \Delta y = 2.5\,\mathrm{m}$, and is depicted in Figure 6 (right).

Figure 7 presents the water depth contours obtained with the different techniques exposed in the previous sections. Cases A and B correspond to the upwind scheme, with linear and quadratic reconstructions, respectively, and the BJ limiter. Cases C and D also correspond to the upwind scheme, with linear and quadratic reconstructions, respectively, and the PC5 limiter.

Even though the use of quadratic reconstruction yields to slightly less dissipative solutions, the improvements are not so relevant considering the increase in computational cost associated to the evaluation of second-order derivatives, and the use of two Gauss points per edge. This fact is probably related to an excessively strong limiting imposed on the second-order derivatives. The sensitivity of the fine scales of the flow to the limiting of high-order derivatives is well known in the context of ENO and WENO schemes [44].

The Lax–Wendroff scheme must be combined with a shock-capturing viscosity model. The methodology proposed in Section 3 includes two free constants, C_h and C_V, and requires the definition of a characteristic length ε. The shape (and quality) of the solution is largely influenced by the adequate choice of such parameters. Cases E and F in Figure 7 depict the contours obtained with $C_h = 1$ and $C_V = 3$, and $C_h = 0.3$ and $C_V = 0.7$, respectively. In the latter case, the advancing front is reasonably well captured, but at the cost of slightly less smooth contours. On the other hand, $C_h = 1$ and $C_V = 3$ yield a smoother but also excessively dissipative solution. This example illustrates an important drawback of this kind of artificial-viscosity-based schemes, where an adequate tuning of the shock-capturing model is fundamental to obtain accurate and stable algorithms for each mesh and problem.

A 3D view of the water surface, obtained with the Roe scheme, linear reconstruction, and BJ limiter, is plotted in Figure 8.

4.1.2. Supercritical flow in a channel with variable width. Let us consider an example of supercritical flow in a symmetrical channel. The initial width of 40 m is constricted from both sides with an angle of 15°. After the constriction there follows a straight channel, being the

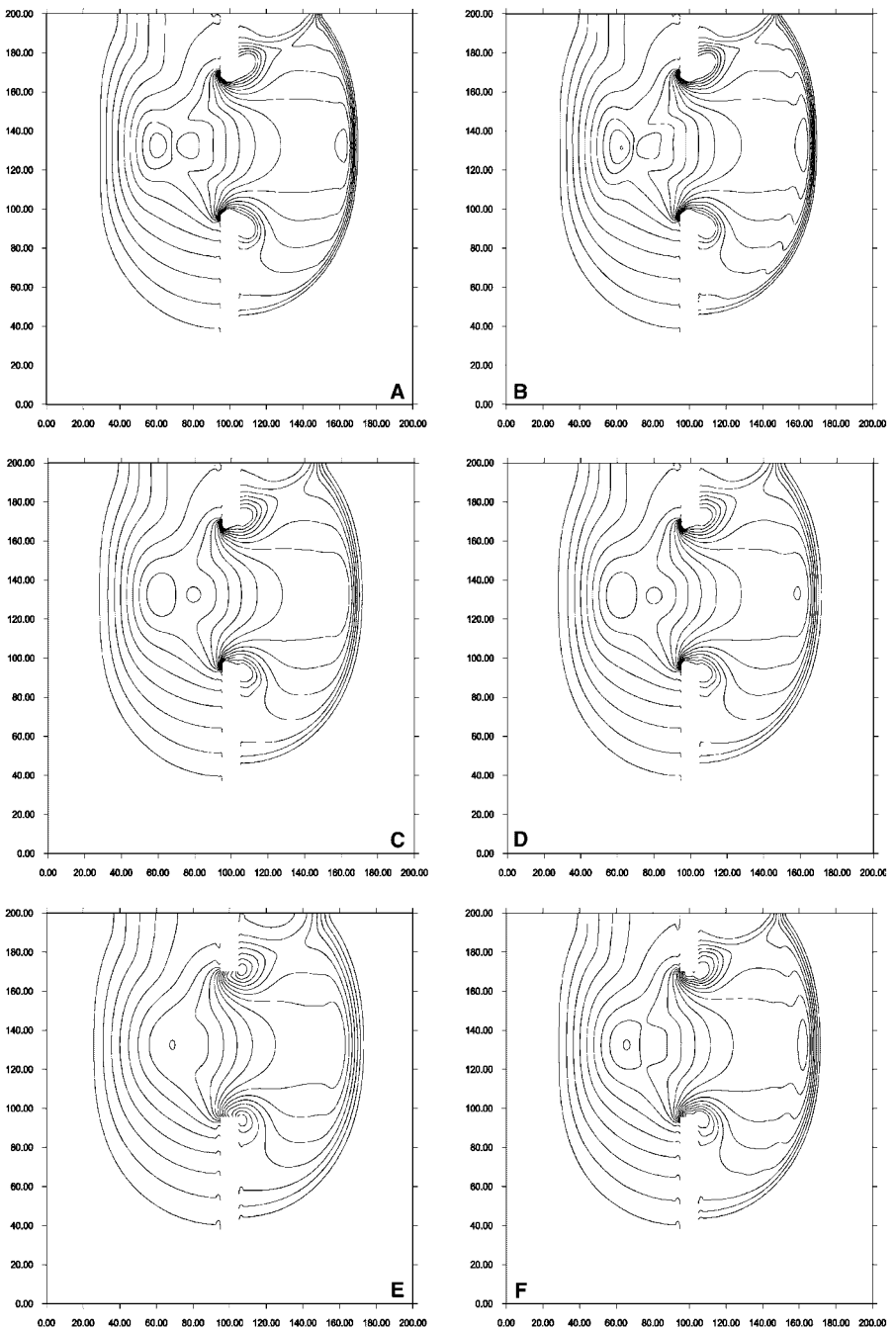

Figure 7. Two-dimensional breaking dam: computed water depth contours at $t = 7.2$ s.

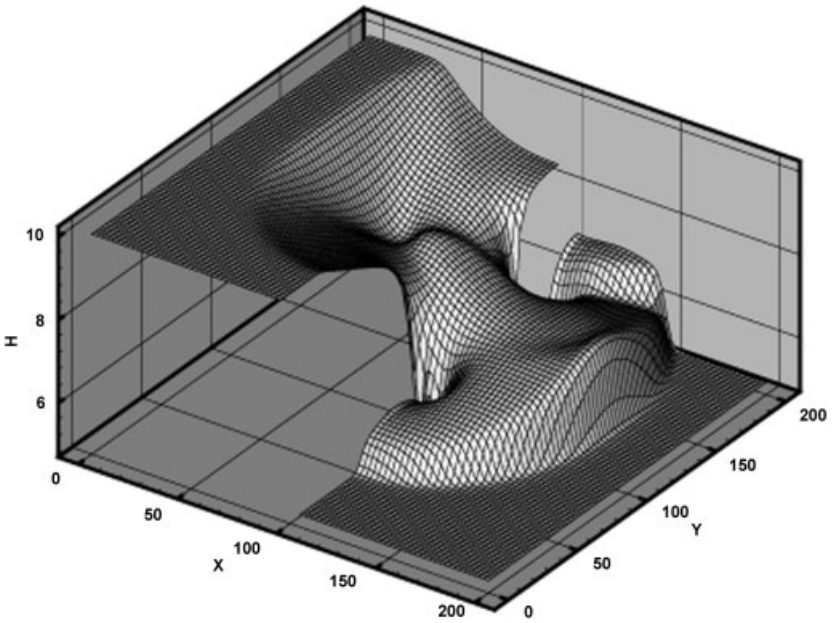

Figure 8. Two-dimensional breaking dam: 3D view of the water surface at $t = 7.2$ s.

