Arch Comput Methods Eng
DOI 10.1007/s11831-008-9017-y

High-order Finite Volume Methods and Multiresolution Reproducing Kernels

Luis Cueto-Felgueroso · Ignasi Colominas

Received: 2 May 2007 / Accepted: 2 May 2007

Abstract This paper presents a review of some of the most successful higher-order numerical schemes for the compressible Navier-Stokes equations on unstructured grids. A suitable candidate scheme would need to be able to handle potentially discontinuous flows, arising from the predominantly hyperbolic character of the equations, and at the same time be well suited for elliptic problems, in order to deal with the viscous terms. Within this context, we explore the performance of Moving Least-Squares (MLS) approximations in the construction 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. The proposed finite volume method is formulated within a continuous spatial representation framework, provided by the MLS approximants, which is "broken" locally (inside each cell) into piecewise polynomial expansions, in order to make use of the specialized finite volume technology for hyperbolic problems. This approach is in contrast with the usual practice in the finite volume literature, which proceeds bottom-up, starting from a piecewise constant spatial representation. 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, and very competitive when compared to other existing procedures.

1 Introduction

High order is fashionable. Whether very high order methods (higher than, say, fourth order) are really needed for flow simulations or not is a different matter. For turbulence simulations, for example, and in particular for Direct Numerical Simulation (DNS), Large Eddy Simulation (LES) or Detached Eddy Simulation (DES), what we really need is methods with high resolution. This expression is not used here in the sense popularized by Harten, second order accuracy for smooth flows and good shock capturing, but rather in the sense of good spectral resolution: high accuracy in Fourier space. We believe this distinction is important because, even though higher order usually means higher resolution (lower dispersion and dissipation), just increasing the order of the truncation error is a quite inefficient way of improving the resolution of a scheme, as those with some experience in finite differences or spectral methods will certainly know. In the context of unstructured grids, however, it is not easy to separate high-order and improved resolution, because most schemes are based on polynomial approximation. In the following, by high-order we mean order higher than second.

In the last decade, there has been a growing interest in the construction of robust and efficient high order schemes for the Navier-Stokes equations on unstructured grids, parallel to the increasing feeling that highly accurate discretization schemes, suitable for complex geometries, are a conditio

L. Cueto-Felgueroso (✉)
Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA
e-mail: lcueto@mit.edu

I. Colominas
School of Civil Engineering, Universidad de A Coruña, Campus de Elviña, 15071, A Coruña, Spain
e-mail: icolominas@udc.es

sine qua non for the development of practical high-fidelity computational approaches, particularly for turbulence simulation and aeroacoustics.

In this paper we present a review of some of the most successful existing methods (or, rather, families of methods), namely: high-order finite volume methods [20, 22, 23, 51, 54, 72, 74, 75, 93, 96], essentially non-oscillatory (ENO and WENO) methods [48, 53, 55–57, 60, 71, 84], discontinuous Galerkin methods [24, 25, 31–35], the so-called spectral (finite) volume method [98–102], and residual distribution or fluctuation splitting methods [4, 6–13, 41, 76, 77, 79, 80]. There is clear difference in terms of spatial approximation between the above schemes. High order finite volume methods construct high order reconstructions of the field variables extracting information from neighbour cells. Essentially non-oscillatory methods use a similar approach, but they use adaptive stencils in order to get non-oscillatory solutions. On the other hand, both discontinuous Galerkin methods and the so-called spectral volume method use piecewise polynomial approximations obtained by means of additional degrees of freedom inside each *element* or cell. The former uses the high-order finite element technology, whereas the latter is based on cell subdivision. Finally, residual distribution schemes are also based on the traditional spatial approximation of continuous finite element methods. Each of these approaches has its own advantages and disadvantages. The "extended" approach of finite volume methods is, in principle, less systematic than the finite-element based approach of discontinuous Galerkin, for instance. This makes the DG approximation more clear, compact and general. On the other hand, high-order reconstructions based on stencils are spatially centered, in contrast with the spatially biased approximation of piecewise polynomial reconstructions, like those of DG. The quality and nature of the reconstruction procedure is critical, as it directly affects not only the accuracy and robustness of the scheme, but also its efficiency, through the pattern of coupling between degrees of freedom.

All the methods considered here are rather "specialized" numerical schemes, in the sense that they are best suited for hyperbolic conservation laws, and the extension to elliptic problems requires, in principle, special attention. We present a review of some of the most successful strategies to extend the above techniques to solve elliptic equations, and discuss their relative advantages and shortcomings.

Within this context, this paper also presents a review on the recent developments on high-order finite volume schemes based on Moving Least-Squares approximation [36, 38, 40]. The crucial difference with respect to other state of the art finite volume methods is the *spatial representation* provided by the MLS approximants. Rather than starting from a piecewise constant field, which is "improved" or reconstructed, we start with a *continuous* high order representation of the solution, which is, for convenience, *"broken"* locally (inside each cell) into piecewise polynomial expansions, in order to make use of the powerful finite volume technology for hyperbolic problems. We have, thus, a dual approximation framework: an underlying continuous high-order reconstruction of the variables, provided by the MLS approximation, and its piecewise polynomial approximation, a "broken" reconstruction inside each cell. The former provides the basic formulation, consistent mass matrix, and viscous fluxes, whereas the latter establishes a direct connection to more traditional Godunov-type high-order finite volume schemes. One of the major advantages of the proposed methodology is, thus, that the solution of elliptic equations fits naturally within its scope.

Originally devised for data processing and surface generation [65], Moving Least-Squares (MLS) became very popular within the *meshfree* community, being widely used both in Eulerian and Lagrangian formulations 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 [52, 66–68, 70]. 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 [38, 40], 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.

The strategy adopted in this study for convection terms follows the ideas of the generalized Godunov method [23, 51, 54], 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 [23, 49, 59, 97]. 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.

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 as a general continuous approximation framework 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.

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 [69], 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 [87].

The outline of the paper is as follows. Section 2 presents the model equations, which will serve as a reference point of the kind of problems we are interested in solving. Section 3 contains a review of some of the most successful state of the art high order schemes for convection-dominated problems on unstructured grids. Section 4 introduces the proposed finite volume formulation, and Sect. 5 is a brief introduction to Moving Least Squares Reproducing Kernel approximation methods, which is completed with some practical implementation issues, presented in Sect. 6. Accuracy tests and representative simulations are exposed in Sects. 7 and 8, respectively, and, finally, our main conclusions are drawn in Sect. 9.

2 Model Problem: The Navier-Stokes Equations

We are interested in high-order methods that are well suited for the solution of mixed hyperbolic/parabolic problems on unstructured meshes. This section presents the compressible Navier-Stokes equations, which are to be used as the prototype model problem. These governing equations can be written as a system of conservation laws, with terms of distinct mathematical structure: terms of hyperbolic character (the "Euler" part of the equations), and terms of elliptic character (the "viscous" part of the equations). The diverse mathematical properties of these terms will be reflected in the discretization methods presented in the following sections.

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

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

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 $S_0 = 110.4$ K is an experimental constant [105].

3 Review of High-order Schemes for the Compressible Navier-Stokes Equations on Unstructured Meshes

This section presents an introduction to some of the most successful state of the art higher (than second) order discretization methods for the compressible Navier-Stokes equations on unstructured grids. The predominantly hyperbolic character of the Navier-Stokes equations has a significant impact on the structure of the schemes described in the following paragraphs. In some sense, all of them are "*specialized*" methods, particularly well suited for hyperbolic problems. On the other hand, this specialization has a negative impact on their suitability for the discretization of elliptic equations or terms of elliptic character. Some of the methods are based on inherently discontinuous spatial representations of the variables, others are based on multidimensional upwinding principles, but none of them are naturally suited for the discretization of higher order terms. We start by describing the structure of these methods for "advection" problems, and then present a review on how "diffusive" terms are addressed.

3.1 High-order Schemes for Hyperbolic Conservation Laws

3.1.1 Introduction: The Generalized Godunov Scheme

The basic first-order finite volume discretization stems from the integral form of the conservation law

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla\cdot\boldsymbol{\mathcal{F}} = \boldsymbol{S} \quad \text{in } \Omega \tag{10}$$

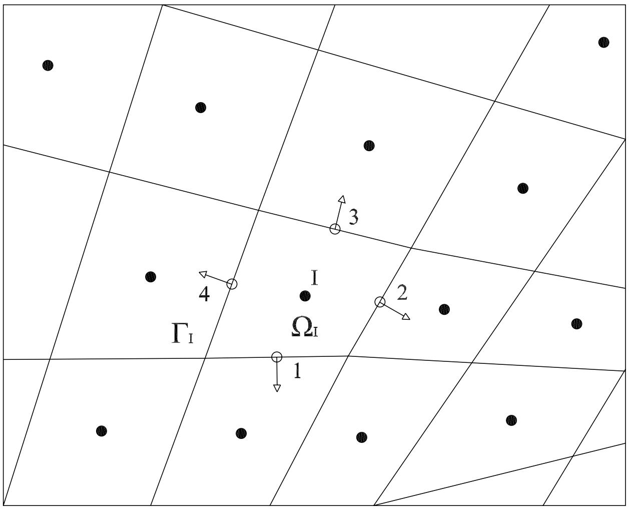


Fig. 1 Cell-centered finite volume discretization

over a control volume Ω_I (Fig. 1)

$$\int_{\Omega_I}\frac{\partial \boldsymbol{u}}{\partial t}\,d\Omega + \int_{\Omega_I}\nabla\cdot\boldsymbol{\mathcal{F}}\,d\Omega = \int_{\Omega_I}\boldsymbol{S}\,d\Omega. \tag{11}$$

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

$$\int_{\Omega_I}\frac{\partial \boldsymbol{u}}{\partial t}d\Omega + \int_{\Gamma_I}\boldsymbol{\mathcal{F}}\cdot\boldsymbol{n}\,d\Gamma = \int_{\Omega_I}\boldsymbol{S}\,d\Omega, \tag{12}$$

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

$$\boldsymbol{\mathcal{F}} = \left(\boldsymbol{F}_x, \boldsymbol{F}_y\right) \tag{13}$$

was 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 order to fix ideas, consider a *cell-centered* approach with quadrilateral control volumes (Fig. 1). The extension to triangular grids is inmediate. From a spatial point of view, the algorithm involves studying the evolution of cell-averaged values of the field variables, and the solution is assumed to be constant within each control volume. Therefore, the underlying spatial representation would be that of a piecewise constant flow field. High order schemes are constructed by substituting the constant representation by a piecewise continuous (usually polynomial) reconstruction of the flow variables inside each cell. In addition, special care must be paid to the discretization of the viscous fluxes, which are functions of the conserved variables, but also of their gradients.

The methodology will be presented within the framework of the numerical method of lines. Focusing on a control volume I, and assuming that suitable approximations to the fluxes (*numerical fluxes*) are available at a set of quadrature points at each edge, the semi-discrete version of (12) reads

$$A_I \frac{d\boldsymbol{U}_I}{dt} + \sum_{iedge=1}^{nedge_I} \sum_{igau=1}^{ngau_I} \boldsymbol{\mathcal{F}} \cdot \boldsymbol{n}\big|_{igau} \mathcal{W}_{igau} = S_I, \tag{14}$$

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, $\mathcal{W}_{igau}$ denotes a quadrature weight and $\boldsymbol{U}_I$ represents the average value of $\boldsymbol{U}$ over the cell I. 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 latter 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 unique 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 [23, 51, 54, 93], 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 cell-averaged information from neighbour 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.
- 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.

The above steps are only meaningful, of course, in the context of explicit time integration schemes, as they reflect the process of residual evaluation. If implicit strategies are preferred, one should add to the above steps the assembly of the Jacobian matrix and solution of the (nonlinear, in general) systems of equations.

As we will see below, 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, if higher order discretizations of elliptic equations or viscous terms follow this path, the stability of the resulting numerical scheme can be seriously compromised.

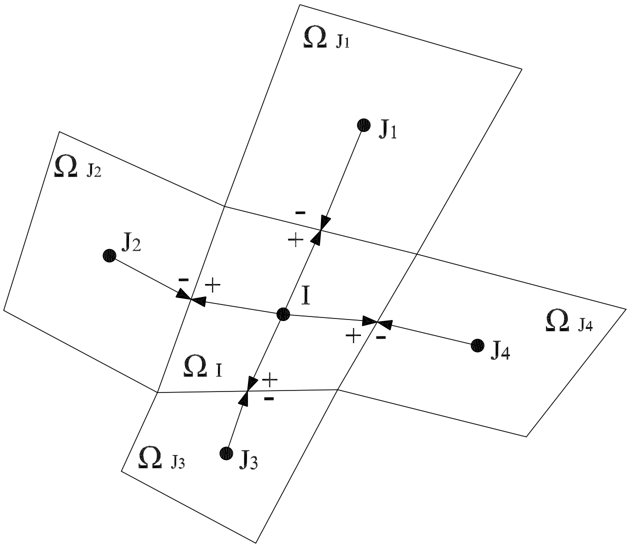


Fig. 2 Extrapolated variables used to evaluate the inviscid fluxes across the interfaces of control volumes Ω_I and $\{\Omega_{J_k}, k = 1, 4\}$

3.1.2 High-order Finite Volume Methods: k-exact Reconstruction and the Cell-average Approach

Assume that, in principle, the piecewise constant cell-averaged solution representation exposed in the previous section is adopted. In order to evaluate the flux $\boldsymbol{\mathcal{F}} \cdot \boldsymbol{n}$, from the states $\boldsymbol{u}^+$ and $\boldsymbol{u}^-$ at each side of the interface, a suitable *numerical flux* $\boldsymbol{\mathcal{H}}(\boldsymbol{u}^+, \boldsymbol{u}^-)$ has to be defined. A practical discretization of (14) then reads

$$A_I \frac{d\boldsymbol{U}_I}{dt} + \sum_{iedge=1}^{nedge_I} \sum_{igau=1}^{ngau_I} \boldsymbol{\mathcal{H}}(\boldsymbol{u}^+, \boldsymbol{u}^-, \boldsymbol{n})|_{igau} \mathcal{W}_{igau} = S_I. \tag{15}$$

As mentioned above, high-order schemes require high-order accurate reconstructed states $\boldsymbol{u}^+$ and $\boldsymbol{u}^-$. Second order schemes (linear reconstruction) have been well studied and the complexity of the reconstruction reduces to the estimation of the gradient of the solution, which is usually accomplished by means of Green-Gauss or simple least-squares procedures [21, 23, 49, 59, 97]. Higher order schemes require more sophisticated strategies, and the number of existing reconstruction procedures is small.

A popular existing approach follows the lines introduced in [20, 22], and further developed for practical applications in [72, 74, 75, 96], about k-exact reconstructions. The idea behind this reconstruction is to compute a polynomial expansion at each cell I, $R_I(\boldsymbol{x} - \boldsymbol{x}_I)$, that preserves the mean value in the control volume I, and that reconstructs exactly polynomials of degree up to k. The coefficients defining this polynomial are obtained by minimizing, in a least-squares sense, the error in the estimate of the cell-averaged values at some neighbour cells (the stencil of the reconstruction).

Note that, although we are in a cell-average contex, the reconstructing polynomial reconstructs point-wise values. It would probably be meaningless to reconstruct cell-averages, since the function "mean value of the solution at each cell" can hardly be thought of as a smooth function of the spatial coordinates.

The expansion $R_I(\boldsymbol{x}-\boldsymbol{x}_I)$ can be written as

$$\begin{aligned} R_I(\boldsymbol{x}-\boldsymbol{x}_I) &= u_I^R(\boldsymbol{x}-\boldsymbol{x}_I) \\ &= u_I + \nabla u_I \cdot (\boldsymbol{x}-\boldsymbol{x}_I) \\ &\quad + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_I)^T \boldsymbol{H}\,(\boldsymbol{x}-\boldsymbol{x}_I) + \cdots, \end{aligned} \tag{16}$$

where u_I^R is the reconstructed value, and ∇u_I and $\boldsymbol{H}$ are, respectively, the gradient and Hessian matrix at the reference point (usually the centroid) of the control volume, $\boldsymbol{x}_I = (x_I, y_I)$. Note that the point value u_I does not coincide, at least in general, with the cell-averaged value. This remark is important in terms of accuracy, as indicated in [74]. The coefficients of this expansion (the approximate derivatives and point value at (x_I, y_I)) are chosen to enforce conservation of the mean on cell I, and to minimize, in some suitable sense, the error in the reconstruction of smooth solutions.

Conservation of the mean inside control volume I implies

$$\frac{1}{A_I}\int_\Omega u_I^R\, d\Omega = u_I \tag{17}$$

which, for the expansion (16), reads

$$\begin{aligned} &\frac{1}{A_I}\int_\Omega u_I^R\, d\Omega \\ &\quad = u_I + \frac{\partial u}{\partial x}\bigg|_I \widetilde{x}_I + \frac{\partial u}{\partial y}\bigg|_I \widetilde{y}_I + \frac{\partial^2 u}{\partial x^2}\bigg|_I \frac{\widetilde{x}_I^2}{2} + \frac{\partial^2 u}{\partial x \partial y}\bigg|_I \widetilde{xy}_I \\ &\qquad + \frac{\partial^2 u}{\partial y^2}\bigg|_I \frac{\widetilde{y}_I^2}{2} + \cdots, \end{aligned} \tag{18}$$

where

$$\widetilde{x^n y^m}_I = \frac{1}{A_I}\int_\Omega (x-x_I)^n (y-y_I)^m\, d\Omega. \tag{19}$$

The coefficients $\{u_I, \frac{\partial u}{\partial x}|_I, \frac{\partial u}{\partial y}|_I, \ldots\}$ are now chosen to minimize the error in the predicted mean values of the function for the control volumes within a certain *stencil* associated to control volume I. Thus, the mean value of the reconstruction corresponding to a control volume J of the stencil can be written as [72, 74, 75]

$$\begin{aligned} &\frac{1}{A_J}\int_{\Omega_J} u_I^R(\boldsymbol{x}-\boldsymbol{x}_I)\, d\Omega \\ &\quad = u_I + \frac{\partial u}{\partial x}\bigg|_I \widehat{x}_{IJ} + \frac{\partial u}{\partial y}\bigg|_I \widehat{y}_{IJ} + \frac{\partial^2 u}{\partial x^2}\bigg|_I \frac{\widehat{x}_{IJ}^2}{2} + \frac{\partial^2 u}{\partial x \partial y}\bigg|_I \\ &\qquad \times \widehat{xy}_{IJ} + \frac{\partial^2 u}{\partial y^2}\bigg|_I \frac{\widehat{y}_{IJ}^2}{2} + \cdots, \end{aligned} \tag{20}$$

where

$$\begin{aligned} \widehat{x^n y^m}_{IJ} = \frac{1}{A_J}\int_{\Omega_J} &((x-x_J)+(x_J-x_I))^n\,((y-y_J) \\ &+ (y_J-y_I))^m\, d\Omega. \end{aligned} \tag{21}$$

As mentioned above, the coefficients are chosen to minimize, in a least squares sense, the difference between the averages of the reconstructing polynomial, (20), and the actual averages u_J. Geometric weights are included to measure the relative importance of the error incurred at each control volume. These weights ω_{IJ} are functions of the distance between the control volume reference points. The resulting least-squares problem reads

$$\begin{aligned} &\begin{pmatrix} \omega_{I1} & \omega_{I1}\widehat{x}_{I1} & \omega_{I1}\widehat{y}_{I1} & \omega_{I1}\widehat{x^2}_{I1} & \omega_{I1}\widehat{xy}_{I1} & \omega_{I1}\widehat{y^2}_{I1} \\ \omega_{I2} & \omega_{I2}\widehat{x}_{I2} & \omega_{I2}\widehat{y}_{I2} & \omega_{I2}\widehat{x^2}_{I2} & \omega_{I2}\widehat{xy}_{I2} & \omega_{I2}\widehat{y^2}_{I2} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \omega_{IN} & \omega_{IN}\widehat{x}_{IN} & \omega_{IN}\widehat{y}_{IN} & \omega_{IN}\widehat{x^2}_{IN} & \omega_{IN}\widehat{xy}_{IN} & \omega_{IN}\widehat{y^2}_{IN} \end{pmatrix} \begin{pmatrix} u_I \\ \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \\ \frac{1}{2}\frac{\partial^2 u}{\partial x^2} \\ \frac{\partial^2 u}{\partial x \partial y} \\ \frac{1}{2}\frac{\partial^2 u}{\partial^2 y} \end{pmatrix} \\ &\quad = \begin{pmatrix} \omega_{I1}u_1 \\ \omega_{I2}u_2 \\ \vdots \\ \omega_{IN}u_N \end{pmatrix} \end{aligned} \tag{22}$$

with the constraint (18). Note that, in principle, the explicit introduction of this constraint requires the solution of the constrained least-squares problem each time the reconstruction is performed (in actual computations, each time the residual is evaluated). The weights are of the form

$$\omega_{IJ} = \frac{1}{|\boldsymbol{x}_J - \boldsymbol{x}_I|^p}, \tag{23}$$

where, typically, $p = 2$. Additional constraints are imposed at the boundaries in order to honor the boundary conditions [72, 74, 75]. This approach provides a reconstruction with exact satisfaction of the boundary conditions. On the other hand, the introduction of constraints in the least-squares problem implies, as mentioned above, the necessity of solving the least-squares problem for each control volume at every residual evaluation.

The above high-order reconstruction procedure involves a piecewise polynomial (discontinuous) representation of the solution. This provides a natural way to implement Riemann solvers for advection-like fluxes, but the discretization of viscous-like fluxes is compromised. At every quadrature point along the control volume boundary, the reconstruction solution and its successive derivatives are double-valued, corresponding to the two discontinuous reconstructions on each side of the interface. The usual choice is to compute the diffusive fluxes using the average values of the solution and its gradients [74], although the stability of the scheme can be compromised by this choice.

One of the main challenges that this approach has to face is related to the conditioning of the least-squares problem, which, particularly in the case of stretched grids (very common, on the other hand, on wall bounded flow problems) may preclude the use of this reconstruction technique.

3.1.3 High-order Essentially Non-oscillatory (ENO and WENO) Methods

The above reconstruction procedure uses fixed stencils, which means that, with independence of the shape of the solution, the reconstruction at a given cell is always performed using the same set of neighbours (*stencil*) and weights. This implies that, in the presence of discontinuities, some TVD-like limiting procedure has to be applied to the high order reconstruction.

The basic idea behind ENO schemes [53, 55, 56, 84] is to use *adaptive stencils* or some *adaptive weighting* of various suitable polynomial reconstructions for each cell (WENO schemes [48, 57, 60, 71]). In the standard ENO approach, a set of stencils comprising different neighbour cells is defined for each cell in the finite volume mesh. For each stencil, a polynomial recovery is constructed, which reconstructs the cell-averages or point values over the cells in the stencil. In the latter case, the polynomials are built is such a way that the reconstruction is conservative. Amongst all the polynomials associated with the various stencils, the less oscillatory (the *smoothest* is some suitable sense) is selected, which requires a certain *smoothness indicator*. If high-order polynomial reconstructions are employed, the resulting finite volume scheme is also high-order accurate. In a later development of ENO schemes, the so-called Weighted ENO (WENO) schemes, the whole set of stencils and their associated reconstructions are used to form a weighted sum of reconstruction polynomials.

The objective in developing this type of schemes is to avoid the introduction of excessive numerical dissipation near discontinuities and smooth extrema, while retaining high-order accuracy and non-oscillatory solutions in those regions. Although ENO and WENO schemes are quite well developed for structured grids, the procedure for unstructured grids is more complicated, and issues related to high cost and insufficient robustness pose important challenges to the practical application of ENO schemes on general meshes. Nonetheless, a considerable amount of progress has been done [2, 3, 5, 14, 43, 53, 58, 73, 88, 89], and this is an interesting and active area of research.

Of particular interest to the methodology proposed in this study is the fact that the development of ENO schemes for unstructured grids has attracted the attention of many researchers towards non-polynomial reconstruction techniques [5, 14, 43, 58, 88], due to the fact that the multidimensional polynomial interpolation problems required for recovery from scattered data are in general badly conditioned.

3.1.4 Discontinuous Galerkin Methods

Discontinuous Galerkin methods have received enormous attention in recent years, and many researchers believe that they will become the preferred discretization method for convection dominated flows in the following years (see [24, 25, 31–35] for an introduction and early developments). They have attracted both finite volume practitioners, who see in DG a natural generalization of finite volume schemes, and finite element analysts, who see in DG a natural bridge between the general Galerkin finite element framework and the powerful Godunov-type finite volume technology, which has been successful in the simulation of flows with strong discontinuities.

The start point is again the conservation law

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{\mathcal{F}} = \boldsymbol{S} \quad \text{in } \Omega \tag{24}$$

with suitable initial and boundary conditions on $\partial\Omega$. Let us introduce the "*broken*" subspace $V(\mathcal{T}^h)$, associated to a suitable decomposition of Ω into a set of non-overlapping *elements*, $\mathcal{T}^h = \{K\}$. In particular, $V(\mathcal{T}^h)$ denotes the space of functions whose restriction to each element K belongs to the Sobolev space $H^1(K)$

$$V = \left\{ v \in L^2(\Omega) \mid v|_K \in H^1(K) \ \forall K \in \mathcal{T}_h \right\}. \tag{25}$$

In order to perform a Galerkin discretization of (24), let us introduce the finite element space

$$V^h = \left\{ v \in L^2(\Omega) \mid v|_K \in \mathcal{P}_p(K) \ \forall K \in \mathcal{T}^h \right\}, \tag{26}$$

where $\mathcal{P}_p(K)$ is the space of polynomials of degree at most p on K. The DG formulation then reads: find $u^h \in V^h$ such that, for all $K \in \mathcal{T}^h$, we have

$$\int_K v \frac{\partial \boldsymbol{u}^h}{\partial t} \, d\boldsymbol{x} - \int_K \nabla v \cdot \boldsymbol{\mathcal{F}}^h \, d\boldsymbol{x} + \int_{\partial K} v \boldsymbol{\mathcal{F}}^h \cdot \boldsymbol{n} \, ds \\ = \int_K v \boldsymbol{S} \, d\boldsymbol{x} \quad \forall v \in \mathcal{P}_p(K). \tag{27}$$

The discontinuous nature of the approximation requires, as in finite volume schemes, the introduction of suitable *numerical fluxes* $\boldsymbol{\mathcal{H}}(\boldsymbol{u}^+, \boldsymbol{u}^-, \boldsymbol{n})$ on the interfaces. This fact suggests that DG is not purely a finite element method, since the Galerkin weak form (27) is not enough to capture the physics underlying the conservation law (24), and some additional technology (upwind fluxes, Riemann solvers, etc.) has to be borrowed from the usual finite volume practice. The resulting scheme for hyperbolic problems is very compact, since interelement couplings are restricted to first neighbours only and, in particular, to those nodes that are shared with the adjacent elements. Unfortunately, this is no longer the case for problems involving higher order derivatives (like elliptic equations, or equations with terms of elliptic character).

