INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2000; **00**:1–6 *Prepared using* nmeauth.cls *[Version: 2002/09/18 v2.02]*

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, M. Casteleiro

Group of Numerical Methods in Engineering, GMNI
Dept. 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 succesive derivatives can be accurately reconstructed using this meshfree technique in a general nodal arrangement. The methodology proposed is used in the construction of low-dissipative high-order high-resolution schemes for the shallow water equations. In particular, second and third-order-reconstruction upwind schemes for unstructured grids based on Roe's flux difference splitting are developed and applied to inviscid and viscous flows. This class of meshfree reconstruction techniques provide 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.

KEY WORDS: Shallow Water dynamics, Finite Volume method, high-resolution schemes, meshfree methods, Moving Least-Squares, 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 Mechanics. In the context of shallow water dynamics, finite element and finite volume discretizations have become very popular in recent years.

Finite element formulations for fluid dynamics are usually elegant and applicable to a wide variety of flow conditions. Unfortunately, their frequent centered 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

*Correspondence to: E.T.S. de Ingenieros de Caminos, Canales y Puertos, Universidad de La Coruña, Campus de Elviña, 15071 La Coruña, SPAIN. Email: ice@udc.es

Received Yesterday
Revised Today

succesful 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 succesfully 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. Zhao et al. [7] proposed an upwind finite volume method on unstructured grids using Osher's scheme. Anastasiou and Chan [8] 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 [9] and Tseng [10]. Liszka and Wendroff [11] introduced composite methods, which combine Lax-Wendroff and Lax-Friedrichs schemes into a multistage algorithm, and Wang and Liu [12] have recently extended the methodology to unstructured triangular meshes.

Various researchers have reported that first and even second order upwind schemes exhibit excessive numerical dissipation when applied to more general flows (not necessarily including shock wave propagation) where turbulent effects are of interest [13],[14],[15], 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 meshes should therefore not introduce exccessive 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 [16].

The endeavour to solve increasingly complex flows has promoted the advent of unstructured meshes as the most efficient approach to mesh highly irregular domains, perform adaptive refinements and capture small scale features of the flow. As far as the development of high order 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. 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.

The authors would like to propose a meshfree technique, the so-called Moving Least Squares (MLS) approximation as an accurate and efficient technique to obtain high order finite volume algorithms on unstructered meshes. 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 [17], the MLS approximation has become very popular among those researchers working in the

class of the so-called *meshless* or *meshfree* 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 played a key role to provide the spatial approximation within an arbitrary cloud of nodes [18]. In this study, second and third-order-reconstruction high-resolution schemes, based on Roe's approximate Riemann solver [19], are developed and tested for inviscid and viscous flow applications. In the latter, the approximation framework provided by these meshfree techniques is especially interesting in the accurate evaluation of the viscous fluxes at the cell faces. Although further work focused on adequate limiting strategies is neccesary to exploit the whole capabilities of the third-order scheme in problems with strong shocks, the preliminary results are encouraging.

For comparison purposes, particularly in the case os viscous flow, a Lax-Wendroff finite volume scheme was also implemented and tested on unstructered grids. In this case, we used again the MLS shape functions to obtain a continuous representation of the field variables and their derivatives within the grid. This Lax-Wendroff scheme is not free from spurious oscillations that may undermine the solution in the presence of shocks. An artificial viscosity model is proposed, in complete analogy to those used in the finite element literature for high speed compressible flows. The resulting scheme possesses accuracy and stability properties very similar to its finite element counterpart, the Taylor-Galerkin FEM, and can be applied to a wide variety of problems of engineering interest, particularly when viscous and turbulent effects are of the utmost importance.

The outline of the paper is as follows. Section 2 presents a brief introduction to some meshless approximation techniques, with special emphasis on Moving Least Squares and Reproducing Kernel methods. The model equations and numerical formulations employed in this study are discussed in section 3. Finally, section 4 is devoted to numerical examples and other practical implementation issues.

2. MESHLESS APPROXIMATION: MOVING LEAST SQUARES

2.1. The idea of a meshfree 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, meshfree 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 succesive 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 computationally efficient algorithms, 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 moving least squares approximations. Further emphasis is placed on the particular technique used in this study, although the reader is referred to the classical meshfree 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 70's with the pioneering works in generalized finite differences and vortex particle methods [20],[21],[22]. However, the strongest influence upon the present trends is commonly attributed to early Smoothed Particle Hydrodynamics (SPH) formulations [23],[24],[25], 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 "meshfree shape functions"), developed using the concept of *kernel estimate*, which is inspired by the following property of the Dirac delta function

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

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

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

and its discrete SPH counterpart $\widehat{u}(\boldsymbol{x})$ is

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

where Ω is the problem domain, which is discretized into a set of n nodes or particles (used as quadrature points in (2)), $W(\boldsymbol{x}-\boldsymbol{x}_j,\rho)$ is a kernel (smoothing) function with compact support centered 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* in the meshfree literature, 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 funtions are most frequent kernels. In analogy with the finite element method, the approximation (3) could be cast in terms of SPH "shape functions", as

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

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

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

The gradient of $\widehat{u}(\boldsymbol{x})$ is evaluated as

$$\boldsymbol{\nabla}_{\boldsymbol{x}}\widehat{u}(\boldsymbol{x}) = \sum_{j=1}^{n} u_j \boldsymbol{\nabla}_{\boldsymbol{x}} N_j(\boldsymbol{x}) = \sum_{j=1}^{n} u_j \boldsymbol{\nabla}_{\boldsymbol{x}} W_j(\boldsymbol{x}) V_j \tag{6}$$

In practice, alternative expressions for $\boldsymbol{\nabla}_{\boldsymbol{x}}\widehat{u}(\boldsymbol{x})$ are frequent in the SPH literature to enforce conservation properties in the discrete equations. Higher order derivatives could be computed in a similar fashion. Note that the reconstructed values of $u(\boldsymbol{x})$ 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 mesh-based data structure (Figure 1).

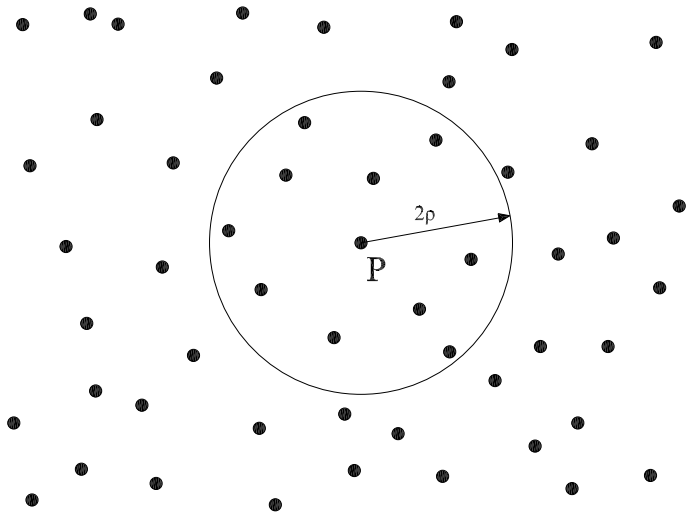

Figure 1. Meshfree approximation: general scheme. Support for reconstruction at P.

This basic approximation structure is retained in other improved interpolation schemes. In this study only Moving Least Squares (MLS) and Reproducing Kernel Particle (RKPM) methods are considered. Although different in their formulation, the resulting numerics are almost identical for both methods, and they can be presented within a common approach.

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

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

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

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

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

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

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

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

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

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

where the vector $\boldsymbol{u}_{\Omega_{\boldsymbol{x}}}$ contains certain nodal parameters of those particles in $\Omega_{\boldsymbol{x}}$, the discrete version of $\mathbf{M}$ is $\mathbf{M}(\boldsymbol{x}) = \mathbf{P}_{\Omega_{\boldsymbol{x}}}\mathbf{W}_{\mathbf{V}}(\boldsymbol{x})\mathbf{P}^T_{\Omega_{\boldsymbol{x}}}$, and matrices $\mathbf{P}_{\Omega_{\boldsymbol{x}}}$ and $\mathbf{W}_{\mathbf{V}}(\boldsymbol{x})$ can be obtained as

$$\boldsymbol{P}_{\Omega_{\boldsymbol{x}}} = \begin{pmatrix} \boldsymbol{p}(\boldsymbol{x}_1) & \boldsymbol{p}(\boldsymbol{x}_2) & \cdots & \boldsymbol{p}(\boldsymbol{x}_{n_{\boldsymbol{x}}}) \end{pmatrix} \tag{12}$$

$$\mathbf{W}_{\mathbf{V}}(\boldsymbol{x}) = diag\left\{W_i(\boldsymbol{x}-\boldsymbol{x}_i)V_i\right\}, \qquad i = 1,\ldots,n_{\boldsymbol{x}} \tag{13}$$

Complete details can be found in [26], [27], [28]. In the above equations, $n_{\boldsymbol{x}}$ denotes the total number of particles within the neighbourhood of point $\boldsymbol{x}$ and V_i and $\boldsymbol{x}_i$ are, respectively, the tributary volume (used as quadrature weight) and coordinates associated to particle i. Note that the tributary volumes of neighbouring particles are included in matrix $\mathbf{W}_{\mathbf{V}}$, obtaining an MLS version of the Reproducing Kernel Particle Method (the so-called MLSRKPM) [26]. Otherwise, we can use $\mathbf{W}$ instead of $\mathbf{W}_{\mathbf{V}}$

$$\mathbf{W}(\boldsymbol{x}) = diag\left\{W_i(\boldsymbol{x}-\boldsymbol{x}_i)\right\}, \qquad i = 1,\ldots,n_{\boldsymbol{x}} \tag{14}$$

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) in (7) the interpolation structure can be identified as

$$\widehat{u}(\boldsymbol{x}) = \boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{P}_{\Omega_{\boldsymbol{x}}}\boldsymbol{W}_V(\boldsymbol{x})\boldsymbol{u}_{\Omega_{\boldsymbol{x}}} = \boldsymbol{N}^T(\boldsymbol{x})\boldsymbol{u}_{\Omega_{\boldsymbol{x}}} \tag{15}$$

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

$$\boldsymbol{N}^T(\boldsymbol{x}) = \boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{P}_{\Omega_{\boldsymbol{x}}}\boldsymbol{W}_V(\boldsymbol{x}) \tag{16}$$

It is most frequent to use a scaled and locally defined polinomial basis, instead of the globally defined $\boldsymbol{p}(\boldsymbol{y})$. Thus, if the shape functions are to be evaluated at point $\boldsymbol{x}$, the basis would be of the form $\boldsymbol{p}(\frac{\boldsymbol{y}-\boldsymbol{x}}{\rho})$. The shape functions are, therefore, of the form

$$\boldsymbol{N}^T(\boldsymbol{x}) = \boldsymbol{p}^T(\boldsymbol{0})\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{p}^T(\boldsymbol{0})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{P}_{\Omega_{\boldsymbol{x}}}\boldsymbol{W}_V(\boldsymbol{x}) \tag{17}$$

Their first derivatives could be computed as

$$\frac{\partial \boldsymbol{N}^T(\boldsymbol{x})}{\partial x_i} = \frac{\partial \boldsymbol{p}^T(\boldsymbol{0})}{\partial x_i}\boldsymbol{C}(\boldsymbol{x}) + \boldsymbol{p}^T(\boldsymbol{0})\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x_i} \tag{18}$$

where

$$\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x_i} = \boldsymbol{C}(\boldsymbol{x})\boldsymbol{W}^{-1}(\boldsymbol{x})\frac{\partial \boldsymbol{W}(\boldsymbol{x})}{\partial x_i}\left(\boldsymbol{I} - \boldsymbol{P}^T_{\Omega_{\boldsymbol{x}}}\boldsymbol{C}(\boldsymbol{x})\right) \tag{19}$$

and expressions for higher order derivatives could be analogously obtained. Fast algorithms to perform these computations have been proposed [20],[29]. Examples of 2D basis functions are the quadratic polinomial basis

$$\boldsymbol{p}(\frac{\boldsymbol{y}-\boldsymbol{x}}{\rho}) = \left(1, z_1, z_2, z_1z_2, z_1^2, z_2^2\right) \tag{20}$$

which provides quadratic completeness, and the cubic basis

$$\boldsymbol{p}(\frac{\boldsymbol{y}-\boldsymbol{x}}{\rho}) = \left(1, z_1, z_2, z_1z_2, z_1^2, z_2^2, z_1^2z_2, z_1z_2^2, z_1^3, z_2^3\right) \tag{21}$$

for cubic completeness. In the above equations, $z_i = (y_i - x_i)/\rho$, and (x_1, x_2) and (y_1, y_2) are, respectively, the cartesian coordinates of $\boldsymbol{x}$ and $\boldsymbol{y}$. The concept of completeness alludes to the ability of the scheme to exactly reproduce polinomials and its derivatives. Thus, if the quadratic basis (20) is used, the approximation verifies

$$\sum_{j=1}^{n} x_j^a y_j^b N_j(\boldsymbol{x}) = x^a y^b, \qquad a \geq 0, b \geq 0, a+b \leq 2 \tag{22}$$

$$\sum_{j=1}^{n} x_j^a y_j^b \frac{\partial N_j(\boldsymbol{x})}{\partial x} = a x^{a-1} y^b, \qquad a \geq 0, b \geq 0, a+b \leq 2 \tag{23}$$

$$\sum_{j=1}^{n} x_j^a y_j^b \frac{\partial N_j(\boldsymbol{x})}{\partial y} = b x^{a} y^{b-1}, \qquad a \geq 0, b \geq 0, a+b \leq 2 \tag{24}$$

and so on. In general, any linear combination of the functions included in the basis $\boldsymbol{p}(\frac{\boldsymbol{y}-\boldsymbol{x}}{\rho})$ is exactly reproduced by the MLS approximation.

