INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2009; **78**:1553–1584
Published online 23 January 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.2538

On the accuracy of finite volume and discontinuous Galerkin discretizations for compressible flow on unstructured grids

X. Nogueira[1], L. Cueto-Felgueroso[2], I. Colominas[1,*,†], H. Gómez[1,3], F. Navarrina[1] and M. Casteleiro[1]

[1]*Group of Numerical Methods in Engineering, GMNI, Department of Applied Mathematics, Civil Engineering School, Universidade da Coruña, Campus de Elviña, 15071, A Coruña, Spain*
[2]*Aerospace Computational Design Laboratory, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 77 Massachusetts Ave., Cambridge, MA 02139, U.S.A.*
[3]*Institute for Computational Engineering and Sciences, The University of Texas at Austin, 1 University Station, C0200, 201 E. 24th Street, Austin, TX 78712, U.S.A.*

SUMMARY

This paper presents a comparison between two high-order methods. The first one is a high-order finite volume (FV) discretization on unstructured grids that uses a meshfree method (moving least squares (MLS)) in order to construct a piecewise polynomial reconstruction and evaluate the viscous fluxes. The second method is a discontinuous Galerkin (DG) scheme. Numerical examples of inviscid and viscous flows are presented and the solutions are compared. The accuracy of both methods, for the same grid resolution, is similar, although the DG scheme requires a larger number of degrees of freedom than the FV–MLS method. Copyright © 2009 John Wiley & Sons, Ltd.

Received 30 October 2007; Revised 3 November 2008; Accepted 27 November 2008

KEY WORDS: compressible flow; finite volume method; discontinuous Galerkin method; high-resolution schemes; high-order methods; moving least-squares approximation; unstructured grids

*Correspondence to: I. Colominas, Universidade da Coruña, Campus de Elviña, 15071, A Coruña, Spain.
†E-mail: icolominas@udc.es

Contract/grant sponsor: Ministerio de Educación y Ciencia; contract/grant numbers: DPI2004-05156, DPI2006-15275, DPI2007-61214
Contract/grant sponsor: FEDER funds
Contract/grant sponsor: Secretaría Xeral de I+D of the Xunta de Galicia; contract/grant numbers: PGIDIT05PXIC118002PN, PGDIT06TAM11801PR
Contract/grant sponsor: University of A Coruña
Contract/grant sponsor: Fundación Caixa Galicia
Contract/grant sponsor: Ministerio de Educación y Ciencia
Contract/grant sponsor: European Comission

1. INTRODUCTION

The field of computational fluid dynamics has developed greatly during the last decade. The increase of computer capabilities has allowed the possibility of solving progressively more complex problems, which require numerical methods with the capability for capturing the flow features in a very accurate way. DNS and LES of turbulent flows, or aeroacoustics simulations are examples of these kind of problems. In aeroacoustics, for example, an accurate solution of flow field is needed due to the very low magnitude of acoustic waves. Indeed, the use of first- and second-order upwind schemes can lead to excessive numerical dissipation that absorbs the acoustic wave propagation and drives the simulation to a wrong or not accurate enough solution. In addition, in problems where turbulent effects play a significant role, the numerical method has to avoid interactions with the turbulence model.

Pseudospectral methods and finite differences are the most natural and commonly used numerical schemes when very high accuracy is needed. They are extremely competitive on simple or moderately complex geometries, and unbeatable, both in terms of accuracy and efficiency, by unstructured-grid approaches. However and even though the use of multiblock grids allows the extension of structured-grid procedures to rather complex geometries, the development of unstructured-grid methodologies is nowadays perceived as the most promising compromise to achieve the dreamed accurate-fast-automated final objective. It is within this context of unstructured-grid methods that the present study originated.

Continuous finite element formulations for fluid dynamics are applicable to a wide variety of flow conditions. Unfortunately, the too frequently required stabilization, with its associated numerical dissipation and lack of robustness, has hampered their widespread use for compressible flow applications.

Finite volume (FV) schemes are widely used, including in industrial problems. Nevertheless, the usual order achieved in these simulations is usually two. This is due to the difficulty of evaluating high-order derivatives of the field variables from scattered, point wise information. These derivatives are required for the reconstruction of field variables by Taylor approximation [1–3]. Moreover, most of multidimensional schemes are extensions of one-dimensional solvers, so if the direction of the flow is not normal to the cell interface, dissipation is a fact [4]. This difficulty is overcome by methods that solve the problem with a Lagrangian approach, the so called meshfree or particle methods. In this context, the smooth particle hydrodynamics method was first introduced by Gingold and Monaghan [5] in astrophysical applications. For acoustic applications, Eldredge, Colonius and Leonard have used a vortex particle method for the calculation of a corotating vortex pair [6]. In [7–9] a method that uses a meshfree technique (moving least squares (MLS)) has been proposed to calculate high-order derivatives of the field variables, achieving a truly multidimensional high-order approach with a FV scheme. In this method, the spatial FV discretization uses the MLS approximation as a kind of 'shape functions' for unstructured grids.

On the other hand, few years ago, discontinuous Galerkin (DG) schemes have attracted the attention of researchers. This method was first applied in the solution of neutron transport problems in 1973 by Reed and Hill [10], and has only been recently extended to general problems in fluid dynamics [11–14]. This technique combines the physics of wave propagation inherent to finite volume methods with the accuracy of high-order polynomial approximations within elements. In opposition to continuous finite elements, this method allows the solution to be discontinuous between elements. The coupling between elements is achieved by means of numerical fluxes, as

DOI: 10.1002/nme

in the FV technique. However, the discontinuous nature of the approximation brings the main drawback of this method: the increasing number of degrees of freedom (dof).

In this paper we compare two high-order methods (the DG and the FV–MLS schemes) by using numerical tests. For the DG method, we take third-order polynomials for the reconstruction of the variables. A Taylor reconstruction up to the third derivative is used for the convective terms on the FV–MLS method. Viscous fluxes are computed directly at the quadrature points with a cubic reconstruction. Therefore, a comparison is made between two fourth-order methods.

We study the accuracy of both methods only for cases with smooth solutions. Notice that the FV–MLS method works well in problems with non-smooth solutions [7–9], by using shock limiter techniques designed for FV solvers.

The outline of the paper is as follows. Section 2 presents the governing equations and Sections 3 and 4 are devoted to numerical formulations. Interpolation methods are examined in Section 5, whereas accuracy test and representative simulations are exposed and discussed in Sections 6 and 7. Finally, conclusions are drawn.

2. GOVERNING EQUATIONS

The two-dimensional Navier–Stokes equations can be written in conservative form as

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

with

$$\mathbf{U}=\begin{pmatrix}\rho\\ \rho v_x\\ \rho v_y\\ \rho E\end{pmatrix} \tag{2}$$

$$\mathbf{F}_x=\begin{pmatrix}\rho v_x\\ \rho v_x^2+p\\ \rho v_x v_y\\ \rho v_x H\end{pmatrix},\quad \mathbf{F}_y=\begin{pmatrix}\rho v_y\\ \rho v_x v_y\\ \rho v_y^2+p\\ \rho v_y H\end{pmatrix} \tag{3}$$

$$\mathbf{F}_x^{\mathrm{V}}=\begin{pmatrix}0\\ \tau_{xx}\\ \tau_{xy}\\ v_x\tau_{xx}+v_y\tau_{xy}-q_x\end{pmatrix},\quad \mathbf{F}_y^{\mathrm{V}}=\begin{pmatrix}0\\ \tau_{xy}\\ \tau_{yy}\\ v_x\tau_{xy}+v_y\tau_{yy}-q_y\end{pmatrix} \tag{4}$$

DOI: 10.1002/nme

with

$$\rho E = \rho e + \tfrac{1}{2}\rho(\mathbf{v}\cdot\mathbf{v}) \tag{5}$$

$$H = E + \frac{p}{\rho} \tag{6}$$

and

$$\begin{aligned} \tau_{xx} &= 2\mu\frac{\partial v_x}{\partial x} - \frac{2}{3}\mu\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}\right) \\ \tau_{yy} &= 2\mu\frac{\partial v_y}{\partial y} - \frac{2}{3}\mu\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}\right) \end{aligned} \tag{7}$$

$$\tau_{xy} = \mu\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right) \tag{8}$$

In the above expressions, $\mathbf{U}$ is the vector of conserved variables, $\mathbf{v}=(v_x, v_y)$ is the velocity, μ is the effective viscosity of the fluid, H is the enthalpy, E is the total energy, e is the total internal energy and ρ is the density. The notation $\mathscr{F}=(\mathbf{F}_x, \mathbf{F}_y)$ and $\mathscr{F}^{\mathrm{V}}=(\mathbf{F}_x^{\mathrm{V}}, \mathbf{F}_y^{\mathrm{V}})$ will be used hereafter in reference to the inviscid and viscous fluxes, respectively.

The thermal flux ($\mathbf{q}=(q_x, q_y)$) is modeled by Fourier's law

$$q_x = -\lambda\frac{\partial T}{\partial x}, \quad q_y = -\lambda\frac{\partial T}{\partial y} \tag{9}$$

Alternative formulations are possible. The authors are working on developing alternative formulations for the calculation of thermal fluxes based on Cattaneo's law [15].

It is assumed that the viscosity depends on the temperature according to Sutherland's law

$$\mu = \mu_{\mathrm{ref}}\frac{T_{\mathrm{ref}} + S_0}{T + S_0}\left(\frac{T}{T_{\mathrm{ref}}}\right)^{1.5} \tag{10}$$

where the subindex ref refers to a reference value and $S_0 = 110.4K$ is an empirical constant.

3. HIGH-ORDER FV METHOD

3.1. Overview

The first discretization analyzed in this study is a high-order unstructured-grid FV method for convection-dominated flows developed by the authors [7–9]. The key ingredient of this scheme is a point-wise local high-order approximation framework, which provides a continuous representation of the solution, and thus allows a simple and efficient discretization of equations with high-order terms.

The usual approach of high-order FV schemes is pragmatic and *bottom-up*. Starting from an underlying piecewise constant representation, a discontinuous reconstruction of the field variables is performed at the cell level. An important practical consequence is that the discretization of higher-order terms requires some kind of *recovery* procedure, which is, almost invariably, inconsistent with

DOI: 10.1002/nme

the aforementioned reconstruction. Our approach is somewhat the opposite. We start from a high-order and highly regular representation of the solution, obtained by means of MLS approximation [16], and well suited for general, unstructured grids. This approach is directly suitable for the discretization of elliptic/parabolic equations and high-order spatial terms. For equations with a predominantly hyperbolic character, the global representation is *broken* locally, at the cell level, into a piecewise polynomial reconstruction, which allows to use the powerful FV technology of Godunov-type schemes for hyperbolic problems (e.g. Riemann solvers, limiters).

3.2. General formulation

Consider a system of conservation laws of the form

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathscr{F}^{\mathrm{H}} + \mathscr{F}^{\mathrm{E}}) = \mathbf{S} \quad \text{in } \Omega \tag{11}$$

supplemented with suitable initial and boundary conditions. The fluxes have been generically split into a hyperbolic-like part, $\mathscr{F}^{\mathrm{H}}$, and an elliptic-like part, $\mathscr{F}^{\mathrm{E}}$. Consider, in addition, a partition of the domain Ω into a set of non-overlapping *control volumes* or *cells*, $\mathscr{T}^h = I$. Furthermore, we define a reference point (*node*), $\mathbf{x}_I$ inside each cell (the cell centroid).

The spatial representation of the solution is as follows: consider a function $\mathbf{u}(\mathbf{x})$, given by its point values, $\mathbf{u}_I = \mathbf{u}(\mathbf{x}_I)$, at the cell centroids, with coordinates $\mathbf{x}_I$. The approximate function $\mathbf{u}^h(\mathbf{x})$ belongs to the subspace spanned by a set of *basis functions* $\{N_I(\mathbf{x})\}$ associated to the nodes, such that $\mathbf{u}^h(\mathbf{x})$ is given by

$$\mathbf{u}^h(\mathbf{x}) = \sum_{j=1}^{n_{\mathbf{x}}} N_j(\mathbf{x}) \mathbf{u}_j \tag{12}$$

which states that the approximation at a point $\mathbf{x}$ is computed using certain $n_{\mathbf{x}}$ surrounding nodes. This set of nodes is referred to as the *cloud* or *stencil* associated to the evaluation point $\mathbf{x}$. In particular, the above approximation is constructed using MLS approximation [16]. Note that, using MLS, the approximate function $\mathbf{u}^h(\mathbf{x})$ is not a polynomial in general. An interesting feature of this MLS approach is the *centered* character of the approximation; thus, avoiding the spatial bias, which is often found in patch-based piecewise polynomial interpolation.

