Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 196 (2007) 4712–4736
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Finite volume solvers and Moving Least-Squares approximations for the compressible Navier–Stokes equations on unstructured grids

Luis Cueto-Felgueroso [a,b], Ignasi Colominas [b,*], Xesús Nogueira [b], Fermín Navarrina [b], Manuel Casteleiro [b]

[a] *Aerospace Computational Design Laboratory, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA*

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

Received 6 June 2006; received in revised form 10 January 2007; accepted 11 June 2007
Available online 16 June 2007

Abstract

This paper explores the approximation power of Moving Least-Squares (MLS) approximations in the context of higher-order finite volume schemes on unstructured grids. The scope of the application of MLS is threefold: (1) computation of high-order derivatives of the field variables for a Godunov-type approach to hyperbolic problems or terms of hyperbolic character, (2) direct reconstruction of the fluxes at cell edges, for elliptic problems or terms of elliptic character, and (3) multiresolution shock detection and selective limiting. A major advantage of the proposed methodology over the most popular existing higher-order methods is related to the viscous discretization. The use of MLS approximations allows the direct reconstruction of high-order viscous fluxes using quite compact stencils, and without introducing new degrees of freedom, which results in a significant reduction in storage and workload. A selective limiting procedure is proposed, based on the multiresolution properties of the MLS approximants, which allows to switch off the limiters in smooth regions of the flow. Accuracy tests show that the proposed method achieves the expected convergence rates. Representative simulations show that the methodology is applicable to problems of engineering interest.
© 2007 Elsevier B.V. All rights reserved.

Keywords: Compressible flow; Finite volume method; High-resolution methods; High-order methods; Moving Least-Squares; Unstructured grids

1. Introduction

This work is motivated by the question about whether piecewise polynomial approximations are the best option for the construction of higher-order Navier–Stokes solvers on unstructured grids. By this we refer to schemes that, either within the finite element framework (such as Discontinuous Galerkin methods [1]), or through suitable cell subdivisions (such as the so-called Spectral Volume method [2]), create new degrees of freedom inside each cell and use them to construct piecewise polynomial interpolants. The most important advantage of this kind of approaches is that, under certain conditions, the high-order interpolation/reconstruction can be formulated in a quite robust and general setting (particularly in the case of DG). Painful viscous discretizations, where additional degrees of freedom must be introduced and solved for, the necessarily high-order grids required and, in the case of higher-order DG, the absence of robust and accurate shock-capturing techniques, are major drawbacks of this approach. By accurate shock-capturing we mean shock resolutions comparable to typical second-order finite volume or residual distribution schemes.

Of course, having the general approximation framework of the finite element method makes DG the "safe" path, and it is understandable that many authors may be willing to pay the price of very high cost and "robustness

* Corresponding author. Tel.: +34 981167000; fax: +34 981167170.
E-mail address: icolominas@udc.es (I. Colominas).

0045-7825/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.cma.2007.06.003

uncertainty'' in order to follow it. This study is aimed at showing that, perhaps only for the more adventurous, there may be other less safe, but maybe better, options.

Originally devised for data processing and surface generation [3], Moving Least-Squares (MLS) became very popular within the *meshfree* community, being widely used, both in Eulerian and Lagrangian formulations, in order to provide spatial approximation. The characteristics of Moving Least-Squares and Reproducing Kernel methods have been extensively analyzed, both from theoretical and purely numerical approaches [4–8]. This class of approximation methods is particularly well suited for the reconstruction of a given function and its successive derivatives from scattered, pointwise data. This fact suggested the incorporation of MLS approximants into finite volume methods on unstructured grids [9], somewhat providing a kind of ''shape functions'' for unstructured-grid finite volume solvers.

The scope of the application of MLS to develop higher-order finite volume schemes, as we understand it, is threefold: (1) computation of high-order derivatives of the field variables for a Godunov-type approach to hyperbolic problems or terms of hyperbolic character, (2) direct reconstruction of the fluxes at cell edges, for elliptic problems or terms of elliptic character, and (3) multiresolution shock detection and selective limiting.

Quite the opposite to most existing high-order finite volume schemes, our approach is ''*top-down*''. Firstly, instead of adopting the cell-average framework, we work with *pointwise* values of the conserved variables, associated to the cell-centroids. Furthermore, our spatial representation, which is provided by the MLS approximants, is *continuous* and already high-order accurate. Note that the discretization of elliptic problems is straightforward within this framework. In order to deal with convection-dominated problems, and to apply the usual finite volume technology for hyperbolic terms, we *break* our continuous representation locally (inside each cell), by means of Taylor series expansions. The resulting scheme has the flavour of a Godunov-type method, but the accurate and clear discretization of elliptic terms is a crucial advantage over most existing approaches.

The strategy adopted for convection terms follows the ideas of the generalized Godunov method [10–12], performing piecewise polynomial reconstructions of the field variables inside each cell, and subsequently using those reconstructed variables as input data for a numerical flux function [12–15]. In practice, the construction of very high-order schemes of this kind has been severely limited by the absence of robust approximation techniques, capable of providing accurate estimates of the successive derivatives of the field variables on unstructured grids. Thus, the concept of high-order scheme is most frequently used in the literature in reference to formally second-order schemes (piecewise linear reconstruction). We believe the use of powerful approximation techniques like MLS may open new perspectives for this kind of schemes.

As mentioned above, a major advantage of the proposed methodology over the most popular existing higher-order methods is related to the viscous discretization. The use of MLS approximations allows the direct reconstruction of high-order viscous fluxes using quite compact stencils, and without introducing new degrees of freedom. This approach is conceptually similar to the successful second-order Multi-Point Flux Approximation (MPFA) methods developed by the petroleum engineering community [18].

Even though well behaved limiters for second-order schemes have been developed, the question for higher-order reconstructions is far from being clear. Therefore, selective shock-capturing is a critical issue in this context. If the limiters are active over the whole domain, their deleterious effect on higher-order derivatives results into a partial (or, quite frequently, complete) loss of the higher-order accuracy of the reconstruction in smooth regions of the flow, virtually taking the method back to second order.

A selective limiting procedure is proposed, based on the multiresolution properties of the MLS approximants [19], which allows to switch off the limiters in smooth regions of the flow. Note that the concept of ''smooth region'' itself is strongly related to the approximation being used, and hence the convenience of an indicator that is of the same order and nature as the approximants. In some sense, this procedure can be regarded as an unstructured grid generalization of the wavelet-based selective filtering proposed by Sjögreen and Yee for finite differences [20].

The outline of the paper is as follows. Section 2 presents the model equations and basic numerical scheme. Section 3 is a brief introduction to Moving Least-Squares Reproducing Kernel approximation methods, which is completed with some practical implementation issues presented in Section 4. Accuracy tests and representative simulations are exposed in Sections 5 and 6 and, finally, our main conclusions are drawn in Section 7.

2. Mathematical model and basic finite volume scheme

2.1. Governing equations

The compressible Navier–Stokes equations for two-dimensional flow, written in cartesian coordinates and in the absence of source terms, can be cast in conservative form as

$$\frac{\partial \boldsymbol{U}}{\partial t}+\frac{\partial(\boldsymbol{F}_x-\boldsymbol{F}_x^V)}{\partial x}+\frac{\partial(\boldsymbol{F}_y-\boldsymbol{F}_y^V)}{\partial y}=\boldsymbol{0} \tag{1}$$

being

$$\boldsymbol{U}=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{pmatrix},\qquad \boldsymbol{F}_x=\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uH\end{pmatrix},\qquad \boldsymbol{F}_y=\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vH\end{pmatrix} \tag{2}$$

the conserved variables and inviscid fluxes, respectively, and

$$\boldsymbol{F}_x^V = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - q_x \end{pmatrix}, \qquad \boldsymbol{F}_y^V = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} - q_y \end{pmatrix} \tag{3}$$

the viscous fluxes. In the above expressions, ρ denotes density, p, pressure and $\boldsymbol{v} = (u, v)$ is the velocity vector. The total energy and enthalpy are given by

$$\rho E = \rho e + \frac{1}{2}\rho \boldsymbol{v}\cdot\boldsymbol{v}, \qquad H = E + \frac{p}{\rho}, \tag{4}$$

where e is the specific internal energy. The viscous stresses are modelled as

$$\begin{aligned} \tau_{xx} &= 2\mu\frac{\partial u}{\partial x} - \frac{2}{3}\mu\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right), \\ \tau_{yy} &= 2\mu\frac{\partial v}{\partial y} - \frac{2}{3}\mu\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right), \\ \tau_{xy} &= \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \end{aligned} \tag{5}$$

where μ is the viscosity. The heat fluxes are assumed to be represented by Fourier's law

$$q_x = -\lambda\frac{\partial T}{\partial x}, \qquad q_y = -\lambda\frac{\partial T}{\partial y}, \tag{6}$$

where T denotes temperature, $\lambda = c_p\mu/P_r$ is the thermal conductivity, c_p the specific heat at constant temperature ($c_p = 1003.5$ for air) and P_r is the Prandtl number ($P_r = 0.72$ for air). The equation of state and temperature for an ideal gas can be written as

$$p = (\gamma - 1)\left(\rho E - \frac{1}{2}\rho\boldsymbol{v}\cdot\boldsymbol{v}\right), \qquad T = \frac{1}{c_v}\frac{p}{\rho(\gamma - 1)}, \tag{7}$$

where c_v is the specific heat at constant volume ($c_v = 716.5$ for air) and $\gamma = \frac{c_p}{c_v}$ is the ratio of specific heats ($\gamma = 1.4$ for air). The speed of sound is given by

$$c = \sqrt{\gamma p/\rho} \tag{8}$$

and the dynamic viscosity μ is assumed to be related to the temperature according to Sutherland's law

$$\mu = \mu_\infty \frac{T + S_0}{T_\infty + S_0}\left(\frac{T}{T_\infty}\right)^{1.5}, \tag{9}$$

where μ_∞ and T_∞ denote freestream viscosity and temperature, respectively, and $S_0 = 110.4K$ is an experimental constant [21].

2.2. Basic finite volume formulation

The basic finite volume discretization stems from the integral form of the conservation laws (1) over a control volume Ω_I (Fig. 1)

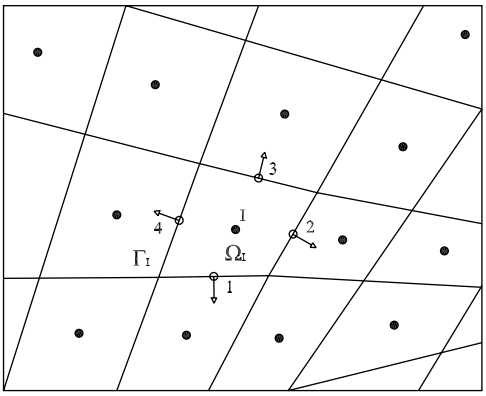

Fig. 1. Cell-centered finite volume discretization.

$$\int_{\Omega_I}\frac{\partial \boldsymbol{U}}{\partial t}\mathrm{d}\Omega + \int_{\Omega_I}\left(\frac{\partial(\boldsymbol{F}_x - \boldsymbol{F}_x^V)}{\partial x} + \frac{\partial(\boldsymbol{F}_y - \boldsymbol{F}_y^V)}{\partial y}\right)\mathrm{d}\Omega = \boldsymbol{0}. \tag{10}$$

Using the divergence theorem, the above expression can be written as

$$\int_{\Omega_I}\frac{\partial \boldsymbol{U}}{\partial t}\mathrm{d}\Omega = \int_{\Gamma_I}(\mathscr{F}^V - \mathscr{F})\cdot\boldsymbol{n}\,\mathrm{d}\Gamma, \tag{11}$$

where $\boldsymbol{n} = (n_x, n_y)$ is the outward pointing unit normal to the control volume boundary Γ_I, and the definitions

$$\mathscr{F} = (\boldsymbol{F}_x, \boldsymbol{F}_y), \qquad \mathscr{F}^V = (\boldsymbol{F}_x^V, \boldsymbol{F}_y^V) \tag{12}$$

are used for the sake of a more compact presentation. The idea behind the finite volume method is to discretize the computational domain into a set of non-overlapping control volumes (*cells*), in which the conservation equations are enforced. In this study a *cell-centered* approach with quadrilateral control volumes was adopted (Fig. 1), although the proposed methodology can be easily extended to more general finite volume discretizations. From a spatial point of view, and in principle, finite volume schemes involve studying the evolution of cell-averaged values of the field variables. Therefore, the underlying spatial representation is that of a piecewise constant flow field. Standard high-order schemes are constructed through the substitution of this piecewise constant representation for a *piecewise continuous* (usually polynomial) reconstruction of the flow variables inside each cell. In addition, and due to the fact that the reconstructed fields are still discontinuous across interfaces, special care must be paid to the discretization of the viscous fluxes, which are functions of the conserved variables, but also of their gradients. According to this description, most existing high-order finite volume schemes work within a cell-average setting, and proceed "*bottom-up*".

Our approach is somewhat the opposite. Firstly, instead of adopting the cell-average framework, we work with *pointwise* values of the conserved variables, associated to the cell-centroids. Furthermore, our spatial representation,

which is provided by the MLS approximants, is *continuous* and already high-order accurate. Note that the discretization of elliptic problems is straightforward within this framework. In order to deal with convection-dominated problems, and to apply the usual finite volume technology for hyperbolic terms, we *break* our continuous representation locally (inside each cell), by means of Taylor series expansions. In this sense, we proceed "*top-down*". The resulting scheme has the flavour of a Godunov-type method, but the accurate and clear discretization of elliptic terms is a crucial advantage of our scheme. More details of the proposed formulation can be found in [22].

Adopting the numerical method of lines, focusing on a control volume I, and assuming that suitable approximations to the inviscid and viscous fluxes are available at a set of quadrature points at each edge, the semi-discrete version of (11) reads

$$A_I \frac{\mathrm{d}\boldsymbol{U}_I}{\mathrm{d}t} = \sum_{iedge=1}^{nedge_I} \sum_{igau=1}^{ngau_I} \left[(\mathscr{F}^V - \mathscr{F}) \cdot \boldsymbol{n} \right]_{igau} \mathscr{W}_{igau}, \tag{13}$$

where A_I is the area of cell I, $nedge_I$ the number of cell edges, $ngau_I$ the number of Gauss quadrature points on each edge, $\mathscr{W}_{igau}$ denotes a quadrature weight and $\boldsymbol{U}_I$ represents, either the average value of $\boldsymbol{U}$ over the cell I (cell-average approach), or the pointwise value of $\boldsymbol{U}$ at the centroid of the cell I. In this latter case, the presence of A_I instead of a consistent *mass matrix* assumes that a mass-lumping has been performed.

It is critical in the development of robust high-order schemes for the Navier–Stokes equations to acknowledge the distinct nature of the inviscid and viscous fluxes. The former is of hyperbolic character, whereas the later is of elliptic character. It is widely accepted that the most powerful schemes for hyperbolic problems are those that take into account, in one way or another, the underlying wave structure of the equations. In the finite volume context, this can be achieved by using upwind *numerical flux* functions, that take as input variables the states on either side of each interface, and return a single numerical flux. First-order schemes use the cell-average values of the variables on each side of the interface as left and right states, whereas higher-order schemes use reconstructed ones, obtained from a certain extrapolation procedure. These ideas are in the basis of the generalized Godunov scheme [10–12], whose implementation involves three major steps in the explicit case:

- Development of piecewise continuous (usually polynomial) reconstructions of the flow variables inside each control volume, using either cell-averaged or pointwise information from neighbour cells. In our case, we use the point values of the variables at the cell centroids. The resulting spatial representation is still discontinuous across interfaces. The presence of discontinuities or steep gradients in the solution may require the use of some limiting strategy.

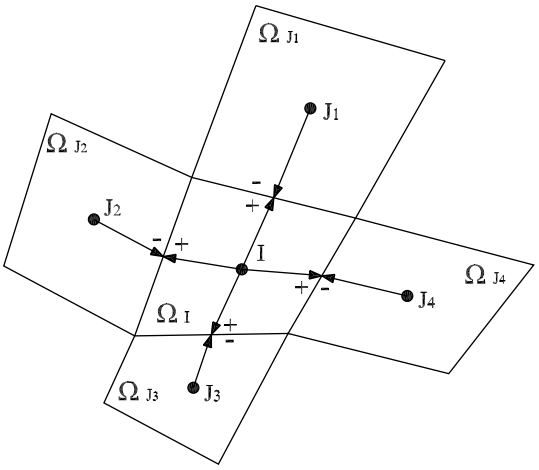

Fig. 2. Extrapolated variables used to evaluate the inviscid fluxes across the interfaces of control volumes Ω_I and $\{\Omega_{J_k}, k = 1,4\}$.

- Evaluation of fluxes at cell edges. The extrapolated left (+) and right (−) states at each edge integration point are used as input data for an approximate Riemann solver (Fig. 2).
- Solution advancement, using appropriate time stepping algorithms.