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

$$W_j(\boldsymbol{x}) = W(\boldsymbol{x}-\boldsymbol{x}_j, \rho) = \frac{\alpha}{\rho^\nu}\begin{cases} 1 - \frac{3}{2}s^2 + \frac{3}{4}s^3 & s \leq 1 \\ \frac{1}{4}(2-s)^3 & 1 < s \leq 2 \\ 0 & s > 2 \end{cases} \tag{25}$$

where $s = \dfrac{\|\boldsymbol{x} - \boldsymbol{x}_j\|}{\rho}$, ν is the number of dimensions and α takes the value $\frac{2}{3}, \frac{10}{7\pi}$ or $\frac{1}{\pi}$ in one, two or three dimensions, respectively. The coeficient α/ρ^ν is a scale factor neccesary only if non-corrected SPH interpolation is being used, to assure the normality property $\int W dV = 1$. We do not use it in our MLS computations. Anisotropic weightings (with rectangular instead of circular supports) for 2D/3D computations can be constructed as tensor-product of one-dimensional kernels as

$$W_j(\boldsymbol{x} - \boldsymbol{x}_j, \rho) = \prod_{n=1}^{\nu} W_j^n(x^n - x_j^n, \rho^n) \tag{26}$$

where x^n is the n-th coordinate of particle $\boldsymbol{x}$. In the above expression we let W_j^n and ρ^n (the one-dimensional kernel function and its caracteristic smoothing length) be different for each spatial dimension.

2.3. Computational aspects. Application to finite volume procedures.

The technique exposed above provides a general approximation framework which can be used in combination with existing finite volume methods. The field variables and their succesive derivatives are reconstructed at certain evaluation points (usually face midpoints or cell centers) using the cell-average information, customarily associated to the cell centers, which are taken here as the *nodes* or *particles* of the meshfree aproximation scheme. The reconstruction involves three major steps:

- Determine the "neighbourhood" of the evaluation point, i.e. which nodes (cell-centers) contribute to the reconstruction process.
- Compute the MLS shape functions and their required derivatives at the evaluation point, as exposed in section 2.2.
- Compute the approximate value of the field variables an their succesive derivatives using the general expressions

$$\widehat{u}(\boldsymbol{x}) = \sum_{j=1}^{n_{\boldsymbol{x}}} u_j N_j(\boldsymbol{x}), \qquad \frac{\partial^\alpha \widehat{u}(\boldsymbol{x})}{\partial x_k^\alpha} = \sum_{j=1}^{n_{\boldsymbol{x}}} u_j \frac{\partial^\alpha N_j(\boldsymbol{x})}{\partial x_k^\alpha} \tag{27}$$

  being $n_{\boldsymbol{x}}$ the number of neighbouring cell-centers whose field values $\{u_j\}$ are used in the reconstruction.

Note that, if a time marching scheme is to be used, the two first steps can be included in the preprocessing phase, as the MLS shape functions do not change in time for a fixed grid.

2.3.1. Cost and algorithmic complexity. The computation of the MLS shape functions is an expensive task if compared, for instance, to the FEM shape functions. However, and for a given finite volume mesh, where the information is stored at cell-centers, the shape functions and their derivatives need to be computed only once, at the beginning of the simulation. The time spent in such task is then negligible if compared to the whole simulation time. On the other hand, the evaluation of gradients and, eventually, other interpolated values at each time step becomes an extremely simple process, using the general expressions (27). In particular

$$\widehat{u}(\boldsymbol{x}) = \sum_{j=1}^{^n\boldsymbol{x}} u_j N_j(\boldsymbol{x}), \qquad \boldsymbol{\nabla}\widehat{u}(\boldsymbol{x}) = \sum_{j=1}^{^n\boldsymbol{x}} u_j \boldsymbol{\nabla} N_j(\boldsymbol{x}) \tag{28}$$

As mentioned before, the locations where the shape functions are evaluated (where the function or its succesive derivatives are reconstructed) depend on the particular finite volume algorithm (e.g. face midpoints, cell-centers...).

The evaluation of the shape functions at a given point involves a series of matrix operations, the most expensive of them being the inversion of the moment matrix $\boldsymbol{M}$. The size of this matrix is $m \times m$, where m is the dimension of the basis $\boldsymbol{p}(\frac{\boldsymbol{y}-\boldsymbol{x}}{\rho})$. For the quadratic basis (20), $m = 3$, $m = 6$ and $m = 10$ in one, two and three dimensions, respectively. Note that the size of $\boldsymbol{M}$ does not depend on the number of neighbours included in the computation of the shape functions.

2.3.2. Searching for neighbours. In most modern high-resolution schemes for unstructured meshes, high (second in most cases) order spatial accuracy is achieved by means of a multidimensional reconstruction of the primitive variables inside each cell. For example, a linear reconstruction could be devised by means of an extrapolation from the cell-averaged (center) value as

$$U(\boldsymbol{x}) = U_M + \boldsymbol{\nabla} U_M \cdot (\boldsymbol{x} - \boldsymbol{x}_M) \tag{29}$$

where $\boldsymbol{\nabla} U_M$ is a cell-centered gradient, usually obtained after a limiting process to avoid oscillations in the presence of shocks. Unlimited gradients are frequently computed using the Green's theorem applied to a scalar

$$\int\int_A \boldsymbol{\nabla} U dA = \oint_B U \boldsymbol{n} dB \tag{30}$$

and assuming that the gradient is constant in the cell

$$\boldsymbol{\nabla} U \approx \frac{1}{A} \oint_B U \boldsymbol{n} dB \tag{31}$$

In [30], several existing techniques for computing this averaged gradient are compared. The basic differences stem from their particular estimate of (31), according to a closed integration path B. The support of this path is called a *stencil*, and determines which (neighbour) cells are taken into account in the computation of (31). In order to obtain accurate gradients, and to alleviate the effects of grid sensitivity, the stencil should be properly wide. The accuracy of the reported schemes in the computation of gradients centered at the centroid of the cell is, at best, $O((\Delta x)^2, (\Delta y)^2)$ (exact linear gradients).

In the MLS approximation there is no reference to an underlying mesh topology, and the concept of "neigbourhood" is completely general. Straightforward choices include using the n nearest cell centers or even simply those within a certain distance from the evaluation point. More sophisticated algorithms (such as stencil-like neighbourhoods) are also perfectly applicable. The approximation can be made of virtually any order, just by adding more monomials to the basis. In this study only quadratic and cubic basis were used (providing exact linear and quadratic gradients, respectively, at *any* evaluation point).

Nevertheless, the cloud of neighbours must verify certain "minimum" requirements, which are mainly related to the inversion of the moment matrix $\boldsymbol{M}$. If the number of neighbours is less than m (the number of functions in the basis), $\boldsymbol{M}$ becomes singular, which implies that more than 6 neighbours are needed in 2D computations with the quadratic basis. In general, the approximation could be poor if $\boldsymbol{M}$ is highly ill-conditioned, so it is convenient to use a number of neighbours greater than the minimum, and with information coming from all possible directions. We have used $14 - -16$ neighbours in the 2D examples shown in this study.

2.3.3. 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 the shape functions are approximated by the first term in (18) as

$$\frac{\partial \boldsymbol{N}^T(\boldsymbol{x})}{\partial x_i} \approx \frac{\partial \boldsymbol{p}^T(\boldsymbol{0})}{\partial x_i}\boldsymbol{C}(\boldsymbol{x}) \tag{32}$$

It has been shown (see [31] and references therein) that the diffuse derivatives of a function $u(\boldsymbol{x})$, given by

$$\frac{\partial u(\boldsymbol{x})}{\partial x_i} \approx \sum_{j=1}^{n\boldsymbol{x}} u_j \frac{\partial N_j(\boldsymbol{x})}{\partial x_i} \tag{33}$$

converge at optimal rate to the exact derivatives. The same procedure can be extended to the succesive derivatives of $u(\boldsymbol{x})$. The fact that the order of the approximation is preserved and the much simpler numerics required in the computation of diffuse derivatives will be exploited later in this study.

3. NUMERICAL SCHEMES FOR THE SHALLOW WATER EQUATIONS

The spatial approximation described above will be used in combination with two different explicit numerical formulations. After introducing the mathematical model, a second-order accurate in time Lax-Wendroff scheme and a suitable shock-capturing viscosity model are presented in section 3.2, whose low-dissipation properties can be fully exploited on unstructured grids using the MLS approximation. In section 3.3, second and third-order-reconstruction high-resolution schemes are developed using Roe's flux difference splitting and a multistage Runge-Kutta time integrator.

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]

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}_x}{\partial x} + \frac{\partial \boldsymbol{F}_y}{\partial y} = \boldsymbol{R}_s + \frac{\partial \boldsymbol{R}_{dx}}{\partial x} + \frac{\partial \boldsymbol{R}_{dy}}{\partial y} \tag{34}$$

being

$$\boldsymbol{U} = \begin{pmatrix} h \\ hu_x \\ hu_y \end{pmatrix} \tag{35}$$

$$\boldsymbol{F}_x = \begin{pmatrix} hu_x \\ hu_x^2 + \frac{1}{2}g(h^2 - H^2) \\ hu_x u_y \end{pmatrix} \qquad \boldsymbol{F}_y = \begin{pmatrix} hu_y \\ hu_x u_y \\ hu_y^2 + \frac{1}{2}g(h^2 - H^2) \end{pmatrix} \tag{36}$$

$$\boldsymbol{R}_{dx} = \begin{pmatrix} 0 \\ 2\nu h \dfrac{\partial u_x}{\partial x} \\ \nu h \left(\dfrac{\partial u_y}{\partial x} + \dfrac{\partial u_x}{\partial y} \right) \end{pmatrix} \qquad \boldsymbol{R}_{dy} = \begin{pmatrix} 0 \\ \nu h \left(\dfrac{\partial u_y}{\partial x} + \dfrac{\partial u_x}{\partial y} \right) \\ 2\nu h \dfrac{\partial u_y}{\partial y} \end{pmatrix} \tag{37}$$

$$\boldsymbol{R}_S = \begin{pmatrix} 0 \\ g(h-H)\dfrac{\partial H}{\partial x} - \dfrac{gn^2|\boldsymbol{u}|u_x}{h^{1/3}} \\ g(h-H)\dfrac{\partial H}{\partial y} - \dfrac{gn^2|\boldsymbol{u}|u_y}{h^{1/3}} \end{pmatrix} \tag{38}$$

In the above, $\boldsymbol{u} = (u_x, u_y)$ 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. The 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

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^n + \Delta t \left(\frac{\partial \boldsymbol{U}}{\partial t} \right)^n + \frac{\Delta t^2}{2} \left(\frac{\partial^2 \boldsymbol{U}}{\partial t^2} \right)^n \tag{39}$$

The time derivatives are expressed in terms of spatial derivatives using the original equation (34), to yield [2]

$$\begin{aligned} \boldsymbol{U}^{n+1} \;=\; & \boldsymbol{U}^n + \Delta t \left(\boldsymbol{R}_s + \frac{\partial \boldsymbol{R}_{di}}{\partial x_i} - \frac{\partial \boldsymbol{F}_i}{\partial x_i} \right)^n + \\ & + \frac{\Delta t^2}{2} \left\{ \boldsymbol{G} \left(\boldsymbol{R}_s - \frac{\partial \boldsymbol{F}_i}{\partial x_i} \right) - \frac{\partial}{\partial x_i} \left[\boldsymbol{A}_i \left(\boldsymbol{R}_s - \frac{\partial \boldsymbol{F}_j}{\partial x_j} \right) \right] \right\}^n \end{aligned} \tag{40}$$

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

$$\frac{\partial \boldsymbol{F}_i}{\partial x_i} = \frac{\partial \boldsymbol{F}_x}{\partial x} + \frac{\partial \boldsymbol{F}_y}{\partial y}, \qquad \frac{\partial \boldsymbol{R}_{di}}{\partial x_i} = \frac{\partial \boldsymbol{R}_{dx}}{\partial x} + \frac{\partial \boldsymbol{R}_{dy}}{\partial y} \tag{41}$$

$$\frac{\partial}{\partial x_i} \left[\boldsymbol{A}_i \left(\boldsymbol{R}_s - \frac{\partial \boldsymbol{F}_j}{\partial x_j} \right) \right] = \frac{\partial}{\partial x} \left[\boldsymbol{A}_x \left(\boldsymbol{R}_s - \frac{\partial \boldsymbol{F}_j}{\partial x_j} \right) \right] + \frac{\partial}{\partial y} \left[\boldsymbol{A}_y \left(\boldsymbol{R}_s - \frac{\partial \boldsymbol{F}_j}{\partial x_j} \right) \right] \tag{42}$$

has been used for simplicity, and

$$\boldsymbol{A}_x = \frac{\partial \boldsymbol{F}_x}{\partial \boldsymbol{U}}, \qquad \boldsymbol{A}_y = \frac{\partial \boldsymbol{F}_y}{\partial \boldsymbol{U}}, \qquad \boldsymbol{G} = \frac{\partial \boldsymbol{R}_s}{\partial \boldsymbol{U}} \tag{43}$$

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

$$\boldsymbol{A}_x = \begin{pmatrix} 0 & 1 & 0 \\ -u_x^2 + gh & 2u_x & 0 \\ -u_x u_y & u_y & u_x \end{pmatrix}, \qquad \boldsymbol{A}_y = \begin{pmatrix} 0 & 0 & 1 \\ -u_x u_y & u_y & u_x \\ -u_y^2 + gh & 0 & 2u_y \end{pmatrix} \tag{44}$$