Consider now the integral form of the system of conservation laws (11) which, for each control volume I, reads as

$$\int_{\Omega_I} \frac{\partial \mathbf{u}}{\partial t} \, \mathrm{d}\Omega + \int_{\Gamma_I} (\mathscr{F}^{\mathrm{H}} + \mathscr{F}^{\mathrm{E}}) \cdot \mathbf{n} \, \mathrm{d}\Gamma = \int_{\Omega_I} \mathbf{S} \, \mathrm{d}\Omega \tag{13}$$

Introducing the component-wise reconstructed function $\mathbf{u}^h$, the spatially discretized counterpart of (13) reads as

$$\int_{\Omega_I} \frac{\partial \mathbf{u}^h}{\partial t} \, \mathrm{d}\Omega + \int_{\Gamma_I} (\mathscr{F}^{h\mathrm{H}} + \mathscr{F}^{h\mathrm{E}}) \cdot \mathbf{n} \, \mathrm{d}\Gamma = \int_{\Omega_I} \mathbf{S}^h \, \mathrm{d}\Omega \tag{14}$$

A direct evaluation of the fluxes in (14) is possible and efficient when the inherent dissipation mechanism is strong enough to overpower the convective terms. In convection-dominated problems, where the character of the equations is predominantly hyperbolic, this centered approach can lead to unstable computations. For this latter type of problems, we introduce a '*broken*' reconstruction,

DOI: 10.1002/nme

$\mathbf{u}_I^{hb}$, which approximates $\mathbf{u}^h(\mathbf{x})$ (and, therefore, $\mathbf{u}(\mathbf{x})$) locally inside each cell I, and is discontinuous across cell interfaces [7, 9]. In general, we require the order of accuracy of the broken reconstruction to be the same as that of the original continuous reconstruction. One possible choice is to use Taylor series expansions; a quadratic reconstruction inside cell I, for example, would read as

$$\mathbf{u}_I^{hb}(\mathbf{x}) = \mathbf{u}_I^h + \nabla \mathbf{u}_I^h \cdot (\mathbf{x} - \mathbf{x}_I) + \tfrac{1}{2}(\mathbf{x} - \mathbf{x}_I)^{\mathrm{T}} \mathbf{H}^h (\mathbf{x} - \mathbf{x}_I) \tag{15}$$

where the gradient $\nabla \mathbf{u}_I^h$ and the Hessian matrix $\mathbf{H}^h$ involve the successive derivatives of the continuous reconstruction $\mathbf{u}^h(\mathbf{x})$, which are evaluated at the cell centroids using MLS. This dual continuous/discontinuous reconstruction of the solution is crucial in order to obtain accurate and efficient numerical schemes for mixed parabolic/hyperbolic problems. The cell-wise broken reconstruction defined here is actually a piecewise continuous approximation to $\mathbf{u}^h$. The advantage is that it allows to make use of Riemann solvers, limiters and other standard FV 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 for the continuous/discontinuous approach can be written as

$$\int_{\Omega_I} \frac{\partial \mathbf{u}^h}{\partial t} \, \mathrm{d}\Omega + \int_{\Gamma_I} \mathbf{H}(\mathbf{u}^{hb+}, \mathbf{u}^{hb-}) \, \mathrm{d}\Gamma + \int_{\Gamma_I} \mathscr{F}^{h\mathrm{E}} \cdot \mathbf{n} \, \mathrm{d}\Gamma = \int_{\Omega_I} \mathbf{S}^h \, \mathrm{d}\Omega \tag{16}$$

where $\mathbf{H}(\mathbf{u}^{hb+}, \mathbf{u}^{hb-})$ is a suitable numerical flux.

3.3. MLS approximation

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

$$u(\mathbf{x}) \approx \hat{u}(\mathbf{x}) = \sum_{i=1}^{m} p_i(\mathbf{x}) \alpha_i(\mathbf{z})|_{\mathbf{z}=\mathbf{x}} = \mathbf{p}^{\mathrm{T}}(\mathbf{x}) \boldsymbol{\alpha}(\mathbf{z})|_{\mathbf{z}=\mathbf{x}} \tag{17}$$

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

$$J(\boldsymbol{\alpha}(\mathbf{z})|_{\mathbf{z}=\mathbf{x}}) = \int_{\mathbf{y} \in \Omega_{\mathbf{x}}} W(\mathbf{z} - \mathbf{y}, h)|_{\mathbf{z}=\mathbf{x}} [u(\mathbf{y}) - \mathbf{p}^{\mathrm{T}}(\mathbf{y}) \boldsymbol{\alpha}(\mathbf{z})|_{\mathbf{z}=\mathbf{x}}]^2 \, \mathrm{d}\Omega_{\mathbf{x}} \tag{18}$$

where $W(\mathbf{z} - \mathbf{y}, h)|_{\mathbf{z}=\mathbf{x}}$ is a *kernel* with compact support (denoted by $\Omega_{\mathbf{x}}$) centered at $\mathbf{z} = \mathbf{x}$.

The parameter h is the smoothing length, which is a measure of the size of the support $\Omega_{\mathbf{x}}$.

The minimization of J leads to

$$\int_{\mathbf{y} \in \Omega_{\mathbf{x}}} \mathbf{p}(\mathbf{y}) W(\mathbf{z} - \mathbf{y}, h)|_{\mathbf{z}=\mathbf{x}} u(\mathbf{y}) \, \mathrm{d}\Omega_{\mathbf{x}} = \mathbf{M}(\mathbf{x}) \boldsymbol{\alpha}(\mathbf{z})|_{\mathbf{z}=\mathbf{x}} \tag{19}$$

$\mathbf{M}(\mathbf{x})$ is the moment matrix, defined by

$$\mathbf{M}(\mathbf{x}) = \int_{\mathbf{y} \in \Omega_{\mathbf{x}}} \mathbf{p}(\mathbf{y}) W(\mathbf{z} - \mathbf{y}, h)|_{\mathbf{z}=\mathbf{x}} \mathbf{p}^{\mathrm{T}}(\mathbf{y}) \, \mathrm{d}\Omega_{\mathbf{x}} \tag{20}$$

DOI: 10.1002/nme

In numerical computations the global domain Ω is represented by a set of *nodes* or *particles*. The integral in (18) is thus evaluated using those nodes inside $\Omega_{\mathbf{x}}$ as quadrature points. In discrete form, the set of parameters $\boldsymbol{\alpha}$ that minimize the functional J are given by

$$\boldsymbol{\alpha}(\mathbf{z})|_{\mathbf{z}=\mathbf{x}} = \mathbf{M}^{-1}(\mathbf{x})\mathbf{P}_{\Omega_{\mathbf{x}}}\mathbf{W}_V(\mathbf{x})\mathbf{u}_{\Omega_{\mathbf{x}}} \tag{21}$$

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

Introducing (21) in (17), the interpolation structure can be identified as

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

In analogy to finite elements, the approximation is written in terms of the MLS 'shape functions' as

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

The basis of functions used in this study comprise scaled and *locally* defined monomials. Thus, in order to reconstruct a one-dimensional function at a location x_I, we use basis of the form

$$\mathbf{p}(\mathbf{x}) = \left(1 \quad \frac{x-x_I}{h} \quad \left(\frac{x-x_I}{h}\right)^2 \quad \cdots \quad \left(\frac{x-x_I}{h}\right)^p\right)^{\mathrm{T}} \tag{24}$$

The functional basis $\mathbf{p}(\mathbf{x})$ is strongly related to the accuracy of the MLS fit. 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 an 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 multidimensions we follow the same idea of p-complete basis, constructed using products of *scaled* and *locally defined* monomials.

The MLS shape functions are data independent and, therefore, for fixed grids they need to be computed *only once* at the preprocessing phase. In order to prevent the moment matrix $\mathbf{M}$ from being singular or ill-conditioned, the cloud of neighbors must fulfill certain 'good neighborhood' requirements. The definition of the cloud (the *MLS stencil*) for each evaluation point is a crucial issue that requires careful attention. 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. Note that these stencils are typically *centered* around the node, and thus the MLS approximation avoids the spatial bias, which is often found in patch-based piecewise polynomial approximations.

The particles needed for the application of the method are identified with the centroids of every cell of the grid. In the case of boundary cells, we add nodes (ghost nodes) placed in the middle of the edge wall.

The approximate derivatives of $u(\mathbf{x})$ can be expressed in terms of the derivatives of the MLS shape functions, which are functions of the derivatives of the polynomial basis $\mathbf{p}(\frac{\mathbf{x}-\mathbf{x}_I}{h})$ and the derivatives of $\mathbf{C}(\mathbf{x})$ [7, 9]. For example, the first-, second- and third-order derivatives of $u(\mathbf{x})$,

DOI: 10.1002/nme

evaluated at $\mathbf{x}_I$, are given by

$$\begin{aligned}
\left.\frac{\partial u(\mathbf{x})}{\partial x_\alpha}\right|_{\mathbf{x}=\mathbf{x}_I} &\approx \sum_{j=1}^{n_{\mathbf{x}_I}} u_j \left.\frac{\partial N_j(\mathbf{x})}{\partial x_\alpha}\right|_{\mathbf{x}=\mathbf{x}_I} \\
\left.\frac{\partial^2 u(\mathbf{x})}{\partial x_\alpha \partial x_\beta}\right|_{\mathbf{x}=\mathbf{x}_I} &\approx \sum_{j=1}^{n_{\mathbf{x}_I}} u_j \left.\frac{\partial^2 N_j(\mathbf{x})}{\partial x_\alpha \partial x_\beta}\right|_{\mathbf{x}=\mathbf{x}_I} \\
\left.\frac{\partial^3 u(\mathbf{x})}{\partial x_\alpha \partial x_\beta \partial x_\gamma}\right|_{\mathbf{x}=\mathbf{x}_I} &\approx \sum_{j=1}^{n_{\mathbf{x}_I}} u_j \left.\frac{\partial^3 N_j(\mathbf{x})}{\partial x_\alpha \partial x_\beta \partial x_\gamma}\right|_{\mathbf{x}=\mathbf{x}_I}
\end{aligned} \tag{25}$$

In this work we have used a polynomial cubic basis in (24), so we can compute until the third derivative. Therefore, the maximum order of the FV–MLS scheme we can get with this basis is four. The polynomial basis also defines the minimum number of neighbors (nodes) needed in the stencil for the MLS approximation. Thus, for the cubic basis and two-dimensional problems at least 10 points are required. However, it is recommended to increase slightly this number in order to avoid an ill conditioning of the moment matrix $\mathbf{M}$. On the other hand, it is possible to obtain higher orders by using quadratic or higher basis. In this case, the number of points in the stencil needs to be increased. In practical applications, however, the maximum attainable orders within the MLS context is up to fourth–sixth order.

3.4. MLS-based reconstruction

Standard high-order FV schemes are constructed through the substitution of a piecewise constant representation for a *piecewise continuous* (usually polynomial) reconstruction of the flow variables inside each cell. Owing to the fact that the reconstructed fields are still discontinuous across interfaces, the discretization of viscous terms must be done carefully. This 'bottom-up' procedure is quite different from the way in that the FV–MLS method works. Here, we work with *pointwise* values of the conserved variables, associated to the cell centroids. The spatial representation provided by the MLS approximants is *continuous* and high-order accurate. In order to deal with convection-dominated problems and to take the most of the FV technology for hyperbolic terms, we *break* the continuous representation inside each cell by means of Taylor expansions ('top-down' procedure). The resulting scheme is like a Godunov method in the convective terms, but with a much clearer and more accurate discretization of elliptic terms. In the following sections, it is shown how the reconstruction process is made.

3.4.1. Reconstruction: *inviscid fluxes.* To apply a FV scheme, a reconstruction scheme to evaluate the value of the variables at the edges of the element and compute the numerical flux is needed.

Using a Taylor series expansion, the linear component-wise reconstruction of the field variables inside each cell I reads as

$$\mathbf{U}(\mathbf{x}) = \mathbf{U}_I + \nabla \mathbf{U}_I \cdot (\mathbf{x} - \mathbf{x}_I) \tag{26}$$

$\mathbf{U}_I$ is the average value of $\mathbf{U}$ over I (associated to the centroid), $\mathbf{x}_I$ denotes the Cartesian coordinates of the centroid and $\nabla \mathbf{U}_I$ is the gradient of the variable at the centroid. The aforementioned gradient is assumed to be constant inside each cell and, therefore, the reconstructed variable

DOI: 10.1002/nme

is still discontinuous across interfaces. Note that we have broken the continuity of the spatial representation of the variable. The first-, second- and third-order derivatives of the field variables will be computed using MLS approximation.