Viscous terms pose a major problem for methods that use piecewise polynomial approximations. Second-order schemes often use the average of the derivatives of the flow variables on either side of the interface to compute the viscous fluxes. Unfortunately, higher-order discretizations of elliptic equations or viscous terms cannot follow this path. One alternative is to decompose the original second-order system into a first-order one, with the consequent introduction of additional degrees of freedom. Another option, and the one that will be adopted in this study, is to perform a reconstruction of the viscous fluxes using information from neighbouring cells. This approach is sometimes thought to require large stencils, therefore being less efficient in practice. It is one of the objectives of this study to show that, with the reconstruction technique proposed, very competitive and efficient schemes for elliptic problems can be devised via multi-point reconstruction.

3. Moving Least-Squares Reproducing Kernel approximations

3.1. General formulation

Consider a function $u(\boldsymbol{x})$ defined in a domain Ω. MLS approximate $u(\boldsymbol{x})$, at a given point $\boldsymbol{x}$, through a weighted least-squares fitting of $u(\boldsymbol{x})$ in a neighbourhood of $\boldsymbol{x}$, as

$$u(\boldsymbol{x}) \approx \hat{u}(\boldsymbol{x}) = \sum_{i=1}^{m} p_i(\boldsymbol{x}) \alpha_i(\boldsymbol{z})|_{z=x} = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x}) \boldsymbol{\alpha}(\boldsymbol{z})|_{z=x}. \tag{14}$$

Note that $\hat{u}(\boldsymbol{x})$ is not a polynomial, in general. In the above expression, $\boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})$ is an m-dimensional basis of functions

(usually polynomials) and $\boldsymbol{\alpha}(\boldsymbol{z})|_{\boldsymbol{z}=\boldsymbol{x}}$ is a set of parameters to be determined, and such that they minimize the following error functional:

$$J(\boldsymbol{\alpha}(\boldsymbol{z})|_{\boldsymbol{z}=\boldsymbol{x}}) = \int_{\boldsymbol{y}\in\Omega_{\boldsymbol{x}}} W(\boldsymbol{z}-\boldsymbol{y},h)|_{\boldsymbol{z}=\boldsymbol{x}}[u(\boldsymbol{y}) - \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{y})\boldsymbol{\alpha}(\boldsymbol{z})|_{\boldsymbol{z}=\boldsymbol{x}}]^2\,\mathrm{d}\Omega_{\boldsymbol{x}} \tag{15}$$

being $W(\boldsymbol{z}-\boldsymbol{y},h)|_{\boldsymbol{z}=\boldsymbol{x}}$ a *kernel* (also weighting, smoothing or window function) with compact support (denoted by $\Omega_{\boldsymbol{x}}$) centered at $\boldsymbol{z}=\boldsymbol{x}$. The parameter h, usually called *smoothing length* or *dilatation parameter* in the meshfree literature, is a certain characteristic measure of the size of the support $\Omega_{\boldsymbol{x}}$ (e.g. kernels with circular supports of radius $2h$). Splines are the most frequent kernels, in particular the cubic spline used in this study

$$W(\boldsymbol{x}-\boldsymbol{y},h) = \begin{cases} 1-\frac{3}{2}s^2+\frac{3}{4}s^3 & s\leqslant 1,\\ \frac{1}{4}(2-s)^3 & 1<s\leqslant 2,\\ 0 & s>2, \end{cases} \tag{16}$$

where $s=\frac{|\boldsymbol{x}-\boldsymbol{y}|}{h}$. In practice, the minimization of (15) provides a means to approximate or reconstruct $u(\boldsymbol{x})$, at any point $\boldsymbol{x}\in\Omega$, from its pointwise value at a number of scattered locations in Ω, which are often called *particles* or *nodes*.

The integral in (15) is evaluated using nodal integration and, given the compact support of the kernel, only those nodes inside $\Omega_{\boldsymbol{x}}$ are involved as quadrature points. After some algebra, the set of parameters $\boldsymbol{\alpha}$ that minimize the functional J are obtained as

$$\boldsymbol{\alpha}(\boldsymbol{z})|_{\boldsymbol{z}=\boldsymbol{x}} = \boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{P}_{\Omega_{\boldsymbol{x}}}\boldsymbol{W}_V(\boldsymbol{x})\boldsymbol{u}_{\Omega_{\boldsymbol{x}}}, \tag{17}$$

where the vector $\boldsymbol{u}_{\Omega_{\boldsymbol{x}}}$ contains the pointwise values of the function to be reproduced, $u(\boldsymbol{x})$, at the $n_{\boldsymbol{x}}$ particles inside $\Omega_{\boldsymbol{x}}$ (Fig. 3)

$$\boldsymbol{u}_{\Omega_{\boldsymbol{x}}} = (\,u(\boldsymbol{x}_1)\quad u(\boldsymbol{x}_2)\quad\cdots\quad u(\boldsymbol{x}_{n_{\boldsymbol{x}}})\,)^{\mathrm{T}}. \tag{18}$$

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

$$\boldsymbol{P}_{\Omega_{\boldsymbol{x}}} = (\,\boldsymbol{p}(\boldsymbol{x}_1)\quad\boldsymbol{p}(\boldsymbol{x}_2)\quad\cdots\quad\boldsymbol{p}(\boldsymbol{x}_{n_{\boldsymbol{x}}})), \tag{19}$$

$$\mathbf{W}_{\mathbf{V}}(\boldsymbol{x}) = \mathrm{diag}\{W_i(\boldsymbol{x}-\boldsymbol{x}_i)V_i\},\quad i=1,\ldots,n_{\boldsymbol{x}}. \tag{20}$$

Complete details can be found in [4,5]. In the above equations, V_i and $\boldsymbol{x}_i$ denote, respectively, the tributary volume (used as quadrature weight) and coordinates associated to node i. Note that the tributary volumes of the neighbouring nodes are included in matrix $\mathbf{W}_{\mathbf{V}}$, obtaining an MLS version of the Reproducing Kernel Particle Method [4]. Otherwise, we can use $\mathbf{W}$ instead of $\mathbf{W}_{\mathbf{V}}$

$$\mathbf{W}(\boldsymbol{x}) = \mathrm{diag}\{W_i(\boldsymbol{x}-\boldsymbol{x}_i)\},\quad i=1,\ldots,n_{\boldsymbol{x}}, \tag{21}$$

which corresponds to the classical MLS approximation (in the nodal integration of the functional (15), the same quadrature weight is associated to all nodes). Introducing (17) in (14), the interpolation structure can be identified as

$$\hat{u}(\boldsymbol{x}) = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{P}_{\Omega_{\boldsymbol{x}}}\boldsymbol{W}(\boldsymbol{x})\boldsymbol{u}_{\Omega_{\boldsymbol{x}}} = \boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{u}_{\Omega_{\boldsymbol{x}}} = \sum_{j=1}^{n_{\boldsymbol{x}}} N_j(\boldsymbol{x})u_j. \tag{22}$$

In analogy to finite elements, the approximation was written in terms of the MLS "shape functions"

$$\boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x}) = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{P}_{\Omega_{\boldsymbol{x}}}\boldsymbol{W}(\boldsymbol{x}), \tag{23}$$

where $N_j(\boldsymbol{x})$ can be seen as the shape function associated to particle j. The functional basis $\boldsymbol{p}(\boldsymbol{x})$ is strongly related to the accuracy of the MLS fit. Theory and numerical evidence [7] show that, for a pth order MLS fit (pth order complete polynomial basis) and general, irregularly spaced points, the nominal order of accuracy for the approximation of a sth order gradient is roughly $(p-s+1)$. In general, any linear combination of the functions included in the basis is exactly reproduced by the MLS approximation.

In 2D, the $p=2$ basis reads

$$\boldsymbol{p}(\boldsymbol{x}) = (\,1\quad x_1\quad x_2\quad x_1x_2\quad x_1^2\quad x_2^2\,)^{\mathrm{T}} \tag{24}$$

and the $p=3$ basis is given by

$$\boldsymbol{p}(\boldsymbol{x}) = (\,1\quad x_1\quad x_2\quad x_1x_2\quad x_1^2\quad x_2^2\quad x_1^2x_2\quad x_1x_2^2\quad x_1^3\quad x_2^3\,)^{\mathrm{T}}. \tag{25}$$

In the above expression, (x_1,x_2) denotes the cartesian coordinates of $\boldsymbol{x}$. To improve the conditioning of the moment matrix, it is most frequent to use scaled and *locally* defined monomials in the basis. Thus, if the shape functions were to be evaluated at a certain point $\boldsymbol{x}_I$, the basis would be of the form $\boldsymbol{p}\left(\frac{\boldsymbol{x}-\boldsymbol{x}_I}{h}\right)$, instead of $\boldsymbol{p}(\boldsymbol{x})$. With this transformation, the MLS shape functions read

$$\boldsymbol{N}^{\mathrm{T}}(\boldsymbol{x}_I) = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{0})\boldsymbol{C}(\boldsymbol{x}_I) = \boldsymbol{p}^{\mathrm{T}}(\boldsymbol{0})\boldsymbol{M}^{-1}(\boldsymbol{x}_I)\boldsymbol{P}_{\Omega_{\boldsymbol{x}_I}}\boldsymbol{W}(\boldsymbol{x}_I), \tag{26}$$

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

$$\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{P}_{\Omega_{\boldsymbol{x}}}\boldsymbol{W}(\boldsymbol{x}). \tag{27}$$

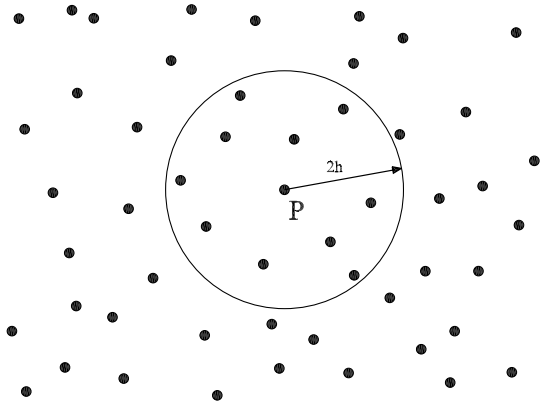

Fig. 3. Meshfree approximation: general scheme. Support for reconstruction at P.

The approximate derivatives of $u(\boldsymbol{x})$ can be expressed in terms of the derivatives of the MLS shape functions, which are functions of the derivatives of the polynomial basis $\boldsymbol{p}\left(\frac{\boldsymbol{x}-\boldsymbol{x}_I}{h}\right)$ and the derivatives of $\boldsymbol{C}(\boldsymbol{x})$ [9,23,24].

The first-order derivatives of the shape functions are computed in this study as full MLS derivatives, whereas second and third-order derivatives are *approximated* by the *diffuse* ones. In the diffuse approach, the successive derivatives of $\boldsymbol{C}(\boldsymbol{x})$ are neglected. Note that the diffuse derivatives of the shape functions are readily obtained once the matrix $\boldsymbol{C}(\boldsymbol{x})$ is computed. Although this approach greatly simplifies the presentation and implementation of the MLS approximants, problems with rough grids may require the use of full derivatives.

More details of the MLS procedure used in this paper can be found in [9,23].

3.2. Computational aspects

The MLS shape functions are data independent and, therefore, for fixed grids they need to be computed *only once* at the preprocessing phase. Note again that *the reconstructed function is not a polynomial*, even in the case when the basis of functions comprises only polynomials.

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 $\boldsymbol{p}(\boldsymbol{x})$. Note that the size of $\boldsymbol{M}$ does not depend on the number of neighbours in the cloud of the evaluation point.

In order to prevent the matrix $\boldsymbol{M}$ from being singular or ill-conditioned, the cloud of neighbours should fulfill certain "good neighbourhood" requirements. Thus, if the number of neighbours is less than m (the number of functions in the basis), $\boldsymbol{M}$ becomes singular. Nevertheless, the approximation could be poor if $\boldsymbol{M}$ is severely ill-conditioned, so it is convenient to use a number of neighbours slightly above the minimum, and with the information coming from as many directions as possible. For rough grids it may be necessary to use anisotropic kernels [22]. The definition of the cloud (the *MLS stencil*) for each evaluation point is an important issue that will be addressed in Sections 4.2 and 4.3. The selection process must be suitable for general unstructured grids, and the stencil should be as compact as possible for the sake of computational efficiency and physical meaning.

Once the cloud of neighbour centroids has been determined, the smoothing length h for isotropic kernels (radial weighting) is set to be proportional to the maximum distance between the evaluation point $\boldsymbol{x}_I$ and its neighbours, as

$$h = k \max(\|\boldsymbol{x}_j - \boldsymbol{x}_I\|). \tag{28}$$

Values of k around 0.6–0.7 seem to be adequate (recall that, using radial weighting, the support of the kernel expands over a circle of radius $2h$).

3.3. Moving-Least Squares vs. piecewise polynomial interpolation

Most existing higher-order schemes are based on piecewise polynomial approximations, which are obtained either within the finite element framework (consider the Discontinuous Galerkin method [1]), or using some suitable form of cell subdivision (such as the so-called Spectral Volume method [2]). Following this approach, higher-order accuracy is achieved by creating new degrees of freedom inside each cell, which are used to construct an interpolating polynomial. This piecewise polynomial interpolation is discontinuous across element interfaces, a feature that is quite convenient in terms of the stability and compactness of the scheme for hyperbolic problems, but also quite inconvenient in terms of the efficiency of the scheme for equations and terms of elliptic character. The way Moving-Least Squares approximations work is rather different, and this section is aimed at shedding some light on its advantages and shortcomings.

Even though the MLS approximants will be later used in a "moving" (centered) sense, Figs. 4 and 5 present some examples of MLS shape functions computed in an "element" sense. By this we mean that, in order to compute the set of p-complete MLS shape functions associated to N points on $[-1,+1]$, the cloud for each point comprises all the N points, instead of using compact supports. This may be useful to give a flavour of the structure of the shape functions, and to have a first comparison to the Lagrange basis.

Fig. 4 presents the computed shape functions for $p = 5$, $N = 9$ (bottom left), and $p = 10$, $N = 15$ (bottom right). The MLS points are uniformly spaced. The Lagrange basis for $p = 5$ and $p = 10$, computed with uniform nodes, are also plotted (top left and top right, respectively). For $p = 10$, it is clear that non-uniform nodes should be used for the Lagrange basis, and the same is true for MLS, although the MLS basis is slightly better behaved. Note that the MLS shape functions do not bear the Kronecker delta property. The smoothing length is $h = 0.6 d_{\max}$, where $d_{\max}$ is the maximum of the distances between the evaluation point and its neighbours. Fig. 5 gives some insight into the effect over the shape functions of changes in the number of points in the cloud N, or in the point distribution. Thus, the shape functions present a better behaviour when more points are added to the cloud (top right). Good non-uniform point distributions have the same effect as in the Lagrange basis (bottom left). Finally, a set of basis functions for irregularly spaced points is presented (bottom right).

This is not, however, the way Moving Least-Squares are usually employed. They are better defined as a "centered" approximation, without reference to an underlying element or patch structure. Thus, the interpolation is based on a "nearest neighbours" or *stencil* structure, which is *local* and *centered* at the evaluation point (the stencil *moves* to the evaluation point). We believe this feature has some

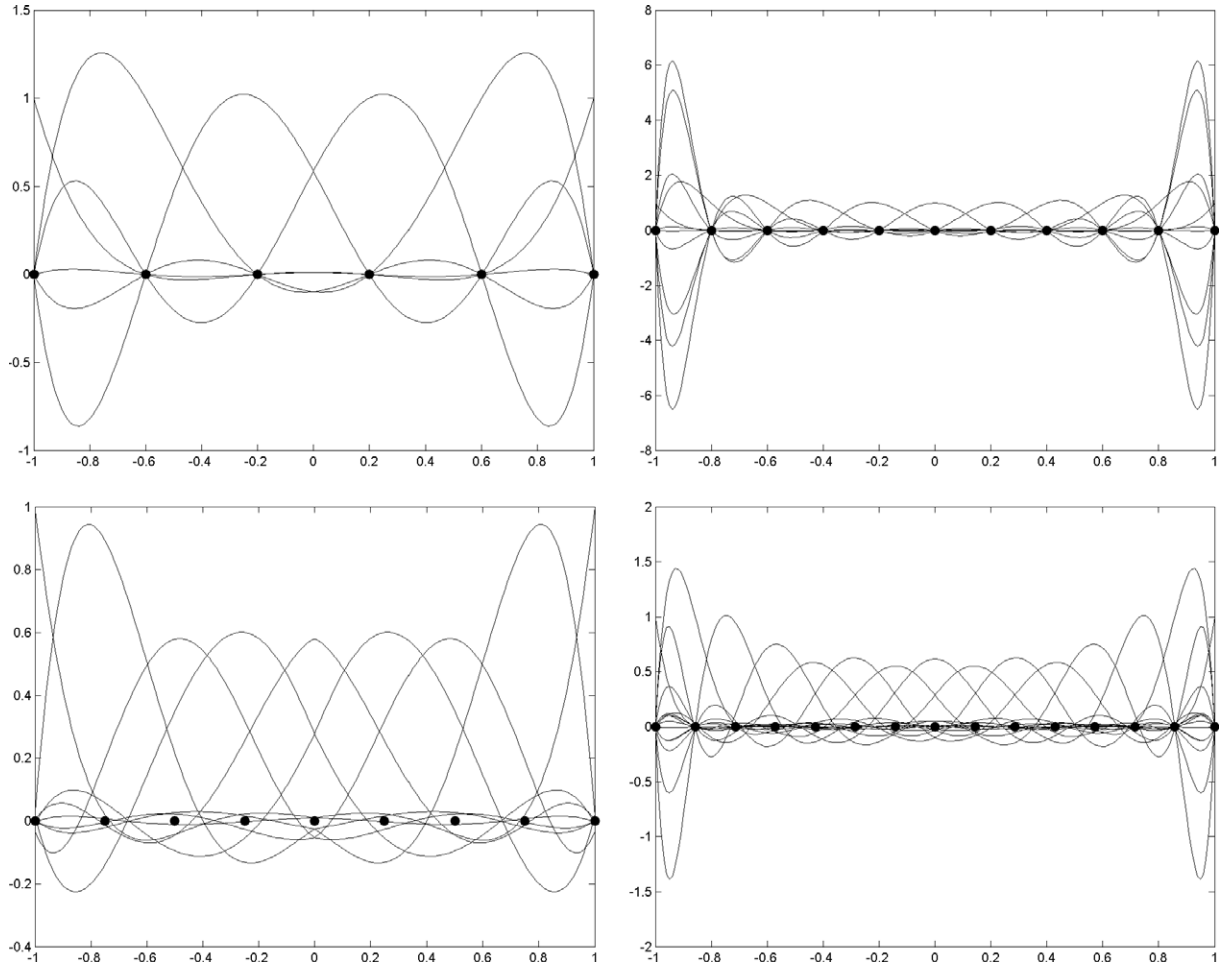

Fig. 4. MLS shape functions for $p = 5$, $N = 9$ (bottom left), and $p = 10$, $N = 15$ (bottom right). Lagrange basis for $p = 5$ (top left) and $p = 10$ (top right).

advantages over piecewise polynomial interpolation. The first one is that, for the same order, higher accuracy can be achieved even with irregular point distributions. Another one is that the interpolation is continuous across interfaces, which will allow the direct computation of high-order viscous fluxes in a multi-point fashion.