Summing over all the elements, and considering conservative schemes, for which the numerical fluxes on a given edge are unique, we obtain the following global expression: find $u^h \in V^h$ such that

$$\begin{aligned}\int_\Omega v\,\frac{\partial \boldsymbol{u}^h}{\partial t}\,d\boldsymbol{x} - \int_\Omega \nabla^h v \cdot \boldsymbol{\mathcal{F}}^h\,d\boldsymbol{x} \\ + \int_{\mathcal{E}} (v^+ - v^-)\,\boldsymbol{\mathcal{H}}\,ds + \int_{\partial\Omega} v\,\boldsymbol{\mathcal{H}}\,ds \\ = \int_K v\,\boldsymbol{S}\,d\boldsymbol{x} \quad \forall v \in \mathcal{P}_p(K),\end{aligned} \tag{28}$$

where $\mathcal{E}$ denotes the union of all the interior edges and ∇^h is the "broken" gradient operator, i.e. $\nabla^h u$ is the function whose restriction to K is ∇u. One of the major problems that discontinuous Galerkin practitioners will have to face in order to make the method fully viable for engineering problems, apart from the very high computational cost associated to DG discretizations, is the fact that, for moderately or very high order schemes (say, from $p = 3$), which is precisely the range when DG is competitive compared to finite volume schemes, the shock-capturing properties of DG schemes are not much better than those of a continuous finite element method, i.e. quite poor. This is due to the fact that the amount of "upwinding" introduced by the use of finite volume-type numerical fluxes at the element edges is not sufficient to yield non-oscillatory solutions. The direct implementation of limiters with such high orders is probably meaningless, and shock-capturing techniques based on artificial viscosities face basically the same challenges as those designed for their continuous counterpart.

3.1.5 Spectral (Finite) Volume Methods

The so-called Spectral Volume method was introduced in a series of papers by Wang et al. [98–102]. The idea is somewhat between classical finite volume schemes and discontinuous Galerkin methods, in the sense that high-order schemes are constructed within the finite volume framework (piewise constant test functions, cell-averaged data representation), but the reconstruction is based on piecewise polynomial approximations, obtained by means of new degrees of freedom created inside each control volume through suitable cell subdivision.

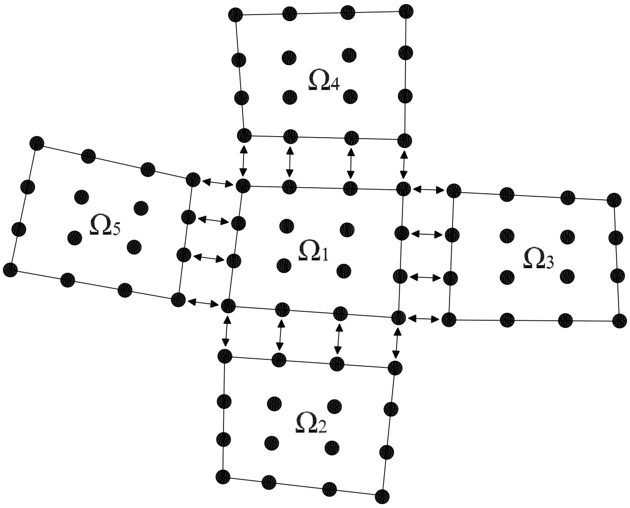


Fig. 3 Discontinuous Galerkin methods: piecewise polynomial approximation

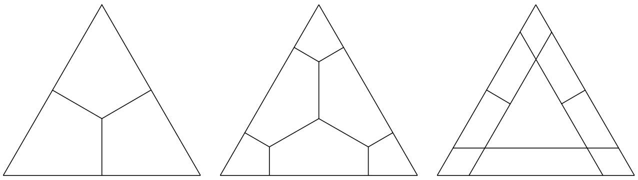


Fig. 4 Spectral (Finite) Volume method: typical cell partitions, $k = 2$ (*left*), $k = 3$ (*center*) and $k = 4$ (*right*)

Assume that the conservation law (24) is solved in a computational domain Ω, discretized into N non-verlapping (triangular) control volumes or cells Ω_i. The design of a numerical scheme with order of accuracy k starts with the partition of each control volume into $m = k(k+1)/2$ subcells, Ω_{ij}. Typical partitions for $k = 2, 3, 4$ are shown in Fig. 4.

The cell-averaged variables for subcell Ω_{ij} are defined as

$$u_{ij} = \frac{1}{A_i}\int_{\Omega_i} u\,d\Omega. \tag{29}$$

Given the cell-averaged variables for all the control volumes inside cell i, a polynomial $p(x, y) \in P^{k-1}$ is constructed, such that $p(x, y)$ is a k-th order approximation to $u(x, y)$ in Ω_i, by enforcing

$$\frac{1}{A_{ij}}\int_{\Omega_{ij}} p(x, y)\,d\Omega = u_{ij}. \tag{30}$$

The final reconstructing polynomial can be cast as

$$p_i(x, y) = \sum_{j=1}^{N} L_{ij} u_{ij}. \tag{31}$$

This reconstruction is then used to formulate finite volume scheme, whereby the integral form of the conservation law is enforced at each subcell. The semi-disctretized equations read

$$A_{ij}\frac{du_{ij}}{dt}+\sum_{e=1}^{E}\int_{\Gamma_e}\mathcal{F}\cdot\boldsymbol{n}\,ds, \tag{32}$$

where E is the number of faces in the subcell i.

The Spectral Volume method is similar to DG in the sense that both are somewhat generalizations of the finite volume method. The crucial differences are that piecewise constant test functions are employed in the SV scheme, and that the solution representation is based on cell-averages, rather than on point values. The polynomial approximation in DG is based on the powerful finite element approximation technology and, therefore, is more accurate and robust than the cell-partition approach of the SV method, which is potentially problematic, due to the difficulty of constructing good partitions for very high-order reconstructions. Both methods have been compared [90, 109], and DG has proven to be consistently more accurate that the SV method. On the other hand, the spectral volume approach has one clear advantage over DG, related to the fact that limiting is applied on a sub-cell basis, which constitutes a very important difference with respect to the rather poor shock-capturing capabilities of very high-order DG methods.

3.1.6 *Residual Distribution Schemes*

The interest in developing fully multidimensional upwind schemes has given rise to a very active and interesting area of research; probably the most successful family of methods developed so far in this context is the class of so-called *Residual Distribution* or *Fluctuation Splitting* schemes [4, 6–13, 41, 76, 77, 79, 80].

The starting point is again the conservation law

$$\frac{\partial u}{\partial t}+\nabla\cdot\mathcal{F}=0\quad\text{in }\Omega \tag{33}$$

and a triangulation of Ω, $\mathcal{T}^h=\{K\}$. The solution is approximated as in the finite element method, in particular using linear interpolation, as

$$u^h(\boldsymbol{x})=\sum u_j N_j(\boldsymbol{x}). \tag{34}$$

The flux balance over each triangle K defines the residual or *fluctuation*, as

$$\phi^K=\int_K\frac{\partial u^h}{\partial t}\,d\boldsymbol{x}=\int_{\partial K}\mathcal{F}\cdot\boldsymbol{n}\,ds. \tag{35}$$

An important concept for RD schemes is the definition of a *conservative linearization* of the flux, which may allow to write

$$\phi^K=\int_{\partial K}\mathcal{F}\cdot\boldsymbol{n}\,ds=\int_K\hat{\boldsymbol{\lambda}}\cdot\nabla u^h\,d\boldsymbol{x}=|K|\hat{\boldsymbol{\lambda}}\cdot\nabla u^h. \tag{36}$$

The above expression exploits the fact that ∇u^h is constant for linear triangles and $\hat{\boldsymbol{\lambda}}$ can be defined as

$$\hat{\boldsymbol{\lambda}}=\frac{1}{|K|}\int_K\boldsymbol{\lambda}\,d\boldsymbol{x}. \tag{37}$$

Residual Distribution schemes are based on distributing fractions of the residual (36) to the nodes of the element (Fig. 5). Using a forward Euler time-stepping for presentation purposes, the basic update scheme becomes

$$u_i^{n+1}=u_i^n-\frac{\Delta t}{S_i}\sum_K\beta_i^K\phi^K, \tag{38}$$

where S_i is the area of the median dual cell around node i and β_i^K the distribution coefficient applied to ϕ^T. Consistency requires that, for each triangle K,

$$\beta_1^K+\beta_2^K+\beta_3^K=1. \tag{39}$$

What defines each RD scheme is the actual way in which the coefficients β_i^K are chosen. There is a quite sound mathematical background concerning the various design criteria behind a particular RD scheme: multidimensional upwinding, positivity, linearity preservation and second order accuracy and continuity. It is interesting that some stabilized finite element methods (Streamline Upwind Petrov Galerkin, SUPG, methods), and finite volume methods, are closely related to RD schemes.

Higher order generalizations have been constructed, based on subtriangulations and extensions of the basic distribution strategy for second order schemes [9, 12]. The extension to quadrilateral meshes poses some difficulties, although some advances towards the extension of the residual distribution ideas to quadrilaterals have been recently published [12, 77].

Residual Distribution schemes are among the least dissipative schemes for hyperbolic problems, and the idea of multidimensional upwinding has proven to yield very good results in this context. Furthermore, their mathematical formulation and analysis is sound and powerful. The main drawbacks of this type of schemes is the use of a *conservative linearization*, which is not always available, and the viscous discretization and extension to higher order accuracy, which are not straightforward.

3.2 A Tale of Two Natures: Discretization of Elliptic Terms

The discretization of terms of elliptic character, such as those arising from diffusive processes, is not straightforward

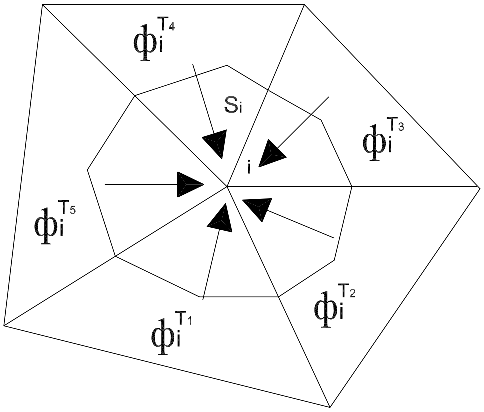


Fig. 5 Residual Distribution Schemes: Residual contributions to node *i*

with any of the aforementioned schemes. They are all naturally suited for hyperbolic problems, but their suitability for advection seems to be somewhat in conflict with a straightforward extension to diffusion problems. The strategies followed to address this issue vary according to the different schemes. Some of them, which have been used with more or less success, are presented in the following paragraphs.

3.2.1 The Standard Finite Volume Approach

The discretization of diffusion terms in finite volume methods is, at least in principle, inconsistent with the advection approach. As indicated in [94], the finite volume approach is basically pragmatic, in the sense that advective and diffusive discretizations are based on different spatial representations of the solution. Thus, the discretization of advective fluxes is based on discontinuous reconstructions, computed from the underlying cell-averaged data. On the other hand, the diffusive fluxes are often computed with cell-averages "reinterpreted" as point-wise function values, and the equivalent reconstruction is not conservative in general.

On unstructured grids, the most common approaches can be presented as

- Methods based on Green-Gauss reconstruction of the gradients. Following this approach, the gradients are evaluated as

$$\nabla u = \frac{1}{A_\Omega} \int_{\partial\Omega} u\,\boldsymbol{n}\,ds, \tag{40}$$

  where the path $\partial\Omega$ varies for the different implementations. The extension of the stencils resulting from this approach is one of its most important drawbacks. Nonetheless, this is the most frequent practice in state of the art second order finite volume codes. An interesting analysis of Green-Gauss gradient reconstructions can be found in [59].
- Methods that use averaged gradients. Thus, the piecewise polynomial reconstructions at each cell are used to compute two "side" gradients at the edges. These gradients, one from each side of the interface, are then averaged and used to evaluate the diffusive fluxes [75].

In the first case, the extension to higher order accuracy results in rapidly increasing complexity and size of the stencils, whereas the averaging approach has robustness issues when extended to higher-order accuracy.

3.2.2 Discontinuous Galerkin for Diffusion

The distinct nature of convection and diffusion processes, which arises in most "specialized" numerical schemes for convection-dominated flows, is also present in discontinuous Galerkin discretizations. Due to the piecewise polynomial approximation employed in DG, special care must be paid to those terms involving higher order derivatives.

The extension of DG methods to elliptic problems is a challenging and very active area of research [16, 17, 26, 28, 34, 78, 83, 107]. A unified framework of analysis of various existing schemes was presented in [17]. From a historical perspective, two main streams of development of discontinuous Galerkin methods for elliptic problems can be identified. On one side, the more recent schemes developed alongside the application of discontinuous Galerkin methods to hyperbolic problems [24, 34, 78]. On the other hand the so-called *interior penalty* (*IP*) methods, developed in the finite element community independently of the development of DG methods for hyperbolic problems [15, 42, 103]. It is shown in [17] that most of these methods, if not all, can be analyzed within a unified framework.

The most popular approach stems from the decomposition of the original problem, involving high order derivatives, into a system of first order equations, by introducing auxiliary variables for the solution gradient or flux. The resulting first order problem is discretized using the standard DG methodology, with a suitable definition of the interelement fluxes. In some cases (for example, when the additional variable is the gradient of the solution), the auxiliary variables can be eliminated locally. Amongst the most stable and accurate schemes are the second method of Bassi and Rebay (BR2 [24]), the Local Discontinuous Galerkin method (LDG [34]), and a newer variant of the latter, the Compact Discontinuous Galerkin method (CDG [78]). The CDG is similar to LDG in its formulation, but slightly more accurate and efficient, since the couplings are reduced to the nearest neighbours.

Consider, for example, the model Poisson problem

$$\begin{aligned} \nabla\cdot(-\kappa\nabla u) &= f && \text{in } \Omega, \\ u &= g_D && \text{on } \partial\Omega_D, \\ \kappa\frac{\partial u}{\partial n} &= g_N && \text{on } \partial\Omega_N, \end{aligned} \tag{41}$$

where $f(\boldsymbol{x})$ is a function on $L^2(\Omega)$ and $\kappa \in L^\infty(\Omega)$. In order to develop a DG discretization, the problem (41) is rewritten as a first order system of equations

$$\begin{aligned} \nabla \cdot \boldsymbol{\sigma} &= f && \text{in } \Omega, \\ \boldsymbol{\sigma} &= -\kappa \nabla u && \text{in } \Omega, \\ u &= g_D && \text{on } \partial\Omega_D, \\ \kappa \frac{\partial u}{\partial n} &= g_N && \text{on } \partial\Omega_N. \end{aligned} \tag{42}$$

Consider again the partition $\mathcal{T}^h$ of Ω. In addition to the space $V(\mathcal{T}^h)$ defined above, we also introduce the "broken" space $\Sigma(\mathcal{T}^h)$, which denotes the space of functions whose restriction to each element K belongs to the Sobolev space $[H^1(K)]$, that is

$$\begin{aligned} V &= \left\{ v \in L^2(\Omega) | v|_K \in H^1(K)\ \forall K \in \mathcal{T}_h \right\}, \\ \Sigma &= \left\{ \boldsymbol{\tau} \in \left[L^2(\Omega)\right]^2 |\boldsymbol{\tau}|_K \in \left[H^1(K)\right]^2\ \forall K \in \mathcal{T}_h \right\} \end{aligned} \tag{43}$$

and the finite element subspaces $V^h \in V$ and $\Sigma^h \in \Sigma$

$$\begin{aligned} V^h &= \left\{ v \in L^2(\Omega) | v|_K \in \mathcal{P}_p(K)\ \forall K \in \mathcal{T}_h \right\}, \\ \Sigma^h &= \left\{ \boldsymbol{\tau} \in \left[L^2(\Omega)\right]^2 \mid \boldsymbol{\tau}|_K \in \left[\mathcal{P}_p(K)\right]^2\ \forall K \in \mathcal{T}_h \right\}. \end{aligned} \tag{44}$$

Now consider DG discretizations of the form

$$\begin{aligned} &\int_K \boldsymbol{\sigma}^h \cdot \boldsymbol{\tau}\, d\boldsymbol{x} = -\int_K u^h \nabla \cdot (\kappa \boldsymbol{\tau})\, d\boldsymbol{x} + \int_{\partial K} \hat{u} \kappa \boldsymbol{\tau} \cdot \boldsymbol{n}\, ds \\ &\quad \forall \boldsymbol{\tau} \in \left[\mathcal{P}_p(K)\right]^2, \\ &\int_K \boldsymbol{\sigma}^h \cdot \nabla v\, d\boldsymbol{x} = -\int_K f\, v\, d\boldsymbol{x} + \int_{\partial K} \hat{\boldsymbol{\sigma}} \cdot \boldsymbol{n}\, v\, ds \\ &\quad \forall v \in \mathcal{P}_p(K). \end{aligned} \tag{45}$$

Then again we need to define numerical fluxes. Analogous expressions (with the location of the κ's changed in the equations) can be obtained for the, somewhat more convenient in the case a general purpose code, situation when the auxiliary variable is ∇u instead of the whole flux. Note that, in both cases, the most common practice is to defined, rather than *numerical fluxes*, some suitable *average states* for the variables, $\hat{u}$ and $\hat{\boldsymbol{\sigma}}$. One may wonder whether this is an appropriate choice in the nonlinear case.

The definition of the above numerical fluxes is critical, and affects not only the stability and accuracy of the scheme, but also its efficiency through the sparsity pattern and structure of the stiffness matrix ([17, 78]).

The LDG and CDG methods define $\hat{u}$ and $\hat{\boldsymbol{\sigma}}$ as

$$\begin{aligned} \hat{\boldsymbol{\sigma}} &= \{\boldsymbol{\sigma}^h\} - C_{11}\left[u^h\right] + \boldsymbol{C}_{12}\left[\boldsymbol{\sigma}\right], \\ \hat{u} &= \{u^h\} - \boldsymbol{C}_{12} \cdot \left[u^h\right] \end{aligned} \tag{46}$$

for the interior edges, and

$$\begin{aligned} \hat{\boldsymbol{\sigma}} &= \boldsymbol{\sigma}^h - C_{11}(u^h - g_D)\boldsymbol{n}, & \hat{u} &= g_D && \text{on } \partial\Omega_D, \\ \hat{\boldsymbol{\sigma}} &= g_N \boldsymbol{n}, & \hat{u} &= u^h && \text{on } \partial\Omega_N \end{aligned} \tag{47}$$

for the boundary edges. In the above expressions, C_{11} is a positive constant and $\boldsymbol{C}_{12}$ is a vector which is determined for each interior edge according to

$$\frac{1}{2}\left(S_{K^+}^{K^-}\boldsymbol{n}^+ + S_{K^-}^{K^+}\boldsymbol{n}^-\right), \tag{48}$$

where $S_{K^+}^{K^-} \in \{0, 1\}$ is a *switch* which is defined for each edge, satisfying

$$S_{K^+}^{K^-} + S_{K^-}^{K^+} = 1. \tag{49}$$

The average and jump operators are given by

$$\{\boldsymbol{\tau}\} = \frac{1}{2}\left(\boldsymbol{\tau}^+ + \boldsymbol{\tau}^-\right), \qquad \{v\} = \frac{1}{2}\left(v^+ + v^-\right), \tag{50}$$

$$[\boldsymbol{\tau}] = \boldsymbol{\tau}^+ \cdot \boldsymbol{n} + \boldsymbol{\tau}^- \cdot \boldsymbol{n}, \qquad [v] = v^+\boldsymbol{n} + v^-\boldsymbol{n}. \tag{51}$$

The viscous discretization in the so-called Spectral Volume methods follow the same basic procedures as discontinuous Galerkin methods [101].

We would like to comment on the approach suggested in [94] for the discretization of diffusive terms using DG. The authors acknowledge the distinct structure of advective and diffusive terms, and propose to construct a "*local smooth recovery*" of the solution at the interfaces, that would remove the discontinuity in a weak sense. This dual continuous-discontinuous spatial representation is also the basis for the higher order finite volume scheme proposed in this study. Our approach is somewhat the opposite, however, since we start from a continuous representation of the solution, which is then "broken" into piecewise polynomial expansions inside each cell, in order to deal with hyperbolic terms.

3.2.3 *Residual Distribution Schemes for Elliptic Problems*

Consider a model "diffusion" term of the form $\mu \Delta u$. For each element K, the idea of the RD discretization would be to distribute, among the nodes of K, the residual over K corresponding to this term, namely

$$\phi^{K,d} = \int_\Omega \mu \Delta u^h\, d\Omega. \tag{52}$$

Using linear (for triangles) or bilinear (for quadrilaterals) elements, and evaluating (52) following the standard finite element interpolation, the above residual is zero and, therefore, the discrete Laplacian operator needs to be defined in a different way.

One option is to used a hybrid discretization, where advective terms are discretized using the standard upwind techniques of the RD approach, whereas the diffusion terms are computed using a standard finite element Galerkin discretization. This approach has the advantages of being inexpensive and keeping the stencil compact [76]. Unfortunately, it is not consistent with the general RD formulation, and leads to schemes that are only first-order accurate [77].

Another option, which retains the residual based approach of RD methods, is to evaluate the viscous fluctuation by means of a contour integral,

$$\phi^{K,d} = \int_{\Omega} \mu \Delta u^h \, d\Omega = \int_{\Gamma} \mu \nabla u \cdot \boldsymbol{n} d\Gamma. \tag{53}$$

The solution gradient needs to be reconstructed at the nodes. To this end, procedures similar to those employed in the finite volume community (such as Green-Gauss reconstruction) are employed. In this approach, the diffusive terms are treated as source terms, and distributed with coefficients based on the convective terms. In turn, the stencil needs to be extended in order to compute the gradients at the nodes.

4 A High-order Finite Volume Scheme Based on Point-wise Data Representation and Moving Least-squares Approximation

This section presents the fundamental structure of the proposed finite volume method. Although the details about the approximation scheme will be presented later, the methodology can be derived by assuming a fairly general approximation framework. We are interested in schemes that acknowledge the distinct nature of convective and diffusive processes. In [94], a methodology to construct discretizations for diffusive terms is presented within the context of DG. The authors argue that the use of a discontinuous approximation, which is adequate to discretize the convective terms, is not well suited for viscous discretizations, and propose a *recovery* procedure to remove the discontinuity in a weak sense. Their idea is, thus, to *"locally recover the underlying smooth solution with sufficient fidelity"*.

Our approach is somewhat the opposite. We will start from a high-order continuous representation of the solution $u^h(\boldsymbol{x})$, obtained by means of Moving Least-Squares approximation, and suitable for general unstructured grids. In order to deal with hyperbolic terms, this continuous reconstruction is *"broken"* locally (inside each cell) into piecewise polynomial approximations to the reconstructed solution, $u^{hb}(\boldsymbol{x} - \boldsymbol{x}_I)$. As a consequence, the "broken" reconstruction is also a high-order approximation to the underlying smooth solution. These piecewise polynomial approximations are constructed by means of Taylor series expansions of $u^h(\boldsymbol{x})$ around the reference point (node) $\boldsymbol{x}_I$, and require the successive derivatives of $u^h(\boldsymbol{x})$, which are also given within the Moving Least-Square framework. The following paragraphs elaborate on the above ideas, and introduce the proposed methodology.

Consider again a system of conservation laws of the form

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \left(\boldsymbol{\mathcal{F}}^I(\boldsymbol{u}) + \boldsymbol{\mathcal{F}}^V(\boldsymbol{u}, \nabla \boldsymbol{u}) \right) = \boldsymbol{S}(\boldsymbol{x}, \boldsymbol{u}, \nabla \boldsymbol{u}) \quad \text{in } \Omega \tag{54}$$

supplemented with suitable initial and boundary conditions. The fluxes have been split into an "inviscid" (hyperbolic) part, and a "viscous" (elliptic) part. Consider, in addition, a partition of the domain Ω into a set of non-overlapping *control volumes* or *cells*, $\mathcal{T}^h = \{I\}$. Furthermore, we define a reference point (*node*), $\boldsymbol{x}_I$, inside each cell (the cell centroid), where the numerical solution $\boldsymbol{u}_I$ is computed.

The general spatial representation of the solution is as follows: consider a function $u(\boldsymbol{x})$, given by its point values, $u_j = u(\boldsymbol{x}_j)$, at the nodes or reference locations, one inside each cell. The approximated or reconstructed function $u^h(\boldsymbol{x})$ is sought for in the subspace spanned by a set of *basis functions* $N_j(\boldsymbol{x})$ associated to the nodes, such that $u^h(\boldsymbol{x})$ is a continuous function of the form

$$u^h(\boldsymbol{x}) = \sum_{j=1}^{n_x} N_j(\boldsymbol{x}) u_j \tag{55}$$

which states that the approximation at a certain point $\boldsymbol{x}$ is computed using $n_{\boldsymbol{x}}$ surrounding nodes. The reconstruction can also be seen, in analogy to finite element spaces, as written in terms of nodal shape functions $N_j(\boldsymbol{x})$. Note that the reconstructed function $u^h(\boldsymbol{x})$, using Moving Least-Squares approximation, is not a polynomial in general.