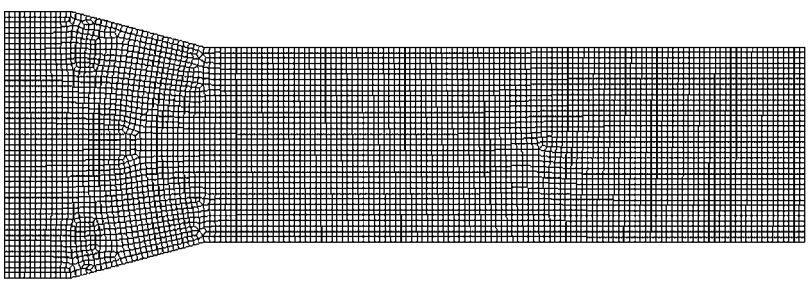

Figure 9. Supercritical flow in a channel with variable width: coarse grid (5878 cells).

total length of the domain 120 m. The imposed inlet flow parameters are: Froude number, $Fr = 3$, and unit depth, $h = 1$ m.

The steady-state flow is characterized by the classical cross-wave structure, with several hydraulic jumps. The solution is initially computed using the grid depicted in Figure 9 (5878 cells). The results obtained with the upwind scheme and various reconstruction techniques are plotted in Figures 10 and 11. Figure 10 shows the water depth contours computed using linear (top) and quadratic (bottom) reconstruction, with the BJ limiter. The results obtained with the PC5 limiter are depicted in Figure 11. The PC5 limiter appears to be slightly more dissipative, although the convergence properties of the scheme are significantly improved.

Figure 12 presents the water depth isolines computed on a finer grid of 23 349 cells with the second-order upwind scheme and BJ limiter, whereas Figure 13 shows a 3D view of the

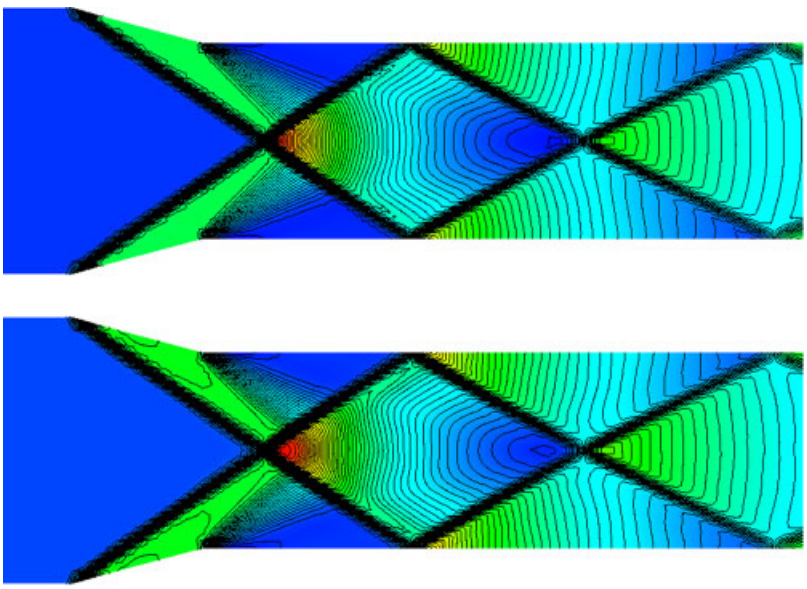

Figure 10. Supercritical flow in a channel with variable width: water depth contours. Linear (top) and quadratic (bottom) reconstructions with BJ limiter.

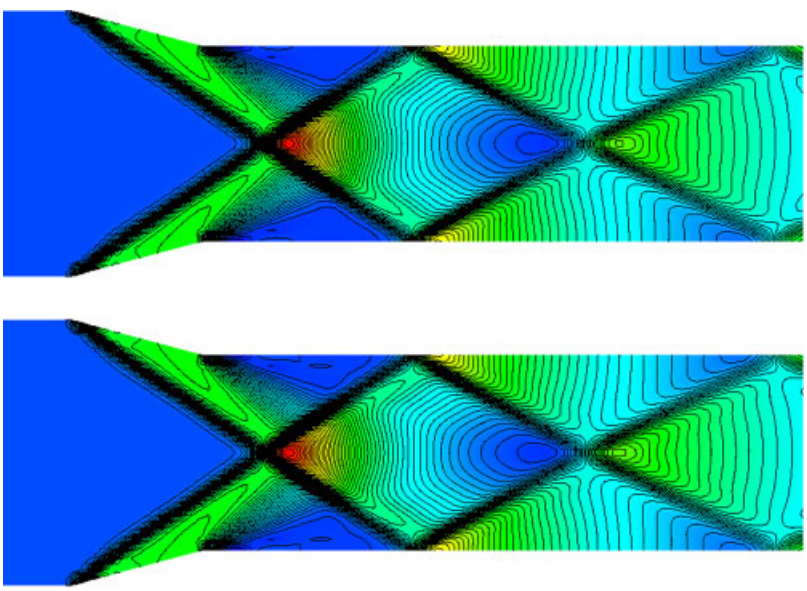

Figure 11. Supercritical flow in a channel with variable width: water depth contours. Linear (top) and quadratic (bottom) reconstructions with PC5 limiter.

corresponding water surface. The results are quite accurate, with hydraulic jumps resolved within 2 cells. Note that it was not possible to obtain comparable results using the Lax–Wendroff scheme.

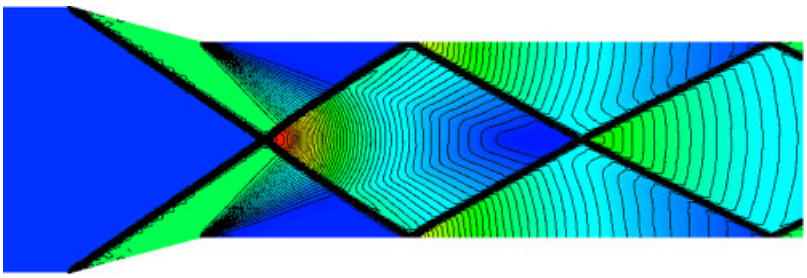

Figure 12. Supercritical flow in a channel with variable width: water depth contours (fine grid, 23 349 cells). Linear reconstruction and BJ limiter.

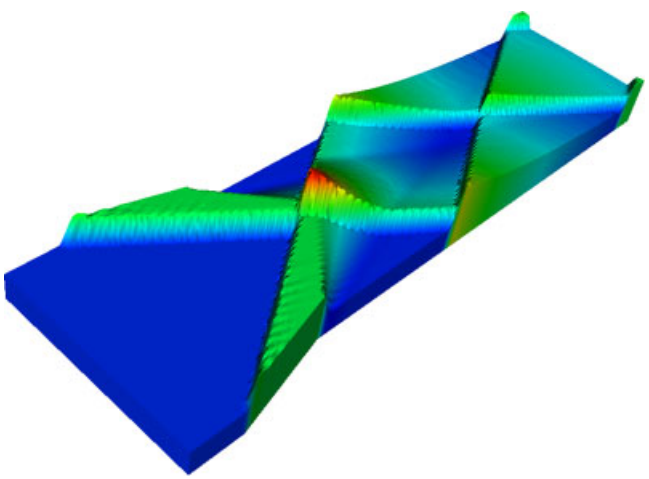

Figure 13. Supercritical flow in a channel with variable width: 3D view of the water surface.