Analogously, the quadratic reconstruction reads as

$$\mathbf{U}(\mathbf{x}) = \mathbf{U}_I + \nabla \mathbf{U}_I \cdot (\mathbf{x} - \mathbf{x}_I) + \tfrac{1}{2}(\mathbf{x} - \mathbf{x}_I)^{\mathrm{T}} \mathbf{H}_I (\mathbf{x} - \mathbf{x}_I) \tag{27}$$

where $\mathbf{H}_I$ is the centroid Hessian matrix.

In case of cubic reconstruction

$$\mathbf{U}(\mathbf{x}) = \mathbf{U}_I + \nabla \mathbf{U}_I \cdot (\mathbf{x} - \mathbf{x}_I) + \tfrac{1}{2}(\mathbf{x} - \mathbf{x}_I)^{\mathrm{T}} \mathbf{H}_I (\mathbf{x} - \mathbf{x}_I) + \tfrac{1}{6} \boldsymbol{\Delta}^2 \mathbf{x}_I^{\mathrm{T}} \mathbf{T}_I (\mathbf{x} - \mathbf{x}_I) \tag{28}$$

with

$$\boldsymbol{\Delta}^2 \mathbf{x}_I^{\mathrm{T}} = ((x - x_I)^2 (y - y_I)^2) \tag{29}$$

$$\mathbf{T}_I = \begin{pmatrix} \dfrac{\partial^3 \mathbf{U}_I}{\partial x^3} & 3\dfrac{\partial^3 \mathbf{U}_I}{\partial x^2 \partial y} \\ 3\dfrac{\partial^3 \mathbf{U}_I}{\partial x \partial y^2} & \dfrac{\partial^3 \mathbf{U}_I}{\partial y^3} \end{pmatrix} \tag{30}$$

For unsteady problems, additional terms must be introduced in (27) and (28) to enforce conservation of the mean, i.e.

$$\frac{1}{A_I} \int_{\mathbf{x} \in \Omega_I} U(\mathbf{x}) \, \mathrm{d}\Omega = U_I \tag{31}$$

Thus, the quadratic reconstruction for unsteady problems reads as

$$\mathbf{U}(\mathbf{x}) = \mathbf{U}_I + \nabla \mathbf{U}_I \cdot (\mathbf{x} - \mathbf{x}_I) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_I)^{\mathrm{T}} \mathbf{H}_I (\mathbf{x} - \mathbf{x}_I) - \frac{1}{2A_I} \left[I_{xx} \frac{\partial^2 \mathbf{U}}{\partial x^2} + 2 I_{xy} \frac{\partial^2 \mathbf{U}}{\partial x \partial y} + I_{yy} \frac{\partial^2 \mathbf{U}}{\partial y^2} \right] \tag{32}$$

with

$$I_{xx} = \int_\Omega (x - x_I)^2 \, \mathrm{d}\Omega, \quad I_{xy} = \int_\Omega (x - x_I)(y - y_I) \, \mathrm{d}\Omega, \quad I_{yy} = \int_\Omega (y - y_I)^2 \, \mathrm{d}\Omega \tag{33}$$

3.4.2. Direct evaluation of the viscous fluxes. The discretization of the viscous terms is a somewhat challenging task for methods that use piecewise polynomial approximations. A popular approach for second-order schemes is to use the average of the derivatives of the flow variables on either side of the interface to compute the viscous fluxes. This is not acceptable, however, for higher-order discretizations. Another option, which is customary in DG methods, is to decompose the original second-order problem into a first-order one, with the subsequent introduction of additional dof, which, in some cases, can be expressed in terms of the primal ones, and therefore 'eliminated'. The proposed FV scheme, through the use of MLS approximations, performs a centered, direct

DOI: 10.1002/nme

evaluation of the viscous fluxes at the quadrature points on the edges using information from neighboring cells. Thus, focusing on the Navier–Stokes equations, the evaluation of the viscous stresses and heat fluxes requires interpolating the velocity vector $\mathbf{v}=(v_x, v_y)$, temperature T, and their corresponding gradients, $\nabla\mathbf{v}$ and ∇T, at each quadrature point $\mathbf{x}_{iq}$. Using MLS approximation, these entities are readily computed as

$$\mathbf{v}_{iq}=\sum_{j=1}^{n_{iq}}\mathbf{v}_j N_j(\mathbf{x}_{iq}),\quad T_{iq}=\sum_{j=1}^{n_{iq}}T_j N_j(\mathbf{x}_{iq}) \tag{34}$$

and

$$\nabla\mathbf{v}_{iq}=\sum_{j=1}^{n_{iq}}\mathbf{v}_j\otimes\nabla N_j(\mathbf{x}_{iq}),\quad \nabla T_{iq}=\sum_{j=1}^{n_{iq}}T_j\nabla N_j(\mathbf{x}_{iq}) \tag{35}$$

where n_{iq} is the number of neighbor centroids given by the stencil. Then, the diffusive fluxes can be computed according to (4).

3.5. The kernel function

In this work the following cubic kernel is used for the FV–MLS calculations

$$W(s)=\begin{cases}1-\frac{3}{2}s^2+\frac{3}{4}s^3, & s\leqslant 1\\ \frac{1}{4}(2-s)^3, & 1<s\leqslant 2\\ 0, & s>2\end{cases} \tag{36}$$

In (36) $s=\frac{|x-x_j|}{h}$ and $h=\kappa\max(\|x_j-x_I\|)$ with $j=1,\ldots,n_{\mathbf{x}_I}$. We have chosen the value of $\kappa=0.7$ for the calculation of the derivatives of the MLS shape functions used in the computation of inviscid fluxes and the value of $\kappa=0.52$ for the direct evaluation of viscous fluxes. For the case of inviscid fluxes, the influence of the value of κ in the properties of the scheme can be examined using a Fourier analysis of the linear convection equation. The dispersion and dissipation of the scheme are related to the real and imaginary parts of the modified wave number. In Figure 1, it is shown the variation of the properties of the scheme with the parameter κ for the 1D convection equation.

For inviscid fluxes, the derivatives of the variables are used to compute a Taylor reconstruction of the variables inside the cell. For the direct evaluation of viscous fluxes or elliptic terms, the behavior is different, since the derivatives or variables that we compute are used directly. If we consider a one-dimensional stencil, Fourier analysis gives the modified wave number for the MLS approximation

$$\text{Modified wave number}=\text{MWN}=(-\text{i})\sum_{l=-P}^{Q}\frac{\partial N_{lj}}{\partial x}\mathrm{e}^{\mathrm{i}k(l\Delta x)} \tag{37}$$

where P is the number of neighbors to the left of the cell I and Q is the number of neighbors to the right of the cell I. Moreover, k is the wave number. In order to get some conclusions, we

DOI: 10.1002/nme

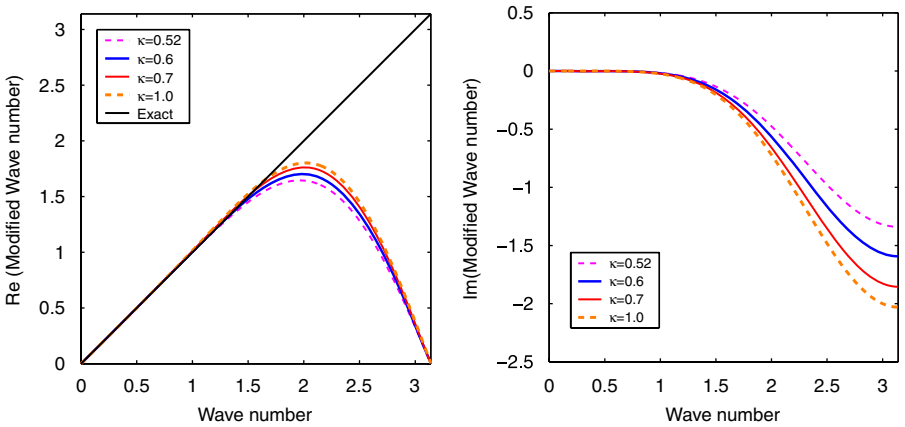

Figure 1. Variation of the properties of the FV–MLS scheme for the 1D linear convection equation. On the left, we show the real part of the modified wave number, related to the dispersion. On the right, we plot the imaginary part of the modified wave number that is related with the dissipation. These results can be related to the behavior of the scheme for the approximation of inviscid fluxes.

particularize for the case of a five-element stencil $P=Q=2$

$$\begin{aligned}\mathrm{MWN} &= (-\mathrm{i})\sum_{l=-P}^{Q}\frac{\partial N_{lj}}{\partial x}\mathrm{e}^{\mathrm{i}k(l\Delta x)}\\ &= \sin(2k\Delta x)\left(\frac{\partial N_{(I+2)}}{\partial x}-\frac{\partial N_{(I-2)}}{\partial x}\right)+\sin(k\Delta x)\left(\frac{\partial N_{(I+1)}}{\partial x}-\frac{\partial N_{(I-1)}}{\partial x}\right)\\ &\quad -\mathrm{i}\left(\cos(2k\Delta x)\left(\frac{\partial N_{(I+2)}}{\partial x}+\frac{\partial N_{(I-2)}}{\partial x}\right)+\cos(k\Delta x)\left(\frac{\partial N_{(I+1)}}{\partial x}+\frac{\partial N_{(I-1)}}{\partial x}\right)+\frac{\partial N_I}{\partial x}\right)\end{aligned} \tag{38}$$

For an isotropic distribution of points and a given stencil for the cubic spline, there is no influence of the parameter κ on the real part of the MLS approximation, since the difference of the shape functions

$$\frac{\partial N_{(I+i)}}{\partial x}-\frac{\partial N_{(I-i)}}{\partial x}$$

is constant. Moreover

$$\frac{\partial N_I}{\partial x}=0 \tag{39}$$

$$\frac{\partial N_{(I+i)}}{\partial x}=-\frac{\partial N_{(I-i)}}{\partial x} \tag{40}$$

Then, the imaginary part of the approximation vanishes, and there is no dissipation. We remark that these results only hold for the kernel (36). Even though for a given stencil the modified wave number is independent of κ, it affects the size of the support and therefore it modifies the weighting

DOI: 10.1002/nme

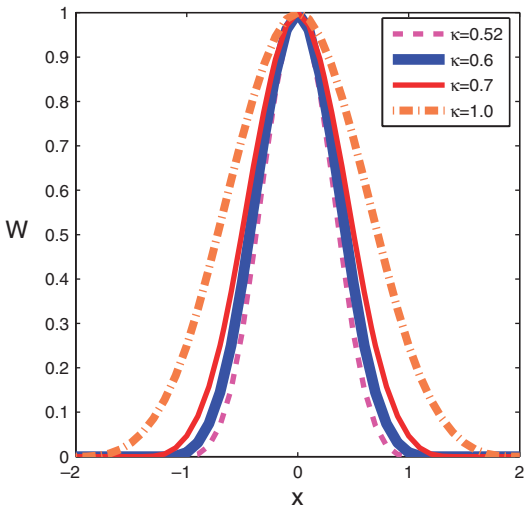

Figure 2. Variation of the shape of the cubic kernel in terms of κ. The kernel is centered in the cell $I=0$. The distance between nodes is $\Delta x=1$.

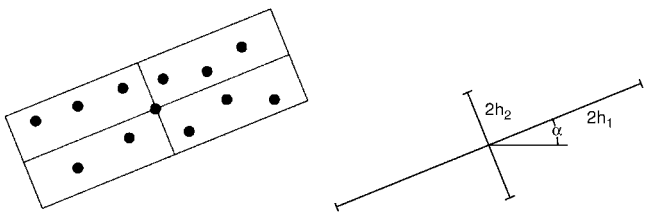

Figure 3. Support with different smoothing lengths (left) and orientation of inertial axis (right).

of the points of the stencil, as we show in Figure 2. Using radial weighting, the support of the kernel expands over a circle of radius $2h$, so it is worth to note that the selection of a value of κ small could lead to a bad conditioning of the moment matrix $\mathbf{M}$. As a practical rule $\kappa \geqslant 0.52$. In unstructured grids, the distribution of nodes is not isotropic and the behavior of the scheme will be affected. It is possible to define an optimum value of κ for each point of the grid, and this subject is currently under research. On the other hand, it is clear that a different selection of the kernel function produces different properties of the scheme.