Consider now the integral form of the system of conservation laws (54) which, for each control volume I, reads

$$\int_{\Omega_I} \frac{\partial \boldsymbol{u}}{\partial t} d\Omega + \int_{\Gamma_I} \left(\boldsymbol{F}^I(\boldsymbol{u}) + \boldsymbol{F}^V(\boldsymbol{u}, \nabla \boldsymbol{u}) \right) \cdot \boldsymbol{n} \, d\Gamma = \int_{\Omega_I} \boldsymbol{S}(\boldsymbol{x}, \boldsymbol{u}, \nabla \boldsymbol{u}) \, d\Omega. \tag{56}$$

Note that this expression can be obtained through a weighted residuals procedure, just by taking test functions that are constant inside each control volume and zero elsewhere. Introducing the component-wise reconstructed function $\boldsymbol{u}^h$, the above expression reads

$$\int_{\Omega_i} \frac{\partial \boldsymbol{u}^h}{\partial t} d\Omega + \int_{\Gamma_i} \left(\boldsymbol{F}^I(\boldsymbol{u}^h) + \boldsymbol{F}^V(\boldsymbol{u}^h, \nabla \boldsymbol{u}^h) \right) \cdot \boldsymbol{n} \, d\Gamma = \int_{\Omega_i} \boldsymbol{S}(\boldsymbol{x}, \boldsymbol{u}^h, \nabla \boldsymbol{u}^h) \, d\Omega. \tag{57}$$

Note that the reconstructed function $\boldsymbol{u}^h(\boldsymbol{x})$ is continuous and, therefore, the numerical scheme (57) is conservative (the normal fluxes are unique). The continuity of the reconstruction is quite convenient for elliptic/parabolic equations (or terms of elliptic character). However, we would like to make use of the classical finite volume technology to discretize hyperbolic equations (or terms of hyperbolic character). To this end, we introduce a *"broken"* reconstruction, $\boldsymbol{u}_I^{hb}(\boldsymbol{x})$, which approximates $\boldsymbol{u}^h(\boldsymbol{x})$ (and, therefore, $\boldsymbol{u}(\boldsymbol{x})$) locally inside each cell I, and is discontinuous across cell interfaces. In general, we require the order of accuracy of the broken reconstruction to be the same as that of the original continuous reconstruction i.e.

$$\|u - u^h\| \leq C_1 h^{k+1} \tag{58}$$

and

$$\|u - u_I^{hb}\|_I \leq C_2 h^{k+1} \tag{59}$$

for some constants k, C_1 and C_2, and a characteristic cell size h. One possible choice is to use Taylor series expansion of the form

$$\begin{aligned}
&\boldsymbol{u}_I^{hb}(x - x_I, y - y_i) \\
&\quad = \boldsymbol{u}_I^h + \nabla \boldsymbol{u}_I^h \cdot (\boldsymbol{x} - \boldsymbol{x}_I) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_I)^T \boldsymbol{H}^h \ (\boldsymbol{x} - \boldsymbol{x}_I) \\
&\qquad + \frac{1}{6}\boldsymbol{\Delta}^2 \boldsymbol{x}_I^T \boldsymbol{T}_I (\boldsymbol{x} - \boldsymbol{x}_I) + \cdots,
\end{aligned} \tag{60}$$

where the gradient ∇u_I^h, the Hessian matrix $\boldsymbol{H}^h$, and

$$\begin{aligned}
&\boldsymbol{\Delta}^2 \boldsymbol{x}_I^T = \left((x - x_I)^2 \quad (y - y_I)^2\right), \\
&\boldsymbol{T}_I = \begin{pmatrix} \frac{\partial^3 \boldsymbol{u}_I}{\partial x^3} & 3\frac{\partial^3 \boldsymbol{u}_I}{\partial x^2 \partial y} \\ 3\frac{\partial^3 \boldsymbol{u}_I}{\partial x \partial y^2} & \frac{\partial^3 \boldsymbol{u}_I}{\partial y^3} \end{pmatrix}
\end{aligned} \tag{61}$$

involve the successive derivatives of the continuous reconstruction $\boldsymbol{u}^h(\boldsymbol{x})$.

This dual continuous/discontinuous reconstruction of the solution is crucial in order to obtain accurate and efficient numerical schemes for mixed hyperbolic/parabolic problems.

The cell-wise broken reconstruction defined here is actually a piecewise continuous approximation to $\boldsymbol{u}^h$. The advantage is that it allows us to make use of Riemann solvers, limiters, and other standard finite volume technologies, while keeping some consistency in terms of functional representation. Thus, the general continuous reconstruction is used to evaluate the viscous (elliptic-like) fluxes, whereas its discontinuous approximation is used to evaluate the inviscid (hyperbolic-like) fluxes.

The final semidiscrete scheme can be written as

$$\begin{aligned}
&\int_{\Omega_I} \frac{\partial \boldsymbol{u}^h}{\partial t}\, d\Omega + \int_{\Gamma_I} \boldsymbol{H}(\boldsymbol{u}^{hb+}, \boldsymbol{u}^{hb-}, \boldsymbol{n})\, d\Gamma \\
&\quad + \int_{\Gamma_I} \boldsymbol{F}^V(\boldsymbol{u}^h, \nabla \boldsymbol{u}^h) \cdot \boldsymbol{n}\, d\Gamma \\
&\quad = \int_{\Omega_I} \boldsymbol{S}(\boldsymbol{x}, \boldsymbol{u}^h, \nabla \boldsymbol{u}^h)\, d\Omega,
\end{aligned} \tag{62}$$

where $\boldsymbol{H}(\boldsymbol{u}^{hb+}, \boldsymbol{u}^{hb-}, \boldsymbol{x})$ is a suitable numerical flux. The direct evaluation of the fluxes corresponding to the elliptic terms assumes that certain continuity requirements on $\boldsymbol{u}^h$, that depend on the order of the derivatives to be discretized, are satisfied.

Note that our dual reconstruction procedure induces a non-diagonal mass matrix. Most existing finite volume schemes recover, in principle, the diagonal structure of the mass matrix, by enforcing reconstructions that preserve the mean. It is actually not clear whether this is actually the case when elliptic terms are present, due to the "pragmatic" flux evaluation approach, which may have some influence on the underlying solution representation, and therefore in the structure of the consistent mass matrix. One advantage of the proposed scheme, at least in terms of formulation, is that the approximation framework is completely clear.

In our case, the structure of the consistent mass matrix is of the form $\boldsymbol{M} = \{m_{ij}\}$, where

$$m_{ij} = \int_{\Omega_i} N_j(\boldsymbol{x})\, d\Omega. \tag{63}$$

As we work with point-wise values, enforcing $m_{ij} = 0$ for $i \neq j$ may have deleterious effects on the accuracy of the reconstructed function $\boldsymbol{u}^h(\boldsymbol{x})$, in addition to increasing the complexity of the design of an approximation scheme. We prefer $\boldsymbol{u}^h$ to be as accurate as possible, even is this produces the inconvenience of a non-diagonal mass matrix.

One possible option to recover the diagonality of the mass matrix (apart from straightforward lumping procedures) is to replace the continuous reconstruction in the time-dependent term by the discontinuous one, and enforce conservation of the mean in this reconstruction $\boldsymbol{u}^{hb}$. This can be accomplished, for example, through the use of zero-mean polynomials in the expansion (60). For example, the quadratic case reads

$$\begin{aligned}
&\boldsymbol{u}_I^{hb}(x - x_I, y - y_i) \\
&\quad = \boldsymbol{u}_I^h + \nabla \boldsymbol{u}_I^h \cdot (\boldsymbol{x} - \boldsymbol{x}_I) + \frac{1}{2}\left((x - x_I)^2 - \widetilde{x^2}\right) \left.\frac{\partial^2 \boldsymbol{u}^h}{\partial x^2}\right|_{\boldsymbol{x}_I} \\
&\qquad + \left((x - x_I)(y - y_I) - \widetilde{xy}\right) \left.\frac{\partial^2 \boldsymbol{u}^h}{\partial x \partial y}\right|_{\boldsymbol{x}_I} \\
&\qquad + \frac{1}{2}\left((y - y_I)^2 - \widetilde{y^2}\right) \left.\frac{\partial^2 \boldsymbol{u}^h}{\partial y^2}\right|_{\boldsymbol{x}_I},
\end{aligned}$$

where

$$\widetilde{x^2} = \frac{1}{A_I} \int_{\Omega_I} (x - x_I)^2\, d\Omega,$$

$$\widetilde{xy} = \frac{1}{A_I}\int_{\Omega_I}(x-x_I)(y-y_I)\,d\Omega, \tag{64}$$

$$\widetilde{y^2} = \frac{1}{A_I}\int_{\Omega_I}(y-y_I)^2\,d\Omega.$$

This specification of the polynomial reconstruction can also be obtained through a suitable numerical quadrature. Following this procedure, the resulting diagonal mass matrix would be consistent with the discretization of the hyperbolic terms (as they use the broken reconstruction), but "lumped" with respect to the viscous discretization. In terms of the diagonality of the mass matrix, this reconstruction based on zero-mean polynomials is equivalent to the mean-conservation restrictions used in other finite volume schemes based on cell averages. Note that, in our context, conservation of the mean appears in a very specific context of the formulation, and does not need to be imposed a priori in the reconstruction. For steady-state computations we do not require the polynomial reconstruction to preserve the mean values.

5 Moving Least-squares Reproducing Kernel Approximations

5.1 General Formulation

Consider a function $u(\boldsymbol{x})$ defined in a domain Ω. Moving Least-Squares (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

$$u(\boldsymbol{x}) \approx \widehat{u}(\boldsymbol{x}) = \sum_{i=1}^{m} p_i(\boldsymbol{x})\alpha_i(\boldsymbol{z})\Big|_{\boldsymbol{z}=\boldsymbol{x}} = \boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{\alpha}(\boldsymbol{z})\Big|_{\boldsymbol{z}=\boldsymbol{x}}, \tag{65}$$

where $\boldsymbol{p}^T(\boldsymbol{x})$ 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

$$J(\boldsymbol{\alpha}(\boldsymbol{z})|_{\boldsymbol{z}=\boldsymbol{x}}) = \int_{\boldsymbol{y}\in\Omega_{\boldsymbol{x}}} W(\boldsymbol{z}-\boldsymbol{y},h)\Big|_{\boldsymbol{z}=\boldsymbol{x}} \times \left[u(\boldsymbol{y}) - \boldsymbol{p}^T(\boldsymbol{y})\boldsymbol{\alpha}(\boldsymbol{z})\Big|_{\boldsymbol{z}=\boldsymbol{x}}\right]^2 d\Omega_{\boldsymbol{x}} \tag{66}$$

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 $\boldsymbol{z}=\boldsymbol{x}$. Note that, even if all the basis functions in $\boldsymbol{p}^T(\boldsymbol{x})$ are polynomials, the reconstructed function $\widehat{u}(\boldsymbol{x})$ is not a polynomial in general.

The parameter h, usually called *smoothing length* or *dilation 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,

$$W(\boldsymbol{x}-\boldsymbol{y},h) = \begin{cases} 1-\frac{3}{2}s^2+\frac{3}{4}s^3, & s\le 1,\\ \frac{1}{4}(2-s)^3, & 1<s\le 2,\\ 0, & s>2, \end{cases} \tag{67}$$

where $s = \frac{|\boldsymbol{x}-\boldsymbol{y}|}{h}$. In practice, the minimization of (66) 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*.

Thus, the minimization of J with respect to the set of parameters $\boldsymbol{\alpha}$ leads to the expression

$$\int_{\boldsymbol{y}\in\Omega_{\boldsymbol{x}}} \boldsymbol{p}(\boldsymbol{y})W(\boldsymbol{z}-\boldsymbol{y},h)\Big|_{\boldsymbol{z}=\boldsymbol{x}} u(\boldsymbol{y})d\Omega_{\boldsymbol{x}} = \boldsymbol{M}(\boldsymbol{x})\boldsymbol{\alpha}(\boldsymbol{z})\Big|_{\boldsymbol{z}=\boldsymbol{x}}, \tag{68}$$

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

$$\boldsymbol{M}(\boldsymbol{x}) = \int_{\boldsymbol{y}\in\Omega_{\boldsymbol{x}}} \boldsymbol{p}(\boldsymbol{y})W(\boldsymbol{z}-\boldsymbol{y},h)\Big|_{\boldsymbol{z}=\boldsymbol{x}} \boldsymbol{p}^T(\boldsymbol{y})d\Omega_{\boldsymbol{x}}. \tag{69}$$

The above integrals are 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

$$\boldsymbol{\alpha}(\boldsymbol{z})\Big|_{\boldsymbol{z}=\boldsymbol{x}} = \boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{P}_{\Omega_{\boldsymbol{x}}}\boldsymbol{W}_V(\boldsymbol{x})\boldsymbol{u}_{\Omega_{\boldsymbol{x}}}, \tag{70}$$

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. 6)

$$\boldsymbol{u}_{\Omega_{\boldsymbol{x}}} = \begin{pmatrix} u(\boldsymbol{x}_1) & u(\boldsymbol{x}_2) & \cdots & u(\boldsymbol{x}_{n_{\boldsymbol{x}}}) \end{pmatrix}^T. \tag{71}$$

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

$$\boldsymbol{P}_{\Omega_{\boldsymbol{x}}} = \boldsymbol{p}(\boldsymbol{x}_1) \quad \boldsymbol{p}(\boldsymbol{x}_2) \quad \cdots \quad \boldsymbol{p}(\boldsymbol{x}_{n_{\boldsymbol{x}}}), \tag{72}$$

$$\boldsymbol{W}_V(\boldsymbol{x}) = \mathrm{diag}\{W_i(\boldsymbol{x}-\boldsymbol{x}_i)V_i\}, \quad i=1,\ldots,n_{\boldsymbol{x}}. \tag{73}$$

Complete details can be found in [67, 70]. 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 $\boldsymbol{W}_V$, obtaining an MLS version of the Reproducing Kernel Particle Method [70]. Otherwise, we can use $\boldsymbol{W}$ instead of $\boldsymbol{W}_V$

$$\boldsymbol{W}(\boldsymbol{x}) = \operatorname{diag}\{W_i(\boldsymbol{x}-\boldsymbol{x}_i)\}, \quad i = 1,\dots,n_{\boldsymbol{x}} \tag{74}$$

which corresponds to the classical MLS approximation (in the nodal integration of the functional (66), the same quadrature weight is associated to all nodes). Introducing (70) in (65), the interpolation structure can be identified as

$$\begin{aligned}\widehat{u}(\boldsymbol{x}) &= \boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{P}_{\Omega_x}\boldsymbol{W}(\boldsymbol{x})\boldsymbol{u}_{\Omega_x}\\ &= \boldsymbol{N}^T(\boldsymbol{x})\boldsymbol{u}_{\Omega_x} = \sum_{j=1}^{n_x} N_j(\boldsymbol{x})u_j.\end{aligned} \tag{75}$$

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

$$\boldsymbol{N}^T(\boldsymbol{x}) = \boldsymbol{p}^T(\boldsymbol{x})\boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{P}_{\Omega_x}\boldsymbol{W}(\boldsymbol{x}), \tag{76}$$

where $N_j(\boldsymbol{x})$ can be seen as the shape function associated to particle j.

The MLS shape functions are not interpolants in general, in the sense that the value of the original function at a certain node i, $u(\boldsymbol{x}_i)$, does not necessarily coincide with the MLS-reconstructed value, $\widehat{u}(\boldsymbol{x}_i)$, i.e.

$$u_i = u(\boldsymbol{x}_i) \neq \widehat{u}(\boldsymbol{x}_i) = \sum_{j=1}^{n_{x_i}} N_j(\boldsymbol{x}_i)u_j \tag{77}$$

in general. In particular,

$$N_j(\boldsymbol{x}_i) \neq \delta_{ij}, \tag{78}$$

where δ_{ij} is the Kroneker delta function.

The functional basis $\boldsymbol{p}(\boldsymbol{x})$ is strongly related to the accuracy of the MLS fit. Theory and numerical evidence [52] 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 (polynomial, 2-complete), reads

$$\boldsymbol{p}(\boldsymbol{x}) = \begin{pmatrix}1 & x_1 & x_2 & x_1x_2 & x_1^2 & x_2^2\end{pmatrix}^T \tag{79}$$

and the $p=3$ basis is given by

$$\boldsymbol{p}(\boldsymbol{x}) = \begin{pmatrix}1 & x_1 & x_2 & x_1x_2 & x_1^2 & x_2^2 & x_1^2x_2 & x_1x_2^2 & x_1^3 & x_2^3\end{pmatrix}^T. \tag{80}$$

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 $\boldsymbol{p}(\frac{\boldsymbol{x}-\boldsymbol{x}_I}{h})$, instead of $\boldsymbol{p}(\boldsymbol{x})$. With this transformation, the MLS shape functions read

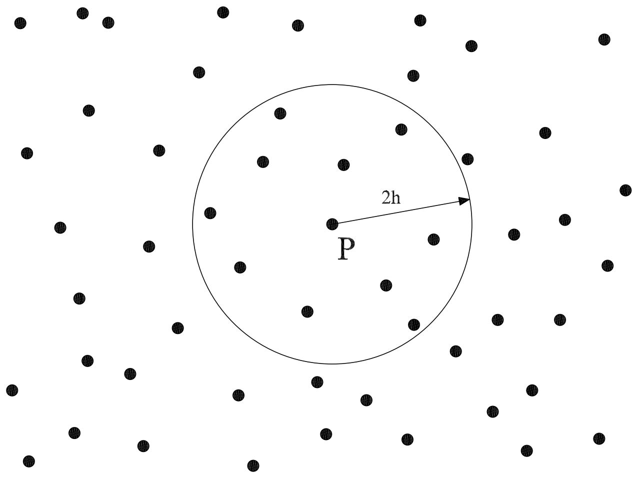


Fig. 6 Meshfree approximation: general scheme. Support for reconstruction at P

$$\boldsymbol{N}^T(\boldsymbol{x}_I) = \boldsymbol{p}^T(\boldsymbol{0})\boldsymbol{C}(\boldsymbol{x}_I) = \boldsymbol{p}^T(\boldsymbol{0})\boldsymbol{M}^{-1}(\boldsymbol{x}_I)\boldsymbol{P}_{\Omega_{x_I}}\boldsymbol{W}(\boldsymbol{x}_I), \tag{81}$$

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

$$\boldsymbol{C}(\boldsymbol{x}) = \boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{P}_{\Omega_x}\boldsymbol{W}(\boldsymbol{x}). \tag{82}$$

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 $\boldsymbol{p}(\frac{\boldsymbol{x}-\boldsymbol{x}_I}{h})$ and the derivatives of $\boldsymbol{C}(\boldsymbol{x})$ [36, 38–40].

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, very rough grids may require the use of full derivatives.

The approximate derivatives of $u(\boldsymbol{x})$ can be expressed in terms of the derivatives of the MLS shape functions. For example, the first and second order derivatives of $u(\boldsymbol{x})$, evaluated at $\boldsymbol{x}_I$, are given by

$$\left.\frac{\partial u(\boldsymbol{x})}{\partial x_\alpha}\right|_{\boldsymbol{x}=\boldsymbol{x}_I} \approx \sum_{j=1}^{n_{x_I}} u_j \left.\frac{\partial N_j(\boldsymbol{x})}{\partial x_\alpha}\right|_{\boldsymbol{x}=\boldsymbol{x}_I} \tag{83}$$

and

$$\left.\frac{\partial^2 u(\boldsymbol{x})}{\partial x_\alpha \partial x_\beta}\right|_{\boldsymbol{x}=\boldsymbol{x}_I} \approx \sum_{j=1}^{n_{x_I}} u_j \left.\frac{\partial^2 N_j(\boldsymbol{x})}{\partial x_\alpha \partial x_\beta}\right|_{\boldsymbol{x}=\boldsymbol{x}_I}. \tag{84}$$

Where the first and second order derivatives of the shape functions can be obtained as

$$\frac{\partial \boldsymbol{N}^T(\boldsymbol{x})}{\partial x_\alpha} = \frac{\partial \boldsymbol{p}^T(\boldsymbol{0})}{\partial x_\alpha}\boldsymbol{C}(\boldsymbol{x}) + \boldsymbol{p}^T(\boldsymbol{0})\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x_\alpha}, \tag{85}$$

$$\begin{aligned}\frac{\partial^2 \boldsymbol{N}^T(\boldsymbol{x})}{\partial x_\alpha \partial x_\beta} &= \frac{\partial^2 \boldsymbol{p}^T(\boldsymbol{0})}{\partial x_\alpha \partial x_\beta}\boldsymbol{C}(\boldsymbol{x}) + \boldsymbol{p}^T(\boldsymbol{0})\frac{\partial^2 \boldsymbol{C}(\boldsymbol{x})}{\partial x_\alpha \partial x_\beta} \\ &\quad + \frac{\partial \boldsymbol{p}^T(\boldsymbol{0})}{\partial x_\alpha}\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x_\beta} + \frac{\partial \boldsymbol{p}^T(\boldsymbol{0})}{\partial x_\beta}\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x_\alpha},\end{aligned} \tag{86}$$

where

$$\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x_\alpha} = \boldsymbol{C}(\boldsymbol{x})\boldsymbol{W}^{-1}(\boldsymbol{x})\frac{\partial \boldsymbol{W}(\boldsymbol{x})}{\partial x_\alpha}\left(\boldsymbol{I} - \boldsymbol{P}_{\Omega_x}^T \boldsymbol{C}(\boldsymbol{x})\right) \tag{87}$$

and

$$\begin{aligned}&\frac{\partial^2 \boldsymbol{C}(\boldsymbol{x})}{\partial x_\alpha \partial x_\beta} \\ &\quad = \frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x_\beta}\boldsymbol{W}^{-1}(\boldsymbol{x})\frac{\partial \boldsymbol{W}(\boldsymbol{x})}{\partial x_\alpha}\left(\boldsymbol{I} - \boldsymbol{P}_{\Omega_x}^T \boldsymbol{C}(\boldsymbol{x})\right) \\ &\qquad - \boldsymbol{C}(\boldsymbol{x})\frac{\partial \boldsymbol{W}(\boldsymbol{x})}{\partial x_\beta}\boldsymbol{W}^{-2}(\boldsymbol{x})\frac{\partial \boldsymbol{W}(\boldsymbol{x})}{\partial x_\alpha}\left(\boldsymbol{I} - \boldsymbol{P}_{\Omega_x}^T \boldsymbol{C}(\boldsymbol{x})\right) \\ &\qquad + \boldsymbol{C}(\boldsymbol{x})\boldsymbol{W}^{-1}(\boldsymbol{x})\frac{\partial^2 \boldsymbol{C}(\boldsymbol{x})}{\partial x_\alpha \partial x_\beta}\left(\boldsymbol{I} - \boldsymbol{P}_{\Omega_x}^T \boldsymbol{C}(\boldsymbol{x})\right) \\ &\qquad - \boldsymbol{C}(\boldsymbol{x})\boldsymbol{W}^{-1}(\boldsymbol{x})\frac{\partial \boldsymbol{W}(\boldsymbol{x})}{\partial x_\alpha}\boldsymbol{P}_{\Omega_x}^T\frac{\partial \boldsymbol{C}(\boldsymbol{x})}{\partial x_\beta}.\end{aligned} \tag{88}$$

More details of the MLS procedure used in this paper can be found in [36, 40].

5.2 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, as exposed below. The definition of the cloud (the *MLS stencil*) for each evaluation point is an important issue that will also be addressed in the following sections. 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 neighours, as

$$h = k \max\left(\|\boldsymbol{x}_j - \boldsymbol{x}_I\|\right). \tag{89}$$

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 $2h$). In the following section we elaborate on the implementation of anisotropic (tensor-product) kernels.

5.3 Anisotropic Node Distributions

The methodology described so far, with radial weighting in two and three dimensions, in not well suited for highly anisotropic node distributions. This situation corresponds, in terms of the application considered in this study, to the presence of cells with high aspect ratios, which is, on the other hand, a quite frequent circumstance in most practical applications of the Navier-Stokes equations. Fortunately, Moving Least Squares can account for this kind of cloud structure, by means of the use of anisotropic kernels.

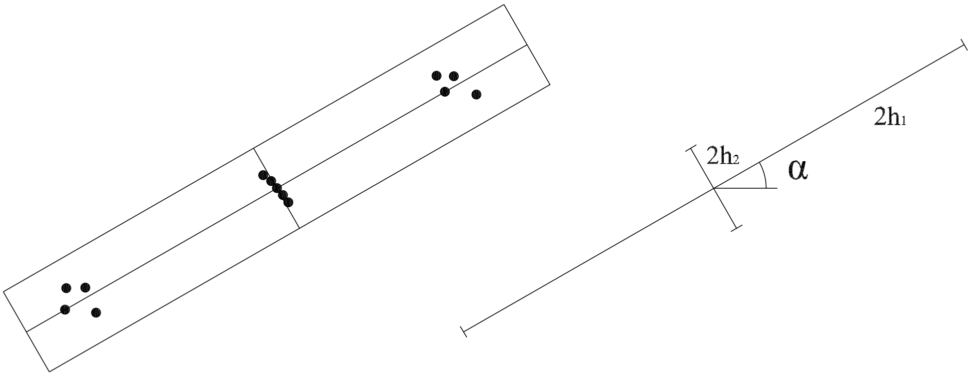
Consider a general node distribution that may include strongly directional features (Fig. 7). Rather than using radial kernels, we will construct our MLS shape functions using tensor-product kernels of the form

$$W^D(\boldsymbol{x} - \boldsymbol{y}, \boldsymbol{h}) = \prod_{i=1}^{D} W(x_i - y_i, h_i), \tag{90}$$

where $W(x - y, h)$ is the basic one-dimensional kernel used to construct the D-dimensional kernel $W^D(\boldsymbol{x} - \boldsymbol{y}, \boldsymbol{h})$, $\boldsymbol{h} = (h_1, \ldots, h_D)$ is the vector of directional smoothing lengths, and $(x_1, \ldots, x_D)$ (resp. $(y_1, \ldots, y_D)$) are the Cartesian coordinates of node $\boldsymbol{x}$ (resp. $\boldsymbol{y}$).

