

# Very high-order method on immersed curved domains for finite difference schemes with regular Cartesian grids

Javier Fernández-Fidalgo<sup>a,\*</sup>, Stéphane Clain<sup>b,\*</sup>, Luis Ramírez<sup>a</sup>, Ignasi Colominas<sup>a</sup>, Xesús Nogueira<sup>a,\*</sup>

<sup>a</sup> Group of Numerical Methods in Engineering, Universidade da Coruña, Campus de Elviña, 15071, A Coruña, Spain

<sup>b</sup> Centro de Matemática, Universidade do Minho, Campus de Azurém, 4800-058 Guimarães, Portugal

Received 2 April 2019; received in revised form 27 November 2019; accepted 7 December 2019

Available online xxx

## Abstract

A new very high-order technique for solving conservation laws with curved boundary domains is proposed. A Finite Difference scheme on Cartesian grids is coupled with an original ghost cell method that provide accurate approximations for smooth solutions. The technology is based on a specific least square method with restrictions that enables to handle general Robin conditions. Several examples in two-dimensional geometries are presented for the unsteady Convection–Diffusion equation and the Euler equations. A fifth-order WENO scheme is employed with matching fifth-order reconstruction at the boundaries. Arbitrary high-order reconstruction for smooth flows is achievable independently of the underlying differential equation since the method works as a black-box dedicated to boundary condition treatment.

© 2019 Elsevier B.V. All rights reserved.

**Keywords:** High-order schemes; Compressible flows; Cartesian grids; Finite difference; Embedded boundary; ROD

## 1. Introduction

Realistic problems take place in arbitrary shaped domains with arbitrary shaped objects inside, such as the wings of an aircraft or the blades of a turbine. Cartesian grid discretizations are very appealing in terms of efficiency and low memory storage but come across serious difficulties when prescribing boundary conditions on non polygonal domains.

Thus, very high-order approximations (of at least third-order) would degenerate into an at most second-order numerical solution if the boundary conditions are included just considering straight edges (for instance by using the edge center values or a simple ghost cell method). Very few technologies were developed to overcome such an issue to recover the optimal convergence order when dealing with curved interfaces or boundaries and most of them lead to a more complex formulation or present strong limitations.

The conventional structured grid approach is to discretize the governing equations (Convection–Diffusion, Euler or Navier–Stokes) on a curvilinear grid that conforms to the boundaries. This way, for simple enough boundary shapes, the task of imposing appropriate boundary conditions is greatly alleviated, since the boundary becomes

\* Corresponding authors.

E-mail addresses: [javier.fernandez1@udc.es](mailto:javier.fernandez1@udc.es) (J. Fernández-Fidalgo), [clain@math.uminho.pt](mailto:clain@math.uminho.pt) (S. Clain), [xnogueira@udc.es](mailto:xnogueira@udc.es) (X. Nogueira).

yet another grid line [1,2]. However, to handle more complex geometries, the grid generation quality turns to be a major issue and a multi-block approach has to be used in order to manipulate simpler geometries. This is still an interesting topic of research for high-order schemes [3–5] because the computed metrics of the transformation can introduce some errors that may impede to achieve the desired order of accuracy. In [6] it is proven that the transformed equations are in conservative form but the metric coefficients may introduce spurious source terms into the equations since the metric coefficients do not numerically satisfy the so-called metric identities, and the numerical representations of derivatives of uniform physical quantities are non-vanishing, thus, impeding high-order schemes to retain their order of accuracy. But there are alternative formulations for the metrics such as the one proposed in [7] that obtain less error.

The Immersed Boundary Method (IBM) provides a large class of methods to handle interfaces or boundaries within the Cartesian grid context. It presents some advantages over the previous approach on simulating flows with moving boundaries involving complicated shapes or topological changes. One subclass is the so-called “continuous forcing” approach deriving from the pioneer work of Peskin [8] and developed by several authors [9–11]. See [12,13] and the references therein for a detailed overview about the IBM. The method is at most of a second-order method due to the regularization of the Dirac  $\delta$  distribution over a small layer in the vicinity of the interface both for the spread and interpolation operations [14]. Nevertheless, it is important to mention the Immersed Boundary Smooth Extension (IBSE) method of Stein et al. [15] that achieves arbitrary order on smooth curved domains for the Laplace operator using Fourier spectral methods. They propose high-order smooth discretizations for the Dirac  $\delta$  distribution that achieve the desired accuracy and derivability, and allow the method to obtain arbitrary high order. The technique was also extended to incompressible flows [16].

Another approach within the IBM framework is the so-called Cartesian cut-cell approach [17,18], that cuts the solid bodies out of the background Cartesian grid. Within this approach, a finite volume solver conducts a flux balance around the edges of the cell. On one hand, strict conservation of mass, momentum and energy is achieved and thereby, the generation of spurious pressure fluctuations that are observed typically with ghost cell methods is avoided. On the other hand, cut cells may be arbitrarily small leading to very small time step due to the CFL constraint and some techniques, such as cell merging or cell mixing, have to be employed to overcome time step stability restrictions in case of an explicit scheme or an ill-conditioned matrix in case of an implicit scheme (see [19] and references therein). It is worth noting that these methods are also second-order accurate, achieving third or higher order in recent developments [20].

The Ghost Cell Immersed Boundary Method (GCIBM) appears at the end of the 1990s [21] and turns out to be a more efficient alternative for handling interfaces or boundaries [22]. In addition with the mirror/image point technique [23], the GCIBM has been widely used and improved in the last decade for compressible flow [24–27], incompressible flow [28–32], heat transfer [22,33–37] among others. Nevertheless, the ghost cell method suffers from two main drawbacks. First, conservation is not fully achieved (especially for moving boundaries) since, as explained by Ghias and coworkers in [23], the IBM can cut through the underlying mesh in an arbitrary manner. The main challenge is to treat the boundaries in a way that does not adversely impact the accuracy and conservation properties of the underlying scheme [38,39]. Secondly, most of the IBM could not go beyond second-order of accuracy.

Curiously, and up to the authors knowledge, very few studies have been dedicated to higher order GCIB methods. Gibou et al. proposed in [40] a fourth order extension of the GCIBM for the Laplace and Heat equation. They use a cubic extrapolation with a shifted interpolation towards the interior of the domain when a computational node is too close to the boundary in order to avoid the important deterioration of the polynomial representation. Another high-order extension was proposed in [41,42] for wave propagation in presence of obstacles. Contrarily to [40], the authors do not exactly fit the polynomial representation with the data but use a more robust Weighted Least-Square interpolation, including the Dirichlet or the Neumann condition in the cost functional. Unfortunately, no convergence test or numerical analysis of the method is provided to assess the accuracy of the method. Along the same lines of the previously described schemes is the work of Sjögreen and coworkers [43], where a technique based on Lagrange interpolation with a limiter which is restricted to second order methods and a single ghost cell is developed. This technique has been extended to high order and multiple ghost cells by Baeza and coworkers in [44,45].

The Inverse Lax–Wendroff method, introduced by Tan and Shu [46], and further developed in [47–49], is an alternative technique to prescribe boundary conditions on arbitrary shaped boundaries. Time derivatives of the boundary condition are converted into space derivatives using the Lax–Wendroff procedure and a Taylor expansion

around a given point of the boundary. This method is able to obtain fifth order convergence for the two-dimensional Euler equations. A crucial point to highlight is that the method explicitly uses the partial differential equation (PDE) and initial conditions in order to populate the ghost points and enforce the desired boundary conditions. It is worth noting that by differentiating the system of PDEs several times with respect to time, it may lead to very cumbersome expressions. In [50], the author propose a third-order accurate GCIBM for hyperbolic systems (Convection and Euler equations with Dirichlet and wall conditions) for one- and two-dimensional geometries with moving boundaries. The recent works of Wang et al. [51], that extends this technique to the simulation of detonation wave propagation for the two-step reactive Euler equations with source terms with complex obstacles; Vilar and Shu [52], that provide a rigorous stability analysis for this technique using different types of boundary conditions on the linear hyperbolic problems for central finite difference schemes; and Dakin et al. [53] for compressible flows using the Lagrange-remap approach are also worth mentioning.

In this work, we propose a new high-order procedure of imposing boundary conditions on arbitrary shaped boundaries. This technique is based on the work of S. Clain et al. [54,55] in finite volumes, where it was applied to the steady Convection–Diffusion equation. Here, we have extended their formulation to time-dependent equations and we develop a new, more flexible way of imposing the boundary conditions via a constrained least-squares polynomial fitting. This methodology works on both structured and unstructured grids, but in this work we will focus on finite difference methods using regular Cartesian grids, where the problem of using complex geometries while maintaining the expected order of accuracy is of special difficulty. This approach, along with the use of ghost points, allows the use of a unique interior scheme for the computations throughout the entire domain, removing the need of switching to biased schemes, usually more unstable than the interior schemes and more troublesome to adapt to boundary conditions. It is worth noting that the present approach is completely independent of the interior scheme, so it can be viewed as a black box that populates the ghost points when a set of appropriate conditions is given.

The structure of the paper is as follows. First, the required elements for the reconstruction process are defined, followed by the explanation of the reconstruction process itself. Then, the different sets of equations and boundary conditions are presented. Afterwards, several numerical examples are shown to prove that the presented technique can obtain arbitrary high-order. Finally, conclusions are drawn.

## 2. Physical and computational domains

Let  $\Omega$  be an open bounded set with  $\gamma_B$  distinct boundaries  $\{\Gamma_j\}_{j=1}^{\gamma_B}$  that are Lipschitz Jordan curves, regular piecewise. We label  $\Omega$  “physical domain” since it corresponds to the real domain where the continuous problem is defined. We introduce a rectangular subset  $\Lambda = [x_W, x_E] \times [y_S, y_N]$  large enough for containing the physical domain  $\Omega$ . An example of physical domain with holes embedded in a larger rectangle is displayed in Fig. 1. The outward unit normal vector defines the positive orientation of the curves such that the tangent and the normal vectors are positively oriented. Domain  $\Omega$  is implicitly given by  $\Omega = \{(x, y); \Psi(x, y) < 0\}$  where the level-set function  $\Psi$  is defined on the whole domain  $\Lambda$ . The boundary is obtained as the zero level-set where the outward normal vector at point  $P = (x_P, y_P)$  reads:

$$\mathbf{n}_P = \frac{\nabla \Psi(x_P, y_P)}{\|\nabla \Psi(x_P, y_P)\|}$$

The rectangular domain  $\Lambda = [x_W, x_E] \times [y_S, y_N]$  is partitioned with a uniform Cartesian grid of  $I \times J$  cells with spacings  $\Delta x = \frac{x_E - x_W}{I}$  and  $\Delta y = \frac{y_N - y_S}{J}$  along the  $x$  and  $y$  directions. For computational purposes, additional cell layers are necessary on each edge of  $\Lambda$  to guarantee that we have enough ghost cells around the domain in order to perform the calculations.

In this work a fifth order WENO scheme for the convective part of the equations will be used, and sixth order central finite differences for the diffusive part. In the present study,  $n_g = 3$  layers of ghost points are needed when dealing just with the convective terms, whereas  $n_g = 6$  are needed when dealing the convective and diffusive parts due to the fact that the second-order derivative is calculated as two iterated first-order derivatives. A brief description of the employed schemes is given in epigraph 5.

We label as  $N_I$  and  $N_J$  the total amount of points in each direction including the ghost layers:

$$N_I = I + 2n_g$$

$$N_J = J + 2n_g$$

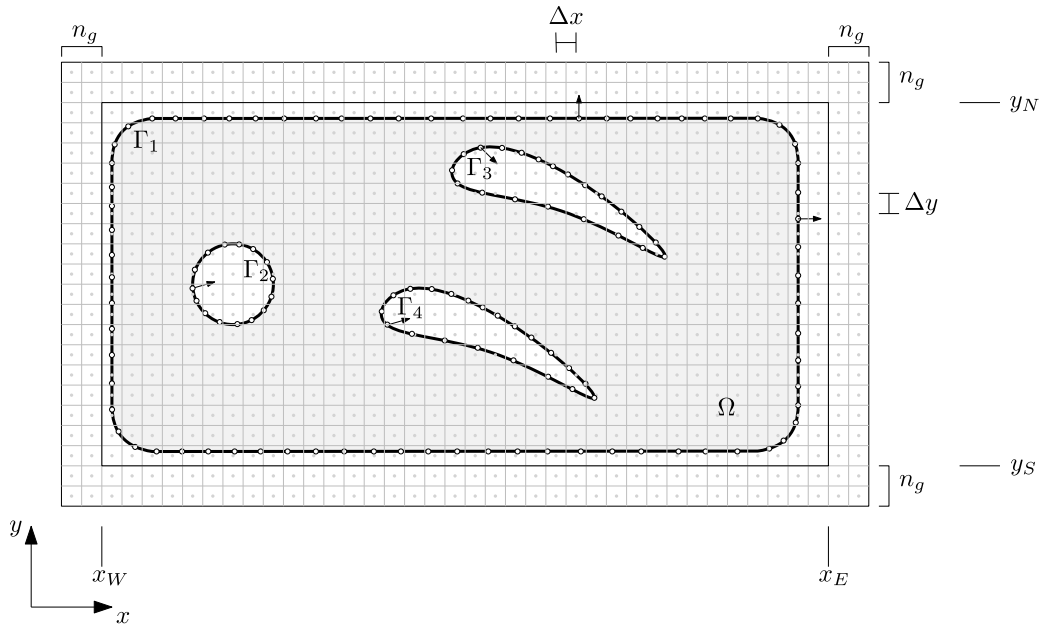


Fig. 1. Example of a two-dimensional physical domain,  $\Omega$ , enclosed by 4 boundaries denoted by  $\Gamma_j$ . Several outward normal vectors and collar points (denoted as hollow circles) for each boundary are plotted. Two additional ghost layers ( $n_g = 2$ ) are considered in this sketch.

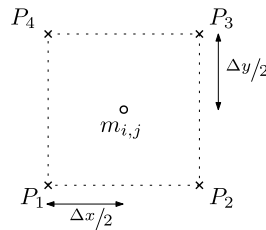


Fig. 2. Elements of the cell  $c_{ij}$ .