3.5.1. Isotropic and non-isotropic kernels. The most frequently used kernels in the literature are splines and exponential functions. Kernels are defined over a support whose size is controlled by the smoothing length h. If the same smoothing length is used for each direction in a two-dimensional problem, the support will be a circle with radius $2h$. Another possibility would be to use different smoothing lengths for every direction. Then, if the kernel is defined as the product of the kernels in every direction, the support will be a rectangle (Figure 3, left), but any other geometric form is possible (i.e. elliptical kernels). In addition, there is the possibility of taking a kernel in the x-direction different of the kernel taken for the y-direction.

DOI: 10.1002/nme

In order to build an anisotropic kernel, we take the clouds of points given by the stencil and calculate their principal inertial axis. Then we can obtain the direction of these axes (Figure 3, right). In two dimensional, this is given by

$$\tan(2\alpha) = \frac{2I_{xy}}{I_{xx} - I_{yy}} \tag{41}$$

Then, changing the coordinates, with $\beta = -\alpha$

$$\mathbf{x}^* = \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix} \mathbf{x} \tag{42}$$

Now, we obtain the smoothing lengths in each new direction as

$$\begin{aligned} h_{x^*} &= \kappa_x \max(\|x_j - x_I\|), \quad j = 1, \ldots, n_{\mathbf{x}_I} \\ h_{y^*} &= \kappa_y \max(\|y_j - y_I\|), \quad j = 1, \ldots, n_{\mathbf{x}_I} \end{aligned} \tag{43}$$

and the rectangular anisotropic kernel will be given by

$$W = W_{x^*} W_{y^*} \tag{44}$$

The derivatives in global coordinates are given by

$$\begin{pmatrix} \dfrac{\partial N}{\partial x} \\ \dfrac{\partial N}{\partial y} \end{pmatrix} = \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix} \begin{pmatrix} \dfrac{\partial N}{\partial x^*} \\ \dfrac{\partial N}{\partial y^*} \end{pmatrix} \tag{45}$$

and the second derivatives by

$$\begin{pmatrix} \dfrac{\partial^2 N}{\partial x^2} & \dfrac{\partial^2 N}{\partial x \partial y} \\ \dfrac{\partial^2 N}{\partial y \partial x} & \dfrac{\partial^2 N}{\partial y^2} \end{pmatrix} = \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix} \begin{pmatrix} \dfrac{\partial^2 N}{\partial x^{*2}} & \dfrac{\partial^2 N}{\partial x^* \partial y^*} \\ \dfrac{\partial^2 N}{\partial y^* \partial x^*} & \dfrac{\partial^2 N}{\partial y^{*2}} \end{pmatrix} \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix} \tag{46}$$

Higher-order derivatives are straightforward.

3.6. Full stencil of the MLS–FV scheme

Figure 5 shows the stencils to compute the MLS shape functions for the fourth-order FV–MLS method. A complete description of the stencils can be found in [8, 9]. For inviscid problems, the stencil of a cell I is obtained as the union of its MLS stencil (Figure 4) and the MLS stencils of its first neighbors. As it can be seen in Figure 5 the full stencil comprises 25 cells. Analogously, the stencil of the 'viscous' discretization is obtained as the union of the MLS stencils associated to all the edges of cell I. This stencil comprises 21 cells.

DOI: 10.1002/nme

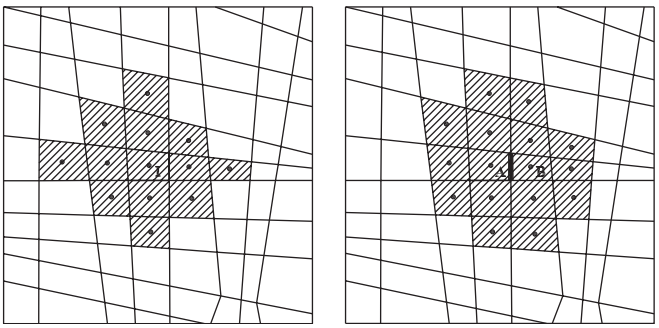

Figure 4. Stencil for interior points (left) and edges (right) for the MLS interpolation.

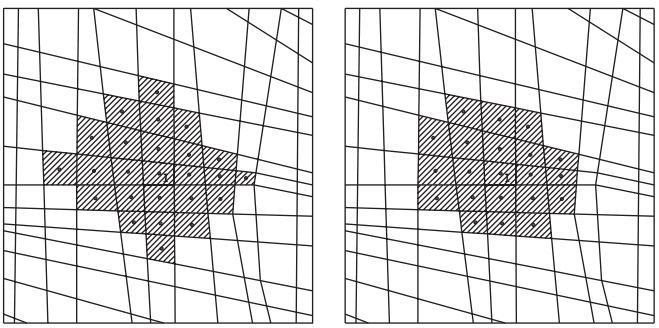

Figure 5. Full fourth-order FV–MLS stencil: Euler and Navier–Stokes (left) and elliptic problems (right).

4. DG METHODS

This section presents a brief introduction to DG methods for convection-dominated problems. For a deeper insight, we refer the reader to [11, 13, 17–20] among others.

The weak form of (1) is

$$\sum_e \left[\int_{\Omega e} \phi \frac{\partial \mathbf{U}_h}{\partial t} \, \mathrm{d}\Omega - \int_{\Omega e} \nabla \phi \cdot \mathbf{F}(\mathbf{U}, \nabla \mathbf{U}) \, \mathrm{d}\Omega + \oint_{\partial \Omega e} \phi \mathbf{n} \cdot \mathbf{F}(\mathbf{U}, \nabla \mathbf{U}) \right] = 0 \quad \forall \phi \tag{47}$$

where $\mathbf{F} = \mathscr{F} - \mathscr{F}^{\mathrm{V}}$. We have split the integral over the domain into the sum over the elements e.

It is important to notice that boundary integrals are referred to the element boundary, and we let the solution to be discontinuous. As a consequence, the flux function in the contour integral is not uniquely defined. As in the FV method, a numerical flux $\mathbf{h}(\mathbf{U}^+, \mathbf{U}^-, \mathbf{n})$ depending on the internal $(+)$ and the external $(-)$ interface values of the variable $\mathbf{U}$ and on the unit normal $\mathbf{n}$ is used. The numerical flux has another important function. It is the coupling between elements, which is missing by the discontinuous nature of the approximation. We can choose any of the numerical fluxes used in FV solvers as numerical flux functions for the convective part. Here, we have chosen the same one as in the FV–MLS method. The problem appears in the definition of numerical fluxes for the viscous surface integrals. In this case, if we use the common approach in FV schemes of central difference discretization for the evaluation of viscous fluxes, the solution

DOI: 10.1002/nme

seems to converge as we refine the mesh. However, it will be wrong and the error will have a component independent of grid size [17]. There are many possibilities to deal with viscous fluxes in a DG framework. Here, we will use the local discontinuous Galerkin (LDG) discretization [18].

We transform the second-order Navier–Stokes equations in a first-order equivalent system as

$$\mathbf{S} = \nabla \mathbf{U} \tag{48}$$

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathscr{F}(\mathbf{U}) - \nabla \cdot \mathscr{F}^{\mathrm{V}}(\mathbf{U}, \mathbf{S}) = 0 \tag{49}$$

We can write the weak form of both equations as

$$\int_{\Omega_e} \mathbf{S}_h \phi \, \mathrm{d}\Omega_e + \int_{\Omega_e} \mathbf{U}_h \nabla \phi(u) \, \mathrm{d}\Omega_e - \oint_{\Gamma_e} \mathbf{U}_h \cdot \mathbf{n} \phi \, \mathrm{d}\sigma = 0 \tag{50}$$

$$\int_{\Omega_e} \phi \frac{\partial \mathbf{U}_h}{\partial t} \, \mathrm{d}\Omega_e - \int_{\Omega_e} \mathscr{F}(\mathbf{U}_h) \nabla \phi \, \mathrm{d}\Omega_e + \int_{\Omega_e} \mathscr{F}^{\mathrm{V}}(\mathbf{U}_h, \mathbf{S}_h) \nabla \phi \, \mathrm{d}\Omega_e + \oint_{\Gamma_e} \mathscr{F}(\mathbf{U}_h) \cdot \mathbf{n} \phi \, \mathrm{d}\sigma - \oint_{\Gamma_e} \mathscr{F}^{\mathrm{V}}(\mathbf{U}_h, \mathbf{S}_h) \cdot \mathbf{n} \phi \, \mathrm{d}\sigma = 0 \tag{51}$$

Following Arnold *et al.* [19] we define on every interior edge the *average* $\{q\}$ and the *jump* $[q]$ operators as follows. Let e be an interior edge shared by two elements Ω_e^+ and Ω_e^-. We define the unit normal vectors $\mathbf{n}^+$ and $\mathbf{n}^-$ on e pointing exterior to Ω_e^+ and Ω_e^-. Then, we set the following:

For q scalar

$$\begin{aligned} \{q\} &= \tfrac{1}{2}(q^+ + q^-) \\ [q] &= q^+ \mathbf{n}^+ + q^- \mathbf{n}^- \end{aligned} \tag{52}$$

For $\boldsymbol{\varphi}$ vector

$$\begin{aligned} \{\boldsymbol{\varphi}\} &= \tfrac{1}{2}(\boldsymbol{\varphi}^+ + \boldsymbol{\varphi}^-) \\ [\boldsymbol{\varphi}] &= \boldsymbol{\varphi}^+ \cdot \mathbf{n}^+ + \boldsymbol{\varphi}^- \cdot \mathbf{n}^- \end{aligned} \tag{53}$$

For the first equation, the LDG scheme defines the following numerical flux. If the edge e is inside the domain Ω, then

$$\mathbf{U}_h \cdot \mathbf{n} = (\{\mathbf{U}\} + \boldsymbol{\beta} \cdot [\mathbf{U}]) \cdot \mathbf{n} \tag{54}$$

and if e lies on the boundary of Ω

$$\mathbf{U}_h \cdot \mathbf{n} = \begin{cases} \mathbf{g}_{\mathrm{D}} \cdot \mathbf{n} & \text{on } \Gamma_{\mathrm{D}} \\ \mathbf{U}^+ \cdot \mathbf{n} & \text{on } \Gamma_{\mathrm{N}} \end{cases} \tag{55}$$

where Γ_{D} and Γ_{N} are the Dirichlet and Neumann boundaries, and $\mathbf{g}_{\mathrm{D}}$ is the value of $\mathbf{U}$ on the Dirichlet boundary.

For the second equation, the inviscid flux is one of the commonly used in FV method. Numerical viscous flux for the edges inside the domain is the following:

$$\mathscr{F}^{\mathrm{V}} \cdot \mathbf{n} = (\{\mathscr{F}^{\mathrm{V}}\} - \boldsymbol{\beta} \cdot [\mathscr{F}^{\mathrm{V}}] - \alpha[\mathbf{U}]) \cdot \mathbf{n} \tag{56}$$

DOI: 10.1002/nme

and for the boundaries

$$\mathscr{F}^{\mathrm{V}} \cdot \mathbf{n} = \begin{cases} (\mathscr{F}^{\mathrm{V}^+} - \alpha(\mathbf{U}^+ - \mathbf{g}_{\mathrm{D}})\mathbf{n}) \cdot \mathbf{n} & \text{on } \Gamma_{\mathrm{D}} \\ \mathbf{g}_{\mathrm{N}} \cdot \mathbf{n} & \text{on } \Gamma_{\mathrm{N}} \end{cases} \tag{57}$$

where $\mathbf{g}_{\mathrm{N}}$ is the value of $\mathscr{F}^{\mathrm{V}}$ on the Neumann boundary.

Following [20] the value of $\alpha = 1$ has been chosen, and the definition of $\boldsymbol{\beta}$ on each edge e is as follows:

$$\boldsymbol{\beta} \cdot \mathbf{n} = \mathrm{sign}(\mathbf{v} \cdot \mathbf{n})/2 \tag{58}$$

where $\mathbf{v}$ is an arbitrary but fixed vector with nonzero components. This vector is taken as $\mathbf{v} = (1, 1)$.

After assembling all the elemental contributions, the system can be written as

$$\mathbf{M} \frac{\mathrm{d}\mathbf{U}_h}{\mathrm{d}t} + \mathbf{R}(\mathbf{U}_h) = 0 \tag{59}$$

where $\mathbf{M}$ is the mass matrix and $\mathbf{R}$ is the residual.