Figs. 6 and 7 present the errors in the interpolation of $u = \sin(2\pi x)$ in the domain $[0,+1]$. The function value is interpolated at 800 points for plotting, using 40 point values for MLS. Several values of p and N will be discussed, and the smoothing length is defined as before. The function is also interpolated using piecewise polynomials, with a number of elements such that the grid resolution h/p is the same as that of the MLS point distribution, and with the nodes placed at the Gauss–Lobatto points. Fig. 6 plots the error distribution for $p = 4$ and $p = 8$. The MLS points are evenly spaced. When the minimum number of neighbours, $p + 1$, is used (top left and bottom left, respectively), the accuracy of MLS for interior nodes is significantly higher than that of the piecewise polynomial interpolation. Furthermore, note that the difference increases with the approximation order. Note that in Fig. 7 the solutions were computed using random points (MLS), and the optimal Gauss–Lobatto node distribution (piecewise polynomial), respectively. For redundant point clouds, $N > p + 1$ (Fig. 6, top right and bottom right), the piecewise polynomial interpolation is more accurate, although the differences for interior points are small. We must point out that generating good non-uniform nodal distributions for high-order piecewise polynomial interpolants is straightforward in 1D (the Gauss–Lobatto points are optimal), but the multidimensional case is far from being so, particularly in the case of methods that use cell subdivisions on triangles.

One of the main shortcomings of MLS approximants is also apparent from Figs. 6 and 7. For $p = 8$, the interpolation errors near the boundaries of the global domain are about an order of magnitude higher than those inside the domain. This is associated to the one-sided MLS approximation, and is more and more pronounced as p is increased. We must point out that suboptimal node distributions for piecewise polynomial interpolations would have the same effect, but in this case on all cells, not just near the domain boundaries as MLS. We believe that this effect is less important for most practical values of p (maybe up to $p = 5$), and that it can be alleviated by the use of ghost points and special boundary kernels.

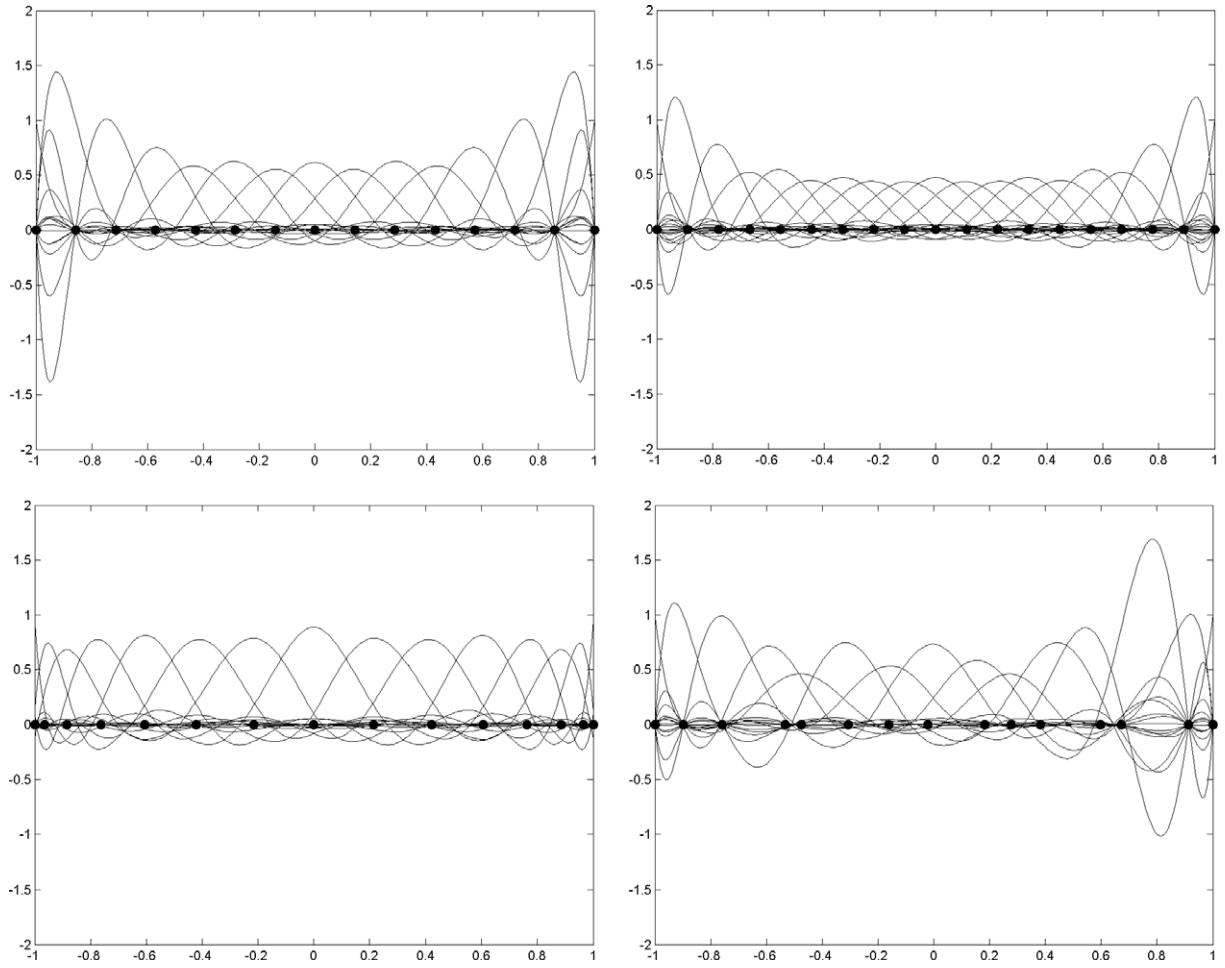

Fig. 5. MLS shape functions for $p = 10$: $N = 15$ uniform points (top left), $N = 19$ uniform points (top right), $N = 15$ Gauss–Lobatto points (bottom left) and $N = 15$ random points (bottom right).

Another potential weakness of MLS is the conditioning of the moment matrix, either due to the cloud structure (highly anisotropic grids), or to very high values of p. In multidimensions the definition of the stencil is probably a more stringent limitation to stay in p's up to 4 or 5 in 2D/3D, which, on the other hand, is enough for most practical applications. The use of anisotropic kernels that adapt to the point distribution helps providing robustness in general grids, but the question is far from being resolved.

3.4. Multiple scale analysis and shock detection

One of the most interesting features of the MLS approximation stems from its natural connection to wavelets and their intrinsic property of multiresolution analysis (see [19] and references therein). Consider a function $u(\boldsymbol{x})$, and define two sets of MLS shape functions, $\boldsymbol{N}^h(\boldsymbol{x})$ and $\boldsymbol{N}^{2h}(\boldsymbol{x})$, computed using two different values of the smoothing length, h and $2h$, which respectively define h-scale and $2h$-scale approximations of the form

$$u_h(\boldsymbol{x}) = \sum_{j=1}^{n} u_j N_j^h(\boldsymbol{x}), \qquad u_{2h}(\boldsymbol{x}) = \sum_{j=1}^{n} u_j N_j^{2h}(\boldsymbol{x}). \tag{29}$$

A set of wavelet functions is obtained as

$$\boldsymbol{\Phi}^{2h}(\boldsymbol{x}) = \boldsymbol{N}^h(\boldsymbol{x}) - \boldsymbol{N}^{2h}(\boldsymbol{x}), \tag{30}$$

which allow the h-scale solution to be expressed as the sum of its low-scale and high-scale complementary parts, as

$$u_h(\boldsymbol{x}) = u_{2h}(\boldsymbol{x}) + \boldsymbol{\Psi}^{2h}(\boldsymbol{x}), \tag{31}$$

where

$$\boldsymbol{\Psi}^{2h}(\boldsymbol{x}) = \sum_{j=1}^{n} u_j \Phi_j^{2h}(\boldsymbol{x}) = \sum_{j=1}^{n} u_j (N_j^h(\boldsymbol{x}) - N_j^{2h}(\boldsymbol{x})). \tag{32}$$

The low-scale $u_{2h}(\boldsymbol{x})$ can be further decomposed using the same rationale. Fig. 8 presents a function $u(x)$, taken from a typical gas dynamics problem, and its corresponding high-scale component $\boldsymbol{\Psi}^{2h}$, obtained using several values of p. Clearly, $\boldsymbol{\Psi}^{2h}$ is a sensitive and powerful indicator of the smoothness of $u(x)$, that can be used as a shock detector or error sensor for adaptive and multiresolution algorithms. In this latter context, it is interesting that $\boldsymbol{\Psi}^{2h}$ is a "*single grid*" detector, that it is naturally suited for unstructured meshes, and that, for smooth functions, it converges

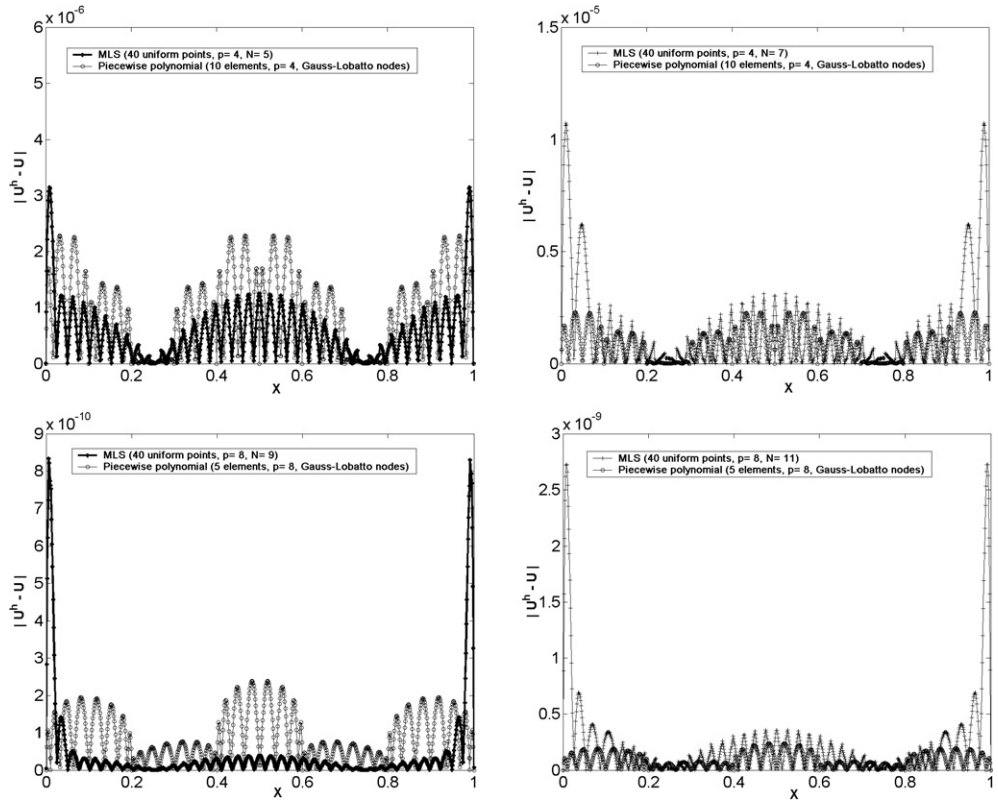

Fig. 6. Error distributions with MLS and piecewise polynomial interpolations, $u(x) = \sin(2\pi x)$. Top, $p = 4$, MLS with $N = 5$ (left) and $N = 7$ (right), 40 uniform points, Lagrange polynomials, 10 elements with Gauss–Lobatto nodes. Bottom, $p = 8$, MLS with $N = 9$ (left) and $N = 7$ (right), 40 uniform points, Lagrange polynomials, 5 elements with Gauss–Lobatto nodes.

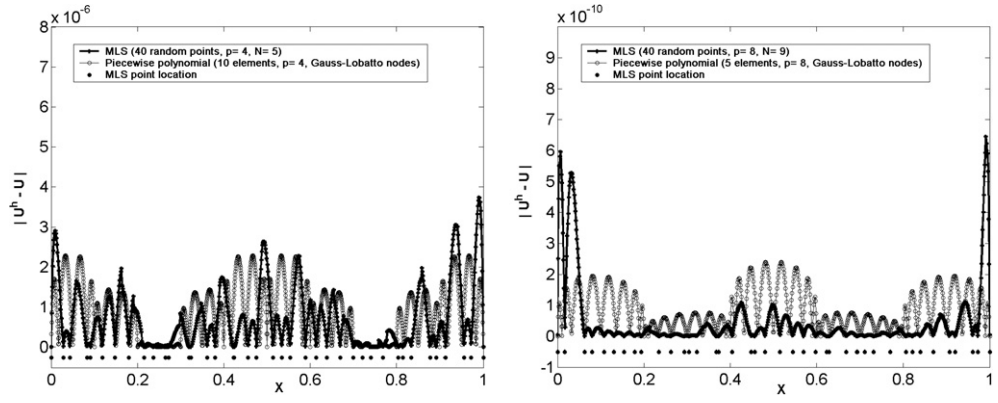

Fig. 7. Error distributions with MLS and piecewise polynomial interpolations, $u(x) = \sin(2\pi x)$. Left, $p = 4$, MLS with $N = 5$, 40 random points, Lagrange polynomials, 10 elements with Gauss–Lobatto nodes. Right, $p = 8$, MLS with $N = 9$, 40 random points, Lagrange polynomials, 5 elements with Gauss–Lobatto nodes.

to zero with the same order as u^h does, $p + 1$ (it is identically zero for polynomials of degree equal or less than p).

We believe that this multiresolution smoothness indicator, and its straightforward incorporation into a code that already uses MLS approximations (one only needs to compute another set of shape functions, but with $2h$ instead of h), is a very attractive feature of the proposed methodology. Even though well behaved limiters for second-order schemes have been developed, the question for higher-order reconstructions is far from being clear. Therefore, selective shock-capturing is a critical issue for higher-order schemes. If the limiters are active over the whole domain, their effect on higher-order derivatives results into a partial (or, quite frequently, complete) loss of the higher-order accuracy of the reconstruction in smooth regions of the flow, virtually taking the method back to second order.