We denote by  $c_{i,j} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$  the cell of centroid  $m(c_{i,j}) = m_{i,j} = (x_i, y_j)$  while the half indices denote the interfaces  $e_{i-1/2,j}, e_{i+1/2,j}, e_{i,j-1/2}, e_{i,j+1/2}$ . Points  $\{P_k(c_{i,j})\}_{k=1}^4$  are the nodes of the cell (see Fig. 2). The grid is a collection of cells  $\mathcal{M} = \{c_{i,j}, i = 1, \dots, N_I, j = 1, \dots, N_J\}$

We characterize the computational domain  $\Omega_\Delta$  as the collection of cells where unknowns will be computed. For example, two scenarios, plotted in Fig. 3, have been considered depending on the level of cell inclusion. In the first one, the computational grid  $\mathcal{M}_\Delta = \{c \in \mathcal{M}_\Delta : m(c) \in \Omega\}$  is based on the centroid inclusion in the physical domain  $\Omega$ . The second one, defined by  $\mathcal{M}_\Delta = \{c \in \mathcal{M}_\Delta : \{P_k(c)\}_{k=1}^4 \in \Omega\}$ , requires that all the cell corners belong to the physical domain. The latter criterion is chosen. So the cells that are partially inside the domain are tagged as ghost cells. Once the collection of cells  $\mathcal{M}_\Delta$  is defined, we simply express the computational domain  $\Omega_\Delta$  as:

$$\Omega_\Delta = \bigcup_{c \in \mathcal{M}_\Delta} c$$

The ghost region,  $\Gamma_\Delta$  is an extension of the computational domain, whose thickness depends on the minimum number of ghost cells  $n_g$  on each direction, necessary so that the centered scheme can be used on any cell  $\mathcal{M}_\Delta$ .

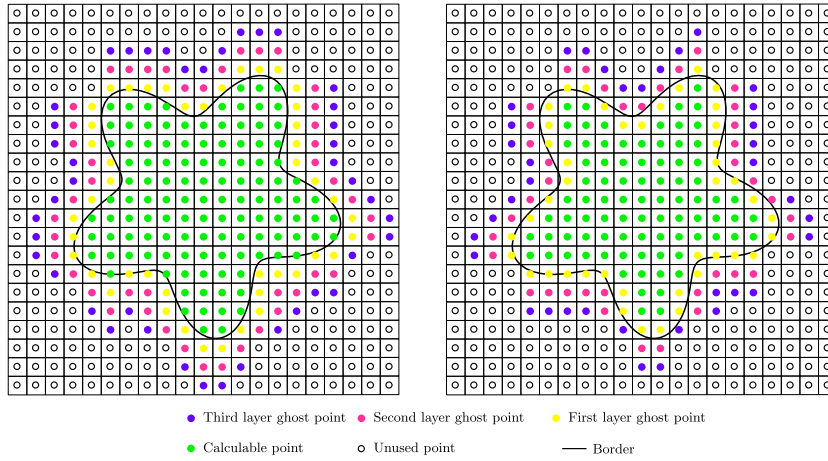


Fig. 3. Comparison between centroid inclusion (left) and cell inclusion (right) criteria. Three layers of ghost points are plotted.

We define the *rook distance*, denoted as  $d_{\mathbb{R}}$ , between two cells  $c_{i,j}$  and  $c'_{p,q}$ , as :

$$d_{\mathbb{R}}(c_{i,j}, c'_{p,q}) = \begin{cases} |q - j| & \text{if } i = p \\ |p - i| & \text{if } j = q \\ \infty & \text{otherwise} \end{cases} \quad (1)$$

This distance gives the number of crossed interfaces from cell  $c$  to cell  $c'$  either horizontally or vertically. If both cells are not on the same column or row, the cells are not related in any way. With the definition of Eq. (1) in hand, the distance between a cell  $c$  and domain  $\Omega_{\Delta}$  is given by

$$\mathcal{D}(c) = \operatorname{argmin}_{c' \subset \Omega_{\Delta}} d_{\mathbb{R}}(c, c').$$

We then define the collection of ghost cells  $c \in \mathcal{M}_{\Delta}^{\text{gh}}$  as

$$c \in \mathcal{M}_{\Delta}^{\text{gh}} \Leftrightarrow 1 \leq \mathcal{D}(c) \leq n_g$$

The ghost region can be partitioned into layers  $\mathcal{M}_{\ell}^{\text{gh}}$ ,  $\ell = \{1, \dots, n_g\}$  using  $\mathcal{D}(c) = \ell$  as displayed in Fig. 3. Please note that all the points of the grid that do not make part of the calculable domain or the ghost region are not taken into account in the calculations.

### 3. The reconstruction Off-site Data method

The Reconstruction Off-site Data (ROD) method was initially introduced for the finite volume method on unstructured meshes where the numerical flux computed on the computational domain faces takes into account the data localized on the physical boundary. In the original method, [54,55], no ghost cells are required but the algorithm has to check if we are dealing with a cell close to the boundary or not, leading to additional tests and reconstruction matrices. The key idea of this work is to use the same scheme for any cell. To this end, ghost cells will be filled with accurate approximations that fulfill the given set of boundary conditions using the ROD technique.

#### 3.1. The collar of points

The physical boundary is discretized into a collection of points where the boundary conditions will be prescribed, referred to as *collar of points*. The way we obtain these points is directly related to the type of mathematical expression that is used to describe the boundary. In the following epigraphs, two different approaches will be described. Namely, a level-set approach and a parametric approach. In both cases, the unitary normal vector  $\mathbf{n}$  and the unitary tangent vector  $\boldsymbol{\tau}$  have positive orientation, as shown in Fig. 4, if the normal vector points outward while the curve is traveled counterclockwise.

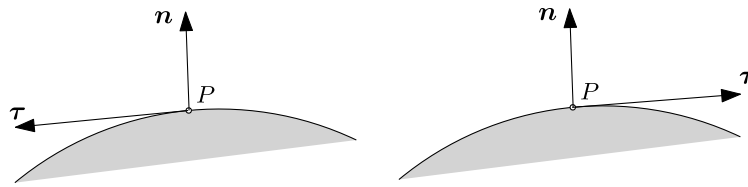


Fig. 4. The curve on the left has positive orientation while the curve on the right has negative orientation.

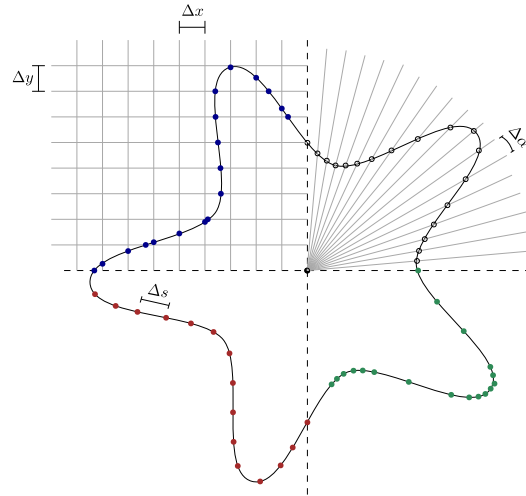


Fig. 5. Different possibilities for obtaining the collar points. Intersection with the grid (top left), intersection with constant angle polar grid (top right), equispaced distribution of points (bottom left), uneven spacing which depends on the curvature of the boundary (bottom right).

Following [53], the border, *i.e.* the physical boundary, is discretized into a collection of points  $\mathcal{C} = \{p_k\}_{k=1}^{N_C}$  that denote a collar of  $N_C$  distinct points, where the boundary conditions will be prescribed. Please note that if the boundary is a closed curve, the first and last point of the collar will coincide, so one of them will not be stored. Point  $p_k$  is expressed in the global coordinate system, denoted as  $(x_k^p, y_k^p)$ , and all the information regarding the unitary normal and tangent vectors and the signed local curvature, denoted respectively as  $\mathbf{n}_k, \boldsymbol{\tau}_k, \kappa_k$ , is stored.

Several procedures to set the collar will be discussed before-hand. A reasonable choice would seem to pick the intersections of the boundary with the grid lines as collar points, or to intersect it with a set of straight lines issued from the barycenter with equal angle spacing in the case of a convex-shaped border. These methods, as seen in the top left and top right parts of Fig. 5, may cause the collar points to be too close to each other, generating ill-conditioned matrices. These procedures are not recommended because the behavior of the point distribution is not controlled in any way, and highly depends on the orientation and shape of the domain. In the following, more robust algorithms for the collar construction are provided. These algorithms can obtain more suitable point distributions regardless of the shape of the boundary and depend solely on the mathematical description of the boundary. Namely, a level-set approach is described in epigraph 3.1.1, while in epigraph 3.1.2 a more general parametric approach is presented.

### 3.1.1. Level-set description

The level-set approach is the first method addressed to define the set of collar points that define the physical boundary. This method is fast and is our first choice in all the numerical examples of this work. However, it is only applicable when the boundary can be defined using an implicit equation.

Assuming that the boundary is defined as:

$$\Gamma = \{(x, y); \Psi(x, y) = 0\}.$$

**Input:** Point  $Q_0 = (x_0, y_0)$  near curve  
**Output:** Point  $p = (x^p, y^p)$  on curve

- 1 choose  $Q_0 = (x_0, y_0)$  near curve, set  $j \leftarrow 0$
- 2 **do**
- 3      $Q_{j+1}(x_{j+1}, y_{j+1}) \leftarrow Q_j(x_j, y_j) - \frac{\Psi(x_j, y_j)}{\|\nabla\Psi(x_j, y_j)\|^2} \nabla\Psi(x_j, y_j)$
- 4      $j \leftarrow j + 1$
- 5 **until**  $\|Q_{j+1} - Q_j\| < tol = 10^{-15}$
- 6  $x^p = x_{j+1}$      ;      $y^p = y_{j+1}$

Fig. 6. Point on Curve algorithm (PoC).

the collar of points is obtained by using the marching algorithm proposed in [56] where two subproblems need to be addressed, namely:

1. Find a curve point when a starting point in the vicinity of the curve is given. That is, given a point  $Q_0 = (x_0, y_0)$  in the vicinity of an implicit curve,  $\Psi(x, y) = 0$ , determine  $p$  that satisfies the level set equation  $\Psi(p) = 0$ .
2. Generate a sequence of curve points. That is, given a point  $p_k(x_k^p, y_k^p)$  which satisfies  $\Psi(p_k) = 0$ , determine another point  $p_{k+1}(x_{k+1}^p, y_{k+1}^p)$  that also satisfies the level-set equation,  $\Psi(p_{k+1}) = 0$ .

It is known that the normal and tangential vectors as well as the curvature can be computed from the level-set function as:

$$\begin{aligned}
 \mathbf{n} &= \frac{\nabla\Psi(x, y)}{\|\nabla\Psi(x, y)\|}, \\
 \boldsymbol{\tau} &= \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \mathbf{n}, \\
 \kappa &= \frac{-\left(\frac{\partial\Psi}{\partial y}\right)^2 \frac{\partial^2\Psi}{\partial x^2} + \frac{\partial\Psi}{\partial x} \frac{\partial^2\Psi}{\partial y \partial x} - \left(\frac{\partial\Psi}{\partial x}\right)^2 \frac{\partial^2\Psi}{\partial y^2}}{\left(\left(\frac{\partial\Psi}{\partial x}\right)^2 + \left(\frac{\partial\Psi}{\partial y}\right)^2\right)^{3/2}}.
 \end{aligned}$$

The first subproblem is solved using a Newton iteration method, as sketched in Fig. 6

The second subproblem, solved using the algorithm in Fig. 7, gives a set of points of arbitrary cardinality, that depends on a certain reference spacing  $\delta$  that in this work is selected as  $\Delta x$ .

The tolerance of the PoC algorithm is set to a value near machine precision (1E-15) because we want the collar points to lie on the curved boundary as accurately as possible. The choice on this tolerance value should not influence the overall accuracy as long as it is chosen low enough.

For the CP algorithm, the value  $\delta$  is related to the proximity of the collar points, since the lower the  $\delta$  value, the closer will be the points of the collar from each other. After some numerical experiments, we have determined that for the majority of the test cases in order to provide a quality discretization, the  $\delta$  value must be comparable to  $\Delta x$ . This situation, however, highly depends on the shape of the boundary. If more complicated, curved shapes are considered, the number of collar points that the use of  $\delta = \Delta x$  yields, could not be enough to capture the curvature of the shape and should be reduced.

In Fig. 8 a schematic representation of several iterations of the previously described algorithms can be seen.

### 3.1.2. Parametric description

For the cases where the level-set approach is not applicable, or when more control over the distribution of points is required, a parametric description can be used. This method allows to have more control over the collar point distribution at the cost of being more computationally demanding than the one of the previous section.

```

Input: Point  $p_1 = (x_1^p, y_1^p)$  on curve
Output: Set of points on curve  $c_k$ 
1 choose  $p_1 = (x_1^p, y_1^p)$  on curve; set  $k \leftarrow 1$ ; set  $(x_1, y_1) = (x_1^p, y_1^p)$ 
2 do
3   /* Evaluate tangent and normal vectors on point  $p_k$  as: */
4    $\mathbf{n}_k \leftarrow \frac{\nabla \Psi(p_k)}{\|\nabla \Psi(p_k)\|}$  ;  $\boldsymbol{\tau}_k \leftarrow (-\partial_y \Psi(p_k), \partial_x \Psi(p_k))$ 
5   /* Find new point outside the curve in the tangent direction */
6    $Q_k(x_k, y_k) \leftarrow p_k(x_k^p, y_k^p) + \delta \boldsymbol{\tau}_k$ 
7   /* Use the Point on Curve algorithm to find next point*/
8    $[Q_k] \rightarrow PoC \rightarrow p_{k+1}$ 
9    $k \leftarrow k + 1$ 
10 until  $\|p_k - p_1\| < \delta$  (and a few iterations have passed)
    
```

Fig. 7. Collar Points algorithm (CP).

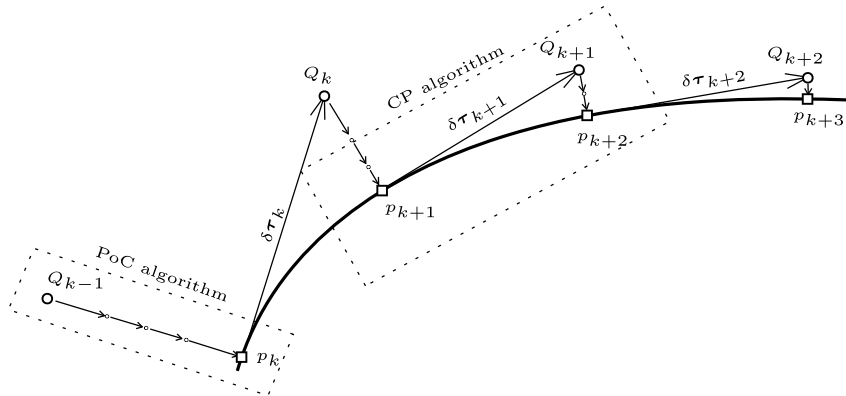


Fig. 8. Sketch of several iterations of the PoC and CP algorithms.