5. PIECEWISE POLYNOMIAL INTERPOLATION VERSUS MLS

Most of the existing high-order schemes are based on piecewise polynomial interpolation. Following this approach, higher-order accuracy is obtained by creating new dof inside each element/cell, which are used to construct an interpolating polynomial. From the perspective of pure interpolation, the reconstructed value at a point inside an element depends solely on the variables at the nodes inside that element. It is clear that for points located at edges and corners of the element, there is a bias in the direction of the incoming information, which necessarily has an impact on the global accuracy of the numerical scheme. Furthermore, even for moderately high-order polynomial interpolations the Runge phenomenon prevents the use of uniform nodal distributions. While an approximation that is discontinuous across interfaces seems somewhat natural for hyperbolic problems, it is rather inconvenient for the discretization of terms involving high-order derivatives.

On the other hand, MLS provide a general, pointwise centered, approximation framework, which may be used in order to construct both higher-order reconstructions in the sense of classical Godunov-type methods, and also accurate discretizations of elliptic-like terms. It is interesting at this point to examine the relative performance of piecewise polynomial and MLS approximations. First, this analysis will provide some insight into the efficiency of the schemes, in terms of how much information is extracted from a given grid. Second, the results of the present analysis may help understanding, at least in part, the relative accuracy in the numerical solution of boundary value problems achieved by techniques that use one method or the other.

An important remark is that, while one may construct polynomial expansions of arbitrarily high orders (provided that a careful choice of expansion basis and collocation points is made), MLS approximation are necessarily restricted to low or moderately high-order expansions. We believe, however, that the attainable orders within the MLS context (up to fourth–sixth order) are high enough for practical unstructured-grid computations.

This section presents a comparison in the approximation of a smooth function given by $u = \sin(2\pi x)\sin(2\pi y)$, defined on the square $[0, 1] \times [0, 1]$. The Cartesian grid for the piecewise polynomial interpolation has 13×13 cubic elements ($p = 3$). With the aim of maintaining the same

DOI: 10.1002/nme

spatial resolution, the MLS grid has been constructed by dividing each edge of the elements by three, and thus comprises 39×39 cells.

In order to compute the L_2 error norm, the function is evaluated at the same points, which are the 4×4 Gauss–Legendre quadrature nodes inside each cell of the MLS grid. As data we take the Gauss–Lobatto points for the piecewise polynomial interpolation and the centroid of each MLS cell for the MLS interpolation. We note that this implies a smaller number of known values for the MLS interpolation (Table I). MLS stencils as the one shown in Figure 4 for interior points have been used. For boundary points and its first neighbors, the stencil is comprised by the nearest 18 neighbors.

The results are shown in Figures 6 and 7. The distribution of the error is more homogeneous for the MLS interpolation and the maximum error is lower for both the variable and its gradient. In Table II we show the L_2 norm of the error. We note that although MLS interpolation uses a smaller number of known points than piecewise polynomial interpolation, L_2 errors for the variable are very similar, and for the gradient MLS achieves better results.

We further analyze the behavior of both interpolation methods with a non Cartesian distribution of the data. The grid for piecewise polynomial interpolation has 16×16 elements and MLS grid has been built as in the structured case. Both grids are shown in Figure 8. Table III shows the

Table I. Known values required for the interpolation.

	Piecewise polynomial interpolation	MLS
Known values	$13 \times 13 \times 16 = 2704$	$39 \times 39 = 1521$
Ghost nodes	—	$39 \times 4 = 156$
Corner ghost nodes	—	4
Total	2704	1681

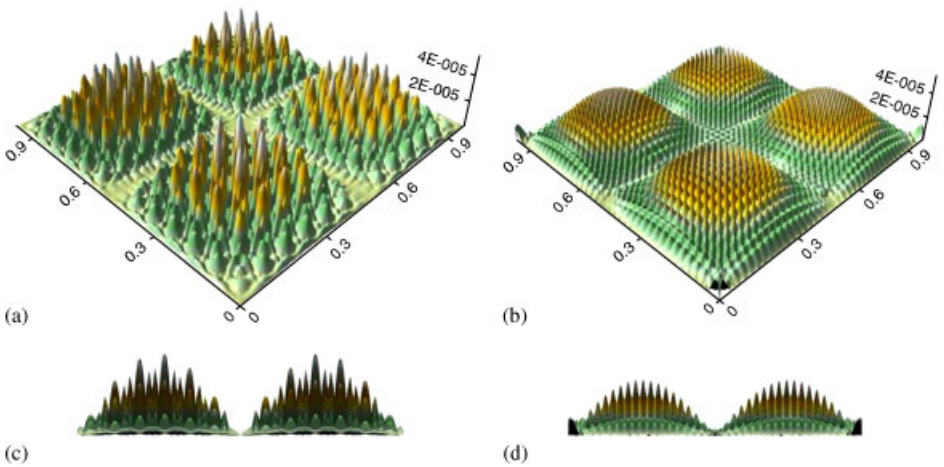

Figure 6. Absolute value of the error for piecewise polynomial interpolation (a, c) and MLS interpolation (b, d), for $u = \sin(2\pi x) \times \sin(2\pi y)$ on a Cartesian grid. a and b show a perspective of the domain, c and d show a front view.

DOI: 10.1002/nme

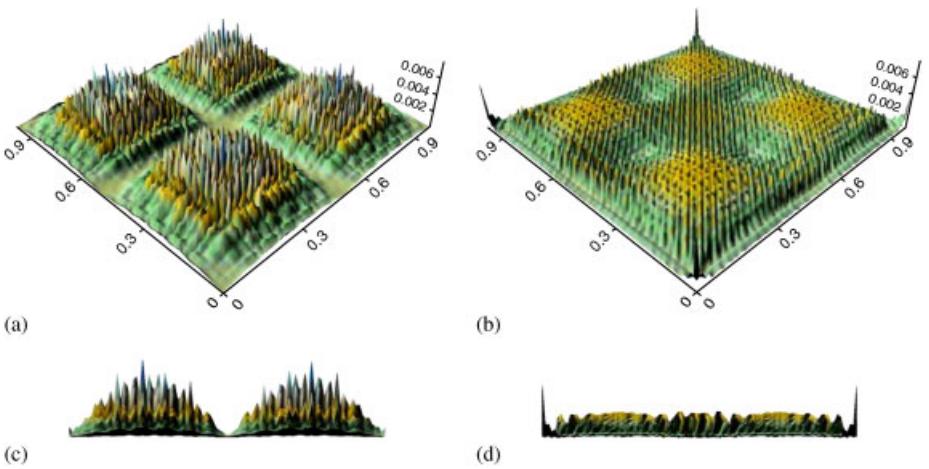

Figure 7. Absolute value of the error for piecewise polynomial interpolation (a, c) and MLS interpolation (b, d), for the gradient of $u = \sin(2\pi x) \times \sin(2\pi y)$ on a Cartesian grid. a and b show a perspective of the domain, c and d show a front view.

Table II. L_2 norm of the errors for the interpolation of $u = \sin(2\pi x) \times \sin(2\pi y)$ and its derivatives on a Cartesian grid.

	Piecewise polynomial interpolation	MLS
U	1.28E−05	1.57E−05
$\frac{\partial U}{\partial x}$	1.12E−03	8.31E−04
$\frac{\partial U}{\partial y}$	1.12E−03	8.31E−04
∇U	1.58E−03	1.18E−03

L_2 norm of the error, which again reveals a similar accuracy, although the MLS approximation achieves the best results in both the variable and derivative. Also, the distribution of the error is more uniform and the maximum errors are smaller. This is shown in Figures 9 and 10.

As pointed out in [9] for the one-dimensional case, the main drawback of MLS is the approximation near boundaries, where the stencils are necessarily biased. In two dimensional, corners are the most difficult points, due to the bad conditioning of moment matrix **M**. Nevertheless, the errors at the boundaries are of the same order of magnitude as those in the rest of the domain, which suggests that a good design of the stencil may overcome, or at least alleviate, the effect of boundaries. We believe that these better results of MLS in comparison with piecewise polynomial interpolation are due to its *local* and *centered* characteristics. These features are specially appealing for non-regular point distributions. It is precisely in this kind of environment where MLS more clearly outperforms piecewise polynomial interpolations. Also, the continuous nature of MLS interpolation across interfaces is a great advantage for the numerical solution of elliptic terms.

DOI: 10.1002/nme

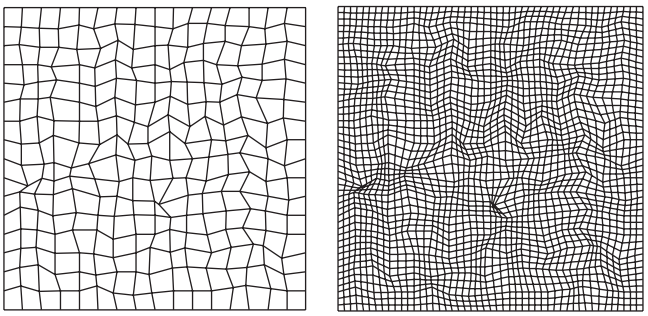

Figure 8. 16×16 non-structured piecewise polynomial interpolation grid (left) and 48×48 MLS grid (right), for $u = \sin(2\pi x) \times \sin(2\pi y)$. MLS grid was built by dividing by three each edge of each element of the grid on the left.

Table III. L_2 norm of the errors for the interpolation of $u = \sin(2\pi x) \times \sin(2\pi y)$ and its derivatives on an unstructured grid.

	Piecewise polynomial interpolation	MLS
U	1.17E−05	1.10E−05
$\frac{\partial U}{\partial x}$	1.27E−03	9.23E−04
$\frac{\partial U}{\partial y}$	1.03E−03	8.32E−04
∇U	1.63E−03	1.24E−03

6. ACCURACY TESTS

This section presents a comparative analysis, in terms of accuracy, of the FV and DG finite element methods introduced in the previous sections. We have preferred to establish a comparison in terms of *accuracy for the same grid resolution*, rather than comparisons of accuracy versus number of dof or accuracy versus cost, which, being perhaps more fair for the FV scheme, do not solve the question about the relative spatial accuracy of either scheme. By grid resolution we refer to h (the typical size of the control volumes in an FV grid), and H/p (the typical size of the DG elements divided by the polynomial order of the approximation). The accuracy in terms of cost depends, to a large extent, on the quality of the implementation. On the other hand, the conclusions of a comparison of accuracy per dof are somewhat incomplete without a rigorous comparison of the cost per dof. We will just remark that, at least in our implementation, the cost per dof is smaller in the case of the FV method.

The chosen test cases are representative for compressible flow simulations, and their scope is twofold:

- Discretization of hyperbolic equations: we present a convergence analysis for Ringleb flow problem.
- Discretization of viscous terms (elliptic problems): we present a convergence analysis for a simple Poisson model problem.

DOI: 10.1002/nme

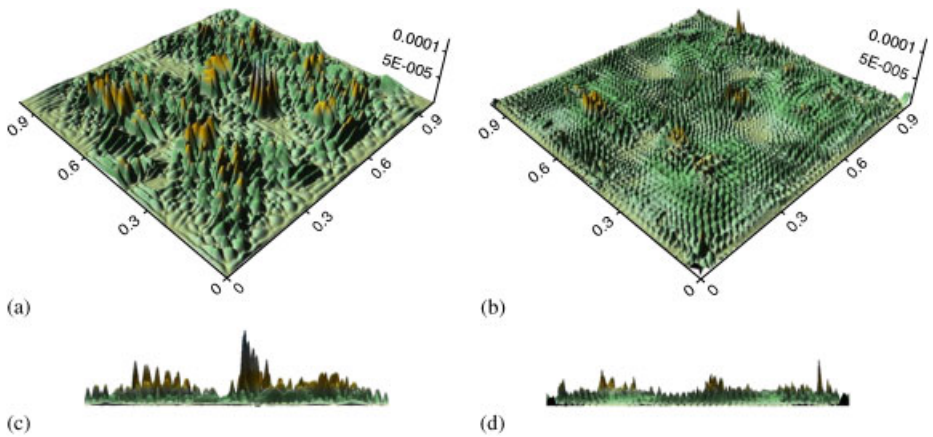

Figure 9. Absolute value of the error for piecewise polynomial interpolation (a, c) and MLS (b, d) interpolation, for $u = \sin(2\pi x) \times \sin(2\pi y)$ on an unstructured grid. a and b show a perspective of the domain, c and d show a front view.