In addition to setting different smoothing lengths for each coordinate direction, we should also include the directionality properties of the cloud of nodes (Fig. 7), i.e. the fact that the principal directions of anisotropy of the cloud may not be aligned with the Cartesian coordinate axis. Focusing our analysis on the two-dimensional case, this can be addressed

Fig. 7 Anisotropic kernels: tensor-product support definition



by the use of rectangular supports (tensor-product kernels) defined by the length of their sides ($L_1 = 4h_1$, $L_2 = 4h_2$) and a rotation angle α. We determine this rotation through an inertial analogy, by looking at the principal directions of the tensor

$$\boldsymbol{I} = \begin{pmatrix} I_{xx} & I_{xy} \\ I_{yx} & I_{yy} \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^{N}(x_j - x_I)^2 & \sum_{j=1}^{N}(x_j - x_I)(y_j - y_I) \\ \sum_{j=1}^{N}(x_j - x_I)(y_j - y_I) & \sum_{j=1}^{N}(y_j - y_I)^2 \end{pmatrix}. \tag{91}$$

In particular, the angle α is given by

$$\tan(2\alpha) = \frac{2I_{xy}}{I_{xx} - I_{yy}}. \tag{92}$$

Once the directionality of the cloud has been characterized, through the computation of α, it is more convenient to compute the MLS shape functions in a rotated space, with the transformed coordinate axis (x^*, y^*) aligned with the principal directions of the tensor $\boldsymbol{I}$. To this end the nodes in the cloud are rotated according to

$$\boldsymbol{x}^* = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \boldsymbol{x}, \tag{93}$$

where $\boldsymbol{x} = (x_1 \ x_2)^T$ stands for coordinates in the *physical* space, and $\boldsymbol{x}^* = (x_1^* \ x_2^*)^T$ are the coordinates in the *transformed* space. Once the coordinates of the nodes in the cloud have been rotated, the MLS shape functions and their derivatives are computed in the transformed space using smoothing lengths

$$h_1^* = k_1 \max(|x_{1j}^* - x_{1I}^*|), \\ h_2^* = k_2 \max(|x_{2j}^* - x_{2I}^*|) \tag{94}$$

and the corresponding tensor-product kernels

$$W(\boldsymbol{x}^* - \boldsymbol{y}^*, \boldsymbol{h}^*) = W(x_1^* - y_1^*, h_1^*)\, W(x_2^* - y_2^*, h_2^*). \tag{95}$$

While the shape functions in the transformed space are equal to the shape functions in the physical space, the derivatives have to be rotated back to the physical space. For example, the first derivatives read

$$\begin{pmatrix} \frac{\partial N}{\partial x_1} \\ \frac{\partial N}{\partial x_2} \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} \frac{\partial N}{\partial x_1^*} \\ \frac{\partial N}{\partial x_2^*} \end{pmatrix} \tag{96}$$

while the second derivatives are computed as

$$\begin{pmatrix} \frac{\partial^2 N}{\partial x_1^2} & \frac{\partial^2 N}{\partial x_1 x_2} \\ \frac{\partial^2 N}{\partial x_1 x_2} & \frac{\partial^2 N}{\partial x_2^2} \end{pmatrix} = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} \frac{\partial^2 N}{\partial x_1^{*2}} & \frac{\partial^2 N}{\partial x_1^* x_2^*} \\ \frac{\partial^2 N}{\partial x_1^* x_2^*} & \frac{\partial^2 N}{\partial x_2^{*2}} \end{pmatrix} \times \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix}. \tag{97}$$

These basic expressions are easily extended to higher order derivatives.

5.4 Moving Least Squares vs. Piecewise Polynomial Interpolation

As mentioned in the above state of the art review, 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 [35]), or using some suitable form of cell subdivision (such as the so-called Spectral Volume method [102]). 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

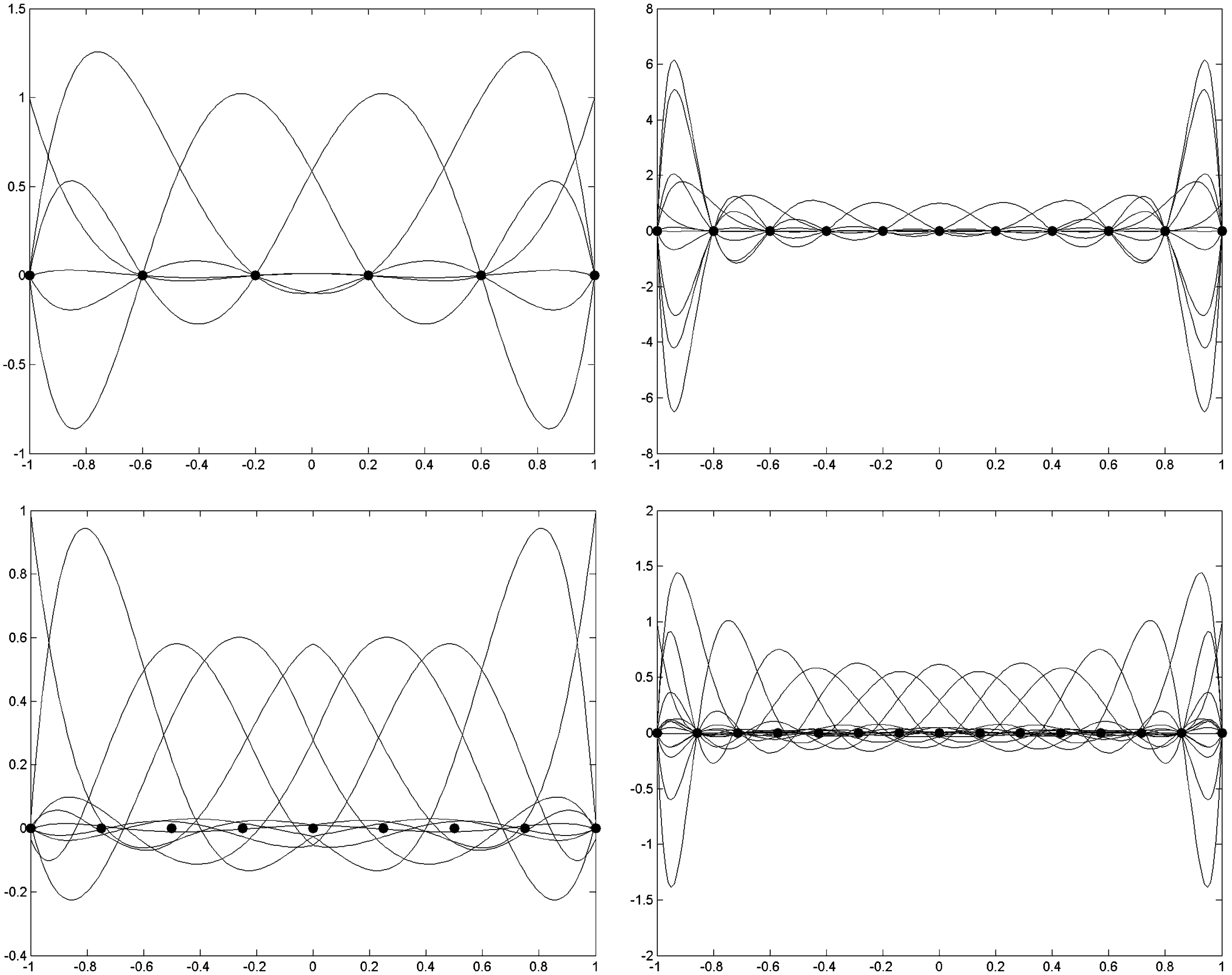


Fig. 8 MLS shape functions for $p = 5$, $N = 9$ (*bottom left*), and $p = 10$, $N = 15$ (*bottom right*). Lagrange basis for $p = 5$ (*top left*) and $p = 10$ (*top right*)

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. 8 and 9 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.

Figure 8 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 $h = 0.6d_{max}$, where d_{max} is the maximum of the distances between the evaluation point and its neighbours. Figure 9 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

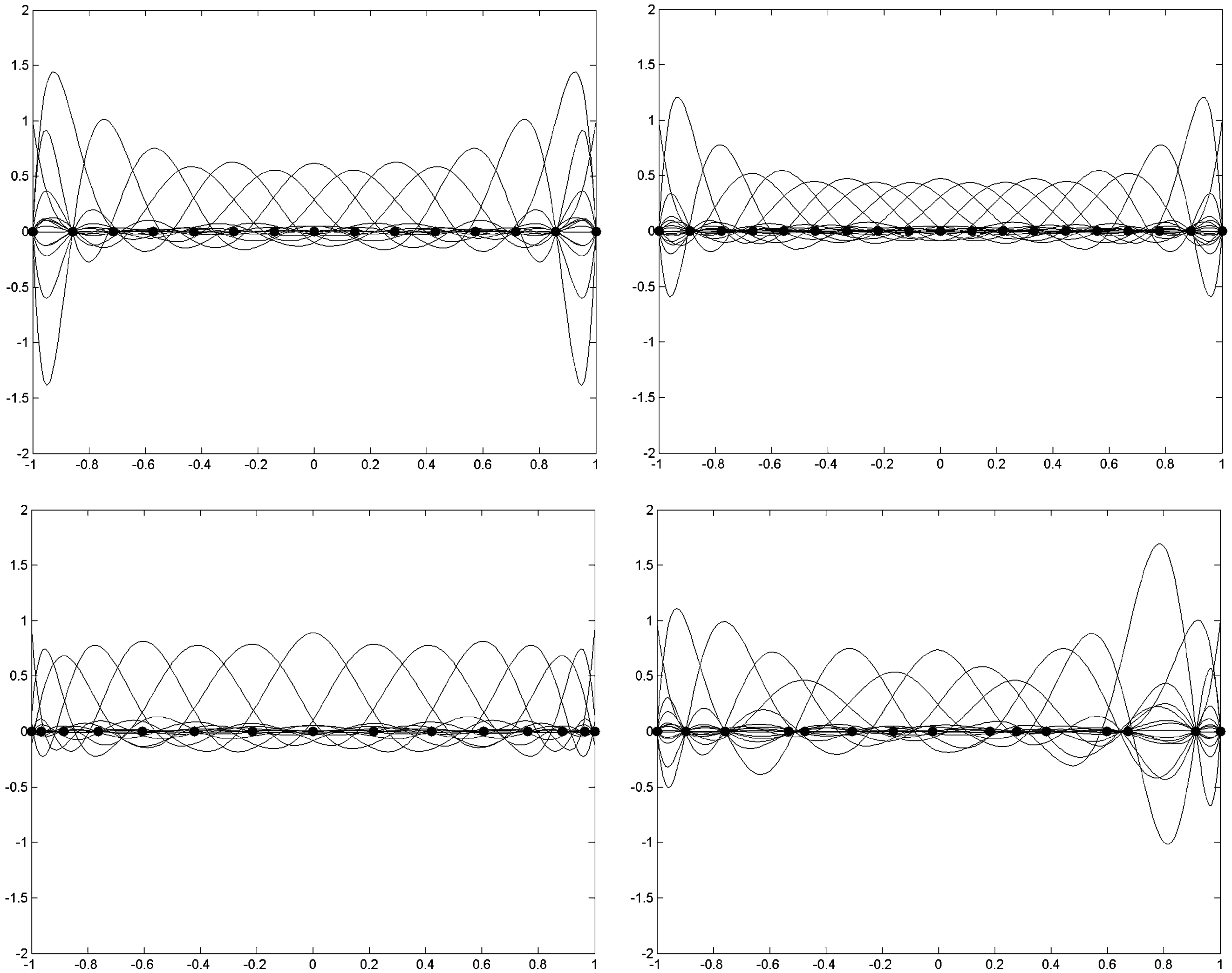


Fig. 9 MLS shape functions for $p = 10$: $N = 15$ uniform points (*top left*), $N = 19$ uniform points (*top right*), $N = 15$ Gauss-Lobatto points (*bottom left*) and $N = 15$ random points (*bottom right*)

"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 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.

Figures 10 and 11 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. Figure 10 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 polynomial order. The same trend is observed for irregularly spaced point distributions, as depicted in Fig. 11. Note that in Fig. 11 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. 10, 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 multidimen-

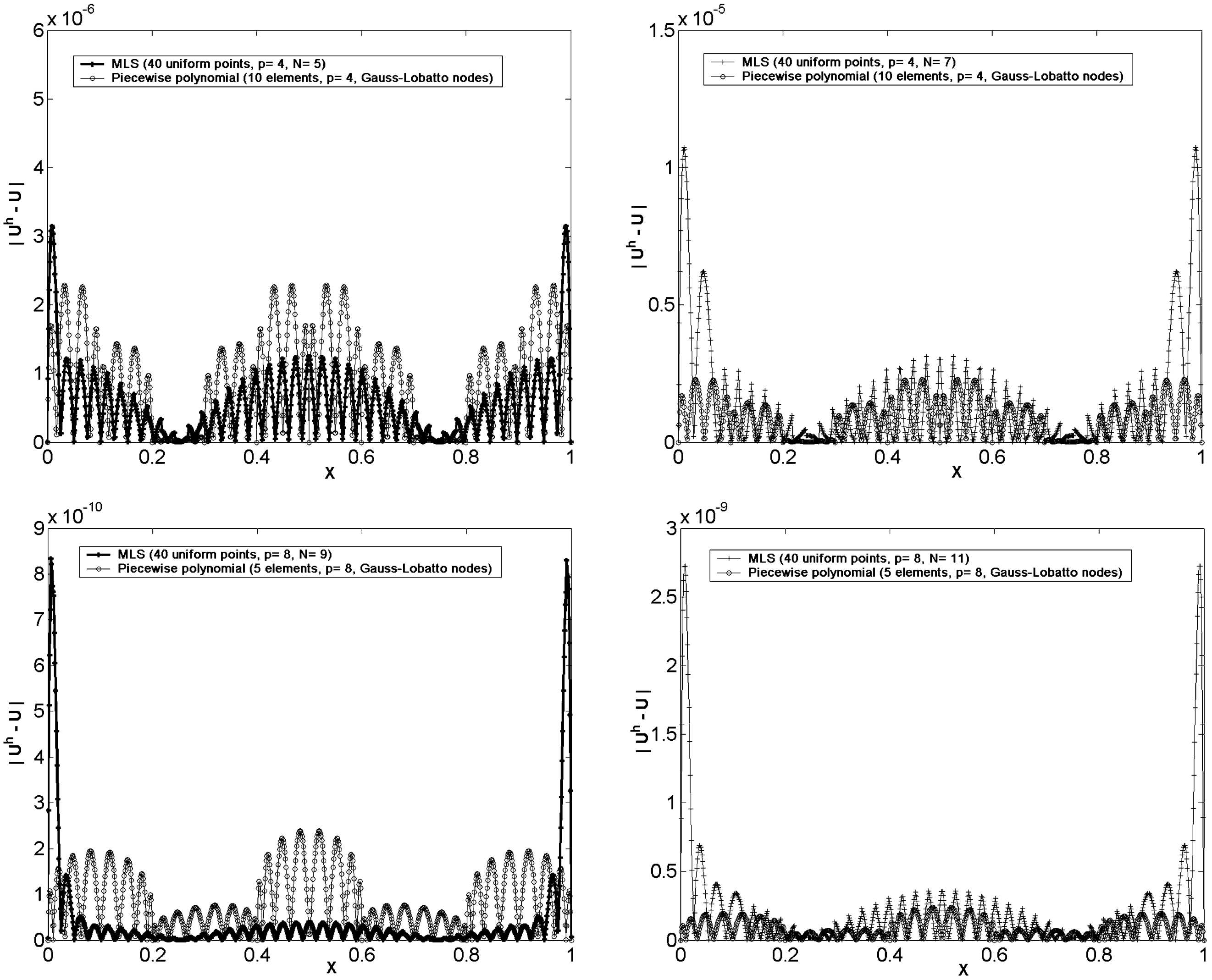


Fig. 10 Error distributions with MLS and piecewise polynomial interpolations, $u(x) = \sin(2\pi x)$. *Top*, $p = 4$, MLS with $N = 5$ (*left*) and $N = 7$ (*right*), 40 uniform points, Lagrange polynomials, 10 elements with Gauss-Lobatto nodes. *Bottom*, $p = 8$, MLS with $N = 9$ (*left*) and $N = 7$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

sional 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. 10–11. 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.

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. Even in 1D, values of p beyond 10 cannot be used in practice due to the extremely ill-conditioned problem they pose near the boundaries. In multidimensions the threshold is even lower, although in practice 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.

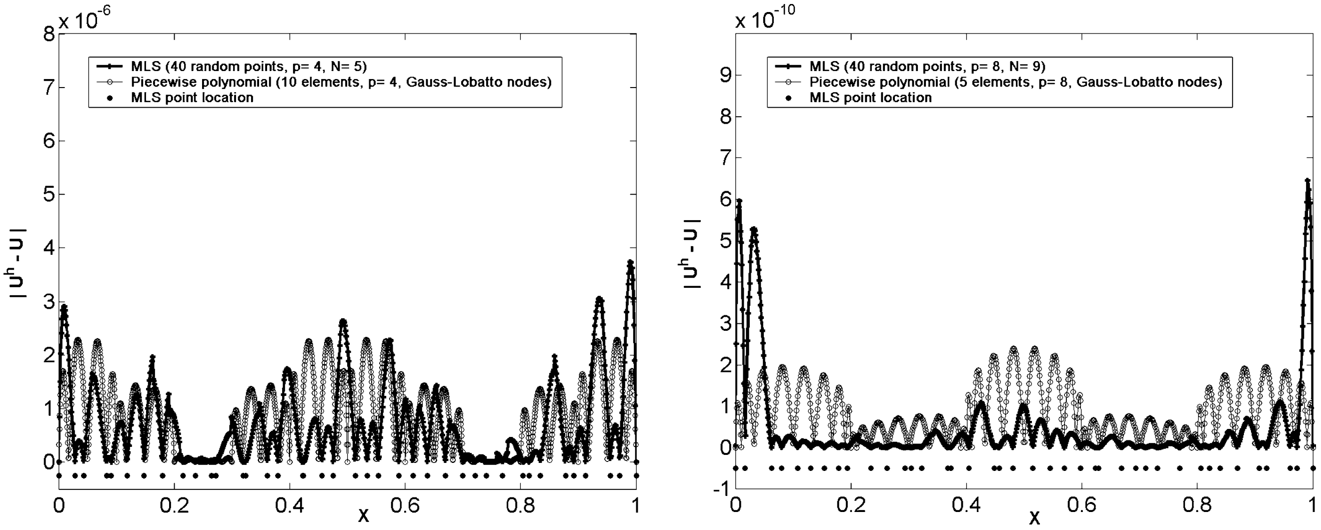


Fig. 11 Error distributions with MLS and piecewise polynomial interpolations, $u(x) = \sin(2\pi x)$. *Left*, $p = 4$, MLS with $N = 5$, 40 random points, Lagrange polynomials, 10 elements with Gauss-Lobatto nodes. *Right*, $p = 8$, MLS with $N = 9$, 40 random points, Lagrange polynomials, 5 elements with Gauss-Lobatto nodes

5.5 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 [69] 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

$$u_h(\boldsymbol{x}) = \sum_{j=1}^{n} u_j N_j^h(\boldsymbol{x}), \qquad u_{2h}(\boldsymbol{x}) = \sum_{j=1}^{n} u_j N_j^{2h}(\boldsymbol{x}). \tag{98}$$

A set of wavelet functions is obtained as

$$\boldsymbol{\Phi}^{2h}(\boldsymbol{x}) = \boldsymbol{N}^h(\boldsymbol{x}) - \boldsymbol{N}^{2h}(\boldsymbol{x}) \tag{99}$$

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

$$u_h(\boldsymbol{x}) = u_{2h}(\boldsymbol{x}) + \boldsymbol{\Psi}^{2h}(\boldsymbol{x}), \tag{100}$$

where

$$\begin{aligned}\boldsymbol{\Psi}^{2h}(\boldsymbol{x}) &= \sum_{j=1}^{n} u_j \Phi_j^{2h}(\boldsymbol{x}) \\ &= \sum_{j=1}^{n} u_j \left(N_j^h(\boldsymbol{x}) - N_j^{2h}(\boldsymbol{x})\right).\end{aligned} \tag{101}$$

The low-scale $u_{2h}(\boldsymbol{x})$ can be further decomposed using the same rationale. Figure 12 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 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 below, the limiters can be switched off in those areas where $\boldsymbol{\Psi}^{2h}$ is lower than a certain threshold, therefore retaining the whole accuracy of the scheme in smooth regions. 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

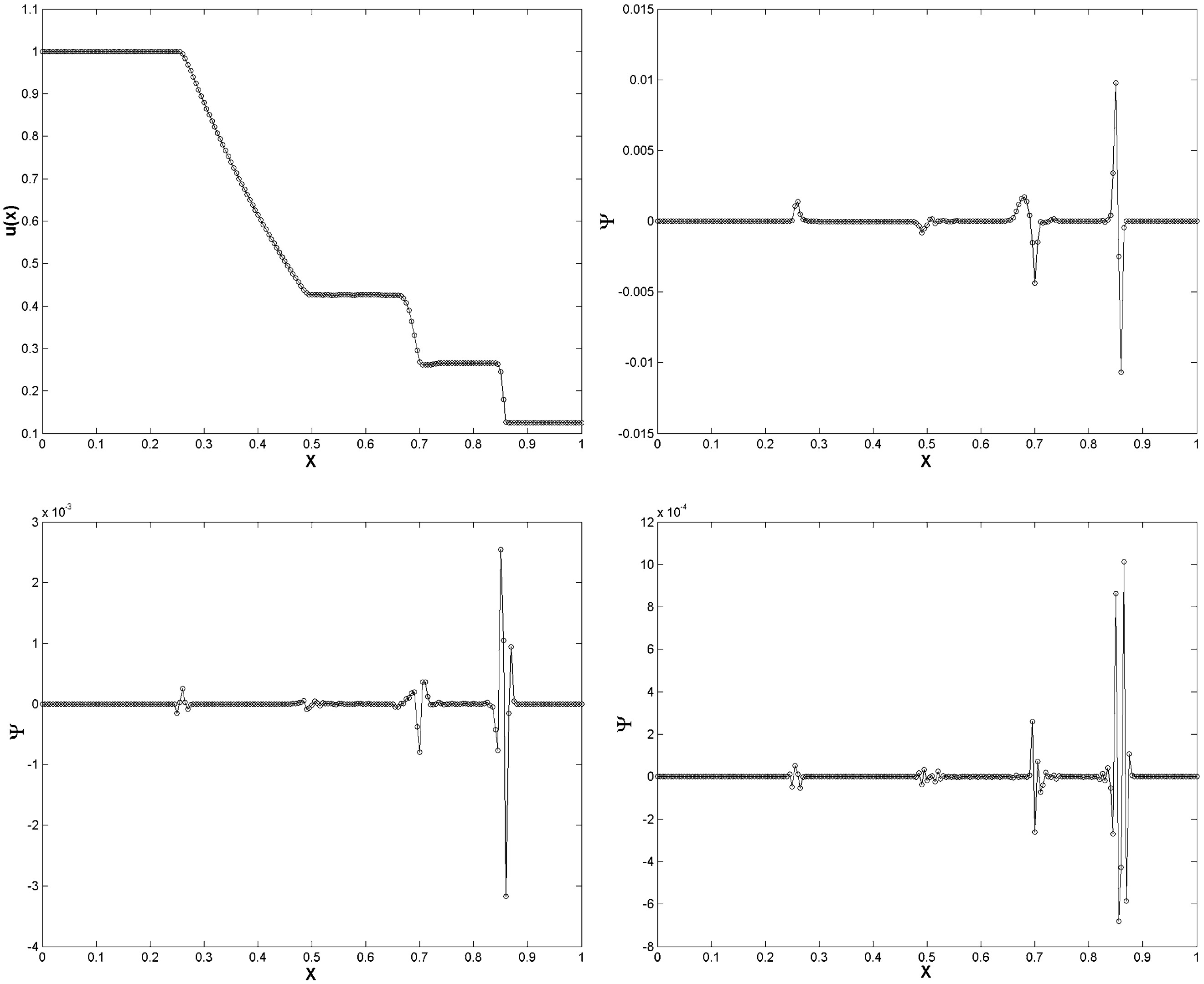


Fig. 12 Multiscale analysis: $u(x)$ (*top left*), and its high-scale component Ψ^{2h}, using $p = 1$, $N = 3$ (*top right*), $p = 3$, $N = 6$ (*bottom left*), and $p = 6$, $N = 9$ (*bottom right*)

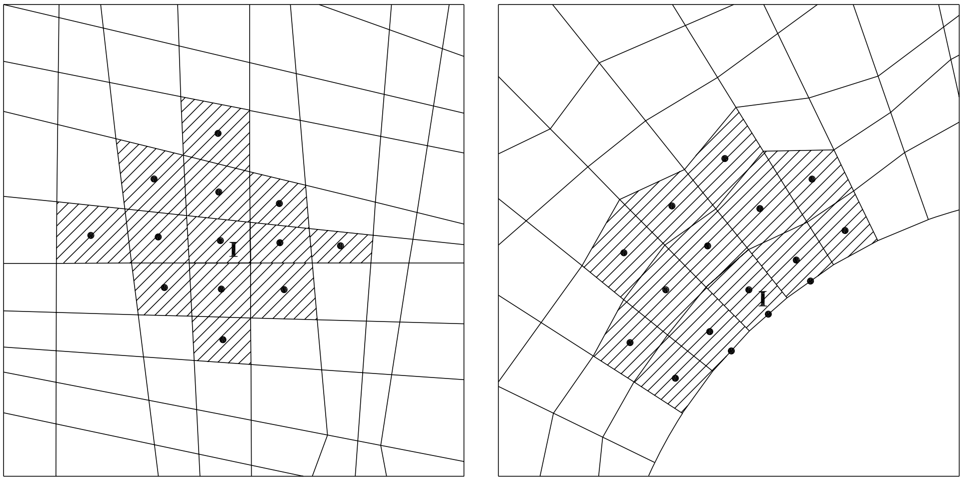
unstructured grid generalization of the wavelet-based selective filtering proposed by Sjögreen and Yee for finite differences [87].