As it is shown in one of the simulations in Section 6, the limiters can be switched off in those areas where $\mathbf{\Psi}^{2h}$ is lower than a certain threshold, therefore retaining the whole accuracy of the scheme in smooth regions. Note that

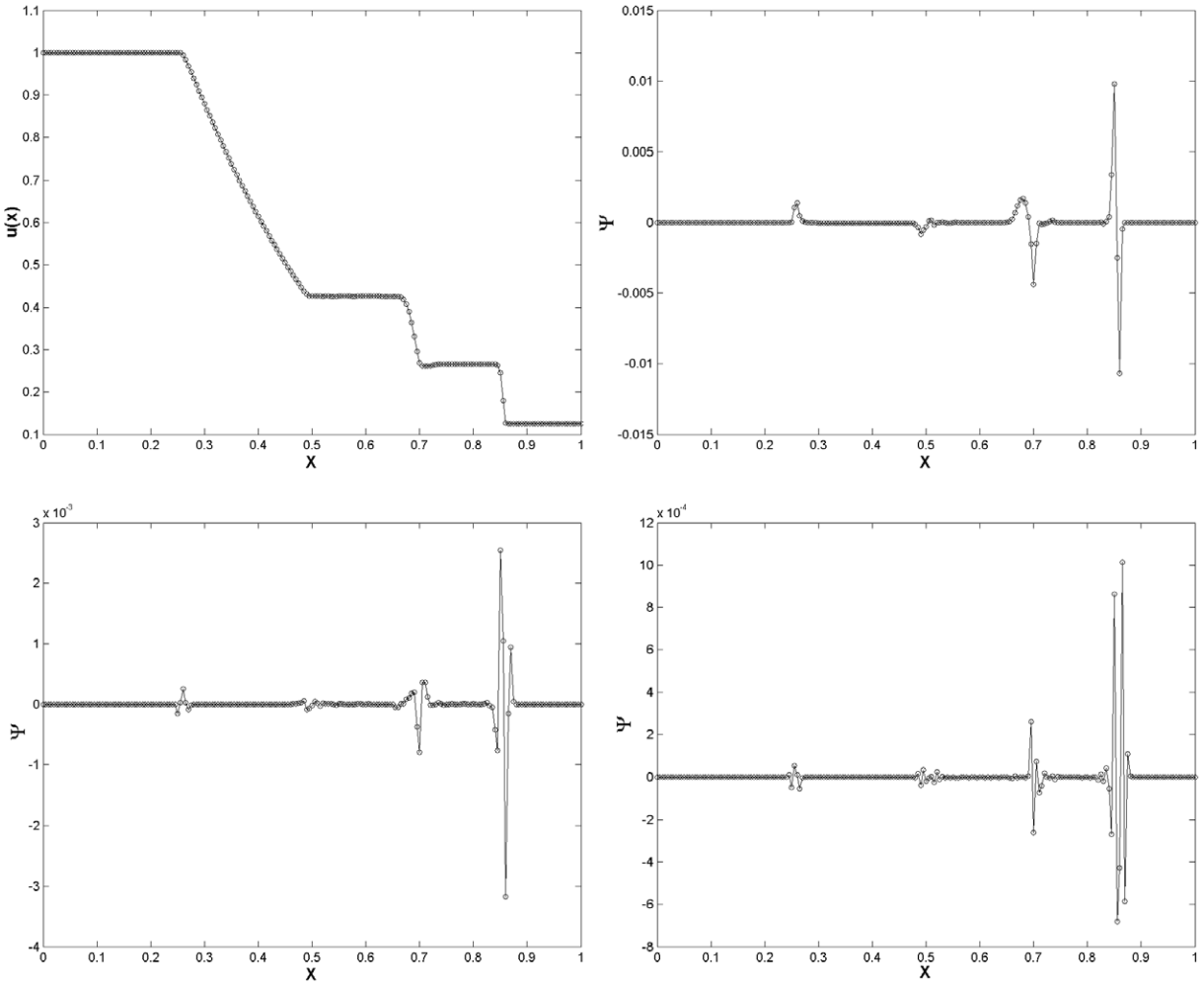

Fig. 8. Multiscale analysis: $u(x)$ (top left), and its high-scale component Ψ^{2h}, using $p = 1$, $N = 3$ (top right), $p = 3$, $N = 6$ (bottom left), and $p = 6$, $N = 9$ (bottom right).

the concept of "smooth region" itself is strongly related to the approximation being used, and hence the convenience of an indicator that is of the same order and nature as the approximants. In some sense, this procedure can be regarded as an unstructured grid generalization of the wavelet-based selective filtering proposed by Sjögreen and Yee for finite differences [20].

3.5. Moving Least-Squares, finite volume solvers, unstructured grids

This paper proposes the use of MLS approximations to construct high-order finite volume schemes on unstructured grids, and its scope is threefold:

- Moving Least-Squares approximants provide a general (continuous) approximation framework. For elliptic problems, or terms of elliptic character, this allows a straightforward, *direct* reconstruction of the fluxes at the interfaces. This procedure yields a single-valued, centered and high-order flux approximation at each edge quadrature point. For example, the viscous Navier–Stokes fluxes require the reconstruction of the conserved variables and their gradients at each quadrature point. We find connections between this approach and the successful second-order Multi-Point Flux Approximation (MPFA) methods developed by the petroleum engineering community (see [18] for an introduction).
- For hyperbolic problems, or terms of hyperbolic character, the generalized Godunov method [10,11,16] is adopted. We use "broken" piecewise polynomial reconstructions based on the MLS general approximation and Taylor series expansions. The successive derivatives of the flow variables at the cell centroids are computed using MLS approximations. Therefore, rather than creating new degrees of freedom inside each cell, we use information from neighbouring cells, in a *centered* (*moving*) fashion.
- The MLS-based multiresolution indicator provides a reliable shock-detection tool for the selective limiting of higher-order discretizations.

4. Practical implementation aspects

4.1. Overview

The following sections elaborate on the practical implementation of the proposed methodology. Conceptually, two aspects of the process should be distinguished:

- How the MLS shape functions and their derivatives are computed; in particular, the choice of the cloud of neighbours for each evaluation point (centroids or edge quadrature points). We call these clouds the *stencil* of the MLS approximation. This choice ultimately determines the *full stencil* of the finite volume method.
- How the MLS shape functions and their derivatives are used to (1) construct high-order reconstructions for a Godunov-type scheme for hyperbolic problems and to (2) directly reconstruct the "viscous" fluxes at the edges, thus obtaining a multipoint-like high-order scheme for elliptic problems.

Sections 4.2 and 4.3 present the MLS stencils used in this study for the cubic basis ($p = 3$).

4.2. $p = 3$ MLS stencils: I. Centroids

Fig. 9 presents the stencil used to compute the $p = 3$ MLS shape functions at the cell centroids. For and interior cell I, the stencil comprises its first and second neighbours (by neighbours we mean cells that share an edge). This gives a 13-point stencil. For boundary cells the stencil comprises those cells that share a vertex with the cell and their first neighbours. A stronger enforcement of the boundary conditions was achieved through the introduction of a set of "zero area" cells attached to the boundary (an approach analogous to the use of so-called *ghost cells* [15]). Note that the centroids of these boundary cells, i.e. the midpoints of those edges lying on the boundary, have been included in the above stencils. During the simulation, the variables at these locations will be either extrapolated or assigned a certain value, depending on the type of boundary condition to be enforced.

4.3. $p = 3$ MLS stencils: II. Edges

Fig. 10 presents the stencil to compute the $p = 3$ MLS shape functions at the edge quadrature points. Given a quadrature point lying in the interface between cells A and B, its stencil comprises those cells sharing the extremum vertices of the edge, and their first neighbours. If both vertices are shared by 4 cells, this is a 16-point stencil. For boundary cells we also include the neighbours of the edge opposite to the boundary, and the corresponding ghost cells.

4.4. Comments on the full stencil of the finite volume scheme

The "inviscid" stencil of a cell I is obtained as the union of its MLS stencil, and the MLS stencils of its first neighbours. Fig. 11(left) depicts the $p = 3$ inviscid stencil for interior cells, which comprises 25 cells. This stencil can be used to construct a fourth-order scheme for the Euler equations.

Analogously, the stencil of the "viscous" discretization is obtained as the union of the MLS stencils associated to all the edges of cell I. Fig. 11(right) depicts the $p = 3$ viscous stencil for interior cells, which comprises 21 cells. This is the stencil of a fourth-order scheme for elliptic problems. The full stencil for Navier–Stokes computations coincides with the inviscid one, as the latter includes the viscous stencil as a subset. Note that, quite the opposite to what is usually thought about finite volume schemes, this stencil is actually quite compact. A comparison with DG in terms of accuracy for the same grid resolution, storage and compactness is presented in [22].

4.5. Specific techniques for hyperbolic terms: "broken" reconstruction and limiting

Reconstruction is usually addressed in finite volume schemes as a bottom-up process, by substituting the piecewise constant representation of the basic first-order scheme by a piecewise polynomial reconstruction of the field variables inside each control volume. In practice, the development of very high-order schemes of this kind has been

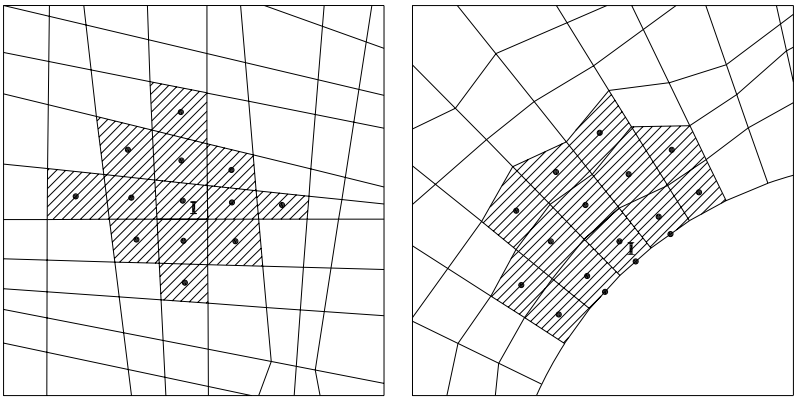

Fig. 9. $p = 3$, MLS stencil: centroids.

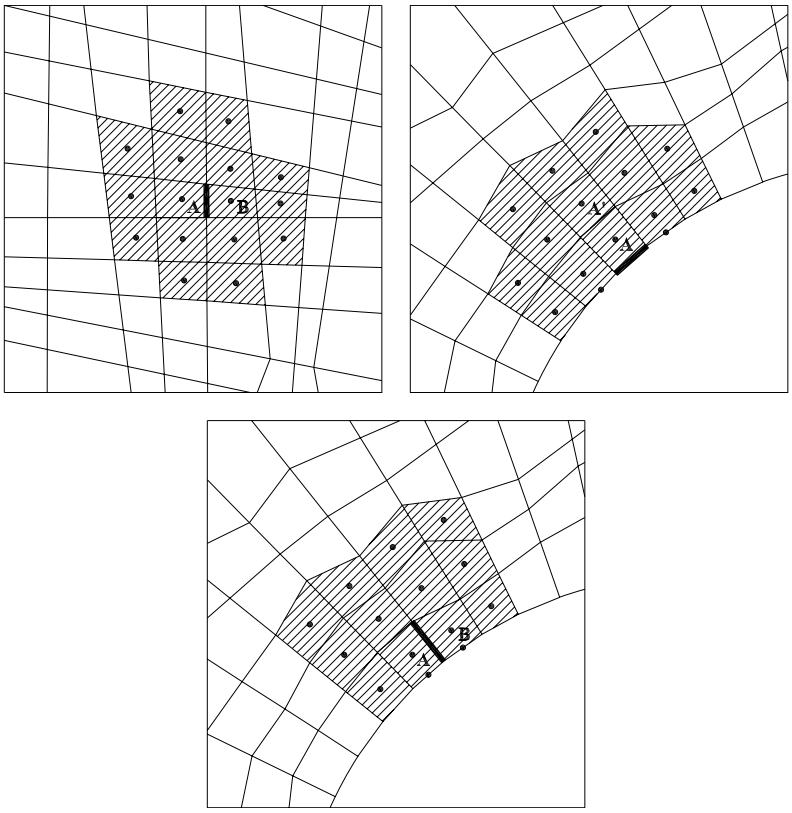

Fig. 10. $p = 3$, MLS stencil: quadrature points on edges.

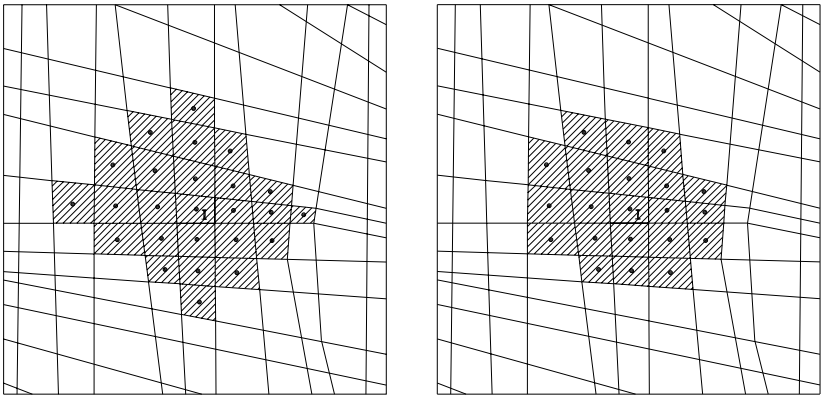

Fig. 11. Fourth-order MLS-FV stencil: Euler and Navier–Stokes (left) and elliptic problems (right).

severely limited by the absence of robust approximation techniques, capable of computing accurate estimates of the successive derivatives of the field variables in the context of unstructured grids [16,17]. Thus, the concept "high-order scheme" is most frequently used in the literature in reference to formally second-order schemes (piecewise linear reconstruction).

In contrast, our approach is top-down, as we define a general continuous approximation framework provided by the Moving Least-Squares approximants, and then compute local discontinuous approximations, which are broken high-order approximations to the underlying continuous solution, to be used in the context of a Godunov-type scheme. In this study, reconstructions of up to fourth-order (cubic) have been tested, although schemes of up to sixth order are expected to be practical in the near future.

The linear component-wise reconstruction of the variables inside cell I reads

$$U(\boldsymbol{x}) = U_I + \nabla U_I \cdot (\boldsymbol{x} - \boldsymbol{x}_I), \tag{33}$$

where U_I stands for the centroid value, $\boldsymbol{x}_I$ denotes spatial coordinates of the centroid of the cell and ∇U_I is a cell-centered gradient. This gradient is assumed to be constant on each cell and, therefore, the reconstructed variables are discontinuous across interfaces.

Analogously, the quadratic reconstruction reads

$$U(\boldsymbol{x}) = U_I + \nabla U_I \cdot (\boldsymbol{x} - \boldsymbol{x}_I) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_I)^{\mathrm{T}} \boldsymbol{H}_I (\boldsymbol{x} - \boldsymbol{x}_I), \tag{34}$$

where $\boldsymbol{H}_I$ is the centroid hessian matrix. Finally, the cubic reconstruction can be written as

$$U(\boldsymbol{x}) = U_I + \nabla U_I \cdot (\boldsymbol{x} - \boldsymbol{x}_I) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_I)^{\mathrm{T}} \boldsymbol{H}_I (\boldsymbol{x} - \boldsymbol{x}_I) + \frac{1}{6}\Delta^2 \boldsymbol{x}_I^{\mathrm{T}} \boldsymbol{T}_I (\boldsymbol{x} - \boldsymbol{x}_I), \tag{35}$$

where

$$\Delta^2 \boldsymbol{x}_I^{\mathrm{T}} = \begin{pmatrix} (x - x_I)^2 & (y - y_I)^2 \end{pmatrix}, \qquad \boldsymbol{T}_I = \begin{pmatrix} \frac{\partial^3 U_I}{\partial x^3} & 3\frac{\partial^3 U_I}{\partial x^2 \partial y} \\ 3\frac{\partial^3 U_I}{\partial x \partial y^2} & \frac{\partial^3 U_I}{\partial y^3} \end{pmatrix}. \tag{36}$$

For unsteady problems, additional terms must be introduced in (34) and (35) to enforce conservation of the mean, i.e.

$$\frac{1}{A_I}\int_{x \in \Omega_I} U(\boldsymbol{x})\,\mathrm{d}\Omega = U_I. \tag{37}$$

The derivatives of the field variables are directly computed at centroids using MLS. Thus, the approximate gradients read

$$\nabla U_I = \sum_{j=1}^{n_{x_I}} U_j \nabla N_j(\boldsymbol{x}_I), \tag{38}$$

where the U_j's stand for variables at the n_{x_I} "neighbour" (in the sense of the MLS stencil) centroids. The second-order derivatives read

$$\frac{\partial^2 U_I}{\partial x^2} = \sum_{j=1}^{n_{x_I}} U_j \frac{\partial^2 N_j(\boldsymbol{x}_I)}{\partial x^2}, \qquad \frac{\partial^2 U_I}{\partial x \partial y} = \sum_{j=1}^{n_{x_I}} U_j \frac{\partial^2 N_j(\boldsymbol{x}_I)}{\partial x \partial y},$$
$$\frac{\partial^2 U_I}{\partial y^2} = \sum_{j=1}^{n_{x_I}} U_j \frac{\partial^2 N_j(\boldsymbol{x}_I)}{\partial y^2}. \tag{39}$$

Finally, the third-order derivatives are written as

$$\frac{\partial^3 U_I}{\partial x^3} = \sum_{j=1}^{n_{x_I}} U_j \frac{\partial^3 N_j(\boldsymbol{x}_I)}{\partial x^3}, \qquad \frac{\partial^3 U_I}{\partial x^2 \partial y} = \sum_{j=1}^{n_{x_I}} U_j \frac{\partial^3 N_j(\boldsymbol{x}_I)}{\partial x^2 \partial y},$$
$$\frac{\partial^3 U_I}{\partial x \partial y^2} = \sum_{j=1}^{n_{x_I}} U_j \frac{\partial^3 N_j(\boldsymbol{x}_I)}{\partial x \partial y^2}, \qquad \frac{\partial^3 U_I}{\partial y^3} = \sum_{j=1}^{n_{x_I}} U_j \frac{\partial^3 N_j(\boldsymbol{x}_I)}{\partial y^3}. \tag{40}$$

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

In the presence of shocks, some limiting procedure is applied to the above derivatives. The choice of adequate multidimensional limiters is critical in order to achieve accurate and non-oscillatory shock-capturing algorithms.