It is assumed now that the boundary  $\Gamma$  is a Jordan curve, that is the image of a continuous map  $t \in [0, 1] \mathbf{r}(t) = (x(t), y(t))^T \rightarrow \mathbb{R}^2$  such that  $\mathbf{r}(0) = \mathbf{r}(1)$  and the restriction on  $[0, 1)$  is one to one (no crossing point). Please note that if the considered curve is not closed, for example because the boundary is divided on several parts with different point distributions, the following expressions still apply. The only difference being that the first and last points will not be the same anymore.

We recall the formulas for tangent vector, normal vector and curvature of a parametric curve

$$\boldsymbol{\tau}(t) = \frac{\mathbf{r}'(t)}{\|\mathbf{r}'(t)\|}$$

$$\mathbf{n}(t) = \frac{\boldsymbol{\tau}'(t)}{\|\boldsymbol{\tau}'(t)\|} = \frac{\mathbf{r}'(t) \times \mathbf{r}''(t)}{\|\mathbf{r}'(t) \times \mathbf{r}''(t)\|}$$

$$\kappa(t) = \frac{\mathbf{r}'(t) \times \mathbf{r}''(t)}{\|\mathbf{r}'(t)\|^3}$$

We calculate the total length of the curve as

$$L = \int_0^1 \|\mathbf{r}'(t)\| dt = \int_0^1 \sqrt{(x'(t))^2 + (y'(t))^2} dt$$



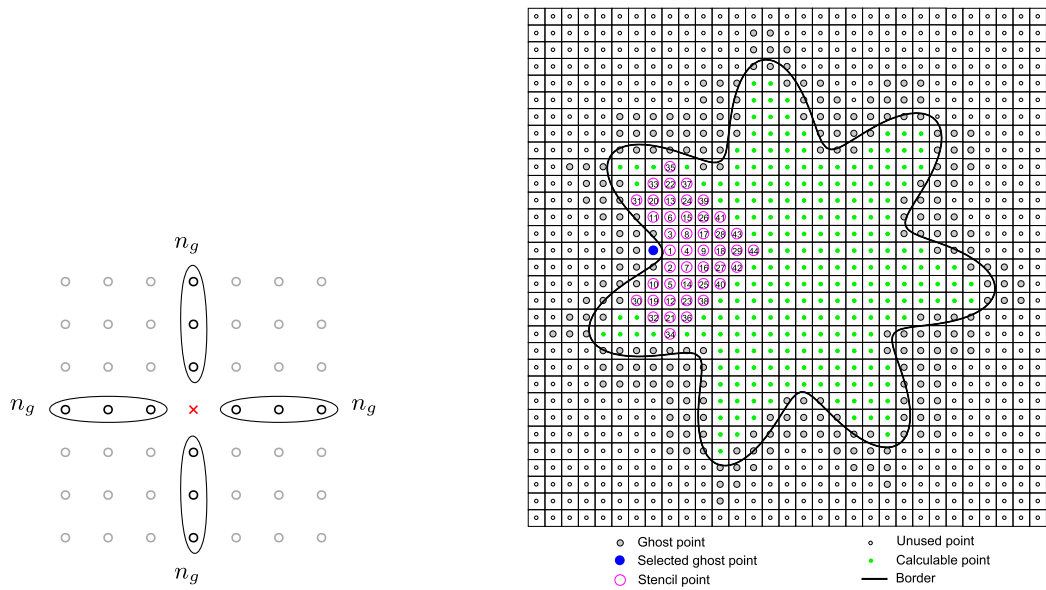


Fig. 9. Nearest ghost point search (left) and depiction of the stencil of a ghost point (right). The cell inclusion criterion is used.

and define  $n_s$  equidistant points with length  $\Delta s = \frac{L}{n_s-1}$  taking the first point at  $t = 0$ , and the last one at  $t = 1$ . One has to determine the parameters  $\{t_k^*\}_{k=1}^{n_s}$  such that:

$$\int_0^{t_k^*} \sqrt{(x'(t))^2 + (y'(t))^2} dt = (k-1)\Delta s \quad k = 1, 2, \dots, n_s. \quad (2)$$

Eq. (2) is numerically solved using a root-finding iterative method (such as the secant, Newton–Raphson or Steffensen algorithms) combined with some numerical quadrature (trapezoid or Simpson’s rule) resulting in the set  $\{t_k^*\}_{k=1}^{n_s}$  and by using the function  $r(t_k^*)$  the corresponding coordinates of the points can be obtained. In Fig. 5 an equidistant point distribution can be seen in the bottom left part. An interesting variant of this approach could be by using a curvature-based procedure we could cluster more points on the areas with larger curvature, as seen in the bottom right part of Fig. 5, but this variant will not be discussed here.

### 3.2. Ghost cell evaluation

Values at the ghost cell centroids are obtained via the Reconstruction Off-site Data procedure which requires two ingredients: the definition of a stencil over the computational domain together with the data located on the collar points that will be taken into account by the polynomial.

#### 3.2.1. Definition of the stencil

Let  $c \in \mathcal{M}_\Delta^{\text{gh}}$  be a ghost cell of centroid  $m$  (see Fig. 9 right). We denote by  $\mathcal{S}_c$  the associated stencil composed of cell  $c' \in \mathcal{M}_\Delta$  situated in the neighborhood of  $c$ . The number of cells in the stencil is related to the order of the two-dimensional  $n$ th degree polynomial reconstruction and, as a rule of thumb, we take  $s = |\mathcal{S}_c|$  as the closest integer to  $\frac{3}{2} \frac{(n+1)(n+2)}{2}$ . Notice that the closest calculable cells are located within a rook length of  $n_g$  (see Fig. 9 left), drastically reducing the computational time spent in the neighbors search.

#### 3.2.2. Collar points selection

Boundary conditions are introduced in the polynomial reconstruction through two collar points that are determined in the following way. We select the nearest collar point to the centroid  $m$ , using the usual euclidean distance:

$$p_1 = \underset{p \in \mathcal{C}}{\operatorname{argmin}} |p - m|.$$

From point  $p_1$ , we label the two neighbor collar points  $p$  and  $q$  respectively and select  $p_2 \in \{p, q\}$  that satisfies the criteria

$$(p_2 - m)(p_1 - m) > 0, \quad (p_2 - m)(p_2 - p_1) > 0.$$

### 3.3. Polynomial reconstruction

A polynomial reconstruction function based on the restricted Least-square sense provides the ghost cell value. We denote by  $\phi_{ij}$  the approximations of an underlying function  $\phi$  at centroids  $m_{ij}$  with  $c_{ij} \in \mathcal{M}_\Delta$ . We need to satisfy the generic boundary condition

$$\alpha(p)\phi(p) + \beta(p)\nabla\phi(p) \cdot \mathbf{n} = g(p), \quad p \in \Gamma, \tag{3}$$

where  $g$  and parameters  $\alpha$  and  $\beta$  are given functions on the boundary.

For a given ghost cell  $c$  with centroid  $m$ , we seek a polynomial

$$\tilde{\phi}_c(x, y) = \boldsymbol{\psi}_c^T(x, y) \cdot \mathbf{a}$$

that is defined as the product of a polynomial basis  $\boldsymbol{\psi}_c(x, y)$  and the unknown coefficient vector  $\mathbf{a}$

$$\boldsymbol{\psi}_c(x, y) = \left\{ \left( \frac{x - x_m}{\Delta x} \right)^{v_1} \left( \frac{y - y_m}{\Delta y} \right)^{v_2} \right\}, \quad |v| \leq n$$

where  $v = (v_1, v_2)$  is a multi-index and  $\mathbf{a} = \{a_v\}$  are the coefficients of the polynomial that approximates the data over the stencil  $\mathcal{S}_c$  and satisfies the constraint (3) on the two collar points  $p_1$  and  $p_2$  exactly.

Relation (3) then reads, in matrix form

$$\overbrace{\begin{pmatrix} \alpha(p_1)\boldsymbol{\psi}_c^T(p_1) + \beta(p_1)\nabla\boldsymbol{\psi}_c^T(p_1) \cdot \mathbf{n}(p_1) \\ \alpha(p_2)\boldsymbol{\psi}_c^T(p_2) + \beta(p_2)\nabla\boldsymbol{\psi}_c^T(p_2) \cdot \mathbf{n}(p_2) \end{pmatrix}}^c \cdot \mathbf{a} = \begin{pmatrix} g(p_1) \\ g(p_2) \end{pmatrix} = \mathbf{g}. \tag{4}$$

On the other hand, we fit the polynomial  $\tilde{\phi}_c$  on the centroids of the stencil leading to an over-determined system of linear equations  $\boldsymbol{\Psi}_c \cdot \mathbf{a} = \boldsymbol{\varphi}_c$  given by

$$\boldsymbol{\psi}_c^T(m_\ell) \cdot \mathbf{a} = \phi_\ell, \quad c_\ell \in \mathcal{S}_c,$$

we solve in the least-square sense, subject to constraint (4). Notice that index  $\ell$  is a local numbering associated to stencil  $\mathcal{S}_c$  to provide a compact matrix structure.

We rewrite the Linear Constrained Least Squares (LCLS) problem under the compact form

$$\begin{aligned} \min_{\mathbf{a}} \|\boldsymbol{\Psi}_c \cdot \mathbf{a} - \boldsymbol{\varphi}_c\|_2^2 \\ \text{subject to } \mathbf{C} \cdot \mathbf{a} = \mathbf{g} \end{aligned}$$

The Lagrange Multipliers technique is employed since it manages the equality constraints in a flexible way. Firstly we define the function to be minimized, that is, the sum of the squared difference between the values of the function  $\phi_\ell$  and the values of the reconstruction polynomial  $\tilde{\phi}_c$  at each cell  $c_\ell$  of the stencil  $\mathcal{S}_c$ .

$$E(\mathbf{a}) = \sum_{c_\ell \in \mathcal{S}_c} \frac{1}{2} (\phi_\ell - \tilde{\phi}_c(m_\ell))^2 = \sum_{c_\ell \in \mathcal{S}_c} \frac{1}{2} (\phi_\ell - \boldsymbol{\psi}_c^T(m_\ell) \cdot \mathbf{a})^2$$

To build the Lagrangian function, we add the constraints of the form of Eq. (3) (that need to be satisfied at the two collar points) times the Lagrange multiplier vector  $\boldsymbol{\lambda}^T = (\lambda_1, \lambda_2)$ . This can be expressed as:

$$\mathcal{L}(\mathbf{a}, \boldsymbol{\lambda}) = E(\mathbf{a}) + \sum_{k=1,2} \lambda_k (\alpha\tilde{\phi}_c + \beta\nabla\tilde{\phi}_c \cdot \mathbf{n} - g)(p_k). \tag{5}$$

Eq. (5) can be rewritten in a more compact form as:

$$\mathcal{L}(\mathbf{a}, \boldsymbol{\lambda}) = E(\mathbf{a}) + \boldsymbol{\lambda}^T (\mathbf{C} \cdot \mathbf{a} - \mathbf{g}) \tag{6}$$

Computing the derivative of the Lagrangian function with respect to  $\mathbf{a}$  and equating to zero, yields the following expression

$$\Psi_c^T \Psi_c \cdot \mathbf{a} + \mathbf{C}^T \cdot \boldsymbol{\lambda} = \Psi_c^T \boldsymbol{\varphi}_c. \tag{7}$$

The derivative of the Lagrangian function with respect to  $\boldsymbol{\lambda}$  equated to zero, yields the already known restriction of Eq. (4). Coupling equations (7) and (4), we obtain the following matrix expression for the LCLS problem.

$$\left( \begin{array}{c|c} \Psi_c^T \Psi_c & \mathbf{C}^T \\ \hline \mathbf{C} & 0 \end{array} \right) \begin{pmatrix} \mathbf{a} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \Psi_c^T \boldsymbol{\varphi}_c \\ \mathbf{g} \end{pmatrix}. \tag{8}$$

The existence of a unique solution is guaranteed assuming that  $\Psi_c$  has full rank. This matrix, which holds the coordinates of the points in the stencil, is a Vandermonde matrix. Thus, it has full rank if and only if all the points considered in the stencil are distinct. Eq. (8) is then solved with

$$\begin{pmatrix} \mathbf{a} \\ \boldsymbol{\lambda} \end{pmatrix} = \underbrace{\left( \begin{array}{c|c} \Psi_c^T \Psi_c & \mathbf{C}^T \\ \hline \mathbf{C} & 0 \end{array} \right)^{-1} \begin{pmatrix} \Psi_c^T & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I} \end{pmatrix}}_{\mathbf{M}} \begin{pmatrix} \boldsymbol{\varphi}_c \\ \mathbf{g} \end{pmatrix}.$$

Notice that the reconstructed value for the ghost cell at point  $m$  is  $\tilde{\varphi}_c(m) = a_1$  so the first row of  $\mathbf{M}$  can be pre-stored beforehand for each ghost point in the mesh, saving a huge amount of computational effort. Since the condition number of the involved matrices could be fairly high, the accuracy of the solution of (8) obtained using a direct solver (such as LU decomposition) may be compromised. To alleviate this problem, we use a QL factorization, which is a variant of the well-known QR factorization (see [57,58]). This technique obtains more accurate results when dealing with matrices with high condition numbers.

#### 4. Model equations and boundary conditions

Different equations or systems along with several boundary conditions will be tested in the numerical section. The two-dimensional Convection–Diffusion equation is a fundamental prototype to experiment since most of the linearized problems result into a combination of this two fundamental operators. The equation has the usual form:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi - \mu \nabla \phi) = s,$$

where  $\phi = \phi(x, y, t)$  is the sought function,  $\mathbf{u} = (u, v)^T$  is the velocity vector,  $\mu$  is the diffusion coefficient and  $s = s(x, y, t)$  is a source term.

On the other hand, the two-dimensional Euler system is a very representative nonlinear problem that concerns a lot of applications. Equations are written under the conservation form

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} = \mathbf{0},$$

where  $\mathbf{U}$  denotes the vector of conservative variables while  $\mathbf{F}$  and  $\mathbf{G}$  are the inviscid flux-vectors expressed as

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho E + p)u \end{pmatrix}, \quad \mathbf{G} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho E + p)v \end{pmatrix}.$$

Function  $\rho$  is the density,  $(u, v)$  are the velocity components along the  $x$  and  $y$  axes,  $p$  is the pressure, and

$$E = \frac{P}{\rho(\gamma - 1)} + \frac{1}{2}(u^2 + v^2)$$

stands for the total energy per unit mass with  $\gamma$  being the ratio of specific heats of the gas/fluid (for an ideal, monoatomic gas,  $\gamma = 7/5$ ).