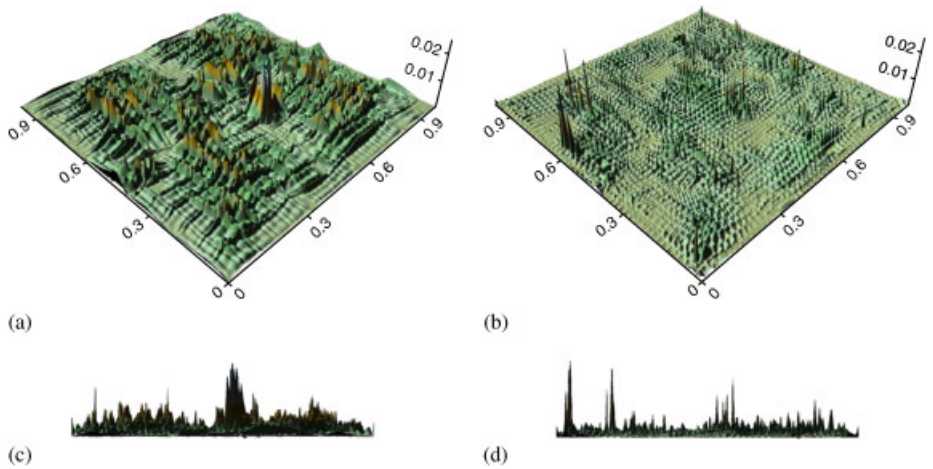

Figure 10. Absolute value of the error for piecewise polynomial interpolation (a, c) and MLS interpolation (b, d), for the gradient of $u = \sin(2\pi x) \times \sin(2\pi y)$ on an unstructured grid. a and b show a perspective of the domain, c and d show a front view.

6.1. Ringleb flow

In this first example, we consider the solution of the Ringleb flow problem. This problem is one of the few non-trivial problems for which a smooth analytical solution of the two-dimensional Euler equations is known. The analytical solution may be obtained using the hodograph transformation, see [21]. The transformation equations between the Cartesian variables (x, y) and the hodograph

DOI: 10.1002/nme

variables (V, ϑ) are

$$\Psi = \frac{1}{V}\sin\vartheta \tag{60}$$

$$c^2 = 1 - \frac{\gamma-1}{2}V^2 \tag{61}$$

$$J = \frac{1}{c} + \frac{1}{3c^3} + \frac{1}{5c^5} - \frac{1}{2}\log\frac{1+c}{1-c} \tag{62}$$

$$\rho = c^{(2/\gamma-1)} \tag{63}$$

$$x = \frac{1}{2\rho}\left[\frac{1}{V^2} - 2\Psi^2\right] + \frac{J}{2} \tag{64}$$

$$y = \pm\frac{\Psi}{\rho V}\cos\vartheta \tag{65}$$

6.1.1. Problem setup. The computational domain is taken inside the regular Ringleb's domain. It is the rectangle $[-1.15, -0.75]\times[0.15, 0.55]$. As boundary condition we set the $(-)$ state of the flux function to be the exact one, as obtained through (60)–(65).

The orders of convergence of both methods are checked on a series of successively refined non-isotropic grids, obtained by half dividing the distance between nodes from the coarsest grid. The grids corresponding to the three refinement levels are summarized in Table IV, and the coarsest one is shown on Figure 11. With this selection of the grid, we want to show the ability of the methods to work on non-isotropic grids, commonly used in CFD to solve boundary layers. Note that, in order to compare the errors at each refinement level, the grids have been chosen in such a way that the spatial resolution is the same or, when a total equivalence is not attainable, as close as possible. Thus, the results on an $n\times n$ FV grid are to be compared with those on an $(n/p)\times(n/p)$ DG grid. Note also that, due to node duplications, the DG grids involve more dof than the FV ones. For scalar problems the number of dof is n^2 for the FV method, whereas (for quadrilateral elements and tensor-product expansions) the DG grids use $n^2(p+1)^2/p^2$ dof. Thus, the ratio DG dof/FV dof is 4, 2.25 and 1.78 for $p=1$, $p=2$ and $p=3$, respectively. We also remark that with this configuration, the grids corresponding to the FV–MLS exhibit a larger skewness rate.

We have used the non-isotropic formulation presented in Section 3.5.1.

Table IV. Grids used for the Ringleb flow problem.

FV–MLS	DG	
All orders	$p=3$	$p=2$
15×60	5×20	8×30
30×120	10×40	15×60
60×240	20×60	30×120

DOI: 10.1002/nme

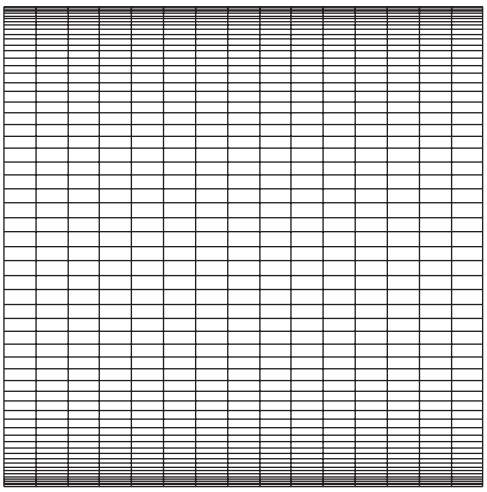

Figure 11. Ringleb flow. Non-isotropic grid with 15×60 elements for the FV scheme.

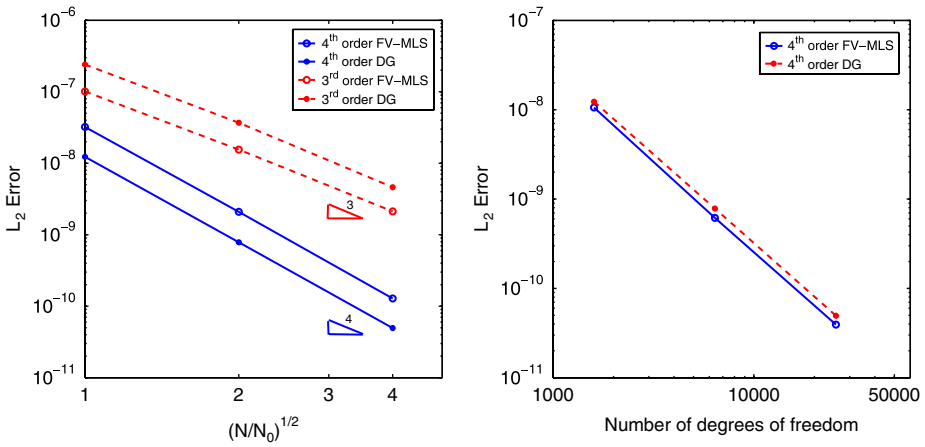

Figure 12. Ringleb flow. Convergence comparison for different reconstruction/polynomial orders (left) and accuracy against number of degrees of freedom for the fourth-order DG and the fourth-order FV schemes (right). N is the number of elements and subindex $_0$ refers to the coarsest grid.

6.1.2. Results. Figure 12 (left) presents a comparison of the convergence performance of the analyzed discretizations. The L_2 errors and numerical convergence rates are broken down in Tables V and VI.

As it can be seen from the above tables, both methods exhibit the correct convergence rates. The results for the fourth-order DG method present the smallest error. However, for the third-order method the FV–MLS gives the best results. This effect may be related to the fact that we use a polynomial cubic basis for the computation of the MLS shape functions, and the approximation of the third derivative may be less accurate than the approximation for the second one. We remark

DOI: 10.1002/nme

Table V. Convergence rates for the Ringleb flow and fourth-order discretizations.

DG fourth order				FV–MLS fourth order			
Grid	dofs	Error	Order	Grid	dofs	Error	Order
5×20	6400	1.23E−08	—	15×60	3600	3.22E−08	—
10×40	25600	7.84E−10	3.97	30×120	14400	2.09E−09	3.94
20×80	102400	4.94E−11	3.99	60×240	57600	1.28E−10	4.03

Table VI. Convergence rates for the Ringleb flow and third-order discretizations.

DG third order				FV–MLS third order			
Grid	dofs	Error	Order	Grid	dofs	Error	Order
8×30	8640	2.41E−07	—	15×60	3600	1.01E−07	—
15×60	32400	3.69E−08	2.84	30×120	14400	1.55E−08	2.698
30×120	129600	4.62E−09	2.99	60×240	57600	2.12E−09	2.946

again that the FV and DG grids have the same spatial resolution and thus the results for the fourth-order FV scheme have been obtained with 1.78 less dof.

In this context, Figure 12 (right) presents a comparison of the accuracy of the fourth-order FV scheme with the fourth-order DG scheme, in terms of accuracy against number of dof.

6.2. Two-dimensional Poisson

This example examines the ability of both methods for the numerical solution of elliptic equations, and is representative of the relative performance in the discretization of the viscous terms in the Navier–Stokes equations.

6.2.1. Problem setup. The boundary value problem is as follows:

Find $u:\Omega\subset\mathbb{R}^2\to\mathbb{R}$ such that

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega \\ u &= g_{\mathrm{D}} \quad \text{on } \Gamma_{\mathrm{D}} \end{aligned} \tag{66}$$

where the source term and the boundary conditions are chosen such that the analytical solution is $u(x,y)=\exp(\alpha\sin(Ax+By)+\beta\cos(Cx+Dy))$.

The parameters chosen for this problem are $A=5.1$, $B=-6.2$, $C=4.3$, $D=3.4$, $\alpha=0.1$, $\beta=0.3$. The problem is solved in $[0,1]\times[0,1]$. The number of cells/elements of the grids is the same as in the previous example. For the FV method, we have used the same kernel as in the Ringleb flow problem. Note that, also in the FV scheme, the stencil is based on the cell edges, since the gradients are computed directly at the quadrature points on the edges.

The magnitude of the errors and convergence rates are shown in Tables VII and VIII.

The expected convergence rates of $p+1$ for the variable u and $p+\frac{1}{2}$ for the gradient are recovered with the L_2 norm. With the L_∞ norm, it seems that the orders of convergence are $p+1$ for the primal variable u and p for the gradient, as the ones obtained in [20] for the LDG scheme. The order of convergence in the gradient is clearly superconvergent with the LDG scheme, leading

DOI: 10.1002/nme

Table VII. 2D Poisson equation: L_2 errors in the solution (u) and in the gradient (s) for the $p=3$ DG and FV methods.

	DG L_2 errors				FV–MLS L_2 errors			
h/ho	Error u	Rate	Error s	Rate	Error u	Rate	Error s	Rate
1	2.50E−04	—	1.47E−03	—	8.34E−05	—	2.60E−03	—
0.5	1.20E−05	4.38	1.03E−04	3.83	5.60E−06	3.90	2.40E−04	3.44
0.25	6.05E−07	4.31	8.02E−06	3.69	3.64E−07	3.94	2.15E−05	3.48
0.125	3.16E−08	4.26	6.61E−07	3.60	2.32E−08	3.97	1.91E−06	3.49

Table VIII. 2D Poisson equation: L_∞ errors in the solution (u) and in the gradient (s) for the $p=3$ DG and FV methods.

	DG L_∞ errors				FV–MLS L_∞ errors			
h/ho	Error u	Rate	Error s	Rate	Error u	Rate	Error s	Rate
1	3.11E−03	—	4.36E−02	—	2.6E−04	—	1.27E−02	—
0.5	1.84E−04	4.08	5.93E−03	2.88	2.01E−05	3.69	1.74E−03	2.87
0.25	1.10E−05	4.07	7.47E−04	2.99	1.34E−06	3.90	2.25E−04	2.95
0.125	7.88E−07	3.80	9.46E−05	2.98	8.56E−08	3.97	2.83E−05	2.99

to a smaller error in the gradient (not in the variable) in finest grids. This result for DG is expected only with Cartesian grids, in which superconvergence results have been reported in [20]. The convergence rate for general grids for LDG is expected to be of order p (see [22], for example). On the other hand, the convergence rate for MLS is expected to be $p+\frac{1}{2}$ on general, irregular grids (see [9]).

The FV discretization is more accurate in terms of the primary variable, both in the L_2 and L_∞ error norms. In terms of the gradient, on the other hand, DG is more accurate in the L_2 norm, while the FV method is more accurate in the L_∞ norm. The differences in the L_∞ norm, both in the primary variable and its gradient, are particularly pronounced. Furthermore, it is worth remarking again that the comparison is made in terms of grid resolution, and therefore the number of dof is significantly higher in the DG case.

6.3. Summary and comments on the accuracy tests

As it has been seen, the FV–MLS results are comparable with the DG results, or even more accurate in both the inviscid and viscous test cases. We believe that this is, at least in part, due to the nature of the interpolation process. In previous sections, it was explained that MLS can be defined as a 'centered' approximation, whereas DG uses an interpolation that is biased. It is clear that with MLS interpolation, the information 'comes' from all directions at every cell (except for the boundary cells, which require ghost points). Another feature that may have an influence on the accuracy gap is the fact that the FV scheme is conservative at a smaller scale than the DG scheme, which is also locally conservative but at the *element level*. On the other hand, the straightforward discretization of the viscous fluxes is one of the greatest advantages of the FV–MLS scheme for practical Navier–Stokes computations.

