

Benchmark solutions

A very high-order finite volume method based on weighted least squares for elliptic operators on polyhedral unstructured grids

Artur G.R. Vasconcelos, Duarte M.S. Albuquerque*, José C.F. Pereira

LAETA, IDMEC, Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

ARTICLE INFO

Article history:

Received 9 August 2017

Revised 29 January 2019

Accepted 5 February 2019

Available online 10 February 2019

Keywords:

Eight-order scheme

Finite volume method (FVM)

Weighted least-squares (WLS)

Unstructured polyhedral grids

Poisson equation

Efficiency study of numerical schemes

ABSTRACT

A very high-order finite volume method is proposed for the solution of the Poisson equation on unstructured grids based on the weighted least-squares method. The new method consists in an up to eight-order accurate face centered reconstruction for the integration of the diffusive fluxes. It uses a new stencil extension algorithm that maintains the local high-order near the boundaries of the computational domain.

The weight function used by the scheme was optimized according to the matrix condition number, convergence order and error magnitude. From this study a new weight function is proposed for this type of reconstruction schemes which achieves the theoretical convergence order for Cartesian, triangular, polyhedral and hybrid grids. An extension for the Neumann boundary type conditions is also shown. A grid quality study, based on the non-orthogonality angle and volume ratio, proves that the scheme's convergence order is not affected by these parameters.

The efficiency criteria, defined by the required memory and solver-run time, indicate that the eight-order scheme is advantageous over the lower order ones, additionally it is demonstrated that the polyhedral grid yields more efficient solutions than the ones obtained with Cartesian or triangular grids.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

The numerical solution of transport phenomena (fluid flow, heat and mass transfer, etc.) in complex geometrical configurations is a subject of continuous development regarding accuracy, robustness and efficiency. The geometrical complexity can be handled with different grid topologies and their issues of robustness and efficiency are relevant for industrial applications. Nowadays the focus is to obtain an accurate solution with a reduced computational cost, which will allow to solve more complex problems and faster. Higher-order (higher than second-order) computation and particularly very high-order (higher than fourth-order) are a demanding issue, motivated by a potential reduction of computational cost for complex computational fluid dynamics (CFD) problems, see e.g. Vermeire et al., [1], this has been showed for structured grids by Zingg et al., [2], and Lipnikov and Manzini, [3], has showed this for unstructured ones.

High-order accurate methods for unstructured grids have historically been focused on hyperbolic equations, see e.g. Lê et al., [4], and Drikakis et al., [5], particularly on advective fluxes. Barth and Frederickson, [6], are among the pioneers on the development of a high-order Finite Volume Methods (FVM) using a quadratic polynomial that is applied for the resolution of the Euler equations. The coupling of Euler system with viscous terms, which requires diffusive schemes was achieved by Ollivier-Gooch et al., [7–10], Ivan and Groth, [11], Felgueroso et al., [12], and Toro and Hidalgo, [13]. Essentially Non-Oscillatory (ENO) fourth order schemes and its variants like WENO or CENO, are very popular for shock resolution, see the works of Ollivier-Gooch, [14], Ivan and Groth, [11], Hu and Shu, [15] and Friedrich, [16].

In finite difference (FD) many high-order accurate methods have been successfully developed for structured grids, see e.g. Visbal and Gaitonde, [17], that have advantages in both terms of computational cost and efficiency for boundary layer simulations, see Deng et al., [18]. Recently, this method was also extended to problems with strong discontinuities and curved boundaries, e.g. Baeza et al. [19]. Still in FD framework the Padé compact schemes are well established for structured grids, see for the example the fourth order scheme of Pereira et al., [20]. The spectral finite difference scheme has been in development, see Liu et al., [21], and

* Corresponding author.

E-mail addresses: artur.vasconcelos@tecnico.ulisboa.pt (A.G.R. Vasconcelos), duartealbuquerque@tecnico.ulisboa.pt (D.M.S. Albuquerque), jcpereira@tecnico.ulisboa.pt (J.C.F. Pereira).

Acronyms

CFD	Computational Fluid Dynamics
DG	Discontinuous Galerking
ENO	Essentially Non-Oscillatory
FD	Finite Difference
FEM	Finite Element Method
FV	Finite Volume
FVM	Finite Volume Method
CV	Control Volume
FLS	Face Least-Squares
FLS2	Face Least Squares 2nd Order Scheme
FLS4	Face Least Squares 4th Order Scheme
FLS6	Face Least Squares 6th Order Scheme
FLS8	Face Least Squares 8th Order Scheme
GQ	Gauss Quadrature
LS	Least Squares
NNZ	Number of Non Zero
SRT	Solver-Run Time
PDE	Partial Differential Equations
WLS	Weighted Least-Squares

Wang et al., [22], another type of schemes with spectral like resolution is the Fourier pseudo-spectral schemes that have been extended to the immersed boundary method, Mariano et al., [23], which can be applied for curved boundaries and bodies.

The large family of high-order finite element methods (FEM) is also well established, see the fourth order accurate works of Marques et al., [24] and the formulations up to tenth order of Bériot et al., [25]. Another popular method is the Discontinuous Galerking (DG) methods which may be found in Shu et al., [26,27], Arnold et al., [28], Dumbser et al., [29], Nogueira et al., [30], and J. Jaśkowiec et al., [31]. Finally an energy stable high-order scheme is applied for both DG and the spectral difference methods, see e.g. Vincent et al., [32].

The development of high-order methods is not limited to hyperbolic equations and in the last years the resolution of parabolic and elliptic problems in unstructured grids has made significant progress, see e.g. Boularas et al., [33], Bertolazzi et al., [34], Droniou, [35]. The range of possible applications varies from Poisson problems, see Batty, [36], heat transfer problems, see e.g. Chantasiriwan, [37], diffusion equations with variable coefficients, see Zhai, [38], or discontinuous coefficients, see e.g. Clain et al., [39]. Diffusive fluxes are also required in the framework of incompressible Navier-Stokes equations and it is important for the computation of velocity and pressure updates by SIMPLE or PISO projection algorithms, see e.g. Nogueira et al., [40], and Guermond et al., [41].

In the last decades, several polynomial reconstruction techniques applied to FVM have been highlighted as for example the fourth-order methods of Ollivier-Gooch et al., [42–44], Cueto-Felgueroso et al., [45–49], and Nogueira et al., [40,50], also sixth-order results have been reported by Clain et al., [51–53]. The objective of this work is to extend the weighted least-squares (WLS) method to very high-order schemes and polyhedral unstructured grids.

The main differences between this approach and the others ones are the point value interpolation and the reconstructions being centered at the face centroid. This new approach does not require an interpolation for the face's fluxes which is used by the cell centered reconstructions. Another advantage in the development of these schemes is the possibility of the resolution of Partial Differential Equations (PDE) with high-order derivatives (higher than second) which is not possible with the classic second order accurate schemes, see e.g. Guo and Lu, [54].

The outline of this paper is as follows: in Section 2 the numerical method is explained, with a special focus on the WLS implicit formulation and on the stencil selection technique for both interior and boundary faces. This will be followed by Section 3 where the selection of the weight function for WLS method is explained and justified by considering three factors: matrix condition number, convergence order and error magnitude. Afterwards, the results section presents the performed numerical tests to demonstrate the effective convergence order of this new very high-order scheme, independently of grid topology. A case with Neumann boundary conditions is also shown. Furthermore, a grid quality study for two parameters, non-orthogonality angle and volume ratio is performed. Finally, an efficiency study based on the memory required and solver-run time (SRT) shows that the eight-order scheme is advantageous over the other lower-order schemes and that the polyhedral grids are superior to the other grid types. The paper ends with summary conclusions.

2. Numerical method

This section describes the FVM and the proposed WLS scheme for diffusive term discretization of a transport equation. Details are provided on the stencil selection and how the global coefficient matrix is constructed according to the boundary condition.

2.1. Poisson equation based on finite volume method

The Poisson equation in the conservative form is given by:

$$\nabla \cdot \nabla \phi = \varphi_\phi, \quad (1)$$

where ϕ is the transported variable and φ_ϕ is the source term.

The Finite Volume (FV) discretization is based on the integral form of the conservative law (1) over a Control Volume (CV), resulting in the following expression:

$$\int_{CV} \nabla \cdot \nabla \phi dV = \int_{CV} \varphi_\phi dV. \quad (2)$$

Applying the Gauss Divergence Theorem, the expression (2) can be written as:

$$\int_{CS} \nabla \phi \cdot d\mathbf{S} = \int_{CV} \varphi_\phi dV, \quad (3)$$

where $d\mathbf{S}$ takes into account the unitary outward normal vector and surface area of the cell's faces.

Fig. 1 shows the typical division of a domain in non-overlapping volumes (cells). The several triangular, quadrilateral and polyhedral cells have the information of the transported variable, ϕ , at each cell centroid.

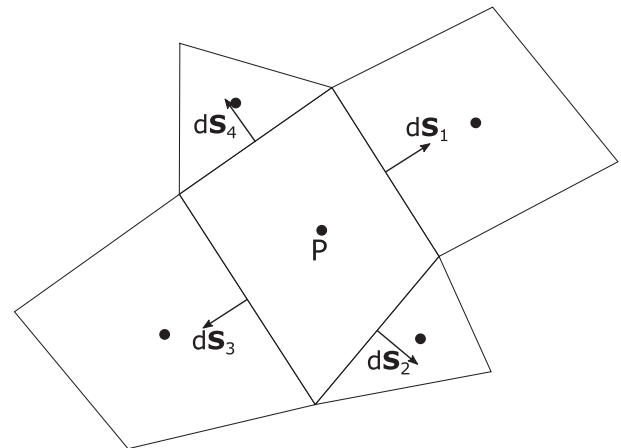


Fig. 1. Cell-centered FV discretization over the cell P (• - Cell Centroid).

Discretization of Eq. (1) with a FVM cell centered approach requires the calculation of the face gradient, $\nabla\phi_f$:

$$\sum_{f \in \mathcal{F}(P)} \sum_{g \in \mathcal{G}(f)} w_g \nabla\phi_g \cdot \mathbf{S}_f = \int_{CV} \phi_\phi dV, \quad (4)$$

where $\mathcal{F}(P)$ is the set of faces of cell P , $\mathcal{G}(f)$ is the set of Gauss points of the face f , \mathbf{S}_f is the face normal vector, and w_G is the weight of Gauss–Legendre Quadrature.

2.2. Least squares approach

2.2.1. Polynomial reconstruction

To compute the solution and its gradients at the integration points, a reconstruction of the unknown primitive variable is performed to obtain a polynomial expansion about the control volumes face with the derived accuracy. This is achieved by an expansion of a p th complete polynomial, centered on the face where the reconstruction is made:

$$\phi_f^R(x, y) = C_1 + C_2(x - x_f) + C_3(y - y_f) + C_4(x - x_f)^2 + C_5(x - x_f)(y - y_f) + C_6(y - y_f)^2 + \dots \quad (5)$$

Expression (5) can be written in a more compact form, a vectorial one, as:

$$\phi_f^R(\mathbf{x}) = \mathbf{d}_f(\mathbf{x})\mathbf{c}_f. \quad (6)$$

The subscript f refers that the reconstruction is made at the face f centroid and $\mathbf{d}_f(\mathbf{x}) = [1, (x - x_f), (y - y_f), (x - x_f)^2, (x - x_f)(y - y_f), (y - y_f)^2, \dots]$, $\mathbf{x}_f = (x_f, y_f)$ is the face centroid coordinates vector, $\mathbf{x} = (x, y)$ is the coordinates vector of a point used for the reconstruction and $\mathbf{c}_f = [C_1, C_2, C_3, C_4, C_5, C_6, \dots]^T$ is the reconstruction vector constants. The above expressions are valid for a p th order complete polynomial and a reconstruction will be made for each face of the domain. This will be called of Face Least Squares (FLS) approach, and the number of terms of the expansion is given by $(p+1)(p+2)/2$. [50]. Table 1 lists the number of terms of the expansion for each polynomial used in this work.

The order of accuracy of the numerical scheme is $p+1$, consequently the linear reconstruction will be 2nd order accurate, the cubic reconstruction will be 4th order accurate, the 5th polynomial will have 6th order accurate and finally the 7th polynomial will be 8th order accurate. The numerical schemes will be called of FLS($p+1$) according to the global order of the implemented method.

2.2.2. General approach

Eq. (6) results in an overdetermined system of linear equations with the form:

$$\mathbf{D}_f \mathbf{c}_f = \phi_s, \quad (7)$$

where \mathbf{D}_f is a combination of $\mathbf{d}_f(\mathbf{x})$ for every point of the reconstruction which results in a matrix with $n_s \times n_{coefs}$ entries. The \mathbf{c}_f is a column vector with n_{coefs} entries and ϕ_s is a column vector with n_s entries, where n_{coefs} is the number of constants of the p th polynomial (unknowns), listed in Table 1, and n_s is the number of

interpolated reconstruction points of the face's stencil that will be explained in detail in next Sections 2.2.3. and 2.2.4.

Since $n_s > n_{coefs}$, the WLS technique is needed to minimize the weighted residual of the problem which is defined as:

$$\min(\mathbf{r}^T \mathbf{W}_{LS_f} \mathbf{r}) = \min([\phi_s - \mathbf{D}_f \mathbf{c}_f]^T \mathbf{W}_{LS_f} [\phi_s - \mathbf{D}_f \mathbf{c}_f]), \quad (8)$$

where $\mathbf{r} = \phi_s - \mathbf{D}_f \mathbf{c}_f$ is the residual of the polynomial fitting and $\mathbf{W}_{LS_f} = \text{diag}(w_{LS1}, \dots, w_{LSn_s})$ is a diagonal matrix where each component is the result of the weighted function at a cell from the stencil.

The minimum of the weighted residual can be obtained by differentiating $\mathbf{r}^T \mathbf{W}_{LS_f} \mathbf{r}$ with respect to \mathbf{c}_f and setting it equal to zero, which yields:

$$\mathbf{D}_f^T \mathbf{W}_{LS_f} \mathbf{D}_f \mathbf{c}_f = \mathbf{D}_f^T \mathbf{W}_{LS_f} \phi_s. \quad (9)$$

For clarity the following matrix product will be defined as $\mathbf{D}_{w_f} = \mathbf{W}_{LS_f} \mathbf{D}_f$ and the Eq. (9) is written as:

$$\mathbf{D}_{w_f}^T \mathbf{D}_f \mathbf{c}_f = \mathbf{D}_{w_f}^T \phi_s. \quad (10)$$

Resulting in a matrix with the same number of entries as matrix \mathbf{D}_f , ($n_s \times n_{coefs}$), and vector \mathbf{c} contains the unknowns of the LS problem. It is fundamental to ensure that the columns of \mathbf{D} are linearly independent. The solution of this problem can be expressed as:

$$\mathbf{c}_f = (\mathbf{D}_{w_f}^T \mathbf{D}_f)^{-1} \mathbf{D}_{w_f}^T \phi_s, \quad (11)$$

with the pseudo inverse matrix defined as $\mathbf{P}_f = (\mathbf{D}_{w_f}^T \mathbf{D}_f)^{-1} \mathbf{D}_{w_f}^T$, the constants of the polynomial reconstruction are computed by:

$$\mathbf{c}_f = \mathbf{P}_f \phi_s. \quad (12)$$

$$\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ \vdots \\ C_{n_{coefs}} \end{bmatrix} = \begin{bmatrix} P_{1,1} & P_{1,2} & P_{1,3} & \dots & P_{1,n_s} \\ P_{2,1} & P_{2,2} & P_{2,3} & \dots & P_{2,n_s} \\ P_{3,1} & P_{3,2} & P_{3,3} & \dots & P_{3,n_s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_{n_{coefs},1} & P_{n_{coefs},2} & P_{n_{coefs},3} & \dots & P_{n_{coefs},n_s} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_{n_s} \end{bmatrix}. \quad (13)$$

After the \mathbf{P} matrix has been determined for every face of the domain, it is possible to proceed with the construction of a linear system for the Poisson problem because the constants of the reconstruction are only dependent of the ϕ_s values.