The integration of (40) over a cell (control volume) Ω yields:

$$\begin{aligned} \int_\Omega \Delta \boldsymbol{U} d\Omega \;\; = \;\; & \Delta t \int_\Omega \left(\boldsymbol{R}_s + \frac{\partial \boldsymbol{R}_{di}}{\partial x_i} - \frac{\partial \boldsymbol{F}_i}{\partial x_i} \right)^n d\Omega + \\ & + \frac{\Delta t^2}{2} \int_\Omega \left\{ \boldsymbol{G} \left(\boldsymbol{R}_s - \frac{\partial \boldsymbol{F}_i}{\partial x_i} \right) - \frac{\partial}{\partial x_i} \left[\boldsymbol{A}_i \left(\boldsymbol{R}_s - \frac{\partial \boldsymbol{F}_j}{\partial x_j} \right) \right] \right\}^n d\Omega \end{aligned} \tag{45}$$

Making use of the divergence theorem, and rearranging,

$$\begin{aligned} \int_\Omega \Delta \boldsymbol{U} d\Omega \;\; = \;\; & \Delta t \int_\Gamma \left(\boldsymbol{\mathcal{R}}_d - \boldsymbol{\mathcal{F}} \right)^n \cdot \boldsymbol{n} \; d\Gamma - \frac{\Delta t^2}{2} \int_\Gamma \boldsymbol{\mathcal{S}}^n \cdot \boldsymbol{n} \; d\Gamma + \\ & + \Delta t \int_\Omega \boldsymbol{R}_s^n d\Omega + \frac{\Delta t^2}{2} \int_\Omega \left[\boldsymbol{G} \left(\boldsymbol{R}_s - \frac{\partial \boldsymbol{F}_i}{\partial x_i} \right) \right]^n d\Omega \end{aligned} \tag{46}$$

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

$$\boldsymbol{\mathcal{F}} = \left(\boldsymbol{F}_x, \boldsymbol{F}_y \right), \quad \boldsymbol{\mathcal{R}}_d = \left(\boldsymbol{R}_{dx}, \boldsymbol{R}_{dy} \right), \quad \boldsymbol{\mathcal{S}} = \left(\boldsymbol{A}_x \left(\boldsymbol{R}_s - \frac{\partial \boldsymbol{F}_j}{\partial x_j} \right), \boldsymbol{A}_y \left(\boldsymbol{R}_s - \frac{\partial \boldsymbol{F}_j}{\partial x_j} \right) \right) \tag{47}$$

In the absence of source terms, $\boldsymbol{R}_s = \boldsymbol{0}$ and equation (46) reduces to

$$\int_\Omega \Delta \boldsymbol{U} d\Omega = \Delta t \int_\Gamma \left(\boldsymbol{\mathcal{R}}_d - \boldsymbol{\mathcal{F}} \right)^n \cdot \boldsymbol{n} \; d\Gamma + \frac{\Delta t^2}{2} \int_\Gamma \left(\boldsymbol{A}_x \frac{\partial \boldsymbol{F}_j}{\partial x_j} n_x + \boldsymbol{A}_y \frac{\partial \boldsymbol{F}_j}{\partial x_j} n_y \right)^n d\Gamma \tag{48}$$

Adopting a standard finite volume discretization for (46), surface integrals are computed using the centerpoints of each cell, where the primitive variables are stored, and boundary integrals are evaluated at certain representative points (e.g. at the center of each face). Thus, the discrete equation for each cell I results

$$\begin{aligned} \Delta \boldsymbol{U}_I A_I = \Delta t \sum_{iface}^{nface} \left(\boldsymbol{\mathcal{R}}_d - \boldsymbol{\mathcal{F}} \right)^n_{iface} \cdot \boldsymbol{n}_{iface} L_{iface} - \frac{\Delta t^2}{2} \sum_{iface}^{nface} \boldsymbol{\mathcal{S}}^n_{iface} \cdot \boldsymbol{n}_{iface} L_{iface} + \\ + \Delta t \boldsymbol{R}^n_{sI} A_I + \frac{\Delta t^2}{2} \left[\boldsymbol{G} \left(\boldsymbol{R}_s - \frac{\partial \boldsymbol{F}_j}{\partial x_j} \right) \right]^n_I A_I \end{aligned} \tag{49}$$

where A_I is the area of cell I, $nface_I$ the number of cell faces, L_{iface} the longitude of face $iface$ and $\boldsymbol{U}_I$ the average value of $\boldsymbol{U}$ over the cell I (associated to the cell center).

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 $\widehat{\phi}(\boldsymbol{x})$ of a function $\phi(\boldsymbol{x})$, given by

$$\widehat{\phi}(\boldsymbol{x}) = \sum_{j=1}^{n_{\boldsymbol{x}}} \phi_j N_j(\boldsymbol{x}) \tag{50}$$

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

$$\boldsymbol{\nabla}\widehat{\phi}(\boldsymbol{x}) = \sum_{j=1}^{n_{\boldsymbol{x}}} \phi_j \boldsymbol{\nabla} N_j(\boldsymbol{x}) \tag{51}$$

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

$$\boldsymbol{F}_x(\boldsymbol{x}_{iface}) = \sum_{j=1}^{n_i} \boldsymbol{F}_{xj} N_j(\boldsymbol{x}_{iface}), \qquad \boldsymbol{F}_y(\boldsymbol{x}_{iface}) = \sum_{j=1}^{n_i} \boldsymbol{F}_{yj} N_j(\boldsymbol{x}_{iface}) \tag{52}$$

where, for simplicity, $n_i = n_{\boldsymbol{x}_{iface}}$ denotes the number of cell-centers taken into account in the reconstruction process. Similarly, other required entities are interpolated as

$$\boldsymbol{R}_s(\boldsymbol{x}_{iface}) = \sum_{j=1}^{n_i} \boldsymbol{R}_{sj} N_j(\boldsymbol{x}_{iface}), \quad \left.\frac{\partial \boldsymbol{F}_k}{\partial x_k}\right|_{\boldsymbol{x}_{iface}} = \sum_{j=1}^{n_i} (\boldsymbol{F}_{xj}, \boldsymbol{F}_{yj}) \cdot \boldsymbol{\nabla} N_j(\boldsymbol{x}_{iface}) \tag{53}$$

$$\boldsymbol{A}_x(\boldsymbol{x}_{iface}) = \sum_{j=1}^{n_i} \boldsymbol{A}_{xj} N_j(\boldsymbol{x}_{iface}), \quad \boldsymbol{A}_y(\boldsymbol{x}_{iface}) = \sum_{j=1}^{n_i} \boldsymbol{A}_{yj} N_j(\boldsymbol{x}_{iface}), \tag{54}$$

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

$$\boldsymbol{\nabla} u_x(\boldsymbol{x}_{iface}) = \sum_{j=1}^{n_i} u_{xj} \boldsymbol{\nabla} N_j(\boldsymbol{x}_{iface}), \quad \boldsymbol{\nabla} u_y(\boldsymbol{x}_{iface}) = \sum_{j=1}^{n_i} u_{yj} \boldsymbol{\nabla} N_j(\boldsymbol{x}_{iface}), \tag{55}$$

or, in compact form,

$$\boldsymbol{\nabla}\boldsymbol{u}(\boldsymbol{x}_{iface}) = \sum_{j=1}^{n_i} \boldsymbol{u}_j \otimes \boldsymbol{\nabla} N_j(\boldsymbol{x}_{iface}), \tag{56}$$

In general, any variable and its gradient can be computed using equations (50) and (51) and the information stored at the cell centers.

3.2.2. Accuracy and implementation. It has been shown [28] that, for a p^{th} order MLS fit (p^{th} order complete polynomial basis) and general, irregularly spaced points, the nominal order of accuracy for the approximation of a q^{th} order gradient is roughly $(p - q + 1)$. Thus, using the quadratic basis (20), the approximation of the convective fluxes at cell faces given by (52) is third order accurate and the diffusive fluxes computed from the velocity gradient (56) are second order accurate. The overall accuracy of the resulting algorithm is, therefore, second order, and one integration point per cell face (the midpoint) is sufficient to compute the boundary integrals (Figure 2 left).

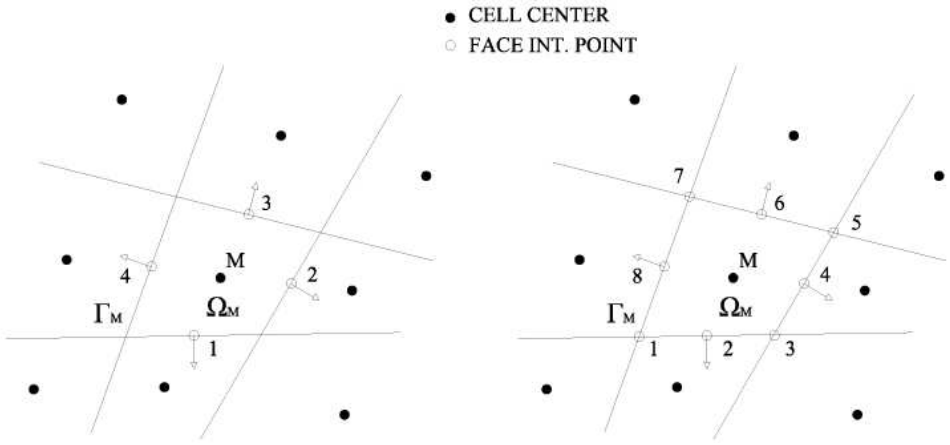

Figure 2. Generic quadrilateral finite volumes.

Third order spatial accuracy could be obtained using the cubic basis (21). In this case a higher order quadrature rule is required for the evaluation of boundary integrals. In 2D applications, the arrangement shown in Figure 2 (right) could be used in combination with the Simpson rule, which is fourth order accurate. Thus, the flux of a vector $\boldsymbol{f}$ through the face 123, F_{123}, would read

$$F_{123} = \int_{\Gamma_{123}} \boldsymbol{f} \cdot \boldsymbol{n} \, d\Gamma \approx \frac{L_{123}}{6} \left[(\boldsymbol{f} \cdot \boldsymbol{n})_1 + 4(\boldsymbol{f} \cdot \boldsymbol{n})_2 + (\boldsymbol{f} \cdot \boldsymbol{n})_3 \right] \tag{57}$$

and so on. However, the cubic MLS scheme requires a high number of neighbours per evaluation point and, therefore, the quadratic MLS approximation (overall second order algorithm) seems to yield the best ratio accuracy/efficiency.

3.2.3. Artificial viscosity. 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 [32]. 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, $\boldsymbol{R}_{dx}^{SC}$ and $\boldsymbol{R}_{dy}^{SC}$, to the right hand side of (34) as

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}_x}{\partial x} + \frac{\partial \boldsymbol{F}_y}{\partial y} = \boldsymbol{R}_s + \frac{\partial (\boldsymbol{R}_{dx} + \boldsymbol{R}_{dx}^{SC})}{\partial x} + \frac{\partial (\boldsymbol{R}_{dy} + \boldsymbol{R}_{dy}^{SC})}{\partial y} \tag{58}$$

where

$$\boldsymbol{R}_{dx}^{SC} = \begin{pmatrix} \nu_h^{SC} \dfrac{\partial h}{\partial x} \\ 2\nu_V^{SC} h \dfrac{\partial u_x}{\partial x} \\ \nu_V^{SC} h \left(\dfrac{\partial u_y}{\partial x} + \dfrac{\partial u_x}{\partial y} \right) \end{pmatrix} \qquad \boldsymbol{R}_{dy}^{SC} = \begin{pmatrix} \nu_h^{SC} \dfrac{\partial h}{\partial y} \\ \nu_V^{SC} h \left(\dfrac{\partial u_y}{\partial x} + \dfrac{\partial u_x}{\partial y} \right) \\ 2\nu_V^{SC} h \dfrac{\partial u_y}{\partial y} \end{pmatrix} \tag{59}$$

and the shock capturing viscosities

$$\nu_h^{SC} = C_h \epsilon^2 \frac{|\boldsymbol{u}| + c}{h} |\boldsymbol{\nabla} h|, \qquad \nu_V^{SC} = C_V \epsilon^2 \frac{|\boldsymbol{u}| + c}{h} |\boldsymbol{\nabla} h| \tag{60}$$

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, $\boldsymbol{u}$ and $\boldsymbol{\nabla} h$, is computed at cell faces using the MLS approximation. In the case of transcritical flows, an entropy fix scheme should also be included in this formulation [33].

3.3. Upwind schemes: high-order reconstruction.

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 [34]. Unfortunately, upwind schemes are frequently associated to an excessive numerical dissipation in more general flows [13], [14], [15], being rather widely regarded as "specialized" methods, and not well suited for more general flows [34].

A quite popular approach to reduce the amount of numerical dissipation of the upwind scheme is the development of a higher-order 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 [30]. It is in this context that the interesting features of meshfree interpolation schemes such as MLS, particularly well suited to provide accurate derivatives on irregularly spaced points [28], can be exploited.