For the Convection–Diffusion problem, the standard Dirichlet, Neumann and Robin type boundary conditions will be considered as given in Eq. (4). For the Euler equations, the Solid Wall BC is of special interest. Following [59] and [13], two alternative sets of compatible boundary conditions for a curved, non-moving solid wall are the following

$$\left\{ \begin{array}{l} u_n = 0 \\ \frac{\partial \rho}{\partial \mathbf{n}} = \left( \frac{\rho}{\gamma p} \right) \frac{\partial p}{\partial \mathbf{n}} \\ \frac{\partial p}{\partial \mathbf{n}} = -\rho u_\tau^2 \kappa \\ \frac{\partial u_\tau}{\partial \mathbf{n}} = u_\tau \kappa \end{array} \right. \tag{9}$$

$$\left\{ \begin{array}{l} u_n = 0 \\ \frac{\partial S}{\partial \mathbf{n}} = 0 \\ \frac{\partial H}{\partial \mathbf{n}} = 0 \\ \frac{\partial u_\tau}{\partial \mathbf{n}} = u_\tau \kappa. \end{array} \right. \tag{10}$$

In the above expressions,  $u_n$  and  $u_\tau$  are the normal and tangential velocity components with respect to the wall while  $\kappa$  stands for the local curvature of the wall. If the center of curvature is placed along the positive direction of the normal vector, then  $\kappa$  is taken as positive. All the variables involved in Eq. (9) have to be evaluated at the wall, hence the order of accuracy of the whole boundary condition relies on the accuracy of the discretization of the normal derivatives and the quality of the approximation of the values of the flow on the RHS. However, Eq. (9) can be rewritten in the form of Eq. (10), by using the entropy  $S$  and enthalpy  $H$  defined as:

$$S = \frac{p}{\rho^\gamma} \quad H = \left( \frac{\gamma}{\gamma - 1} \right) \frac{p}{\rho} + \frac{1}{2} (u^2 + v^2)$$

With Eq. (10), only the curvature at the wall is required, since the tangential velocity is prescribed as a Robin-type boundary condition. Since we are using a level-set approach we know a smooth distribution of normal and tangent vectors at every point of the mesh, so each velocity vector on every point of the stencil is projected into its normal and tangential components by using the vectors calculated with the level-set, and those components are used separately to reconstruct the normal and tangent velocities, respectively. This way, we avoid creating a discontinuity as the traditional mirror technique does, and very high-order of accuracy can be achieved for smooth flows.

## 5. Numerical schemes

As previously noted, for all the problems (unless otherwise stated), a fifth WENO scheme will be employed for the inviscid reconstruction, sixth order central finite differences for the diffusive fluxes, a fifth order ROD reconstruction for the boundaries, and six layers of ghost points. In the following, a brief description of the employed schemes is given.

### 5.1. Spatial discretization

The implementation of the fifth-order WENO scheme follows the original of Jiang and Shu in [60] with the mappings proposed by Henrick et al. later on in [61]. This scheme is usually called WENO5M and its implementation can also be consulted in [62]. In the following, a brief description for the scalar version of the scheme will be given, but we encourage the interested reader to check the aforementioned references for a more in-depth description an generalization to the Euler equations.

The key idea behind this scheme is to obtain a high-order non-oscillatory interpolation at the interfaces  $e_{i\pm 1/2,j}$  and approximate the derivative, in a dimension by dimension fashion, as

$$\left. \frac{\partial h}{\partial x} \right|_{i,j} = \frac{h_{i+1/2,j} - h_{i-1/2,j}}{\Delta x}$$

dropping the  $j$  index for simplicity, since the same procedure applies for the other dimensions, the WENO5 scheme calculates the  $h_{i+1/2}$  as a linear combination of three candidate stencils:

$$h_{i+1/2} = \omega^1 h_{i+1/2}^1 + \omega^2 h_{i+1/2}^2 + \omega^3 h_{i+1/2}^3$$

where

$$h_{i+1/2}^1 = \frac{2}{6}h_{i-2} - \frac{7}{6}h_{i-1} + \frac{11}{6}h_i$$

$$h_{i+1/2}^2 = -\frac{1}{6}h_{i-1} + \frac{5}{6}h_i + \frac{2}{6}h_{i+1}$$

$$h_{i+1/2}^3 = \frac{2}{6}h_i + \frac{5}{6}h_{i+1} - \frac{1}{6}h_{i+2}$$

and the nonlinear weights  $\omega^k$  are computed as

$$\omega^k = \frac{\alpha^k}{\alpha^1 + \alpha^2 + \alpha^3}, \quad \alpha_k = \frac{d^k}{(\varepsilon + \beta^k)^2} \quad k = 1, 2, 3$$

where  $\varepsilon = 10^{-6}$ , and the optimal coefficients for this scheme are

$$d^1 = 1/10, d^2 = 6/10, d^3 = 3/10.$$

Finally, the smoothness detectors can be written as:

$$\beta^1 = \frac{13}{12} (h_{j-2} - 2h_{j-1} + h_j)^2 + \frac{1}{4} (h_{j-2} - 4h_{j-1} + 3h_j)^2$$

$$\beta^2 = \frac{13}{12} (h_{j-1} - 2h_j + h_{j+1})^2 + \frac{1}{4} (h_{j-1} - h_{j+1})^2$$

$$\beta^3 = \frac{13}{12} (h_j - 2h_{j+1} + h_{j+2})^2 + \frac{1}{4} (3h_j - 4h_{j+1} + h_{j+2})^2$$

The WENO5M is used with a global Lax–Friedrichs flux splitting and the characteristic version is employed in the case of the Euler equations.

For the diffusive part of the Convection–Diffusion equation a standard centered sixth order Finite Difference scheme is used. The derivative can be expressed as:

$$\left. \frac{\partial h}{\partial x} \right|_{i,j} = \sum_{k=-3}^3 s_k h_{i+k,j}$$

where the coefficients (that fulfill the property  $s_{-k} = -s_k$ ) are

$$s_0 = 0, \quad s_1 = 3/4, \quad s_2 = -3/20, \quad s_3 = 1/60.$$

As before, the same procedure can be applied in a dimension-by-dimension fashion.

Please note that by using a ghost point approach, the WENO5M scheme behaves near the boundaries in the exact same way as in the interior of the domain.

### 5.2. Temporal integration

For the temporal integration, the method of lines (MoL) is employed. Thus, the PDE can be transformed into a system of Ordinary Differential Equations (ODEs) by solving for the temporal derivative. It reads:

$$\frac{\partial \mathbf{Y}}{\partial t} = \mathbf{R}$$

where  $Y$  in this case can be either  $\phi$  in the 2D Convection–Diffusion equation of  $U$  in the 2D Euler equations, and  $R$  stands for the remaining right-hand side of the corresponding PDE.

With the PDE expressed in this fashion, a fourth-order Runge–Kutta scheme is employed to advance the solution in time from time step  $n$  to  $n + 1$  as:

$$\begin{aligned} Y^{(1)} &= Y^{(n)} + \frac{1}{2} \Delta t R^{(n)} \\ Y^{(2)} &= Y^{(n)} + \frac{1}{2} \Delta t R^{(1)} \\ Y^{(3)} &= Y^{(n)} + \Delta t R^{(2)} \\ Y^{(n+1)} &= Y^{(n)} + \frac{1}{6} \Delta t (R^{(n)} + 2R^{(1)} + 2R^{(2)} + R^{(3)}) \end{aligned}$$

where the short-hand notation  $R^{(k)} \equiv R(Y^{(k)})$  has been used.

### 6. Numerical tests

In order to assess the present approach, several test cases are run and the  $L_1$ ,  $L_2$  and  $L_\infty$  error norms are employed to measure the error of the obtained solution with respect to the analytic one. These norms for a scalar function  $\phi$  read

$$\begin{aligned} L_1(\phi) &= \frac{1}{|\Omega_\Delta|} \sum_{c_k \in \mathcal{M}_\Delta} |\phi_k| |c_k|, \\ L_2(\phi) &= \left( \frac{1}{|\Omega_\Delta|} \sum_{c_k \in \mathcal{M}_\Delta} |\phi_k|^2 |c_k| \right)^{\frac{1}{2}}, \\ L_\infty(\phi) &= \max_{c_k \in \mathcal{M}_\Delta} |\phi_k|, \end{aligned}$$

where  $\phi_k = \phi(m_k)$  stands for any approximation while  $|c_k|$  is the measure of the cell. Similar definition holds for domain  $\Gamma_\Delta$ .

#### 6.1. Reconstruction order assessment

A first validation is carried out to check that, assuming exact values of function  $\phi$  at the centroid of  $c_k \in \mathcal{M}_\Delta$  and at the collar points, we get an accurate approximation of  $\phi$  at the ghost cells’ centroid after the reconstruction. To this end, we consider two domains  $\mathcal{D}_S$  and  $\mathcal{D}_N$  depicted in Fig. 10, that are implicitly defined with the level-set functions

$$\begin{aligned} \mathcal{D}_S &= \{(x, y) \in \mathbb{R}^2; \quad \Psi_S(x, y) < 0\}, \\ \mathcal{D}_N &= \{(x, y) \in \mathbb{R}^2; \quad \Psi_{N_1}(x, y) \Psi_{N_2}(x, y) < 0\}, \end{aligned}$$

where functions  $\Psi_S$ ,  $\Psi_{N_1}$  and  $\Psi_{N_2}$  are given by

$$\begin{aligned} \Psi_S(x, y) &= x^2 + y^2 - \left[ 0.8 + 0.2 \cos \left( 7 \left( \arctan \left( \frac{y}{x} \right) - \frac{\pi}{4} \right) \right) \right]^2, \\ \Psi_{N_1}(x, y) &= x^2 + y^2 - \left[ 0.85 + 0.15 \cos \left( 5 \left( \arctan \left( \frac{y}{x} \right) - \frac{\pi}{3} \right) \right) \right]^2, \\ \Psi_{N_2}(x, y) &= x^2 + y^2 - \left[ 0.35 + 0.10 \cos \left( 4 \left( \arctan \left( \frac{y}{x} \right) - \frac{\pi}{4} \right) \right) \right]^2. \end{aligned}$$

The smooth function  $\phi(x, y) = \sin(x + y) \cos(x - y)$  is used to compute a Robin-type boundary condition  $g(p) = \alpha \phi(p) + \beta \nabla \phi(p) \cdot \mathbf{n}(p)$  taking  $\alpha = \beta = 1$  in the examples. The reconstruction process is carried out onto the ghost cells and the exact error of the reconstruction is measured using the aforementioned norms. Collar points are constructed in two ways: (1) using Eq. (2) and referred to as *evenly spaced collar*; (2) PoC algorithm described in algorithm 7 (see Fig. 11).

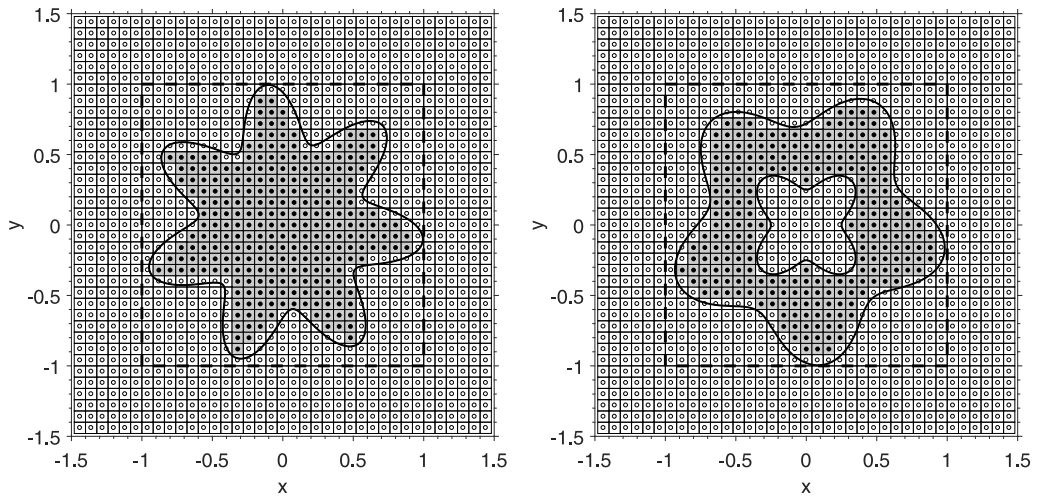


Fig. 10. Sketches of domains  $\mathcal{D}_S$  (left) and  $\mathcal{D}_N$  (right). The calculable domain is shadowed for a  $25 \times 25$  grid. The original domain  $A$  is enclosed within a dashed line, outside that region the ghost layers can be seen.

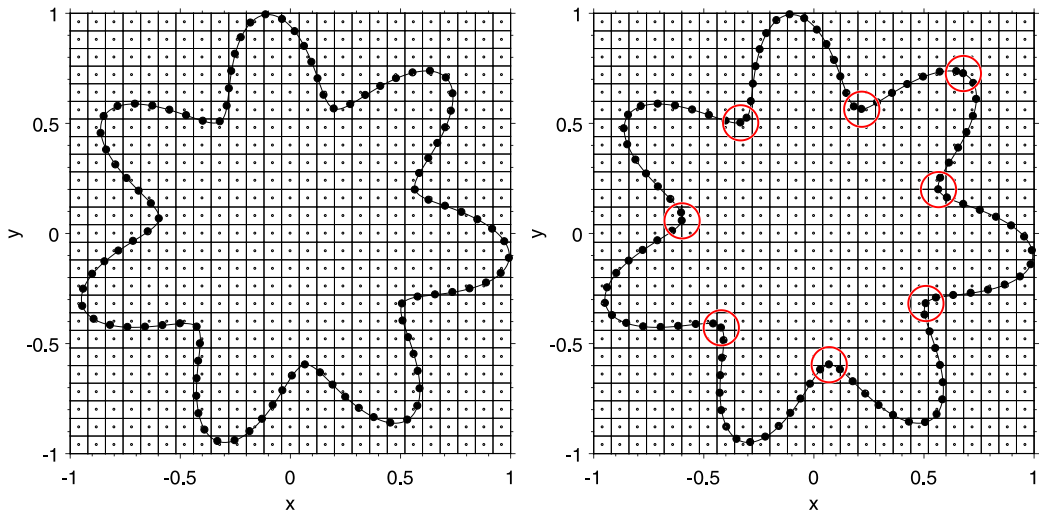


Fig. 11. Obtained collar of points for the same reference length  $s = \frac{1}{25}$  using the evenly spacing technique (left) and the PoC algorithm (right). The zones where the PoC algorithm produces a more uneven distribution are marked with a circle.

We report in Tables 1 and 2 the errors obtained with the three norms for the two methods. We get almost identical errors even though the points are not located in the same places. The fifth order of the reconstruction is retained. In the following examples we will use the PoC algorithm unless otherwise noted because it is faster in the computations.