Afterwards is it possible to compute the diffusive flux by derivation of the reconstructed polynomial, which results in:

$$\nabla\phi_f^R(\mathbf{x}) = (\nabla\mathbf{d}_f(\mathbf{x}))\mathbf{c}_f, \quad (14)$$

combining the above equation with Eq. (12) yields:

$$\nabla\phi_f^R(\mathbf{x}) = (\nabla\mathbf{d}_f(\mathbf{x}))(\mathbf{P}_f \phi_s), \quad (15)$$

where $\nabla\mathbf{d}$ in vector form:

$$\nabla\mathbf{d}_f(x, y) = \begin{bmatrix} 0 & 1 & 0 & 2(x - x_f) & (y - y_f) & 0 & \dots \\ 0 & 0 & 1 & 0 & (x - x_f) & 2(y - y_f) & \dots \end{bmatrix}, \quad (16)$$

introducing the matrix \mathbf{T} as:

$$\mathbf{T}_f(\mathbf{x}) = (\nabla\mathbf{d}_f(\mathbf{x}))\mathbf{P}_f \quad (17)$$

Table 1
Number of terms of the Taylor expansion required for a p th order polynomial at two dimensional (2D) cases.

p th Order polynomial	1	3	5	7
Number of terms	3	10	21	36

with $2 \times n_s$ entries, Eq. (15) can be expressed as $\nabla \phi_f^R(\mathbf{x}) = \mathbf{T}_f(\mathbf{x})\phi_s$ and with the following matrix form:

$$\begin{bmatrix} \frac{\partial \phi_f^R}{\partial x} \\ \frac{\partial \phi_f^R}{\partial y} \end{bmatrix} = \begin{bmatrix} T_{x,1} & T_{x,2} & T_{x,3} & \dots & T_{x,n_s} \\ T_{y,1} & T_{y,2} & T_{y,3} & \dots & T_{y,n_s} \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_{n_s} \end{bmatrix}. \quad (18)$$

The integration of the fluxes through the faces of the CV in Eq. (4) requires the use of the corresponding Gauss–Legendre Quadrature with the same order of the reconstructed polynomial [55]. Consequently the left side of expression for the Poisson equation is given by:

$$\sum_{f \in \mathcal{F}(P)} \nabla \phi_f \cdot \mathbf{S}_f = \sum_{f \in \mathcal{F}(P)} \left(\sum_{g \in \mathcal{G}(f)} w_{Gg} \mathbf{T}_f(\mathbf{x}_g) \phi_s \right) \cdot \mathbf{S}_f, \quad (19)$$

where $\mathcal{G}(f)$ is the set of Gauss–Legendre points of face f , w_{Gg} is the weight and \mathbf{x}_g are the coordinates of each Gauss–Legendre point g . From the above expression, the information for the construction of the global matrix is available, and each entry of the matrix \mathbf{A} can be written in the following form:

$$\mathbf{A}_{ij} = \sum_{f \in \mathcal{F}(i)} \left(\sum_{g \in \mathcal{G}(f)} w_{Gg} \mathbf{t}_{fj}(\mathbf{x}_g) \right) \cdot \mathbf{S}_f, \quad (20)$$

where i is the CV where the discretization is applied, j is the neighbour cell that is used for the face f reconstruction and the vector \mathbf{t}_{fj} is the column of the face matrix \mathbf{T}_f (presented in expression (17)). Each line of the global matrix \mathbf{A}_{ij} corresponds to the discretization of the cell i and each column corresponds to the contribution from the cell j to the discretization. The global matrix is solved with the BIGSTAB solver and the ILU(0) preconditioner. Other details related with the matrix of least-squares systems are described in the following works [56–59].

2.2.3. Stencil selection algorithm for interior faces

The stencil is a set of points used for the polynomial reconstruction and its choice is important to achieve the scheme's accuracy. The stencil selection for interior faces that are free from the influence of the boundaries is described in this section and the faces influenced by the domain boundaries is described in the next section.

The polynomial reconstruction is face centered and every face has an associated stencil, represented by $\mathcal{N}(f)$. This stencil should be as compact as possible and should have enough points in each direction to maintain the required order. In addition the stencil size, n_s should be higher than the number of coefficients, n_{coefs} , this is required in order to have an overdetermined WLS problem [7,12]. The number of coefficients or terms is listed in Table 1, for each polynomial order.

To have the most compact stencil in the unstructured grids framework, the concept of cell neighbouring by vertex will be used. A discussion of the advantages of vertex neighbours over the face ones is done in [57]. The cells that have at least one common vertex with the considered face are classified as first vertex neighbours of face f . As for second order vertex neighbours, they are cells that share at least one vertex with a cell that is a first vertex neighbour, and so on for higher vertex neighbours. Fig. 2(a) and (b) shows two examples of the vertex neighbours that are used in the reconstructions of a face for the cases of a Cartesian grid and a polyhedral irregular one, respectively.

Table 2 lists the order of vertex neighbours necessary to build the stencil for each of the polynomials used in this work. The fourth vertex neighbours are used in the reconstruction of the 7th order polynomial.

Table 2

Order of vertex neighbours required for a p th order polynomial.

p th Order polynomial	1	3	5	7
Order of vertex neighbours	1	2	3	4

Tests performed to verify if a p th polynomial reconstruction actually builds a p th complete polynomial have shown that to reconstruct, successfully, the high-order terms it is necessary to have in each axis direction at least $p + 1$ cells. For the faces far enough from the boundaries this requirement was always achieved with the concept of vertex neighbouring, see Fig. 2.

2.2.4. Stencil selection algorithm for faces near the computational domain

The treatment given to the boundaries face depends if its type is Dirichlet or Neumann. In the case of a Neumann type, the flux through the face is known and it is not necessary to perform the face reconstruction because the required value in Eq. (4) can be directly applied, by taking into account the analytical gradient at the Gauss points.

For a Dirichlet boundary, the distribution of the computational variable is known and each face can be added to the reconstruction stencil in order to use this information. As a consequence the stencils near the boundaries, will include both cells and boundary faces. In summary, if a cell from the stencil has any boundary face, they are automatically added to the stencil and each value from the face's centroid is considered in the reconstruction. This was done, to avoid an increase of the condition number and a bad distribution of points of the least-squares problem. For example, in eight-order scheme if all Gauss points were added, a total $4 \times 9 = 36$ points would come from the boundary faces, which is excessive when compared to the representative 8 polynomial terms of specific direction. Afterwards, from the reconstructed polynomial is possible to apply a high-order quadrature at the boundary faces.

In addition, it is necessary these stencils must satisfy the requirements mentioned previously: $n_s > n_{coefs}$, $n_x^s > p + 1$ and $n_y^s > p + 1$, where n_x^s and n_y^s are the number of points of the stencil in each axis direction, XX and YY respectively. When these conditions are not achieved, it is necessary to perform a stencil extension to accomplish the referred conditions.

Once the initial stencil is constructed the following expression allows to know its size in XX direction:

$$n_x^s = \frac{L_x^s}{h_{ref}^s} + 1.6, \quad (21)$$

where $L_x^s = x_{max}^s - x_{min}^s$ is the length in XX direction of the stencil and it is computed from the centroid coordinates of the stencil's cells and faces. The 1.6 is a factor that takes into account that the cell and face centroids can be used to the stencil's length. If no face boundaries were being considered a factor of 1.0 plus a tolerance of 0.1 would be enough for the n_x^s computation. The mean reference length of the stencil, h_{ref}^s , is given by:

$$h_{ref}^s = \frac{1}{n_{cells}^s} \sum_{i \in \mathcal{N}_c(f)} h_{ref}^i, \quad (22)$$

where n_{cells}^s is the number of cells in the stencil, $\mathcal{N}_c(f)$ is the set of cells from the stencil of face f and h_{ref}^i is the hydraulic diameter of cell i . The calculation in the YY axis direction is performed similarly and Eq. (21) is valid for both types of boundary conditions. When n_x^s or n_y^s is lower than $p + 1$, the stencil will be extended in the respective direction. This operation is done for any face and

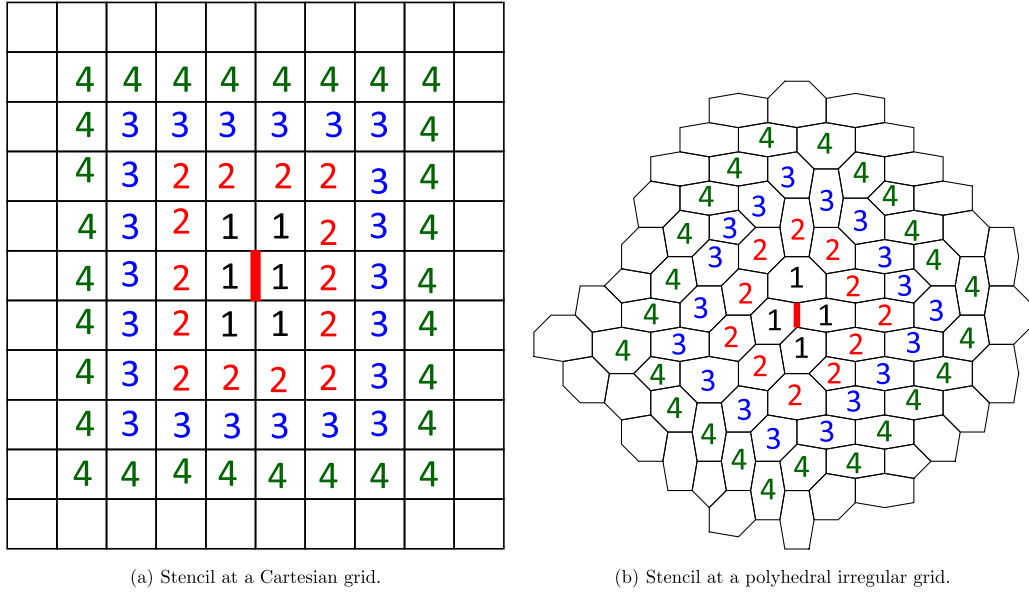


Fig. 2. Example of the different vertex neighbours of a face, marked in red, where the numbers inside of each cell represents its neighbouring order. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

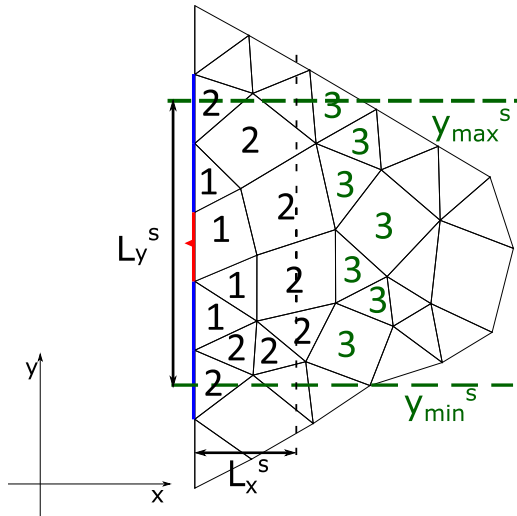


Fig. 3. Example of a stencil extension in an elliptic hybrid grid for a cubic reconstruction of the boundary face marked with red color and the other additional boundaries faces are marked with blue color. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

repeated after an extension to verified if the current stencil can maintain the required order.

Fig. 3 shows an example of a stencil extension performed on an elliptic hybrid grid, for a cubic reconstruction at the boundary face marked with the red color. The initial stencil includes the first and second order vertex neighbours, numbered with the black color and the additional boundary faces are marked with the blue color. In the YY direction the current stencil is long enough and will not grow in that direction, but in the XX direction it is necessary to perform a stencil expansion. To avoid cells in the YY direction both a lower and upper bounds of the stencil are computed, which are defined by y_{\min}^s and y_{\max}^s , respectively, and both represented by the dotted lines in the Fig. 3. Finally, the new cells added to the stencil after the extension are represented in Fig. 3 with the green color and the number 3.

In the case of Neumann boundary faces the diffusive flux is known and their contribution to the WLS problem is slightly different of the Dirichlet case. The line in the \mathbf{D}_f matrix referred to the boundary face, b , is included in the form of $\nabla \mathbf{d}_f(\mathbf{x}_b) \cdot \mathbf{n}_b$ instead of \mathbf{d}_f and the entry for the boundary face in the vector ϕ_f will be $\nabla \phi_b \cdot \mathbf{n}_b$ instead of ϕ_s , where \mathbf{n}_b is the boundary face outward unitary normal.

2.3. Source term

A high-order Gauss–Legendre Quadrature (GLQ) for the integral of the source term $\varphi_{\phi P}$ is also required. For 2D space the GLQ can be directly applied for quadrilateral and triangular 2D forms and for tetrahedral 3D forms. Since this work is related with 2D unstructured grids, it became necessary to divide the non triangular cells into several triangles, like the example of Fig. 4(a), where every triangle has a common vertex, the cell centroid.

To determine the Gauss integration points coordinates for each triangular element, simplex coordinates, $(\zeta_1, \zeta_2, \zeta_3, \dots)$ are used. A simplex region uses the number of points given by the dimension space plus one, so for the 2D space will be necessary 3 points and the area coordinates are obtained according to Akin [60].