4.5.1. Limiters: I. Monotonicity enforcement

Barth and Jespersen [12] have proposed an extension of Van Leer's scheme [25] which is suitable for unstructured grids. The basic idea is to enforce "*monotonicity*" in the reconstructed solution. In this context, monotonicity implies that no new extrema are created by the reconstruction process [12]. The enforcement is local, in the sense that only certain neighbour cells are considered for the "no new extrema" criterion.

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

$$U(\boldsymbol{x})_I = U_I + \nabla U_I \cdot (\boldsymbol{x} - \boldsymbol{x}_I) \tag{41}$$

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

$$U(\boldsymbol{x})_I = U_I + \Phi_I \nabla U_I \cdot (\boldsymbol{x} - \boldsymbol{x}_I), \tag{42}$$

where Φ_I is a slope limiter ($0 \leqslant \Phi_I \leqslant 1$) such that the reconstruction (42) satisfies

$$U^{\min} \leqslant U(\boldsymbol{x})_I \leqslant U^{\max} \tag{43}$$

being

$$U^{\min} = \min_{j \in \mathscr{A}_I}(U_j), \qquad U^{\max} = \max_{j \in \mathscr{A}_I}(U_j), \tag{44}$$

where $\mathscr{A}_I$ is the set of "neighbour" cells. In practice, the restriction (43) is only enforced at the quadrature points on the edges of cell I; thus, for each quadrature point q, its associated slope limiter Φ_I^q is computed in terms of the unlimited extrapolated value U_I^q, as

$$\Phi_I^q = \begin{cases} \min\left(1, \frac{U^{\max} - U_I}{U_I^q - U_I}\right) & U_I^q - U_I > 0, \\ \min\left(1, \frac{U^{\min} - U_I}{U_I^q - U_I}\right) & U_I^q - U_I < 0, \\ 1 & U_I^q - U_I = 0 \end{cases} \tag{45}$$

and, finally,

$$\Phi_I = \min_q(\Phi_I^q). \tag{46}$$

In the case of the quadratic reconstruction (34), a similar limiting strategy is adopted

$$U(\boldsymbol{x}) = U_I + \Phi_I\left(\nabla U_I \cdot (\boldsymbol{x} - \boldsymbol{x}_I) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_I)^{\mathrm{T}} \boldsymbol{H}_I (\boldsymbol{x} - \boldsymbol{x}_I)\right), \tag{47}$$

where the limiter Φ_I is obtained following the same procedure exposed above for the linear case. An analogous expression can be used for the cubic reconstruction.

In this study the neighbourhood to determine the extrema $U^{\min}$ and $U^{\max}$ comprises the reconstruction cell I and its first-order neighbours (Fig. 12A). In the following, the above limiter will be referred to as "*BJ limiter*".

4.5.2. Limiters: II. Averaged derivatives

This section presents a general strategy to obtain limited gradients and hessian matrices. Thus, the limited gradient

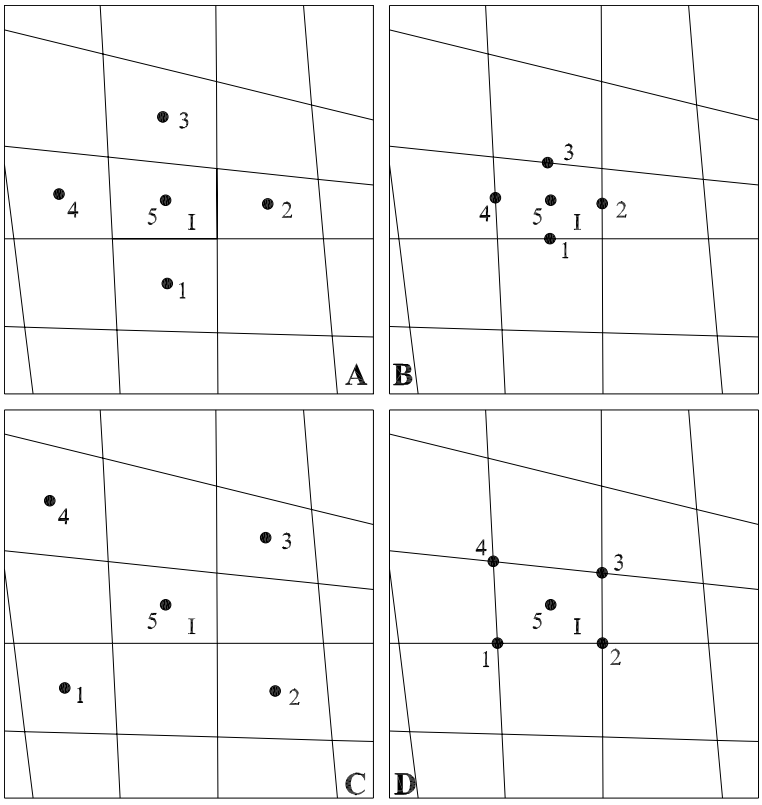

Fig. 12. Neighbourhoods for the limiting of the reconstruction inside cell I.

associated to a certain cell I, ∇U_I is obtained as a weighted average of a series of *representative* gradients, as

$$\nabla U_I = \sum_{k=1}^{N} \omega_k \nabla U_k, \tag{48}$$

where $\{\nabla U_k,\ k = 1,\ldots,N\}$ is a set of unlimited gradients, used as a basis to construct the limited one. In an approach similar to that exposed in [15], the weights $\{\omega_k,\ k = 1,\ldots,N\}$ are given by

$$\omega_k(g_1, g_2, \ldots, g_N) = \frac{\prod_{i\neq k}^{N} g_i + \epsilon^{N-1}}{\sum_{j=1}^{N}\left(\prod_{i\neq j}^{N} g_i\right) + N\epsilon^{N-1}}, \quad k = 1,\ldots,N, \tag{49}$$

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

$$g_i = \left(\frac{\partial^2 U_i}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 U_i}{\partial x\,\partial y}\right)^2 + \left(\frac{\partial^2 U_i}{\partial y^2}\right)^2, \quad i = 1,\ldots,N. \tag{50}$$

Some existing limiters could be considered to be included in this family. Van Rosendale [26] has proposed an extension to three gradients of Van Albada's limiter [27]. This limiter was used on unstructured triangular grids and its general structure is that of (48) with $N = 3$. The representative gradients are evaluated at the three *vertices* of the cell. Jawahar and Kamath [15] proposed a limiter with $N = 3$, with averaged gradients computed from the unlimited gradients evaluated at the *centroids* of the adjacent cells on triangular meshes. Furthermore, the denominators in (49) are slightly different in this case.

For quadrilateral cells we propose a limiter based on (48) and (49) with $N = 5$; i.e. the limited derivatives are obtained as a weighted average of five unlimited derivatives. Fig. 12 presents four suitable configurations to determine such *representative derivatives*. In this study only the configuration given by 12A will be considered. In the following, the above limiter will be referred to as "*PC5 limiter*".

4.6. Numerical convective fluxes

The numerical inviscid fluxes in (13) are obtained using Roe's flux difference splitting [28]. For this purpose, left ($\boldsymbol{U}^+$) and right ($\boldsymbol{U}^-$) states are defined on each face. The numerical flux is then computed as

$$(\boldsymbol{F}_x, \boldsymbol{F}_y)\cdot\boldsymbol{n} = \frac{1}{2}\left[(\boldsymbol{F}_x(\boldsymbol{U}^+), \boldsymbol{F}_y(\boldsymbol{U}^+)) + (\boldsymbol{F}_x(\boldsymbol{U}^-), \boldsymbol{F}_y(\boldsymbol{U}^-))\right]\cdot\boldsymbol{n} - \frac{1}{2}\sum_{k=1}^{3}\tilde{\alpha}_k|\tilde{\lambda}_k|\tilde{\boldsymbol{r}}_k, \tag{51}$$

where $\{\tilde{\lambda}_k,\ k=1,4\}$ and $\{\tilde{\boldsymbol{r}}_k,\ k=1,4\}$ are, respectively, the eigenvalues and eigenvectors of the approximate jacobian $\widetilde{\boldsymbol{J}}(\boldsymbol{U}^+,\boldsymbol{U}^-)$

$$\tilde{\lambda}_1=\tilde{\boldsymbol{v}}\cdot\boldsymbol{n}-\tilde{c},\quad \tilde{\lambda}_2=\tilde{\lambda}_3=\tilde{\boldsymbol{v}}\cdot\boldsymbol{n},\quad \tilde{\lambda}_4=\tilde{\boldsymbol{v}}\cdot\boldsymbol{n}+\tilde{c}, \tag{52}$$

$$(\tilde{\boldsymbol{r}}_1\tilde{\boldsymbol{r}}_2\tilde{\boldsymbol{r}}_3\tilde{\boldsymbol{r}}_4)=\begin{pmatrix} 1 & 0 & 1 & 0\\ \tilde{u}-\tilde{c}n_x & -\tilde{c}n_y & \tilde{u} & \tilde{u}+\tilde{c}n_x\\ \tilde{v}-\tilde{c}n_y & \tilde{c}n_x & \tilde{v} & \tilde{v}+\tilde{c}n_y\\ \widetilde{H}-\tilde{c}\tilde{\boldsymbol{v}}\cdot\boldsymbol{n} & \tilde{c}(\tilde{v}n_x-\tilde{u}n_y) & \frac{1}{2}(\tilde{u}^2+\tilde{v}^2) & \widetilde{H}+\tilde{c}\tilde{\boldsymbol{v}}\cdot\boldsymbol{n}\end{pmatrix} \tag{53}$$

and the corresponding wave strengths $\{\tilde{\alpha}_k,\ k=1,4\}$

$$\begin{aligned}
\tilde{\alpha}_1&=\frac{1}{2\tilde{c}^2}[\Delta(p)-\tilde{\rho}\tilde{c}(\Delta(u)n_x+\Delta(v)n_y)],\\
\tilde{\alpha}_2&=\frac{\tilde{\rho}}{\tilde{c}}[\Delta(v)n_x-\Delta(u)n_y],\\
\tilde{\alpha}_3&=-\frac{1}{\tilde{c}^2}[\Delta(p)-\tilde{c}^2\Delta(\rho)],\\
\tilde{\alpha}_4&=\frac{1}{2\tilde{c}^2}[\Delta(p)+\tilde{\rho}\tilde{c}(\Delta(u)n_x+\Delta(v)n_y)],
\end{aligned} \tag{54}$$

where $\Delta(\cdot)=(\cdot)^--(\cdot)^+$, $\boldsymbol{n}=(n_x,n_y)$ is the outward pointing unit normal to the interface, and the Roe-average values $\tilde{\boldsymbol{v}}=(\tilde{u},\tilde{v})$ and $\widetilde{H}$ (computed using $\boldsymbol{U}^+$ and $\boldsymbol{U}^-$) are defined as

$$\begin{aligned}
\tilde{u}&=\frac{u^+\sqrt{\rho^+}+u^-\sqrt{\rho^-}}{\sqrt{\rho^+}+\sqrt{\rho^-}},\qquad \tilde{v}=\frac{v^+\sqrt{\rho^+}+v^-\sqrt{\rho^-}}{\sqrt{\rho^+}+\sqrt{\rho^-}},\\
\widetilde{H}&=\frac{H^+\sqrt{\rho^+}+H^-\sqrt{\rho^-}}{\sqrt{\rho^+}+\sqrt{\rho^-}}.
\end{aligned} \tag{55}$$

On the other hand, the average values $\tilde{\rho}$ and $\tilde{c}$ are computed as

$$\tilde{\rho}=\sqrt{\rho^+\rho^-},\qquad \tilde{c}^2=(\gamma-1)\left[\widetilde{H}-\frac{1}{2}\left(\tilde{u}^2+\tilde{v}^2\right)\right]. \tag{56}$$

4.7. Viscous fluxes

As mentioned before, one of the major advantages of the proposed method is that we use the MLS approximants as a global (centered) reconstruction procedure to evaluate the viscous fluxes at the quadrature points on the edges. This procedure provides a single high-order flux and, therefore, it is not necessary to create new degrees of freedom to compute the derivatives of the variables at the cell edges.

Recall that the evaluation of the viscous stresses and heat fluxes requires interpolating the velocity vector $\boldsymbol{v}=(u,v)$, temperature T, and their corresponding gradients, $\nabla\boldsymbol{v}$ and ∇T, at each quadrature point $\boldsymbol{x}_{iq}$. Using MLS approximation, these entities are readily computed as

$$\boldsymbol{v}_{iq}=\sum_{j=1}^{n_{iq}}\boldsymbol{v}_jN_j(\boldsymbol{x}_{iq}),\qquad T_{iq}=\sum_{j=1}^{n_{iq}}T_jN_j(\boldsymbol{x}_{iq}) \tag{57}$$

and

$$\nabla\boldsymbol{v}_{iq}=\sum_{j=1}^{n_{iq}}\boldsymbol{v}_j\otimes\nabla N_j(\boldsymbol{x}_{iq}),\qquad \nabla T_{iq}=\sum_{j=1}^{n_{iq}}T_j\nabla N_j(\boldsymbol{x}_{iq}), \tag{58}$$

where n_{iq} is the number of neighbour centroids (in the sense of the MLS stencil). Once the above information has been interpolated, the diffusive fluxes can be computed, according to (3).

4.8. Flux integration

One quadrature point (the midpoint) was used in the case of linear reconstruction, whereas two and three Gauss points were respectively used in the case of quadratic and cubic reconstructions.

4.9. Time integration

We use the third-order TVD-Runge–Kutta algorithm proposed by Shu and Osher [29]. Given the field variables U^n at the previous time step n, the algorithm proceeds in three stages to obtain the updated field variables U^{n+1}, as

$$\begin{aligned}
U^1&=U^n+\Delta tL(U^n),\\
U^2&=\frac{3}{4}U^n+\frac{1}{4}U^1+\frac{1}{4}\Delta tL(U^1),\\
U^{n+1}&=\frac{1}{3}U^n+\frac{2}{3}U^2+\frac{2}{3}\Delta tL(U^2),
\end{aligned} \tag{59}$$

where the operator $L(\cdot)$, which represents the time derivative given by (13), reads

$$L(\boldsymbol{U})=\frac{1}{A}\sum_{iedge=1}^{nedge}\sum_{igau=1}^{ngau}[(\mathscr{F}^V-\mathscr{F})\cdot\boldsymbol{n}]_{igau}\mathscr{W}_{igau}. \tag{60}$$

5. Accuracy tests

This section presents some convergence results of the proposed finite volume method with Moving Least-Squares approximations. The tests are intended to assess the performance of the methodology with respect to two distinct areas of its scope: high-order variable reconstruction for Godunov-type schemes, and high-order, multipoint viscous flux evaluation.