4.1.3. Hypercritical flow past a cylinder. In this last inviscid case we consider hypercritical flow past a circular cylinder. The flow parameters at the inflow are: Froude number, $Fr = 6$ and unit water depth, $h = 1$ m. Figure 14 shows the geometry and computational grid (4719 cells) of the problem. The steady-state flow was computed using the upwind scheme with linear reconstruction.

Figures 15 and 16 depict the Froude number (left) and water depth (right) contours, obtained with the BJ and PC5 limiters, respectively. In this case, the behaviour of the more dissipative PC5 limiter is superior to that of the BJ limiter which, even though providing extremely sharp shock capturing (the hydraulic jump is contained within two cells in the frontal area), seems to be unable to completely remove the oscillations near the strong shock.

Figure 17 presents a close-up view of the Froude number contours in front of the cylinder for the BJ (left) and PC5 (right) limiters.

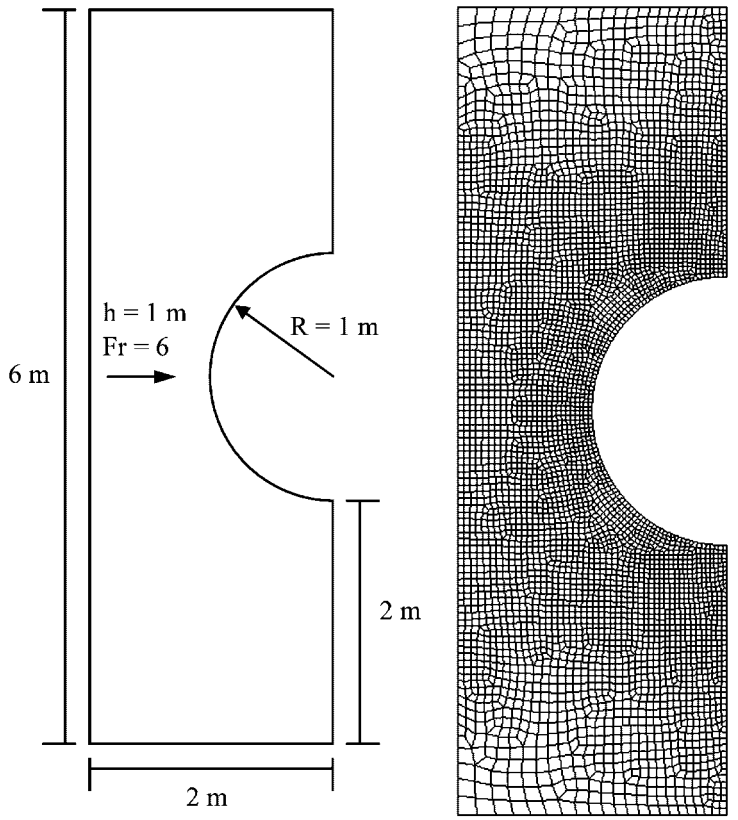

Figure 14. Hypercritical flow past a cylinder ($Fr = 6$): geometry (left) and computational grid (4719 cells, right).

4.2. Viscous flows

The practical interest of the proposed methodologies depend, to a large extent, on the ability of the schemes to provide accurate solutions in problems were subtle viscous effects are of interest. Given its low inherent dissipation, the Lax–Wendroff scheme is expected to perform satisfactorily, whereas the development of high-order reconstructions will be critical to achieve comparable performance in the case of upwind schemes.

The first example intends to evaluate the influence of the limiting procedure on the accuracy of viscous computations, whereas the second one involves a smooth flow where limiters are not needed, thus providing a closer insight into the intrinsic numerical dissipation of the different schemes.

4.2.1. Supercritical viscous flow near a wall. In analogy with a classical benchmark test for compressible flow solvers, we consider viscous supercritical flow near a solid wall. The problem set-up is outlined in Figure 18. The free stream flow parameters are: Froude number, $Fr = 1.5$, unit depth, $h = 1\,\mathrm{m}$ and Reynolds number $Re = 1000$, referred to a unit reference length, $L = 1\,\mathrm{m}$. No-slip boundary conditions were applied along the wall boundary,

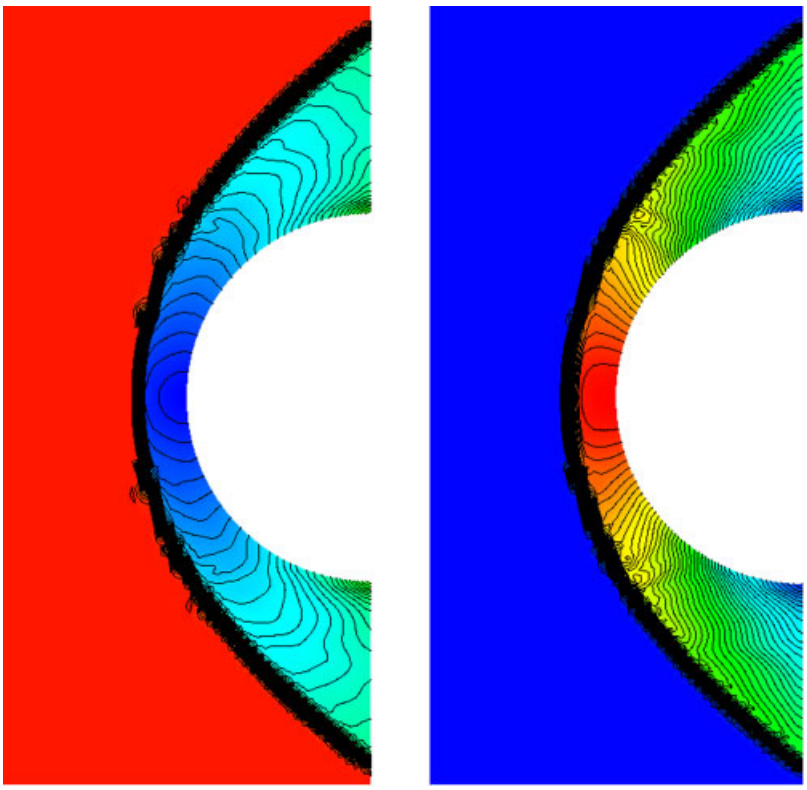

Figure 15. Hypercritical flow past a cylinder ($Fr = 6$): Froude number (left) and water depth (right) isolines. Linear reconstruction and BJ limiter.

$y = 0$, $0.2 \leqslant x \leqslant 0.8$. The flow pattern includes a shock front starting from the leading edge of the wall and a boundary layer (assumed here to be laminar) due to the presence of the no-slip condition.

The problem has been solved on two different meshes, plotted in Figures 19 and 20 (left). The first is a structured non-uniform mesh, whereas the second is a (roughly) adapted, fully unstructured mesh. In the first case, and given the mesh structure, the MLS shape functions were computed using anisotropic weighting according to (23).

Figures 21 and 22 show the computed Froude number profiles along the outlet section with, respectively, limited (BJ limiter) and unlimited reconstructions in the Roe schemes. For the centred scheme (Lax–Wendroff) the artificial viscosity model was used, with $C_h = C_v = 0.1$. In the case of limited reconstructions, the centred scheme is less dissipative that the upwind methods. On the other hand, the results improve dramatically when the limiters are turned off. The solution obtained using quadratic reconstruction is slightly better than that provided by the linear one.