This section presents a low-dissipative upwind scheme, based on Roe's flux difference splitting, applied to the set of shallow water equations on unstructured meshes. Second and third-order-reconstruction schemes 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 (34)

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}_x}{\partial x} + \frac{\partial \boldsymbol{F}_y}{\partial y} = \boldsymbol{R}_s + \frac{\partial \boldsymbol{R}_{dx}}{\partial x} + \frac{\partial \boldsymbol{R}_{dy}}{\partial y} \tag{61}$$

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

$$\int_\Omega \frac{\partial \boldsymbol{U}}{\partial t} d\Omega = \int_\Gamma (\boldsymbol{\mathcal{R}}_d - \boldsymbol{\mathcal{F}}) \cdot \boldsymbol{n} \, d\Gamma + \int_\Omega \boldsymbol{R}_s d\Omega \tag{62}$$

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

$$\boldsymbol{\mathcal{F}} = (\boldsymbol{F}_x, \boldsymbol{F}_y), \quad \boldsymbol{\mathcal{R}}_d = (\boldsymbol{R}_{dx}, \boldsymbol{R}_{dy}) \tag{63}$$

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

$$\frac{\partial \boldsymbol{U}_I}{\partial t} = \frac{1}{A_I} \sum_{iface=1}^{nface_I} [(\boldsymbol{\mathcal{R}}_d - \boldsymbol{\mathcal{F}}) \cdot \boldsymbol{n}]_{iface} \, L_{iface} + \boldsymbol{R}_{sI} \tag{64}$$

where A_I is the area of cell I, $nface_I$ the number of cell faces, L_{iface} the longitude of face $iface$ and $\boldsymbol{U}_I$ the average value of $\boldsymbol{U}$ over the cell I (customarily associated to the cell center). Standard ODE solvers can be applied to (64). We have used Shu's third order Runge-Kutta algorithm, which is compatible with TVD, TVB and ENO schemes [35]

$$\begin{aligned} U^1 &= U^n + \Delta t L(U^n) \\ U^2 &= \frac{3}{4}U^n + \frac{1}{4}U^1 + \frac{1}{4}\Delta t L(U^1) \\ U^{n+1} &= \frac{1}{3}U^n + \frac{2}{3}U^2 + \frac{2}{3}\Delta t L(U^2) \end{aligned} \tag{65}$$

In the above equations, the operator $L(\cdot)$ represents the right hand side of (64). The diffusive fluxes are evaluated using the same procedure as in the Lax-Wendroff scheme, computing velocity gradients at cell faces by means of the MLS approximation. The numerical convective fluxes are obtained using Roe's flux difference splitting [19] . For this purpose, left ($\boldsymbol{U}^-$) and right ($\boldsymbol{U}^+$) states are defined on each face. The numerical flux is then computed as [36]

$$(\boldsymbol{F}_x, \boldsymbol{F}_y) \cdot \boldsymbol{n} = \frac{1}{2}\left[\left(\boldsymbol{F}_x\left(\boldsymbol{U}^-\right), \boldsymbol{F}_y\left(\boldsymbol{U}^-\right)\right) + \left(\boldsymbol{F}_x\left(\boldsymbol{U}^+\right), \boldsymbol{F}_y\left(\boldsymbol{U}^+\right)\right)\right] \cdot \boldsymbol{n} - \frac{1}{2}|\widetilde{\boldsymbol{J}}|\left(\boldsymbol{U}^+ - \boldsymbol{U}^-\right) \tag{66}$$

where $\widetilde{\boldsymbol{J}}\left(\boldsymbol{U}^+, \boldsymbol{U}^-\right)$ is an approximate flux jacobian, satisfying certain matrix properties [36]. Equation (66) can be also written as [37]

$$(\boldsymbol{F}_x, \boldsymbol{F}_y) \cdot \boldsymbol{n} = \frac{1}{2}\left[\left(\boldsymbol{F}_x\left(\boldsymbol{U}^-\right), \boldsymbol{F}_y\left(\boldsymbol{U}^-\right)\right) + \left(\boldsymbol{F}_x\left(\boldsymbol{U}^+\right), \boldsymbol{F}_y\left(\boldsymbol{U}^+\right)\right)\right] \cdot \boldsymbol{n} - \frac{1}{2}\sum_{k=1}^{3} \widetilde{\alpha}_k |\widetilde{\lambda}_k| \widetilde{\boldsymbol{r}}_k \tag{67}$$

where $\{\widetilde{\lambda}_k, \ k = 1,3\}$ and $\{\widetilde{\boldsymbol{r}}_k, \ k = 1,3\}$ are, respectively, the eigenvalues and eigenvectors of the approximate jacobian $\widetilde{\boldsymbol{J}}\left(\boldsymbol{U}^+, \boldsymbol{U}^-\right)$

$$\widetilde{\lambda}_1 = \widetilde{u}_x n_x + \widetilde{u}_y n_y + \widetilde{c}, \quad \widetilde{\lambda}_2 = \widetilde{u}_x n_x + \widetilde{u}_y n_y, \quad \widetilde{\lambda}_3 = \widetilde{u}_x n_x + \widetilde{u}_y n_y - \widetilde{c} \tag{68}$$

$$\widetilde{\boldsymbol{r}}_1 = \begin{pmatrix} 1 \\ \widetilde{u}_x + \widetilde{c} n_x \\ \widetilde{u}_y + \widetilde{c} n_y \end{pmatrix}, \quad \widetilde{\boldsymbol{r}}_2 = \begin{pmatrix} 0 \\ -\widetilde{c} n_y \\ \widetilde{c} n_x \end{pmatrix}, \quad \widetilde{\boldsymbol{r}}_2 = \begin{pmatrix} 1 \\ \widetilde{u}_x - \widetilde{c} n_x \\ \widetilde{u}_y - \widetilde{c} n_y \end{pmatrix} \tag{69}$$

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

$$\begin{aligned} \widetilde{\alpha}_1 &= \frac{\Delta h}{2} + \frac{1}{2\widetilde{c}} \left(\Delta \left(h u_x \right) n_x + \Delta \left(h u_y \right) n_y - \left(\widetilde{u}_x n_x + \widetilde{u}_y n_y \right) \Delta h \right) \\ \widetilde{\alpha}_2 &= \frac{1}{\widetilde{c}} \left(\left(\Delta \left(h u_y \right) - \widetilde{u}_y \Delta \left(h \right) \right) n_x - \left(\Delta \left(h u_x \right) - \widetilde{u}_x \Delta \left(h \right) \right) n_y \right) \\ \widetilde{\alpha}_3 &= \frac{\Delta h}{2} - \frac{1}{2\widetilde{c}} \left(\Delta \left(h u_x \right) n_x + \Delta \left(h u_y \right) n_y - \left(\widetilde{u}_x n_x + \widetilde{u}_y n_y \right) \Delta h \right) \end{aligned} \tag{70}$$

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

$$\widetilde{u}_x = \frac{u_x^+ \sqrt{h^+} + u_x^- \sqrt{h^-}}{\sqrt{h^+} + \sqrt{h^-}}, \quad \widetilde{u}_y = \frac{u_y^+ \sqrt{h^+} + u_y^- \sqrt{h^-}}{\sqrt{h^+} + \sqrt{h^-}}, \quad \widetilde{c} = \sqrt{g \left(h^+ + h^- \right) / 2} \tag{71}$$

A first order scheme is obtained by setting $\boldsymbol{U}^-$ and $\boldsymbol{U}^+$ to be the variables at the left and right cell centers. 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 *higher order schemes*) can be devised by choosing "better" values for the left and right states.

3.3.1. Higher order schemes: reconstruction and limiting. As mentioned before, the amount of artificial dissipation can be reduced by expanding the piecewise constant representation which the finite volume philosophy entails. This can be addressed by (astutely) choosing "closer" values for $\boldsymbol{U}^-$ and $\boldsymbol{U}^+$, and several extrapolation procedures are feasible. In this study, a multidimensional reconstruction of the field variables inside each cell is obtained by means of Taylor series expansions. Unfortunately, these higher-order extensions of the basic linear algorithm are not free from oscillations and some kind of limiting procedure is required in the presence of shocks.

Second order accuracy is achieved by means of a linear reconstruction inside left and right cells, as

$$\boldsymbol{U}^-(\boldsymbol{x}) = \boldsymbol{U}_{M^-} + \boldsymbol{\nabla U}_{M^-} (\boldsymbol{x} - \boldsymbol{x}_{M^-}) \tag{72}$$

$$\boldsymbol{U}^+(\boldsymbol{x}) = \boldsymbol{U}_{M^+} + \boldsymbol{\nabla U}_{M^+} (\boldsymbol{x} - \boldsymbol{x}_{M^+}) \tag{73}$$

where $\boldsymbol{U}_{M^-}$ and $\boldsymbol{U}_{M^+}$ stand for left and right cell-averaged (center) values of the variables, $\boldsymbol{x}_{M^-}$ and $\boldsymbol{x}_{M^+}$ are the spatial coordinates of left and right cell centerpoints and $\boldsymbol{\nabla U}_{M^-}$ and $\boldsymbol{\nabla U}_{M^+}$ are cell-centered gradients. These gradients $\boldsymbol{\nabla U}_{M^-}$ and $\boldsymbol{\nabla U}_{M^+}$ are assumed to be constant

on each cell and, therefore, the reconstructed variables are discontinuous across interfaces (i.e. we still have a Riemann problem on each face).

In the presence of shocks, limiting is applied to the cell-centered gradients in (72)–(73). The choice of adequate multidimensional limiters is critical in order to achieve accurate and non-oscillatory shock capturing algorithms, and quite frequently related to the type of finite volume considered. In this study a cell-centered approach with quadrilateral volumes, as shown in Figure 2 (left), has been adopted. Limited gradients are constructed in a similar fashion to that proposed by Van Rosendale [30], adapted to quadrilateral cells, where the cell-centered gradient $\boldsymbol{\nabla} U_M$ is obtained as a weighted average of the unlimited face gradients

$$\boldsymbol{\nabla} U_M = \omega_1 \boldsymbol{\nabla} U_1 + \omega_2 \boldsymbol{\nabla} U_2 + \omega_3 \boldsymbol{\nabla} U_3 + \omega_4 \boldsymbol{\nabla} U_4 \tag{74}$$

where the weights ω_1, ω_2, ω_3 and ω_4 are given by the multidimensional limiter function [30]. Using an extension of the Van Albada limiter, we set

$$\omega_1 (g_1, g_2, g_3, g_4) = \frac{g_2 g_3 g_4 + \epsilon^3}{g_1 g_2 g_3 + g_1 g_2 g_4 + g_1 g_3 g_4 + g_2 g_3 g_4 + 4\epsilon^3} \tag{75}$$

$$\omega_2 (g_1, g_2, g_3, g_4) = \frac{g_1 g_3 g_4 + \epsilon^3}{g_1 g_2 g_3 + g_1 g_2 g_4 + g_1 g_3 g_4 + g_2 g_3 g_4 + 4\epsilon^3} \tag{76}$$

$$\omega_3 (g_1, g_2, g_3, g_4) = \frac{g_1 g_2 g_4 + \epsilon^3}{g_1 g_2 g_3 + g_1 g_2 g_4 + g_1 g_3 g_4 + g_2 g_3 g_4 + 4\epsilon^3} \tag{77}$$

$$\omega_4 (g_1, g_2, g_3, g_4) = \frac{g_1 g_2 g_3 + \epsilon^3}{g_1 g_2 g_3 + g_1 g_2 g_4 + g_1 g_3 g_4 + g_2 g_3 g_4 + 4\epsilon^3} \tag{78}$$

where g_1, g_2, g_3 and g_4 are functions of the unlimited face gradients and ϵ is a small number, introduced to prevent division by zero. Two different choices for $\{g_k,\ k = 1, \ldots, 4\}$ were compared, namely

$$g_k = \|\boldsymbol{\nabla} U_k\|^2 \tag{79}$$

and

$$g_k = \|\boldsymbol{\nabla} U_k\| \tag{80}$$

The first expression penalizes high gradients more intensely, yielding a more dissipative limiting, whereas the second yields a less strict limiting but not well suited for very strong shocks. The unlimited gradients $\{\boldsymbol{\nabla} U_k,\ k = 1, \ldots, 4\}$ are computed at each face midpoint using the MLS approximation

$$\boldsymbol{\nabla} U_k = \sum_{j=1}^{n_{\boldsymbol{x}_k}} U_j \boldsymbol{\nabla} N_j(\boldsymbol{x}_k) \tag{81}$$

where the U_j's stand for variables at the $n_{\boldsymbol{x}_k}$ "neighbour" cell centers, and $\boldsymbol{x}_k$ denotes the cartesian coordinates of the face midpoint k. Note that, following this scheme, the diffusive fluxes and the gradient computation necessary for the reconstruction (72)–(73) are treated

within a unified approach (i.e. only the derivatives of the MLS shape functions at the face midpoints are needed).

The numerical dissipation can be further reduced by considering higher-order Taylor series expansions. A third order reconstruction is developed using cell-centered second derivatives to perform a quadratic expansion of the field variables inside each cell, as

$$U(\boldsymbol{x}) = U_M + \boldsymbol{\nabla} U_M(\boldsymbol{x} - \boldsymbol{x}_M) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_M)^T \boldsymbol{H}_M(\boldsymbol{x} - \boldsymbol{x}_M) - \frac{1}{2}\left[I_{xx}\frac{\partial^2 U}{\partial x^2} + 2I_{xy}\frac{\partial^2 U}{\partial x \partial y} + I_{yy}\frac{\partial^2 U}{\partial y^2}\right] \tag{82}$$

where $\boldsymbol{H}_M$ is the cell-centered hessian matrix and

$$I_{xx} = \int_\Omega (x - x_M)^2 d\Omega, \quad I_{xy} = \int_\Omega (x - x_M)(y - y_M) d\Omega, \quad I_{yy} = \int_\Omega (y - y_M)^2 d\Omega \tag{83}$$

The above integrals can be easily computed for quadrilateral and triangular cells. The last term in (82) has been added to ensure that the average value of the reconstructed variables over cell Ω_i is the center value U_{M_i}, i.e.

$$\frac{1}{A_i}\int_{\boldsymbol{x}\in\Omega_i} U(\boldsymbol{x})\, d\Omega = U_{M_i} \tag{84}$$

Note that the introduction of the terms (83) does not reduce the order of the approximation given by (82). For the second order derivatives we have used the same limiting procedure presented above for the first order gradients, (74)–(78). The development of better limiters for this third-order reconstruction is currently in progress.