5.1. Hyperbolic problems: Ringleb flow

Ringleb flow is an exact solution of the Euler equations, obtained by means of the hodograph method [30]. The problem is solved on the square $[-1.15,-0.75]\times[+0.15,+0.55]$, imposing the exact value of the conserved variables on the boundary. Linear, quadratic and cubic reconstructions are developed by means of MLS derivatives, as exposed above. A refinement study was carried out using a sequence of four nested grids, the coarsest of which is

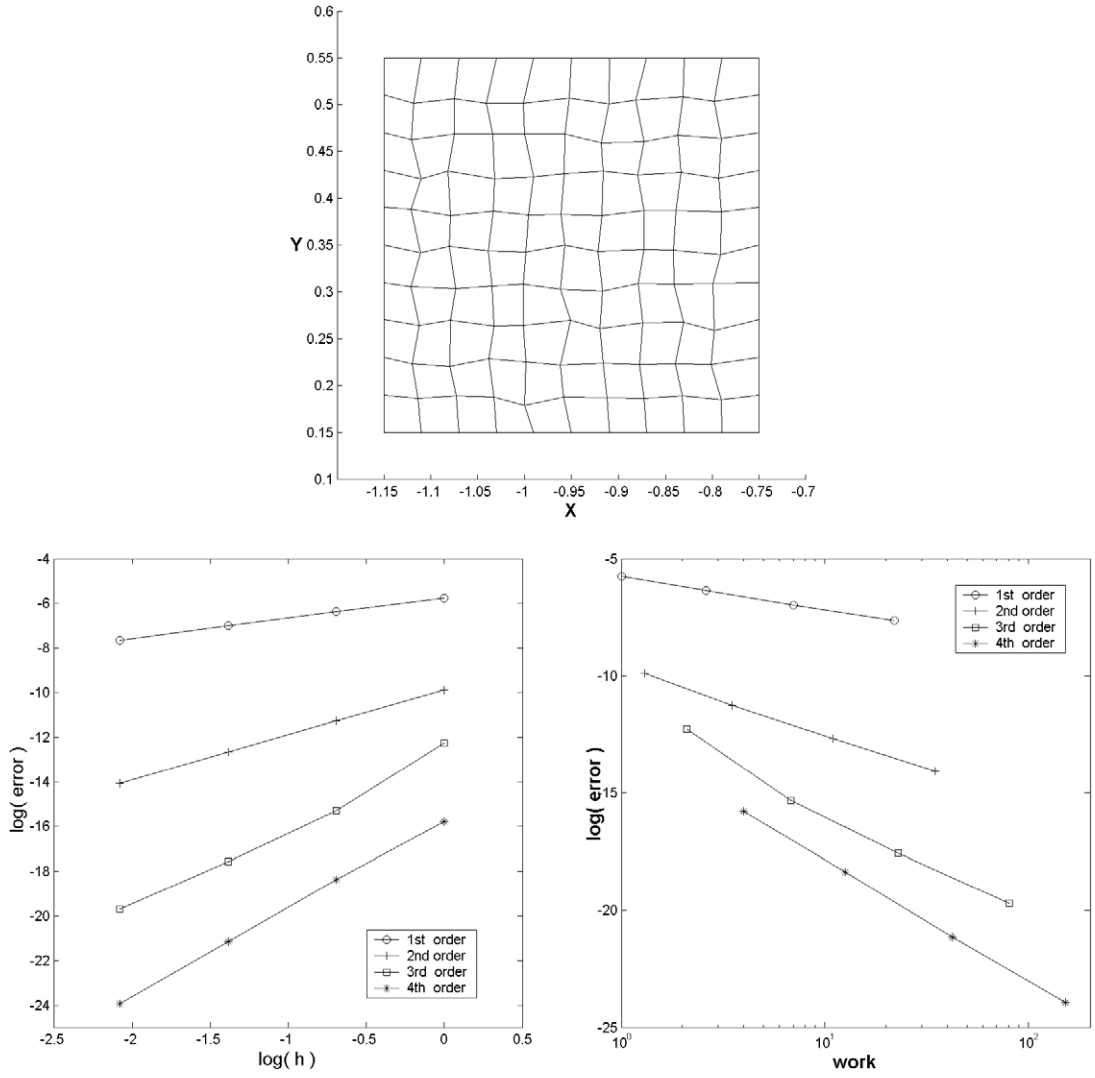

Fig. 13. Coarse grid level and convergence results for Ringleb flow.

showed in Fig. 13(top), along with the convergence curves, which are broken down in Table 1.

All linear, quadratic and cubic reconstructions exhibit the correct second, third and fourth-orders of convergence, respectively, as expected. One, two, and three Gauss quadrature points per edge have been employed for the second, third, and fourth-order schemes, respectively. In addition, Fig. 13(bottom right) presents a comparison of the different reconstructions with respect to accuracy vs. workload. The cpu times are expressed in terms of time units per time step of the Runge–Kutta integrator, and normalized with respect to the cpu time associated to a time step of the first-order scheme (no reconstruction) on the 10×10 grid, which is taken as the reference workload, $Work = 1$. The benefits and efficiency of the higher-order reconstructions are quite apparent. Comparing the second and fourth-order reconstructions, for example, we see that, for the same grid, the accuracy of the latter is about three orders of magnitude higher than that of the former, with a cpu increase of a factor of four. Moreover, most of the additional cpu time associated to the fourth-order scheme is due to the use of three quadrature points per edge, and

Table 1
Fourth-order results

Grid	Linear rec.			Quadratic rec.			Cubic rec.		
	Work	L_2 error	Slope	Work	L_2 error	Slope	Work	L_2 error	Slope
10×10	1.3	5.04×10^{-5}		2.1	4.71×10^{-6}		4	1.39×10^{-7}	
20×20	3.5	1.28×10^{-5}	1.98	6.8	2.23×10^{-7}	4.40	12.6	1.06×10^{-8}	3.71
40×40	11	3.14×10^{-6}	2.03	23	2.34×10^{-8}	3.25	42.5	6.60×10^{-9}	4.01
80×80	35	7.81×10^{-7}	2.01	80.5	2.80×10^{-9}	3.06	152	4.07×10^{-10}	4.02

therefore more flux evaluations, and not to the higher-order reconstruction itself.

5.2. Viscous discretization. A first "worked" example: 1D Poisson

This first "viscous" example is intended to provide a flavour of the power of the proposed method for the numerical solution of elliptic equations, and for viscous flux evaluation as a particular application. The finite volume discretization of the 1D boundary-value problem

$$\frac{\mathrm{d}}{\mathrm{d}x}\left(-\kappa\frac{\mathrm{d}u}{\mathrm{d}x}\right)=f(x),\quad x\in[0,1], \tag{61}$$

$$u(0)=0,\quad u(1)=0, \tag{62}$$

reads, for each control volume Ω_i

$$q_{i+1/2}-q_{i-1/2}=\int_{\Omega_i} f(x)\,\mathrm{d}x, \tag{63}$$

where $q_{i+1/2}$ (resp. $q_{i-1/2}$) is the numerical flux evaluated at the interface between control volumes i and $i+1$ (resp. i and $i-1$), i.e. a suitable approximation to $-\kappa\frac{\mathrm{d}u}{\mathrm{d}x}\big|_{i+1/2}$ (resp. $-\kappa\frac{\mathrm{d}u}{\mathrm{d}x}\big|_{i-1/2}$). Computing the gradients of the variables directly at the midpoints using MLS approximation, $q_{i+1/2}$ and $q_{i-1/2}$ can be expressed in a multi-point fashion, as

$$q_{i+1/2}=-\kappa\sum_{j=1}^{N^+}u_j\frac{\mathrm{d}N_j}{\mathrm{d}x}\bigg|_{i+1/2},\qquad q_{i-1/2}=-\kappa\sum_{j=1}^{N^-}u_j\frac{\mathrm{d}N_j}{\mathrm{d}x}\bigg|_{i-1/2}, \tag{64}$$

where for the sake of simplicity, κ was assumed to be constant, and N^+ (resp. N^-) is the number of "neighbours" of point $i+1/2$ (resp. $i-1/2$). Denoting by S^+ (resp. S^-) the cloud associated to $i+1/2$ (resp. $i-1/2$), the stencil of the finite volume method given by (63) is $S^+\cup S^-$ (Fig. 14). For centered clouds and $N^+=N^-=N$, the stencil of an interior cell comprises $N+1$ neighbours, since S^+ and S^- overlap (Fig. 14).

With all the above in mind, the model problem (61) is solved with $\kappa=1$ and $f(x)=4\pi^2\sin(2\pi x)$, which corresponds to $u(x)=\sin(2\pi x)$.

Note that only diffusive fluxes are present and, therefore, special attention must be paid to the proper enforcement of the Dirichlet boundary conditions. To this end, a ghost cell is placed at each boundary, $x=0$ and $x=1$. These ghost cells are given the exact boundary value, and included in the cloud generation process. This (somewhat weak) enforcement of the Dirichlet boundary conditions has proved to be very effective in the elliptic problems analyzed in this study.

The fully discrete equations constitute a linear system of equations of the form

$$\sum_{j=1}^{N^+}u_j\frac{\mathrm{d}N_j}{\mathrm{d}x}\bigg|_{i+1/2}-\sum_{j=1}^{N^-}u_j\frac{\mathrm{d}N_j}{\mathrm{d}x}\bigg|_{i-1/2}=2\pi(\cos(2\pi x_{i+1/2})-\cos(2\pi x_{i-1/2})) \tag{65}$$

for each control volume i. The coefficient matrix is straightforwardly assembled once the "cloud" derivatives $\frac{\mathrm{d}N_j}{\mathrm{d}x}\big|_{i+1/2}$ and $\frac{\mathrm{d}N_j}{\mathrm{d}x}\big|_{i-1/2}$ are computed at the midpoints of each control volume. The bandwith of this matrix is obviously $N+1$ (see Fig. 14). Note that the cloud associated to each midpoint is unique and, therefore, the numerical flux is also unique at each midpoint, which yields a conservative scheme. This feature of uniqueness of the computed flux is quite convenient, in the sense that higher-order schemes can be achieved without decomposing the original second-order problem into a set of several first-order ones.

The convergence results for various MLS orders are presented in Table 2, in terms of the order of the polynomial basis, p, and number of neighbours for each midpoint, N.

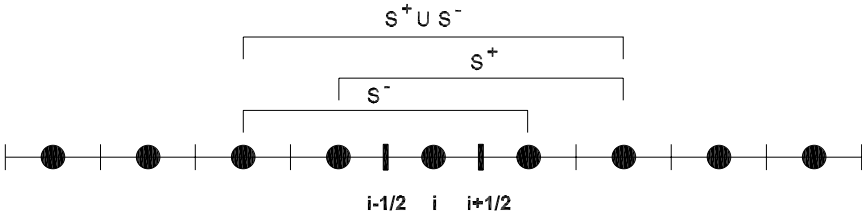

Fig. 14. Stencil for multi-point MLS-based viscous flux computation.

Table 2
Convergence results for viscous discretizations: 1D Poisson equation

Cells	MLS order (N = number of neighbours)							
	$p=3$ ($N=6$)		$p=4$ ($N=6$)		$p=5$ ($N=8$)		$p=6$ ($N=9$)	
	Error	Slope	Error	Slope	Error	Slope	Error	Slope
10	1.20×10^{-2}		8.02×10^{-4}		3.08×10^{-3}		4.16×10^{-4}	
20	3.91×10^{-4}	4.94	3.31×10^{-5}	4.60	3.52×10^{-5}	6.45	1.55×10^{-6}	8.07
40	2.00×10^{-5}	4.29	8.53×10^{-7}	5.28	3.60×10^{-7}	6.61	1.31×10^{-8}	6.89
80	1.39×10^{-6}	3.85	2.01×10^{-8}	5.41	4.33×10^{-9}	6.38	1.15×10^{-10}	6.83

Table 3
Convergence results for the 1D Poisson equation: tenth order scheme

Cells	MLS order (N)	
	$p = 9$ ($N = 12$)	
	Error	Slope
20	1.32×10^{-7}	
30	2.16×10^{-9}	10.16
40	1.14×10^{-10}	10.22

The expected $p + 1$ convergence order is achieved in all cases considered. Recall that, as mentioned before, the stencil for interior cells is $N + 1$. As a final example, Table 3 presents the convergence results for $p = 9$. A tenth-order scheme is realized, with a stencil of 13 and just one degree of freedom per cell.

5.3. Viscous discretization: 2D Darcy flow

Consider the elliptic equation

$$\begin{aligned} &\nabla \cdot (-\boldsymbol{K}\nabla p) = f \quad \text{in } \Omega, \\ &p = p^D \quad \text{on } \Gamma^D, \\ &\boldsymbol{v} \cdot \boldsymbol{n} = h \quad \text{on } \Gamma^N, \end{aligned} \tag{66}$$

which is a prototype for the pressure equation in porous media flow, $\boldsymbol{K}$ is the permeability tensor and $\boldsymbol{v} = -\boldsymbol{K}\nabla p$ is the Darcy velocity. The problem (66) is solved in $[0,+1]\times[0,1+]$ using the $p = 3$ MLS-FV scheme, with $k_{11} = k_{22} = 1$, $k_{12} = k_{21} = 0$, and a source term and boundary conditions such that $p = \sin(2\pi x)\sin(2\pi y)$. The exact Dirichlet boundary condition is enforced on Γ. A refinement study was carried out on a sequence of uniform and random grids. The coarse random grid and first level refinement are plotted in Fig. 15. The convergence results are presented in Fig. 16 and Table 4. Orders 4 and 3.5 in pressure and velocity, respectively, are realized for both the uniform and irregular grids. Velocities seem to be slightly superconvergent in the gradient (3.5 instead of 3), but this phenomenon requires further analysis.

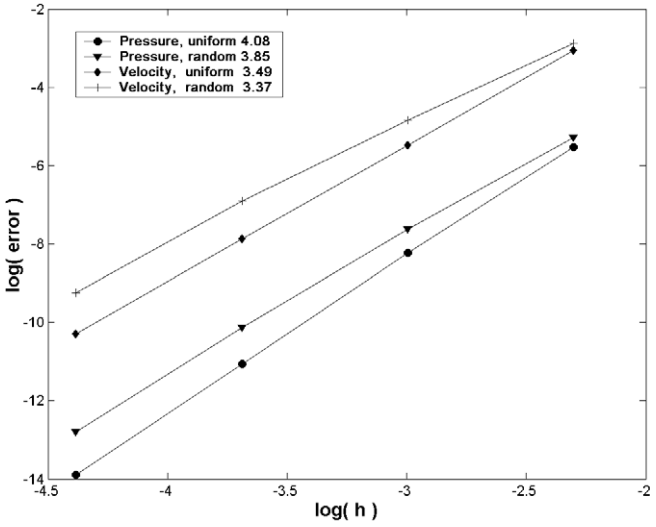

Fig. 16. 2D Darcy flow: convergence for the $p = 3$ MLS-FV scheme.

6. Representative simulations

6.1. Selective limiting and the multiresolution detector

This 1D Euler example intends to provide some insight into the behaviour of the limited higher-order reconstructions in the presence of shocks and smooth flow regions. In addition, it serves as an example of the performance of the multiresolution-based selective limiting procedure exposed in Section 3.4. The Shu–Osher case [31] is solved using 400 cells in $[-5,+5]$, with initial conditions

$$\begin{aligned} &(\rho_R, u_R, p_R) = (3.857, 2.629, 10, 333), \\ &(\rho_L, u_L, p_L) = (1 + 0.2\sin(5x), 0, 1). \end{aligned} \tag{67}$$

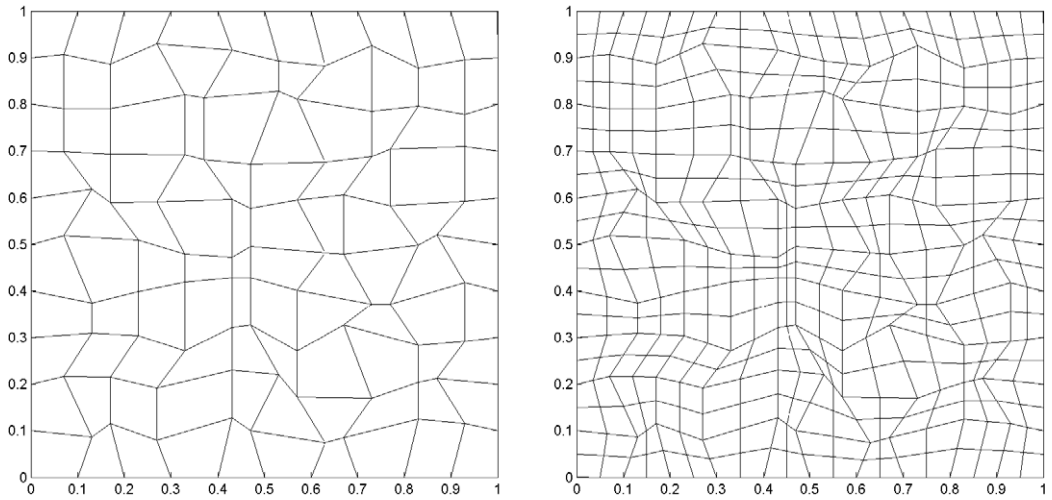

Fig. 15. Coarse random grid for 2D Darcy flow (left) and one level refinement (right).

Table 4
Convergence results for viscous discretizations: Darcy flow ($p = 3$ MLS-FV scheme)

Grid	Pressure				Velocity			
	Uniform		Random		Uniform		Random	
	Error	Slope	Error	Slope	Error	Slope	Error	Slope
10×10	3.98×10^{-3}		5.08×10^{-3}		4.73×10^{-2}		5.66×10^{-2}	
20×20	2.65×10^{-4}	3.91	4.91×10^{-4}	3.37	4.20×10^{-3}	3.49	7.84×10^{-3}	2.85
40×40	1.57×10^{-5}	4.08	4.01×10^{-5}	3.61	3.80×10^{-4}	3.47	1.00×10^{-3}	2.97
80×80	9.31×10^{-7}	4.08	2.78×10^{-6}	3.85	3.39×10^{-5}	3.49	9.68×10^{-5}	3.37

The solution is advanced until $t = 1.8$ using linear, quadratic and cubic reconstructions, with the limiters being active everywhere (Fig. 17). In order to focus our analysis on the reconstructions and limiting alone, all the derivatives are computed using $p = 3$ MLS with $N = 7$ points per cloud and $h = 0.55 d_{\max}$.

Even though the higher-order schemes perform fairly better than the second-order one, it is clear that the limiters are introducing excessive dissipation when applied to the higher-order derivatives. Actually, the third-order scheme seems to perform better than the fourth-order one.