Even though the limiting procedure adds more dissipation in the smooth regions of the flow, the results are reasonably close to those of the unlimited reconstructions. The Froude number contours for the upwind scheme with unlimited quadratic reconstructions are plotted in Figures 19 and 20 (right).

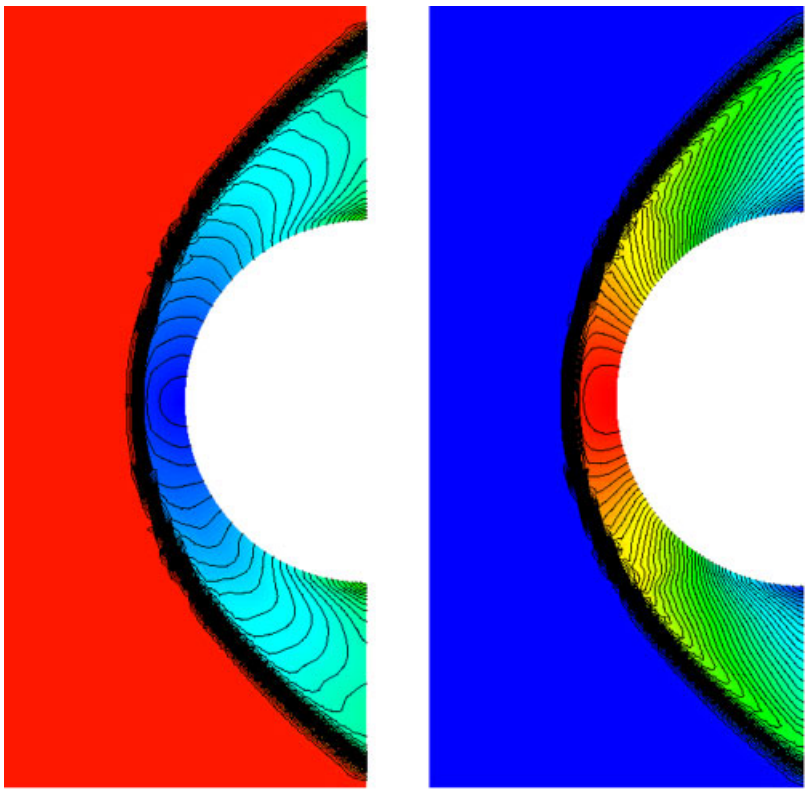

Figure 16. Hypercritical flow past a cylinder ($Fr = 6$): Froude number (left) and water depth (right) isolines. Linear reconstruction and PC5 limiter.

4.2.2. Lid-driven cavity flow. Although this is not a standard test in the shallow water literature (and is probably devoid of any hydraulic meaning), in our opinion this problem is very useful to assess the ability of the different numerical schemes to capture fine viscous features of the flow. The problem set-up is completely analogous to the classical cavity flow problem used to validate incompressible Navier–Stokes solvers. A unit square domain with flat, frictionless bottom is considered. The boundary conditions imposed are $h = 1\,\mathrm{m}$, $q_x = 1\,\mathrm{m}^2/s$ and $q_y = 0$ on $y = 1\,\mathrm{m}$ (including the corners) and solid walls ($q_x = 0$ and $q_y = 0$) elsewhere. Unit water depths were also imposed on the inferior corners. The computational grid is shown in Figure 23, and is comprised of 61×61 non-uniform cells. The grid has been refined near solid walls to take into account the thin boundary layer.

The problem was solved for Reynolds numbers of 1000 and 10 000. For comparison purposes, the solutions were also obtained on the same mesh using the finite element Taylor–Galerkin explicit formulation developed by Peraire [1, 2].

Given the absence of shocks, the Lax–Wendroff scheme was used without the introduction of any added dissipation. It was not necessary to use limiters in the reconstruction process for the upwind schemes. Given the presence of highly stretched cells near the walls, anisotropic weighting was used for the MLS computations.

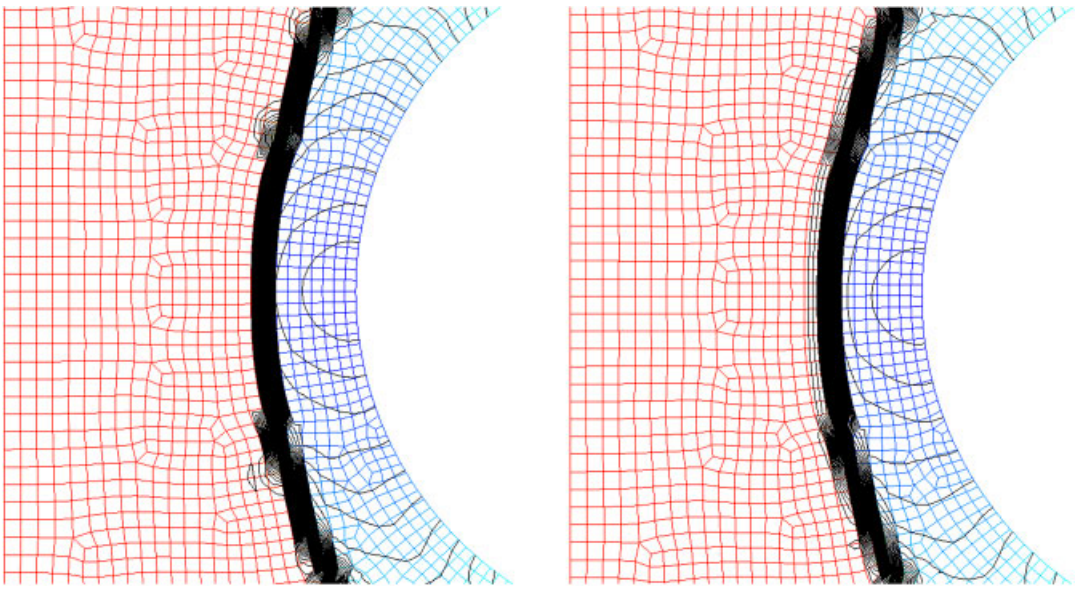

Figure 17. Hypercritical flow past a cylinder ($Fr = 6$): Froude number isolineas (detail). Linear reconstruction, BJ (left) and PC5 (right) limiters.

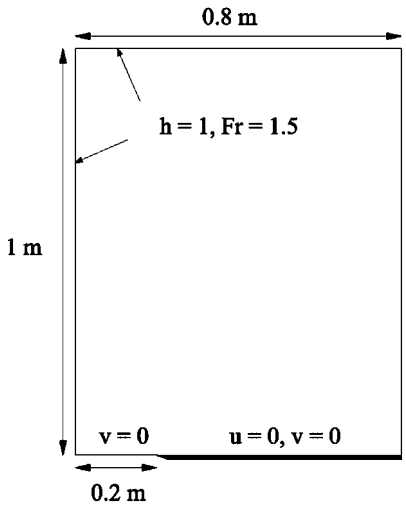

Figure 18. Supercritical viscous flow near a wall: problem set-up.

The streamlines for the Lax–Wendroff and the upwind scheme with quadratic reconstruction are depicted in Figures 24 and 25. The horizontal velocity (u_x) profiles along $x = 0.5$ m for the Taylor–Galerkin, Lax–Wendroff and upwind schemes are plotted in Figures 26 and 27.