The integration of a triangle is computed by:

$$I_{\Delta}(f) = S_{\Delta} \sum_{i \in \mathcal{G}(\Delta)} w_{G_i} f(\zeta_1, \zeta_2, \zeta_3), \quad (23)$$

where ζ_i denote the simplex coordinates of the triangle, the respective area is computed by $S_{\Delta} = (\mathbf{x}_1 - \mathbf{x}_3) \times (\mathbf{x}_2 - \mathbf{x}_3)/2$ and $\mathcal{G}(\Delta)$ is the set of Gauss points of the triangle Δ . The final expression for the integration of the source term is given by:

$$\int_{CV} \varphi_{\phi} dV = \sum_{i \in \mathcal{T}(P)} \left(S_{\Delta_i} \sum_{j \in \mathcal{G}_i(\Delta_i)} w_{G_j} \varphi_{\phi}(\mathbf{x}_j) \right), \quad (24)$$

where $\mathcal{T}(P)$ is the set of triangular elements that result from the cell P division. The Gauss points coordinates of each triangular element are expressed as:

$$\mathbf{x}_j = \zeta \mathbf{x}_v = \begin{bmatrix} \zeta_1 & \zeta_2 & \zeta_3 \end{bmatrix} \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix} \quad (25)$$

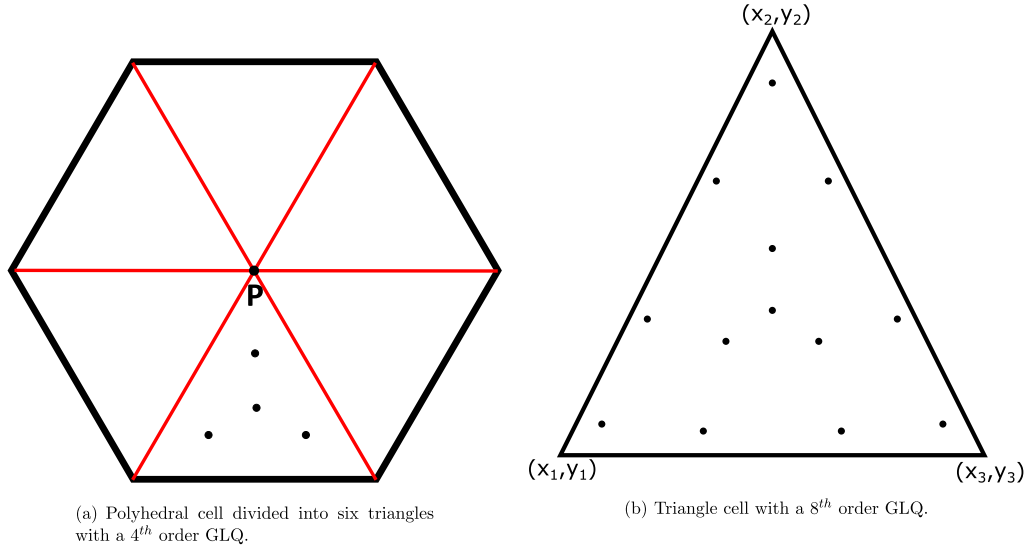


Fig. 4. Example of a division of a polyhedral cell in triangles in order to perform the numerical integration of source term at cell P , in red is the face's sub-triangles. The dots represent the Gauss points of the GLQ. At the right, an example of the 8th order GLQ applied to a triangle cell. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3

Number of Gauss points required for a triangular element with a p th order reconstruction.

p th Order polynomial	1	3	5	7
Number of Gauss points - n_G	1	4	7	13

where the \mathbf{x}_v is the vertex coordinates of the triangular element. Table 3 lists the number of required Gauss points for each reconstruction order in a triangular element.

3. Selection of the weight function

In this section several numerical tests are presented to investigate the WLS weight function, and its performance is compared by using two different grid types. According to Clain et al., [52] and Chenoweth et al., [61] the weight function and shape factor can influence the properties of the matrix \mathbf{A} , accuracy order and error magnitude of the scheme.

In the present work three weight functions, w_{LS_1} , w_{LS_2} and w_{LS_3} , are compared. The first one is given by:

$$w_{LS_1}(\mathbf{x}_p) = \frac{1}{(d_{fp})^k}, \quad (26)$$

where $d_{fp} = \sqrt{(x_p - x_f)^2 + (y_p - y_f)^2}$ is the distance between the face f and the cell centroid P and k is the shape factor, typically is equal 2. This function was used in the works of Ollivier-Gooch et al., [7], and Cueto-Felgueroso et al. [46]. The second one is given by:

$$w_{LS_2}(\mathbf{x}_p) = \frac{\exp\left(-\left(\frac{kd_{fp}}{2d_{fmax}}\right)^2\right) - \exp(-k^2)}{1 - \exp(-k^2)}, \quad (27)$$

where $d_{fmax} = \max(d_{fp})$ is the maximum distance between a cell of the stencil and the face centroid. This function was used in the work of [47]:

$$w_{LS_3}(\mathbf{x}_p) = \frac{\exp\left(-\left(\frac{kd_{fp}}{d_{fmax}}\right)^2\right) - \exp(-k^2)}{1 - \exp(-k^2)}, \quad (28)$$

where the stencil radius, d_{fmax} is used instead of the stencil diameter, $2d_{fmax}$, that was used in the previous weight function w_{LS_3} .

In the section, the following analytical solution of the Poisson equation was considered:

$$\phi(x, y) = \exp\left(-\frac{(x - 0.5)^2 + (y - 0.5)^2}{0.0175}\right), \quad (29)$$

and the source term φ_ϕ is computed according to $\varphi_\phi = \nabla^2 \phi$.

Cartesian grids are obtained with grid refinements, from a $h_{ref} = 1.00E - 01$ to $h_{ref} = 1.56E - 03$. The coarser grid contains 100 cells and the finest one 409,600 cells, an example of these grids is presented in Fig. 5(a).

The hybrid grid is built from a triangular one, where 20% of the cells are randomly selected to merge with one face neighbour resulting in a quadrilateral cell, afterwards an elliptical softener is applied to the grid. The main goal of this last step is to change the grid from its original cell distribution. Six grid refinements were considered, the coarser grid has 167 cells and the finest one has 200,905 cells. An example of this grid type is showed in Fig. 5(b). This grid type was selected as a robustness test of the proposed diffusive schemes and weighted functions.

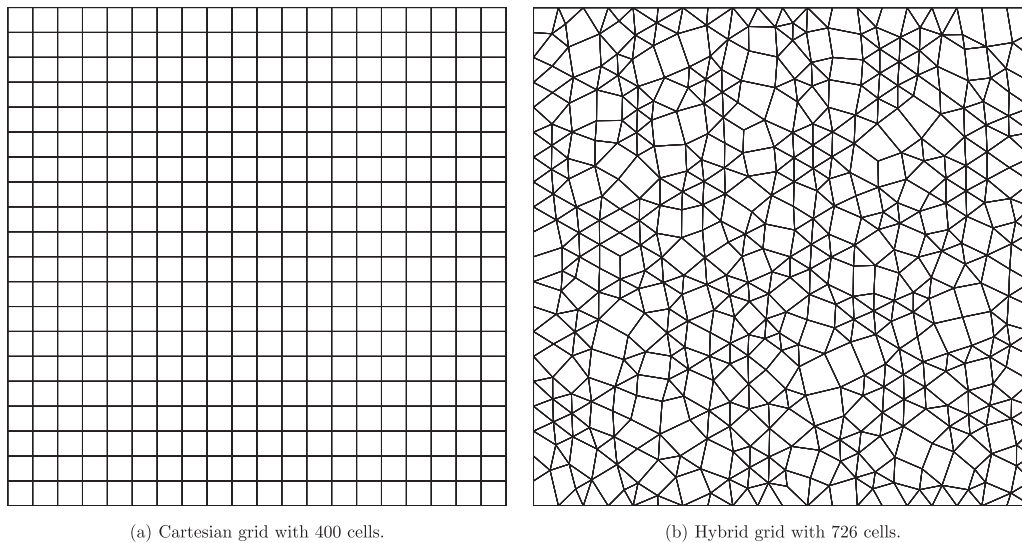
3.1. Properties of matrix \mathbf{A}

According to Clain et al., [52], the global matrix must satisfy several requirements, such as the main diagonal should be negative and the inverse of the global matrix should have all the entries with the same signal. These requirements were successfully verified for all studied weight functions.

The matrix condition number, $\text{cond}(\mathbf{A})$, is the ratio between the largest and smallest eigenvalues from the matrix. When the condition number is very high, it means that the matrix is close to be singular and ill conditioned. An estimation of the matrix condition number, that results from the discretization of the Poisson equation in a uniform grid, is given by [36,62]:

$$\text{cond}(\mathbf{A}) = \frac{4}{\pi^2 h^2}. \quad (30)$$

The above expression only depends on the grid size, h , and was deduced for 2nd order finite difference method. For a Cartesian

**Fig. 5.** Example of a Cartesian and hybrid grid.**Table 4**The matrix condition number obtained with w_{LS_1} for the Cartesian grid (left) and hybrid grid (right).

k	Cartesian grid				Hybrid grid			
	FLS2	FLS4	FLS6	FLS8	FLS2	FLS4	FLS6	FLS8
0	2.80E+03	7.98E+03	4.22E+04	7.23E+04	4.69E+05	1.54E+05	1.11E+06	4.34E+06
2	6.74E+02	9.69E+02	1.33E+03	1.73E+03	1.33E+03	2.98E+03	4.17E+03	4.31E+03
4	8.73E+02	1.34E+03	1.52E+03	1.84E+03	2.11E+03	3.50E+03	4.26E+03	5.45E+03
6	9.28E+02	1.41E+03	1.53E+03	1.83E+03	2.48E+03	8.79E+04	4.36E+03	6.22E+03
8	9.40E+02	1.37E+03	1.57E+03	1.83E+03	2.69E+03	6.00E+03	5.76E+03	6.82E+03

Table 5The matrix condition number obtained with w_{LS_2} for the Cartesian grid (left) and hybrid grid (right).

k	Cartesian grid				Hybrid grid			
	FLS2	FLS4	FLS6	FLS8	FLS2	FLS4	FLS6	FLS8
3	7.09E+02	2.24E+03	3.93E+05	4.12E+05	5.31E+04	4.45E+05	5.08E+06	5.37E+06
4	8.72E+02	9.63E+02	6.75E+04	2.19E+05	1.73E+03	3.01E+05	1.40E+07	3.00E+08
5	9.30E+02	1.01E+03	1.47E+03	1.92E+05	1.96E+03	2.43E+04	6.61E+06	2.99E+06

Table 6The matrix condition number obtained with w_{LS_3} for the Cartesian grid (left) and hybrid grid (right).

k	Cartesian grid				Hybrid grid			
	FLS2	FLS4	FLS6	FLS8	FLS2	FLS4	FLS6	FLS8
3	NaN	1.07E+03	1.48E+03	2.38E+03	2.22E+03	3.53E+03	7.07E+05	9.15E+07
4	NaN	1.31E+03	1.48E+03	1.85E+03	3.45E+03	5.13E+05	5.12E+03	5.05E+04
5	NaN	1.40E+03	1.66E+03	1.92E+03	4.24E+03	9.88E+03	8.11E+03	1.11E+04
6	NaN	1.43E+03	1.75E+03	2.08E+03	4.46E+03	1.90E+04	1.20E+04	2.36E+04
8	NaN	1.47E+03	1.21E+04	3.40E+04	NaN	1.92E+05	1.66E+06	1.34E+05

grid with 1600 cells and $h_{ref} = 2.5E - 02$ and a hybrid grid with 2969 cells and $h_{ref} = 1.60E - 02$, it gives $\text{cond}(\mathbf{A}) = 6.48E + 02$ and $\text{cond}(\mathbf{A}) = 1.59E + 03$, respectively. These values are taken for reference purposes and to be compared with the effective matrix condition numbers obtained with the different weight functions. Tables 4–6 list the obtained condition numbers for the three considered weight functions using different shape factors, k , for all schemes at study.

Tables 4–6 show that the shape factor k has negligible influence in the matrix condition number for Cartesian grids. For the hybrid grids it is possible to verify that the shape factor influences the matrix condition number. The majority of the results are consistent with the reference value obtained from Eq. (30).

Table 4 lists the matrix condition number for the weight function w_{LS_1} . For $k = 0$ (an unweighted LS problem) the condition number is one or two orders of magnitude higher than the reference values for both considered grids and consequently this shape factor will not be considered anymore for the w_{LS_1} function. Other values of k give conditions numbers with similar orders of magnitude of Eq. (30).

Table 5 lists the condition number for weight function w_{LS_2} which has the highest condition numbers, specifically three orders of magnitude greater than the reference values for FLS6 and FLS8, and consequently will not be considered. The w_{LS_2} was used by Nogueira, [40,47–49] for a fourth-order scheme on both quadrilateral and triangular grids. The present results show that for the

Table 7

The \mathcal{O}_1 of the schemes for Cartesian and hybrid grids with the weight function w_{LS_1} , where the (-) indicates that a converged solution was not achieved.

Shape Factor - k	Cartesian grid				Hybrid grid			
	FLS2	FLS4	FLS6	FLS8	FLS2	FLS4	FLS6	FLS8
2	2.00	3.92	5.94	7.75	2.03	4.03	6.03	7.50
4	2.00	4.01	6.03	8.25	2.05	4.08	5.94	7.72
6	2.00	3.98	6.06	8.36	2.05	-	6.01	7.51
8	2.00	3.99	6.12	8.09	2.05	-	5.96	7.58

Table 8

The \mathcal{O}_1 of the schemes for Cartesian and hybrid grids with the weight function w_{LS_1} , where (-) indicates that a converged solution was not achieved and (*) indicates the discarded shape factors from previous analysis.

Shape Factor - k	Cartesian grid				Hybrid grid			
	FLS2	FLS4	FLS6	FLS8	FLS2	FLS4	FLS6	FLS8
3	-	3.97	*	*	-	3.88	*	*
4	-	*	6.02	7.96	-	*	5.73	-
5	-	3.99	6.02	8.10	-	2.53	5.86	6.70
6	-	3.99	6.02	7.25	-	-	4.73	6.90

hybrid grid and for the FLS6 and FLS8 schemes, this weight function produces large matrix condition numbers, which will lead to a ill-posed problem. Due to these facts, the w_{LS_2} function will not be considered in the next sections.

Table 6 lists the condition number for the weight function w_{LS_3} . For a Cartesian grid this function cannot be used for FLS2 scheme because there are some entries in the main diagonal of the global matrix are close to zero, reducing the convergence properties from the matrix. But since our interest is mainly in high-order schemes, it still is considered as a possible choice. For the other schemes the condition number has the same order of magnitude of the reference value and it is very similar to the one obtained with the function w_{LS_1} . In the hybrid grid, the condition number for some k 's is two orders of magnitude greater than the reference value and consequently they will be discarded from this study, namely $k = 8$ for all schemes, $k = 4$ for the FLS4 scheme and $k = 3$ for both FLS6 and FLS8 schemes.

3.2. Numerical schemes accuracy

In this section, the convergence order will be analysed for the case of the w_{LS_1} and w_{LS_2} weighted function. Table 7 lists the average convergence order for all schemes when using the weight function w_{LS_1} . For the Cartesian grid the results show that for every k value the theoretical order of convergence is achieved. The hybrid grid displays cases where the order of accuracy is slightly lower than the theoretical value. It is worth mentioning that the hybrid grid is a robust test and that for the FLS4 and k equal to 6 and 8 the solver did not achieve a converged solution. This result is expected since Table 4 show high condition numbers for these cases, which justifies the lack of convergence and as a consequence these values of k will be discarded.

Table 8 lists the average convergence order for all schemes with the weight function w_{LS_3} . For the Cartesian grid the theoretical order of convergence is almost achieved for all the shape factors. However for the hybrid grid this weight function never achieved the theoretical order of convergence for the FLS8 scheme and the overall results are not as consisted as in the case of the w_{LS_1} . Failing this robustness test here performed and as consequence the w_{LS_3} will not be selected as a weight function. From the analysis of Tables 7 and 8 the best choice for weight function is the w_{LS_1} . Another advantage of the w_{LS_1} is that the evaluation of polynomials functions has a lower computational cost than the evaluation of exponential functions.

3.3. Numerical error magnitude

The previous sections suggests that the w_{LS_1} is the best choice for the WLS weight function, but the best shape factor, k , was not justified yet. To achieved this, the influence of k in the numerical error magnitude will be studied, in this section.