3.3.2. Dissipation additions. Some additional dissipation can be necessary in certain applications. This procedure is frequent in the context of flux difference splitting schemes for gas dynamics in order to break expansion shocks and as a cure for the "carbuncle" phenomenon in supersonic flow past blunt bodies. Yee proposed a formula for Roe's Riemann solver where the eigenvalues λ_k are replaced by $Q(\lambda_k, \delta_k)$, as [38]

$$Q_k(\lambda_k, \delta_k) = \begin{cases} \lambda_k & |\lambda_k| < \delta_k \\ \frac{1}{2}\left[sign(\lambda_k)\frac{\lambda_k^2 + \delta_k^2}{\delta_k} + \lambda_k\right] & |\lambda_k| \geq \delta_k \end{cases} \tag{85}$$

where

$$\delta_k = \delta_k^* \left(|v_n| + c\right) \tag{86}$$

being v_n the velocity normal to the grid face and c the face wave speed. The parameters δ_k^* control the amount of artificial dissipation. This formula has been found excessively dissipative for the v_n waves [38], and thus not suitable for viscous flow. Inspired by the work of Peery and Imlay, Lin [38] proposed a modification for structured meshes which uses the second difference of the pressure to tune the dissipation. For the $v_n + c$ and $v_n - c$ waves,

$$\delta_k = \left(|v_n| + c\right)\left(k_1 + k_2 k_p\right) \tag{87}$$

instead of (86), whereas for the linear waves

$$\delta_k = (|v_n| + c)\,(k_3 k_p) \tag{88}$$

The coefficient k_p is defined at each cell (i,j) as

$$(k_p)_{i,j} = \frac{1}{2}\left(\left|\frac{P_{i+1,j} - 2P_{i,j} + P_{i-1,j}}{P_{i+1,j} + 2P_{i,j} + P_{i-1,j}}\right| + \left|\frac{P_{i,j+1} - 2P_{i,j} + P_{i,j-1}}{P_{i,j+1} + 2P_{i,j} + P_{i,j-1}}\right|\right) \tag{89}$$

where P denotes pressure. We propose a different expression for k_p, as

$$k_p = \epsilon \frac{|\boldsymbol{\nabla} P|}{P} \tag{90}$$

where ϵ is a certain characteristic length. For shallow water problems, we write

$$k_p = \epsilon \frac{|\boldsymbol{\nabla} h|}{h} \tag{91}$$

This parameter is defined in a continuous fashion (in practice it is only required at each interface) and computed using the MLS shape functions, which makes this procedure suitable for unstructured meshes. In this study we define the characteristic length ϵ as

$$\epsilon = \frac{1}{2}\left(\sqrt{A^+} + \sqrt{A^-}\right) \tag{92}$$

being A^+ and A^- the area of the cells on each side of the interface.

3.3.3. Computational aspects: diffuse derivatives. The evaluation of full second order derivatives of the MLS shape functions is a tedious and computationally expensive task. Instead, we have used the concept of diffuse derivative to approximate the second order derivatives required in the third order reconstruction (82). Thus, we write

$$\frac{\partial^2 U(\boldsymbol{x})}{\partial x_\alpha x_\beta} \approx \sum_{j=1}^{n_{\boldsymbol{x}}} U_j \frac{\partial^2 N_j(\boldsymbol{x})}{\partial x_\alpha x_\beta} \tag{93}$$

where the second derivatives of the shape functions are approximated by

$$\frac{\partial^2 \boldsymbol{N}^T(\boldsymbol{x})}{\partial x_\alpha x_\beta} \approx \frac{\partial^2 \boldsymbol{p}^T(\boldsymbol{0})}{\partial x_\alpha x_\beta} \boldsymbol{C}(\boldsymbol{x}) \tag{94}$$

and straightforwardly computed once the matrix $\boldsymbol{C}(\boldsymbol{x})$, given by (17) and required to compute the MLS shape functions, is known. Our numerical experiments show that, using the third order reconstruction (82), the results obtained with diffuse second derivatives are almost identical to those obtained with full derivatives. This seems to be due to the fact that the order of accuracy of the full derivatives is preserved in the diffuse approach.

Full MLS first derivatives have been used in the examples shown below, in order to retain the whole accuracy of the MLS approximation (not only the order of convergence) in the evaluation of the viscous fluxes. Nevertheless, in the case of inviscid flows, the numerical experiments show again that diffuse first derivatives could be used for reconstruction purposes with no noticeable loss of accuracy.

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 analyzed in the case of viscous flow at moderate and 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 oportunity to assess the quality of the results provided by the high order upwind schemes.

Special attention is paid to the proposed third-order-reconstruction Roe scheme, in both inviscid and viscous flow applications. In the latter case, the low-dissipation properties of this algorithm look particularly interesting, with substantial improvements with respect to the second-order scheme. The unstructered quadrilateral meshes were generated using the code GEN4U, based on the formulation proposed by Sarrate [39].

4.1. Inviscid flows.

4.1.1. 2D 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 3 and corresponds to two reservoirs, with water levels $h_1 = 10\ m$ and $h_2 = 5\ m$ 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 $t = 7.2\ s$ (90 time steps) was obtained using the Lax-Wendroff and Roe schemes.

In both cases the MLS shape functions were computed using the quadratic basis (20), with circular supports of radius 2ρ, being $\rho = 1.2d$ and d the typical grid size; in this case $d = \Delta x = \Delta y = 2.5\ m$.

Second and third-order reconstructions, as exposed in section 3, were developed in order to obtain higher-order schemes. Slope limiting was carried out using (74)–(78) with either (79) or (80). First order derivatives are full derivatives (given by (18)), whereas second order derivatives were approximated by the diffuse ones (dropping the succesive derivatives of $\boldsymbol{C}(\boldsymbol{x})$).

The results for the second order scheme with limited gradients are shown in Figure 4. The sharp features of the flow are well captured by this scheme without oscillations and it seems that the use of (79) yields a slightly less dissipative limiting procedure.

The third-order-reconstruction scheme with limited first and second-order derivatives does not appear to produce a significant improvement of the results (Figure 5). We suspect that this fact is related to the rather strong limiting applied to the second 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 [40]. Further insight is provided by the solutions obtained with unlimited derivatives. This problem involves a rather mild shock and can be solved (at least using MLS approximation to compute the derivatives) without the introduction of limiters. In this case, first and second-order derivatives are computed directly at cell centers. The water surface contours for the second and third-order schemes are plotted in Figure 6. The second-order scheme without limiters yields a good solution, accurately capturing the shock front.

The third-order scheme produces a less smooth solution, but the front is even sharper than that of the second-order scheme.

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. The shape (and quality) of the solution is largely influenced by the adequate choice of such parameters. Figure 7 depicts the contours obtained with $C_h = 0.3$ and $C_V = 0.7$ (left), and $C_h = 1$ and $C_V = 3$ (right). In the first 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.

The water surface obtained with the second order Roe scheme is plotted in Figure 8.

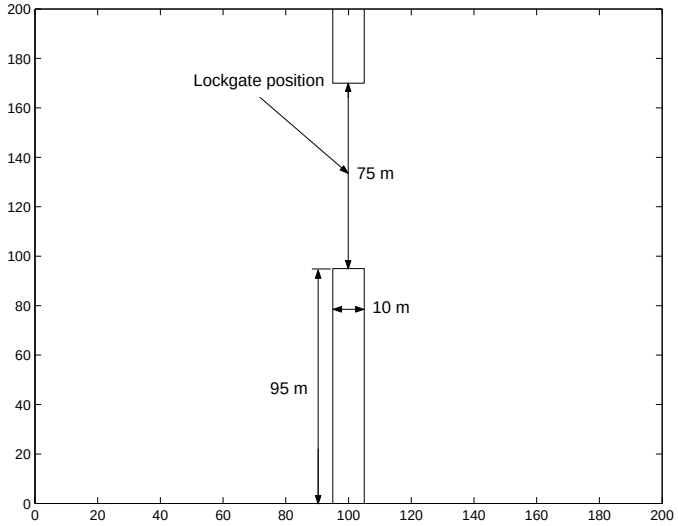

Figure 3. 2D break dam problem set up.

4.1.2. Supercritical flow in a channel with variable width. Given its multiscale nature, the MLS approximation possesses nice properties to be used in adaptive and multigrid strategies as a means to exchange information between coarse and refined meshes. To illustrate this point, 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 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 was obtained using the second order Roe scheme. The resuts obtained with the third order reconstruction were almost identical and are not shown. It was not possible to obtain solutions of comparable quality using the Lax-Wendroff scheme.

A first rough solution is computed using a coarse mesh (Figure 9, 917 cells). This solution is the interpolated using the MLS approximation on fine grids, and the computation is continued

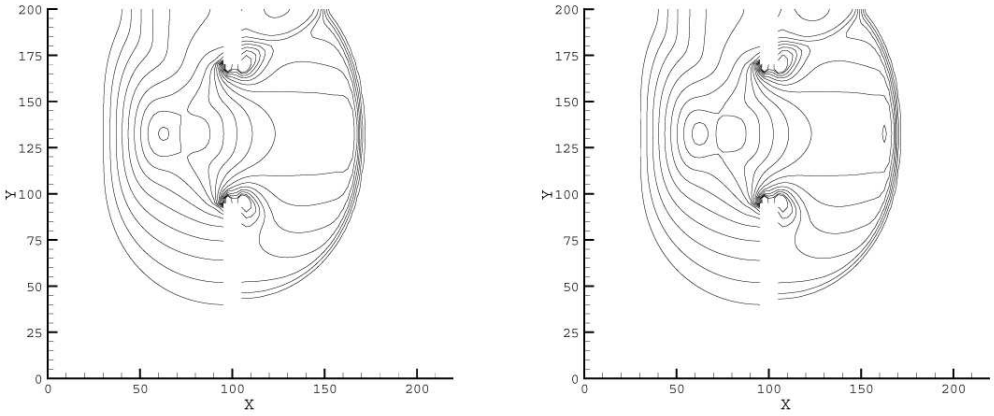

Figure 4. 2D break dam problem: surface contours at $t = 7.2\ s$, second order Roe scheme. Limited gradients using (79) (left) and (80) (right).

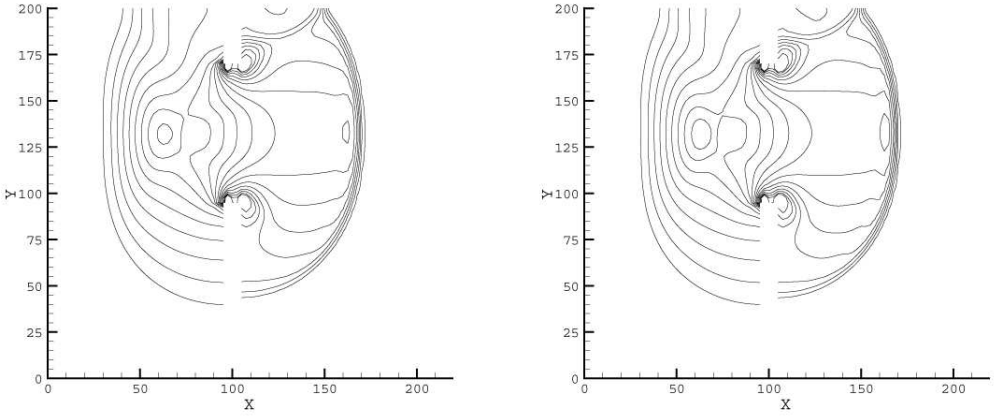

Figure 5. 2D break dam problem: surface contours at $t = 7.2\ s$, third order Roe scheme. Limited derivatives using (79) (left) and (80) (right).

until convergence. Two different grids were used: a uniformly refined mesh with 23349 cells (Figure 11), and an adapted mesh 9216 cells (Figures 10 and 12). Figure 13 depicts a 3D view of the water surface. The classical cross-wave structure of the flow can be easily identified and the hydraulic jumps are quite well captured..

4.1.3. Supercritical flow past a cylinder. In this last inviscid case we consider supercritical flow past a circular cylinder. The flow parameters at the inflow are: Froude number, $Fr = 4$ and unit water depth, $h = 1\ m$. The flow is impulsively started and some dissipation was needed to avoid instabilities in the early stages of the simulation. Thus, the procedure presented in section 3.3.2 was used at the beginning of the simulation, setting $k_2 = 2$ and $k_3 = 1$ and no base

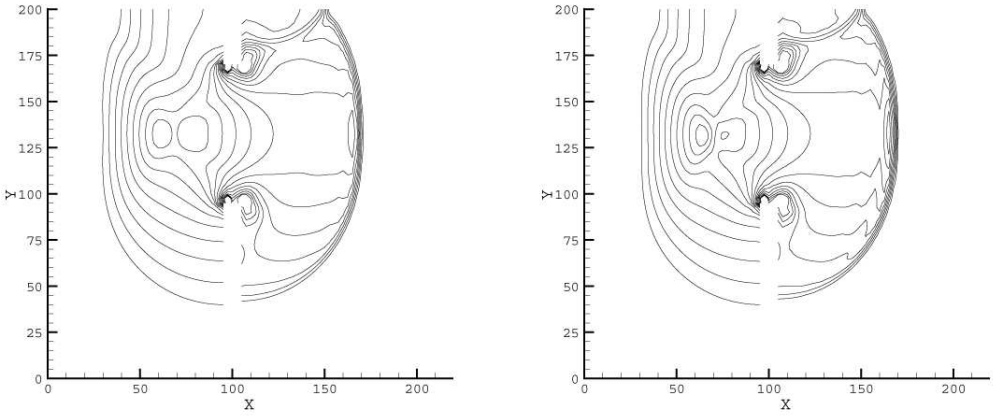

Figure 6. 2D break dam problem: surface contours at $t = 7.2$ s, second (left) and third (right) order Roe schemes with unlimited derivatives.