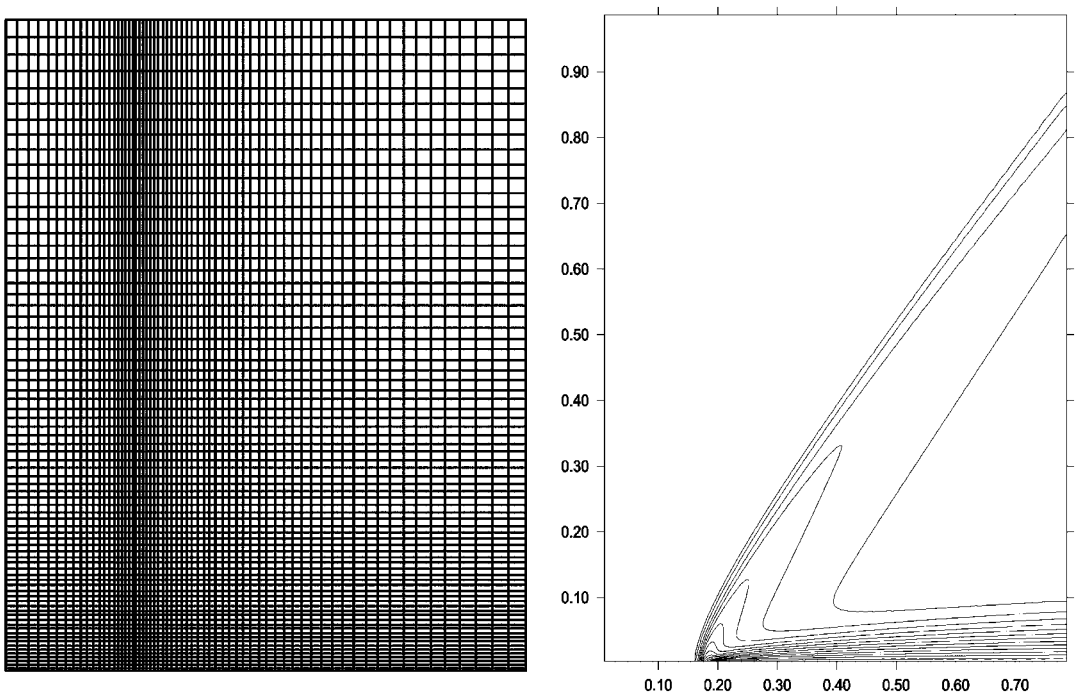

Figure 19. Supercritical viscous flow near a wall: structured grid (4875 cells, left) and Froude number isolines (right). Roe scheme with unlimited quadratic reconstruction.

The different schemes yield almost identical solutions at moderate Reynolds numbers ($Re = 1000$, Figure 26). However, at high Reynolds numbers ($Re = 10\,000$, Figure 27) the dissipation properties of each scheme become more evident. The results obtained with the FEM Taylor–Galerkin and the proposed MLS–Lax–Wendroff schemes are very similar, although the latter predicts slightly higher maximum horizontal velocities. The second and third-order-reconstruction upwind schemes provide acceptable solutions for $Re = 10\,000$. However, the maximum horizontal velocities are somewhat displaced from their correct position closer to the wall, affecting the shape of the profile. This effect could be associated to an excess of crosswind dissipation introduced by the upwind scheme. It is remarkable that this excessive dissipation is substantially reduced by the proposed third-order reconstruction. The solution obtained with this latter approach is in better agreement with the solution obtained by using the Taylor–Galerkin and MLS–Lax–Wendroff schemes.

5. CONCLUSIONS

In this paper it is explored the ability of a mesh-free interpolation technique (namely, the moving least-squares method) to be used in combination with finite volume discretizations on

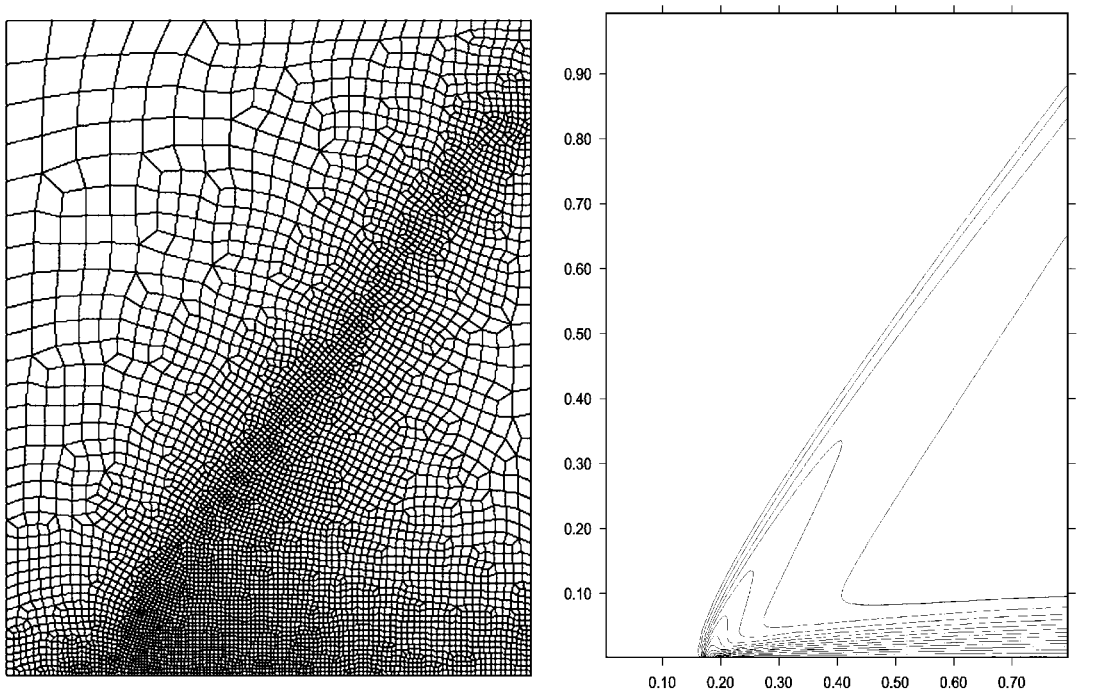

Figure 20. Supercritical viscous flow near a wall: structured grid (5426 cells, left) and Froude number isolines (right). Roe scheme with unlimited quadratic reconstruction.