The w_{LS_1} with a shape factor k equal to two is a very popular choice, see e.g. Ollivier-Gooch et al., [7,43], and Cueto-Felgueroso et al.,[46]. The numerical error of the three high-order schemes for several shape factors, with a Cartesian grid with 25,600 cells and a hybrid grid with 47,345 cells is shown in Fig. 6. The results suggest a significant difference between $k = 2$ and the others shape factors. From Fig. 6(a) the best shape factor for the Cartesian grid is $k = 8$, however for the FLS4 scheme the best result appears with $k = 4$ for the hybrid grid, see Fig. 6(b). Consequently one may conclude that there is a relation between the order of convergence of the scheme and the shape factor. Due to this, the shape factor will be equal to the order of scheme, $k = o$, where $o = p + 1$.

Table 9 and Fig. 6 compare the error $\|e\|_1$ for $k = 2$ and $k = o$. When k is equal to the order of the scheme, the solution's accuracy is improved, especially for the FLS8. The error obtained with $k = 2$ could be reduced substantially from 3 to 37 times, depending on the considered scheme and grid. Similar conclusions presented in Table 9 can be derived from the maximum error, $\|e\|_\infty$. From now on the weight function will be defined by:

$$w_{LS}(\mathbf{x}_p) = \frac{1}{(d_{fp})^o}, \quad (31)$$

which is the selected weight function for the proposed scheme.

4. Numerical schemes verification

In this section several numerical tests are presented to investigate the influence of grid type, boundary conditions type and grid quality parameters on the proposed schemes accuracy. This will be followed by an efficiency study, where the required memory and computing solver-run time (SRT) will be compared for the different scheme's order and grid types.

4.1. Influence of grid type

Numerical experiments were performed with four grid types: Cartesian, triangular, polyhedral and hybrid.

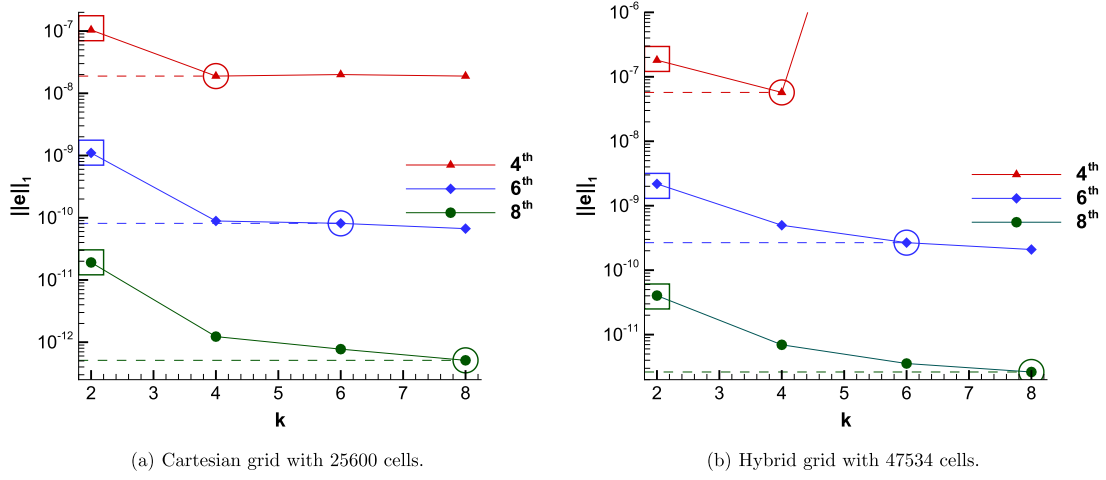


Fig. 6. The $\|e\|_1$ for all high-order schemes, where the \circ is the proposed shape factor, $k = o$, and the \square is the shape factor used by most authors, $k = 2$.

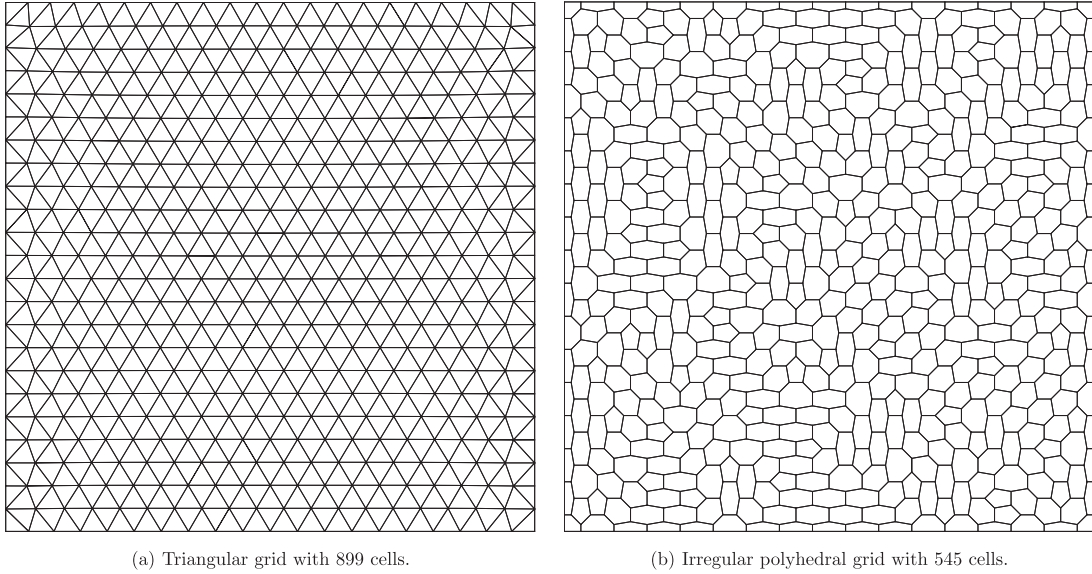


Fig. 7. Examples of a triangular and a polyhedral grids.

Table 9

The $\|e\|_1$ for a Cartesian grid with 25,600 cells and a hybrid grid with 47,534 cells for $k = 2$ and $k = o$, where the last line is the ratio between the errors of the two shape factors.

k	Cartesian grid (25600 cells)			Hybrid grid (47534 cells)		
	FLS4	FLS6	FLS8	FLS4	FLS6	FLS8
2	1.04E-07	1.10E-09	1.91E-11	1.81E-07	2.18E-09	4.03E-11
o	1.89E-08	8.12E-11	5.11E-13	5.72E-08	2.67E-10	2.63E-12
Ratio	5.4899	13.4811	37.4512	3.1584	8.1606	15.3334

The triangular grid was generated with DISTMESH code, [63], using a constant grid size. Fig. 7(a) shows a regular grid for interior cells, only the cells near the lateral boundaries are irregular, the coarsest grid has 211 cells and the finest one has 91,772 cells. The polyhedral grid is formed by irregular pentagonal and hexagonal cells, the coarsest grid contains 145 cells and the finest one 131,585 cells.

The numerical verification was performed in a domain $[0, 1] \times [0, 1]$ with Dirichlet boundary conditions and the following analytical solution:

$$\phi(x, y) = \psi_1 - \psi_2 - \psi_3 + \psi_4, \quad (32)$$

where ψ_i is an auxiliary exponential function with the following form:

$$\psi_i = \exp(-120((x - x_i)^2 + (y - y_i)^2)) \quad (33)$$

and the different auxiliary functions are centered in the following coordinates $(x_1, y_1) = (\frac{1}{3}, \frac{1}{3})$, $(x_2, y_2) = (\frac{1}{3}, \frac{2}{3})$, $(x_3, y_3) = (\frac{2}{3}, \frac{1}{3})$ and $(x_4, y_4) = (\frac{2}{3}, \frac{2}{3})$.

Fig. 8(a) and (b) show the convergence curves of the schemes with the Cartesian grid. For the finest grid the error of the FLS8 scheme is close to the machine truncation error. Tables 10 and 11 list the errors norm-1 and norm- ∞ , respectively. The effective convergence order is virtually equal to the theoretical value

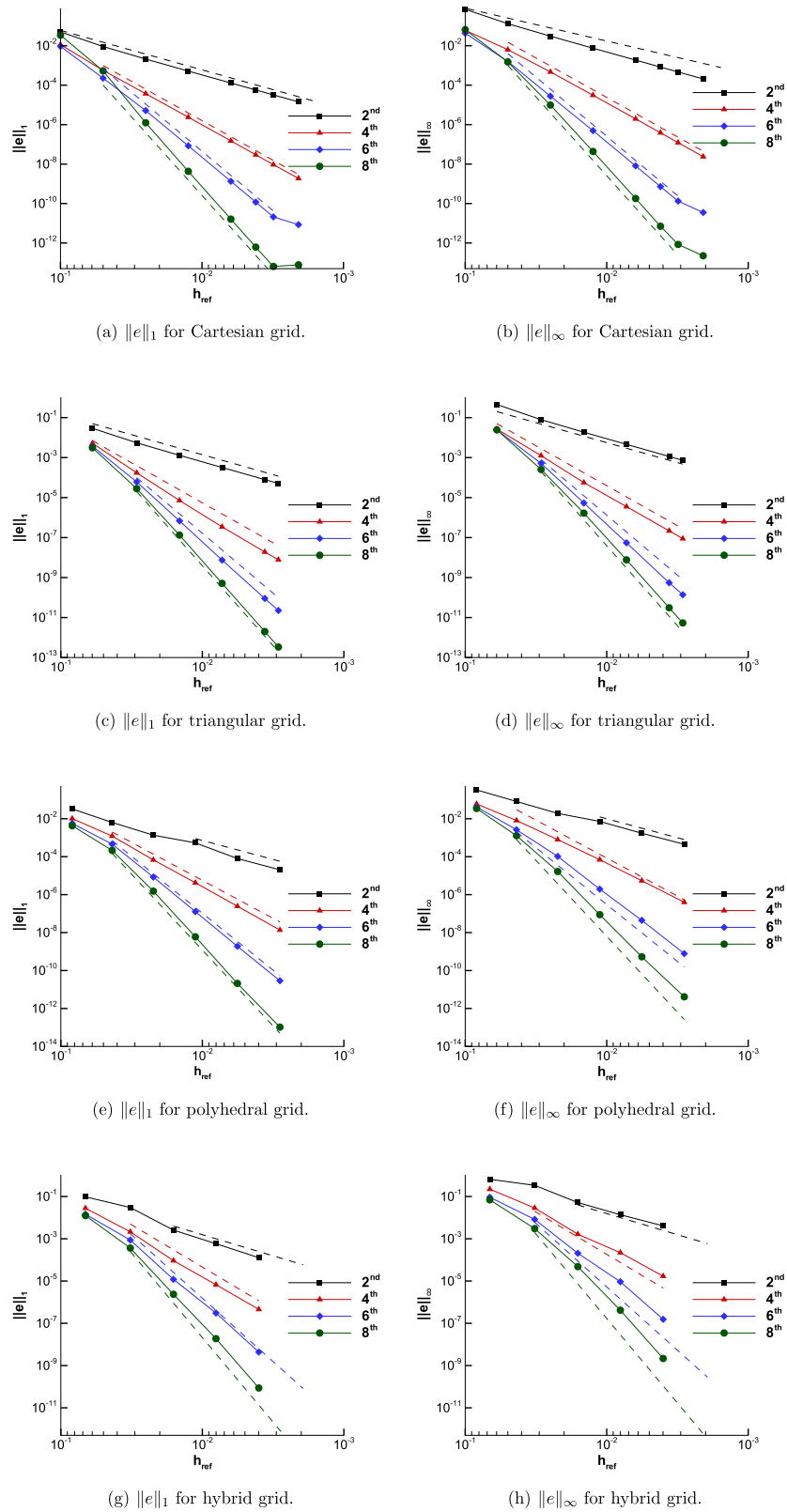


Fig. 8. Convergence curves for all grid types of $\|e\|_1$ and $\|e\|_\infty$ for all schemes, where the dotted line represents the theoretical slope of each scheme.

Table 10
Obtained $\|e\|_1$ and \mathcal{O}_1 for the Cartesian grid.

Cartesian grid		FLS2		FLS4		FLS6		FLS8	
n_c	h_{ref}	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1
25600	6.25E-03	1.29E-04	2.00	1.50E-07	3.99	1.34E-09	5.97	1.62E-11	8.05
57600	4.17E-03	5.72E-05	2.00	2.96E-08	4.00	1.18E-10	5.99	6.17E-13	8.07
102400	3.13E-03	3.22E-05	2.00	9.36E-09	4.00	2.10E-11	6.00	6.45E-14	7.85

Table 11
Obtained $\|e\|_\infty$ and \mathcal{O}_∞ for the Cartesian grid.

Cartesian grid		FLS2		FLS4		FLS6		FLS8	
n_c	h_{ref}	$\ e\ _\infty$	\mathcal{O}_∞	$\ e\ _\infty$	\mathcal{O}_∞	$\ e\ _\infty$	\mathcal{O}_∞	$\ e\ _\infty$	\mathcal{O}_∞
25600	6.25E-03	1.87E-03	2.01	1.96E-06	3.98	8.10E-09	5.93	1.80E-10	7.92
57600	4.17E-03	8.28E-04	2.01	3.86E-07	4.01	7.22E-10	5.96	7.04E-12	7.99
102400	3.13E-03	4.66E-04	2.00	1.22E-07	4.01	1.33E-10	5.89	8.42E-13	7.38

Table 12
Obtained $\|e\|_1$ and \mathcal{O}_1 for the triangular grid.

Triangular grid		FLS2		FLS4		FLS6		FLS8	
n_c	h_{ref}	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1
14632	7.25E-03	3.20E-04	2.02	3.44E-07	4.37	7.47E-09	6.50	5.15E-10	8.00
58698	3.62E-03	7.94E-05	2.01	1.91E-08	4.16	9.03E-11	6.36	2.00E-12	8.00
91772	2.90E-03	5.08E-05	2.00	7.74E-09	4.05	2.25E-11	6.22	3.35E-13	8.00

Table 13
Obtained $\|e\|_\infty$ and \mathcal{O}_∞ for the triangular grid.

Triangular grid		FLS2		FLS4		FLS6		FLS8	
n_c	h_{ref}	$\ e\ _\infty$	\mathcal{O}_∞	$\ e\ _\infty$	\mathcal{O}_∞	$\ e\ _\infty$	\mathcal{O}_∞	$\ e\ _\infty$	\mathcal{O}_∞
14632	7.25E-03	4.60E-03	2.01	3.54E-06	3.98	5.48E-08	6.60	7.70E-09	7.75
58698	3.62E-03	1.14E-03	2.01	2.17E-07	4.02	5.49E-10	6.64	3.08E-11	7.96
91772	2.90E-03	7.30E-04	2.00	8.84E-08	4.03	1.38E-10	6.19	5.38E-12	7.81

Table 14
Obtained $\|e\|_1$ and \mathcal{O}_1 for the irregular polyhedral grid.

Polyhedral grid		FLS2		FLS4		FLS6		FLS8	
n_c	h_{ref}	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1
8321	1.12E-02	5.35E-04	1.37	4.23E-06	4.07	1.27E-07	6.15	5.97E-09	8.14
33024	5.65E-03	7.97E-05	2.77	2.45E-07	4.15	1.89E-09	6.12	2.09E-11	8.23
131585	2.83E-03	2.04E-05	1.98	1.36E-08	4.19	2.89E-11	6.06	1.04E-13	7.69

Table 15
Obtained $\|e\|_\infty$ and \mathcal{O}_∞ for the irregular polyhedral grid.