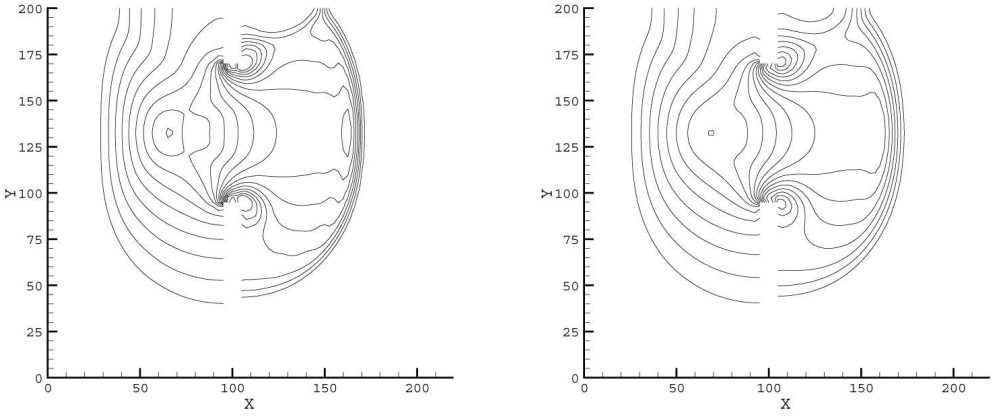

Figure 7. 2D break dam problem: surface contours at $t = 7.2$ s, Lax-Wendroff scheme with $C_h = 0.3$ and $C_V = 0.7$ (left), and $C_h = 1$ and $C_V = 3$ (right).

dissipation ($k_1 = 0$). After a few cycles, no additional dissipation was required ($k_1 = 0$, $k_2 = 0$, $k_3 = 0$). The computational mesh (4719 cells) and Froude number contours are depicted in Figure 14 (left and center, respectively). The shock in front of the cylinder is resolved within two cells. In order to analyze the ability of the nonlinear dissipation scheme exposed in section 3.3.2 to locate discontinuities, the distribution of k_p in the steady state is shown in Figure 14 (right). Note that the additional dissipation, if required, would be controlled by the parameter k_p, and negligible in the smooth regions of the flow.

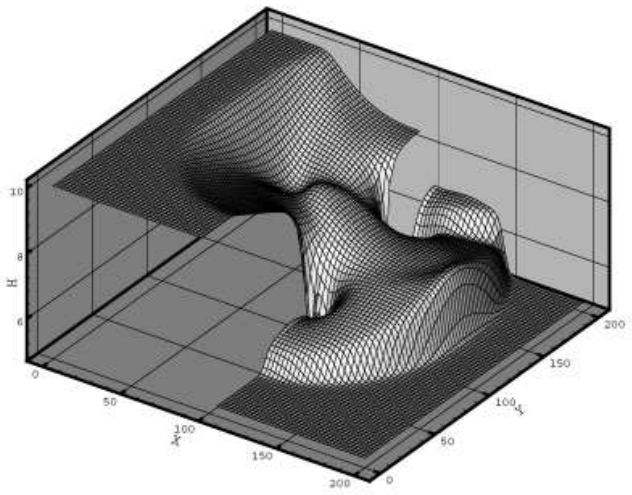

Figure 8. 2D break dam problem: water surface at $t = 7.2\ s$ (second order Roe scheme).

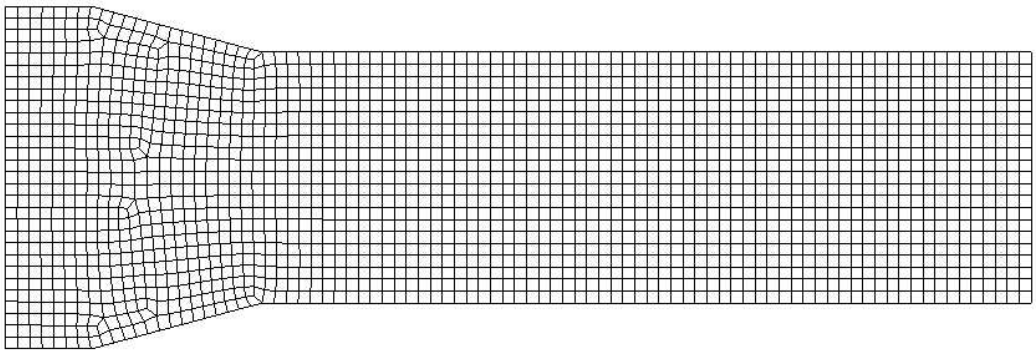

Figure 9. Supercritical channel flow: coarse mesh (917 cells).

4.2. Viscous flows.

We are also interested in flows where viscous and turbulence effects are of the utmost importance. A suitable numerical method to solve such problems on unstructured meshes 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.

It is crucial to assess whether high-resolution schemes such as Roe's and its high order extensions are suitable for general viscous flows. For this purpose, two different test cases are analyzed in this section, comparing the results with those of the low dissipative Lax-Wendroff scheme.

The first example intends to evaluate the influence of the limiting procedure on the accuracy

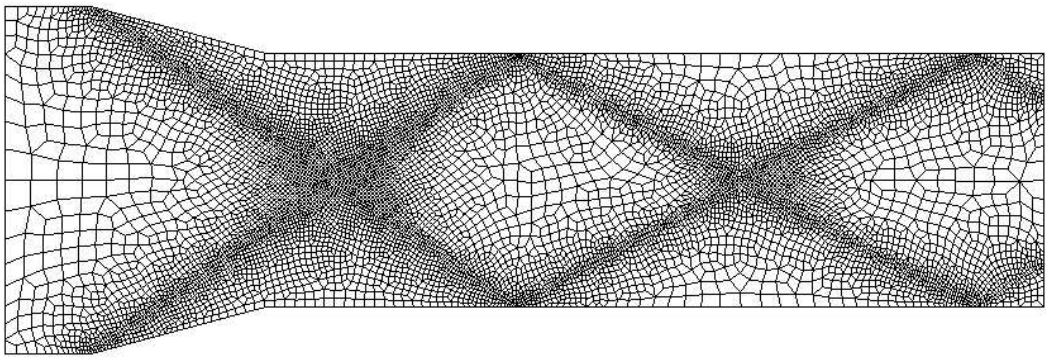

Figure 10. Supercritical channel flow: adapted mesh (9216 cells).

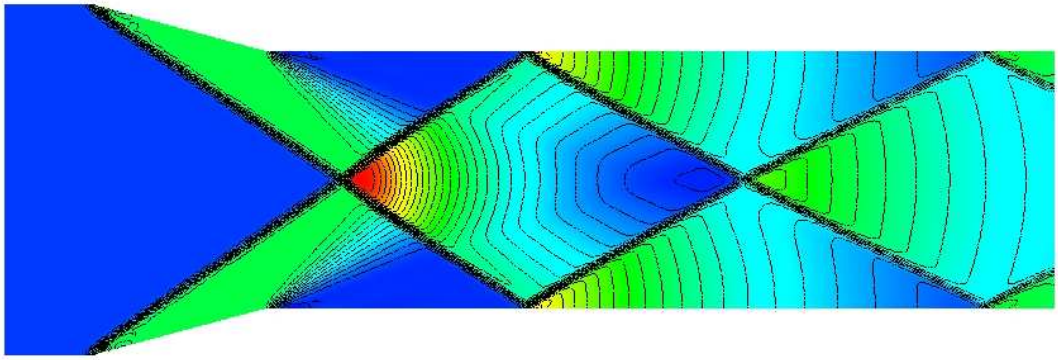

Figure 11. Supercritical channel flow: water surface contours (fine mesh).

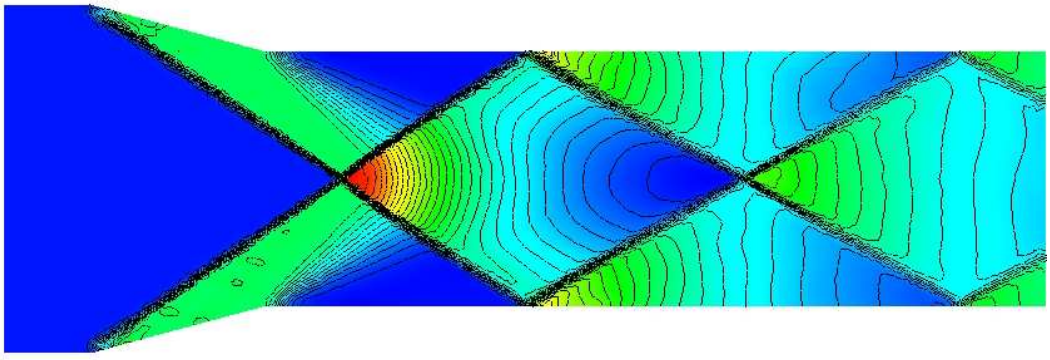

Figure 12. Supercritical channel flow: water surface contours (adapted mesh).

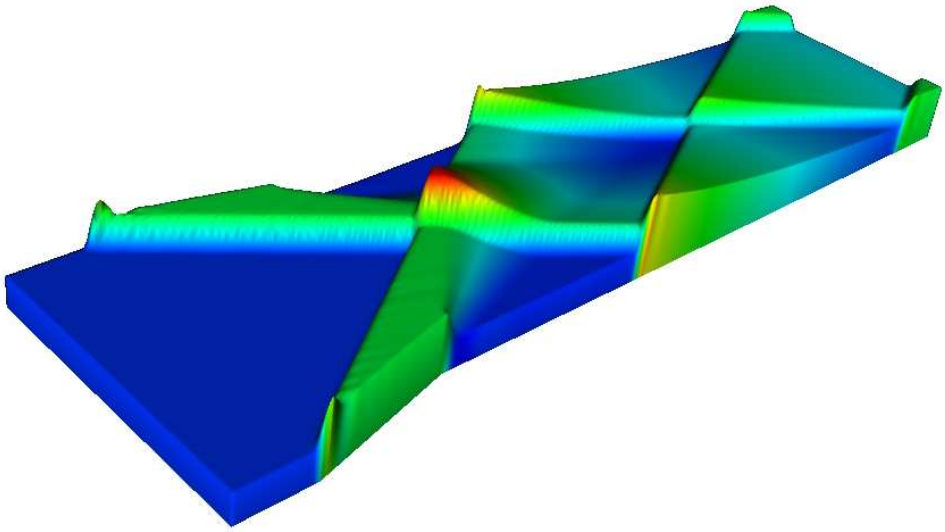

Figure 13. Supercritical channel flow: 3D view of the computed water surface.

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 statement is exposed in Figure 15. The free stream flow parameters are: Froude number, $Fr = 1.5$, unit depth, $h = 1\ m$ and Reynolds number $Re = 1000$, referred to a unit reference length, $L = 1\ m$. No-slip boundary conditions were applied along the wall boundary, $y = 0,\ 0.2 \leq x \leq 0.8$. 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 was run on two different meshes, plotted in Figures 16 and 17. The first is a structured non-uniform mesh, whereas the second is a (roughly) adapted, fully unstructered mesh. In the first case, and given the mesh structure, the MLS shape functions were computed using anisotropic weighting, according to (26).

Figures 18 and 19 show the computed Froude number profiles along the outlet section, with respectively limited and unlimited derivatives to develop the second and third order reconstructions in the Roe schemes. For the Lax-Wendroff scheme the artificial viscosity model exposed in section 3.2.3 was used, with $C_h = C_v = 0.1$. The unlimited Roe schemes look less dissipative than the Lax-Wendroff scheme with artificial viscosity. The solution provided by the third order scheme is slightly better than that of the second order scheme even at this moderate Reynolds number.

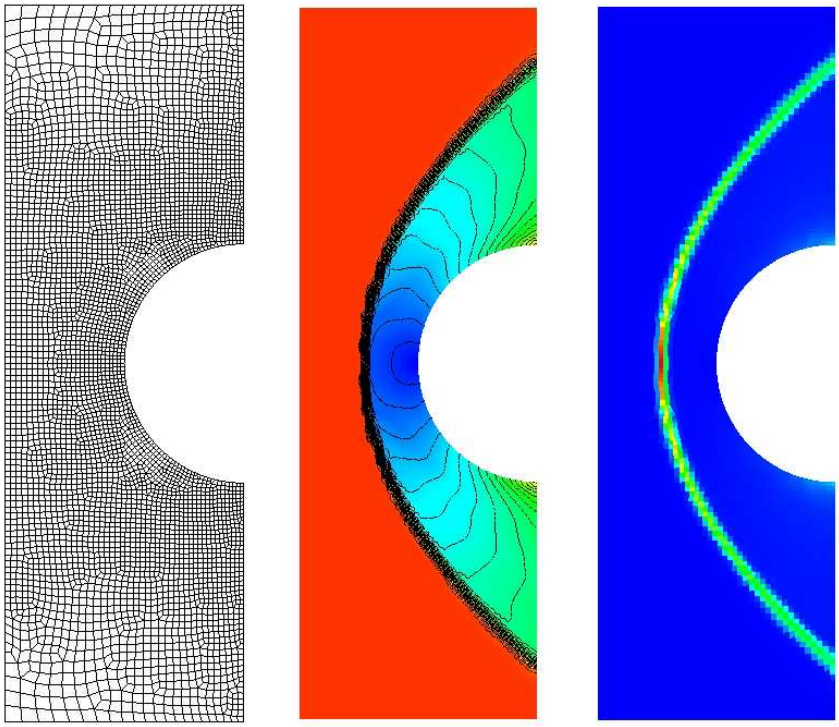

Figure 14. Supercritical flow past a cylinder ($Fr = 4$): computational mesh (4719 cells, left), Froude number contours (center) and distribution of k_p (right)

Even though the limiting procedure adds more dissipation in the smooth regions of the flow, the results and reasonably close to those of the unlimited reconstructions. The Froude number contours for the third order Roe scheme are plotted in Figures 16 (right) and 17 (right).