### 6.2. Two-dimensional scalar convection–diffusion equation

Let us consider the smooth function  $\phi(x, y, t) = e^{-t} \sin(x + y) \cos(x - y)$ . For a constant velocity and diffusion parameter, the analytic source term reads

$$s(x, y, t) = \frac{1}{2} e^{-t} (2u \cos 2x + 2v \cos 2y + (4\mu - 1) (\sin 2x + \sin 2y)).$$

We test several kinds of boundary conditions to assess the accuracy of the method. We begin with the simple advection problem by setting  $\mathbf{u} = (1, 1)$  and  $\mu = 0$ . We prescribe the Dirichlet boundary condition on the

**Table 1**  
Results for 5th order ROD reconstruction with a 32 point stencil for  $\mathcal{D}_S$  domain.

Domain $\mathcal{D}_S$	Evenly spaced collar						PoC Algorithm					
	$L_1$	$L_2$	$L_\infty$	$r_1$	$r_2$	$r_\infty$	$L_1$	$L_2$	$L_\infty$	$r_1$	$r_2$	$r_\infty$
25	1.15E-04	2.53E-04	2.64E-03	-	-	-	1.51E-04	3.39E-04	2.75E-03	-	-	-
50	2.71E-06	4.84E-06	2.33E-05	5.42	5.71	6.82	2.79E-06	4.98E-06	2.43E-05	5.76	6.09	6.83
100	8.85E-08	1.54E-07	6.73E-07	4.94	4.97	5.12	8.98E-08	1.58E-07	6.92E-07	4.96	4.98	5.13
200	2.65E-09	4.72E-09	2.27E-08	5.06	5.03	4.89	2.67E-09	4.75E-09	2.26E-08	5.07	5.05	4.94
400	8.30E-11	1.48E-10	8.00E-10	5.00	4.99	4.83	8.35E-11	1.49E-10	7.97E-10	5.00	4.99	4.83
800	2.55E-12	4.56E-12	2.66E-11	5.02	5.02	4.91	2.55E-12	4.56E-12	2.79E-11	5.03	5.03	4.84

**Table 2**  
Results for 5th order ROD reconstruction with a 32 point stencil for  $\mathcal{D}_N$  domain.

Domain $\mathcal{D}_N$	Evenly spaced collar						PoC Algorithm					
	$L_1$	$L_2$	$L_\infty$	$r_1$	$r_2$	$r_\infty$	$L_1$	$L_2$	$L_\infty$	$r_1$	$r_2$	$r_\infty$
25	8.97E-05	1.65E-04	9.53E-04	-	-	-	1.05E-04	2.13E-04	1.48E-03	-	-	-
50	2.71E-06	4.89E-06	2.28E-05	5.05	5.08	5.39	2.67E-06	4.75E-06	2.56E-05	5.31	5.49	5.85
100	7.62E-08	1.37E-07	7.29E-07	5.15	5.16	4.97	7.73E-08	1.38E-07	6.79E-07	5.11	5.10	5.24
200	2.47E-09	4.47E-09	2.29E-08	4.95	4.94	4.99	2.49E-09	4.53E-09	2.41E-08	4.96	4.93	4.81
400	7.55E-11	1.36E-10	7.58E-10	5.03	5.04	4.92	7.55E-11	1.36E-10	7.31E-10	5.04	5.06	5.04
800	2.33E-12	4.20E-12	2.82E-11	5.02	5.02	4.75	2.34E-12	4.21E-12	2.75E-11	5.01	5.01	4.73

**Table 3**  
Errors and convergence rates for the pure convection case  $\mathbf{u} = (1, 1)$  and  $\mu = 0$  with a Dirichlet boundary condition.

Mesh	$\mathcal{D}_S$						$\mathcal{D}_N$					
	$L_1$	$L_2$	$L_\infty$	$r_1$	$r_2$	$r_\infty$	$L_1$	$L_2$	$L_\infty$	$r_1$	$r_2$	$r_\infty$
25	3.31E-05	9.17E-05	6.42E-04	-	-	-	3.28E-05	4.77E-05	2.10E-04	-	-	-
50	4.05E-07	5.84E-07	2.25E-06	6.35	7.29	8.16	5.30E-07	7.10E-07	2.39E-06	5.95	6.07	6.46
100	1.26E-08	1.97E-08	8.21E-08	5.00	4.89	4.78	2.03E-08	2.88E-08	1.59E-07	4.71	4.62	3.91
200	4.19E-10	6.78E-10	3.31E-09	4.91	4.86	4.63	5.67E-10	8.19E-10	4.19E-09	5.16	5.14	5.24
400	1.27E-11	2.12E-11	1.26E-10	5.05	5.00	4.71	1.77E-11	2.62E-11	1.42E-10	5.00	4.96	4.88
800	3.78E-13	6.07E-13	3.77E-12	5.07	5.13	5.07	5.43E-13	8.16E-13	5.26E-12	5.02	5.01	4.76

aforementioned domains  $\mathcal{D}_S$  and  $\mathcal{D}_N$ , and report in Table 3 the errors and convergence rates. We mention that the expected order of accuracy is retained for both types of curved domains for all the norms. We recall that the mesh does not fit the physical domain, but the boundary is fully recovered thanks to the ROD method.

To highlight the impact of the ROD technique in the boundary treatment and the benefits in using very high-order schemes, we compare in Fig. 12 the computational time versus  $L_2$ -norm of the error for the second- and fifth-order ROD method while preserving the WENO 5 scheme for the calculation cells. The computational time required to obtain a given error is significantly lower (several orders of magnitude) using the fifth-order ROD method than the second-order one.

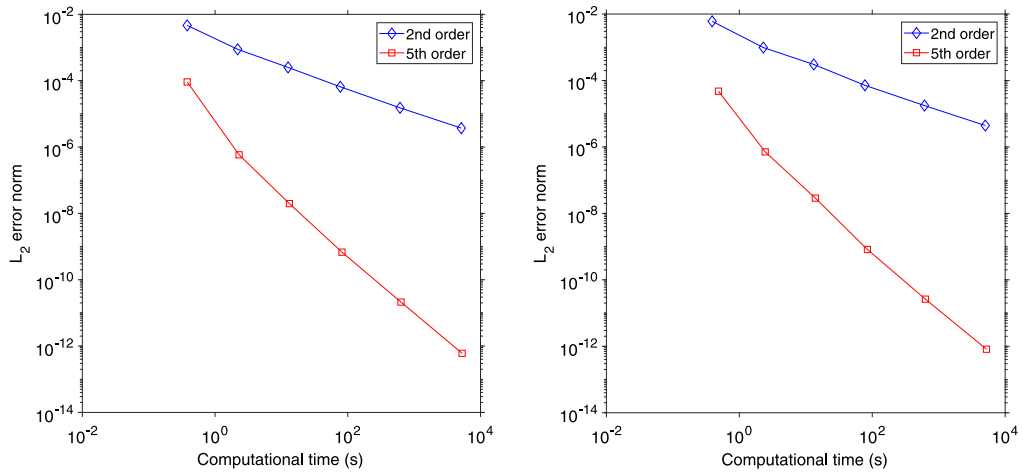
The second test deals with the Heat equation setting  $\mathbf{u} = (0, 0)$  and  $\mu = 1$ . In Table 4 the errors and convergence rates for the Dirichlet boundary condition are presented, while in Table 5 are the Neumann boundary condition results. For the Dirichlet boundary condition, optimal orders are obtained with a sixth-order finite difference for the diffusive part and a fifth-order method for the ROD. On the other hand, we observe a loss of one order of accuracy for the Neumann boundary condition. We state that if a truly  $n$ th order reconstruction is desired for the Neumann BC, a  $(n + 1)$ th-degree polynomial has to be used.

The last test case, proposed in [54], investigates the effects of a boundary layer on the reconstruction. The domain is represented in Fig. 13 and is made up of two concentric circles of radii  $r_E = 1.0$  and  $r_I = 0.5$  respectively. The velocity is a radial function with constant modulus  $u$  and the diffusion  $\mu = 1$  is normalized.

A manufactured solution with a boundary layer on both boundaries is stated as

$$\phi(x, y) = a \left( e^{ur'(x,y)} + e^{-ur'(x,y)} + b \right),$$





**Fig. 12.** Pure convection. Dirichlet boundary condition. Computational time vs.  $L_2$  error norm plot for  $\mathcal{D}_S$  domain (left) and  $\mathcal{D}_N$  domain (right).

**Table 4**

Results for the Heat equation taking  $\mathbf{u} = (0, 0)$  and  $\mu = 1$  with Dirichlet BC.

Mesh	$\mathcal{D}_S$						$\mathcal{D}_N$					
	$L_1$	$L_2$	$L_\infty$	$r_1$	$r_2$	$r_\infty$	$L_1$	$L_2$	$L_\infty$	$r_1$	$r_2$	$r_\infty$
25	1.65E-05	2.88E-05	2.03E-04	-	-	-	3.68E-05	5.99E-05	3.24E-04	-	-	-
50	3.09E-07	4.62E-07	3.59E-06	5.74	5.96	5.82	2.54E-07	4.23E-07	4.01E-06	7.18	7.15	6.34
100	5.08E-09	7.41E-09	8.81E-08	5.92	5.96	5.35	4.53E-09	6.16E-09	5.60E-08	5.81	6.10	6.16
200	1.28E-10	1.66E-10	2.11E-09	5.31	5.48	5.38	1.19E-10	1.60E-10	1.40E-09	5.26	5.26	5.32
400	2.42E-12	3.70E-12	5.69E-11	5.73	5.49	5.21	3.68E-12	4.80E-12	6.32E-11	5.01	5.06	4.47

**Table 5**

Results for the Heat equation taking  $\mathbf{u} = (0, 0)$  and  $\mu = 1$  with Neumann BC.

Mesh	$\mathcal{D}_S$						$\mathcal{D}_N$					
	$L_1$	$L_2$	$L_\infty$	$r_1$	$r_2$	$r_\infty$	$L_1$	$L_2$	$L_\infty$	$r_1$	$r_2$	$r_\infty$
25	2.12E-04	4.25E-04	2.80E-03	-	-	-	2.36E-04	2.84E-04	6.04E-04	-	-	-
50	5.51E-06	7.18E-06	2.10E-05	5.27	5.89	7.06	1.20E-05	1.41E-05	3.29E-05	4.30	4.33	4.20
100	2.78E-07	3.80E-07	1.33E-06	4.31	4.24	3.99	6.70E-07	7.91E-07	1.92E-06	4.16	4.16	4.10
200	1.76E-08	2.47E-08	8.23E-08	3.98	3.95	4.01	3.93E-08	4.56E-08	1.03E-07	4.09	4.12	4.22
400	9.32E-10	1.33E-09	4.51E-09	4.24	4.22	4.19	2.41E-09	2.67E-09	5.77E-09	4.03	4.09	4.15

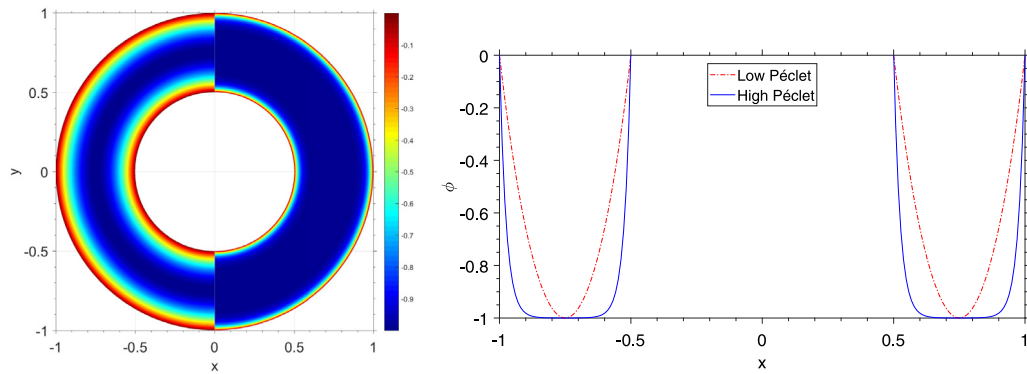
where

$$a = \frac{1}{e^u + e^{-u} - 2},$$

$$b = -e^u - e^{-u},$$

$$r'(x, y) = \frac{2\sqrt{x^2 + y^2} - (r_E + r_I)}{r_E - r_I}$$

while the source term is analytically computed. Following [54], we simulate a low Péclet number  $Pe = u/\mu$  situation by setting  $u = 1$ , and a high Péclet number situation with  $u = 10$ . Errors and convergence rates are presented in Table 6. In both cases, we report the optimal convergence errors but with a noticeable difference between low and high Péclet cases due to the boundary layer size. Indeed, the coarse meshes hardly achieve a correct resolution in



**Fig. 13.** Steady-state circular Convection–Diffusion. On the left, the annulus domain and surface plot of the manufactured solutions for  $u = 1$  (Low Pe) case (left half) and  $u = 10$  (High Pe) (right half) are presented. On the right a cut along the horizontal axis is shown.

**Table 6**  
Results for the steady-state circular convection–diffusion.

Mesh	Low Pe						High Pe					
	$L_1$	$L_2$	$L_\infty$	$r_1$	$r_2$	$r_\infty$	$L_1$	$L_2$	$L_\infty$	$r_1$	$r_2$	$r_\infty$
50	7.44E-05	1.09E-04	8.15E-04	–	–	–	9.85E-02	1.01E-01	1.90E-01	–	–	–
100	1.86E-06	3.02E-06	3.34E-05	5.32	5.17	4.61	8.22E-03	8.91E-03	2.99E-02	3.58	3.50	2.67
200	8.01E-08	1.32E-07	2.12E-06	4.54	4.51	3.98	5.61E-04	6.08E-04	2.85E-03	3.87	3.87	3.39
400	2.58E-09	4.06E-09	7.74E-08	4.95	5.03	4.77	3.43E-05	3.58E-05	1.73E-04	4.03	4.09	4.05
800	7.85E-11	1.30E-10	3.30E-09	5.04	4.97	4.55	1.27E-06	1.32E-06	7.65E-06	4.75	4.76	4.49

the latter situation resulting into rough approximations and large errors. Finer meshes are required to recover the full order.

### 6.3. Euler equations

The ROD technique is extended to systems of conservation laws either by reconstructing the conserved variables’ vector  $\mathbf{U} = (\rho, \rho u, \rho v, \rho E)^T$  or the primitive vector  $\mathbf{W} = (\rho, u, v, p)^T$ . The choice of the variables for the reconstruction depends on the nature of the boundary conditions. Moreover, in some specific situations, quantities such as entropy or normal velocity would be considered for the ROD method in place of the traditional primitive variables.