Selective limiting is then introduced. Thus, the derivatives on cell I are only limited whenever the high-scale component verifies $|\Psi| > 0.04|u_{\max} - u_{\min}|$, where $u_{\max}$ (resp. $u_{\min}$) is the maximum (resp. minimum) value of the sensed variable (density) within the cloud of cell I. Fig. 18 presents

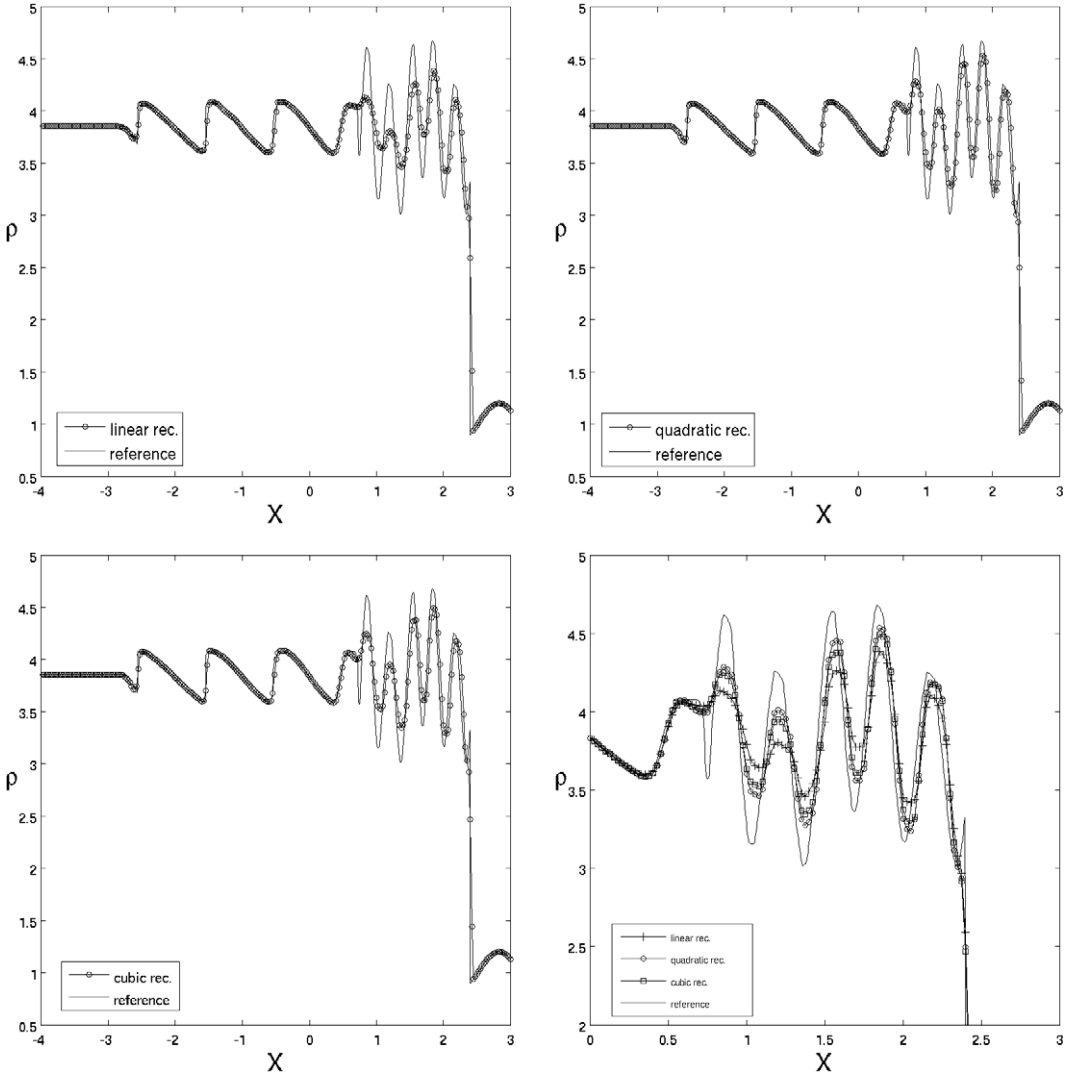

Fig. 17. Shu–Osher problem, 400 cells, limiters active everywhere. Density profiles computed with the second (top left), third (top right) and fourth (bottom left) order schemes. Detailed comparison (bottom right).

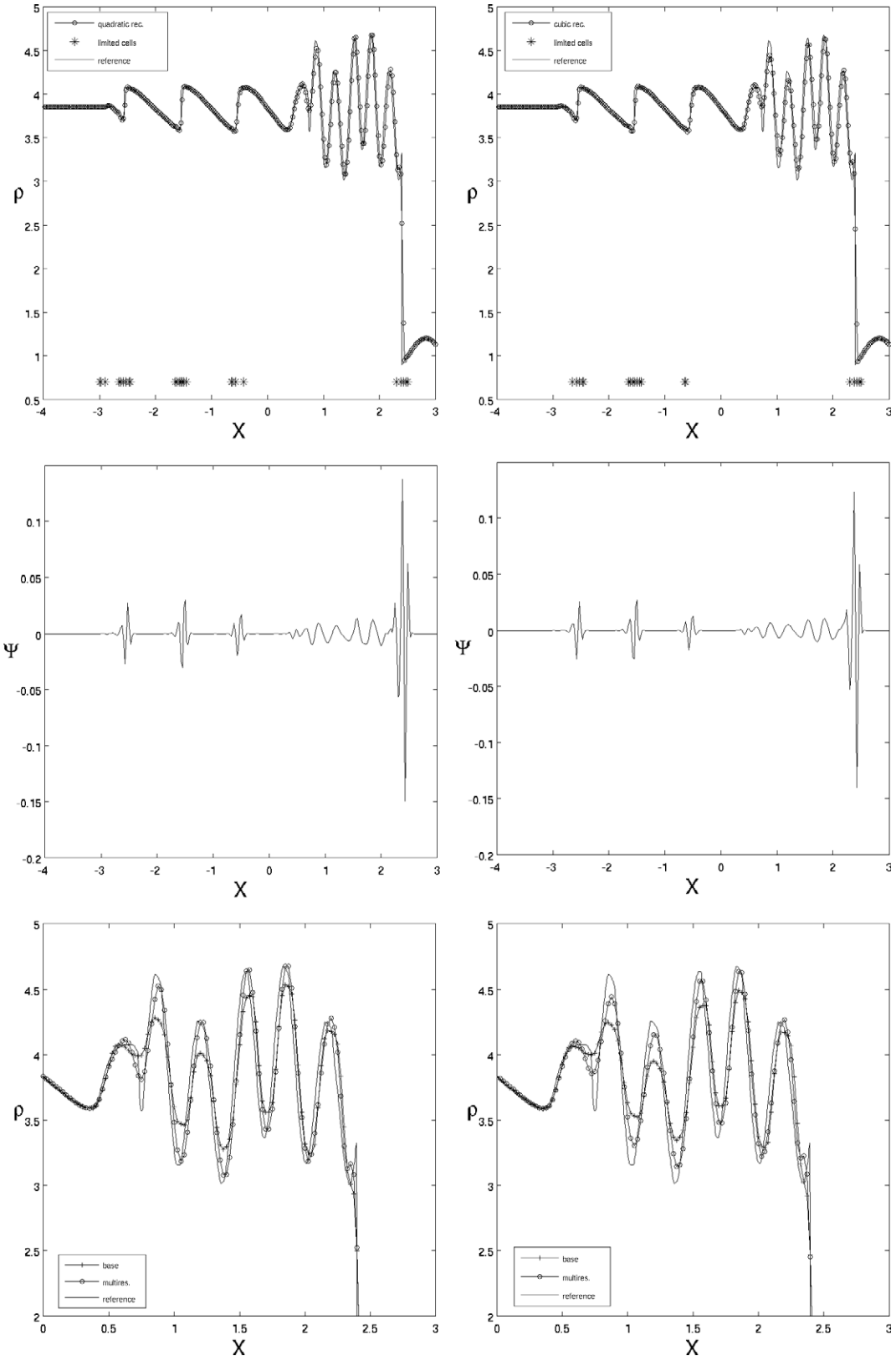

Fig. 18. Shu–Osher problem, 400 cells, multiresolution-based selective limiting. Density profiles computed with the third and fourth-order schemes (top left and top right, respectively), high-scale components of the density (center) and detailed comparison of the densities with and without selective limiting (bottom).

the computed density profiles and location of the limited cells (top), and the high-scale component of the density (center) for the quadratic and cubic reconstructions. The response of the indicator Ψ is interesting by itself. A

comparison of the computed densities with limiters everywhere and with selective limiting is also plotted (bottom). The results are good, comparable only to those of high order ENO/WENO schemes.

6.2. Inviscid flow: airfoils

6.2.1. A subsonic test

The problem set up corresponds to a subsonic flow around a NACA 0012 airfoil. The freestream Mach number is $M = 0.63$ and the angle of attack is $\alpha = 2°$. The computational grid is rather coarse (5322 cells). Given the poor mesh resolution near the leading and trailing edges, the inherent dissipation associated to each reconstruction becomes apparent through the inspection of the Mach number isolines.

Fig. 19 presents a close-up view of the Mach number isolines obtained by using linear (A), quadratic (B) and cubic (C and D) reconstructions. The inviscid fluxes have been integrated using one, two and either two (C) or three (D) Gauss points per edge, for the linear, quadratic and cubic reconstructions, respectively. The solution provided by the linear reconstruction clearly shows an anomalous pseudo-viscous behaviour of the Mach number contours near the surface. The entropy layer is dramatically reduced by the increase of the order of the reconstruction. Note that the grid was not modified near the airfoil for the higher-order schemes, and therefore straight edges are used in the boundary cells. The maximum entropy production reduces from $\Delta S_{\max} = 0.03336$ (linear reconstruction) to $\Delta S_{\max} = 0.00772$ (cubic reconstruction), where S is given by

$$S = \ln\left(\frac{h^{\frac{\gamma}{\gamma-1}}}{p}\right), \quad h = \gamma\left(E - \frac{1}{2}(u^2 + v^2)\right). \tag{68}$$

6.2.2. Two transonic examples

A non-adapted finer grid (12243 cells) has been used to solve two transonic test cases: (I) $M = 0.8$, $\alpha = 1.25°$, and (II) $M = 0.85$, $\alpha = 1°$. Figs. 20 and 21 show the results for test cases I and II, respectively, using quadratic reconstruction and either the BJ or the PC5 limiter: Mach number isolines, pressure isolines and surface pressure coefficient C_p distribution. Both limiters provide sharp shock-capturing (one interior cell) and clear slip lines, although the PC5 limiter appears to be slightly more dissipative.

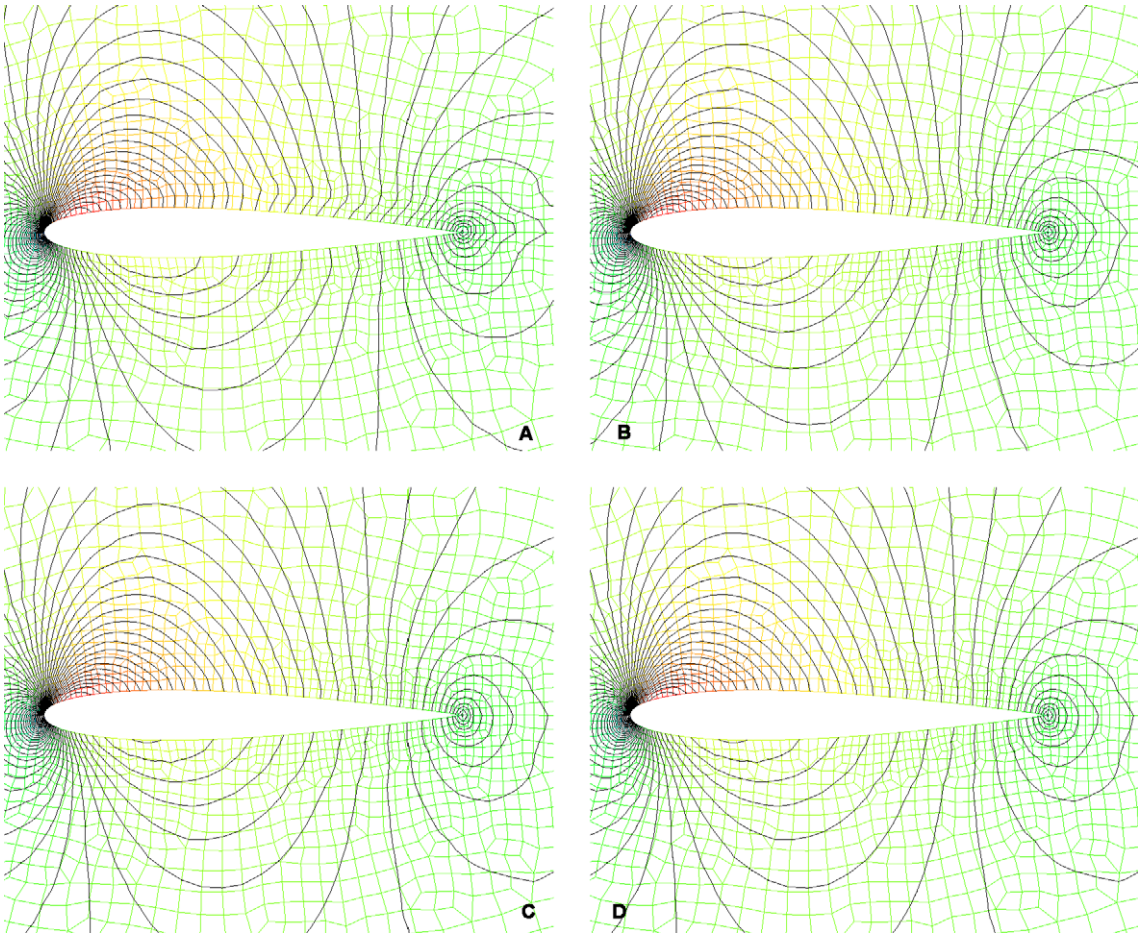

Fig. 19. Subsonic inviscid flow around a NACA 0012 airfoil ($M = 0.63$, $\alpha = 2.0°$): close-up view of the Mach number contours obtained with linear (A), quadratic (B) and cubic (with 2 and 3 Gauss points per edge, C and D) reconstructions.

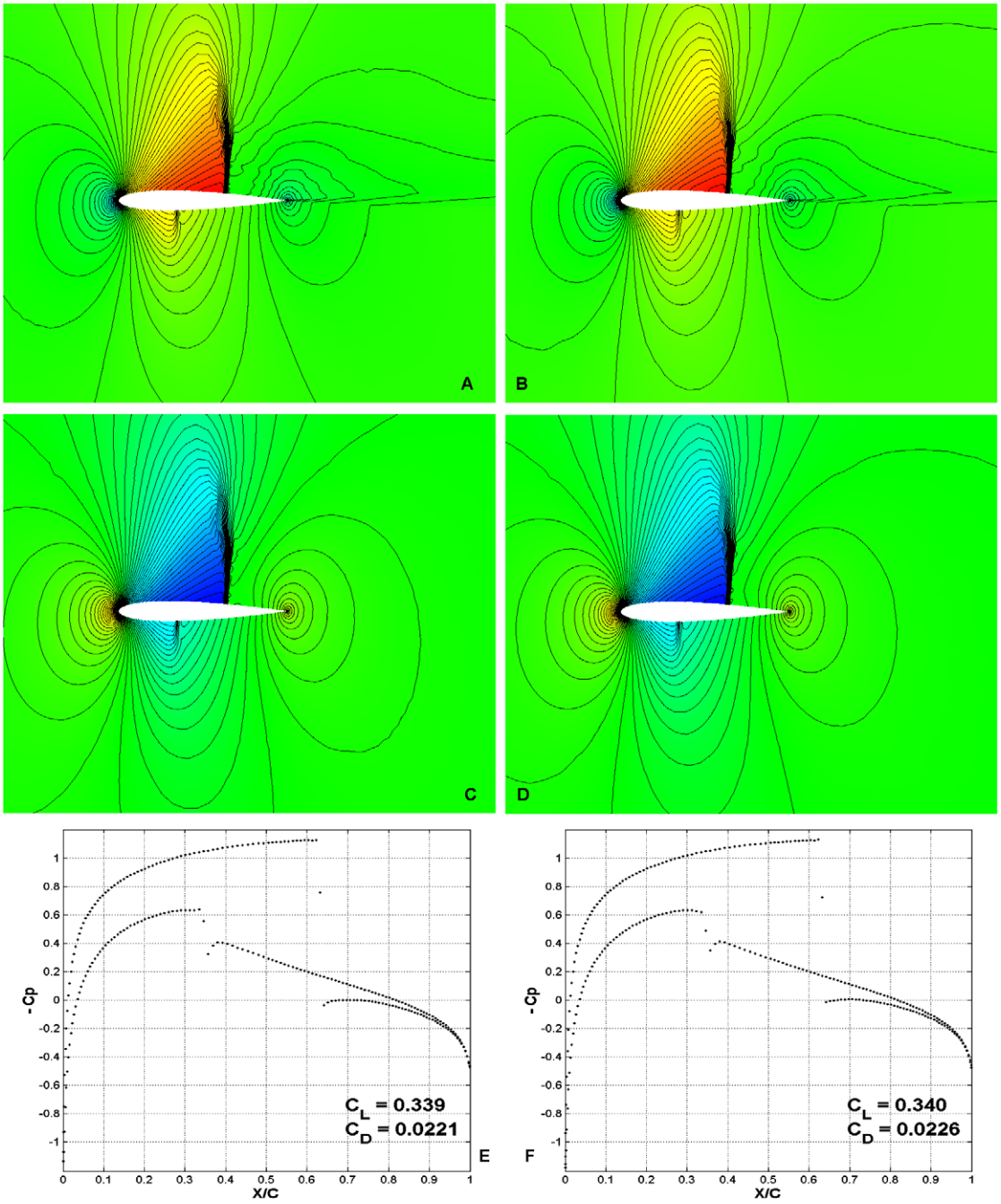

Fig. 20. Inviscid flow around a NACA 0012 airfoil ($M = 0.8$, $\alpha = 1.25°$): results obtained using *quadratic reconstruction* with either the BJ limiter (A–C–E) or the PC5 limiter (B–D–F). Mach number contours (A and B), pressure contours (C and D) and surface pressure coeffcients C_p (E and F).