5.6 Moving Least-Squares, Finite Volume Solvers, Unstructured Grids

The proposed methodology uses 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 [1] for an introduction). This approximation framework induces a non-diagonal mass matrix.
- For hyperbolic problems, or terms of hyperbolic character, the generalized Godunov method [21, 51, 54] is adopted. We use "broken" piecewise polynomial reconstructions based on the MLS general approximation and Taylor series expansions. The successive derivatives of the conserved variables at the cell centroids are computed using MLS approximations. Therefore, rather than creating new degrees of freedom inside each cell, we use in-

Fig. 13 $p = 3$ MLS stencil: centroids



formation 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.

6 Practical Implementation Aspects

6.1 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) develop 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.

The following sections elaborate on the MLS stencils used in this study for the cubic basis ($p = 3$).

6.2 $p = 3$ MLS Stencils: I. Centroids

Figure 13 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* [59]). 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.

6.3 $p = 3$ MLS Stencils: II. Edges

Figure 14 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.

6.4 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. Figure 15 (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. Figure 15 (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

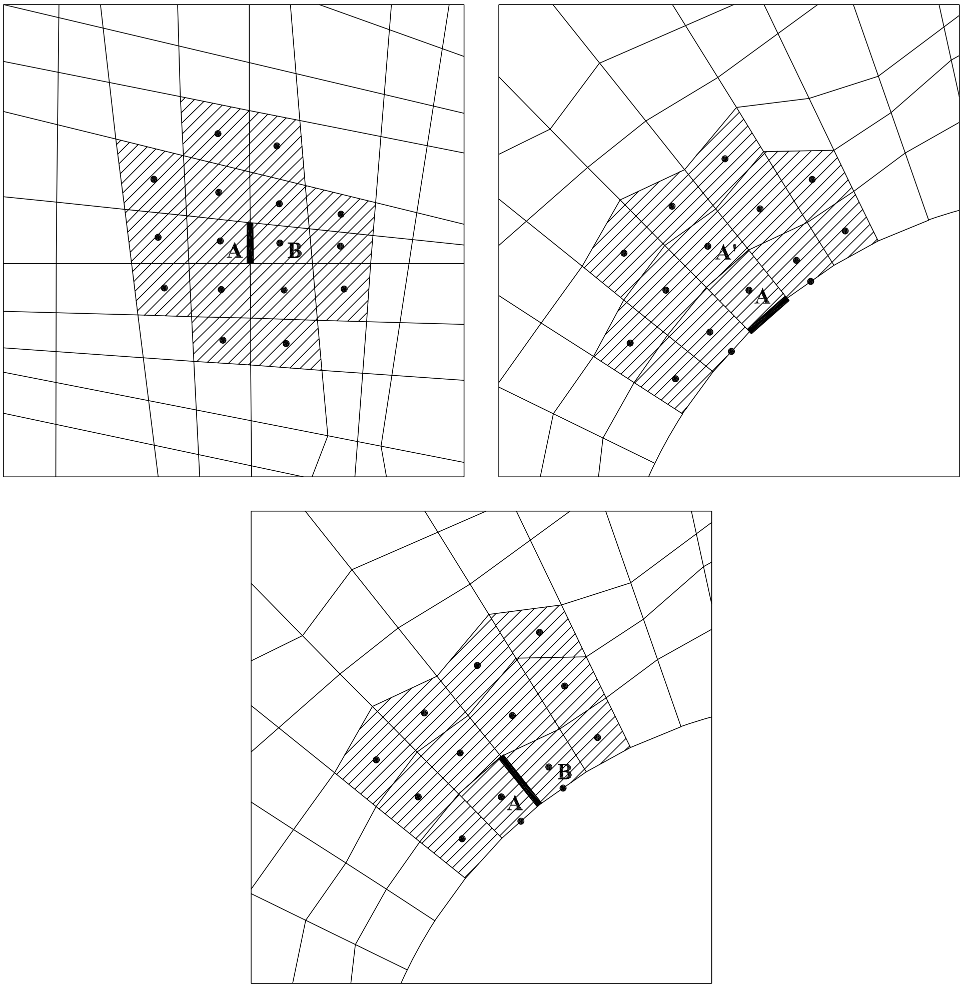


Fig. 14 $p = 3$ MLS stencil: quadrature points on edges

stencil as a subset. Note that, quite the opposite to what is usually thought about finite volume schemes, this stencil is actually quite compact. In the examples section we present a comparison with the DG stencil for elliptic problems.

6.5 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 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. 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 $u^h(\boldsymbol{x})$, provided by the Moving Least-Squares approximants, and then compute local discontinuous approximations $u_I^{hb}(\boldsymbol{x})$, 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

$$u_I^{hb}(\boldsymbol{x}) = u_I + \nabla u_I \cdot (\boldsymbol{x} - \boldsymbol{x}_I), \qquad (102)$$

where u_I stands for the centroid value, $\boldsymbol{x}_I$ denotes spatial coordinates of the centroid of the cell and $\boldsymbol{\nabla} u_I$ 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

$$u_I^{hb}(\boldsymbol{x}) = u_I + \nabla u_I \cdot (\boldsymbol{x} - \boldsymbol{x}_I) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_I)^T \boldsymbol{H}_I (\boldsymbol{x} - \boldsymbol{x}_I), \qquad (103)$$

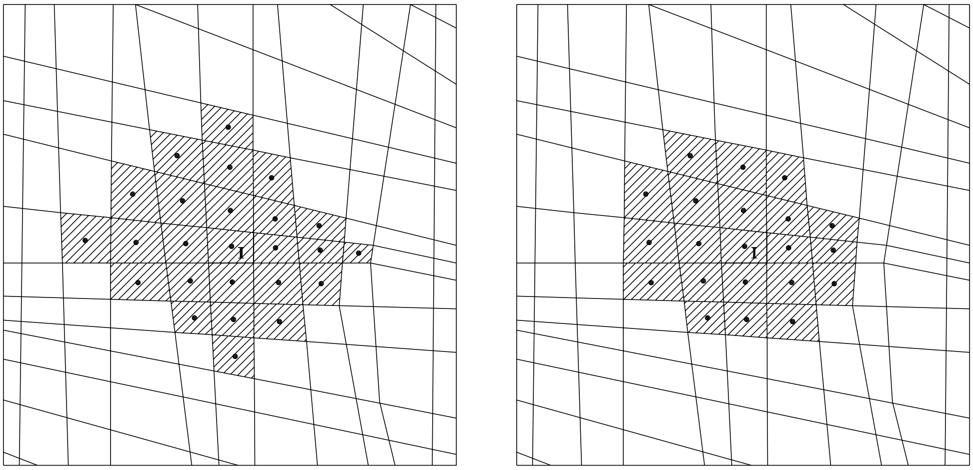


Fig. 15 Full fourth order MLS-FV stencil: Euler and Navier-Stokes (*left*) and elliptic problems (*right*)

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

$$u_I^{hb}(\boldsymbol{x}) = u_I + \nabla u_I \cdot (\boldsymbol{x} - \boldsymbol{x}_I) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}_I)^T \boldsymbol{H}_I (\boldsymbol{x} - \boldsymbol{x}_I) + \frac{1}{6}\boldsymbol{\Delta}^2 \boldsymbol{x}_I^T \boldsymbol{T}_I (\boldsymbol{x} - \boldsymbol{x}_I), \tag{104}$$

where

$$\boldsymbol{\Delta}^2 \boldsymbol{x}_I^T = \left((x - x_I)^2 \quad (y - y_I)^2 \right),$$

$$\boldsymbol{T}_I = \begin{pmatrix} \frac{\partial^3 u_I}{\partial x^3} & 3\frac{\partial^3 u_I}{\partial x^2 \partial y} \\ 3\frac{\partial^3 u_I}{\partial x \partial y^2} & \frac{\partial^3 u_I}{\partial y^3} \end{pmatrix}. \tag{105}$$

For unsteady problems, additional terms must be introduced in (103) and (104) to enforce conservation of the mean, i.e.

$$\frac{1}{A_I} \int_{\boldsymbol{x} \in \Omega_I} u(\boldsymbol{x})\, d\Omega = u_I \tag{106}$$

as exposed in Sect. 4. The derivatives of the field variables are directly computed at centroids using MLS. Thus, the approximate gradients read

$$\nabla u_I = \sum_{j=1}^{n_{\boldsymbol{x}_I}} u_j \nabla N_j(\boldsymbol{x}_I), \tag{107}$$

where the u_j's stand for variables at the $n_{\boldsymbol{x}_I}$ "neighbour" (in the sense of the MLS stencil) centroids. The second order derivatives read

$$\frac{\partial^2 u_I}{\partial x^2} = \sum_{j=1}^{n_{\boldsymbol{x}_I}} u_j \frac{\partial^2 N_j(\boldsymbol{x}_I)}{\partial x^2}, \qquad \frac{\partial^2 u_I}{\partial x \partial y} = \sum_{j=1}^{n_{\boldsymbol{x}_I}} u_j \frac{\partial^2 N_j(\boldsymbol{x}_I)}{\partial x \partial y}, \tag{108}$$
$$\frac{\partial^2 u_I}{\partial y^2} = \sum_{j=1}^{n_{\boldsymbol{x}_I}} u_j \frac{\partial^2 N_j(\boldsymbol{x}_I)}{\partial y^2}.$$

Finally, the third order derivatives are written as

$$\frac{\partial^3 u_I}{\partial x^3} = \sum_{j=1}^{n_{\boldsymbol{x}_I}} u_j \frac{\partial^3 N_j(\boldsymbol{x}_I)}{\partial x^3}, \qquad \frac{\partial^3 u_I}{\partial x^2 \partial y} = \sum_{j=1}^{n_{\boldsymbol{x}_I}} u_j \frac{\partial^3 N_j(\boldsymbol{x}_I)}{\partial x^2 \partial y},$$
$$\frac{\partial^3 u_I}{\partial x \partial y^2} = \sum_{j=1}^{n_{\boldsymbol{x}_I}} u_j \frac{\partial^3 N_j(\boldsymbol{x}_I)}{\partial x \partial y^2}, \qquad \frac{\partial^3 u_I}{\partial y^3} = \sum_{j=1}^{n_{\boldsymbol{x}_I}} u_j \frac{\partial^3 N_j(\boldsymbol{x}_I)}{\partial y^3}. \tag{109}$$

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.

6.5.1 Limiters: I. Monotonicity Enforcement

Barth and Jespersen [23] have proposed an extension of Van Leer's scheme [93] which is suitable for unstructured grids. The basic idea is to enforce "*monotonicity*" in the reconstructed solution. In this context, monotonicity implies that no new extrema are created by the reconstruction process [23]. The enforcement is local, in the sense that only certain neighbour cells are considered for the "no new extrema" criterion.

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

$$u_I^{hb}(\boldsymbol{x})_I = u_I + \nabla u_I \cdot (\boldsymbol{x} - \boldsymbol{x}_I) \tag{110}$$

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

$$u_I^{hb}(\boldsymbol{x})_I = u_I + \Phi_I \nabla u_I \cdot (\boldsymbol{x} - \boldsymbol{x}_I), \tag{111}$$

where Φ_I is a slope limiter ($0 \leq \Phi_I \leq 1$) such that the reconstruction (111) satisfies

$$u^{\min} \leq u_I^{hb}(\boldsymbol{x})_I \leq u^{\max} \tag{112}$$

being

$$u^{\max} = \min_{j\in\mathcal{A}_I}(u_j), \qquad u^{\max} = \max_{j\in\mathcal{A}_I}(u_j), \tag{113}$$

where $\mathcal{A}_I$ is the set of "neighbour" cells. In practice, the restriction (112) 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

$$\Phi_I^q = \begin{cases} \min(1, \frac{u^{\max}-u_I}{u_I^q-u_I}), & u_I^q - u_I > 0, \\ \min(1, \frac{u^{\min}-u_I}{u_I^q-u_I}), & u_I^q - u_I < 0, \\ 1, & u_I^q - u_I = 0 \end{cases} \tag{114}$$

and, finally,

$$\Phi_I = \min_q(\Phi_I^q). \tag{115}$$

In the case of the quadratic reconstruction, (103), a similar limiting strategy is adopted

$$u_I^{hb}(\boldsymbol{x}) = u_I + \Phi_I\Big(\nabla u_I\cdot(\boldsymbol{x}-\boldsymbol{x}_I) + \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_I)^T\boldsymbol{H}_I(\boldsymbol{x}-\boldsymbol{x}_I)\Big), \tag{116}$$

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. 16-A). In the following, the above limiter will be referred to as "*BJ limiter*".

6.5.2 Limiters: II. Averaged Derivatives

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

$$\nabla u_I = \sum_{k=1}^N \omega_k \nabla u_k, \tag{117}$$

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 [59], the weights $\{\omega_k,\ k=1,\ldots,N\}$ are given by

$$\omega_k(g_1,g_2,\ldots,g_N) = \frac{\prod_{i\neq k}^N g_i + \epsilon^{N-1}}{\sum_{j=1}^N\left(\prod_{i\neq j}^N g_i\right) + N\epsilon^{N-1}}, \quad k=1,\ldots,N, \tag{118}$$

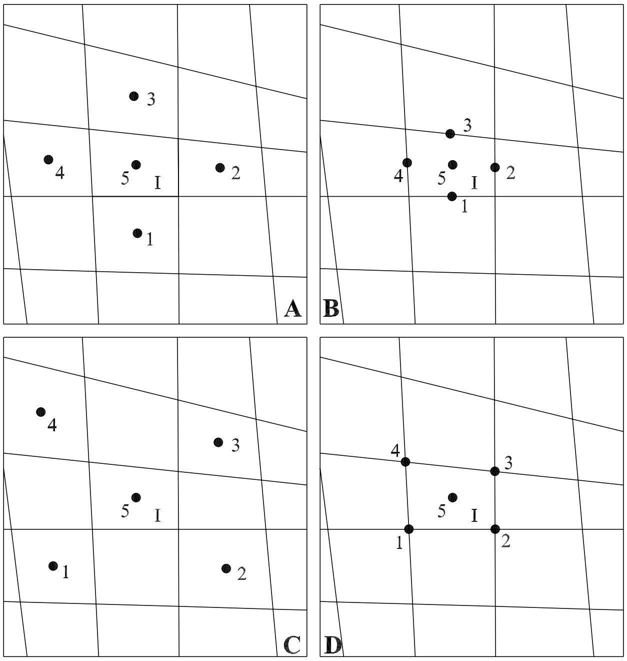


Fig. 16 Neighbourhoods for the limiting of the reconstruction inside cell I

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

$$g_i = \left(\frac{\partial^2 u_i}{\partial x^2}\right)^2 + 2\left(\frac{\partial^2 u_i}{\partial x\partial y}\right)^2 + \left(\frac{\partial^2 u_i}{\partial y^2}\right)^2, \quad i=1,\ldots,N. \tag{119}$$

Some existing limiters could be considered to be included in this family. Van Rosendale [95] has proposed an extension to three gradients of Van Albada's limiter [92]. This limiter was used on unstructured triangular grids and its general structure is that of (117) with $N=3$. The representative gradients are evaluated at the three *vertices* of the cell. Jawahar and Kamath [59] 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 (118) are slightly different in this case.

For quadrilateral cells we propose a limiter based on (117)–(118) with $N=5$; i.e. the limited derivatives are obtained as a weighted average of five unlimited derivatives. Figure 16 presents four suitable configurations to determine such *representative derivatives*. In this study only the configuration given by 16-A will be considered. In the following, the above limiter will be referred to as "*PC5 limiter*".

6.6 Numerical Convective Fluxes

For exposition purposes, we present the numerical convective fluxes corresponding to the model problem (Navier-Stokes equations), which are obtained using Roe's flux difference splitting [81]. For this purpose, left ($\boldsymbol{u}^+$) and right ($\boldsymbol{u}^-$) states are defined on each face. The numerical flux is then computed as

$$\begin{aligned}\left(\boldsymbol{F}_x, \boldsymbol{F}_y\right) \cdot \boldsymbol{n} &= \frac{1}{2}\left[\left(\boldsymbol{F}_x\left(\boldsymbol{u}^+\right), \boldsymbol{F}_y\left(\boldsymbol{u}^+\right)\right) + \left(\boldsymbol{F}_x\left(\boldsymbol{u}^-\right), \boldsymbol{F}_y\left(\boldsymbol{u}^-\right)\right)\right] \cdot \boldsymbol{n} \\ &\quad - \frac{1}{2}\sum_{k=1}^{3} \widetilde{\alpha}_k |\widetilde{\lambda}_k| \widetilde{\boldsymbol{r}}_k, \end{aligned} \tag{120}$$

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

$$\begin{aligned}&\widetilde{\lambda}_1 = \widetilde{\boldsymbol{v}} \cdot \boldsymbol{n} - \widetilde{c}, \qquad \widetilde{\lambda}_2 = \widetilde{\lambda}_3 = \widetilde{\boldsymbol{v}} \cdot \boldsymbol{n}, \\ &\widetilde{\lambda}_4 = \widetilde{\boldsymbol{v}} \cdot \boldsymbol{n} + \widetilde{c},\end{aligned} \tag{121}$$

$$\begin{aligned}&(\widetilde{\boldsymbol{r}}_1\ \widetilde{\boldsymbol{r}}_2\ \widetilde{\boldsymbol{r}}_3\ \widetilde{\boldsymbol{r}}_4) \\ &= \begin{pmatrix} 1 & 0 & 1 & 0 \\ \widetilde{u} - \widetilde{c}n_x & -\widetilde{c}n_y & \widetilde{u} & \widetilde{u} + \widetilde{c}n_x \\ \widetilde{v} - \widetilde{c}n_y & \widetilde{c}n_x & \widetilde{v} & \widetilde{v} + \widetilde{c}n_y \\ \widetilde{H} - \widetilde{c}\,\widetilde{\boldsymbol{v}} \cdot \boldsymbol{n} & \widetilde{c}(\widetilde{v}n_x - \widetilde{u}n_y) & \frac{1}{2}(\widetilde{u}^2 + \widetilde{v}^2) & \widetilde{H} + \widetilde{c}\,\widetilde{\boldsymbol{v}} \cdot \boldsymbol{n} \end{pmatrix}\end{aligned} \tag{122}$$

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

$$\begin{aligned}\widetilde{\alpha}_1 &= \frac{1}{2\widetilde{c}^2}\left[\Delta(p) - \widetilde{\rho}\widetilde{c}\left(\Delta(u)\, n_x + \Delta(v)\, n_y\right)\right], \\ \widetilde{\alpha}_2 &= \frac{\widetilde{\rho}}{\widetilde{c}}\left[\Delta(v)\, n_x - \Delta(u)\, n_y\right], \\ \widetilde{\alpha}_3 &= -\frac{1}{\widetilde{c}^2}\left[\Delta(p) - \widetilde{c}^2\Delta(\rho)\right], \\ \widetilde{\alpha}_4 &= \frac{1}{2\widetilde{c}^2}\left[\Delta(p) + \widetilde{\rho}\widetilde{c}\left(\Delta(u)\, n_x + \Delta(v)\, n_y\right)\right],\end{aligned} \tag{123}$$

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 $\widetilde{\boldsymbol{v}} = (\widetilde{u}, \widetilde{v})$ and $\widetilde{H}$ (computed using $\boldsymbol{U}^+$ and $\boldsymbol{U}^-$) are defined as

$$\begin{aligned}&\widetilde{u} = \frac{u^+\sqrt{\rho^+} + u^-\sqrt{\rho^-}}{\sqrt{\rho^+} + \sqrt{\rho^-}}, \qquad \widetilde{v} = \frac{v^+\sqrt{\rho^+} + v^-\sqrt{\rho^-}}{\sqrt{\rho^+} + \sqrt{\rho^-}}, \\ &\widetilde{H} = \frac{H^+\sqrt{\rho^+} + H^-\sqrt{\rho^-}}{\sqrt{\rho^+} + \sqrt{\rho^-}}.\end{aligned} \tag{124}$$

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

$$\widetilde{\rho} = \sqrt{\rho^+\rho^-}, \qquad \widetilde{c}^2 = (\gamma - 1)\left[\widetilde{H} - \frac{1}{2}\left(\widetilde{u}^2 + \widetilde{v}^2\right)\right]. \tag{125}$$

6.7 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 auxiliary 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 $\boldsymbol{v}$, temperature T, and their corresponding gradients, $\boldsymbol{\nabla v}$ and $\boldsymbol{\nabla} T$, at each quadrature point $\boldsymbol{x}_{iq}$. Using MLS approximation, these entities are readily computed as

$$\boldsymbol{v}_{iq} = \sum_{j=1}^{n_{iq}} \boldsymbol{v}_j N_j(\boldsymbol{x}_{iq}), \qquad T_{iq} = \sum_{j=1}^{n_{iq}} T_j N_j(\boldsymbol{x}_{iq}) \tag{126}$$

and

$$\boldsymbol{\nabla v}_{iq} = \sum_{j=1}^{n_{iq}} \boldsymbol{v}_j \otimes \boldsymbol{\nabla} N_j(\boldsymbol{x}_{iq}), \qquad \boldsymbol{\nabla} T_{iq} = \sum_{j=1}^{n_{iq}} T_j \boldsymbol{\nabla} N_j(\boldsymbol{x}_{iq}), \tag{127}$$

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).

6.8 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.

6.9 Time Integration

6.9.1 Explicit Schemes

For explicit time integration we use the third order TVD-Runge-Kutta algorithm proposed by Shu and Osher [85]. Given the field variables $\boldsymbol{u}^n$ at the previous time step n, the algorithm proceeds in three stages to obtain the updated field variables $\boldsymbol{u}^{n+1}$, as

$$\begin{aligned}\boldsymbol{u}^1 &= U^n + \Delta t L(\boldsymbol{u}^n),\\ \boldsymbol{u}^2 &= \frac{3}{4}\boldsymbol{u}^n + \frac{1}{4}\boldsymbol{u}^1 + \frac{1}{4}\Delta t L(\boldsymbol{u}^1),\\ \boldsymbol{u}^{n+1} &= \frac{1}{3}\boldsymbol{u}^n + \frac{2}{3}\boldsymbol{u}^2 + \frac{2}{3}\Delta t L(\boldsymbol{u}^2),\end{aligned} \tag{128}$$

where the operator $L(\cdot)$, which represents the spatial discretization operator, reads

$$L(\boldsymbol{u}) = \frac{1}{A}\sum_{iedge=1}^{nedge}\sum_{igau=1}^{ngau}\left[\left(\mathcal{F}^V - \mathcal{F}\right)\cdot\boldsymbol{n}\right]_{igau}\mathcal{W}_{igau}. \tag{129}$$

6.9.2 Implicit Schemes

Implicit time integration can be used either as a time marching approach for unsteady problems, or as a relaxation technique for steady state computations. For most implicit schemes, in addition to the residual $\boldsymbol{R}$, we need to evaluate the Jacobian $\partial\boldsymbol{R}/\partial\boldsymbol{u}$. The approximation framework in terms of shape functions allows a clear formulation of the Jacobian in the proposed methodology.

For presentation purposes, we will analyze the construction of the Jacobian for a nonlinear parabolic problem, the Cahn-Hilliard equation [27]. Numerical results of the application of the proposed finite volume method to the numerical solution of this challenging problem will be presented in Sect. 8. To the authors' knowledge, this is the first practical finite volume scheme developed for this equation. The stiffness of this problem precludes the use of explicit time-stepping for practical applications, and requires the use of efficient adaptive implicit time-stepping procedures [37].