#### 6.3.1. Entropy wave

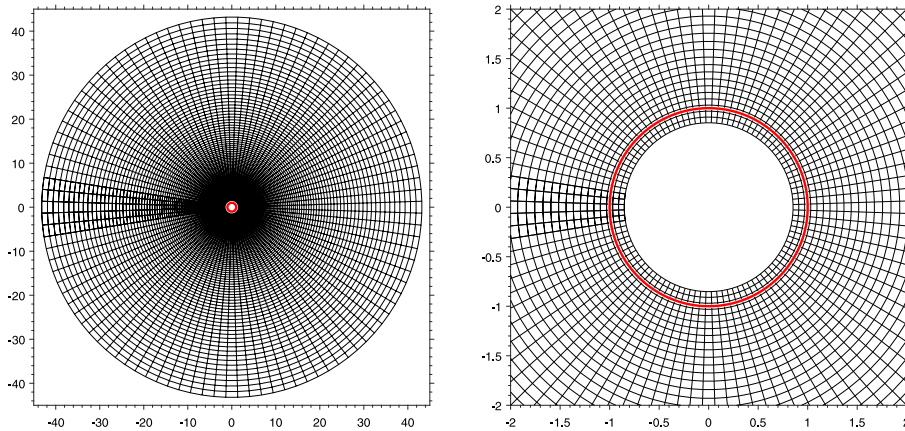
The two-dimensional entropy wave problem [63] simulates the convection of a density wave at a constant free-stream velocity while the pressure remain unchanged. The analytic solution is given by

$$\begin{aligned} \rho(x, y, t) &= \rho_\infty + A \sin[\pi(x + y - (u_\infty + v_\infty)t)] \\ u(x, y, t) &= u_\infty \\ v(x, y, t) &= v_\infty \\ p(x, y, t) &= p_\infty \end{aligned}$$

Simulations are carried out on the two domains  $\mathcal{D}_S$  and  $\mathcal{D}_N$ , already presented in Fig. 10, with  $A = 0.2$ ,  $\rho_\infty = 1$ ,  $u_\infty = 1$ ,  $v_\infty = 1$  and  $p_\infty = 1$  until the final time  $t = 0.5$ . Dirichlet BC is prescribed and the reconstruction is performed using the conservative variables. We report in Table 7 the errors and convergence rates for the two geometries with different grids and the full fifth order of convergence is obtained with the three norms. The Dirichlet condition given on the physical domain is perfectly well translated into ghost cell values and allows to achieve the optimal order even if the boundary condition is not located at points of the computational grid.

**Table 7**  
Results for the 2D entropy wave.

Mesh	$\mathcal{D}_S$						$\mathcal{D}_N$					
	$L_1$	$L_2$	$L_\infty$	$r_1$	$r_2$	$r_\infty$	$L_1$	$L_2$	$L_\infty$	$r_1$	$r_2$	$r_\infty$
25	2.24E-04	4.01E-04	1.61E-03	–	–	–	2.21E-04	3.35E-04	1.11E-03	–	–	–
50	4.25E-06	6.89E-06	3.50E-05	5.72	5.86	5.52	4.58E-06	7.04E-06	4.44E-05	5.60	5.57	4.65
100	1.41E-07	2.47E-07	1.15E-06	4.91	4.80	4.93	1.56E-07	2.42E-07	1.29E-06	4.88	4.87	5.10
200	4.11E-09	6.70E-09	3.36E-08	5.11	5.20	5.09	5.22E-09	9.04E-09	8.06E-08	4.90	4.74	4.01
400	1.28E-10	2.40E-10	3.21E-09	5.01	4.80	3.39	1.55E-10	2.79E-10	2.96E-09	5.07	5.02	4.77
800	4.02E-12	7.30E-12	8.14E-11	4.99	5.04	5.30	5.03E-12	9.31E-12	1.01E-10	4.95	4.91	4.88



**Fig. 14.** 2D subsonic flow around a circular cylinder. Complete mesh (left) and detail near the cylinder, marked with a red thicker line (right). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

**6.3.2. 2D subsonic flow around a circular cylinder**

The proposed technique is adapted to curvilinear coordinates where the computational space is a Cartesian equispaced grid with  $\Delta\xi$  and  $\Delta\eta$  increments in the  $\xi$  and  $\eta$  directions. The only difference with the methodology presented so far is that one has to account for the periodicity in the overlapping direction by populating the ghost points with the corresponding values from the other side of the cut, and extending the previously explained techniques to take them into account.

We consider a curvilinear grid made up of concentric circumferences and straight lines emanating from the center of the cylinder, placed at the origin of coordinates. A uniform distribution of points along the circumferences is set, while an exponential distribution of the form  $\xi(x) = \frac{e^{\beta x} - 1}{e^\beta - 1}$  is used to locate the points on the radii. The value of  $\beta$  is set to 3 and it controls the amount of clustering near the cylinder wall,  $x$  being a uniform distribution of points between 0 and 1. This way, more points are placed close to the cylinder in order to bound the entropy errors. The mesh displayed in Fig. 14 is made up of 100 points on each direction.

The problem is also solved with a finite difference adaptation of the curvature corrected symmetry technique (CCST) [59]. Supposing rectilinear wall boundaries, Eq. (11) can be used, since it is a simplification of Eq. (9) with zero curvature ( $\kappa = 0$ ). Applying Eq. (11) when the curvature is different than zero, however, leads to undesired entropy growths around the wall because the curved boundary is modeled as piece-wise rectilinear. The CCST technique takes into account the curvature in the pressure equation, and correctly formulates the entropy preservation for the calculation of the density, see Eq. (12), but the tangential velocity is approximated as the tangential velocity component of the nearest cell to the wall. This causes all the other variables, that rely on the approximation of the tangential velocity, to be approximated only to second order of accuracy. This drawback is overcome by the

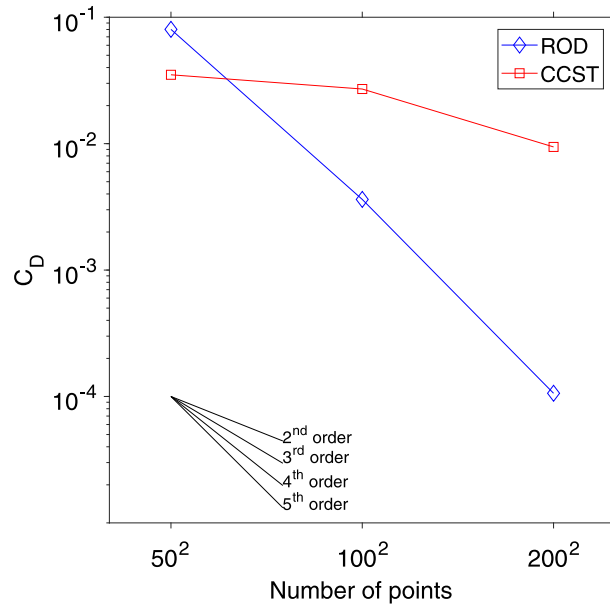


Fig. 15. 2D subsonic flow around a cylinder. Convergence rate of drag coefficient  $C_D$ .

proposed approach as it will be shown in Fig. 15.

$$\left\{ \begin{array}{l} u_n = 0 \\ \frac{\partial \rho}{\partial \mathbf{n}} = 0 \\ \frac{\partial p}{\partial \mathbf{n}} = 0 \\ \frac{\partial u_\tau}{\partial \mathbf{n}} = 0 \end{array} \right. \quad (11)$$

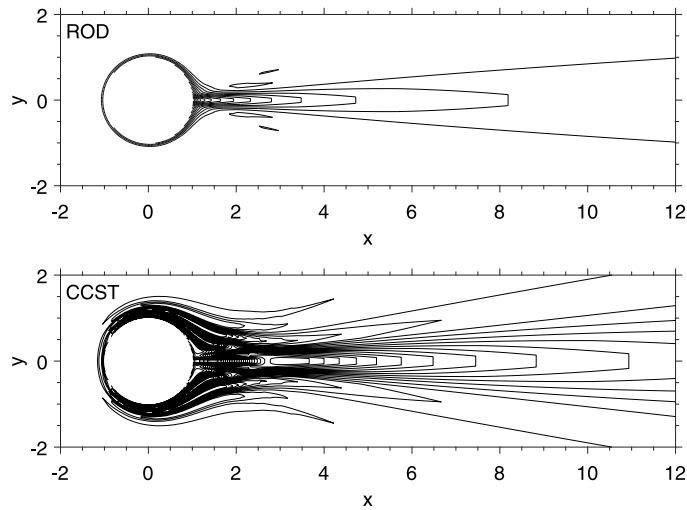
$$\left\{ \begin{array}{l} u_n = 0 \\ \frac{\partial \rho}{\partial \mathbf{n}} = \left( \frac{\rho}{\gamma p} \right) \frac{\partial p}{\partial \mathbf{n}} \\ \frac{\partial p}{\partial \mathbf{n}} = -\rho u_\tau^2 \kappa \\ \frac{\partial u_\tau}{\partial \mathbf{n}} = 0 \end{array} \right. \quad (12)$$

Three layers of ghost points have been added in order to use the WENO 5 scheme. It is important to notice that given the exponential nature of the grid stretching, the ghost points do not lie in the exact symmetrical position of the corresponding real points. As we shall see, this is not a problem for the ROD technique but for the traditional mirror technique, a Lagrange interpolation has to be used in order to locate the mirror points, otherwise we would be using a very low order piece-wise constant interpolation.

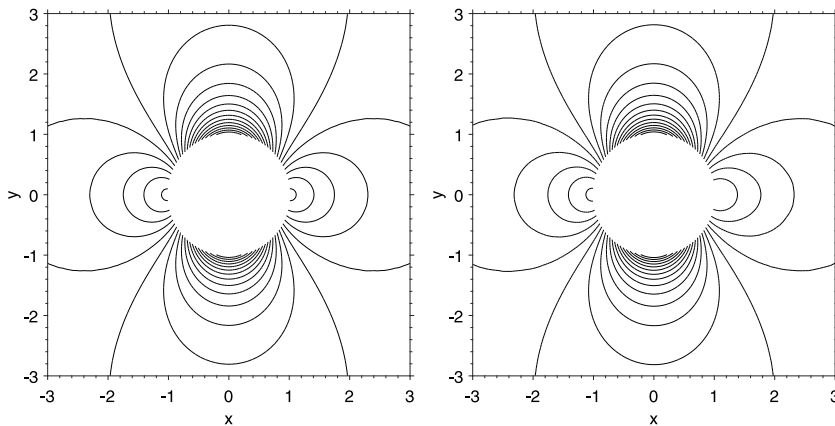
Since the drag coefficient of the exact solution computed along the surface of the cylinder is zero, we use it to estimate the convergence rates of the schemes. The drag coefficient is given by

$$C_D = \frac{1}{2} \int_0^{2\pi} \frac{p(\theta) - p_\infty}{\frac{1}{2} \rho_\infty V_\infty^2} \cos \theta d\theta.$$

where  $\rho_\infty = 1.0$ ,  $V_\infty = 0.1$  and  $p_\infty = 1/\gamma$  are the density, velocity modulus and pressure at the free-stream. The mach number for this test case is 0.1. Fig. 15 shows that the order of accuracy of the ROD BC is fifth-order, while the CCST BC technique provides second-order error of accuracy.



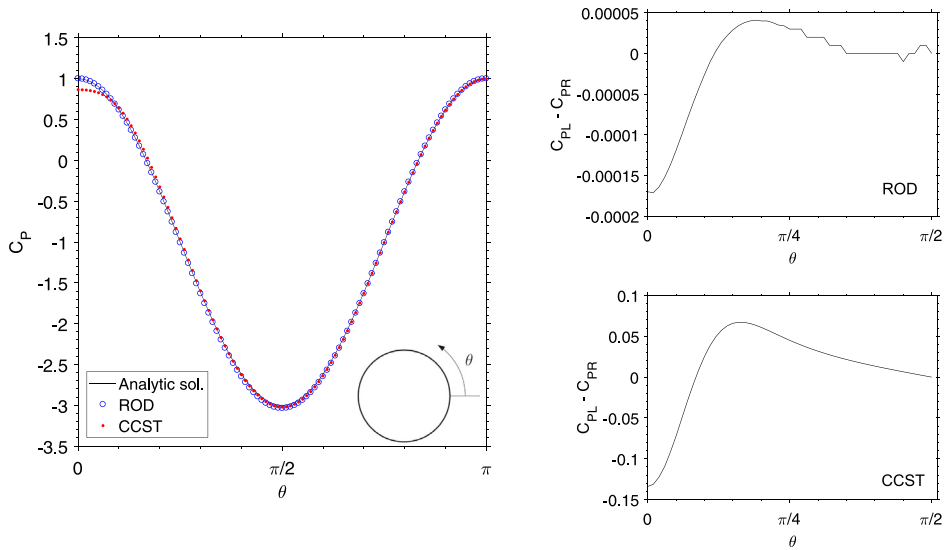
**Fig. 16.** 2D subsonic flow around a cylinder. 30 equispaced produced entropy contours from  $-7.00E-6$  to  $7.00E-6$ . ROD results (top) and CCST results (bottom) for a  $200 \times 200$  curvilinear grid.



**Fig. 17.** 2D subsonic flow around a cylinder. 30 equispaced density contours from 0.98 to 1.01. ROD results (left) and CCST results (right) for a  $200 \times 200$  curvilinear grid.

The way the CCST technique handles the boundary condition (12), although it is an improvement with respect to the approach given by Eq. (11), it generates a spurious entropy growth on the rear part of the cylinder that is convected downstream, as seen in Fig. 16, and makes the solution impossible to converge to a proper steady-state. The ROD technique, in turn, accounts for a more accurate entropy preservation, since it is embedded in the way the boundary conditions are calculated. This is the reason why the ROD solution in Fig. 17 appears more left–right symmetric, while the CCST cannot achieve this condition. This is apparent in Fig. 18 where a small undershoot of the CCST scheme can be seen at  $\theta = 0$ . Conversely, the ROD technique achieves a solution closer to the analytic one.

This test case illustrates the importance of modeling the boundary conditions in a physically correct and numerically accurate way to achieve high-order results. Using the new solid wall boundary condition proposed in Eqs. (9) and (10) correctly models the physical behavior of the test case. Moreover, using the proposed methodology we achieve high-order accuracy. On the other hand, the CCST technique constitutes an extension of the traditional mirror technique of Eq. (11) to impose the solid wall boundary condition in the form given by Eq. (12). However, it fails to correctly account for the curvature when obtaining the tangential velocity at the wall. This leads to excessive entropy generation.



**Fig. 18.** 2D subsonic flow around a cylinder. Obtained  $C_p$  distribution for surface for a  $200 \times 200$  curvilinear grid (left). Since the analytic solution is symmetric around  $\pi/2$ , on the right the  $C_p$  difference between the left half ( $C_{pL}$ ) and the right half ( $C_{pR}$ ) of the plot is plotted for both schemes. Please note that both plots on the right do not share the same vertical scale.