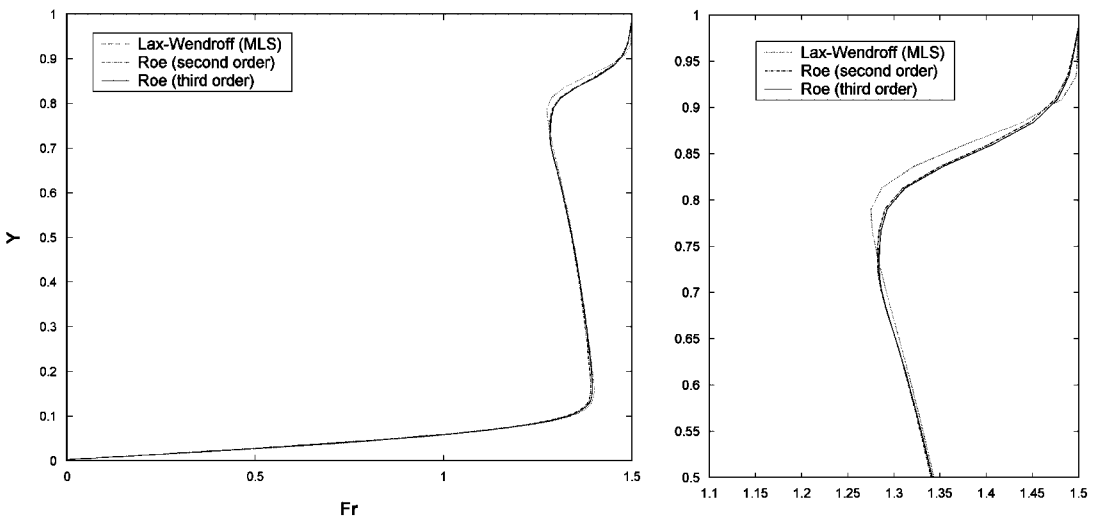

Figure 21. Supercritical viscous flow near a wall: Froude number profiles along $x = 0.8\,\mathrm{m}$ (left) and detail (right). Limited reconstructions.

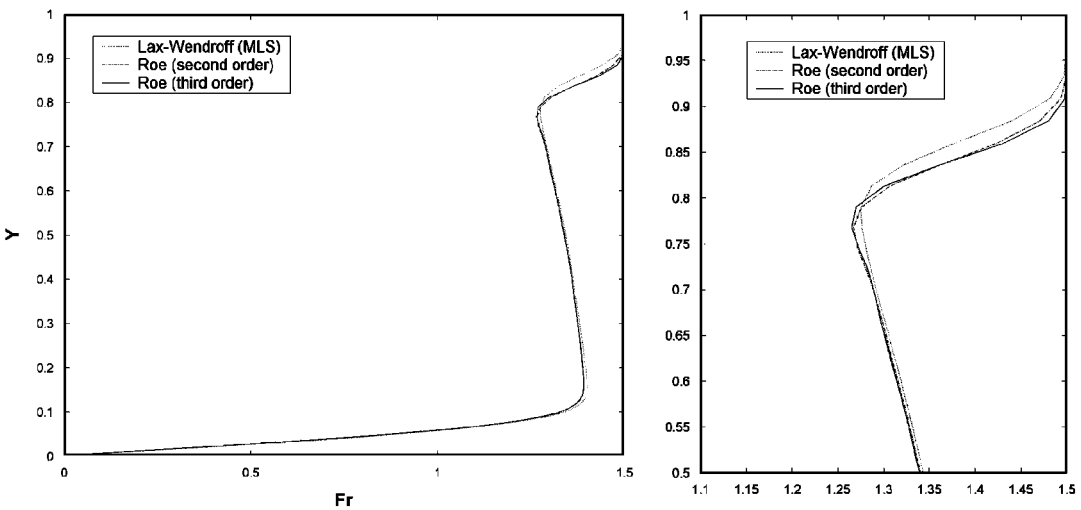

Figure 22. Supercritical viscous flow near a wall: Froude number profiles along $x = 0.8\,\mathrm{m}$ (left) and detail (right). Unlimited reconstructions.

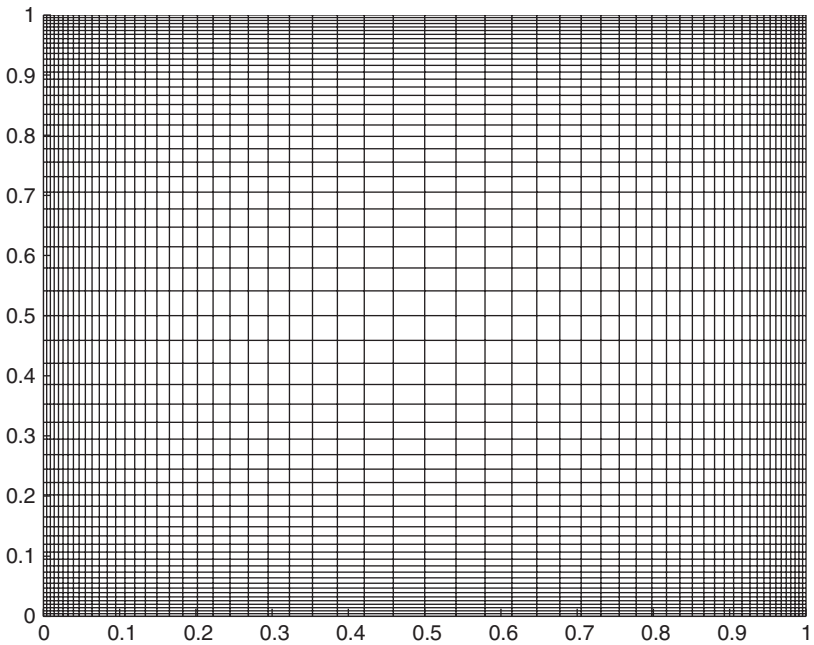

Figure 23. Lid-driven cavity flow: computational grid (61×61 cells).

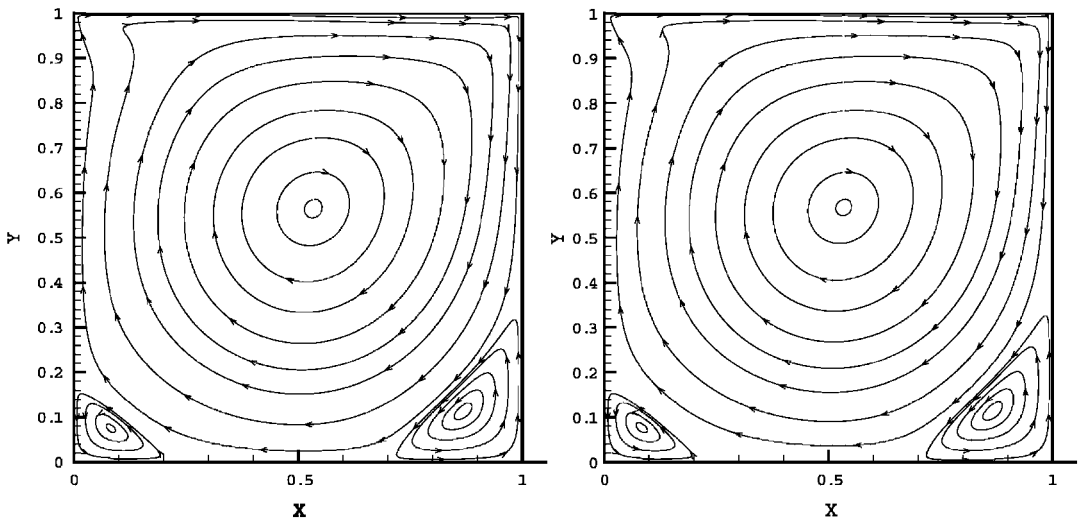

Figure 24. Lid-driven cavity flow: streamlines for test case $Re = 1000$. MLS–Lax–Wendroff scheme (left) and Roe scheme with quadratic reconstruction (right).