Polyhedral grid		FLS2		FLS4		FLS6		FLS8	
n_c	h_{ref}	$\ e\ _\infty$	\mathcal{O}_∞	$\ e\ _\infty$	\mathcal{O}_∞	$\ e\ _\infty$	\mathcal{O}_∞	$\ e\ _\infty$	\mathcal{O}_∞
8321	1.12E-02	7.24E-03	1.43	6.93E-05	3.58	1.95E-06	5.83	8.80E-08	7.69
33024	5.65E-03	1.72E-03	2.09	5.39E-06	3.72	4.42E-08	5.51	5.28E-10	7.45
131585	2.83E-03	4.51E-04	1.94	3.97E-07	3.78	7.78E-10	5.86	4.11E-12	7.04

for the norm-1 and in close agreement for the norm- ∞ . The error norms with the FLS8 are almost nine orders of magnitude lower than those obtained with the second-order scheme. Fig. 8(c) and (d), and Tables 12 and 13 list similar values of the error norms and its effective convergence order for the finest triangular grid.

The results obtained for irregular polyhedral grids (see Fig. 8(e) and (f), and Tables 14 and 15) also display the expected order of accuracy. A regular polyhedral grid was also tested, the conclusions about the several schemes order of convergence are identical to the irregular polyhedral grid but the values of the numerical error was one order of magnitude lower.

Tests with an elliptic hybrid grid were also performed and presented in Tables 16 and 17, and Fig. 8(g) and (h). The results confirm that the proposed method is very robust and that the convergence order is close to the theoretical one.

Fig. 9 shows a comparison of the $\|e\|_1$ evolution with the reference length using the FLS4 and FLS8 schemes and different grid types. Fig. 9(a) refers to the FLS4 scheme and for the same h_{ref} the regular grids display lower errors than the irregular ones. For the smaller reference length the triangular, Cartesian and regular polyhedral grids give identical errors. After the hybrid grid, the irregular polyhedral presents the worse behaviour. However this changes for the FLS8 scheme, see Fig. 9(b), where the triangular grid has

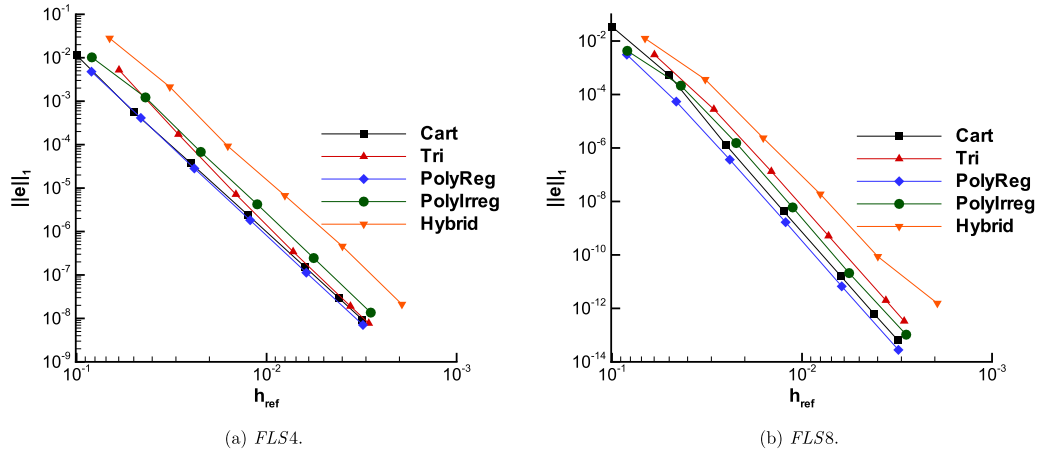


Fig. 9. Convergence curves of the error norm-1 with the h_{ref} with the FLS4 and FLS8 schemes for different grid types.

Table 16
Obtained $\|e\|_1$ and \mathcal{O}_1 for the hybrid grid.

Hybrid grid		FLS2		FLS4		FLS6		FLS8	
n_c	h_{ref}	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1
2969	1.60E-02	2.57E-03	3.50	9.23E-05	4.46	1.20E-05	6.10	2.36E-06	7.17
11827	8.01E-03	5.84E-04	2.14	6.72E-06	3.79	3.06E-07	5.31	1.88E-08	6.99
47534	3.99E-03	1.27E-04	2.19	4.00E-07	4.05	4.35E-09	6.11	8.68E-11	7.73

Table 17
Obtained $\|e\|_\infty$ and \mathcal{O}_∞ for the hybrid grid.

Hybrid grid		FLS2		FLS4		FLS6		FLS8	
n_c	h_{ref}	$\ e\ _\infty$	\mathcal{O}_∞	$\ e\ _\infty$	\mathcal{O}_∞	$\ e\ _\infty$	\mathcal{O}_∞	$\ e\ _\infty$	\mathcal{O}_∞
2969	1.60E-02	5.19E-02	2.67	1.65E-03	4.06	2.07E-04	5.24	4.80E-05	5.90
11827	8.01E-03	1.37E-02	1.92	2.20E-04	2.91	9.26E-06	4.50	4.17E-07	6.87
47534	3.99E-03	4.09E-03	1.74	1.66E-05	3.72	1.54E-07	5.88	2.16E-09	7.56

Table 18
Obtained $\|e\|_1$ and \mathcal{O}_1 with mixed boundary condition types in the irregular polyhedral grid for all numerical schemes.

Polyhedral grid		FLS2		FLS4		FLS6		FLS8	
n_c	h_{ref}	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1
145	8.31E-02	1.78E-01	-	5.04E-03	-	4.98E-04	-	1.36E-04	-
545	4.34E-02	3.77E-02	2.39	2.19E-04	4.83	9.26E-06	6.14	6.77E-07	8.17
2113	2.22E-02	7.26E-03	2.46	1.24E-05	4.28	1.63E-07	6.02	2.27E-09	8.49
8321	1.12E-02	1.14E-03	2.72	7.05E-07	4.21	2.05E-09	6.42	1.44E-11	7.42
33024	5.65E-03	1.84E-04	2.65	4.32E-08	4.06	3.07E-11	6.11	-	-

the worse behaviour (after the hybrid one) and the regular polyhedral grid presents the best evolution.

4.2. The influence of the neumann boundary condition type

For the case with Neumann boundary conditions, numerical tests were performed for two grid types: polyhedral and hybrid ones. The analytical solution of the Poisson equation is solved in a square 1×1 domain and the Neumann boundary condition is applied at $x = 0$ and $x = 1$. Dirichlet boundary condition is applied at the remaining boundaries. For this case, the used analytical solution is given by:

$$\phi(x, y) = \cos(3\pi x) \sin(2\pi y) \quad (34)$$

Fig. 10 shows the convergence curves for all schemes with polyhedral and hybrid grids. Tables 18 and 19 list the values of norm-1 and its effective convergence order for all schemes with the polyhedral and hybrid grids, respectively. The results point out that the theoretical convergence order is always achieved for both grids, being in some cases over convergent for the coarser grids, specially

for polyhedral grids. The Neumann boundary condition type yields higher numerical errors than the ones obtained with only Dirichlet boundary conditions.

A more detailed observation of the results shows a saturation of the solver around $1.00E-13$ for the sixth and eighth-order schemes, this situation was already noticed by Boularas et al., [33], when using trigonometric functions in the analytical solution of the Poisson equation.

4.3. Influence of the grid quality parameters

In this section a grid quality study was performed based in two parameters: the non-orthogonality angle and the volume ratio.

4.3.1. Influence of the non-orthogonality angle

The influence of non-orthogonality angle on numerical error was described by Juretić and Gosman [64] and in this section a constant non-orthogonality angle, α , is imposed to the vertical faces of a Cartesian grid, resulting in a twist grid. In this study

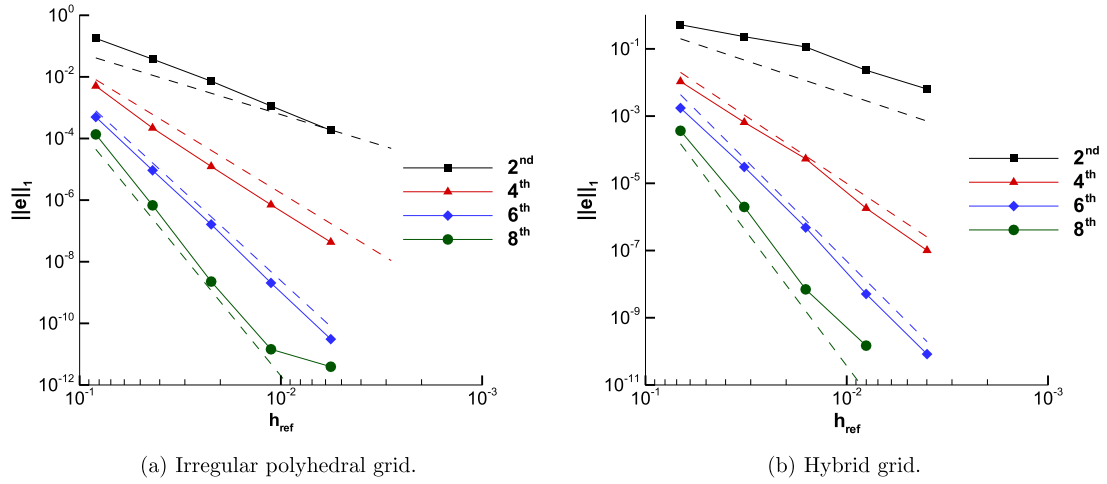


Fig. 10. Convergence curves of the mean error for mixed boundary condition types for all schemes in irregular polyhedral and hybrid grids.

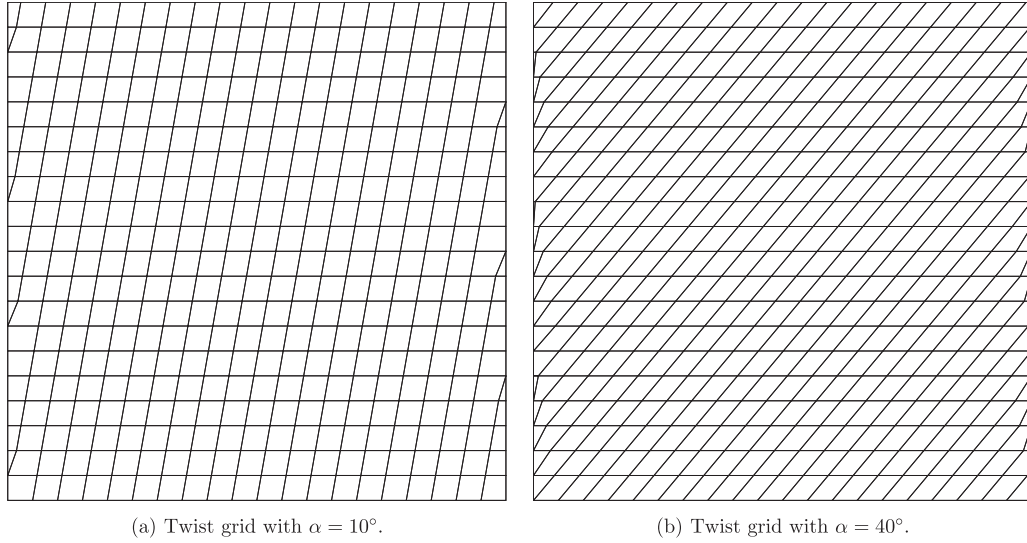


Fig. 11. Example of two twist grids with an imposed non-orthogonality angle α of 10° and 40° .

Table 19

Obtained $\|e\|_1$ and \mathcal{O}_1 with mixed boundary condition types in the hybrid grid for all numerical schemes.

Hybrid grid		FLS2		FLS4		FLS6		FLS8	
n_c	h_{ref}	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1
167	6.70E-02	5.23E-01	-	1.07E-02	-	1.74E-03	-	3.66E-04	-
726	3.23E-02	2.31E-01	1.12	6.53E-04	3.83	3.06E-05	5.53	1.98E-06	7.15
2969	1.60E-02	1.14E-01	1.00	5.37E-05	3.55	4.83E-07	5.90	6.99E-09	8.03
11827	8.01E-03	2.31E-02	2.31	1.81E-06	4.90	5.10E-09	6.58	1.48E-10	5.57
47534	3.99E-03	6.39E-03	1.85	1.01E-07	4.14	8.28E-11	5.92	-	-

five imposed angles were used: 0° , 10° , 20° , 30° and 40° . For these tests the same analytical solution of Section 4.1 was used, which is defined by the expression (32). Fig. 11 shows two twist grids with an imposed non-orthogonality angle of 10° and 40° . At the boundaries, some vertices were merged together to avoid small triangles that caused a local numerical error due to their very small volume.

Fig. 12. shows the numerical error distribution obtained with the FLS8 for two grids with a non-orthogonality angle of 0° and 40° . From their comparison it is possible to conclude that the numerical error suffers a rotation according to the imposed 40° of the non-orthogonality angle and that the boundaries did not contribute to the numerical error distribution.

Fig. 13 shows the numerical error in all twist grids at study with the FLS4 and FLS8 schemes. It reveals an almost constant convergence order that is independent of the imposed non-orthogonality angle, despite the increasing error magnitude for each successive α .

Table 20 lists the obtained error ratios (r) between a grid with an imposed α and a regular one for the finest grid at study. The error ratio, r is defined by:

$$r_i = \frac{\|e\|_i^\alpha}{\|e\|_i^0}, \quad (35)$$

where α is the imposed non-orthogonality angle of the grid and i is the norm used.

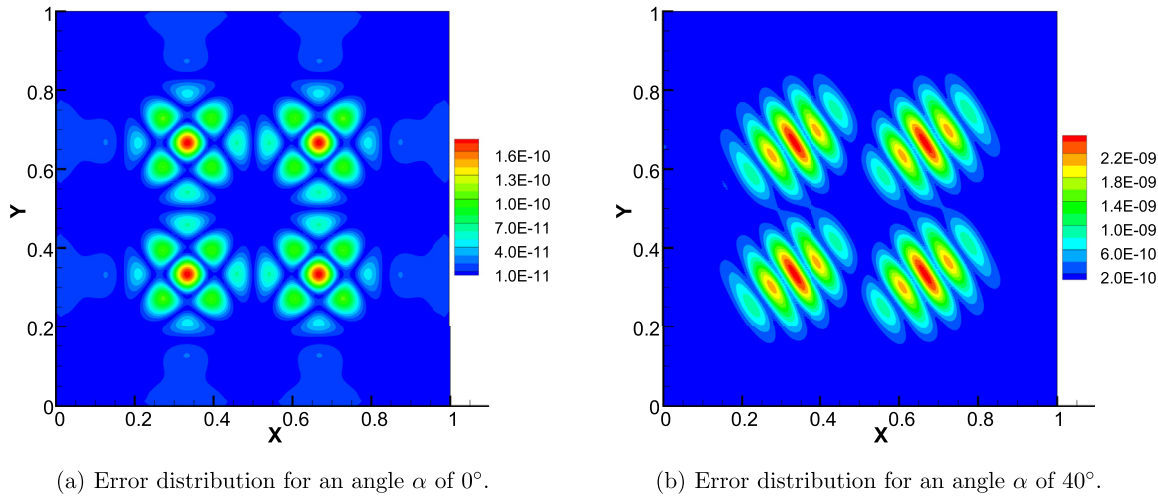


Fig. 12. Error distribution of the FLS8 scheme for two grids with non-orthogonality angle α of 0° and 40° , respectively. The present grid resolution is equivalent to a 160×160 Cartesian grid.