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), we have found this problem 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 botton is considered. The boundary conditions imposed are $h = 1\ m$, $q_x = 1\ m^2/s$ and $q_y = 0$ on $y = 1\ m$ (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 mesh employed is shown in Figure 20, and consists in 61x61 non-uniform cells. The grid has been refined near solid walls to account for the thin boundary layer.

The problem was solved for Reynolds numbers of 1000 and 10000. 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

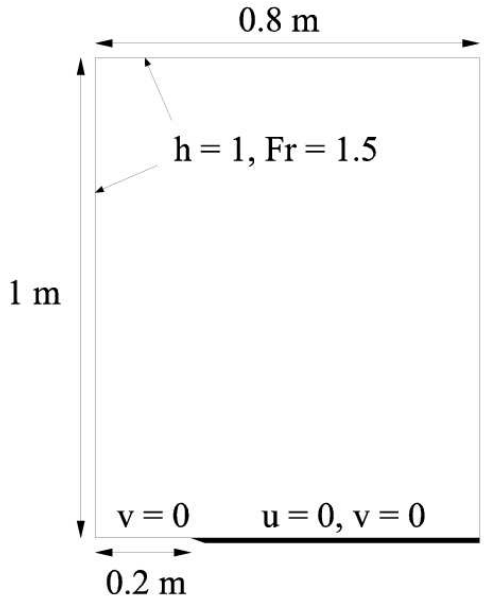

Figure 15. Supercritical viscous flow near a wall: problem set up.

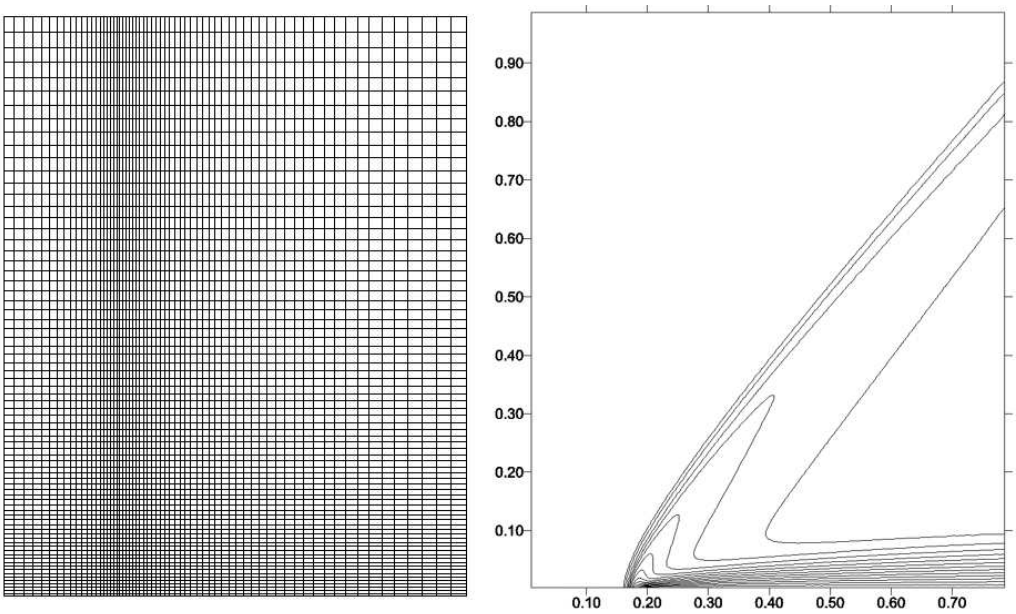

Figure 16. Supercritical viscous flow near a wall: structured computational mesh (4875 cells) and Froude number contours (right). Third order Roe scheme.

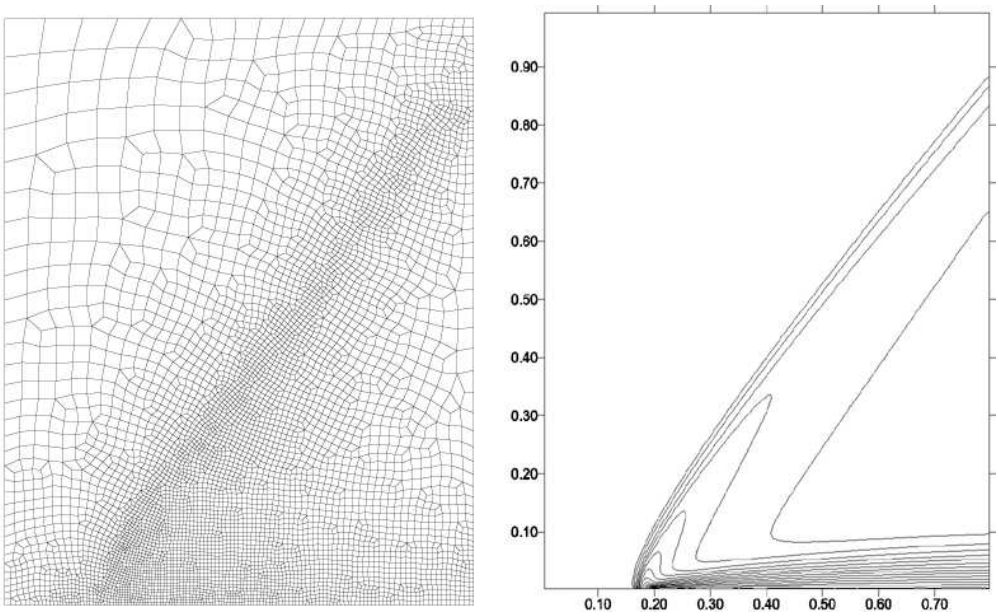

Figure 17. Supercritical viscous flow near a wall: unstructured computational mesh (5426 cells) and Froude number contours (right). Third order Roe scheme.

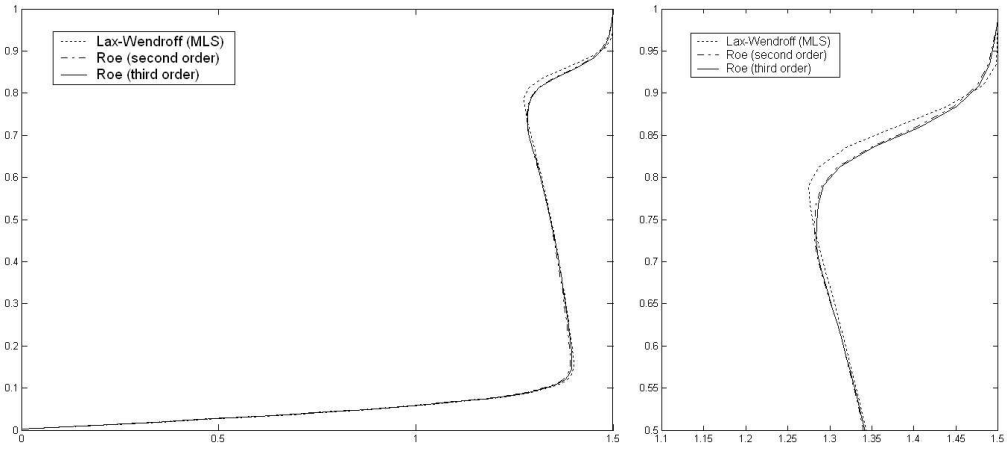

Figure 18. Supercritical viscous flow near a wall: Froude number profiles along $x = 0.8\ m$ (left) and close up comparison (right). Limited reconstructions.

any artificial dissipation model, and it was not necessary to use limiters in the reconstruction process for the Roe schemes. Instead of using the limiting formula (74), the corresponding linear and quadratic reconstructions (given by (72)–(73) and (82) respectively) were performed with first and second-order derivatives computed directly at cell centers using the MLS shape

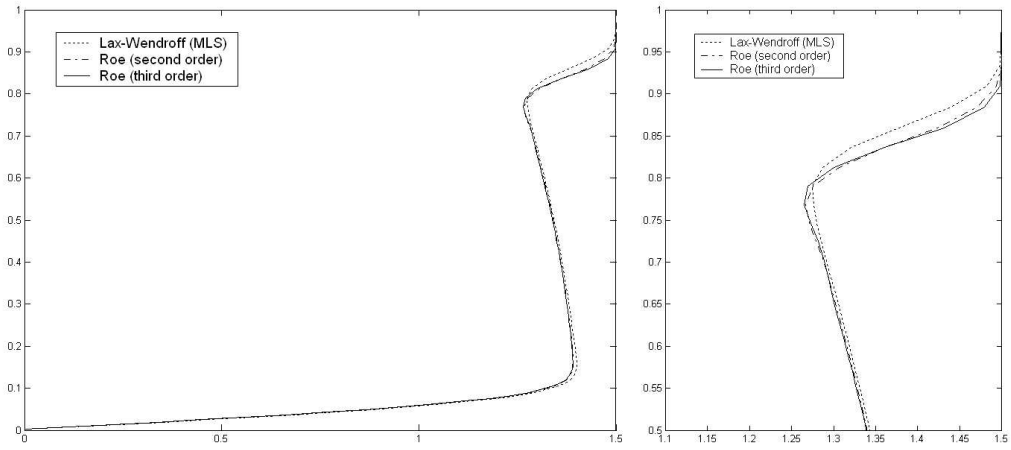

Figure 19. Supercritical viscous flow near a wall: Froude number profiles along $x = 0.8\ m$ (left) and close up comparison (right). Unlimited reconstructions.

functions. All first derivatives (the face derivatives used for the viscous fluxes and the cell-center derivatives used for the reconstruction) are full MLS derivatives, whereas the second-order derivatives used in the quadratic reconstruction are diffuse ones. Given the presence of highly stretched cells near the walls, anisotropic weighting was used according to (26).

The streamlines for the Lax-Wendroff and third-order Roe scheme are depicted in Figures 21–22. The horizontal velocity (u_x) profiles along $x = 0.5\ m$ for the Taylor-Galerkin, Lax-Wendroff and Roe schemes are plotted in Figures 23–24.

The different schemes yield almost identical solutions at moderate Reynolds numbers ($Re = 1000$, Fig. 23). However, at high Reynolds numbers ($Re = 10000$, Fig. 24) the dissipation properties of each scheme become more evident. The results obtained with the FEM Taylor-Galerkin and the proposed FV Lax-Wendroff schemes are very similar, although the latter predicts slightly higher maximum horizontal velocities, which could be considered a more accurate solution.

The second and third-order-reconstruction Roe schemes provide acceptable solutions for $Re = 10000$. 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 (Figure 24). The solution obtained with this latter approach is in better agreement with that of the more general Taylor-Galerkin and Lax-Wendroff schemes.

5. CONCLUSIONS

In this paper we explored the ability of a meshfree interpolation technique (namely, the Moving Least-Squares method) to be used in combination with finite volume discretizations on

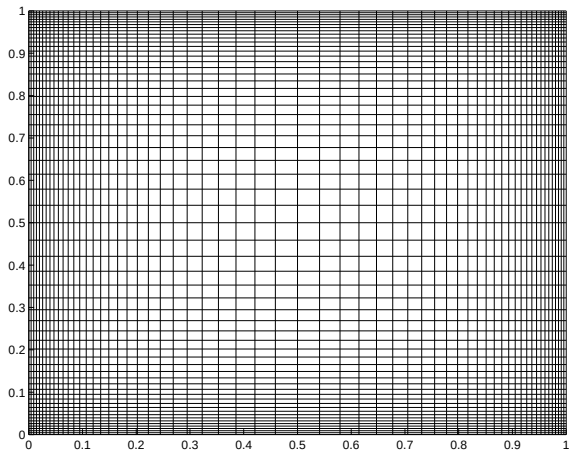

Figure 20. Lid driven cavity flow: computational mesh.