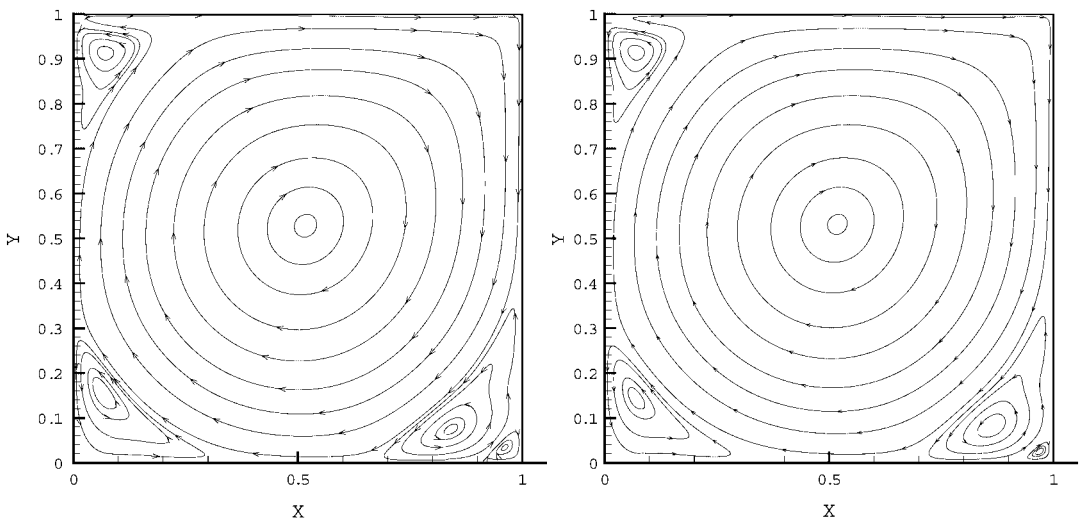

Figure 25. Lid-driven cavity flow: streamlines for test case $Re = 10\,000$. MLS–Lax–Wendroff scheme (left) and Roe scheme with quadratic reconstruction (right).

unstructured grids. The resulting numerical schemes have been applied to the resolution of the set of shallow water equations.

Special emphasis has been placed in the development of low-dissipative high-order upwind schemes, given their robustness and shock-capturing ability. In addition, and making use of the general approximation framework provided by the MLS approach, a centred in space

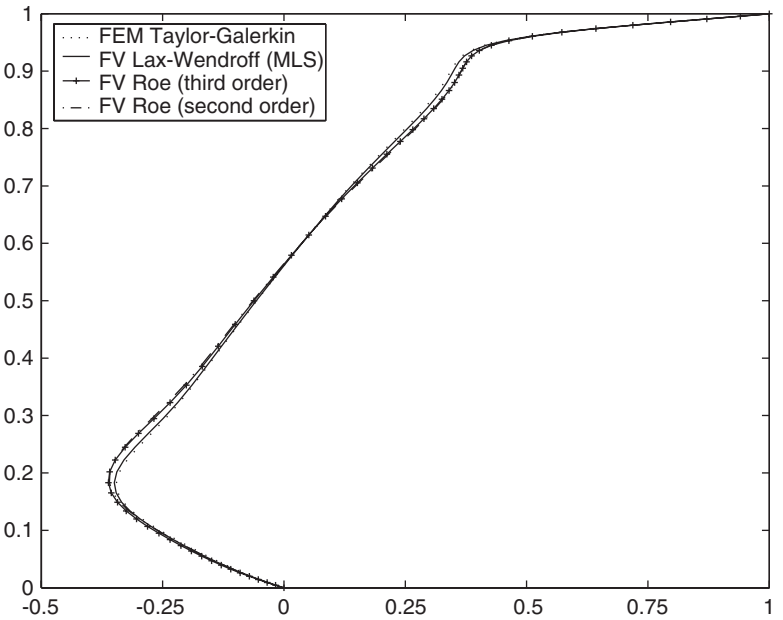

Figure 26. Lid-driven cavity flow: horizontal velocity u_x along $x = 0.5$ for $Re = 1000$.

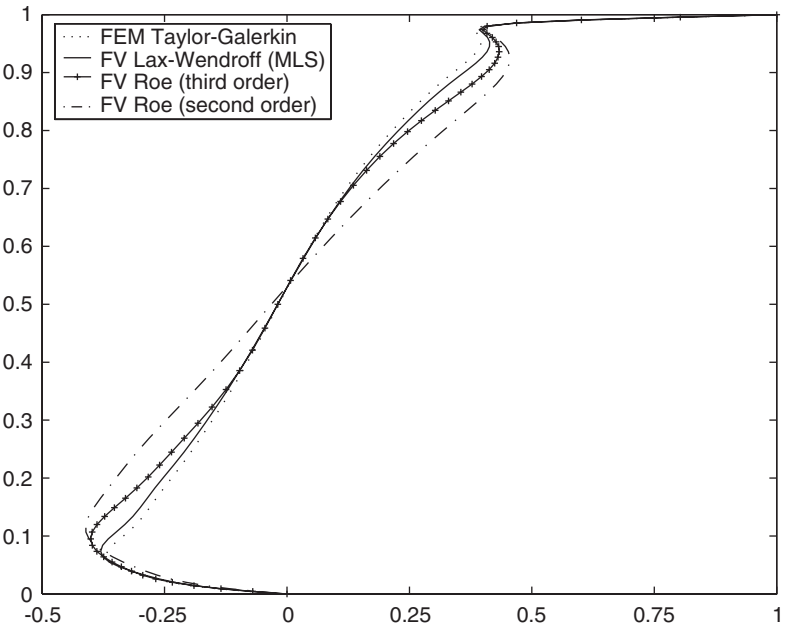

Figure 27. Lid-driven cavity flow: horizontal velocity u_x along $x = 0.5$ for $Re = 10\,000$.

Lax–Wendroff scheme has been developed, with accuracy and stability properties very similar to those of its FEM counterpart, the Taylor–Galerkin method. A shock-capturing artificial viscosity model has been additionally proposed for this Lax–Wendroff scheme.

The development of high-order upwind schemes on unstructured grids involves high-order reconstruction of the field variables or fluxes inside each cell. The main difficulty at this point is associated to the accurate evaluation of gradients and, eventually, higher order derivatives.

In this study it is shown that the class of approximation methods employed in the 'mesh-free' literature (of which the MLS technique is one example) is particularly well suited for such purpose, allowing, in addition, an accurate computation of the viscous fluxes. They provide a robust and general approximation framework being an interesting alternative to the more or less 'heuristic' existing techniques.

As expected, the performance of the more 'specialized' Roe schemes in complex flows involving shock waves is superior to that of the centred scheme with the proposed artificial viscosity model. The development of adequate limiting strategies for high-order derivatives appears to be critical in order to exploit the full potential of very high-order reconstructions applied to discontinuous flows.

On the other hand, the MLS–Lax–Wendroff scheme provides less dissipative solutions in the case of smooth viscous flows. However, the third-order-reconstruction Roe scheme developed here shows substantial improvements with respect to the second-order scheme. A more complete assessment of the suitability of this scheme for turbulent flow computations is currently in progress.

ACKNOWLEDGEMENTS

This work has been partially supported by the 'Ministerio de Educación y Ciencia' of the Spanish Government (Grant DPI#2004-05156) cofinanced with FEDER funds, the 'Xunta de Galicia' (Grants # PGDIT01PXI11802PR and PGIDIT03PXIC118002PN) and the University of La Coruña.

Mr Cueto-Felgueroso gratefully acknowledges the support received from 'Fundación de la Ingeniería Civil de Galicia' and 'Colegio de Ingenieros de Caminos, Canales y Puertos'. Most of this paper was written while Mr. Cueto-Felgueroso was visiting the University of Wales Swansea during the first semester of 2004. The support received from 'Caixanova' and the kind hospitality offered by Prof. Javier Bonet and his research group are gratefully acknowledged.