DOI: 10.1002/nme

7. REPRESENTATIVE SIMULATIONS

In this section, our comparative analysis is extended to more practical unstructured-grid simulations. As in the previous cases, cubic polynomials for the DG case and cubic reconstruction for the FV–MLS method are used in order to obtain an equivalent order of accuracy.

It is difficult to compare exactly these methods, because their way of reconstructing the variables is different. DG uses high-order polynomials to reconstruct the solution within the element. On the other side, FV–MLS approach uses a Taylor reconstruction for the inviscid fluxes with the derivatives calculated by using the information of neighbor nodes. As explained before, viscous fluxes are computed directly at the quadrature points. In these examples, both methods have been compared by *keeping the spatial resolution of the grid constant*. Thus, the FV–MLS grid is built from the DG grid, by constructing the FV–MLS elements from the nodes of the DG elements. DG nodes are placed at the Gauss–Lobatto points. In this way, from a DG element with 16 nodes (third-order polynomial reconstruction $p=3$), we obtain 9 MLS elements, as it is shown in Figure 13. Cell-centered nodes are used for the FV–MLS grid. We note that this comparison in terms of accuracy for the same grid resolution is unfair for the FV–MLS scheme, because it entails a smaller number of dofs for this scheme. Nevertheless, we prefer this approximation in order to try to answer the question about the relative spatial accuracy of either scheme.

7.1. Subsonic flow past an NACA 0012 profile

In this section, the inviscid subsonic flow past an NACA 0012 airfoil is computed. The Mach number is $M=0.63$ and the angle of attack is two degrees.

7.1.1. Problem setup. The computational (unstructured) grids are described in Table IX. One of the key features of a grid for this type of simulations is the number of cells/nodes on the surface of the profile. In the case of the DG grid, 48 $p=3$ elements have been placed to lie adjacent to the surface, which corresponds to 192 nodes over the surface of the airfoil. On the other hand, the FV–MLS grid has 144 cells on the surface. Both grids are shown in Figure 14. Note that the FV grid has been obtained by dividing the DG grid, in such a way that both grids have exactly the same spatial resolution. Freestream boundary conditions have been imposed at a distance of 30 chords.

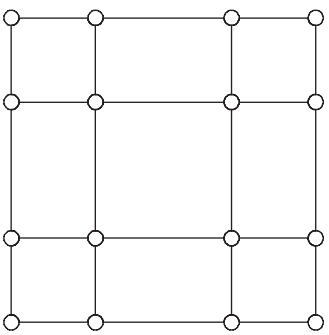

Figure 13. DG nodes (o) at Gauss–Lobatto points and FV–MLS elements from a single DG element.

DOI: 10.1002/nme

Table IX. Grid description for NACA0012 inviscid case.

	DG	FV–MLS
Elements	803	7227
One variable dofs	12848	7227
Total dofs	51392	28908
Elements over profile	48	144
Nodes over profile	192	144

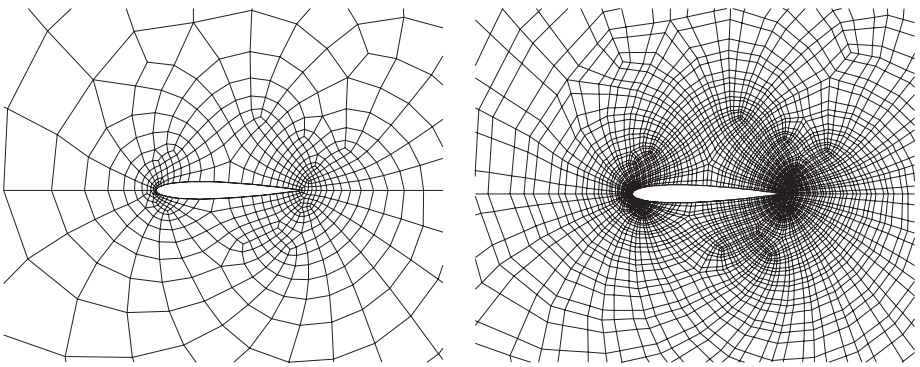

Figure 14. NACA 0012. Detail of the mesh for the DG case (left) and FV–MLS case (right).

7.1.2. Numerical aspects. The calculations have been carried on with $p=3$ polynomial approximations for the DG case and a cubic reconstruction in the FV–MLS case. As few elements are used to define the geometry in the DG case, curved elements are necessary if an accurate approximation of the geometry is desired. Moreover, Bassi and Rebay [23] showed that the DG method suffers from lack of accuracy in the case of a slip flow boundary condition, and that this effect could be minimized by using high-order approximations of the geometry. For the FV–MLS method we simply use a piecewise linear approximation of the boundaries. In this case the effect of curved boundaries would be appreciated over the directions of the normal vectors at the edge quadrature points, and would be required if the full accuracy of the scheme was to be exploited.

As for the MLS approximation, we use the cubic kernel defined by Equation (36) and, due to the irregularity of the grid, we use rectangular anisotropic weighting.

7.1.3. Results. Figure 15 shows the Mach number contours given by both methods, which are practically identical. Figure 16 presents the pressure coefficient (C_p) profiles, as well as the corresponding lift and drag coefficients, respectively C_L and C_D, whose reference values are $C_L=0.335$ and $C_D=0$. The differences between the DG and FV solutions are minimal at this grid resolution which is, on the other hand, typical of aerodynamic computations. The errors in the lift coefficients with respect to the reference ones may be attributed, to a certain extent, to the straightforward enforcement of the freestream boundary conditions.

Another important feature that is worth analyzing is the entropy generation. An inviscid flux of this kind is isentropic, so any generation of entropy in the solution is an evidence of dissipation introduced by the numerical scheme. Focusing on the entropy generation, we can see that at zones

DOI: 10.1002/nme

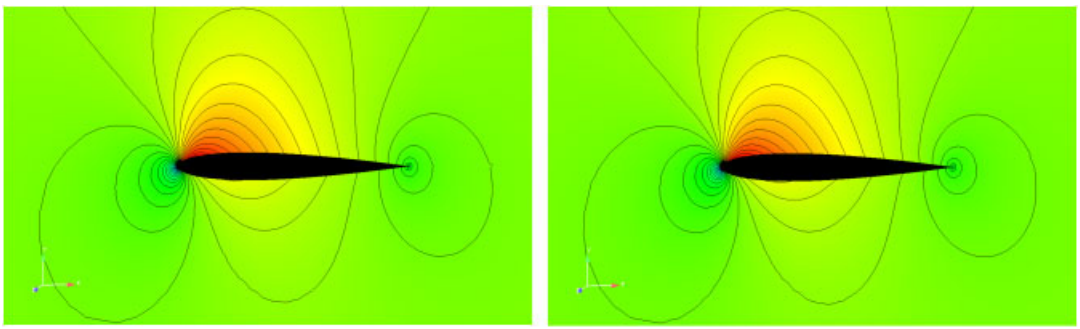

Figure 15. NACA 0012. Mach isolines for the DG (left) and FV–MLS (right) schemes.

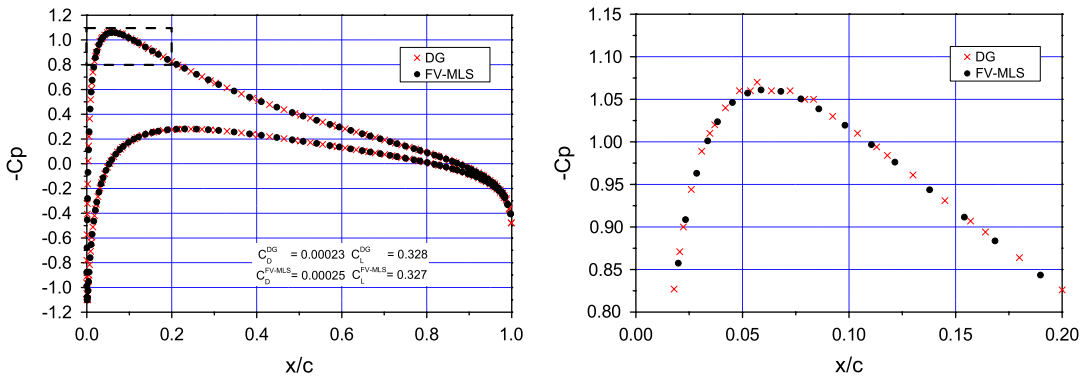

Figure 16. NACA 0012. Pressure coefficient and detail.

with higher curvature over the airfoil we have more generation of entropy with the FV–MLS method. The total amount of entropy generation is reduced by refining the mesh at the leading edge, so this problem may be related to the use of rectilinear cells with the FV–MLS method. Another source of numerical dissipation near the airfoil surface could be related to the MLS stencils in those areas, which may affect the accuracy of the approximation. In any case, as shown in Figure 17, the entropy production is quite small and its maximum value is only slightly larger than the DG one.

7.2. Subsonic viscous flow past an NACA 0012 profile

This example examines the behavior of both methods for full Navier–Stokes simulations. The Mach number is set as $M = 0.8$ and the angle of attack is 10 degrees, with a Reynolds number of $Re = 500$.

DOI: 10.1002/nme

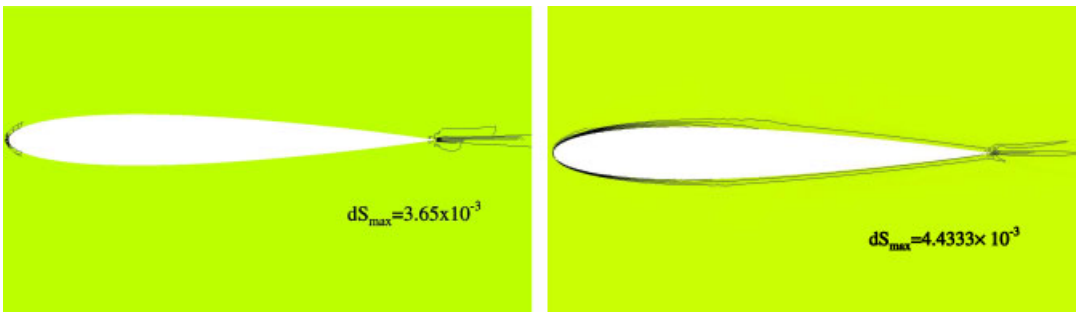

Figure 17. NACA 0012. Entropy generation DG (left) FV–MLS (right).

Table X. Grid description for NACA0012 viscous case. The viscous dofs are referred to those introduced by the auxiliary variable on the LDG discretization.

	DG	FV–MLS
Elements	2013	18 117
Inviscid dofs	128 832	72 468
Viscous dofs	193 248	0
Total dofs	322 080	72 468
Elements over profile	100	300
Nodes over profile	400	300

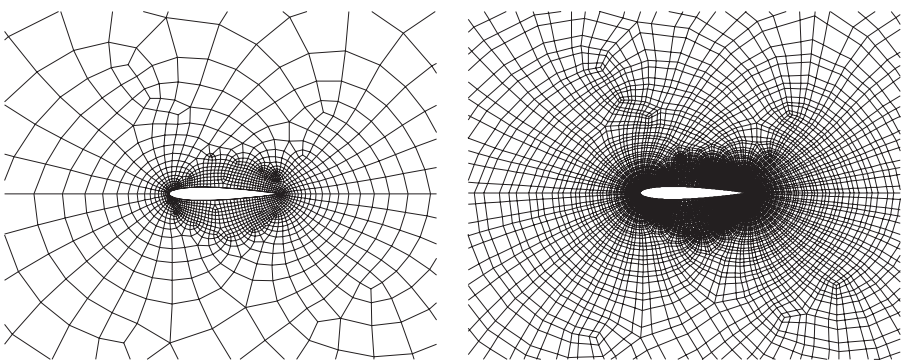

Figure 18. NACA 0012 $Re = 500$. Detail of the mesh for the DG case (left) and FV–MLS case (right).

7.2.1. Problem setup. The computational (unstructured) grids are described in Table X and Figure 18. The DG grid comprises 2013 $p = 3$ elements, with 100 elements placed over the profile. The FV–MLS mesh has the same spatial resolution, with a total of 18 117 control volumes.

As in the inviscid case, freestream boundary conditions have been placed at a distance of $30c$, and c is the cord of the profile.