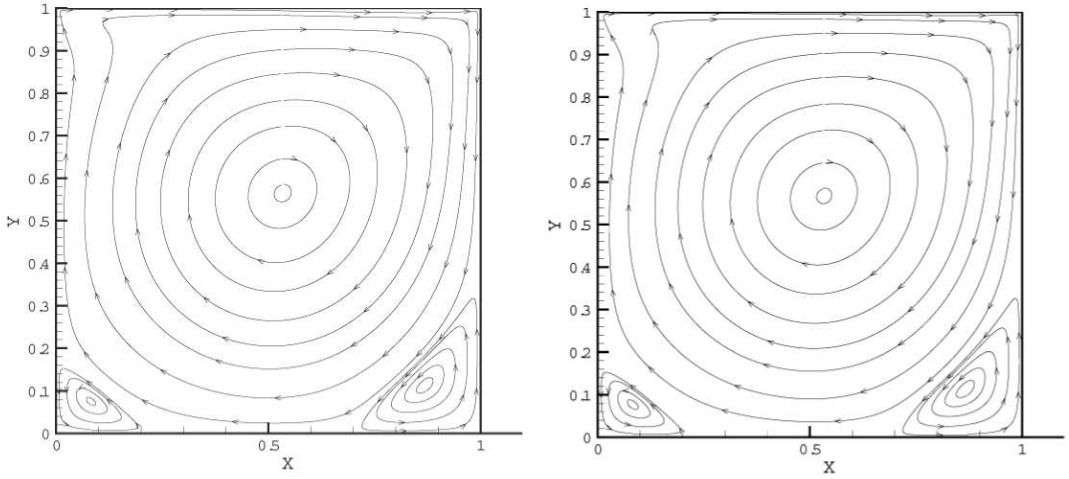

Figure 21. Lid driven cavity flow: streamlines for $Re = 1000$. Lax-Wendroff (left) and third order Roe (right) schemes.

unstructured grids. The resulting numerical schemes were applied to the resolution of the set of shallow water equations. We were particularly interested 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 second-order accurate in time Lax-Wendroff scheme was developed, with accuracy and stability properties very similar to those of its FEM counterpart, the Taylor-Galerkin method. A shock-capturing artificial viscosity model was proposed for this Lax-Wendroff scheme.

The development of high order upwind schemes on unstructured grids involves high-order

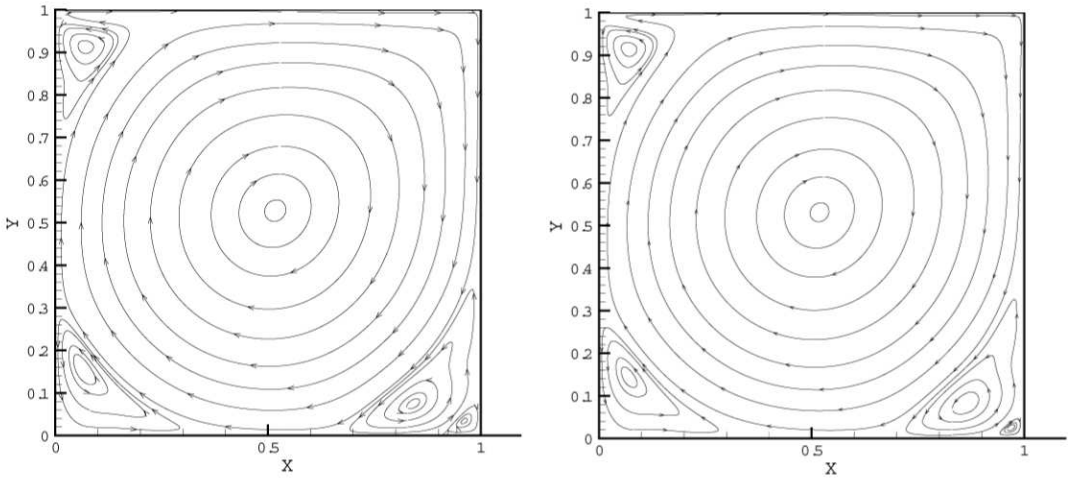

Figure 22. Lid driven cavity flow: streamlines for $Re = 10000$. Lax-Wendroff (left) and third order Roe (right) schemes.

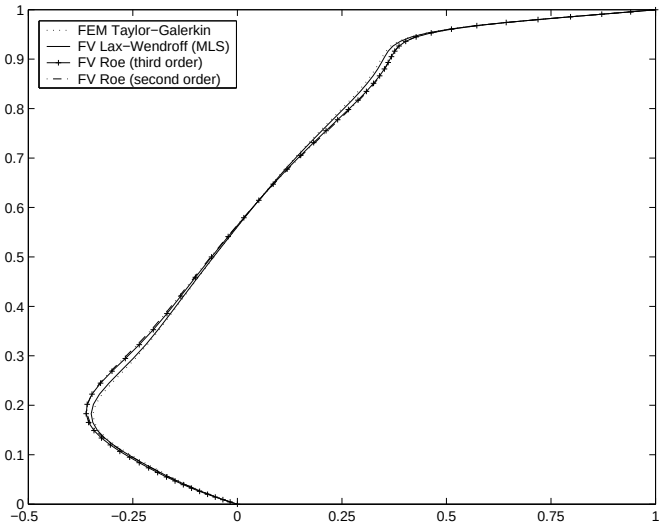

Figure 23. Lid driven cavity flow: profiles of u_x along $x = 0.5$ for $Re = 1000$.

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. It was shown in this study that the class of approximation methods employed in the "meshfree" 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

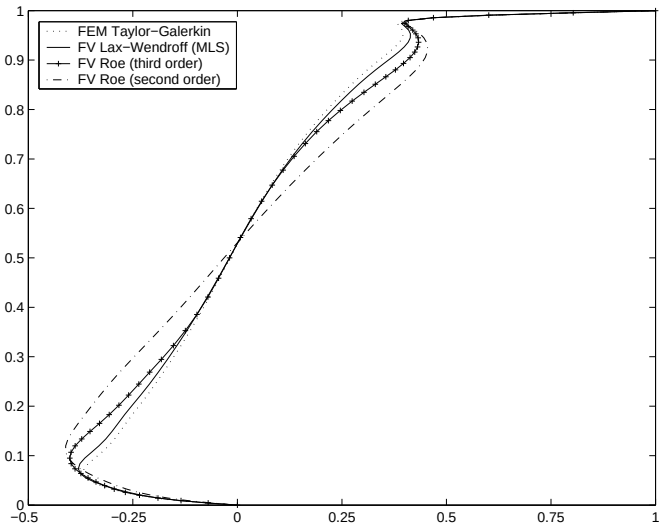

Figure 24. Lid driven cavity flow: profiles of u_x along $x = 0.5$ for $Re = 10000$.

a robust and general approximation framework which contitutes 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 Lax-Wendroff scheme with the proposed artificial viscosity model. On the other hand, the upwind schemes yield more dissipative solutions in the cavity flow test. However, the third-order-reconstruction Roe scheme developed here shows substantial improvements with respect to the second-order scheme, in good agreement with the results obtained with the more general Taylor-Galerkin and Lax-Wendroff algorithms. A more complete assessment of the suitability of this scheme for turbulent flow computations is currently in progress. The results obtained in this study for viscous flow at high Reynolds numbers are encouraging.

ACKNOWLEDGEMENTS

This work has been partially supported by the SGPICT of the "Ministerio de Ciencia y Tecnología" of the Spanish Government (Grant DPI# 2001-0556), 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. J. Peraire. A finite element method for convection dominated flows. PhD Thesis. University of Wales Swansea (1986).
2. J. Peraire, O.C. Zienkiewicz, K. Morgan. Shallow water problems: a general explicit formulation. *International Journal for Numerical Methods in Engineering.* **22**:547–574 (1986).
3. M. Quecedo, M. Pastor. A reappraisal of Taylor-Galerkin algorithm for drying-wetting areas in shallow water computations. *International Journal for Numerical Methods in Fluids.* **38**:515–531 (2002).
4. M. Quecedo, M. Pastor. Finite element modelling of free surface flows on inclined and curved beds. *Journal of Computational Physics.* **189**:45–62 (2003).
5. T.W.H. Sheu, C.C. Fang. High resolution finite-element analysis of shallow water equations in two dimensions. *Computer Methods in Applied Mechanics and Engineering.* **190**:2581–2601 (2001).
6. F. Alcrudo, P. García-Navarro. A high-resolution Godunov-type scheme in finite volumes for the 2D shallow-water equations. *International Journal for Numerical Methods in Fluids.* **16**:489–505 (1993).
7. D.H. Zhao, H.W. Shen, G.Q. Tabious III, J.S. Lai, W.Y. Tan. Finite-volume two-dimensional unsteady-flow model for river basins. *ASCE Journal of Hydraulic Engineering.* **120**:864–883 (1994).
8. K. Anastasiou, C.T. Chan. Solution of the 2D shallow water equations using the finite volume method on unstructured triangular meshes. *International Journal for Numerical Methods in Fluids.* **24**:1225–1245 (1997).
9. K. Hu, C.G. Mingham, D.M. Causon. A bore-capturing finite volume method for open-channel flows. *International Journal for Numerical Methods in Fluids.* **28**:1241–1261 (1998).
10. M.H. Tseng. Explicit finite volume non-oscillatory schemes for 2D transient free-surface flows. *International Journal for Numerical Methods in Fluids.* **30**:831–843 (1999).
11. R. Liszka, B. Wendroff. Two-dimensional shallow water equations by composite schemes. *International Journal for Numerical Methods in Fluids.* **30**:461–479 (1999).
12. J. Wang, R. Liu. The composite finite volume method on unstructured meshes for the two-dimensional shallow water equations. *International Journal for Numerical Methods in Fluids.* **37**:933–949 (2001).
13. J. Fe. A finite volume numerical formulation for shallow water turbulent flows. PhD Thesis. Universidad de A Coruña (2004).
14. T. Bui. A parallel, finite-volume algorithm for large-eddy simmulation of turbulent flows. *Computers and Fluids.* **29**:877–915 (2000).
15. S.Y. Lin, Y.F. Chen, S.C. Shih. Numerical study of MUSCL schemes for computational aeroacoustics. *AIAA–97–0023.* (1997).
16. V. Daru, C. Tenaud. High order one-step monotonicity-preserving schemes for unsteady compressible flow calculations. *Journal of Computational Physics.* **193**:563–594 (2004).
17. P. Lancaster, K. Salkauskas. Surfaces generated by moving least squares methods. *Math. of Comp.* **155**:141–158 (1981).
18. L. Cueto-Felgueroso, I. Colominas, G. Mosqueira, F. Navarrina, M. Casteleiro. On the Galerkin formulation of the SPH method. *International Journal for Numerical Methods in Engineering.* **60**:1475–1512 (2004).
19. P.L. Roe. Approximate Riemann solvers, parameter vectors and difference schemes. *Journal of Computational Physics.* **43**:357–372 (1981).
20. T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering.* **139**:3–47 (1996).
21. A.J. Chorin. Numerical study of slightly viscous flow. *Journal of Fluid Mechanics.* **57** (1973).
22. G.H. Cottet, P.D. Koumoutsakos. Vortex methods: theory and practice. Cambridge University Press, Cambridge, UK (2000).
23. L.B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal.* **82**:1013 (1977).
24. R.A. Gingold, J.J. Monaghan. Smoothed Particle Hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society.* **181**:378 (1977).
25. J.J. Monaghan. An introduction to SPH. *Computer Physics Communications.* **48**:89–96 (1988).
26. W.K. Liu, S. Li, T. Belytschko. Moving least-squares reproducing kernel methods: (I) Methodology and Convergence. *Computer Methods in Applied Mechanics and Engineering* **143**:113–154 (1997).
27. L. Cueto-Felgueroso. A unified analysis of meshless methods: formulation and applications. Technical Report (in Spanish), Universidad de La Coruña, (2002).
28. A. Gossler. Moving Least-Squares: a numerical differentiation method for irregularly spaced calculation points. SANDIA Report, SAND2001-1669 (2001).
29. P. Breitkopf, A. Rassineux, G. Touzot, P. Villon. Explicit form and efficient computation of MLS shape functions and their derivatives. *International Journal for Numerical Methods in Engineering.* **48**:451–466 (2000).
30. P. Jawahar, H. Kemath. A high-resolution procedure for Euler and Navier-Stokes computations on unstructured grids. *Journal of Computational Physics.* **164**:165–203 (2000).
31. A. Huerta, Y. Vidal, P. Villon. Pseudo-divergence-free element free Galerkin method for incompressible

fluid flow. *Computer Methods in Applied Mechanics and Engineering.* **193**:1119–1136 (2004).
32. P. Nithiarasu, O.C. Zienkiewicz, B.V.K. Satya Sai, K. Morgan, R. Codina, M. Vázquez. Shock capturing viscosities for the general fluid mechanics algorithm. *International Journal for Numerical Methods in Fluids.* **28**:1325–1353 (2000).
33. J. Burguete, P. García-Navarro. Efficient construction of high-resolution TVD consevative schemes for equations with source terms: application to shallow water flows. *International Journal for Numerical Methods in Fluids.* **37**:209–248 (2001).
34. B.V.K. Satya Sai, O.C. Zienkiewicz, M.T. Manzari, P.R.M. Lyra, K. Morgan. General purpose versus special algorithms for high-speed flows with shocks. *International Journal for Numerical Methods in Fluids.* **27**:57–80 (1998).
35. H.C. Yee, N.D. Sandham, M.J. Djomehri. Low-dissipative high-order shock-capturing methods using characteristic-based filters. *Journal of Computational Physics.* **150**:199–238 (1999).
36. P. Brufau, M.E. Vázquez-Cendón, P. García-Navarro. A numerical model for the flooding and drying of irregular domains. *International Journal for Numerical Methods in Fluids.* **39**:247–275 (2002).
37. M.E. Hubbard. Multidimensional slope limiters for MUSCL-type finite volume schemes on unstructured grids. *Journal of Computational Physics.* **155**:54–74 (1999).
38. H. C. Lin. Dissipation additions to flux difference splitting. *Journal of Computational Physics.* **117**:20–27 (1995).
39. J. Sarrate, A. Huerta. Efficient unstructured quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering.* **49**:1327–1350 (2000).
40. S. Serna, A. Marquina. Power ENO methods: a fifth-order accurate Weighted Power ENO method. *Journal of Computational Physics.* **194**:632–658 (2004).