REFERENCES

1. Peraire J. A finite element method for convection dominated flows. *Ph.D. Thesis*, University of Wales Swansea, 1986.
2. Peraire J, Zienkiewicz OC, Morgan K. Shallow water problems: a general explicit formulation. *International Journal for Numerical Methods in Engineering* 1986; **22**:547–574.
3. Quecedo M, Pastor M. A reappraisal of Taylor–Galerkin algorithm for drying-wetting areas in shallow water computations. *International Journal for Numerical Methods in Fluids* 2002; **38**:515–531.
4. Quecedo M, Pastor M. Finite element modelling of free surface flows on inclined and curved beds. *Journal of Computational Physics* 2003; **189**:45–62.
5. Sheu TWH, Fang CC. High resolution finite-element analysis of shallow water equations in two dimensions. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:2581–2601.
6. Alcrudo F, García-Navarro P. A high-resolution Godunov-type scheme in finite volumes for the 2D shallow-water equations. *International Journal for Numerical Methods in Fluids* 1993; **16**:489–505.
7. Chippada S, Dawson CN, Martnez ML, Wheeler MF. A Godunov-type finite volume method for the system of shallow water equations. *Computer Methods in Applied Mechanics and Engineering* 1998; **151**:105–129.

8. Zhao DH, Shen HW, Tabious III GQ, Lai JS, Tan WY. Finite-volume two-dimensional unsteady-flow model for river basins. *Journal of Hydraulic Engineering* (ASCE) 1994; **120**:864–883.
9. Anastasiou K, Chan CT. Solution of the 2D shallow water equations using the finite volume method on unstructured triangular meshes. *International Journal for Numerical Methods in Fluids* 1997; **24**:1225–1245.
10. Hu K, Mingham CG, Causon DM. A bore-capturing finite volume method for open-channel flows. *International Journal for Numerical Methods in Fluids* 1998; **28**:1241–1261.
11. Tseng MH. Explicit finite volume non-oscillatory schemes for 2D transient free-surface flows. *International Journal for Numerical Methods in Fluids* 1999; **30**:831–843.
12. Liszka R, Wendroff B. Two-dimensional shallow water equations by composite schemes. *International Journal for Numerical Methods in Fluids* 1999; **30**:461–479.
13. Wang J, Liu R. The composite finite volume method on unstructured meshes for the two-dimensional shallow water equations. *International Journal for Numerical Methods in Fluids* 2001; **37**:933–949.
14. Fe J. Application of the finite volume method to the numerical resolution of the shallow water equations including turbulent stresses (in Spanish). *Ph.D. Thesis*, Universidade da Coruña, 2005.
15. Bui T. A parallel, finite-volume algorithm for large-eddy simulation of turbulent flows. *Computers and Fluids* 2000; **29**:877–915.
16. Lin SY, Chen YF, Shih SC. Numerical study of MUSCL schemes for computational aeroacoustics. *AIAA-97-0023*, 1997.
17. Daru V, Tenaud C. High order one-step monotonicity-preserving schemes for unsteady compressible flow calculations. *Journal of Computational Physics* 2004; **193**:563–594.
18. Godunov SK. A difference method for the numerical calculation of discontinuous solutions of hydrodynamic equations. *Mathematics of the USSR-Sbornik* 1959; **47**(3):271–306.
19. Harten A, Lax P, Van Leer B. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review* 1983; **25**:35–61.
20. Barth TJ. Aspects of unstructured grids and finite-volume solvers for the Euler and Navier–Stokes equations. *VKI Lecture Series 1994-05*, 1995.
21. Lancaster P, Salkauskas K. Surfaces generated by moving least squares methods. *Mathematics of Computation* 1981; **155**:141–158.
22. Cueto-Felgueroso L, Colominas I, Mosqueira G, Navarrina F, Casteleiro M. On the Galerkin formulation of the SPH method. *International Journal for Numerical Methods in Engineering* 2004; **60**:1475–1512.
23. Barth TJ, Jespersen DC. The design and application of upwind schemes on unstructured meshes. *AIAA-89-0366*, 1989.
24. Roe PL. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics* 1981; **43**:357–372.
25. Belytschko T, Krongauz Y, Organ D, Fleming M, Krysl P. Meshless methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering* 1998; **139**:3–47.
26. Chorin AJ. Numerical study of slightly viscous flow. *Journal of Fluid Mechanics* 1973; **57**:785–796.
27. Lucy LB. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal* 1977; **82**:1013–1024.
28. Gingold RA, Monaghan JJ. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 1977; **181**:375–389.
29. Monaghan JJ. An introduction to SPH. *Computer Physics Communications* 1988; **48**:89–96.
30. Liu WK, Li S, Belytschko T. Moving least-squares reproducing kernel methods. I. Methodology and convergence. *Computer Methods in Applied Mechanics and Engineering* 1997; **143**:113–154.
31. Cueto-Felgueroso L. A unified analysis of meshless methods: formulation and applications (in Spanish). *Technical Report*, Universidad de La Coruña, 2002.
32. Gossler A. Moving least-squares: a numerical differentiation method for irregularly spaced calculation points. *SANDIA Report*, *SAND2001-1669*, 2001.
33. Cueto-Felgueroso L. Particles, finite volumes and unstructured grids: numerical simulation of fluid dynamics problems (in Spanish). *Ph.D. Thesis*, Universidade da Coruña, 2005.
34. Huerta A, Vidal Y, Villon P. Pseudo-divergence-free element free Galerkin method for incompressible fluid flow. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:1119–1136.
35. Jawahar P, Kemath H. A high-resolution procedure for Euler and Navier–Stokes computations on unstructured grids. *Journal of Computational Physics* 2000; **164**:165–203.

36. Nithiarasu P, Zienkiewicz OC, Satya Sai BVK, Morgan K, Codina R, Vázquez M. Shock capturing viscosities for the general fluid mechanics algorithm. *International Journal for Numerical Methods in Fluids* 2000; **28**:1325–1353.
37. Burguete J, García-Navarro P. Efficient construction of high-resolution TVD conservative schemes for equations with source terms: application to shallow water flows. *International Journal for Numerical Methods in Fluids* 2001; **37**:209–248.
38. Satya Sai BVK, Zienkiewicz OC, Manzari MT, Lyra PRM, Morgan K. General purpose versus special algorithms for high-speed flows with shocks. *International Journal for Numerical Methods in Fluids* 1998; **27**:57–80.
39. Shu C-W, Osher S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics* 1988; **77**:439–471.
40. Brufau P, Vázquez-Cendón ME, García-Navarro P. A numerical model for the flooding and drying of irregular domains. *International Journal for Numerical Methods in Fluids* 2002; **39**:247–275.
41. Hubbard ME. Multidimensional slope limiters for MUSCL-type finite volume schemes on unstructured grids. *Journal of Computational Physics* 1999; **155**:54–74.
42. Van Leer B. Towards the ultimate conservative difference scheme V. A second order sequel to Godunov's method. *Journal of Computational Physics* 1979; **32**:101–136.
43. Sarrate J, Huerta A. Efficient unstructured quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering* 2000; **49**:1327–1350.
44. Serna S, Marquina A. Power ENO methods: a fifth-order accurate Weighted Power ENO method. *Journal of Computational Physics* 2004; **194**:632–658.