### 6.3.3. Simple wave hitting a steady disk

The benchmark proposed in [13] consists in a simple wave propagating around a steady disk in a constant medium. The disk has radius equal to 0.1 and its center is placed in  $(x_c, y_c) = (0.6, 0.5)$  and the initial conditions for  $(\rho, u, v, p)$  are

$$(\rho, u, v, p) = \begin{cases} (\tilde{\rho}, \tilde{u}, \tilde{v}, \tilde{p}) & \text{if } x < 0.6 \\ (1, 0, 0, 1) & \text{otherwise.} \end{cases}$$

The values of the primitive variables at free-stream are  $\rho_0 = 1$ ,  $p_0 = 1$  and  $c_0 = \sqrt{\gamma}$ . A Gaussian wave that travels to the right is set as

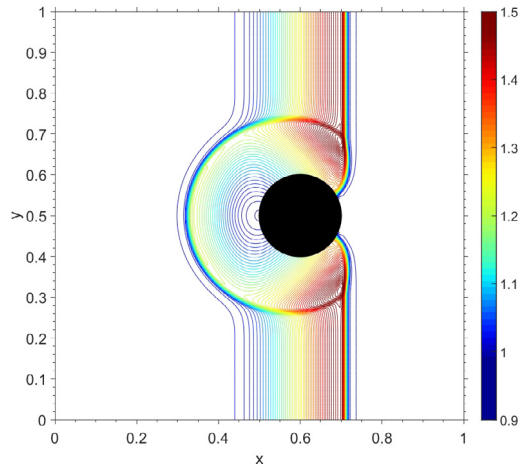
$$\begin{aligned} \tilde{\rho}(x, y) &= \rho_0 \left( 1 + \frac{\gamma - 1}{2c_0} \tilde{u}(x, y) \right)^{\frac{2}{\gamma-1}}, \\ \tilde{u}(x, y) &= \frac{1}{2} e^{-\frac{(x - 0.35)^2}{0.005}}, \\ \tilde{v}(x, y) &= 0, \\ \tilde{p}(x, y) &= p_0 \left( \frac{\tilde{\rho}(x, y)}{\rho_0} \right)^\gamma. \end{aligned}$$

The ROD procedure is only applied on the disk while simple extrapolation is used on the outer square boundary since the wave does not interact with the boundaries during the simulation, which is carried out until the final time  $t = 0.2$ . Fig. 19 shows that the fifth order ROD technique achieves a smooth solution without any spurious wiggles, that have been reported by other authors [13].

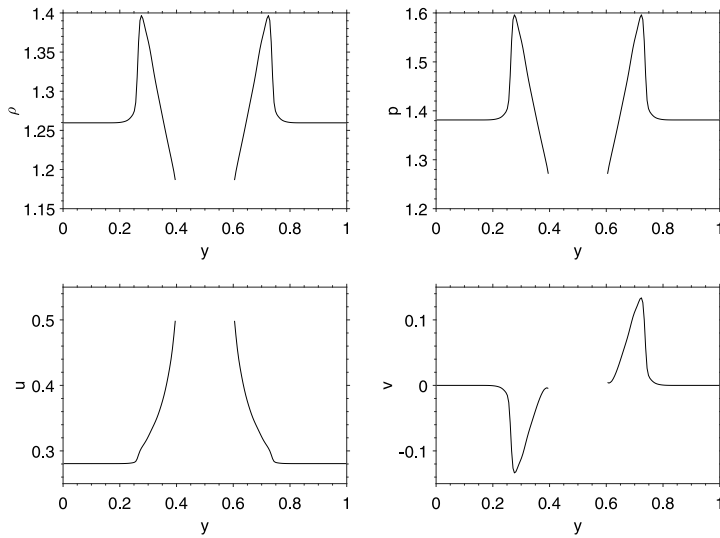
In Figs. 20 and 21, plots along the lines  $y = 0.5$  and  $x = 0.6$  of the primitive variables  $\rho, u, v, p$  are presented.

### 6.3.4. Subsonic Ringleb's flow

Ringleb's flow is a hodograph solution to the Euler equations that is used to assess the accuracy of an Euler code, as in [64]. This test case is presented here because it has an analytic solution and it takes place in a domain limited by non-trivial boundaries. In previous examples the level-set approach explained in epigraph 3.1.1 has been



**Fig. 19.** Simple wave hitting a steady disk. 60 equispaced density contours from 0.98 to 1.59. ROD results for a  $200 \times 200$  Cartesian grid.



**Fig. 20.** Simple wave hitting a steady disk. Primitive variables plot for  $x = 0.6$ .

used to describe the boundaries, but for this case, the mathematical description of the boundaries does not allow to use this approach, so the parametric approach of 3.1.2 is used instead.

For each point we know the velocity’s modulus  $V$  and its angle  $\theta$ . We can obtain the streamline coordinate  $\psi = (\sin \theta)/V$ , and for each pair  $(V, \psi)$ , we can obtain the  $(x, y)$  coordinates as:

$$x(V, \psi) = \frac{1}{\rho} \left( \frac{1}{2V^2} - \psi^2 \right) + \frac{J}{2},$$

$$y(V, \psi) = \pm \frac{\psi}{\rho V} \sqrt{1 - V^2 \psi^2},$$

with

$$J = \frac{1}{c} + \frac{1}{3c^3} + \frac{1}{5c^5} - \frac{1}{2} \ln \left( \frac{1+c}{1-c} \right),$$

$$c = \sqrt{1 - \frac{\gamma - 1}{2} V^2}.$$

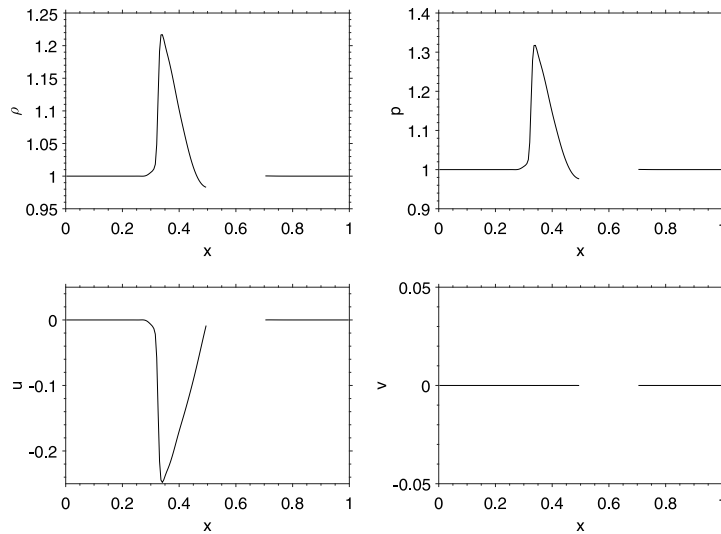


Fig. 21. Simple wave hitting a steady disk. Primitive variables plot for  $y = 0.5$ .

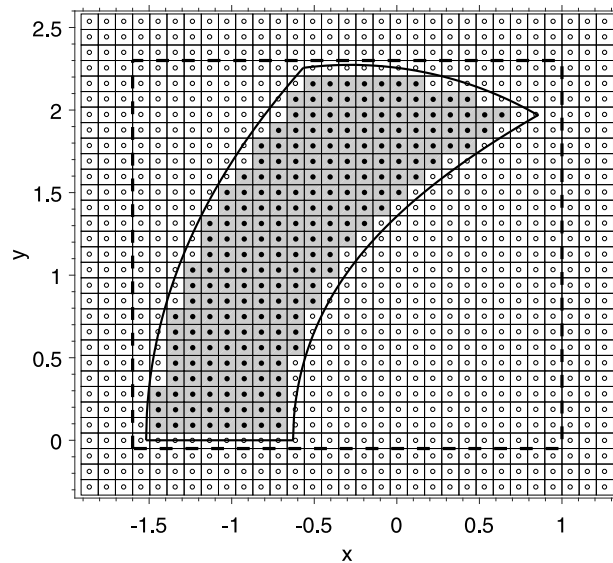


Fig. 22. Subsonic Ringleb's flow. Domain setup, the original domain  $\Lambda$  is enclosed within a dashed line, outside that region the ghost layers can be seen. The original domain  $\Lambda$  is enclosed within a dashed line, outside that region the ghost layers can be seen.  $25 \times 25$  grid.

In [63] a detailed way of calculating the analytic solution for a generic point  $(x, y)$  using a fixed-point algorithm is explained. The primitive variables can be expressed in terms of the hodographic variables  $(V, \psi)$  as

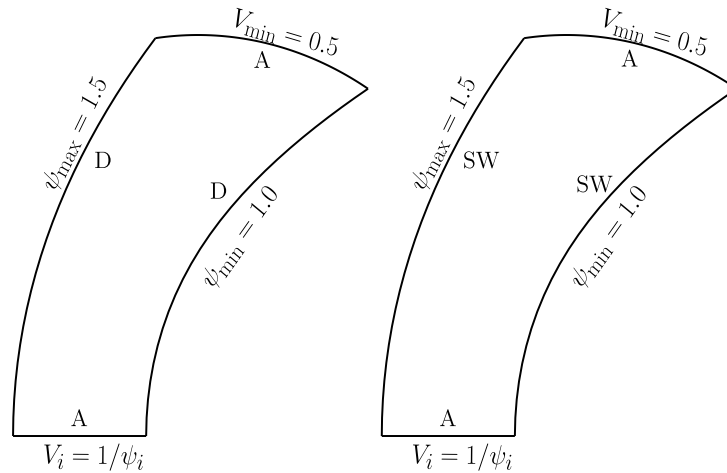
$$\rho = c^{\frac{2}{\gamma-1}}, \quad u = -V \cos \theta, \quad v = -V \sin \theta, \quad p = \rho \frac{c^2}{\gamma}.$$

We use a Cartesian mesh over the domain  $[-1.6, 1.0] \times [-0.05, 2.3]$  that covers a part of the subsonic domain of the Ringleb's flow detailed in Fig. 22. For this test case, a three-layer ghost region is used.

Two different configurations are used for this test case.

- (a) The analytic solution is used on the top and bottom borders while the ROD procedure with Dirichlet BC is used on the left and right borders.





**Fig. 23.** Subsonic Ringleb's flow. Domain configurations. Configuration (a) uses analytic solution on the top and bottom borders while ROD reconstruction with Dirichlet BC on the left and right borders (left). Configuration (b) uses analytic solution on the top and bottom borders while ROD reconstruction with Solid Wall BC on the left and right borders (right).

(b) The analytic solution on the top and bottom borders while the ROD reconstruction with a Solid Wall BC on the left and right borders.

The setup can be seen in Fig. 23 while in Fig. 22 the calculable domain is plotted for a  $25 \times 25$  grid. We remark that on the inflow and outflow boundaries, which correspond to the top and bottom boundaries respectively, the exact analytic solution will be imposed, *i.e.* the ROD procedure is not used for these boundaries.

We have a biparametric definition of the domain, where the right and left boundaries correspond to a constant streamline  $\psi_{\min} = 1.0$  and  $\psi_{\max} = 1.5$  respectively. The top boundary is the isotach corresponding to  $V = V_{\min}$ , and the bottom boundary is the line  $y = 0$ , which is a symmetry axis. With this choice of values, the simulation takes place entirely in a subsonic part of the flow.

At the boundaries an equal point distribution can be obtained using the procedure explained in Eq. (2), since now the equations depend on a single parameter  $V$ .

In order to obtain a smooth distribution of normal and tangent vectors without a level-set function, we take advantage of the fact that we can obtain the  $(V, \psi)$  coordinates from any  $(x, y)$  pair (also detailed in [63]). Since the walls are located on streamlines with constant value of  $\psi$ , we obtain the unitary normal and tangent vectors (and the signed curvature in the case of the boundaries) using  $V$  as the only parameter while using the corresponding value of  $\psi$  for each point. Once we have obtained a set of normal and tangent unitary vectors, the calculations of the normal and tangent velocities is performed.

We start from the steady-state solution at every point, and carry out the simulation until  $t = 0.5$ . We present in Table 8 the errors and convergence rates for the different configurations. For this test case, we obtain the optimal order of accuracy for configurations (a) and (b) while Fig. 24 displays the isovalues of the pressure and a density color map.

For both configurations satisfactory results are obtained for this test case as well, showing that by using a parametric description of the boundary the ROD procedure is still able to recover the expected order of accuracy.

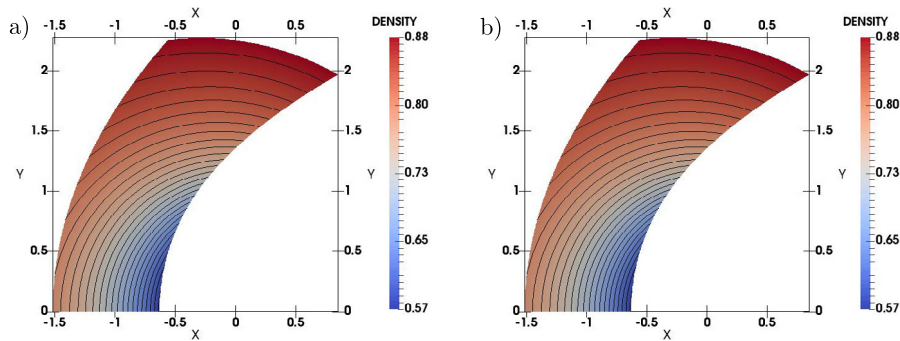
## 7. Conclusions

In this work we have presented a methodology to obtain high-order reconstructions on curved boundaries employing simple Cartesian meshes. A new way to impose the boundary conditions is presented, via the use of ghost points with constrained least squares polynomial fitting. The methodology proposed in this work could be extended to unstructured grids using the framework presented in [54,55]. Moreover, this methodology can be easily extended to systems of conservation laws such as the Euler equations and to arbitrarily distributed curvilinear grids, if necessary. In the case of the Euler equations, a new technique to impose the Solid Wall BC has also been explained

**Table 8**

Results for the Subsonic Ringleb's flow.

Mesh	Configuration (a)						Configuration (b)					
	$L_1$	$L_2$	$L_\infty$	$r_1$	$r_2$	$r_\infty$	$L_1$	$L_2$	$L_\infty$	$r_1$	$r_2$	$r_\infty$
25	5.51E-05	9.69E-05	3.93E-04	–	–	–	3.19E-05	4.69E-05	1.48E-04	–	–	–
50	1.99E-06	4.24E-06	3.19E-05	4.79	4.51	3.62	2.99E-06	6.16E-06	4.02E-05	3.42	2.93	1.88
100	9.10E-08	1.90E-07	1.40E-06	4.45	4.48	4.51	1.08E-07	2.18E-07	1.45E-06	4.79	4.82	4.80
200	2.27E-09	4.88E-09	4.98E-08	5.32	5.28	4.81	3.31E-09	6.56E-09	5.08E-08	5.03	5.05	4.83
400	9.50E-11	1.98E-10	1.83E-09	4.58	4.63	4.77	1.23E-10	2.50E-10	2.51E-09	4.75	4.71	4.34
800	2.83E-12	5.82E-12	7.58E-11	5.07	5.09	4.59	3.74E-12	7.55E-12	7.93E-11	5.04	5.05	4.98

**Fig. 24.** Subsonic Ringleb's flow. 30 equispaced pressure contours from 0.30 to 0.60. Results for configuration (a) (left) and (b) (right) for a  $100 \times 100$  grid.

and it has been proved that for smooth flows, arbitrary high order can be achieved, even for borders of variable curvature.