Consider a binary mixture and let the concentration of one of its constituents be denoted by c, $0 < c < 1$. The concentration of the other constituent is, therefore, $1 - c$. Transport of mass in the mixture is governed by the parabolic (Cahn-Hilliard) equation

$$\frac{\partial c}{\partial t} = \nabla\cdot\left(b(c)\,\nabla\left(-\gamma\,\Delta c + \Psi'(c)\right)\right) \tag{130}$$

supplemented with suitable initial and boundary conditions, where $b(c) \geq 0$ is the diffusion mobility, $\Psi(c)$ is the homogeneous free energy density, and γ is a positive constant. The Cahn-Hilliard equation was originally proposed by Cahn and Hilliard [27] to model spinodal decomposition and coarsening phenomena in binary alloys. In this study we focus on the case of degenerate mobility

$$b(c) = c(1-c) \tag{131}$$

and logarithmic potential,

$$\Psi(c) = A(c\log(c) + (1-c)\log(1-c)) + Bc(1-c). \tag{132}$$

The finite volume formulation of (130) reads, for each control volume [37]

$$\int_{\Omega_I}\frac{\partial c}{\partial t}\,d\Omega - \int_{\Gamma_I} b(c)\,\nabla\left(-\gamma\,\Delta c + \Psi'(c)\right)\cdot\boldsymbol{n}\,d\Gamma = 0. \tag{133}$$

The challenge for any finite volume scheme is how to compute the flux

$$\boldsymbol{f}\cdot\boldsymbol{n} = b(c)\,\nabla\left(-\gamma\,\Delta c + \Psi'(c)\right)\cdot\boldsymbol{n}. \tag{134}$$

Considering a constant value of γ, the above expression can be written as

$$\boldsymbol{f}\cdot\boldsymbol{n} = b(c)\begin{pmatrix} -\gamma(\frac{\partial^3 c}{\partial x^3} + \frac{\partial^3 c}{\partial y^2 x}) + \frac{\partial\Psi'(c)}{\partial x}\\ -\gamma(\frac{\partial^3 c}{\partial x^2 y} + \frac{\partial^3 c}{\partial y^3}) + \frac{\partial\Psi'(c)}{\partial y}\end{pmatrix}\cdot\boldsymbol{n}. \tag{135}$$

Following the proposed MLS-based finite volume scheme, the above fluxes can be directly, and uniquely, evaluated at each interface quadrature point. This is accomplished by setting mobilities

$$b(c)\big|_{ig} = \sum_{j=1}^{n} N_j(\boldsymbol{x}_{ig})b(c_j) \tag{136}$$

whereas the third derivatives of the concentrations are computed as

$$\begin{aligned}&\left.\frac{\partial^3 c}{\partial x^a y^b}\right|_{ig} = \sum_{j=1}^{n}\left.\frac{\partial^3 N_j}{\partial x^a y^b}\right|_{ig} c_j,\\ &a + b = 3,\ a \geq 0,\ b \geq 0,\end{aligned} \tag{137}$$

and the free energy terms

$$\begin{aligned}\left.\frac{\partial\Psi'(c)}{\partial x}\right|_{ig} &= \sum_{j=1}^{n}\left.\frac{\partial N_j}{\partial x}\right|_{ig}\Psi'(c_j),\\ \left.\frac{\partial\Psi'(c)}{\partial y}\right|_{ig} &= \sum_{j=1}^{n}\left.\frac{\partial N_j}{\partial y}\right|_{ig}\Psi'(c_j).\end{aligned} \tag{138}$$

The $O(\Delta x^4)$ time step stability limit precludes the use of explicit time-stepping for practical purposes. The use of MLS approximants, with its shape function structure, simplifies the linearization of the fluxes (135). It follows from the above definitions that the contribution to the Jacobian due to each stencil node j of a certain edge quadrature point ig corresponding to a cell I is given by

$$\begin{aligned}J_{Ij}^{ig} &= -\frac{\partial\boldsymbol{f}^{ig}}{\partial c_j}\omega_{ig},\\ \boldsymbol{n} &= -b(c)\big|_{ig}\begin{pmatrix} -\gamma(\frac{\partial^3 N_j}{\partial x^3} + \frac{\partial^3 N_j}{\partial y^2 x}) + \frac{\partial N_j}{\partial x}\Psi''(c_j)\\ -\gamma(\frac{\partial^3 N_j}{\partial x^2 y} + \frac{\partial^3 N_j}{\partial y^3}) + \frac{\partial N_j}{\partial y}\Psi''(c_j)\end{pmatrix}\omega_{ig}\cdot\boldsymbol{n}\end{aligned}$$

Table 1 Convergence results for Ringleb flow

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

$$- N_j b'(c_j) \begin{pmatrix} -\gamma(\frac{\partial^3 c}{\partial x^3} + \frac{\partial^3 c}{\partial y^2 x}) + \frac{\partial \Psi'(c)}{\partial x} \\ -\gamma(\frac{\partial^3 c}{\partial x^2 y} + \frac{\partial^3 c}{\partial y^3}) + \frac{\partial \Psi'(c)}{\partial y} \end{pmatrix} \omega_{ig} \cdot \boldsymbol{n}, \tag{139}$$

where ω_{ig} is the quadrature weight associated to the quadrature point ig.

7 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 solution of hyperbolic problems, and high-order solution of elliptic problems.

7.1 Hyperbolic Problems: Ringleb Flow

Ringleb flow is an exact solution of the Euler equations, obtained by means of the hodograph method [82]. 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 showed in Fig. 17 (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. 17 (bottom right) presents a comparison of the different reconstructions with respect to accuracy versus 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 therefore more flux evaluations, and not to the higher order reconstruction itself.

7.2 Elliptic Problems: Comparison with Discontinuous Galerkin Methods

In this section the accuracy test in [78] is reproduced. It will serve as a comparison between the proposed scheme and three DG discretizations for elliptic problems, namely the Local Discontinuous Galerkin (LDG) method [34], an improved variant of LDG, the so-called Compact Discontinuous Galerkin (CDG) method [78], and the second scheme of Bassi and Rebay (BR2) [24].

The two-dimensional model problem is

$$\begin{aligned} &\nabla \cdot (-\nabla u) = f \quad \text{in } \Omega, \\ &u = u_D \quad \text{on } \Gamma \end{aligned} \tag{140}$$

where Ω is the unit square, $[0, 1] \times [0, 1]$, and Γ its boundary. We use Dirichlet boundary conditions and a source term such that the solution of (140) is

$$u(x, y) = \exp[\alpha \sin(ax + by) + \beta \cos(cx + dy)], \tag{141}$$

where $\alpha = 0.1$, $\beta = 0.3$, $a = 5.1$, $b = -6.2$, $c = 4.3$ and $d = 3.4$. The Dirichlet boundary conditions are enforced weakly by the use of ghost boundary points which have associated the exact boundary value.

The results in [78] were computed using a series of successively refined meshes, obtained by splitting a regular $n \times n$ Cartesian grid into $2n^2$ triangles, giving uniform element sizes of $h = 1/n$. They consider nodal basis of degree p, with the nodes uniformly distributed, and therefore the spatial resolution is h/p.

In order to compare these DG results with those of the proposed FV scheme we need to establish a way of generating the FV grids whose solutions are to be compared to the

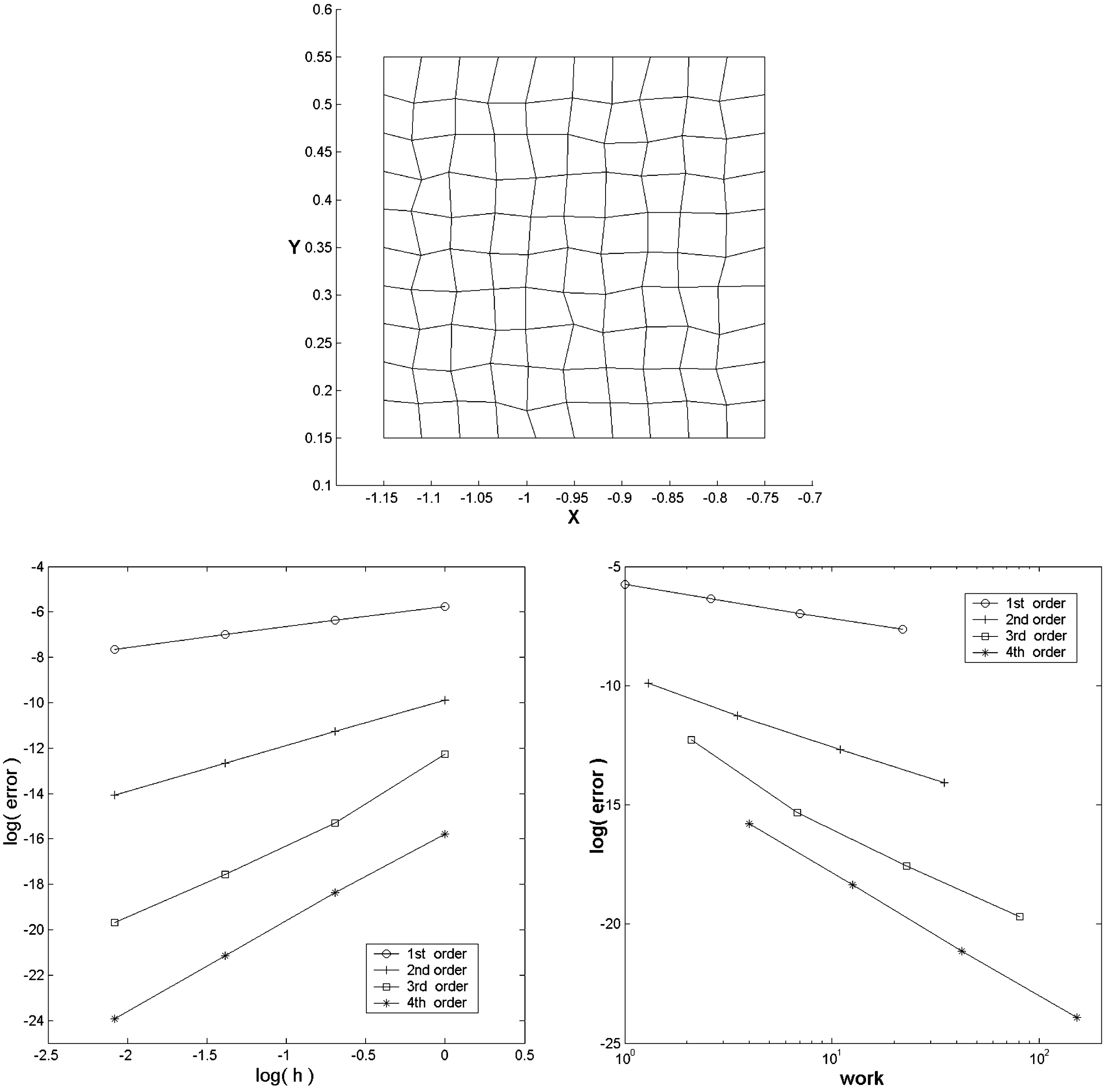


Fig. 17 Coarse grid level and convergence results for Ringleb flow

reported ones. In this study we make a comparison in terms of accuracy for the *same grid resolution*. Following this criterion, the results reported in [78] on a grid level n, with $2n^2$ p-elements, are to be compared with FV results obtained on a Cartesian grid of $(pn) \times (pn)$ quadrilaterals. We will restrict our analysis to the case $p = 3$, which is the FV scheme we are interested in for practical applications.

Note that, however, for the same grid resolution and $p = 3$, the DG triangular grids comprise a total of $20n^2$ primal degrees of freedom, whereas the total number of degrees of freedom for the FV scheme is $9n^2$, thus a factor of 2.2. Furthermore, this analysis excludes the auxiliary degrees of freedom required in DG which, for this simple problem, can be written in terms of the primal unknowns. This "elimination" of the auxiliary variables involves, nonetheless, a non-negligible cost. We prefer this comparison in terms of *accuracy for the same grid resolution*, rather than comparisons of accuracy versus number of degrees of freedom or accuracy versus cost, which, being more fair for the FV scheme, do not solve the question about the relative spatial accuracy of either scheme. This study aims at shedding some light on the question about whether DG is actually more accurate

Table 2 2D Poisson, $p = 3$: L_2 errors and convergence rates in primal variable. Comparison between the proposed FV scheme and three DG schemes, namely: CDG, LDG and BR2

Scheme	$n = 4$ (12×12)	$n = 8$ (24×24)	$n = 16$ (48×48)	$n = 32$ (96×96)	Rate
FV	2.02×10^{-4}	1.40×10^{-5}	9.07×10^{-7}	5.78×10^{-8}	4.0
CDG	2.44×10^{-4}	1.71×10^{-5}	1.10×10^{-6}	7.03×10^{-8}	4.0
LDG	3.81×10^{-4}	2.04×10^{-5}	1.18×10^{-6}	7.23×10^{-8}	4.0
BR2	3.77×10^{-4}	2.47×10^{-5}	1.52×10^{-6}	9.46×10^{-8}	4.0

Table 3 2D Poisson, $p = 3$: L_2 errors and convergence rates of the gradient of the primal variable, $p = 3$. Comparison between the proposed FV scheme and CDG

Scheme	$n = 4$ (12×12)	$n = 8$ (24×24)	$n = 16$ (48×48)	$n = 32$ (96×96)	Rate
FV	3.24×10^{-3}	2.78×10^{-4}	2.49×10^{-5}	2.15×10^{-6}	3.5
CDG	3.01×10^{-3}	3.63×10^{-4}	4.37×10^{-5}	5.36×10^{-6}	3.0

Table 4 Memory requirements and compactness measure for the proposed FV scheme, CDG, LDG and BR2

Scheme	Memory requirements per block	ndof per block (primal dof for DG)	Average entries per row	Ratio
FV	$9 \times 21 = 189$	9	21	1
CDG	$2 \times 220 = 440$	$2 \times 10 = 20$	22	1.05
LDG	$2 \times 236 = 472$	$2 \times 10 = 20$	23.6	1.12
BR2	$2 \times 292 = 584$	$2 \times 10 = 20$	29.2	1.39

and more compact than FV schemes, and the only way is to compare with the same grid resolution. The conclusions in terms of computational effort and efficiency will be straightforward, and are left to the reader.

In order to make the FV results reproducible, the stencils for the computation of the fluxes at the interfaces are defined as the 16 nearest centroids (the distances are computed with respect to the centerpoint of the edge). This definition coincides, for Cartesian grids and interior edges, with the one presented in Sect. 6. It is clearly sub-optimal for edges near boundaries, but easy to reproduce. We used radial kernels with the cubic spline (67), and smoothing lengths $h = 0.52 \max(|\boldsymbol{x}_j - \boldsymbol{x}_q|)$, where $\boldsymbol{x}_q$ is the quadrature (evaluation) point.

Table 2 presents a comparison of the L_2 error norm for the different grid sizes, the CDG results correspond to the *consistent switch*. All schemes achieve the correct fourth order of accuracy. The proposed FV scheme is slightly, but consistently, more accurate than the DG methods at all refinement levels, with differences varying from 20% with respect to the most accurate DG scheme (CDG), to almost a factor of 2 with respect to BR2.

Table 3 presents a comparison of the errors in the gradient (L_2 error of the gradient). Only the CDG results are reported in [78]. The differences are even more pronounced, due in part to the superconvergent properties of the proposed finite volume scheme.

It is also interesting to analyze the storage requirements of the different schemes. Following the above "same grid resolution" criterion, we consider an interior block of 2 DG p-elements, and the corresponding p^2 control volumes. Table 4 is based on the data reported in [78] for the $p = 3$ case, including the results for the proposed FV scheme. For each block of 9 cells, and with the 21-point stencil presented in Sect. 6, the finite volume scheme requires $9 \times 21 = 189$ entries in the coefficient matrix. With the same grid resolution (the same block), the CDG, LDG and BR2 schemes require 440, 472 and 584 entries, respectively. Of course, a significant part of the increased number of entries is due to the fact that, as mentioned above, the DG schemes involve 2.22 times more degrees of freedom (equations). A good measure to assess the *net* compactness of the different schemes is to compute the *average number of entries* in each row of the coefficient matrix. This value corresponds to dividing the number of block entries between the number of degrees of freedom per block (primal degrees of freedom in the case of DG). Even in this measure the proposed finite volume scheme outperforms the DG schemes, the differences varying between 5% in the case of CDG and 39% with respect to BR2. This results challenge the somewhat widespread ideas about the non-compactness of finite volume methods, as opposed to the compactness of DG.

The results of this study are clear: even for the same grid resolution, the proposed FV scheme is consistently more ac-

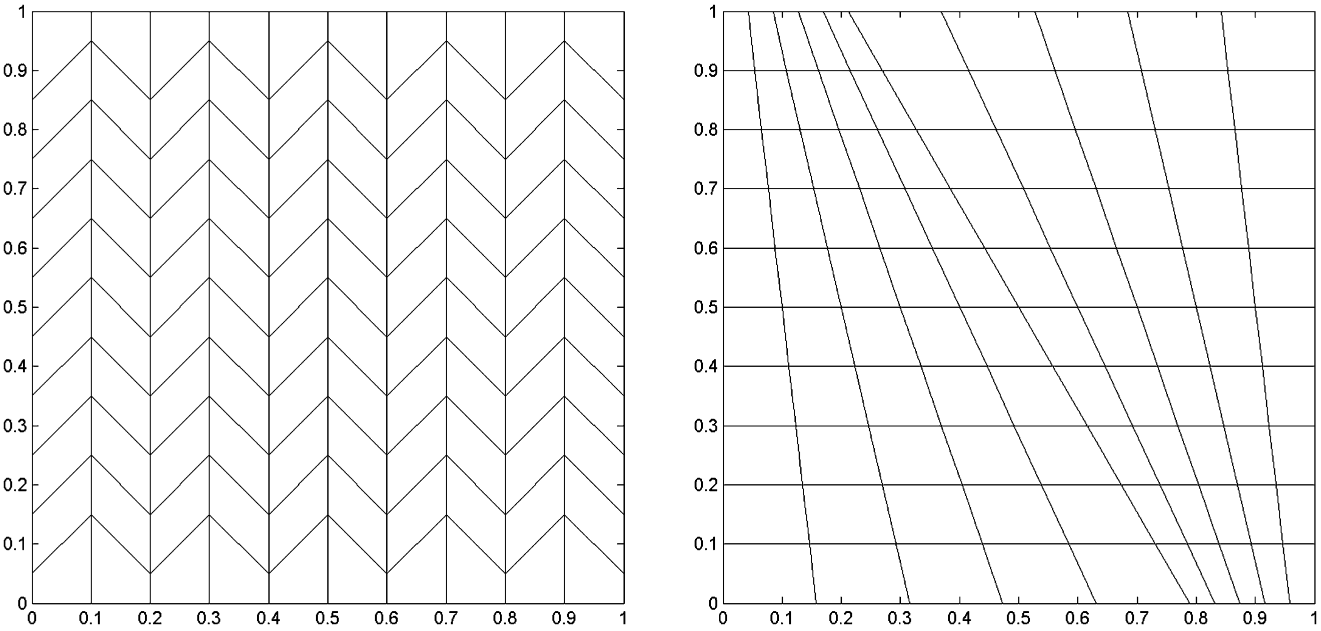


Fig. 18 Coarse grids for 2D Darcy flow: Chevron (*left*) and skewed (*right*)

curate than the analyzed DG schemes. Furthermore, it requires less than half the number of degrees of freedom (2.22 times less rows in the coefficient matrix), and the average stencil is smaller (average number of non-zero entries on each row of the coefficient matrix). The cost analysis should also include the fact that the DG discretizations require the introduction of auxiliary degrees of freedom, that need to be solved for in terms of the primal unknowns.

7.3 Further Analysis of the Superconvergence of the FV Scheme for Elliptic Problems: 2D Darcy Flow

In order to assess whether the superconvergence in the gradient of the previous section is due to the use of a Cartesian grid, or there is the possibility of a property of the scheme, consider the elliptic equation

$$\begin{aligned} &\nabla\cdot(-\boldsymbol{K}\nabla p)=f \quad \text{in } \Omega,\\ &p=p^D \quad \text{on } \Gamma^D, \\ &\boldsymbol{v}\cdot\boldsymbol{n}=h \quad \text{on } \Gamma^N, \end{aligned} \tag{142}$$

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 (142) 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 the analytical solution is $p=\sin(2\pi x)\sin(2\pi y)$. The exact Dirichlet boundary condition is enforced on Γ. A refinement study was carried out on a sequence of uniform, random, Chevron and skewed grids. The prototype Chevron and skewed grids are plotted in Fig. 18. The coarse random grid and its first level of refinement are plotted on Fig. 19. The convergence results are presented in Tables 5 and 6, and confirm that, also on irregular grids, the scheme is fourth order accurate in the primal variable (pressure) and superconvergent (order 3.5) in the gradient (velocity).

8 Representative Simulations

8.1 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 Sect. 3.4. The Shu-Osher test case [86] is solved using 400 cells in $[-5,+5]$, with initial conditions

$$\begin{aligned} &(\rho_R,u_R,p_R)=(3.857,2.629,10,333)\,,\\ &(\rho_L,u_L,p_L)=(1+0.2\sin(5x),0,1)\,. \end{aligned} \tag{143}$$

The solution is advanced until $t=1.8$, using linear, quadratic and cubic reconstructions, with the limiters being active everywhere (Fig. 20). 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 $h=0.55d_{max}$.

Even though the higher order schemes perform fairly better than the second-order one, it is clear that the limiters are

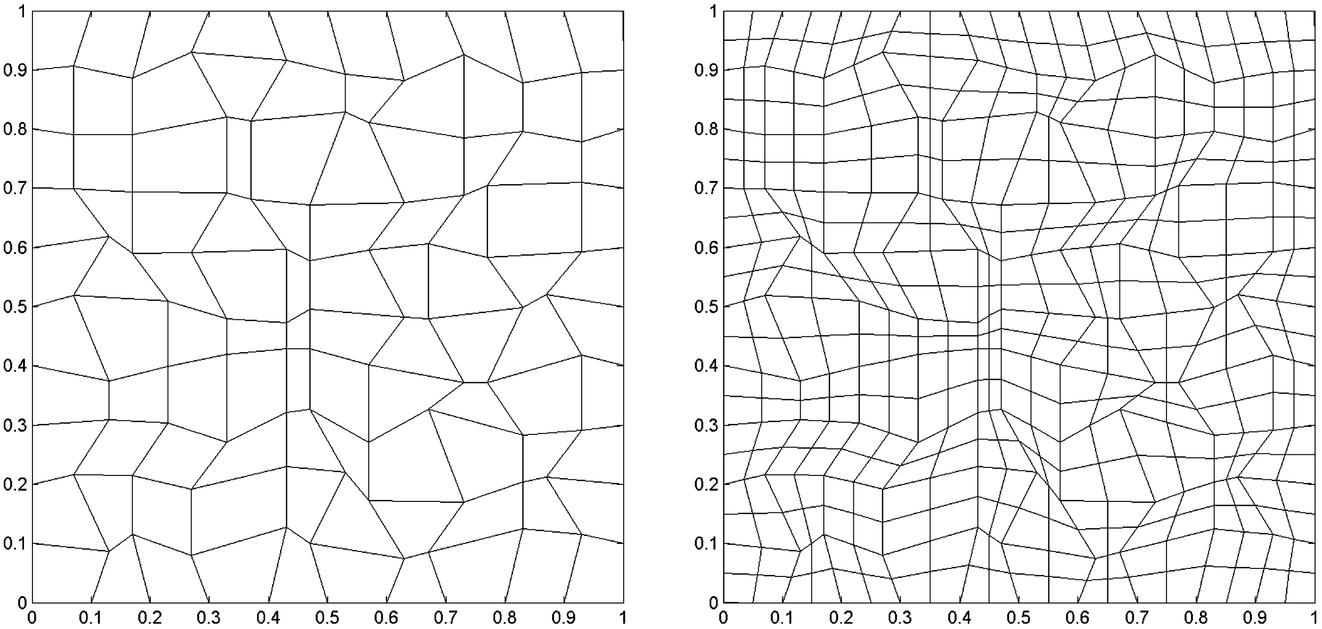


Fig. 19 Coarse random grid for 2D Darcy flow (*left*) and one level refinement (*right*)

Table 5 Darcy flow: Convergence results for the uniform and random grids ($p = 3$ 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}		3.97×10^{-2}		5.66×10^{-2}	
20 × 20	2.65×10^{-4}	3.91	4.91×10^{-4}	3.37	3.80×10^{-3}	3.39	7.84×10^{-3}	2.85
40 × 40	1.57×10^{-5}	4.08	4.01×10^{-5}	3.61	3.52×10^{-4}	3.43	1.00×10^{-3}	2.97
80 × 80	9.34×10^{-7}	4.07	2.78×10^{-6}	3.85	3.17×10^{-5}	3.47	9.68×10^{-5}	3.37

Table 6 Darcy flow: Convergence results for the Chevron and skewed grids ($p = 3$ MLS-FV scheme)

Grid	Pressure				Velocity			
	Chevron		Skewed		Chevron		Skewed	
	Error	Slope	Error	Slope	Error	Slope	Error	Slope
10 × 10	3.98×10^{-3}		6.33×10^{-3}		3.97×10^{-2}		6.33×10^{-2}	
20 × 20	2.65×10^{-4}	3.91	4.99×10^{-4}	3.67	3.80×10^{-3}	3.39	5.77×10^{-3}	3.46
40 × 40	1.56×10^{-5}	4.09	3.41×10^{-5}	3.87	3.52×10^{-4}	3.43	5.22×10^{-4}	3.47
80 × 80	9.34×10^{-7}	4.06	2.24×10^{-6}	3.93	3.17×10^{-5}	3.47	4.68×10^{-5}	3.48

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. Figure 21 presents 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.