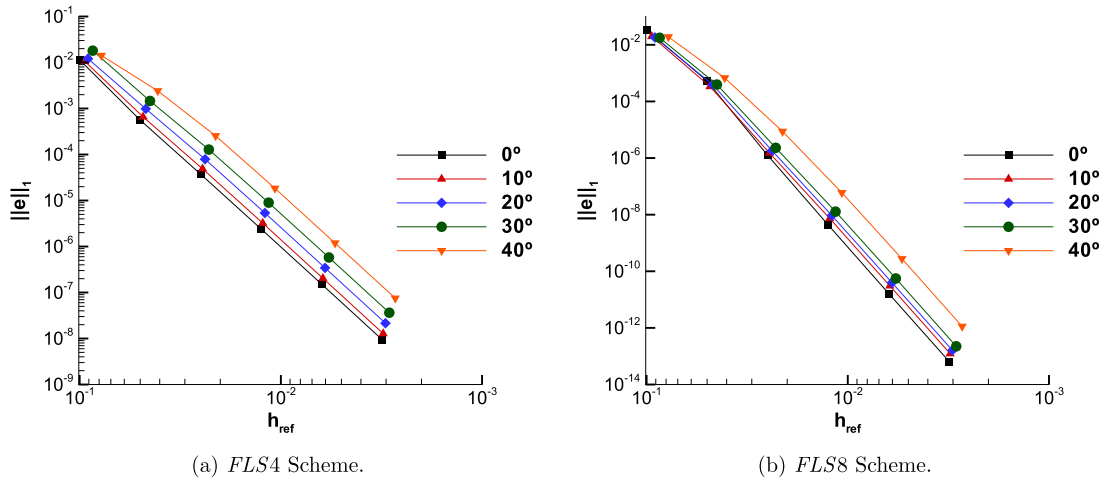


Fig. 13. Convergence curve of the $\|e\|_1$ for FLS4 and FLS8 schemes in all twist grids at study.

Table 20
Ratio of mean and maximum errors for all schemes, the ratio is computed from the results of a twist grid and of a Cartesian grid with 102,400 cells.

Non-orthogonality angle - α	FLS2		FLS4		FLS6		FLS8	
	r_1	r_∞	r_1	r_∞	r_1	r_∞	r_1	r_∞
10°	1.01	1.00	1.37	0.89	2.06	2.85	1.91	1.62
20°	1.05	1.01	2.30	1.42	3.17	4.25	2.51	1.93
30°	1.16	1.03	3.89	2.08	4.24	5.33	3.48	2.39
40°	1.28	1.04	8.02	4.29	11.68	15.60	17.55	12.92

FLS4 scheme displays a mean error ratio of almost one order of magnitude when the non-orthogonality angle is equal to 40° . For the same angle, the FLS6 and FLS8 schemes have ratios superior to one order magnitude for both mean and maximum error.

4.3.2. Influence of volume ratio

To study the influence of the cell's volume ratio in the scheme's accuracy, several grids were generated by imposing random displacements to a Cartesian grid.

The volume ratio grid parameter measures the ratio between the maximum and minimum cell volume in a grid, it can be calculated by:

$$\frac{V_{\max}}{V_{\min}} = \frac{(h_{ref} + 2\Gamma)^2}{(h_{ref} - 2\Gamma)^2} \quad (36)$$

where Γ is the displacement applied to a face and it is given by $\Gamma = \gamma h_{ref}$, where γ is a relative percentage to the reference length of the cell. Four different percentages, 0%, 10%, 20% and 30%, were used. In the worse case scenario, the grids will have a volume ratio of 2.25, 5.44 and 16, respectively, see Fig. 14.

Fig. 14 shows two examples of these grids, one with a displacement of 10% and another one with a displacement of 30%. The analytical solution used in this case is given by Eq. (29) and the errors of the FLS4 and FLS8 schemes are presented in Fig. 15. The expected schemes convergence order was achieved for all grids at study.

Table 21 lists the ratios, r , of the mean and maximum errors of the finest grid, defined by Eq. (35) but this time α should be replaced by γ in the formula. The difference between the successive volume ratios is more pronounced for the maximum error. Partic-

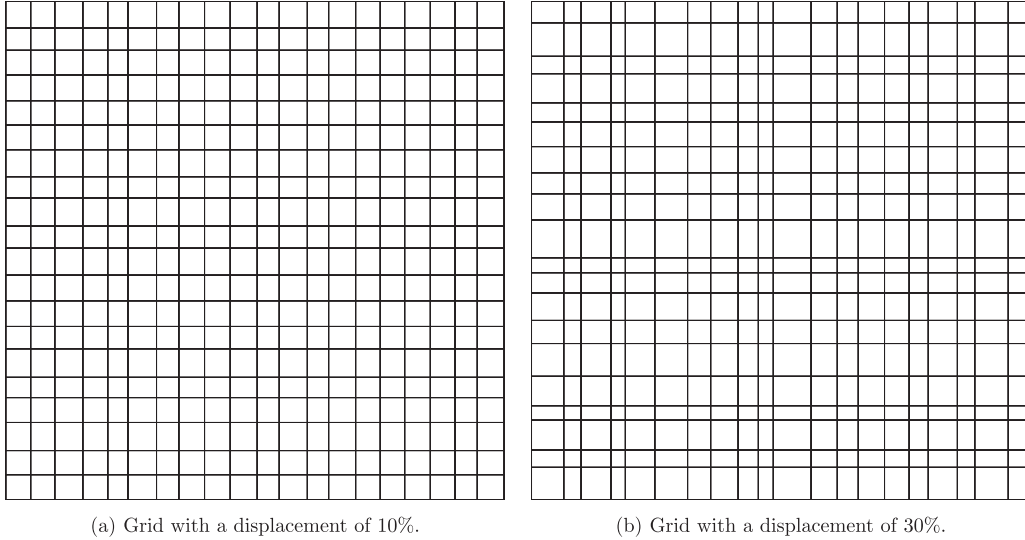


Fig. 14. Examples of two displacements percentages applied to a Cartesian grid with 400 cells.

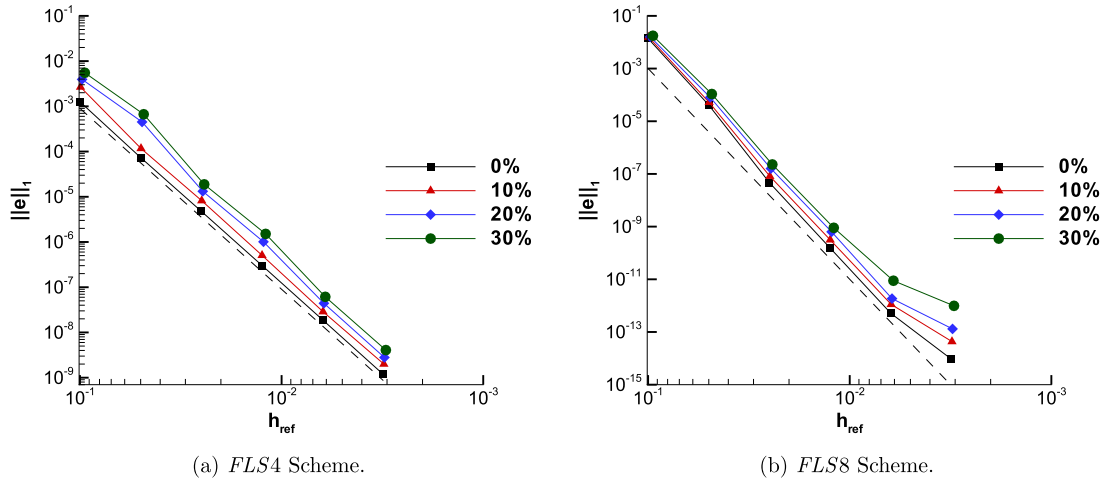


Fig. 15. Convergence curves for FLS4 and FLS8 schemes in all perturbed grids.

Table 21

Ratio of mean and maximum errors for all schemes between the grid with an imposed displacement and a Cartesian one, both with 25,600 cells.

Imposed displacement - γ (%)	FLS2		FLS4		FLS6		FLS8	
	r_1	r_∞	r_1	r_∞	r_1	r_∞	r_1	r_∞
10%	1.01	1.54	1.52	4.44	1.86	10.35	2.20	8.19
20%	1.09	1.55	2.33	4.72	2.81	15.15	3.62	11.05
30%	1.22	1.78	3.24	7.53	3.89	22.30	17.28	30.61

ularly for the FLS6 and FLS8 schemes the error could be one order of magnitude greater than for the case of the Cartesian grid.

The results showed in Tables 20 and 21 suggest that the schemes accuracy is more sensitive to the volume ratio than to the non-orthogonality angle. An relevant conclusion of this section, is that the FLS8 scheme has a low numerical accuracy penalization when using a non-orthogonality angle of 30° or an imposed displacement of 20%.

4.4. Scheme performance with boundary layer grids

Despite the proposed scheme being a diffusive one, its performance with grids specific for boundary layers can be relevant for

several fluid flow related computations. To test this aspect an analytical solution, that mimics an boundary layer velocity distribution, from the work of Eca et al. [65] was selected:

$$\phi(x, y) = u = \sum_{i=1}^3 \alpha_i \tanh(a_i Re^{1-b_i} x^{b_i} y) \quad (37)$$

where the used coefficients α_i , a_i and b_i are the ones presented in for the base flow of the reference. As for the Reynolds number, Re , it is equal to 5.0E6 in this test case.

As for grids specific to boundary layer problems, the ones generated by the NASA Langley Research Center [66] for the flat plate turbulent flow case were used in this section. To avoid any influence of cell size variation in the x direction, all cells that at the left

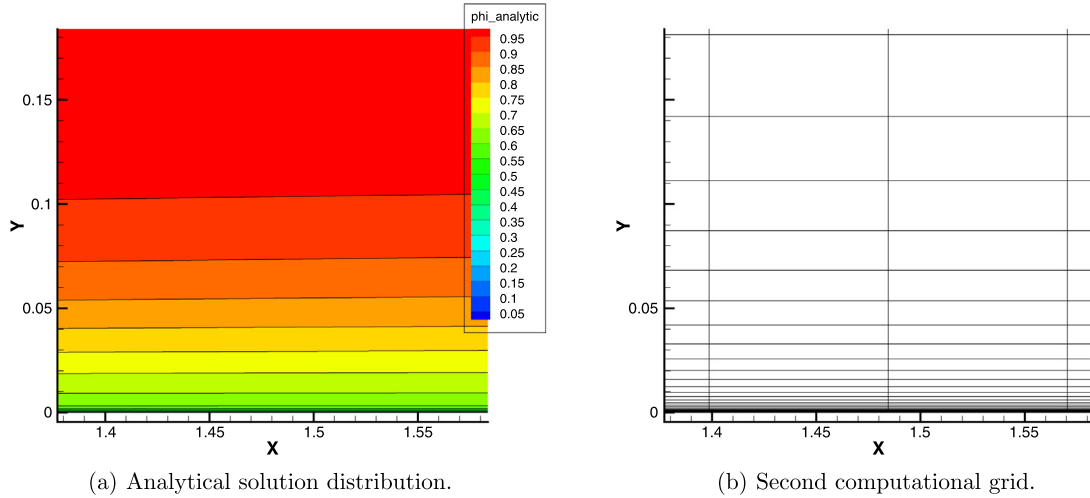


Fig. 16. Zoom of the analytical solution [65] and of the second computational grid of the database [66] used for the present boundary layer test case.

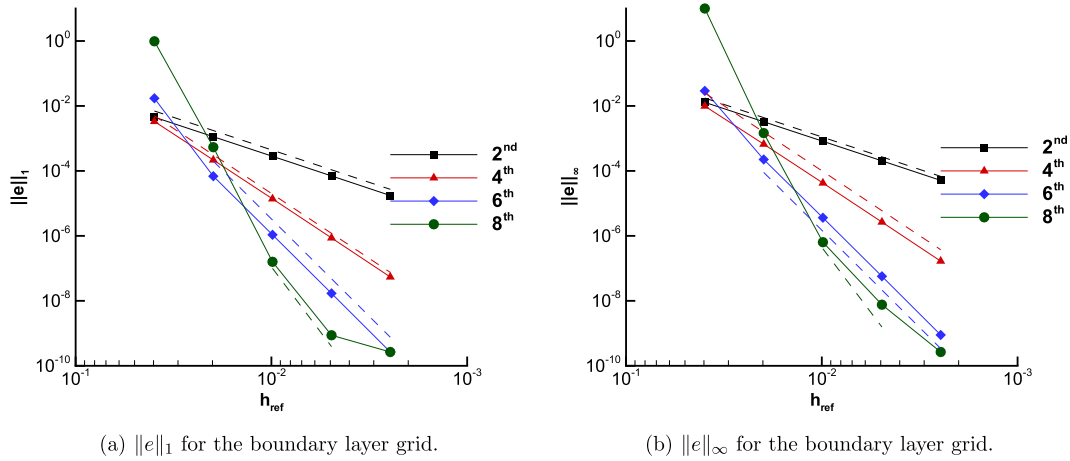


Fig. 17. Convergence curve of the $\|e\|_1$ and the $\|e\|_\infty$ for all schemes at study in the boundary layer grids from the [66] database and a specific analytical solution from [65].

of the line $X = 0.970084$ were deleted from these grids. This way, the final grids only present stretching in the y direction which are along place's normal. Notice that all grids of these database have at least a vertex located at this coordinate, which will define the left boundary of the used computational domain. Since the cells are now anisotropic, the boundary extension algorithm had to be modified to consider a different mean length in the x and y directions. Fig. 16 shows a zoom of the analytical solution and of the second computational grid used in this test case.

Fig. 17 shows the mean and maximum error for the four schemes at study using the analytical solution and the respective grids selected for the boundary layer problem. Both second and fourth order schemes converged quickly to the expected theoretical convergence order. While the sixth and eighth order schemes have converged to the expected order, they show an over-convergence behaviour for the initial grids. This is more evident in the case of the eighth order scheme. Tables 22 and 23 complement this analysis and show the computed convergence order between each successive grid refinement. This test case proves the ability of the current schemes, combined with their own stencil extension algorithm, to compute the viscous fluxes in a boundary layer like grid and with a similar u -velocity distribution.

4.5. Efficiency remarks

In this section the results of the efficiency study are presented. Two criteria are used: the error level as function of the number of non-zero (NNZ) entries of the global matrix, see Lipnikov and Manzini, [3], and the error level as function of solver-run time (SRT), see Ollivier-Gooch, [67].