DOI: 10.1002/nme

7.2.2. Numerical aspects. The calculations have been carried on with $p=3$ polynomial approximations for the DG case and a cubic reconstruction in the FV–MLS case. As in the Euler flow calculations, curved boundaries have been used for the DG case, and rectangular, anisotropic weighting for the MLS approximation.

The increase in the total number of dofs for the DG case is larger in this case than in Euler calculations. This is caused by the introduction of the auxiliary variable $\mathbf{S}$ in the transformation of the second-order system of PDEs into a first-order equivalent one (see Equations (48) and (49)). Most DG discretizations of the viscous terms are designed in such a way that these dof can be expressed in terms of the primal dof and thus eliminated, although they entail an additional workload that is not negligible.

7.2.3. Results. As observed in Figures 19–22, the results are very similar, virtually identical. As control variables, we have chosen the pressure coefficient and the skin friction coefficient. The results agree with those published in [24]. In this case, the main advantage of the use of FV–MLS is that at least the same accuracy than DG method is obtained, but with a number of dofs considerably smaller.

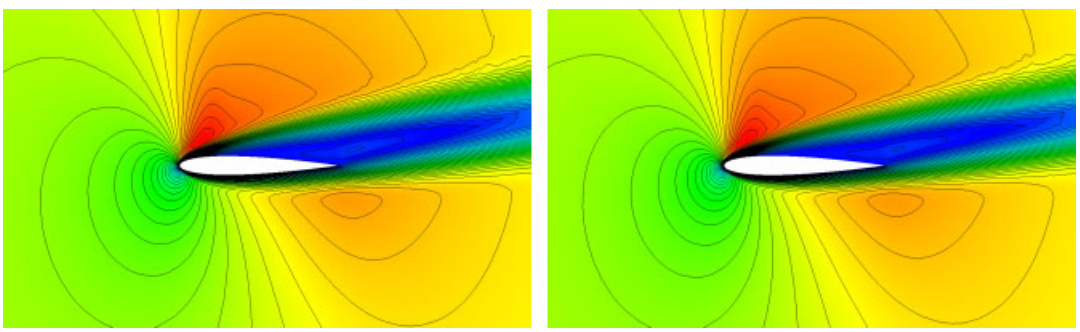

Figure 19. NACA 0012 $Re=500$. Mach isolines for the DG case (left) and FV–MLS case (right).

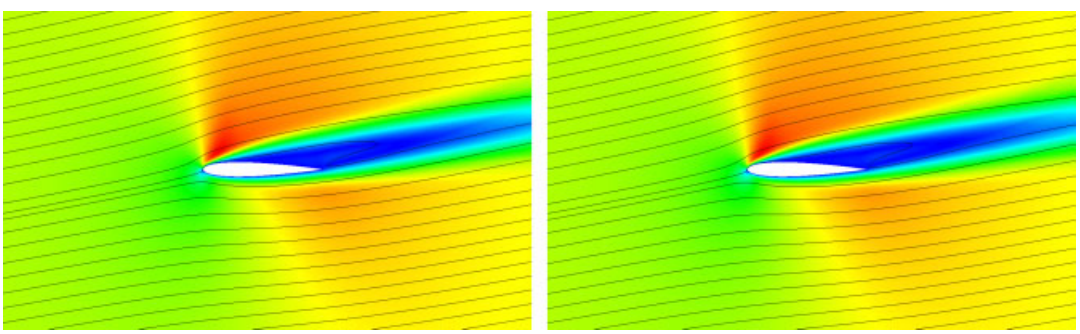

Figure 20. NACA 0012 $Re=500$. Streamlines for the DG case (left) and FV–MLS case (right).

DOI: 10.1002/nme

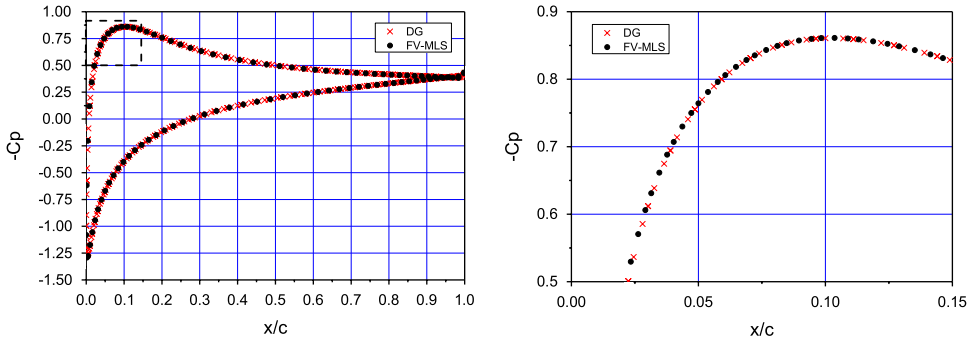

Figure 21. NACA 0012 $Re = 500$. Pressure coefficient and detail. Some FV–MLS points have been skipped for clarity.

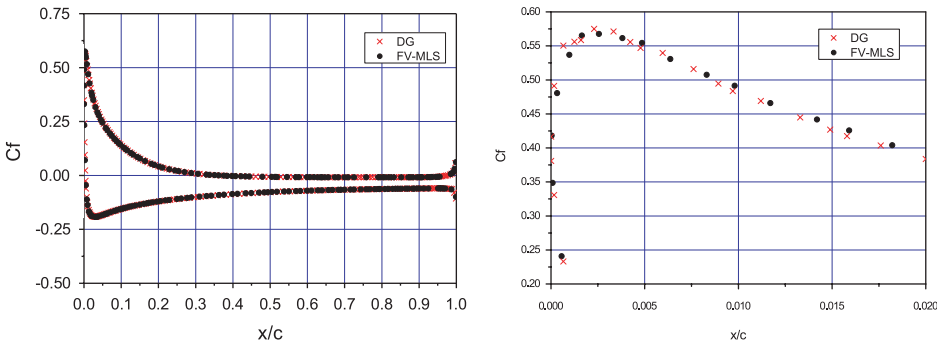

Figure 22. NACA 0012 $Re = 500$. Friction coefficient. Some FV–MLS points have been skipped for clarity.

8. CONCLUSIONS

In this paper, two high-order methods have been compared. The first one is a meshfree interpolation technique (namely, the MLS method) used in combination with FV discretization (FV–MLS) and the second one is the DG method.

We have shown the results given by both methods in many cases, for both inviscid and viscous flows. For these cases we have checked that the convergence of both methods is the expected one. Moreover, results for the FV–MLS method have an accuracy comparable with the solutions given by the DG method, at least with the use of third-order polynomial elements and for the same spatial resolution of the grids. In this case, the FV–MLS fourth-order method requires a number of dof that is roughly 1.78 times smaller than the DG method, for the resolution of the Euler equations. This is more remarkable for Navier–Stokes computations. In this case, the number of dof for DG is increased more than four times. Nevertheless, most DG discretizations of the viscous terms are designed in such a way that these dof can be expressed in terms of the primal dof and thus eliminated, although they entail an additional workload that is not negligible.

DOI: 10.1002/nme

We believe that one advantage of the FV–MLS method is its *local* and *centered* features and its *continuous* nature. Another feature that may have an influence is the fact that the FV scheme is conservative at a smaller scale than the DG scheme.

Finally, we would like to note that even though we have only shown cases with smooth solutions, FV–MLS method can be used in cases with non-smooth solutions. In this case, the behavior of the FV–MLS is similar to any FV solver with shock limiters. In the case of DG method, this is still an open problem.

ACKNOWLEDGEMENTS

This work has been partially supported by the *Ministerio de Educación y Ciencia* of the Spanish Government (#DPI2004-05156, #DPI2006-15275 and #DPI2007-61214) cofinanced with FEDER funds, the *Secretaría Xeral de I + D* of the *Xunta de Galicia* (Grants #PGIDIT05PXIC118002PN and # PGDIT06TAM11801PR) and the *University of A Coruña*. X. Nogueira was granted by the *Fundación Caixa Galicia*. Dr Gómez and Dr Cueto-Felgueroso gratefully acknowledge the support provided by *Ministerio de Educación y Ciencia* through the postdoctoral fellowships program. Dr Gómez and X. Nogueira also want to acknowledge the support provided by the European Commission through Marie Curie Actions.

REFERENCES

1. Godunov SK. A difference scheme for numerical solution of discontinuous solution of hydrodynamic equations. *Matematicheskii Sbornik* 1969; **47**(3):271–306 (Translated in US Joint Publications Research Service, JPRS 7226).
2. Harten A, Lax P, Van Leer B. On upstream differencing and Godunov-type schemes for hyperbolic conservation laws. *SIAM Review* 1983; **25**:35–61.
3. Barth TJ. *Aspects of Unstructured Grids and Finite-volume Solvers for the Euler and Navier–Stokes Equations*. VKI Lecture Series Monographs: Computational Fluid Dynamics, 25th Computational Fluid Dynamics, VKI LS 1994-05, 1995.
4. Toro EF. *Riemann Solvers and Numerical Methods for Fluid Dynamics. A Practical Introduction* (2nd edn). Springer: Berlin, 1999.
5. Gingold RA, Monaghan JJ. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society* 1977; **181**:378.
6. Eldredge JD, Colonius T, Leonard A. A vortex particle method for two-dimensional compressible flow. *Journal of Computational Physics* 2002; **179**:371–399.
7. Cueto-Felgueroso L, Colominas I, Fe J, Navarrina F, Casteleiro M. High order finite volume schemes on unstructured grids using moving least squares reconstruction. Application to shallow waters dynamics. *International Journal for Numerical Methods in Engineering* 2006; **65**:295–331.
8. Cueto-Felgueroso L, Colominas I, Nogueira X, Navarrina F, Casteleiro M. Finite volume solvers and moving least-squares approximations for the compressible Navier–Stokes equations on unstructured grids. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**:4712–4736.
9. Cueto-Felgueroso L, Colominas I. High-order finite volume methods and multiresolution reproducing kernels. *Archives of Computational Methods in Engineering* 2008; **15**(2):185–228.
10. Reed WH, Hill TR. Triangular mesh methods for the neutron transport equation. *Technical Report LA-UR-73-479*, Los Alamos Scientific Laboratory, Los Alamos, NM, 1973.
11. Bassi F, Rebay S. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. *Journal of Computational Physics* 1997; **131**:267–279.
12. Bassi F, Crivellini A, Rebay S, Savini M. Discontinuous Galerkin solution of the Reynolds-averaged Navier–Stokes and $k-\omega$ turbulence model equations. *Computers and Fluids* 2005; **34**:507–540.
13. Baumann CE, Oden JT. A discontinuous *hp* finite element method for the Euler and Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 2004; **45**:1083–1106.
14. Dolejsi V. On the discontinuous Galerkin method for the numerical solution of the Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 1999; **31**:79–95.

DOI: 10.1002/nme

15. Gómez H, Colominas I, Navarrina F, Casteleiro M. A finite element formulation for a convection–diffusion equation based on Cattaneo's law. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**(9–12):1757–1766. DOI: 10.1016/j.cma.2006.09.016.
16. Lancaster P, Salkauskas K. Surfaces generated by moving least squares methods. *Mathematics of Computation* 1981; **155**:141–158.
17. Zhang M, Shu C-W. An analysis of three different formulations of the discontinuous Galerkin method for diffusion equations. *Mathematical Models and Methods in Applied Sciences* 2003; **13**(3):395–413.
18. Cockburn B, Shu C-W. The local discontinuous Galerkin finite element method for convection–diffusion systems. *SIAM Journal on Numerical Analysis* 1998; **35**:2440–2463.
19. Arnold DN, Brezzi F, Cockburn B, Marini D. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis* 1998; **39**(5):1749–1779.
20. Cockburn B, Kanschat G, Perugia I, Schtzau D. Superconvergence of the local discontinuous Galerkin method for elliptic problems on cartesian grids. *SIAM Journal on Numerical Analysis* 2001; **39**(1):264–285.
21. Chiocchia G. Exact solutions to transonic and supersonic flows. *Technical Report AR-211*, *AGARD*, 1985.
22. Persson P, Peraire J. An efficient low memory implicit DG algorithm for time dependent problems. *44th AIAA Aerospace Sciences Meeting*, *AIAA-2006-0113*, Reno, Nevada, 2006.
23. Bassi F, Rebay S. High-order accurate discontinuous finite element solution of the 2D Euler equations. *Journal of Computational Physics* 1997; **138**:251–285.
24. Bristeau MO, Glowinski R, Periaux J, Viviand H (eds). *Numerical Simulation of Compressible Navier–Stokes Flows*. Notes on Numerical Fluids Mechanics, vol. 18(2). Friedrich Vieweg und Sohn: Brunswick, Federal Republic of Germany, 1987.