6.3. Viscous flow

6.3.1. Shock wave impingement on a spatially evolving mixing layer

We reproduce the example presented in [32]. An oblique shock impacts on a spatially developing mixing layer. The flow is fully supersonic at the outflow, so no explicit outflow boundary conditions are required. The problem domain is the rectangle $0 \leqslant x \leqslant 200$ and $-20 \leqslant y \leqslant 20$, with inflow velocities specified as a hyperbolic tangent profile

$$u = 2.5 + 0.5\tanh(2y). \tag{69}$$

Hence, the velocity of the upper stream is $u_1 = 3$, whereas the velocity of the lower stream is $u_2 = 2$. The convective Mach number, defined as $\frac{u_1 - u_2}{c_1 + c_2}$, where c_1 and c_2 are the free stream sound speeds, is equal to 0.6.

The shear layer is excited by adding a periodic fluctuation to the vertical component of the velocity inflow, as

$$v' = \sum_{k=1}^{2} a_k \cos\left(\frac{2\pi k t}{T} + \phi_k\right) e^{\left(\frac{-y^2}{b}\right)}, \tag{70}$$

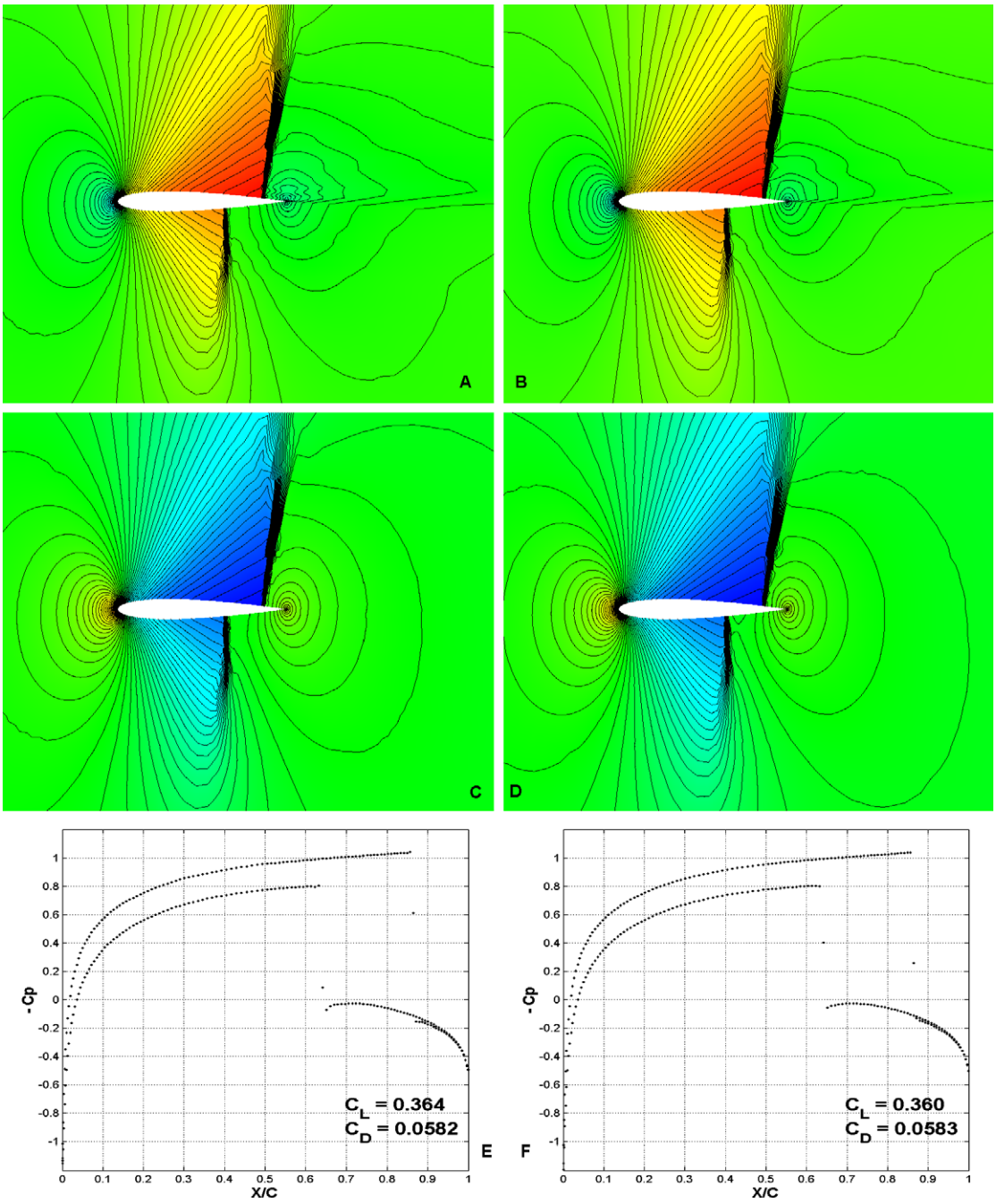

Fig. 21. Inviscid flow around a NACA 0012 airfoil ($M = 0.85$, $\alpha = 1°$): results obtained using *quadratic reconstruction* with either the BJ limiter (A–C–E) or the PC5 limiter (B–D–F). Mach number contours (A and B), pressure contours (C and D) and surface pressure coeffcients C_p (E and F).

where $b = 10$ and $T = \frac{\lambda}{u_c}$, being $u_c = 2.68$ is the convective velocity, defined by $u_c = \frac{u_1 c_2 + u_2 c_1}{c1 + c2}$, and $\lambda = 30$ the wavelength. For $k = 1$ we take $a_1 = 0.05$ and $\phi_1 = 0$. For $k = 2$, $a_2 = 0.05$ and $\phi_2 = \pi/2$.

The reference density is taken as the average of the two free streams and the reference pressure is given by

$$p_R = \frac{(\rho_1 + \rho_2)(u_1 - u_2)^2}{2}. \tag{71}$$

Under the assumption that both streams have equal stagnation enthalpies, the local speed of sound reads

$$c^2 = c_1^2 + \frac{(\gamma - 1)}{2}(u_1^2 - u_2^2). \tag{72}$$

Equal pressure through the mixing layer is assumed. The following values are used at the inflow (left boundary):

$$p_0 = 0.3327, \qquad H_0 = 5.211, \qquad \mu_0 = 5 \times 10^{-4}, \tag{73}$$

whereas on the upper boundary we set

$$u = 2.9709, \qquad v = -0.1367, \qquad \rho = 2.1101, \qquad p = 0.4754. \tag{74}$$

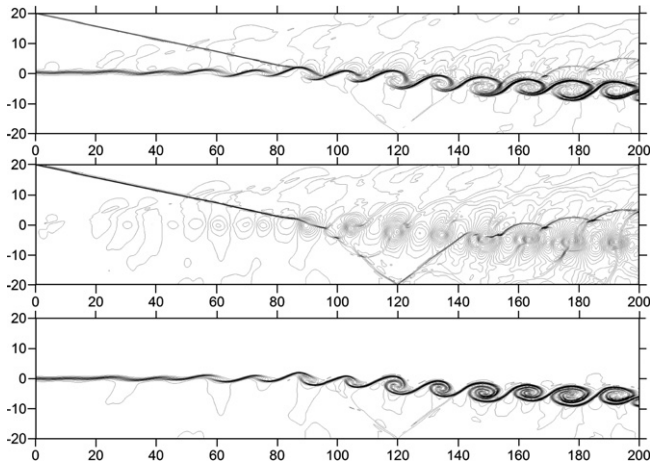

Fig. 22. Shock wave impingement on a mixing layer at $t = 120$. Fourth-order results on the 600×300 grid. Contours of density (top), pressure (center) and temperature (bottom).

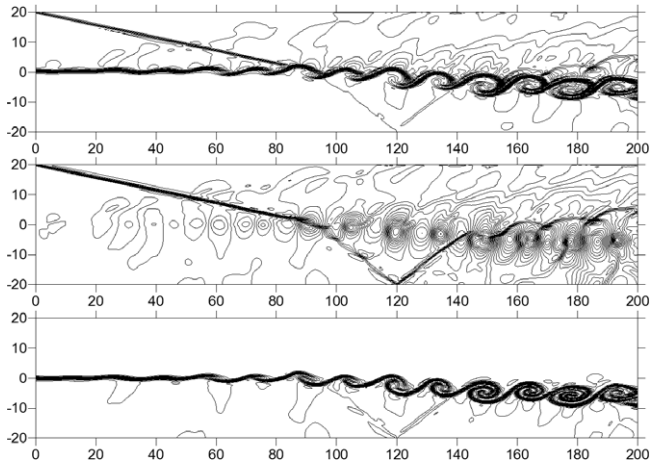

Fig. 23. Shock wave impingement on a mixing layer at $t = 120$. Fourth-order results on the 400×100 grid. Contours of density (top), pressure (center) and temperature (bottom).

On the lower boundary, a slip wall condition was specified. With this problem setup, an oblique shock originates from the top left corner, impacting the shear layer around $x = 90$. The shock wave reflects at the lower wall and passes back through the deflected shear layer.

The problem was run using the fourth-order scheme on two grids of 400×100 and 600×300 cells. Figs. 22 and 23 show the contours of density (top), pressure (center) and temperature (bottom) on the fine and coarse grids, respectively. On both grids the fourth-order scheme is capable of capturing the fine scale features of the flow, such as the formation of shocklets or the splitting in two of the vortex core located at $x = 148$, caused by its interaction with the reflected shock wave.

7. Conclusions

This paper explored the approximation power of MLS approximations in the context of higher-order finite volume schemes on unstructured grids. The scope of the application of MLS is threefold: (1) computation of high order derivatives of the field variables for a Godunov-type approach to hyperbolic problems or terms of hyperbolic character, (2) direct reconstruction of the fluxes at cell edges, for elliptic problems or terms of elliptic character, and (3) multiresolution shock detection and selective limiting.

A major advantage of the proposed methodology over the most popular existing higher-order methods is related to the viscous discretization. The use of MLS approximations allows the direct reconstruction of high order viscous fluxes using quite compact stencils, and without introducing new degrees of freedom, which results in a significant reduction in storage and workload.

A selective limiting procedure is proposed, based on the multiresolution properties of the MLS approximants, which allows to switch off the limiters on smooth regions of the flow.

Accuracy tests show that the proposed method achieves the expected convergence rates. Representative simulations show that the methodology is applicable to problems of engineering interest.

Acknowledgements

This work has been partially supported by the ''Ministerio de Educación y Ciencia'' of the Spanish Government (grants #DPI2004-05156 and #DPI2006-15275) cofinanced with FEDER funds, and by the ''Xunta de Galicia'' (grants #PGIDIT05PXIC118002PN and #PGDIT06-TAM11801PR).

The research of Dr. Luis Cueto-Felgueroso is supported by the ''Ministerio de Educación y Ciencia'' through its program of postdoctoral scholarships. Furthermore, it is also gratefully acknowledged the financial support received in the past from ''Colegio de Ingenieros de Caminos, Canales y Puertos'', ''Fundación de la Ingeniería Civil de Galicia'' and ''Caixanova''. Mr. Xesús Nogueira gratefully acknowledges the financial support received from ''Fundación Caixa Galicia''.

References

[1] B. Cockburn, C.W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Scient. Comput. 6 (2001) 173–261.

[2] Z.J. Wang, L. Zhang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional Euler equations, J. Comput. Phys. 194 (2004) 716–741.

[3] P. Lancaster, K. Salkauskas, Surfaces generated by moving least squares methods, Math. Comput. 155 (1981) 141–158.

[4] W.K. Liu, S. Li, T. Belytschko, Moving least-squares reproducing kernel methods: (I) methodology and convergence, Comput. Methods Appl. Mech. Engrg. 143 (1997) 113.

[5] S. Li, W.K. Liu, Moving least-squares reproducing kernel methods: (II) Fourier analysis, Comput. Methods Appl. Mech. Engrg. 139 (1996) 159–193.

[6] S. Li, W.K. Liu, Reproducing kernel hierarchical partition of unity, part I – formulation and theory, Int. J. Numer. Methods Engrg. 45 (1999) 251–288.
[7] A. Gossler, Moving Least-Squares: a numerical differentiation method for irregularly spaced calculation points, SANDIA Report, SAND2001-1669, 2001.
[8] D. Levin, The approximation power of moving least-squares, Math. Comput. 67 (224) (1998) 1517–1531.
[9] L. Cueto-Felgueroso, I. Colominas, J. Fe, F. Navarrina, M. Casteleiro, High order finite volume schemes on unstructured grids using Moving Least-Squares reconstruction. Application to shallow water dynamics, Int. J. Numer. Methods Engrg. 65 (2006) 295–331.
[10] S.K. Godunov, A difference method for the numerical calculation of discontinuous solutions of hydrodynamic equations, Mat. Sbornik. 47 (3) (1959) 271–306.
[11] A. Harten, P. Lax, B. Van Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM Rev. 25 (1983) 35–61.
[12] T.J. Barth, D.C. Jespersen, The design and application of upwind schemes on unstructured meshes, AIAA-89-0366, 1989.
[13] N.T. Frink, Upwind scheme for solving the Euler equations on unstructured tetrahedral meshes, AIAA J. 30 (1) (1992) 70.
[14] V. Venkatakrishnan, Convergence to steady state solutions of the Euler equations on unstructured grids with limiters, J. Comput. Phys. 118 (1995) 120–130.
[15] P. Jawahar, H. Kemath, A high-resolution procedure for Euler and Navier–Stokes computations on unstructured grids, J. Comput. Phys. 164 (2000) 165–203.
[16] T.J. Barth, Aspects of unstructured grids and finite-volume solvers for the Euler and Navier–Stokes equations, VKI Lecture Series 1994-05, 1995.
[17] J.M. Vaassen, P. Wautelet, J.A. Essers, A quadratic reconstruction scheme for hypersonic reacting flows on unstructured meshes, in: ECCOMAS Computational Fluid Dynamics Conference, Swansea, Wales.
[18] I. Aavatsmark, An introduction to multipoint flux approximations for quadrilateral grids, Comput. Geosci. 6 (2002) 405–432.
[19] W.K. Liu, W. Hao, Y. Chen, S. Jun, J. Gosz, Multiresolution reproducing kernel particle methods, Comput. Mech. 20 (1997) 295–309.
[20] B. Sjögreen, H.C. Yee, Multiresolution wavelet based adaptive numerical dissipation control for high order methods, J. Scient. Comput. 20 (2004) 211–255.
[21] F.M. White, Viscous Fluid Flow, second ed., McGraw-Hill, New York, 1991.
[22] L. Cueto-Felgueroso, I. Colominas, High-order finite volume methods and multiresolution reproducing kernels, Arch. Comput. Methods Engrg., submitted for publication.
[23] L. Cueto-Felgueroso, Particles, finite volumes and unstructured grids: numerical simulation of fluid dynamics problems, Ph.D. thesis, Universidad de A Coruña, 2005. <http://www.tesisenred.net/> (in Spanish).
[24] L. Cueto-Felgueroso, I. Colominas, G. Mosqueira, F. Navarrina, M. Casteleiro, On the Galerkin formulation of the SPH method, Int. J. Numer. Methods Engrg. 60 (2004) 1475–1512.
[25] B. van Leer, Towards the ultimate conservative difference scheme V. A second order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101.
[26] J. Van Rosendale, Floating shock fitting via Lagrangian adaptive meshes, ICASE 94-89 (1989).
[27] G.D. Van Albada, B. Van Leer, W.W. Roberts, A comparative study of computational methods in cosmic gas dynamics, Astron. Astrophys. 108 (1982) 76.
[28] P.L. Roe, Approximate Riemann solvers, parameter vectors and difference schemes, J. Comput. Phys. 43 (1981) 357–372.
[29] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[30] A.H. Shapiro, The Dynamics and Thermodynamics of Compressible Fluid Flow, Ronald Press Co., New York, 1953.
[31] C.W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, II, J. Comput. Phys. 83 (1989) 32–78.
[32] H.C. Yee, N.D. Sandham, M.J. Djomehri, Low-dissipative high-order shock-capturing methods using characteristic-based filters, J. Comput. Phys. 150 (1999) 99–238.