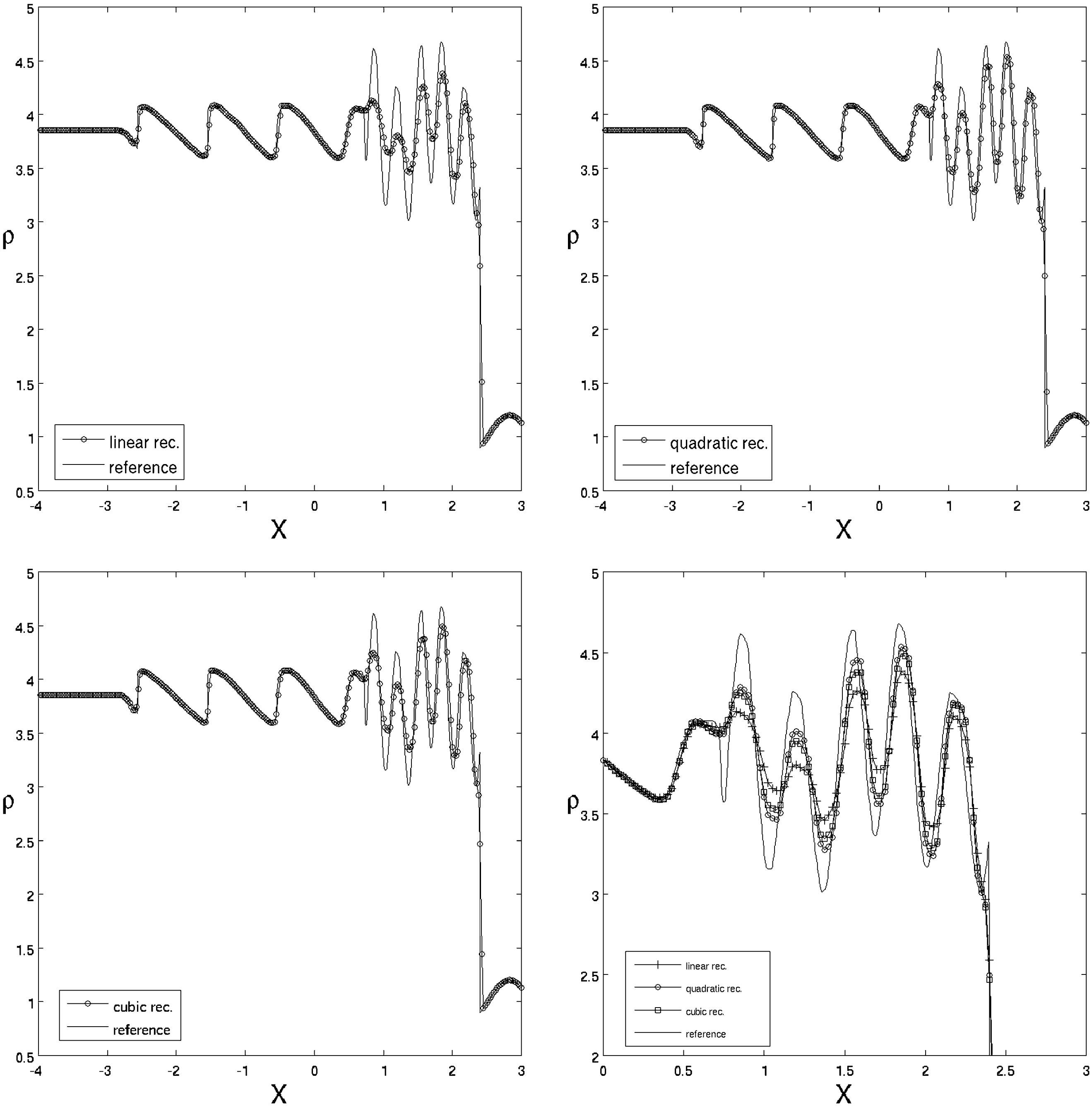


Fig. 20 Shu-Osher problem, 400 cells, limiters active everywhere. Density profiles computed with the second (*top left*), third (*top right*) and fourth (*bottom left*) order schemes. Detailed comparison (*bottom right*)

8.2 Inviscid Flow: Anisotropic Grids

In order to assess the ability of the MLS reconstruction to handle meshes with strong anisotropies, a compression corner problem is solve on a highly stretched grid. The free stream Mach number is $M = 3$. The problem is solved with cubic reconstruction and anisotropic kernels, as exposed in Sect. 5. The computational grid and pressure contours are plotted in Fig. 22. The maximum aspect ratio of the cells near the wall is about 2000.

8.3 Inviscid Flow: Airfoils

8.3.1 A Subsonic Test

The problem set up corresponds to a subsonic flow around a NACA 0012 airfoil. The freestream Mach number is

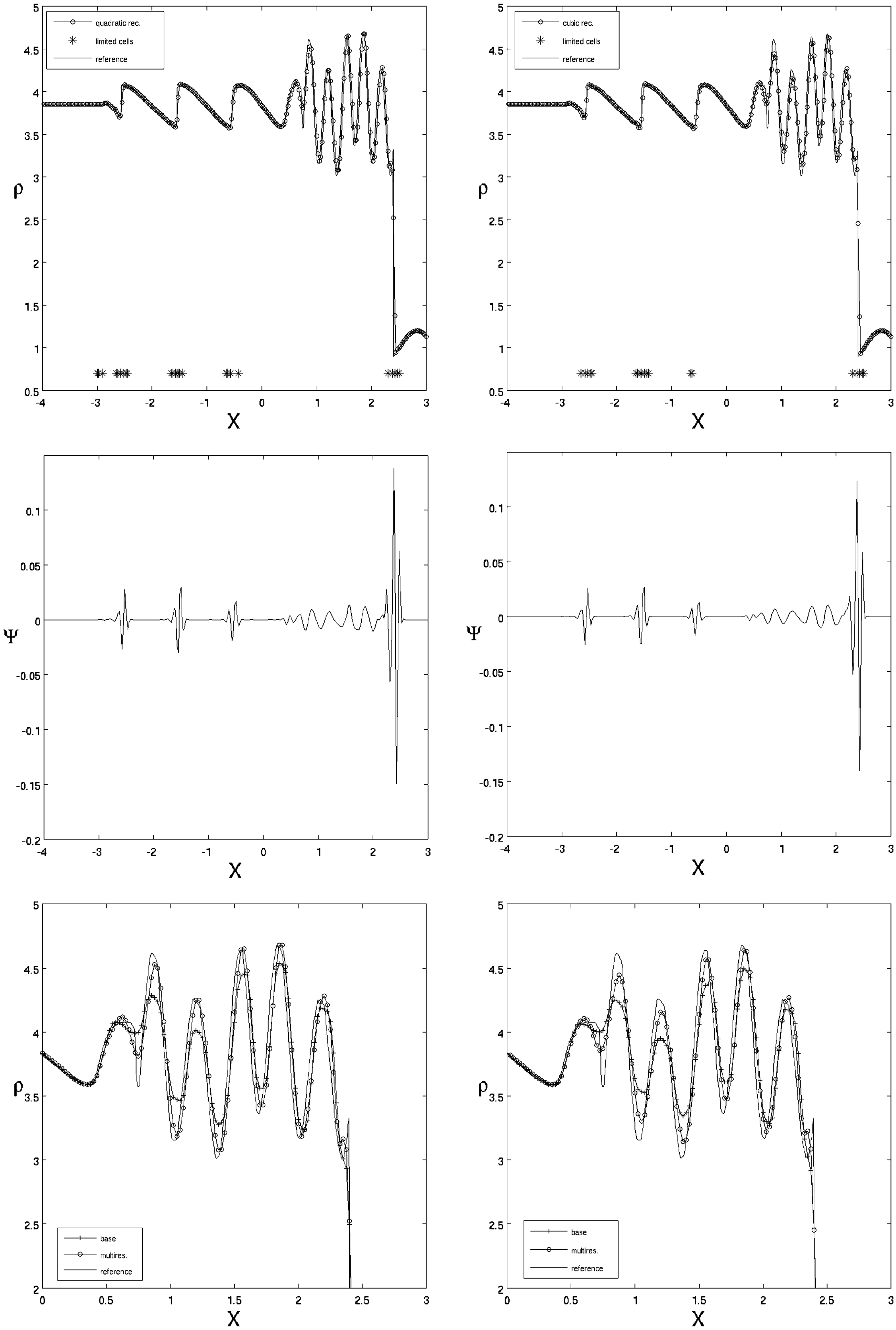


Fig. 21 Shu-Osher problem, 400 cells, multiresolution-based selective limiting. Density profiles computed with the third and fourth order schemes (*top left* and *top right*, respectively), high-scale components of the density (*center*) and detailed comparison of the densities with and without selective limiting (*bottom*)

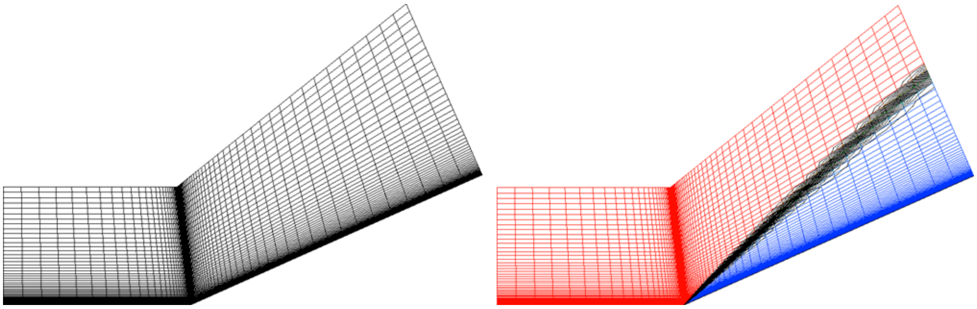


Fig. 22 Highly stretched grids: compression corner $M = 3$. Computational mesh (*left*) and pressure contours (*right*)

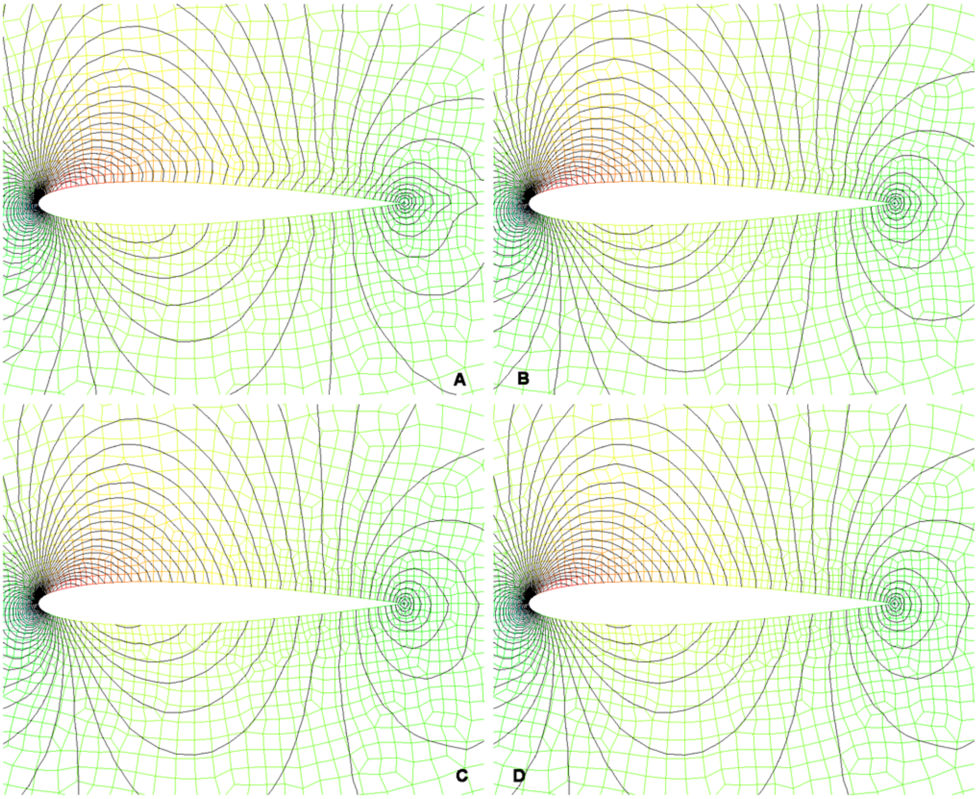


Fig. 23 Subsonic inviscid flow around a NACA 0012 airfoil ($M = 0.63, \alpha = 2.°$): close-up view of the Mach number contours obtained with linear (A), quadratic (B) and cubic (with 2 and 3 Gauss points per edge, C and D) reconstructions

$M = 0.63$ 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.

Figure 23 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 pseudoviscous 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

$$S = \ln\left(\frac{h^{\frac{\gamma}{\gamma-1}}}{p}\right), \quad h = \gamma\left(E - \frac{1}{2}(u^2 + v^2)\right). \qquad (144)$$

8.3.2 *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) $M = 0.85$, $\alpha = 1°$. Figures 24 and 25 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-

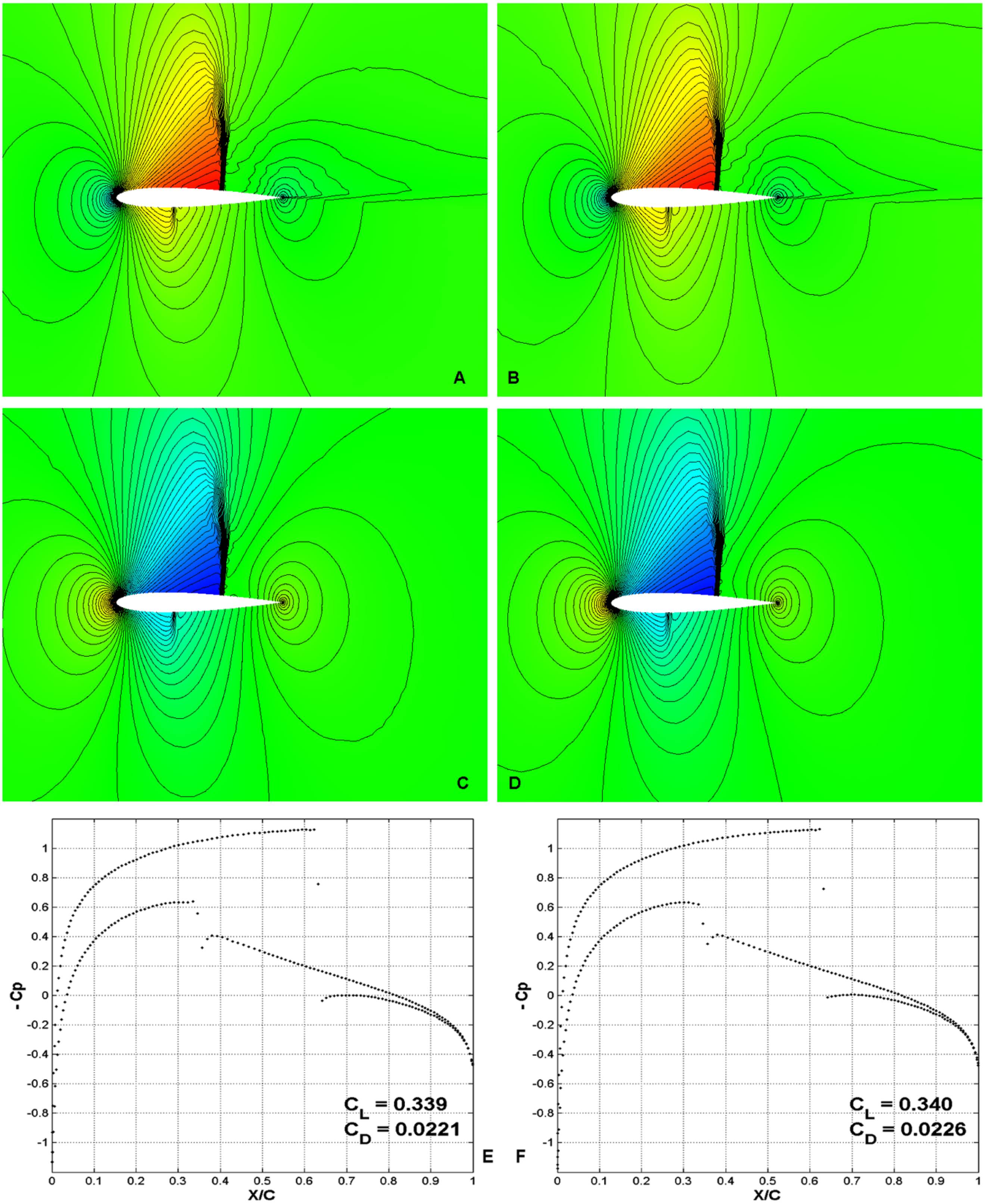


Fig. 24 Inviscid flow around a NACA 0012 airfoil ($M = 0.8$, $\alpha = 1.25°$): results obtained using *quadratic reconstruction* with either the BJ limiter (*A*–*C*–*E*) or the PC5 limiter (*B*–*D*–*F*). Mach number contours (*A*–*B*), pressure contours (*C*–*D*) and surface pressurecoefficients C_p (*E*–*F*)

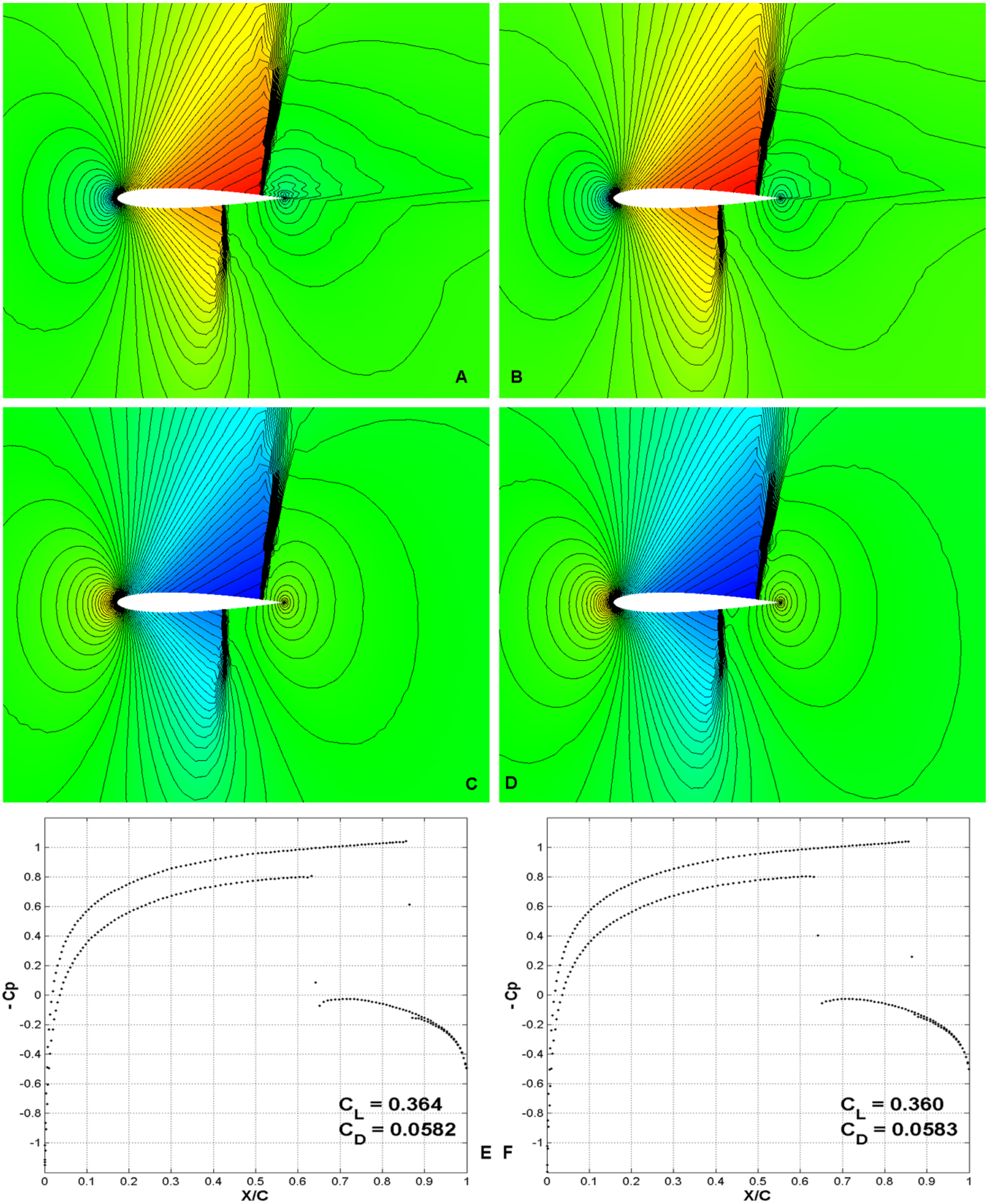


Fig. 25 Inviscid flow around a NACA 0012 airfoil ($M = 0.85$, $\alpha = 1°$): results obtained using *quadratic reconstruction* with either the BJ limiter (A–C–E) or the PC5 limiter (B–D–F). Mach number contours (A–B), pressure contours (C–D) and surface pressure coefficients C_p (E–F)

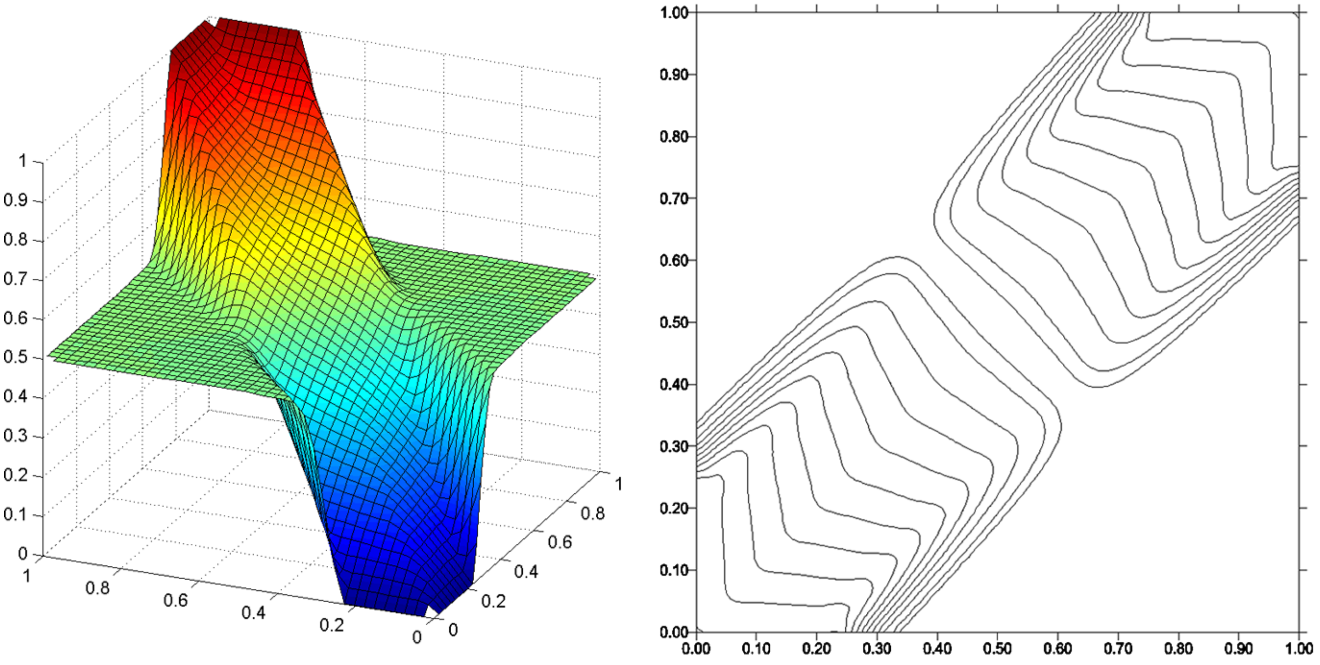


Fig. 26 2D Darcy flow with highly anisotropic, homogeneous permeability (40 × 40 grid)

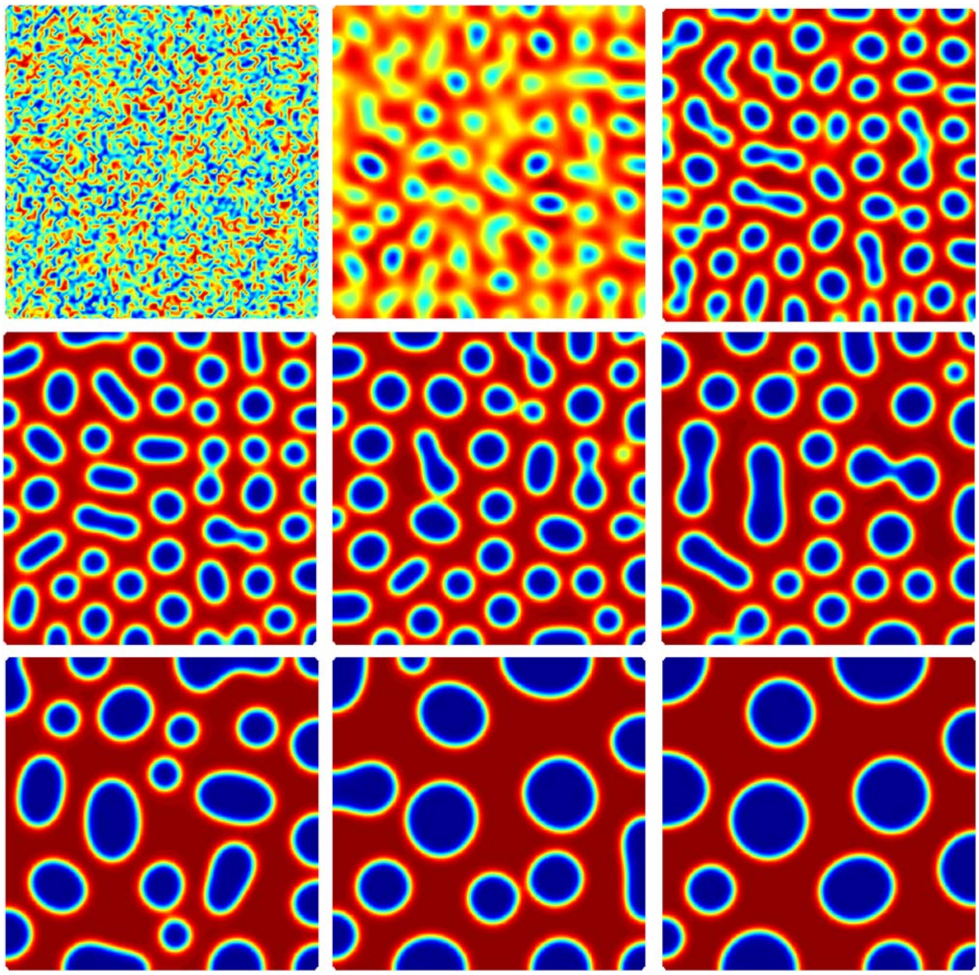


Fig. 27 2D Cahn-Hilliard equation. Snapshots of the solution at various time levels. *From left to right*, and *from top to bottom*: $t = 0$, $t = 2 \cdot 10^{-6}$, $t = 4 \cdot 10^{-6}$, $t = 8 \cdot 10^{-6}$, $t = 1.6 \cdot 10^{-5}$, $t = 3.2 \cdot 10^{-5}$, $t = 6.4 \cdot 10^{-5}$, $t = 1.28 \cdot 10^{-4}$, $t = 2.56 \cdot 10^{-4}$

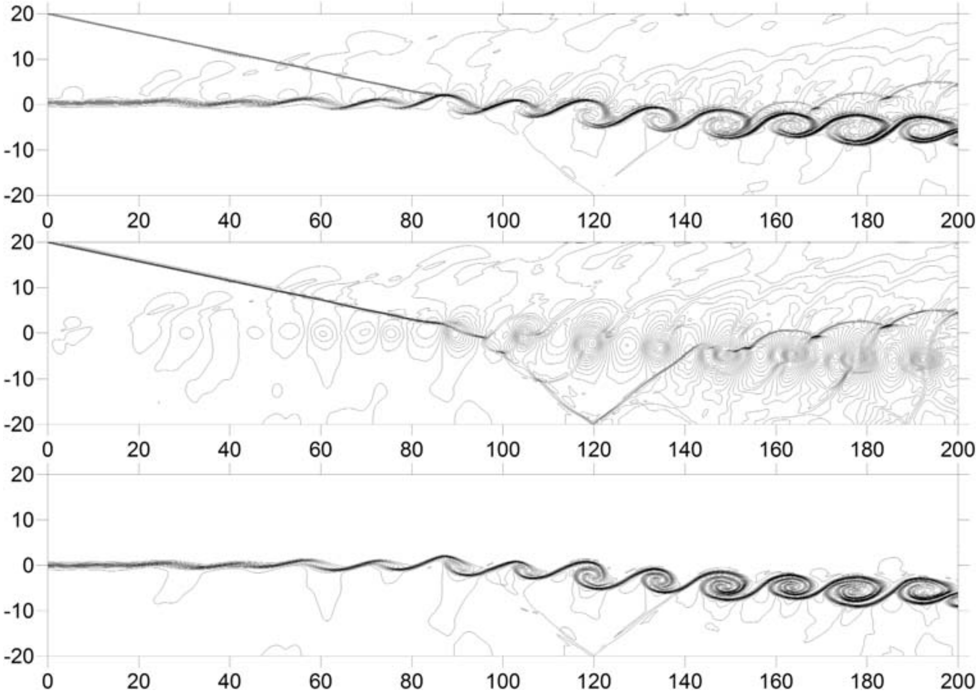


Fig. 28 Shock wave impingement on a mixing layer at $t = 120$. Fourth order results on the 600×300 grid. Contours of density (*top*), pressure (*center*) and temperature (*bottom*)

capturing (one interior cell) and clear slip lines, although the PC5 limiter appears to be slightly more dissipative.