4.5.1. Number of non-zeros (NNZ)

A considerable reduction of the error norms was obtained with the very high-order scheme, however the stencil is larger which increases the computational cost when using the same grid, consequently the efficiency is a prime concern of high-order numerical schemes. Denoting by N_z the number of non-zero entries in the global matrix, Lipnikov and Manzini, [3], have reported that the error norm decays as $\mathcal{O}(h^q) \sim \mathcal{O}(N_z^{q/D})$, where D is the space dimension. So for a two dimensional space, the error norm should decay $q/2$ with N_z . The order of convergence can be calculated by:

$$\mathcal{O}_n^z = \frac{\log_{10} \|e\|_{n_1} - \log_{10} \|e\|_{n_2}}{\log_{10} N_{z_1} - \log_{10} N_{z_2}}, \quad (38)$$

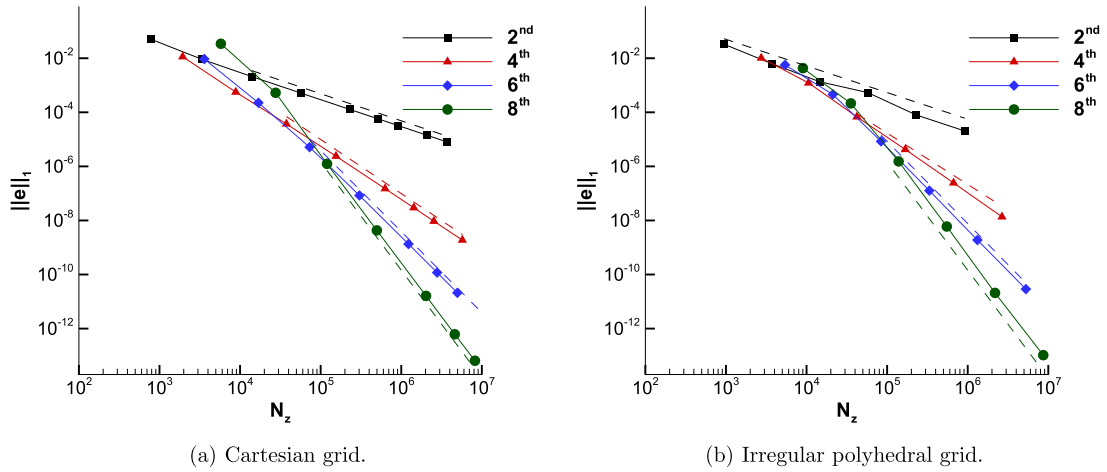


Fig. 18. Convergence curves of error norm-1 with N_z for the Cartesian and polyhedral grids with all schemes, where the dotted line represents the theoretical slope of each scheme.

Table 22

Obtained $\|e\|_1$ and \mathcal{O}_1 for the boundary layer grid.

Boundary layer grid		FLS2		FLS4		FLS6		FLS8	
n_c	h_{ref}	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1	$\ e\ _1$	\mathcal{O}_1
576	1.92E-02	1.13E-03	2.00	2.17E-04	3.92	6.87E-05	7.93	5.37E-04	10.79
2304	9.87E-03	2.82E-04	2.00	1.38E-05	3.97	1.08E-06	5.98	1.59E-07	11.71
9216	4.93E-03	7.02E-05	2.00	8.64E-07	4.00	1.70E-08	5.99	8.73E-10	7.51

Table 23

Obtained $\|e\|_\infty$ and \mathcal{O}_∞ for the boundary layer grid.

Boundary layer grid		FLS2		FLS4		FLS6		FLS8	
n_c	h_{ref}	$\ e\ _\infty$	\mathcal{O}_∞	$\ e\ _\infty$	\mathcal{O}_∞	$\ e\ _\infty$	\mathcal{O}_∞	$\ e\ _\infty$	\mathcal{O}_∞
576	1.92E-02	3.26E-03	2.12	6.66E-04	4.16	2.25E-04	7.49	1.46E-03	10.07
2304	9.87E-03	8.17E-04	2.06	4.24E-05	4.10	3.62E-06	6.15	6.39E-07	11.52
9216	4.93E-03	2.04E-04	2.03	2.66E-06	4.06	5.70E-08	6.09	7.60E-09	6.50

Table 24

\mathcal{O}_1^z and $\|e\|_1$ for the Cartesian grid with the FLS2, FLS4 and FLS8 schemes.

n_c	FLS2			FLS4			FLS8		
	N_z	$\ e\ _1$	\mathcal{O}_1^z	N_z	$\ e\ _1$	\mathcal{O}_1^z	N_z	$\ e\ _1$	\mathcal{O}_1^z
25600	2.28E+05	1.29E-04	1.00	6.30E+05	1.50E-07	1.97	2.03E+06	1.62E-11	3.97
57600	5.16E+05	5.72E-05	1.00	1.43E+06	2.96E-08	1.99	4.61E+06	6.17E-13	4.00
102400	9.18E+05	3.22E-05	1.00	2.54E+06	9.36E-09	1.99	8.21E+06	6.45E-14	3.90

Table 25

\mathcal{O}_1^z and $\|e\|_1$ the irregular polyhedral grid with FLS2, FLS4 and FLS8 schemes.

n_c	FLS2			FLS4			FLS8		
	N_z	$\ e\ _1$	\mathcal{O}_1^z	N_z	$\ e\ _1$	\mathcal{O}_1^z	N_z	$\ e\ _1$	\mathcal{O}_1^z
8321	5.77E+04	5.35E-04	0.68	1.68E+05	4.23E-06	2.01	5.50E+05	5.97E-09	4.04
33024	2.30E+05	7.97E-05	1.38	6.67E+05	2.45E-07	2.07	2.18E+06	2.09E-11	4.11
131585	9.19E+05	2.04E-05	0.98	2.66E+06	1.36E-08	2.09	8.66E+06	1.04E-13	3.84

where the superscript z means that the convergence order is related with N_z , which allows to estimate the amount of memory required for a given error level.

Table 24 lists the $\|e\|_1$ error decay for different N_z values and the respective convergence order, \mathcal{O}_1^z , with second, fourth and eighth-order schemes for Cartesian grids. Fig. 18(a) shows the results of $\|e\|_1$ with N_z for all schemes with the respective theoretical convergence order, which was achieved for every considered case.

Table 25 lists the error norm-1 as a function of N_z and the \mathcal{O}_1^z for FLS2, FLS4 and FLS8 schemes for the last three refinements of

the irregular polyhedral grid. Fig. 18(b) shows that the values of the convergence order are in close agreement with the theoretical values. These results support that a high-order method can be more efficient than a second-order one because it requires less computational resources to achieve the same accuracy level.

Table 26 lists the required computational resources to obtain a given error level in a Cartesian grid and in an irregular polyhedral grid with all schemes at study. The size of the global matrix is estimated by multiplying N_z with 12 bytes, 8 bytes to allocate the value and 4 bytes to allocate the position in the matrix. The ratio is computed by dividing the size of the matrix from the considered

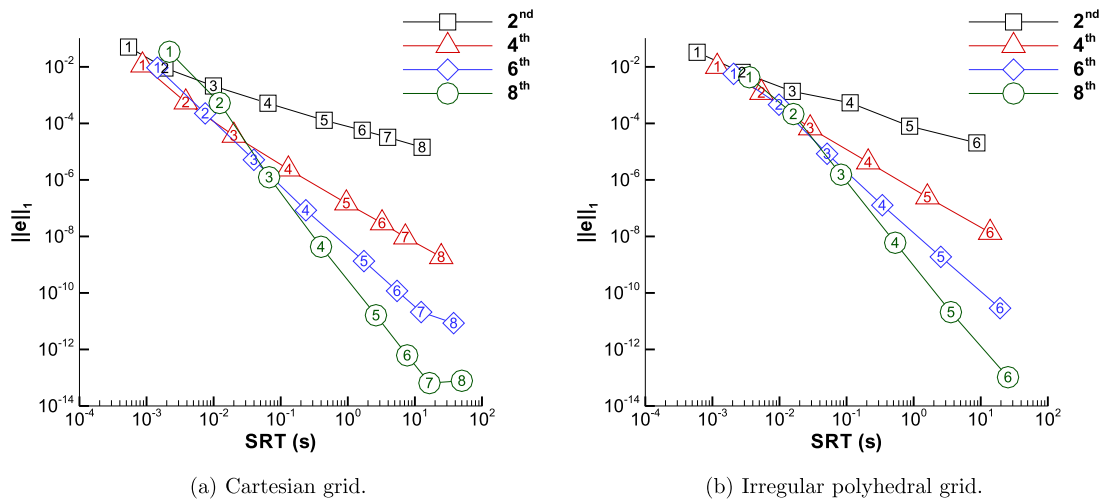


Fig. 19. Mean error with SRT diagram for the proposed schemes with Cartesian and irregular polyhedral grids.

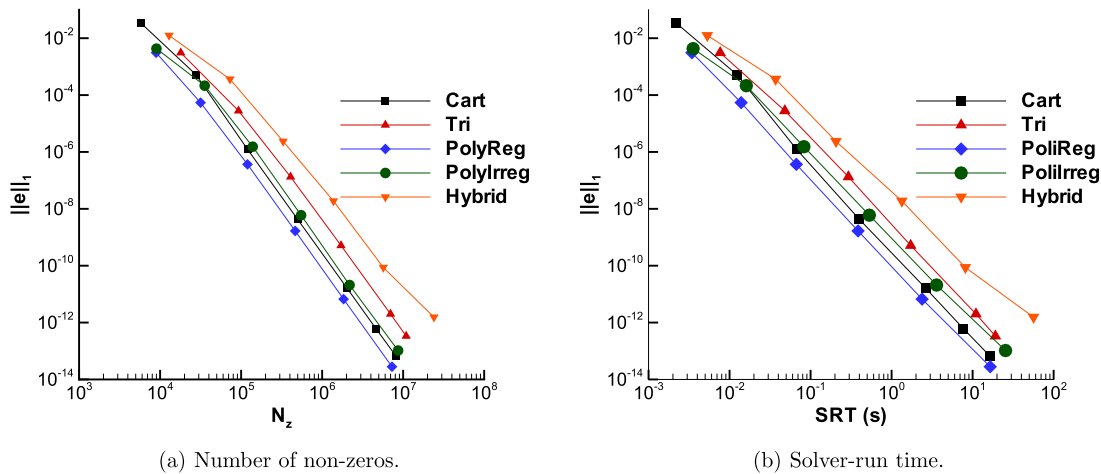


Fig. 20. Convergence curves of the error norm-1 with respect to N_z (a) and SRT (b) with the FLS8 scheme for the different grid types at study.

Table 26

Comparison of the required computational resources for all schemes with a norm-1 level error of $4.30\text{E}-09$ in a Cartesian grid and for $5.97\text{E}-09$ in an irregular polyhedral grid, where $1\text{Gb} = 1024\text{Mb}$.

Scheme	Cartesian grid $\ e\ _1 = 4.03\text{E}-09$				Irregular polyhedral grid $\ e\ _1 = 5.97\text{E}-09$			
	N_z	n_{cells}	size(A)	Ratio	N_z	n_{cells}	size(A)	Ratio
FLS8	4.98E+05	6.40E+03	5.70Mb	1.00	5.50E+05	8.32E+03	6.30Mb	1.00
FLS6	8.30E+05	1.76E+04	9.49Mb	1.66	9.08E+05	2.27E+04	10.40Mb	1.65
FLS4	3.77E+06	1.52E+05	43.17Mb	7.57	3.94E+06	1.95E+05	45.10Mb	7.16
FLS2	6.95E+09	7.74E+08	77.63Gb	13946.16	3.59E+09	5.13E+08	40.07Gb	6512.97

scheme by the size of the matrix from the FLS8 scheme. The results show that the global matrix for FLS8 scheme occupies three to four orders of magnitude less computational storage than the FLS2, 7 times less than the FLS4 and almost half of the space than the FLS6. These results show that the very high-order schemes will allow to obtain numerical solutions with higher accuracy than the second and fourth-order methods, when using the same storage memory.

In the previous table, the high selected error level is important when using transient simulations where the accumulation of spatial errors can caused deviations of the final numerical solution. For other types of applications, a higher error level may be enough for an acceptable accuracy. In Table 27, the same exercise is done

for an error level of $1.00\text{E}-04$. For the Cartesian grids, the FLS8 is not more advantageous than its fourth and sixth order counterpart. For the irregular polyhedral grid, the eight order scheme is still advantageous than the fourth one. Overall the sixth order scheme required less storage memory than the other schemes, and the FLS2 required almost 10 times the memory of the other high-order schemes. This last fact proves the advantages of using these type of schemes over the second-order ones.

4.5.2. Solver-run time (SRT)

Another way to analyse the method efficiency is through the computing time required to achieved a converged solution, see e.g. [67]. For this study the BICGSTAB solver was used, Fig. 19(a) and

Table 27

Comparison of the required computational resources for all schemes with a norm-1 level error of $1.00E-04$ in a Cartesian grid and an irregular polyhedral grid.

Scheme	Cartesian grid $\ e\ _1 = 1.00E-04$		Irregular polyhedral grid $\ e\ _1 = 1.00E-04$	
	N_z	Ratio	N_z	Ratio
FLS8	4.13E+04	1.00	2.87E+04	1.00
FLS6	1.23E+04	0.33	1.26E+04	0.44
FLS4	2.22E+04	0.54	3.21E+04	1.22
FLS2	2.95E+05	7.14	1.71E+05	5.96

Table 28

NNZ and SRT required for a $\|e\|_1$ of $1.00E-10$ for FLS8 and all grid types.

Grid	Number of non-zeros				Solver-run time	
	N_z	$\frac{N_z}{n_{cells}}$	size(A)	Ratio	SRT (s)	Ratio
Cart	1.29E+06	80.48	14.72Mb	1.37	1.42E+00	1.46
Tri	2.59E+06	119.27	29.59Mb	2.76	2.98E+00	3.05
PolyReg	9.37E+05	55.35	10.72Mb	1.00	9.77E-01	1.00
PolyIrreg	1.49E+06	65.83	17.04Mb	1.59	2.12E+00	2.17
Hybrid	5.49E+06	119.40	62.78Mb	5.86	7.76E+00	7.95

(b) show the results obtained for the Cartesian and irregular polyhedral grids, respectively.

Fig. 19 shows the errors level as function of the computing time required for convergence, the several used were enumerated. For a certain level of error, the FLS8 scheme displays the fastest time, being the fastest scheme for both grid types. The results shows that the error decays approximately with an $1/3$ power law of time, $\mathcal{O}(h^q) \sim \mathcal{O}(t_{solver}^{q/3})$, according to these results.

To obtain a level of error of $5.00E-09$, the SRT spent with the FLS8 is about half the obtained with FLS6, 30 times less than the FLS4 and 10 millions times less than the FLS2, these estimations are both valid for Cartesian and polyhedral grids.

4.5.3. Influence of grid type in the FLS8 efficiency

The efficiency of FLS8 for different grid types is compared, using the same approach of the previous sections. Fig. 20 shows evolution of the $\|e\|_1$ with respect to NNZ and SRT for the FLS8 but this time with all grids at study. Fig. 20(a) suggests that polyhedral grids optimize the computational resources and Fig. 20(b) suggests that faster solutions are obtained with regular polyhedral grids than with the other grid types. The triangular topology is the one with the worse behaviour, already expected because of its matrix's size.

Table 28 lists the computational resources required, for both criteria, for an error norm-1 of $1.00E-10$ for FLS8 and all grid types. The regular polyhedral grid is the one that has the lowest computational cost. The triangular grid requires more memory than the other grid types. This happens because each vertex connect to a high number of neighbouring triangles increasing the cells per stencil and consequently the required memory for the global matrix.