The extension of this methodology to the Navier–Stokes equations will be the subject of a future work.

## Acknowledgments

This work has been partially supported by the Ministerio de Economía y Competitividad (grant #DPI2015-68431-R) and #RTI2018-093366-B-I00 of the Ministerio de Ciencia, Innovación y Universidades of the Spanish Government and by the Consellería de Educación e Ordenación Universitaria of the Xunta de Galicia (grants #GRC2014/039 and #ED431C 2018/41), cofinanced with FEDER, Spain funds and the Universidade da Coruña, Spain. J. Fernández-Fidalgo gratefully acknowledges the contributions of the IACOBUS Program, Spain and the INDITEX-UDC, Spain grant that have partially financed the present work. S. Clain acknowledges the financial support by FEDER – Fundo Europeu de Desenvolvimento Regional, Portugal, through COMPETE 2020 – Programa Operacional Fatores de Competitividade, and the National Funds through FCT — Fundação para a Ciência e a Tecnologia, Portugal, project No. UID/FIS/04650/2013 and project No. POCI-01-0145-FEDER-028118.

## References

- [1] T. Nonomura, D. Terakado, Y. Abe, K. Fujii, A new technique for freestream preservation of finite-difference WENO on curvilinear grid, *Comput. & Fluids* 107 (2015) 242–255.
- [2] A.J. Christlieb, X. Feng, Y. Jiang, Q. Tang, A high-order finite difference WENO scheme for ideal magnetohydrodynamics on curvilinear meshes, *SIAM J. Sci. Comput.* 40 (4) (2018) A2631–A2666.
- [3] M.R. Visbal, D.P. Rizzetta, Large-eddy simulation on curvilinear grids using compact differencing and filtering schemes, *J. Fluids Eng.* 124 (4) (2002) 836.
- [4] M.R. Visbal, D.V. Gaitonde, On the use of higher-order finite-difference schemes on curvilinear and deforming meshes, *J. Comput. Phys.* 181 (1) (2002) 155–185.
- [5] Y. Jiang, C.W. Shu, M. Zhang, Free-stream preserving finite difference schemes on curvilinear meshes, *Methods Appl. Anal.* 21 (1) (2014) 1–30.
- [6] J.F. Thompson, *Numerical Grid Generation: Foundations and Applications*, Elsevier Science Pub. Co., 1985.

- [7] D. Kopriva, Metric identities and the discontinuous spectral element method on curvilinear meshes, *J. Sci. Comput.* 26 (3) (2006) 301–327.
- [8] C. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* 25 (1977) 220–252.
- [9] D. Goldstein, R. Handler, L. Sirovich, Modeling a no-slip flow boundary with an external force field, *J. Comput. Phys.* 105 (1993) 354–366.
- [10] E.M. Saiki, S. Biringen, Numerical simulation of a cylinder in uniform flow: application of a virtual boundary method, *J. Comput. Phys.* 123 (1996) 450–465.
- [11] P. Ouro, L. Cea, L. Ramírez, X. Nogueira, An immersed boundary method for unstructured meshes in depth averaged shallow water models, *Internat. J. Numer. Methods Fluids* 81 (11) (2015) 672–688.
- [12] R. Boukharfane, F.H.E. Ribeiro, Z. Bouali, A. Mura, A combined ghost-point-forcing / direct-forcing immersed boundary method (IBM) for compressible flow simulations, *Comput. & Fluids* 162 (2018) 91–112.
- [13] A. Chertock, A. Coco, A. Kurganov, G. Russo, A Second-order finite-difference method for compressible fluids in domains with moving boundaries, in: *Communications in Computational Physics, Commun. Comput. Phys.* 23 (2018) 230–263.
- [14] H.S. Udaykumar, R. Mittal, W. Shyy, Solid-liquid phase front computations in the sharp interface limit on fixed grids, *J. Comput. Phys.* 153 (1999) 535–574.
- [15] D.B. Stein, R.D. Guy, B. Thomases, Immersed boundary smooth extension: A high-order method for solving PDE on arbitrary smooth domains using fourier spectral methods, *J. Comput. Phys.* 304 (2016) 252–274.
- [16] D.B. Stein, R.D. Guy, B. Thomases, Immersed Boundary Smooth Extension (IBSE): A high-order method for solving incompressible flow on arbitrary smooth domains, *J. Comput. Phys.* 335 (2017) 155–178.
- [17] P. Huang, H. Wu, Y. Xiao, An unfitted interface penalty finite element method for elliptic interface problems, *Comput. Methods Appl. Mech. Engrg.* 323 (2017) 439–460.
- [18] N. Gokhale, N. Nikiforakis, R. Klein, A dimensionally split Cartesian cut cell method for hyperbolic conservation laws, *J. Comput. Phys.* 364 (2018) 186–208.
- [19] B. Muralidharan, S. Menon, A high-order adaptive Cartesian cut-cell method for simulation of compressible viscous flow over immersed bodies, *J. Comput. Phys.* 321 (2016) 342–368.
- [20] W.G. Dettmer, C. Kadapa, D. Perić, A stabilised immersed boundary method on hierarchical b-spline grids, *Comput. Methods Appl. Mech. Engrg.* 311 (2016) 415–437.
- [21] R. Fedkiw, A. Marquina, B. Merriman, An isobaric fix for the overheating problem in multimaterial compressible flows, *J. Comput. Phys.* 148 (1999) 545–578.
- [22] F. Gibou, R.P. Fedkiw, L.T. Cheng, M. Kang, A second-order accurate symmetric discretization of the Poisson equation on irregular domains, *J. Comput. Phys.* 176 (1999) 205–227.
- [23] R. Ghias, R. Mittal, H. Dong, A sharp interface immersed boundary method for compressible viscous flows, *J. Comput. Phys.* 225 (1) (2007) 528–553.
- [24] M.D. de Tullio, P.D. Palma, G. Iaccarino, G. Pascazio, M. Napolitano, An immersed boundary method for compressible flows using local grid refinement, *J. Comput. Phys.* 225 (2) (2007) 2098–2117.
- [25] P.D. Palma, M.D. de Tullio, G. Pascazio, M. Napolitano, An immersed-boundary method for compressible viscous flows, *Comput. & Fluids* 35 (7) (2006) 693–702.
- [26] C. Brehm, C. Hader, H.F. Fasse, A locally stabilized immersed boundary method for the compressible Navier-Stokes equations, *J. Comput. Phys.* 295 (2015) 475–504.
- [27] Y. Qu, R. Shi, R.C. Batra, An immersed boundary formulation for simulating high-speed compressible viscous flows with moving boundary, *J. Comput. Phys.* 357 (2018) 672–691.
- [28] H.S. Udaykumar, R. Mittal, P. Rampunggoon, A. Khanna, A sharp interface cartesian grid method for simulating flows with complex moving boundaries, *J. Comput. Phys.* 174 (2001) 345–380.
- [29] Y.H. Tseng, J.H. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, *J. Comput. Phys.* 192 (2) (2003) 593–623.
- [30] R. Mittal, H. Dong, M. Bozkurttas, F.M. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flow with complex boundaries, *J. Comput. Phys.* 227 (2008) 4825–4852.
- [31] M.D. Griffith, J.S. Leontini, Sharp interface immersed boundary methods and their application to vortex-induced vibration of a cylinder, *J. Fluids Struct.* 72 (2017) 38–58.
- [32] D.C. Lo, C.P. Lee, I.F. Lin, An efficient immersed boundary method for fluid flow simulations with moving boundaries, *Appl. Math. Comput.* 328 (2018) 312–337.
- [33] A. Mark, E. Svenning, F. Edelvik, An immersed boundary method for simulation of flow with heat transfer, *Int. J. Heat Mass Transfer* 56 (2013) 424–435.
- [34] J. Xia, K. Luo, J. Fan, A ghost-cell based high-order immersed boundary method for inter-phase heat transfer simulation, *Int. J. Heat Mass Transfer* 75 (2014) 302–312.
- [35] J. Xia, K. Luo, J. Fan, Simulating heat transfer from moving rigid bodies using high-order ghost-cell based immersed-boundary method, *Int. J. Heat Mass Transfer* 89 (2015) 856–865.
- [36] K. Luo, Z. Zhuang, J. Fan, N.E.L. Haugen, A ghost-cell immersed boundary method for simulations of heat transfer in compressible flows under different boundary conditions, *Int. J. Heat Mass Transfer* 92 (2016) 708–717.
- [37] A. Coco, G. Russo, Second order finite-difference ghost-point multigrid methods for elliptic problems with discontinuous coefficients on an arbitrary interface, *J. Comput. Phys.* 361 (2018) 299–330.
- [38] J.H. Seo, R. Mittal, A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations, *J. Comput. Phys.* 230 (2011) 7347–7363.

- [39] S.M.H. Karimian, M. Ardakani, Immersed boundary method for the solution of 2D inviscid compressible flow using finite volume approach on moving cartesian grid, *J. Appl. Fluid Mech.* 4 (2) (2011) 27–36.
- [40] F. Gibou, R. Fedwik, A fourth order accurate discretization for the Laplace Heat equation on arbitrary domains with application to Stephan problem, *J. Comput. Phys.* 202 (2005) 577–601.
- [41] J.H. Seo, R. Mittal, A high-order immersed boundary method for acoustic wave scattering and low-Mach number flow-induced in complex geometries, *J. Comput. Phys.* 230 (2011) 1000–1019.
- [42] H.X. Luo, R. Mittal, X. Zheng, S.A. Bielałowicz, R.J. Walsh, J.K. Hahne, An immersed-boundary method for flow–structure interaction in biological systems with application to phonation, *J. Comput. Phys.* 227 (9303–9332) (2008).
- [43] B. Sjögren, N.A. Petersson, A Cartesian embedded boundary method for hyperbolic conservation laws, *Commun. Comput. Phys.* 2 (2007) 1199–1219.
- [44] A. Baeza, P. Mulet, D. Zorío, High order boundary extrapolation technique for finite difference methods on complex domains with cartesian meshes, *J. Sci. Comput.* 66 (2) (2015) 761–791.
- [45] A. Baeza, P. Mulet, D. Zorío, High order weighted extrapolation for boundary conditions for finite difference methods on complex domains with cartesian meshes, *J. Sci. Comput.* 69 (1) (2016) 170–200.
- [46] S. Tan, C.W. Shu, Inverse Lax-Wendroff procedure for numerical boundary conditions of conservation laws, *J. Comput. Phys.* 229 (21) (2010) 8144–8166.
- [47] S. Tan, C.W. Shu, A high order moving boundary treatment for compressible inviscid flows, *J. Comput. Phys.* 230 (15) (2011) 6023–6036.
- [48] S. Tan, C. Wang, C.-W. Shu, J. Ning, Efficient implementation of high order inverse Lax–Wendroff boundary treatment for conservation laws, *J. Comput. Phys.* 231 (6) (2012) 2510–2527.
- [49] S. Tan, C.W. Shu, Inverse Lax-Wendroff procedure for numerical boundary conditions of hyperbolic equations: survey and new developments, *Adv. Appl. Math. Model. Comput. Sci.* (2013) 41–63.
- [50] X. Zeng, C. Farhat, A systematic approach for constructing higher-order immersed boundary and ghost fluid methods for fluid–structure interaction problems, *J. Comput. Phys.* 231 (7) (2012) 2892–2923.
- [51] C. Wang, J. Ding, S. Tan, W. Han, High order numerical simulation of detonation wave propagation through complex obstacles with the inverse Lax-Wendroff treatment, *Commun. Comput. Phys.* 18 (5) (2015) 1264–1281.
- [52] F. Vilar, C.W. Shu, Development and stability analysis of the inverse lax-wendroff boundary treatment for central compact schemes, *ESAIM Math. Modelling Numer. Anal.* 49 (1) (2015) 39–67.
- [53] G. Dakin, B. Després, S. Jaouen, Inverse Lax-Wendroff boundary treatment for compressible Lagrange-remap hydrodynamics on Cartesian grids, *J. Comput. Phys.* 353 (2018) 228–257.
- [54] R. Costa, S. Clain, R. Loubère, G.J. Machado, Very high-order accurate finite volume scheme on curved boundaries for the two-dimensional steady-state convection–diffusion equation with Dirichlet condition, *Appl. Math. Model.* 54 (2018) 752–767.
- [55] R. Costa, J.M. Nóbrega, S. Clain, G.J. Machado, R. Loubère, Very high-order accurate finite volume scheme for the convection-diffusion equation with general boundary conditions on arbitrary curved boundaries, *Internat. J. Numer. Methods Engrg.* 117 (2) (2018) 188–220.
- [56] E. Hartmann, On the curvature of curves and surfaces defined by normalforms, *Comput. Aided Geom. Design* 16 (5) (1999) 355–376.
- [57] E. Anderson, Z. Bai, J. Dongarra, Generalized QR factorization and its applications, *Linear Algebra Appl.* 162–164 (1992) 243–271.
- [58] W. Gander, Algorithms for the QR-decomposition, *Res. Rep* 80 (2) (1980) 1251–1268, <https://inf.ethz.ch/personal/gander/papers/qrneu.pdf>.
- [59] Z.J. Wang, Y. Sun, Curvature-based wall boundary condition for the euler equations on unstructured grids, *Amer. Inst. Aeronaut. Astronaut. J.* 41 (1) (2003) 27–33.
- [60] G.S. Jiang, C.W. Shu, Efficient implementation of weighted ENO schemes, *J. Comput. Phys.* 126 (1) (1996) 202–228.
- [61] A.K. Henrick, T.D. Aslam, J.M. Powers, Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points, *J. Comput. Phys.* 207 (2) (2005) 542–567.
- [62] D. Ghosh, J.D. Baeder, Compact reconstruction schemes with weighted ENO limiting for hyperbolic conservation laws, *SIAM J. Sci. Comput.* 34 (3) (2012) A1678–A1706.
- [63] K. Masatsuka, *I do like CFD*, Vol. 1, second ed., K. Masatsuka, 2013, v. 1.
- [64] W.J. Coirier, K.G. Powell, An accuracy assessment of Cartesian-mesh approaches for the Euler equations, *J. Comput. Phys.* 117 (1) (1995) 121–131.