8.4 Complex Elliptic/Parabolic Problems

8.4.1 Darcy Flow with Highly Anisotropic Permeability

Consider again the Darcy flow model problem (142). Practical reservoir simulations pose a great challenge to any discretization method due to the fact that the coefficient tensor $\boldsymbol{K}$ is in general highly variable and anisotropic. In particular, most existing schemes produce non-physical oscillations in the presence of strongly anisotropic fields, which cannot be eliminated by (reasonable) grid refinement. The search for schemes with improved monotonicity properties constitutes a very active area of research in the oil industry [1, 29, 44, 63, 64]. Given the nature of the equations, the authors believed that the anomalous behaviour of many existing schemes in the presence of extreme anisotropy ratios was due, at least in part, to their lack of accuracy in the computation of the interface fluxes (computation of gradients). This section presents and example of the performance of the proposed methodology for this kind of schemes, and somewhat confirms the authors' beliefs.

The problem is solved in $[0, +1] \times [0, +1]$; the permeability tensor $\boldsymbol{K}$ is such that its eigenvalues are $a = 10^6$ and $b = 1$, and the first eigenvalue is rotated and angle $\alpha = \pi/4$, i.e.

$$\boldsymbol{K} = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \begin{pmatrix} a & 0 \\ 0 & b \end{pmatrix} \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix}. \tag{145}$$

We impose Dirichlet boundary conditions in a certain area around the corners (0, 0 and (1, 1), more precisely $p(0, y) = 0$ for $y \leq 0.25$, $p(x, 0) = 0$ for $x \leq 0.25$, $p(1, y) = 1$ for $y \geq 0.75$ and $p(x, 1) = 1$ for $x \geq 0.75$. Otherwise, Neumann boundary conditions ($\boldsymbol{v} \cdot \boldsymbol{n} = 0$) are specified.

The results obtained in a grid of 40×40 cells are plotted in Fig. 26. The solution is monotone and the sharp layers are correctly captured by the scheme. We believe that the correct behaviour of the method is due to the high quality of the reconstructed gradients. This example also suggests that, in some cases, we may be able, and should try to, tackle our numerical problems by improving the accuracy of the scheme, rather than by decreasing it (adding dissipation). In other words, *resolving, rather than "stabilizing"*.

8.4.2 Cahn-Hilliard Equation

The model problem (130) presented in Sect. 6 is solved using the proposed finite volume method [37]. The Cahn-Hilliard equations have been solved using finite elements [18, 19, 45–47, 61, 62], finite differences [50, 91] and, more recently, discontinuous Galerkin methods [30, 104, 106]. We solve the second example in [104]. The simulation is performed in $[-0.5, +0.5] \times [-0.5, +0.5]$, with a uniform grid of 80×80 control volumes. The logarithmic potential (132) was used, with constants $A = 3000$, $B = 9000$,

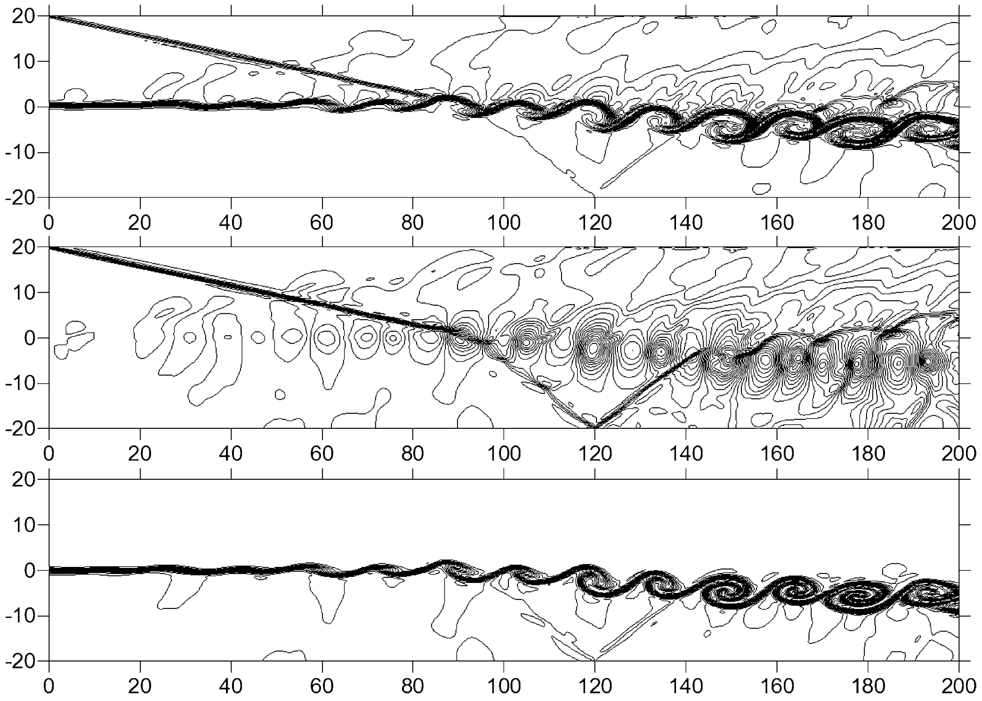


Fig. 29 Shock wave impingement on a mixing layer at $t = 120$. Fourth order results on the 400 × 100 grid. Contours of density (*top*), pressure (*center*) and temperature (*bottom*)

and $\gamma = 1$. The initial conditions are given by an average concentration $c = 0.63$, with random perturbations between -0.05 and $+0.05$. We use implicit time integration, with a simple backward Euler scheme. The concentrations at different time levels are depicted in Fig. 27. The result agree qualitatively with those reported in [104, 106].

8.5 Navier-Stokes Equations

8.5.1 Shock Wave Impingement on a Spatially Evolving Mixing Layer

We reproduce the example presented in [108]. 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 $0 \le x \le 200$ and $-20 \le y \le 20$, with inflow velocities specified as a hyperbolic tangent profile

$$u = 2.5 + 0.5 \tanh(2y). \tag{146}$$

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

$$v' = \sum_{k=1}^{2} a_k \cos\left(\frac{2\pi k t}{T} + \phi_k\right) e^{\left(\frac{-y^2}{b}\right)}, \tag{147}$$

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:

$$p_R = \frac{(\rho_1 + \rho_2)(u_1 - u_2)^2}{2}. \tag{148}$$

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

$$c^2 = {c_1}^2 + \frac{(\gamma - 1)}{2}\left({u_1}^2 - {u_2}^2\right). \tag{149}$$

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

$$p_0 = 0.3327, \qquad H_0 = 5.211, \qquad \mu_0 = 5 \times 10^{-4} \tag{150}$$

whereas on the upper boundary we set

$$\begin{aligned} u &= 2.9709, & v &= -0.1367, \\ \rho &= 2.1101, & p &= 0.4754. \end{aligned} \tag{151}$$

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. Figures 28 and

29 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.

9 Concluding Remarks

This paper explored 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.

The proposed selective limiting procedure, based on the multiresolution properties of the MLS approximants, allows to switch off the limiters on smooth regions of the flow, thus reducing the excessive dissipation associated to the use of limiters everywhere in the domain.

Accuracy tests show that the proposed method achieves the expected convergence rates and is a competitive alternative to other existing schemes. Representative simulations show that the methodology is applicable to problems of engineering interest.

Acknowledgements The research of LCF is supported by the Spanish Ministry of Science and Education (MEC), through its program of postdoctoral scholarships. 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" is also gratefully acknowledged.

This work has been partially funded by the SGPICT of the Spanish Ministry of Science and Education (MEC) (#DPI2004-05156 and #DPI2007-61214) co-financed with FEDER funds, as well as by the "Secretaría Xeral de I+D" of the "Xunta de Galicia" (# PGIDIT05PXIC118002PN and # PGDIT06TAM11801PR) and by the "Fundación de la Ingeniería Civil de Galicia" and the University of La Coruña, through research grants and fellowships.

References

1. Aavatsmark I (2002) An introduction to multipoint flux approximations for quadrilateral grids. Comput Geosci 6:405–432
2. Abgrall R (1994) On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. J Comput Phys 114:45–58
3. Abgrall R (1994) An essentially non-oscillatory reconstruction procedure on finite-element type meshes: application to compressible flows. Comput Methods Appl Mech Eng 116:95–101
4. Abgrall R (2001) Toward the ultimate conservative scheme: following the quest. J Comput Phys 167:277–315
5. Abgrall R, Sonar T (1997) On the use of Mülbach expansions in the recovery step of ENO methods. Numer Math 76:1–25
6. Abgrall R, Barth TJ (2002) Weighted residual distribution schemes for conservation laws via adaptive quadrature. SIAM J Sci Comput 24:732–769
7. Abgrall R, Mezine M (2003) Construction of second order accurate monotone and stable residual distribution schemes for unsteady flow problems. J Comput Phys 188:16–55
8. Abgrall R, Roe PL (2003) Construction of very high order fluctuation schemes. J Sci Comput 19:3–36
9. Abgrall R, Roe PL (2003) High order fluctuation schemes on triangular meshes. J Sci Comput 19:3–36
10. Abgrall R (2006) Essentially non-oscillatory Residual Distribution schemes for hyperbolic problems. J Comput Phys 214:773–808
11. Abgrall R, Mezine M (2004) Construction of second order accurate monotone and stable residual distribution schemes for steady problems. J Comput Phys 195:474–507
12. Abgrall R, Adrianov N, Mezine M (2005) Towards very high-order accurate schemes for unsteady convection problems on unstructured meshes. Int J Numer Methods Fluids 47:679–691
13. Abgrall R, Marpeau F (2007) Residual distribution schemes on quadrilateral meshes. J Sci Comput 30:131–175
14. Aboiyar T, Georgoulis EH, Iske A (2006) High order WENO finite volume schemes using polyharmonic spline reconstruction. In: International conference on numerical analysis and approximation theory, NAAT2006, pp 1–14
15. Arnold DN (1982) An interior penalty finite element method with discontinuous elements. SIAM J Numer Anal 19:742–760
16. Arnold DN, Brezzi F, Cockburn B, Marini LD (1997) A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. J Comput Phys 131:267–279
17. Arnold DN, Brezzi F, Cockburn B, Marini LD (2001) Unified analysis of discontinuous Galerkin methods for elliptic problems. SIAM J Numer Anal 39:1749–1779
18. Barret JW, Blowey JF (1997) Finite element approximation of a model for phase separation of a multi-component alloy with non-smooth free energy. Numer Math 77:1–34
19. Barret JW, Blowey JF, Garcke H (1999) Finite element approximation of the Cahn-Hilliard equation with degenerate mobility. SIAM J Numer Anal 37:286–318
20. Barth TJ (1993) Recent developments in high-order k-exact reconstruction on unstructured meshes. AIAA paper 93-0668
21. Barth TJ (1995) Aspects of unstructured grids and finite-volume solvers for the Euler and Navier-Stokes equations. VKI lecture series 1994-05
22. Barth TJ, Frederickson PO (1990) Higher-order solution of the Euler equations on unstructured grids using quadratic reconstruction. AIAA paper 90-0013
23. Barth TJ, Jespersen DC (1989) The design and application of upwind schemes on unstructured meshes. AIAA-89-0366
24. Bassi F, Rebay S (1997) A higher-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. J Comput Phys 131:267–279
25. Bassi F, Rebay S (1997) A higher-order accurate discontinuous finite element solution of the 2D Euler equations. J Comput Phys 138:251–285

26. Brezzi F, Manzini G, Marini LD, Pietra P, Russo A (2000) Discontinuous Galerkin approximations for elliptic problems. Numer Methods Partial Differ Equ 16:365–378
27. Cahn JW, Hilliard JE (1958) Free energy of non-uniform systems. I. Interfacial free energy. J Chem Phys 28:258–267. 1–34
28. Castillo P (2002) Performance of discontinuous Galerkin methods for elliptic PDEs. SIAM J Sci Comput 24:524–547
29. Chen Q-Y, Wan J, Yang Y, Mifflin RT (2007) Enriched multipoint flux approximation for general grids. J Comput Phys 227:1701–1721
30. Choo SM, Lee YJ (2005) A discontinuous Galerkin method for the Cahn-Hilliard equation. J Appl Math Comput Phys 18:113–126
31. Cockburn B, Lin SY, Shu C-W (1989) TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems. J Comput Phys 84:90–113
32. Cockburn B, Shu C-W (1989) TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. Math Comput 52:411–435
33. Cockburn B, Hou S, Shu C-W (1990) TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. Math Comput 54:545–581
34. Cockburn B, Shu C-W (1998) The local discontinuous Galerkin method for time-dependent convection-diffusion systems. SIAM J Numer Anal 35:2440–2463
35. Cockburn B, Shu C-W (2001) Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. J Sci Comput 16:173–261
36. Cueto-Felgueroso L (2005) Particles, finite volumes and unstructured grids: numerical simulation of fluid dynamics problems. PhD Thesis (in Spanish), Universidad de A Coruña. http://www.tesisenred.net/
37. Cueto-Felgueroso L, Peraire J (2007) A time-adaptive finite volume method for the Cahn-Hilliard and Kuramoto-Sivashinsky equations. J Comput Phys, submitted
38. Cueto-Felgueroso L, Colominas I, Nogueira X, Navarrina F, Casteleiro M (2007) Finite volume solvers and moving least-squares approximations for the compressible Navier-Stokes equations on unstructured grids. Comput Methods Appl Mech Eng 196:4712–4736
39. Cueto-Felgueroso L, Colominas I, Mosqueira G, Navarrina F, Casteleiro M (2004) On the Galerkin formulation of the SPH method. Int J Numer Methods Eng 60:1475–1512
40. Cueto-Felgueroso L, Colominas I, Fe J, Navarrina F, Casteleiro M (2006) High order finite volume schemes on unstructured grids using moving least-squares reconstruction. Application to shallow water dynamics. Int J Numer Methods Eng 65:295–331
41. Deconinck H, Paillere H, Struijs R, Roe PL (1993) Multidimensional upwind schemes based on fluctuation-splitting for systems of conservation laws. Comput Mech 11:323–340
42. Douglas J, Dupont T (1976) Interior penalty procedures for elliptic and parabolic Galerkin methods. Lecture Notes in Phys, vol 58. Springer, Berlin
43. Durlofsky LJ, Engquist B, Osher S (1992) Triangle based adaptive stencils for the solution of hyperbolic conservation laws. J Comput Phys 98:64–73
44. Eigestad G, Klausen R (1998) Convergence of the MPFA O-method: numerical experiments for discontinuous media. Numer Methods Partial Differ Equ 21:1079–1098
45. Elliott CM, French DM, Milner FA (1989) A second order splitting method for the Cahn-Hilliard equation. Numer Math 54:575–590
46. Elliott CM, French DM (1989) A nonconforming finite element method for the two-dimensional Cahn-Hilliard equation. SIAM J Numer Anal 26:884–903
47. Feng X, Prohl A (2004) Analysis of a fully discrete finite element method for the phase field model and approximation of its sharp interface limits. Math Comput 73:541–567
48. Friedrich O (1998) Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. J Comput Phys 144:194–212
49. Frink NT (1992) Upwind scheme for solving the Euler equations on unstructured tetrahedral meshes. AIAA J 30:70
50. Furihata D (2001) A stable and conservative finite difference scheme for the Cahn-Hilliard equation. Numer Math 87:675–699
51. Godunov SK (1959) A difference method for the numerical calculation of discontinuous solutions of hydrodynamic equations. Mat Sb 47:271–306
52. Gossler A (2001) Moving least-squares: a numerical differentiation method for irregularly spaced calculation points. SANDIA Report, SAND2001-1669
53. Harten A, Chakravarthy S (1991) Multi-dimensional ENO schemes for general geometries. ICASE Report No. 91-76
54. Harten A, Lax P, Van Leer B (1983) On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. SIAM Rev 25:35–61
55. Harten A, Osher S (1987) Uniformly high order accurate nonoscillatory schemes I. SIAM J Numer Anal 24:279–309
56. Harten A, Engquist B, Osher S, Chakravarthy S (1987) Uniformly high order accurate essentially nonoscillatory schemes III. J Comput Phys 71:231–303
57. Hu C, Shu C-W (1999) Weighted essentially non-oscillatory schemes on triangular meshes. J Comput Phys 150:97–127
58. Iske A, Sonar T (1996) On the structure of function spaces of optimal recovery of point functions for ENO schemes by radial basis functions. Numer Math 74:177–201
59. Jawahar P, Kamath H (2000) A high-resolution procedure for Euler and Navier-Stokes computations on unstructured grids. J Comput Phys 164:165–203
60. Jiang GS, Shu C-W (1996) Efficient implementation of weighted ENO schemes. J Comput Phys 126:202–228
61. Kay D, Welford R (2006) A multigrid finite element solver for the Cahn-Hilliard equation. J Comput Phys 212:288–304
62. Kim D, Kang K, Lowengrub J (2004) Conservative multigrid methods for Cahn-Hilliard fluids. J Comput Phys 193:511–543
63. Klausen R, Russell T (2004) Relations among some locally conservative discretization methods which handle discontinuous coefficients. Comput Geosci 8:341–377
64. Klausen R, Winther R (2006) Convergence of multi point flux approximations on quadrilateral grids. Numer Methods Partial Differ Equ 22:1438–1454
65. Lancaster P, Salkauskas K (1981) Surfaces generated by moving least squares methods. Math Comput 155:141–158
66. Levin D (1998) The approximation power of moving least-squares. Math Comput 67:1517–1531
67. Li S, Liu WK (1996) Moving least-squares reproducing kernel methods: (II) Fourier analysis. Comput Methods Appl Mech Eng 139:159–193
68. Li S, Liu WK (1999) Reproducing kernel hierarchical partition of unity, part I—formulation and theory. Int J Numer Methods Eng 45:251–288
69. Liu WK, Hao W, Chen Y, Jun S, Gosz J (1997) Multiresolution reproducing kernel particle methods. Comput Mech 20:295–309
70. Liu WK, Li S, Belytschko T (1997) Moving least-squares reproducing kernel methods: (I) methodology and convergence. Comput Methods Appl Mech Eng 143:113–154
71. Liu XD, Osher S, Chan T (1994) Weighted essentially non-oscillatory schemes. J Comput Phys 115:200–212
72. Nejat A, Ollivier-Gooch CF, Michalak K (2007) Accuracy assessment methodology for a high-order unstructured finite volume solver. In: 18th AIAA computational fluid dynamics conference. American Institute of Aeronautics and Astronautics

73. Ollivier-Gooch CF (1997) High-order ENO schemes for unstructured meshes based on least-squares reconstruction. AIAA paper 97-0540
74. Ollivier-Gooch CF, Nejat A, Michalak K (2007) On obtaining high-order finite volume solutions to the Euler equations on unstructured meshes. In 18th AIAA computational fluid dynamics conference. American Institute of Aeronautics and Astronautics
75. Ollivier-Gooch CF, Van Altena M (2002) A high-order accurate unstructured mesh finite volume scheme for the advection-diffusion equation. J Comput Phys 181:729–752
76. Paillere H, Boxho J, Degrez G, Deconinck H (1996) Residual distribution schemes for advection and advection-diffusion problems on quadrilateral cells. Int J Numer Methods Fluids 23:923–936
77. De Palma P, Pascazio G, Rubino DT, Napolitano M (2006) Multidimensional upwind residual distribution schemes for the convection-diffusion equation. J Comput Phys 218:159–199
78. Peraire J, Persson P-O (2007) A Compact Discontinuous Galerkin method for elliptic problems. SIAM J Sci Comput, submitted
79. Ricchiuto M., Csík Á., Deconinck H (2005) Residual distribution for general time dependant conservation laws. J Comput Phys 209:249–289
80. Roe PL (1994) Multidimensional upwinding. Motivation and concepts. VKI lecture series 1994–1995
81. Roe PL (1981) Approximate Riemann solvers, parameter vectors and difference schemes. J Comput Phys 43:357–372
82. Shapiro AH (1953) The dynamics and thermodynamics of compressible fluid flow. Ronald Press, New York
83. Sherwin SJ, Kirby RM, Pieró J, Taylor RL, Zienkiewicz OC (2006) On 2D elliptic discontinuous Galerkin methods. Int J Numer Methods Eng 65:752–784
84. Shu C-W, Osher S (1988) Efficient implementation of essentially non-oscillatory shock-capturing schemes III. J Comput Phys 77:439–471
85. Shu C-W, Osher S (1988) Efficient implementation of essentially non-oscillatory shock-capturing schemes. J Comput Phys 77:439–471
86. Shu C-W, Osher S (1989) Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. J Comput Phys 83:32–78
87. Sjögreen B, Yee HC (2004) Multiresolution wavelet based adaptive numerical dissipation control for high order methods. J Sci Comput 20:211–255
88. Sonar T (1998) On families of pointwise optimal finite volume ENO approximations. SIAM J Numer Anal 35:2350–2369
89. Sonar T (1997) On the construction of essentially non-oscillatory finite volume approximations to hyperbolic conservation laws on general triangulations: polynomial recovery, accuracy and stencil selection. Comput Methods Appl Mech Eng 140:157–181
90. Sun Y, Wang ZJ (2004) Evaluation of discontinuous Galerkin and spectral volume methods for scalar and system conservation laws on unstructured grids. Int J Numer Methods Fluids 45:819–838
91. Sun ZZ (1995) A second-order accurate linearized finite difference scheme for the two-dimensional Cahn-Hilliard equation. Math Comput 64:1463–1471
92. Van Albada GD, Van Leer B, Roberts W (1982) A comparative study of computational methods in cosmic gas dynamics. Astron Astrophys 108:76–84
93. Van Leer B (1997) Towards the ultimate conservative difference scheme V. A second order sequel to Godunov's method. J Comput Phys 135:229–248
94. Van Leer B, Nomura S (1997) Discontinuous Galerkin for diffusion. In: 17th AIAA computational fluid dynamics conference. AIAA 2005-5108
95. Van Rosendale J (1989) Floating shock fitting via Lagrangian adaptive meshes. ICASE 94–89
96. Vaassen JM, Wautelet P, Essers JA (2001) A quadratic reconstruction scheme for hypersonic reacting flows on unstructured meshes. In: ECCOMAS computational fluid dynamics conference, Swansea, Wales, UK
97. Venkatakrishnan V (1995) Convergence to steady state solutions of the Euler equations on unstructured grids with limiters. J Comput Phys 118:120–130
98. Wang ZJ, Liu Y (2002) Spectral (finite) volume method for conservation laws on unstructured grids II: extension to two-dimensional scalar equation. J Comput Phys 179:665–697
99. Wang ZJ, Liu Y (2004) Spectral (finite) volume method for conservation laws on unstructured grids III: one-dimensional systems and partition optimization. J Sci Comput 20:137–157
100. Wang ZJ, Liu Y (2004) Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional systems. J Comput Phys 194:716–741
101. Wang ZJ, Liu Y (2006) Spectral (finite) volume method for conservation laws on unstructured grids VI: extension to viscous flow. J Comput Phys 215:41–58
102. Wang ZJ (2002) Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation. J Comput Phys 178:210–251
103. Wheeler M (1978) An elliptic collocation finite element method with interior penalties. SIAM J Numer Anal 15:152–161
104. Wells GN, Kuhl E, Garikipati K (2006) A discontinuous Galerkin method for the Cahn-Hilliard equation. J Comput Phys 218:860–877
105. White FM (1991) Viscous fluid flow, 2nd edn. McGraw-Hill, New York
106. Xia YX, Xu Y, Shu C-W (2007) Local discontinuous Galerkin methods for the Cahn-Hilliard type equations. J Comput Phys 227:472–491
107. Yan J, Shu C-W (2002) Local discontinuous Galerkin methods for partial differential equations with higher order derivatives. J Sci Comput 17:27–47
108. Yee HC, Sandham ND, Djomehri MJ (1999) Low-dissipative high-order shock-capturing methods using characteristic-based filters. J Comput Phys 150:199–238
109. Zhang M, Shu C-W (2005) An analysis of and a comparison between the discontinuous Galerkin and the spectral finite volume methods. Comput Fluids 34:581–592