The combination that minimizes the SRT is the regular polyhedral grid applied to the FLS8 scheme, see Table 28, where the comparison of the SRT for all grid types is showed. The triangular grid is one of the most used grid types nowadays but the FLS8 with the polyhedral grid is almost three times faster than with the triangular one.

5. Conclusions

A new finite volume very high-order scheme is proposed for the solution of the Poisson equation and the computation of diffusive fluxes through the use of polynomial reconstructions. A detailed

description of the new stencil selection technique is explained, for both cases of interior and boundary faces, in order to reconstruct properly the high-order terms. A parametric study was performed to select the weight function of the WLS reconstruction method, which to the authors knowledge it is new for this type of methods.

The presented method only requires one reconstruction per face independently of the scheme's order. This is an advantage compared with other approaches that use a reconstruction per Gauss point. The main differences between this approach and the others ones is the point value formulation and reconstructions centered at face centroid. This can be considered as an advantage because it does not require an interpolation for the face flux that is used by the cell centered reconstructions.

The performed tests allowed to verify the effective convergence order of the second, fourth, sixth and eighth-order schemes flux based on reconstruction with the WLS method. In the framework of unstructured finite volume it is, on the Authors knowledge, the first time that 8th effective convergence order is reported for the FV Poisson equation solution with any arbitrary unstructured grid.

The weight function optimization allowed to decrease the numerical error of the method up to two orders of magnitude. An extension for Neumann boundary conditions is proposed and its impact in the convergence order is showed to be similar to the theoretical one. The grid quality study allowed to conclude that a non-orthogonality angle lower than 30° and a volume ratio lower than 5.44 it does not affect severely the accuracy of the solution, when considering a eight-order Fig. 12 scheme.

Finally, it is shown that the eight-order scheme is the one that requires less memory and solver-run time, when compared with the other schemes. Another major conclusion is that the polyhedral grid type is the one that showed superior results for the two efficiency at study. These results are expected because the stencils for this grid type are smaller and more compact than the ones obtained with the other grid types.

Acknowledgments

This work was supported by FCT, through IDMEC, under LAETA, project UID/EMS/ 50022/2013. The present results are also a contribution to the research project: High-order immersed boundary for moving boundary problems "HIBforMBP" with the reference PTDC/EME-EME/32315/2017 and supported by FCT.

References

- [1] Vermeire BC, Witherden FD, Vincent PE. On the utility of GPU accelerated high-order methods for unsteady flow simulations: a comparison with industry-standard tools. *J Comput Phys* 2017;334:497–521.
- [2] Zingg DW, Rango SD, Nemec M, Pulliam TH. Comparison of several spatial discretizations for the Navier-Stokes equations. *J Comput Phys* 2000;160:683–704.
- [3] Lipnikov K, Manzini G. A high-order mimetic method on unstructured polyhedral meshes for the diffusion equation. *J Comput Phys* 2014;272:360–85.
- [4] Lê TH, Gouez JML, Garnier E. High accuracy flow simulations: advances and challenges for future needs in aeronautics. *Comput Fluids* 2011;43(1):90–7.
- [5] Drikakis D, Kwak D, Kiris CC. Computational aerodynamics: advances and challenges. *Aeronaut J* 2016;120:13–36.
- [6] Barth T, Frederickson P. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. 28th Aerospace Sciences Meeting. Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics; 1990.
- [7] Ollivier-Gooch C, Altena MV. A high-Order-Accurate unstructured mesh finite-volume scheme for the advection-Diffusion equation. *J Comput Phys* 2002;181:729–52.
- [8] Michalak C, Ollivier-Gooch C. Unstructured high-Order accurate finite-volume solutions of the Navier-Stokes equations. 47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition. Aerospace Sciences Meetings. American Institute of Aeronautics and Astronautics; 2009.
- [9] Jalali A, Sharbatdar M, Ollivier-Gooch C. Accuracy analysis of unstructured finite volume discretization schemes for diffusive fluxes. *Comput Fluids* 2014;101:220–32.

- [10] Sejkan CB, Ollivier-Gooch CF. Improving finite-volume diffusive fluxes through better reconstruction. *Comput Fluids* 2016;139:216–32.
- [11] Ivan L, Groth CPT. High-order solution-adaptive central essentially non-oscillatory (CENO) method for viscous flows. *J Comput Phys* 2014;257:830–62.
- [12] Cueto-Felgueroso L, Colomina I, Nogueira X, Navarrina F, Casteleiro M. Finite volume solvers and moving least-squares approximations for the compressible Navier-Stokes equations on unstructured grids. *Comput Methods Appl Mech Eng* 2007;196:4712–36.
- [13] Toro EF, Hidalgo A. ADER Finite volume schemes for nonlinear reaction-diffusion equations. *Appl Numer Math* 2009;59(1):73–100.
- [14] Ollivier-Gooch C. Higher-order ENO schemes for unstructured meshes based on least-squares reconstruction. 35th Aerospace Sciences Meeting and Exhibit. American Institute of Aeronautics & Astronautics; 1997. Aerospace Sciences Meetings.
- [15] Hu C, Shu C-W. Weighted essentially non-oscillatory schemes on triangular meshes. *J Comput Phys* 1999;150:97–127.
- [16] Friedrich O. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. *J Comput Phys* 1998;144:194–212.
- [17] Visbal MR, Gaitonde DV. On the use of higher-order finite-difference schemes on curvilinear and deforming meshes. *J Comput Phys* 2002;181(1):155–85.
- [18] Deng X, Mao M, Tu G, Zhang H, Zhang Y. High-order and high accurate CFD methods and their applications for complex grid problems. *Commun Comput Phys* 2012;11(04):1081–102.
- [19] Baeza A, Mulet P, Zorío D. High order weighted extrapolation for boundary conditions for finite difference methods on complex domains with cartesian meshes. *J Sci Comput* 2016;69(1):170–200.
- [20] Pereira JMC, Kobayashi MH, Pereira JCF. A fourth-order-accurate finite volume compact method for the incompressible Navier Stokes solutions. *J Comput Phys* 2001;167(1):217–43.
- [21] Liu Y, Vinokur M, Wang ZJ. Spectral difference method for unstructured grids I: basic formulation. *J Comput Phys* 2006;216:780–801.
- [22] Wang ZJ, Liu Y, May G, Jameson A. Spectral difference method for unstructured grids II: extension to the euler equations. *J Sci Comput* 2007;32:45–71.
- [23] Mariano FP, Moreira LDQ, Silveira-Neto Ad, da Silva CB, Pereira JCF. A new incompressible Navier-Stokes solver combining Fourier pseudo-spectral and immersed boundary methods. *Comput Model Eng Sci (CMES)* 2010;59(2):181–216.
- [24] Marques AN, Nave J-C, Rosales RR. High-order solution of Poisson problems with piecewise constant coefficients and interface jumps. *J Comput Phys* 2017;335:497–515.
- [25] Bériot H, Prinn A, Gabard G. Efficient implementation of high-order finite elements for Helmholtz problems. *Int J Numer Methods Eng* 2016;106(3):213–40.
- [26] Cockburn B, Shu C-W. The local discontinuous Galerkin method for time-Dependent convection-Diffusion systems. *SIAM J Numer Anal* 1998;35(6):2440–63.
- [27] Yan J, Shu C-W. Local discontinuous Galerkin methods for partial differential equations with higher order derivatives. *J Sci Comput* 2002;17(1):27–47.
- [28] Arnold DN, Brezzi F, Cockburn B, Marini LD. Unified analysis of discontinuous Galerkin methods for elliptic problems. *J Numer Anal* 2002;39(5):1749–79.
- [29] Dumbser M, Käser M, Toro EF. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes V. Local time stepping and p-adaptivity. *Geophys J Int* 2007;171(2):695–717.
- [30] Nogueira X, Cueto-Felgueroso L, Colomina I, Gómez H, Navarrina F, Casteleiro M. On the accuracy of finite volume and discontinuous Galerkin discretizations for compressible flow on unstructured grids. *Int J Numer Methods Eng* 2009;78(13):1553–84.
- [31] Jaśkowiec J, Pluciński P, Stankiewicz A. Discontinuous Galerkin method with arbitrary polygonal finite elements. *Finite Elem Anal Des* 2016;120:1–17.
- [32] Vincent PE, Castonguay P, Jameson A. A new class of high-order energy stable flux reconstruction schemes. *J Sci Comput* 2011;47(1):50–72.
- [33] Boularas A, Clain SL, Baudoin F. A sixth-order finite volume method for diffusion problem with curved boundaries. *Appl Math Model* 2017;42:401–22.
- [34] Bertolazzi E, Manzini G. A unified treatment of boundary conditions in least-square based finite-volume methods. *Comput Math Appl* 2005;49(11):1755–65.
- [35] Droniou J. Finite volume schemes for diffusion equations: introduction to and review of modern methods. *Math Models Methods Appl Sci* 2014;24(08):1575–619.
- [36] Batty C. A cell-centred finite volume method for the Poisson problem on non-graded quadrees with second order accurate gradients. *J Comput Phys* 2017;331:49–72.
- [37] Chantasiwan S. Methods of fundamental solutions for time-dependent heat conduction problems. *Int J Numer Methods Eng* 2006;66(1):147–65.
- [38] Zhai S, Weng Z, Feng X. An adaptive local grid refinement method for 2D diffusion equation with variable coefficients based on block-centered finite differences. *Appl Math Comput* 2015;268:284–94.
- [39] Costa R, Clain S, Machado GJ. Finite volume scheme based on cell-vertex reconstructions for anisotropic diffusion problems with discontinuous coefficients. Springer International Publishing; 2014. p. 87–102. ISBN 978-3-319-09144-0.
- [40] Ramírez L, Nogueira X, Khelladi S, Chassaing J-C, Colomina I. A new high-order finite volume method based on moving least squares for the resolution of the incompressible navier-Stokes equations on unstructured grids. *Comput Methods Appl Mech Eng* 2014;278:883–901.
- [41] Guemond JL, Mineev P, Shen J. An overview of projection methods for incompressible flows. *Comput Methods Appl Mech Eng* 2006;195:6011–45.
- [42] Nejat A, Ollivier-Gooch C. A high-order accurate unstructured finite volume Newton-Krylov algorithm for inviscid compressible flows. *J Comput Phys* 2008;227(4):2582–609.
- [43] Ollivier-Gooch C, Jalali A. Higher-Order finite volume solution reconstruction on highly anisotropic meshes. In: 21st AIAA Computational Fluid Dynamics Conference, Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics; 2013.
- [44] Jalali A, Ollivier-Gooch C. Higher-order unstructured finite volume RANS solution of turbulent compressible flows. *Comput Fluids* 2017;143:32–47.
- [45] Cueto-Felgueroso L, Colomina I, Fe J, Navarrina F, Casteleiro M. High-order finite volume schemes on unstructured grids using moving least-squares reconstruction. application to shallow water dynamics. *Int J Numer Methods Eng* 2006;65:295–331.
- [46] Cueto-Felgueroso L, Colomina I. High-order finite volume methods and multiresolution reproducing kernels. *Arch Comput Methods Eng* 2008;15:185–228.
- [47] Nogueira X, Colomina I, Cueto-Felgueroso L, Khelladi S. On the simulation of wave propagation with a higher-order finite volume scheme based on reproducing Kernel methods. *Comput Methods Appl Mech Eng* 2010;199:1471–90.
- [48] Nogueira X, Cueto-Felgueroso L, Colomina I, Gómez H. Implicit large Eddy simulation of non-wall-bounded turbulent flows based on the multiscale properties of a high-order finite volume method. *Comput Methods Appl Mech Eng* 2010;199:615–24.
- [49] Nogueira X, Cueto-Felgueroso L, Colomina I, Navarrina F, Casteleiro M. A new shock-capturing technique based on moving least squares for higher-order numerical schemes on unstructured grids. *Comput Methods Appl Mech Eng* 2010;199:2544–58.
- [50] Chassaing J-C, Khelladi S, Nogueira X. Accuracy assessment of a high-order moving least squares finite volume method for compressible flows. *Comput Fluids* 2013;71:41–53.
- [51] Clain SL, Machado GJ, Pereira RMS. A new very high-order finite volume method for the 2D convection diffusion problem on unstructured meshes. IV Conferência Nacional em Mecânica dos Fluidos, Termodinâmica e Energia. Laboratório Nacional de Engenharia Civil; 2012.
- [52] Clain SL, Machado GJ, Nóbrega JM, Pereira RMS. A sixth-order finite volume method for multidomain convection-diffusion problem with discontinuous coefficients. *Comput Methods Appl Mech Eng* 2013;267:43–64.
- [53] Clain S, Machado G. A very high-order finite volume method for the time-dependent convection diffusion problem with Butcher Tableau extension. *Comput Math Appl* 2014;68(10):1292–311.
- [54] Guo F, Lü S. Unconditional stability of alternating difference schemes with intrinsic parallelism for two-dimensional fourth-order diffusion equation. *Comput Math Appl* 2016;71(10):1944–59.
- [55] Pina H. Métodos numéricos. 1st ed. McGraw-Hill de Portugal; 1995. ISBN 972-8298-04-8.
- [56] Magalhães JPP, Albuquerque DMS, Pereira JMC, Pereira JCF. Adaptive mesh finite-volume calculation of 2D lid-cavity corner vortices. *J Comput Phys* 2013;243:365–81.
- [57] Albuquerque DMS, Pereira JMC, Pereira JCF. Residual least-squares error estimate for unstructured h-adaptive meshes. *Numer Heat Transfer Part B* 2015;67(3):187–210.
- [58] Martins DMC, Albuquerque DMS, Pereira JCF. Continuity constrained least-squares interpolation for SFO suppression in immersed boundary methods. *J Comput Phys* 2017;336:608–26.
- [59] Martins DMC, Albuquerque DMS, Pereira JCF. On the use of polyhedral unstructured grids with moving immersed boundary method. *Comput Fluids* 2018;174:78–88.
- [60] Akin JE. Finite element analysis with error estimators. 1st ed. Elsevier; 2005. ISBN 978-0-7506-6722-2.
- [61] Chenoweth SKM, Soria J, Ooi A. A singularity-avoiding moving least squares scheme for two-dimensional unstructured meshes. *J Comput Phys* 2009;228:5592–619.
- [62] Demmel JW. Applied numerical linear algebra. SIAM; 1997. ISBN 978-0-89871-389-3.
- [63] Persson P-O, Strang G. A simple mesh generator in MATLAB. *SIAM Rev* 2004;46:329–45.
- [64] Juretić F, Gosman AD. Error analysis of the finite-volume method with respect to mesh type. *Numer Heat Transfer Part B* 2010;57(6):414–39.
- [65] Eca L, Hoekstra M, Vaz G. Manufactured solutions for steady-flow Reynolds-averaged Navier-Stokes solvers. *Int J Comput Fluid Dyn* 2012;26(5):313–32.
- [66] Rumsey C. Turbulence modeling resource, NASA Langley Research Center, http://turbmodels.larc.nasa.gov/flatplate_grids.html.
- [67] Nejat A, Ollivier-Gooch C. Effect of discretization order on preconditioning and convergence of a high-order unstructured newton-GMRES solver for the Euler equations. *J Comput Phys* 2008;227(4):2366–86.